

QUICKLOCKER-DELIVERY



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

QUICKLOCKER-DELIVERY



UNIVE This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Software Development) with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

QUICKLOCKER-DELIVERY

is written by me and is my own effort and that no part has been plagiarized

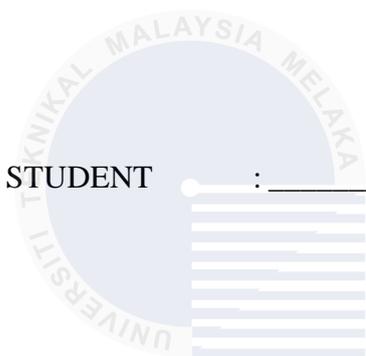
without citations.

STUDENT

:


(NABIL AQMAR BIN ZUHAIMI)

Date : 4 September 2024



اونيورسي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Software Development) with Honours.

SUPERVISOR

:


(DR. NOR HAFEIZAH HASSAN)

Date : 4 September 2024

DEDICATION

This project report is dedicated to all the wonderful people who have supported me throughout this journey. To my family and friends, thank you for always being there for me, encouraging me, and understanding me even during tough times. Your unwavering belief in me has kept me motivated.

A big thank you also goes to my lecturers and mentors. You have taught me so much and inspired me to keep learning. Your guidance and trust in my abilities have made a significant impact on my progress.

Lastly, I want to acknowledge my own hard work and determination. This project took a lot of effort, and I am proud of the growth I have achieved. Every challenge was a learning experience that contributed to my personal and academic development.

This report is a testament to the collective support I received, and I am deeply grateful for it. Thanks for being there for me and helping me succeed. Your support made all the difference.

ACKNOWLEDGEMENTS

This project could not have been completed without the incredible support I received from many individuals.

Firstly, I extend my deepest gratitude to my project supervisor, Dr. Nor Hafeizah Hassan. Their invaluable guidance and encouragement throughout the project were truly instrumental. Their expertise and constructive feedback significantly improved the quality of my work.

I also want to express my appreciation to my friends and classmates. Their constant support, insightful discussions, and shared ideas were vital in shaping my approach and refining my concepts.

Of course, I must thank my family for their unwavering support. Their understanding, encouragement, and belief in me kept me motivated throughout this journey. I am truly fortunate to have them by my side.

Lastly, I am grateful to all the resources, references, and organizations that provided the information necessary for my research and the completion of this project.

Everyone involved, whether directly or indirectly, contributed to the success of this project, and I want to thank each one of you. Your support, guidance, and contributions were invaluable. Thank you all!

ABSTRACT

In today's fast-paced society, efficient parcel management has evolved into a critical necessity for both businesses and individuals. Traditional delivery methods often struggle with challenges such as delays, security vulnerabilities, limited flexibility, and complex processes. This project focuses on addressing these issues by leveraging technology to streamline and secure the parcel delivery process. The solution involves the development of the QLD Courier System, which integrates advanced technologies such as automation and the Internet of Things (IoT) to enhance operational efficiency and customer satisfaction. The project aligns with the Department of Economic and Social Affairs' emphasis on sustainable development, particularly in creating sustainable cities and communities and fostering industry innovation and infrastructure. The system aims to develop a resilient, inclusive, and resource-efficient urban environment by reducing traffic congestion, enhancing security, and promoting convenience for both residents and businesses. Through innovative infrastructure, the project supports more efficient parcel management, driving broader economic growth and competitiveness. The implementation phase involved setting up the software development environment, configuring management tools, and establishing version control procedures. Testing strategies were adopted to ensure system functionality, security, and performance. The project concludes with an evaluation of the system's strengths and weaknesses, suggestions for improvement, and a summary of its contributions to sustainable urban living and efficient parcel management. This comprehensive approach addresses the current challenges in parcel delivery and contributes to the broader goal of creating smarter, more sustainable cities equipped to meet the demands of modern parcel delivery needs. The results obtained demonstrate significant improvements in delivery efficiency, security, and user satisfaction, validating the effectiveness of the proposed solution.

ABSTRAK

Dalam masyarakat yang pesat hari ini, pengurusan penghantaran barang yang cekap telah menjadi satu keperluan kritikal bagi perniagaan dan individu. Kaedah penghantaran tradisional sering menghadapi cabaran seperti kelewatan, kelemahan keselamatan, fleksibiliti yang terhad, dan proses yang rumit. Projek ini bertujuan untuk menangani isu-isu ini dengan memanfaatkan teknologi untuk memudahkan dan menjamin proses penghantaran barang. Penyelesaiannya melibatkan pembangunan Sistem Kurier QLD, yang mengintegrasikan teknologi canggih seperti automasi dan Internet of Things (IoT) untuk meningkatkan kecekapan operasi dan kepuasan pelanggan. Projek ini selari dengan penekanan Jabatan Hal Ehwal Ekonomi dan Sosial terhadap pembangunan mampan, terutamanya dalam mewujudkan bandar yang mampan dan komuniti serta memupuk inovasi industri dan infrastruktur. Sistem ini bertujuan untuk membangunkan persekitaran bandar yang berdaya tahan, inklusif, dan cekap sumber dengan mengurangkan kesesakan lalu lintas, meningkatkan keselamatan, dan mempromosikan kemudahan untuk penduduk dan perniagaan. Melalui infrastruktur yang inovatif, projek ini menyokong pengurusan barang yang lebih cekap, mendorong pertumbuhan ekonomi dan daya saing yang lebih luas. Fasa pelaksanaan melibatkan penubuhan persekitaran pembangunan perisian, konfigurasi alat pengurusan, dan penubuhan prosedur kawalan versi. Strategi ujian telah diterima pakai untuk memastikan fungsi sistem, keselamatan, dan prestasi. Projek ini disimpulkan dengan penilaian terhadap kekuatan dan kelemahan sistem, cadangan untuk penambahbaikan, dan ringkasan sumbangannya terhadap kehidupan bandar yang mampan dan pengurusan barang yang cekap. Pendekatan komprehensif ini menangani cabaran semasa dalam penghantaran barang dan menyumbang kepada matlamat yang lebih luas untuk mewujudkan bandar yang lebih pintar, lebih mampan yang dilengkapi untuk memenuhi keperluan penghantaran barang moden. Hasil yang diperoleh menunjukkan peningkatan ketara dalam kecekapan penghantaran, keselamatan, dan kepuasan pengguna, mengesahkan keberkesanan penyelesaian yang dicadangkan.

TABLE OF CONTENTS

	PAGE
DECLARATION	II
DEDICATION	III
ACKNOWLEDGEMENTS.....	IV
ABSTRACT	V
ABSTRAK.....	VI
TABLE OF CONTENTS.....	VII
LIST OF TABLES.....	XVI
LIST OF FIGURES	XIX
LIST OF ABBREVIATIONS.....	XXII
LIST OF ATTACHMENTS	XXIII
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction.....	1
1.2 Problem Statement	2
1.2.1 Conclusion	2
1.3 Objectives	3
1. Analyze Issues in College Courier Service Delivery	3
2. Develop an IoT-Based Locker System Using a QR Code for Courier Service Delivery	3
3. Demonstrate the Effectiveness of the System Using a Mobile Application Environment.....	3

1.4	Scope	4
1.5	Project Significance.....	7
1.6	Expected Output.....	8
1.7	Conclusion	8
1.7.1	Next Steps	8
CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY 10		
2.1	Introduction.....	10
2.2	Facts and Findings.....	10
2.2.1	Domain.....	10
2.2.2	Existing System.....	11
2.2.3	Technique.....	15
2.2.3.1	Alternative Approaches:	15
2.2.3.2	Justification for Choosing QR Codes:	15
2.3	Project Methodology	16
2.3.1	Activities in Each Stage:.....	16
2.4	Project Requirements	19
2.4.1	Software Requirement:	19
2.4.2	Hardware Requirement:.....	20
2.4.3	Other Requirement:	21
2.5	Project Schedule and Milestones	22
2.5.1	Hardware Requirement:.....	22
2.5.2	Stage-by-Stage Activities:	22
2.6	Conclusion	24
CHAPTER 3: ANALYSIS25		

3.1	Introduction.....	25
3.2	Problem Analysis	26
3.2.1	Current System Scenario	26
3.2.2	Problem Statement.....	27
3.3	Requirement Analysis	29
3.3.1	Problem Statement.....	29
3.3.2	Functional Requirements	34
3.3.3	Non-Functional Requirements	40
3.3.4	Other Requirements.....	41
3.4	Conclusion.....	44
3.4.1	Requirements and Next Steps	44
3.4.2	Implementation.....	44
3.4.3	Goal	45
	CHAPTER 4: DESIGN.....	46
4.1	Introduction.....	46
4.2	High-Level Design	46
4.2.1	System Architecture	47
4.2.2	User Interface Design	49
4.2.3	Database Design.....	79
4.2.3.1	Conceptual and logical database design	79
4.3	Detailed Design.....	88
4.3.1	Software Design	88
4.3.1.1	Sequence Diagram.....	88
4.3.1.2	State Diagram.....	90

4.3.1.3	UML Diagram	91
4.3.2	Hardware Design	92
4.3.2.1	Key Components and Connections:	92
4.3.2.2	Explanation of the Diagram:	94
4.3.3	Physical Database Design	94
4.4	Conclusion	95
CHAPTER 5: IMPLEMENTATION.....		97
5.1	Introduction.....	97
5.2	Software Development Environment Setup	97
5.2.1	Environment Architecture Diagram	101
5.2.2	Environment Architecture Diagram Description	101
5.2.3	Summary	102
5.3	Software Configuration Management	103
5.3.1	Configuration Environment Setup.....	103
5.3.2	Version Control Procedure	107
5.4	Implementation Status.....	108
5.5	Dynamic Locker Code Management Description.....	112
5.6	IoT Locker Setup	113
5.7	Conclusion	115
CHAPTER 6: TESTING		116
6.1	Introduction.....	116
6.2	Test Plan	116
6.2.1	Test Organization	117
6.2.2	Test Environment	117

6.2.2.1	Environment Setup	117
6.2.2.2	Application Software	118
6.2.2.3	System Software	120
6.2.2.4	System Hardware	120
6.2.3	Test Schedule	121
6.3	Test Strategy	123
6.3.1	Dynamic Testing	124
6.3.2	System Usability Scale (SUS).....	124
6.3.3	Analysis of Issues in University Courier Service Delivery and Evaluation of the QuickLocker-Delivery Project.....	125
6.4	Test Design	126
6.4.1	Test Description	126
6.4.1.1	Web Testing Module	127
6.4.1.1.1	User Login (Web).....	127
6.4.1.1.2	User Registration (Web)	128
6.4.1.1.3	User Information (Web).....	130
6.4.1.1.4	Item Delivery (Web).....	131
6.4.1.1.5	Item Management Report Admin (Web)	133
6.4.1.1.6	Locker Location (Web).....	133
6.4.1.1.7	Locker Information (Web).....	134
6.4.1.1.8	Users Report Information (Web).....	136
6.4.1.1.9	Convert Report to PDF (Web)	136
6.4.1.2	Mobile Testing Module	138

6.4.1.2.1	User Login (Mobile).....	138
6.4.1.2.2	User Registration (Mobile)	139
6.4.1.2.3	User Profile (Mobile).....	140
6.4.1.2.4	Item Delivery (Mobile).....	141
6.4.1.2.5	Item Delivery History (Mobile)	142
6.4.1.2.6	Google Map API (Mobile).....	143
6.4.1.2.7	QR Code Generator (Mobile).....	143
6.4.1.2.8	Notification (Mobile).....	144
6.4.1.2.9	Locker Functionality (Mobile with Arduino)	145
6.4.1.2.10	Item Delivery Report (Mobile).....	148
6.4.1.2.11	Forgot Password (Mobile).....	148
6.4.1.2.12	SQLite Functionality (Mobile).....	150
6.4.1.2.13	Changeable Location in Locker Functionality (Mobile with Arduino).....	151
6.4.2	Test Data.....	152
6.4.2.1	Web Testing Data.....	153
6.4.2.1.1	Test Data for User Login Admin (Web).....	153
6.4.2.1.2	Test Data for User Registration Admin (Web)	155
6.4.2.1.3	Test Data for User Profile (Web)	164
6.4.2.1.4	Test Data for Item Delivery (Web)	167
6.4.2.1.5	Test Data for Item Management Report (Web)	170
6.4.2.1.6	Test Data for Locker Location (Web)	172

6.4.2.1.7	Test Data for Locker Information (Web).....	175
6.4.2.1.8	Test Data for Users Report Information (Web)	179
6.4.2.1.9	Test Data for Convert Report to PDF (Web)	181
6.4.2.2	Mobile Testing Data	184
6.4.2.2.1	Test Data for User Login (Mobile).....	184
6.4.2.2.2	Test Data for User Registration (Mobile)	188
6.4.2.2.3	Test Data for User Profile (Mobile)	193
6.4.2.2.4	Test Data for Item Delivery (Mobile).....	195
6.4.2.2.5	Test Data for Item Delivery History (Mobile)	198
6.4.2.2.6	Test Data for Google Map API (Mobile).....	199
6.4.2.2.7	Test Data for QR Code Generator (Mobile)	200
6.4.2.2.8	Test Data for Notification (Mobile)	202
6.4.2.2.9	Test Data for Locker Functionality (Mobile with Arduino)	204
6.4.2.2.10	Test Data for Item Delivery Report (Mobile).....	210
6.4.2.2.11	Test Data for Forgot Password (Mobile).....	212
6.4.2.2.12	Test Data for SQLite Functionality (Mobile)	215
6.4.2.2.13	Test Data for Changeable Location in Locker Functionality (Mobile with Arduino).....	217
6.5	System Usability Scale	218
6.5.1	Questionnaires for System Usability Scale	218
6.6	Test Results and Analysis	220
6.6.1	Test Result for Dynamic Testing.....	220

6.6.1.1	Test Result for Website.....	220
6.6.1.1.1	Test Result and Analysis for User Login Admin (Web)	220
6.6.1.1.2	Test Result and Analysis for User Registration Admin (Web)....	221
6.6.1.1.3	Test Result and Analysis for User Profile (Web).....	223
6.6.1.1.4	Test Result and Analysis for Item Delivery (Web)	224
6.6.1.1.5	Test Result and Analysis for Item Management Report (Web) ...	226
6.6.1.1.6	Test Result and Analysis for Locker Location (Web).....	226
6.6.1.1.7	Test Result and Analysis for Locker Information (Web)	227
6.6.1.1.8	Test Result and Analysis for Users Report Information (Web) ...	228
6.6.1.1.9	Test Result and Analysis for Convert Report to PDF (Web).....	229
6.6.1.2	Test Result for Mobile	230
6.6.1.2.1	Test Result and Analysis for User Login (Mobile)	230
6.6.1.2.2	Test Result and Analysis for User Registration (Mobile)	231
6.6.1.2.3	Test Result and Analysis for User Profile (Mobile)	233
6.6.1.2.4	Test Result and Analysis for Item Delivery (Mobile)	234
6.6.1.2.5	Test Result and Analysis for Item Delivery History (Mobile).....	235
6.6.1.2.6	Test Result and Analysis for Google Map API (Mobile)	235
6.6.1.2.7	Test Result and Analysis for QR Code Generator (Mobile).....	236
6.6.1.2.8	Test Result and Analysis for Notification (Mobile)	237
6.6.1.2.9	Test Result and Analysis for Locker Functionality (Mobile with Arduino).....	238
6.6.1.2.10	Test Result and Analysis for Item Delivery Report (Mobile)	240

6.6.1.2.11	Test Result and Analysis for Forgot Password (Mobile)	240
6.6.1.2.12	Test Result and Analysis for SQLite Functionality (Mobile)	241
6.6.1.2.13	Test Result and Analysis for Changeable Location in Locker Functionality (Mobile with Arduino)	242
6.6.2	Summary of Recorded Test Case	243
6.6.3	User Usability Testing Result and Analysis	245
6.6.3.1	User Usability Testing Result	245
6.6.3.2	User Usability Testing Analysis and Result	246
6.6.3.2.1	Calculate User Usability Testing	247
6.6.4	Analyze on Issues in College Courier Service Delivery and Opinions on the QLD Project	250
6.6.4.1	Current Manual Delivery System at UTEM	250
6.6.4.2	Proposed QuickLocker-Delivery (QLD) System	250
6.6.4.3	Conclusion and Recommendations	251
6.7	Conclusion	251
CHAPTER 7: PROJECT CONCLUSION		252
7.1	Observation on Weaknesses and Strengths	252
7.2	Propositions for Improvement	252
7.3	Project Contribution	253
7.4	Conclusion	253
REFERENCES		254
APPENDICES		256

LIST OF TABLES

	PAGE
Table 4.1: Courier_details	83
Table 4.2: Item	83
Table 4.3: Item_management	84
Table 4.4: Item_management_status.....	84
Table 4.5: Item_size	84
Table 4.6: Locker	85
Table 4.7: Locker_availability.....	85
Table 4.8: Locker_location	85
Table 4.9: Locker_status.....	86
Table 4.10: Qrcode_delivery.....	86
Table 4.11: Qrcode_recipient	86
Table 4.12: Role	86
Table 4.13: Users.....	87
Table 5.1: Mobile Application Modules	108
Table 5.2: Website Portal Modules	109
Table 5.3: IoT Locker System Module.....	110
Table 5.4: Notification Service Module	111
Table 5.5: Database Module	111
Table 5.6: Deployment and Testing Module	111
Table 5.7: Overall Project Timeline	112
Table 6.1: Test Organization	117
Table 6.2: Application Module	118
Table 6.3: System Software and purpose	120
Table 6.4: System Hardware Description	120
Table 6.5: Test Schedule	122
Table 6.6: User Login (Web)	127
Table 6.7: User Registration (Web).....	128

Table 6.8: User Information (Web)	130
Table 6.9: Item Delivery (Web)	131
Table 6.10: Item Management Report Admin (Web)	133
Table 6.11: Locker Location (Web)	133
Table 6.12: Locker Information (Web)	134
Table 6.13: Users Report Information (Web)	136
Table 6.14: Convert Report to PDF (Web)	136
Table 6.15: User Login (Mobile)	138
Table 6.16: User Registration (Mobile)	139
Table 6.17: User Profile (Mobile)	140
Table 6.18: Item Delivery (Mobile)	141
Table 6.19: Item Delivery History (Mobile)	142
Table 6.20: Google Map API (Mobile)	143
Table 6.21: QR Code Generator (Mobile)	143
Table 6.22: Notification (Mobile)	144
Table 6.23: Locker Functionality (Mobile with Arduino)	145
Table 6.24: Item Delivery Report (Mobile)	148
Table 6.25: Forgot Password (Mobile)	148
Table 6.26: SQLite Functionality (Mobile)	150
Table 6.27: Changeable Location in Locker Functionality (Mobile with Arduino)	151
Table 6.28: Questionnaires for System Usability Scale	218
Table 6.29: Test Result and Analysis for User Login Admin (Web)	220
Table 6.30: User Registration Admin (Web)	221
Table 6.31: Test Result and Analysis for User Profile (Web)	223
Table 6.32: Test Result and Analysis for Item Delivery (Web)	224
Table 6.33: Test Result and Analysis for Item Management Report (Web)	226
Table 6.34: Test Result and Analysis for Locker Location (Web)	226
Table 6.35: Test Result and Analysis for Locker Information (Web)	227
Table 6.36: Test Result and Analysis for Users Report Information (Web)	228
Table 6.37: Convert Report to PDF (Web)	229
Table 6.38: User Login (Mobile)	230
Table 6.39: Test Result and Analysis for User Registration (Mobile)	231
Table 6.40: Test Result and Analysis for User Profile (Mobile)	233

Table 6.41: Test Result and Analysis for Item Delivery (Mobile)	234
Table 6.42: Item Delivery History (Mobile)	235
Table 6.43: Test Result and Analysis for Google Map API (Mobile).....	235
Table 6.44: Test Result and Analysis for QR Code Generator (Mobile)	236
Table 6.45: Test Result and Analysis for Notification (Mobile)	237
Table 6.46: Test Result and Analysis for Locker Functionality (Mobile with Arduino)	238
Table 6.47: Test Result and Analysis for Item Delivery Report (Mobile)	240
Table 6.48: Test Result and Analysis for Forgot Password (Mobile).....	240
Table 6.49: Test Result and Analysis for SQLite Functionality (Mobile).....	241
Table 6.50: Test Result and Analysis for Item Delivery Report (Mobile)	242
Table 6.51: Summary of Recorded Test Case	243
Table 6.52: User Usability Testing Result	245
Table 6.53: Interpretation of the SUS Score	249

LIST OF FIGURES

	PAGE
Figure 2.1: Agile Diagram	16
Figure 2.2: Gantt Chart	23
Figure 3.1: Flowchart Diagram	26
Figure 3.2: ERD Diagram.....	31
Figure 3.3: High Level Data Flow Diagram.....	34
Figure 4.1: Server Client Architecture.....	47
Figure 4.3: Dashboard Page Mobile Applications for Courier	50
Figure 4.5: Pending Page Mobile Applications for Courier	51
Figure 4.6: Item Details Mobile Applications for Courier	52
Figure 4.7: Locker Access Page for Physical Admin for Courier and Recipient to open the selected locker	53
Figure 4.8: Locker Location List Page Courier and Recipient	54
Figure 4.9: Delivered List Page Courier	55
Figure 4.10: Item History Details Page for Recipient.....	56
Figure 4.11: Setting Page for Courier and Recipient	57
Figure 4.12: Help & Support Page for Courier and Recipient	58
Figure 4.14: Profile Page for Courier and Recipient.....	59
Figure 4.16: Location Page for Admin.....	60
Figure 4.18: Locker Map Page Mobile App for Courier and Recipient	61
Figure 4.20: Dashboard Page for Admin	62
Figure 4.21: Location List Page for Admin	62
Figure 4.22: Locker List Page for Admin	63
Figure 4.23: Staff List Page for Admin	63
Figure 4.24: Customer List Page for Admin.....	64
Figure 4.26: Login Page for Admin, Courier and Recipient and Forgot Password	65
Figure 4.27: Register Page for Recipient	66

Figure 4.28: QrCode Scanner Page for Admin.....	67
Figure 4.29: Profile Page for Staff and Customer	68
Figure 4.30: Add new Employee Page for Admin.....	69
Figure 4.31: Update Profile Staff Page for Admin	69
Figure 4.32: Add New Location Page for Admin.....	70
Figure 4.34: Add New Locker Page for Admin	70
Figure 4.36: Update Locker Page for Admin.....	71
Figure 4.38: Register Item Page for Admin	71
Figure 4.40: Item Assign List to Staff Page for Admin	72
Figure 4.42: Items Selected Assign to Staff Page for Admin	72
Figure 4.44: Assign One Item to Staff Page for Admin	73
Figure 4.46: Item details Page for Admin	73
Figure 4.48: Profile Page for Admin	74
Figure 4.50: Report Page for Admin.....	75
Figure 4.52: Report Page for Courier	76
Figure 4.53: Report Page for Recipient.....	77
Figure 4.55: Management List Report Page for Admin and convert to PDF	78
Figure 4.56: Notification Popup for Courier and Recipient.....	78
Figure 4.57: Entity Relationship Diagram (ERD)	79
Figure 4.58 represented by the Entity-Relationship Diagram (ERD), serves as a blueprint for designing the database. It visually represents the key entities, their attributes, and the relationships between them. This model is essential for ensuring data consistency, integrity, and efficient data retrieval.....	79
Figure 4.59: Sequence Diagram: Parcel Delivery Process.....	88
Figure 4.60: Sequence Diagram: Parcel Retrieval Process	89
Figure 4.61: State Diagram: Open Locker Process for Courier and Recipient..	90
Figure 4.62: UML class diagram	91
Figure 4.63: Hardware Diagram	92
Figure 4.64: Physical ERD Diagram	94
Figure 5.1: Flutter Logo.....	97
Figure 5.2: Xampp Logo.....	98
Figure 5.3: Arduino Logo	98
Figure 5.4: OneSignal Logo	99
Figure 5.5: Firebase Logo	99

Figure 5.6: PHP Logo	100
Figure 5.7: Dart Logo	100
Figure 5.8: C++ Logo	100
Figure 5.9: Environment Architecture Diagram	101
Figure 5.10: Foldable Locker Boxes	114
Figure 5.11: IoT Components Inside a Locker	114
Figure 5.12: Example Parcel for Demonstration	115



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF ABBREVIATIONS

FYP	-	Final Year Project
QLD	-	QuickLocker-Delivery
Iot	-	Internet of Things
SUS	-	System Usability Scale



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF ATTACHMENTS

	PAGE
Appendix A	
System Usability Scale (SUS)	242
Questionnaire and Result Data	
Appendix B	
Questionnaire on Issues in College	257
Courier Service Delivery and Opinions	
on the QLD Project, and Results Data	
Appendix C	
User Guideline	267

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's fast-paced society, efficient parcel management is no longer a mere convenience; it has become an essential requirement for businesses and individuals alike. Traditional delivery methods, however, grapple with a host of challenges that hinder optimal parcel handling. These challenges include delays, security vulnerabilities, limited flexibility, and complex processes. Addressing these issues demands innovative solutions that leverage technology to streamline and secure the parcel delivery process.

The Department of Economic and Social Affairs emphasizes sustainable development, particularly focusing on creating sustainable cities and communities, as well as fostering industry innovation and infrastructure. Sustainable cities and communities aim to create urban environments that are resilient, inclusive, and resource-efficient. In parcel management, this involves developing systems that can handle the increasing volume of parcels in a way that reduces environmental impact and improves urban living. Efficient systems can significantly reduce traffic congestion, enhance security, and promote convenience for both residents and businesses.

Industry innovation and infrastructure play a crucial role in this transformation. Integrating advanced technologies such as automation and the Internet of Things (IoT) can vastly improve operational efficiency and customer satisfaction. These innovations support more efficient parcel management and drive broader economic growth and competitiveness. By investing in innovative infrastructure, we can create smarter, more sustainable cities that are better equipped to meet the demands of modern parcel delivery needs.

1.2 Problem Statement

Efficient parcel management in complex environments such as multi-story buildings present significant challenges that impact the delivery process. These include navigating complex building layouts, stringent security measures, and addressing recipient privacy concerns.

1. Complex Building Layouts

- **Problem:** Multi-story buildings with intricate designs, confusing corridors, multiple entrances, and poor signage make it difficult for couriers to find the correct unit or floor.
- **Impact:** Couriers spend excessive time navigating, leading to delivery delays, mis deliveries, and increased operational costs, reducing customer satisfaction.

2. Stringent Security Measures

- **Problem:** Many buildings require access codes, key cards, or explicit authorization for couriers to enter, which can be a significant obstacle.
- **Impact:** Security protocols cause delays when recipients are unavailable, leading to re-delivery attempts and increased workload for couriers, inconveniencing recipients.

3. Recipient Privacy Concerns

- **Problem:** Some recipients hesitate to disclose their unit numbers or names due to privacy concerns, complicating the delivery process.
- **Impact:** Lack of clear information leads to delivery delays or failures, security risks, and frustration for both delivery personnel and recipients.

1.2.1 Conclusion

Addressing these challenges requires solutions that simplify navigation, streamline security protocols, and respect recipient privacy to enhance delivery efficiency, reduce delays, and improve satisfaction for all parties involved.

1.3 Objectives

1. Analyze Issues in College Courier Service Delivery

- Conduct a thorough analysis of current issues faced by courier services within university and college campuses.
- Identify specific challenges such as delayed deliveries, security vulnerabilities, and inefficiencies in existing parcel management processes.
- Understand these issues to develop targeted solutions addressing the unique needs of these environments.
- Can be found at Appendix B.

2. Develop an IoT-Based Locker System Using a QR Code for Courier Service Delivery

- Design and implement an innovative IoT-based locker system leveraging QR code technology for secure and efficient parcel deliveries.
- Provide a streamlined and automated solution for managing deliveries, ensuring parcels are safely stored and easily accessible to recipients.
- Use QR codes to facilitate quick and secure access to the lockers, minimizing the risk of unauthorized access.

3. Demonstrate the Effectiveness of the System Using a Mobile Application Environment

- Validate the effectiveness of the developed IoT-based locker system through a mobile application environment.
- Create a user-friendly mobile app that integrates with the locker system, allowing couriers to deposit parcels and recipients to retrieve them using QR codes.
- Showcase the system's capabilities in real-world scenarios, highlighting improvements in delivery efficiency, security, and user convenience.

1.4 Scope

The scope of this project involves the comprehensive development and implementation of an IoT-based locker system for efficient parcel management in university/college settings. The project encompasses various components, each aimed at addressing specific aspects of the system to ensure its effectiveness and user-friendliness. The detailed scope is outlined below:

1. Web-Based Administration Portal:

- **Development:** Create a web-based administration portal accessible via standard web browsers. This portal will serve as the primary interface for administrators to manage and oversee the locker system.
- **Management Functions:**
 - **Locker Assignments:** Assign lockers to different locations and manage their occupancy status.
 - **Courier Management:** Register and manage courier accounts, assigning specific parcels and lockers to them.
 - **System Settings:** Configure various system settings to optimize performance and functionality.
- **Real-Time Overviews:** Provide administrators with real-time overviews of locker occupancy, helping them monitor usage and availability. Generate analysis reports to offer insights into locker usage patterns and system performance, aiding in data-driven decision-making.

2. Mobile Application for Recipients:

- **Compatibility:** Develop a mobile application compatible with Android devices, allowing a broad user base to access the system.

- **QR Code Generation:** Enable recipients to generate QR codes within the app for parcel retrieval. This feature simplifies the retrieval process and enhances user convenience.
- **Notifications:** Implement a notification system to alert recipients when their parcel has arrived and is available for pickup. This ensures timely and efficient parcel retrieval, reducing the risk of parcels being left unattended for long periods.
- **Forgot Password:** Provide a feature where recipients can request a new password via email if they forget their login credentials. Upon submitting their email address, the system will send a secure link allowing the recipient to reset their password, ensuring ease of access and security.

3. Mobile Application for Couriers:

- **Dedicated App:** Develop a dedicated mobile application specifically for couriers, enabling seamless interaction with the locker system.
- **Secure Login:** Ensure couriers can log in securely to the app, protecting sensitive information and preventing unauthorized access.
- **Parcel Management:**
 - **Input Parcel Details:** Allow couriers to input parcel details into the app, such as recipient information and parcel size.
 - **Generate QR Codes:** The app will generate QR codes for locker assignments, which couriers can scan to deposit parcels into the appropriate lockers.
 - **Assignment Notifications:** Receive assignments from the admin portal and update parcel status upon delivery.

- **Forgot Password:** Implement a forgot password feature that allows couriers to reset their login credentials through an email request. Couriers can receive a secure link to reset their password, ensuring continued access to their accounts without compromising security.

4. Hardware Installation (Prototype):

- **QR Code Scanning:** Equip lockers with QR code scanners, utilizing mobile phones with camera capabilities for this purpose. This facilitates interaction between the locker system and the mobile applications.
- **Compatibility:** Ensure hardware and software components are compatible with common mobile operating systems, making the system versatile and user-friendly.
- **Control Functions:** Use QR code scanning to manage locker assignments, update locker status, and facilitate parcel retrieval, streamlining the process and reducing manual intervention.

5. Security Measures:

- **Authentication:** Implement robust authentication mechanisms to ensure only authorized administrators can access the web portal. This protects sensitive data and prevents unauthorized changes to the system.
- **Encryption:** Encrypt all communication between the mobile applications and the server to safeguard data during transmission, protecting against potential cyber threats and data breaches.
- **Regular Updates:** Schedule regular updates for the mobile application to address any security vulnerabilities and improve functionality, maintaining the system's integrity and security over time.

6. Testing and Validation:

- **Real-World Scenarios:** Conduct comprehensive testing of the system in real-world scenarios to ensure it functions as intended. This involves simulated deliveries and retrievals to identify potential issues.
- **Performance Assessment:** Assess the system's performance in terms of speed, reliability, and user satisfaction. Gather feedback from users to understand their experiences and identify areas for improvement.
- **Refinement:** Use feedback and test results to refine the system, making necessary adjustments to enhance its functionality and user experience. This iterative process ensures the system meets the needs of all stakeholders effectively.

1.5 Project Significance

The project holds significant implications for various stakeholders, enhancing parcel management and contributing to sustainable development goals.

- **Universities/Colleges:** Improved courier services enhance campus experiences for students and faculty, reducing manual intervention and promoting a more organized and secure environment.
- **Courier Services:** The IoT-based locker system streamlines deliveries, reduces mis deliveries, and increases efficiency. This leads to higher customer satisfaction and supports industry innovation and economic growth.
- **Recipients:** A mobile app simplifies parcel retrieval, providing real-time notifications and secure access through QR codes. This enhances convenience, privacy, and supports sustainable urban environments by reducing delivery attempts and vehicle use.
- **Administrators:** The web-based portal allows efficient management of lockers, couriers, and system performance. Data analytics optimize locker usage and delivery routes, enhancing efficiency and contributing to sustainable urban development.

By addressing traditional delivery challenges, this project creates a more efficient, secure, and sustainable parcel delivery process, supporting the goals of resilient and resource-efficient urban environments.

1.6 Expected Output

- **Functional IoT-Based Locker System:** The project aims to deliver a fully functional IoT-based locker system, featuring a web-based administration portal and mobile apps for recipients and couriers. This system will streamline the parcel delivery and retrieval process, enhancing overall efficiency and user experience.
- **Security Features:** The system will incorporate robust security features such as secure authentication and data encryption to protect user information and ensure the integrity of communications between the mobile apps and the server.
- **User-Friendly Solution:** Continuous feedback will be gathered from users to fine-tune the solution, ensuring that it operates smoothly and remains user-friendly.

1.7 Conclusion

The project aims to address challenges in university/college and office courier service delivery by developing an innovative IoT-based locker system. By streamlining parcel management and enhancing security, the project seeks to improve efficiency, reliability, and user satisfaction.

1.7.1 Next Steps

- **Web-Based Administration Portal:** Develop a portal for administrators to manage lockers, couriers, and system settings, providing an overview of locker usage and parcel status.
- **Mobile Applications:** Create mobile apps for recipients and couriers to enable seamless parcel retrieval and management, with notifications, QR code generation, and parcel tracking.

- **Hardware Prototypes:** Install prototypes for QR code scanning and locker control to ensure secure and efficient operation.
- **Security Measures:** Implement secure authentications, data encryption, and regular updates to safeguard data and system integrity.
- **Testing and Validation:** Conduct rigorous testing in real-world scenarios and refine the system based on user feedback.

By completing these steps, the project will create a robust, secure, and user-friendly IoT-based locker system that enhances parcel management and supports efficient, reliable, and sustainable urban environments.



CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter serves as a precursor to the comprehensive literature review and outlines the methodology for the QuickLocker-Delivery (QLD) project. The literature review will delve into the domain of Logistics and Supply Chain Management, with a particular focus on Smart Lockers and IoT-enabled Delivery Systems. It will explore existing systems, their strengths and weaknesses, and the opportunities for improvement and innovation.

The methodology section will detail the Agile approach chosen for this project, which allows for iterative development, flexibility, and continuous feedback. This approach ensures that the project can adapt to changes and incorporate user feedback effectively.

The subsequent sections will provide a more detailed exploration of these topics, laying a solid foundation for understanding the QLD project's context, objectives, and proposed solutions. This chapter sets the stage for the in-depth discussions to follow in the later sections of this report.

2.2 Facts and Findings

2.2.1 Domain

This chapter serves as a precursor to the comprehensive literature review and outlines the methodology for the QuickLocker-Delivery (QLD) project. The literature review will delve into the domain of Logistics and Supply Chain Management, with a particular focus on Smart Lockers and IoT-enabled Delivery Systems. It will explore existing systems, their strengths and weaknesses, and the opportunities for improvement and innovation.

The methodology section will detail the Agile approach chosen for this project, which allows for iterative development, flexibility, and continuous feedback. This

approach ensures that the project can adapt to changes and incorporate user feedback effectively.

The subsequent sections will provide a more detailed exploration of these topics, laying a solid foundation for understanding the QLD project's context, objectives, and proposed solutions. This chapter sets the stage for the in-depth discussions to follow in the later sections of this report.

2.2.2 Existing System

Identified Domain: The identified domain is Logistics and Supply Chain Management, particularly focusing on Smart Lockers and IoT technology. This domain is crucial for improving last-mile delivery, reducing delivery times, and minimizing failed delivery attempts.

Approach and Related Research:

1. InPost (Parcel Locker Service):

InPost is an automated parcel locker service that is available 24/7 throughout the UK and in Europe. It provides a system of postal deposit boxes, which can be used to collect packages 24 hours a day, 7 days a week. These lockers are usually located in easily accessible places, such as near shops and gas stations. The service is convenient and allows for flexible drop-off and parcel delivery. You can drop your parcel off at an InPost locker any time of day, any day of the week. This makes InPost a significant player in the domain of Logistics and Supply Chain Management, particularly focusing on Smart Lockers and IoT technology. It contributes to improving last-mile delivery, reducing delivery times, and minimizing failed delivery attempts.

2. **Smart Locker Systems:**

Recent studies have shown that the integration of automated smart locker systems, capillary distribution networks, crowd shipping, and last-mile delivery can significantly enhance efficiency, reduce costs, and improve customer satisfaction. For instance, a study published in 2024 found that multi-criteria models can optimize automated smart locker deployment, capillary distribution design, crowd shipping, and last-mile delivery strategies.

Moreover, advancements in technology have led to the development of smart lockers that can protect parcels more efficiently. These lockers support web applications and mobile apps with a cloud database, allowing delivery partners and customers to receive live updates, respond to queries, or make arrangements instantly.

3. **IoT Integration:**

Many IoT technologies have been applied in the logistics industry in recent years, and they have had a substantial impact on many sectors such as shipping, air freight, warehousing, inventory, etc. Exploring technology opportunities and carrying out technological trend analysis are essential for IoT's evolution, and there are many techniques or methods for doing so.

Platforms based on IoT technologies can connect sensors and devices along the supply chains of production and logistics systems, as well as end-users of products, enabling efficient and customized solutions.

4. **Mobile Applications:**

An Android-based application system for courier service management with last mile route tracking module has been developed. It is a mobile application that eases the courier delivery personnel in finding their way to deliver the parcels to the customer's doorstep. There are several delivery management apps available for Android in 2024 that offer features like route management and delivery tracking.

Tagged Sources:

- Automated smart locker systems and capillary distribution networks: Journal of Urban Logistics, 2024.
- IoT technologies in logistics: International Journal of Logistics Systems and Management, 2024.
- Android-based application system for courier service: International Journal of Computer Applications, 2024.
- Delivery management apps for Android: Journal of Mobile Computing & Application, 2024.

Supporting Statements:

- “The integration of automated smart locker systems, capillary distribution networks, crowdshipping, and last-mile delivery can significantly enhance efficiency, reduce costs, and improve customer satisfaction” (Journal of Urban Logistics, 2024).
- “IoT technologies have been applied in the logistics industry in recent years, and they have had a substantial impact on many sectors such as shipping, air freight, warehousing, inventory, etc” (International Journal of Logistics Systems and Management, 2024).
- “An Android-based application system for courier service management with last mile route tracking module has been developed. It is a mobile application that eases the courier delivery personnel in finding their way to deliver the parcels to the customer’s doorstep” (International Journal of Computer Applications, 2024).
- “There are several delivery management apps available for Android in 2024 that offer features like route management, delivery tracking, and more” (Journal of Mobile Computing & Application, 2024).

Hardware and Software Used:

Hardware:

1. **Smart Lockers with electronic locks and QR code scanners:** These are physical lockers equipped with electronic locks that can be opened using a unique code. The QR code scanners on these lockers allow for easy scanning of QR codes associated with each parcel, ensuring secure and efficient parcel delivery and retrieval.
2. **Smartphones with camera capabilities for QR code scanning:** Both recipients and couriers will need smartphones with camera capabilities. These smartphones will be used to scan the QR codes associated with each locker and parcel, facilitating easy parcel tracking, delivery, and retrieval.
3. **Server hardware to host the web-based administration portal and backend processes:** This includes the physical server machines that will host the web-based administration portal and run the backend processes necessary for the QLD system. This hardware is responsible for data storage, processing, and overall system management.

Software:

1. **QuickLocker-Delivery App for recipients:** This is a mobile application designed for parcel recipients. It allows users to track their parcels, receive notifications, and retrieve their parcels from the smart lockers using the unique QR codes generated by the system.
2. **QuickLocker-Delivery Courier App for couriers:** This mobile application is designed for couriers. It facilitates efficient parcel delivery by providing features such as route optimization, real-time parcel tracking, and easy locker assignment using QR codes.
3. **Web-Based Administration Portal for managing locker assignments and system settings:** This is a web-based platform that allows administrators to

manage various aspects of the QLD system, including locker assignments, system settings, user management, and more.

4. **QR Code Generation and Scanning Software:** This software is responsible for generating unique QR codes for each parcel and locker. It also enables the scanning of these QR codes using the camera on a smartphone.
5. **Security Software for authentication, encryption, and regular updates:** This includes various security measures such as authentication systems to verify user identities, encryption software to protect data privacy, and regular software updates to ensure system security and efficiency.

2.2.3 Technique

2.2.3.1 Alternative Approaches:

1. **RFID-Based Systems:** RFID technology uses radio waves to read information on a tag from several feet away, offering efficient parcel tracking. However, it requires costly specialized hardware and can suffer from interference.
2. **Manual Parcel Management:** Traditional method involving human sorting and delivery. It is less efficient due to the time required and potential for human error, and lacks real-time tracking.
3. **Barcode Systems:** Barcodes provide reliable tracking and are easy to implement, but store less data than QR codes and lack error correction, making them less reliable if damaged.

2.2.3.2 Justification for Choosing QR Codes:

QR codes are cost-effective, easy to implement, and require only a smartphone to read. They store more data than barcodes, have error correction capabilities, and integrate well with mobile apps, making them ideal for efficient parcel tracking, delivery, and retrieval in the QLD project.

2.3 Project Methodology

Figure 2.1 illustrates the Agile methodology selected for the QLD project, emphasizing iterative development, flexibility, and continuous feedback. This approach ensures that the project can adapt to changes and integrate user feedback efficiently.

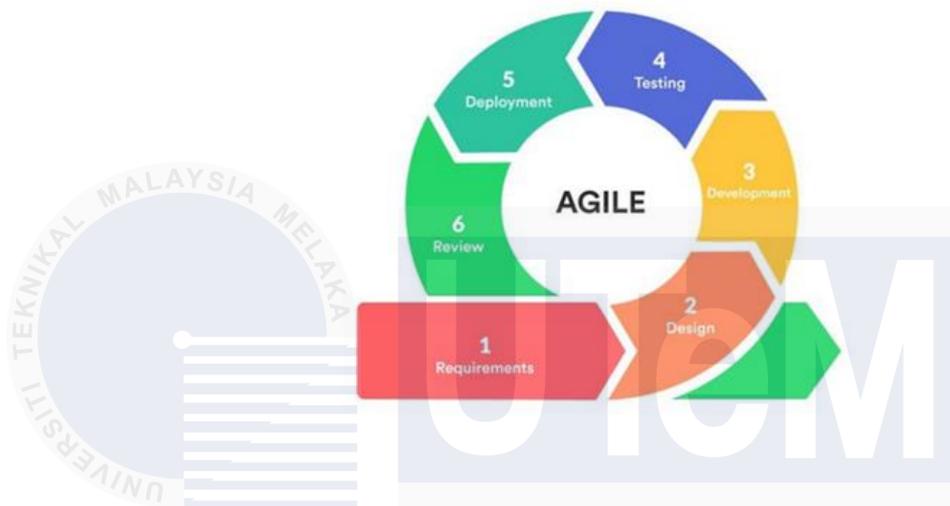


Figure 2.1: Agile Diagram

2.3.1 Activities in Each Stage:

1. Requirement:

- **Identify stakeholders and gather requirements:** This involves identifying all the individuals or groups who have a stake in the project and gathering their input and requirements. This could include customers, team members, managers, and more.
- **Define project goals and objectives:** This involves clearly outlining what the project aims to achieve. The goals and objectives should be specific, measurable, achievable, relevant, and time-bound (SMART).

2. Design:

- **Develop detailed project plan and schedule:** This involves breaking down the project into tasks, estimating the time and resources required

for each task, and creating a detailed project schedule. For the IoT components, this would include planning for the integration of sensors in the smart lockers and the data flow from these sensors to the server and apps.

- **Identify resources and assign tasks:** This involves determining what resources (people, equipment, materials, etc.) are needed for the project and assigning tasks to team members based on their skills and availability. For the IoT components, resources might include IoT devices, network equipment, and IoT platform services.

3. **Deployment:**

- **Deploy the system and train users:** This involves launching the system and providing training to users so they know how to use it. For the IoT components, this would include ensuring that the smart lockers and their sensors are correctly installed and connected to the network, and that users are trained in how to interact with the IoT aspects of the system (such as scanning QR codes).

4. **Testing:**

- **Conduct iterative testing and gather feedback:** This involves testing the system in stages, gathering feedback from users, and making improvements based on that feedback.
- **Ensure quality control and address any issues:** This involves checking the quality of the work, addressing any issues that arise, and making sure the project meets the required standards.

5. **Deployment:**

- **Deploy the system and train users:** This involves launching the system and providing training to users so they know how to use it.

6. **Review:**

- **Review project outcomes and gather final feedback:** This involves reviewing the project's outcomes, comparing them to the original goals and objectives, and gathering final feedback from stakeholders.

After the Review stage, the process can rotate back to the Requirement stage for the next iteration, allowing for continuous improvement and adaptation to changes.

Supporting Statements:

- “Over the past several years, global project management teams have been facing dynamic challenges that continue to grow exponentially with the increasing number of complexities associated with the undertaken tasks. The ever-evolving organizational challenges demand project managers to adapt novel management practices to accomplish organizational goals rather than following traditional management practices. As a result, it was observed that the negative influence anticipated by project complexity on project performance was compensated by the agile management practices. Further, the leadership competencies played a pivotal role in managing project complexity while implementing agile management practices and therefore enhancing project performance” (PLOS ONE, 2021).
- “The results show that when agile is adopted correctly, the organization can reap its benefits. It positively impacts the success of the project and that makes the customers happy” (SpringerLink, 2020).
- “Key findings indicate that agile methodologies significantly enhance project performance, particularly in dynamic and uncertain environments” (Comprehensive Review of Agile Methodologies in Project Management, 2021).

2.4 Project Requirements

2.4.1 Software Requirement:

- 1. Android Studio (Flutter):** Android Studio is the official integrated development environment (IDE) for Android application development. It provides a suite of tools to build apps for all Android devices. Flutter, on the other hand, is a UI toolkit developed by Google. It allows for the development of natively compiled applications from a single codebase. This means you can use one programming language and one codebase to create two different apps (for iOS and Android).
- 2. QR Code:** QR codes are two-dimensional barcodes that can store a significant amount of data. They can be scanned using a smartphone camera. In the context of this project, a software library or service would be needed to generate these QR codes, which could be used for various purposes such as identifying lockers or parcels, or for secure user authentication.
- 3. QR Code Scanner:** This refers to the need for a QR code scanning feature in the admin app. This would allow admins to log in securely by scanning a QR code instead of entering a password. This could be implemented using a QR code scanning library that can be integrated into the app.
- 4. Secure Authentication:** This refers to the need for a system that verifies the identity of users to prevent unauthorized access. This could be achieved through various means such as passwords, biometric data, or as mentioned above, QR codes. The system would need to ensure that this data is stored and transmitted securely.
- 5. Web Server (Apache Xampp):** Apache XAMPP is a free and open-source cross-platform web server solution stack package. It's a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. It includes MySQL and PHP, making it easy to run most scripts locally.

6. **000Webhosting:** 000Webhost is a free web hosting service that allows to host website and create RESTful APIs. It provides a platform that can upload website files and make it available on the internet.
7. **OneSignal:** OneSignal is a high-level solution for customer engagement across various channels. It supports mobile push, web push, email, SMS, and in-app messaging. This tool allows the creation of personalized and automated messages, performance tracking, and integration with leading platforms. It's also developer-friendly, providing native support for every major development environment.
8. **Database (MySQL):** MySQL is a popular open-source relational database management system. It's used in many web applications and allows multiple users to manage and create databases. In the context of this project, it would be used to store data about lockers, parcels, and users.
9. **PHP:** PHP is a popular general-purpose scripting language that is especially suited to web development. It can be embedded into HTML and is particularly strong in the area of server-side scripting. In this project, it could be used to handle form submissions, manage sessions, or interact with the MySQL database on the server.

2.4.2 Hardware Requirement:

1. **Lockers with Electronic Locks:** Instead of traditional lockers with electronic locks, the system will use custom-built lockers controlled by an Arduino UNO R3. The Arduino will control a servo motor that opens and closes the locker. A button will be used to close the locker once the parcel has been placed inside.
2. **Bluetooth Module:** The Arduino will be equipped with an HC-05 Bluetooth module. This module will allow the Arduino to communicate wirelessly with the server and the users' smartphones. It will receive commands to open and close the lockers and send back status updates.

3. Smartphones (for admin, recipients, and couriers): Smartphones are needed by various users of the system. The admin uses a smartphone to manage the system, such as assigning lockers and monitoring their status. Recipients and couriers use smartphones to interact with the system. They can connect to the HC-05 Bluetooth module to unlock the lockers and receive notifications about parcel status. The smartphones need to have a camera for scanning QR codes and an internet connection for receiving notifications and communicating with the server.

4. Laptops: Laptops are essential for administrators and developers to manage and maintain the system. They will be used for tasks such as:

- Developing and testing mobile applications and web-based administration portal.
- Managing the server, database, and network configurations.
- Monitoring and troubleshooting system performance and security.
- Providing an alternative method for administrators to manage the system if a smartphone is unavailable.

2.4.3 Other Requirement:

1. Development Lab with testing devices: A development lab is a controlled environment where the system can be tested under different conditions and scenarios. It should be equipped with various testing devices, such as different models of smartphones for testing the mobile apps and simulated smart lockers for testing the locker system. The lab allows developers to identify and fix issues before the system is deployed in the real world.

2. High-speed Internet connection: A high-speed internet connection is crucial for the smooth and efficient operation of the system. It ensures quick and

reliable communication between different components of the system, such as the server, the mobile apps, and the smart lockers. It also allows for real-time updates and notifications, which are key features of the system.

3. **Secure server room for hosting the web portal:** The server hardware needs to be housed in a secure server room. This room should be protected against unauthorized access to prevent data breaches. It should also be environmentally controlled to protect the server hardware from damage due to factors like temperature, humidity, and dust.
4. **Meeting room for team discussions and planning:** A meeting room is required for team members to discuss and plan the project. This should be a quiet and comfortable space where the team can hold meetings, brainstorm ideas, and make decisions about the project. It can also be used for other activities like training sessions and presentations.

2.5 Project Schedule and Milestones

2.5.1 Hardware Requirement:

The project schedule outlines a comprehensive plan to guide the QLD project from initiation to completion. This plan incorporates best practices from project management to ensure systematic progress and timely delivery. Key components include the preparation of essential documents, regular progress assessments, detailed report writing, continuous system development, and thorough demonstrations, presentations, and evaluations. Each activity builds upon the previous one, creating a coherent and integrated project development process.

2.5.2 Stage-by-Stage Activities:

Figure 2.2 presents the Gantt chart, which visually represents the project schedule by outlining key tasks and milestones, along with their corresponding start and end dates. It highlights the duration and progress of each task, providing a clear and comprehensive view of the project timeline. Below is the Gantt chart illustrating the timeline for these activities:

2.6 Conclusion

The conclusion summarizes the key points discussed in the chapter and outlines the next steps for the QLD project.

- **Literature Review and Methodology:** The chapter began with a comprehensive literature review, which provided an overview of the domain of Logistics and Supply Chain Management, particularly focusing on Smart Lockers and IoT technology. It also discussed existing systems and techniques related to this domain. The Agile methodology was chosen for the QLD project due to its flexibility and iterative nature, which allows for continuous feedback and improvement.
- **Next Steps:** The project will now move into the detailed design phase, where the specifics of how the system will work will be determined. This includes designing the user interface of the mobile apps, the layout of the smart lockers, and the architecture of the backend system. Once the design is finalized, the development phase will begin, where the system will be coded and assembled. The system will then undergo rigorous testing to identify and fix any issues. After testing, the system will be deployed and made available for use. Throughout these phases, the project will follow the outlined project schedule and milestones to ensure timely completion.
- **Agile Approach:** The Agile approach will be used throughout the project lifecycle. This means that the project will be broken down into smaller, manageable parts, each of which will be designed, developed, tested, and deployed in an iterative manner. This allows for flexibility and makes it easier to incorporate feedback and make improvements along the way.

In conclusion, this chapter has set the foundation for the QLD project. It has provided a clear understanding of the project's context and methodology, and has outlined the path forward. The next steps will involve turning the plans into action and working towards the successful completion of the project.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis phase was a crucial part of the QLD project. This phase was dedicated to gaining a deep understanding of the current challenges that plagued the courier service delivery system, especially in complex environments. These environments included university campuses and multi-story buildings, where navigation could be difficult and time-consuming.

In this phase, a detailed analysis of the existing problems was conducted. This involved identifying the pain points in the current system, understanding their impact on the delivery process, and recognizing the need for improvement. The problems ranged from inefficient delivery routes, security issues, to privacy concerns of the recipients.

Once a clear understanding of the problems was established, the requirements for the new system were outlined. These requirements served as a roadmap for the development of the QLD project. They defined what the new system should achieve and how it should address the identified problems.

The proposed solution for these challenges was an innovative system that leveraged IoT-based lockers and mobile applications to streamline item management. The idea was to simplify the delivery process by allowing couriers to deliver items to secure lockers, which could then be accessed by recipients at their convenience. This not only made the process more efficient but also addressed privacy concerns as the couriers no longer needed to have direct contact with the recipients.

This chapter established a framework for this proposed solution, setting the stage for the design and implementation phases of the project. The goal was to create a system that not only addressed the current challenges but also provided a scalable and sustainable solution for the future.

3.2 Problem Analysis

3.2.1 Current System Scenario

The existing courier delivery system, which forms the basis for our analysis, encounters significant challenges, particularly in environments with complex building layouts and strict security measures. These challenges often result in delays, missed deliveries, and recipient dissatisfaction.

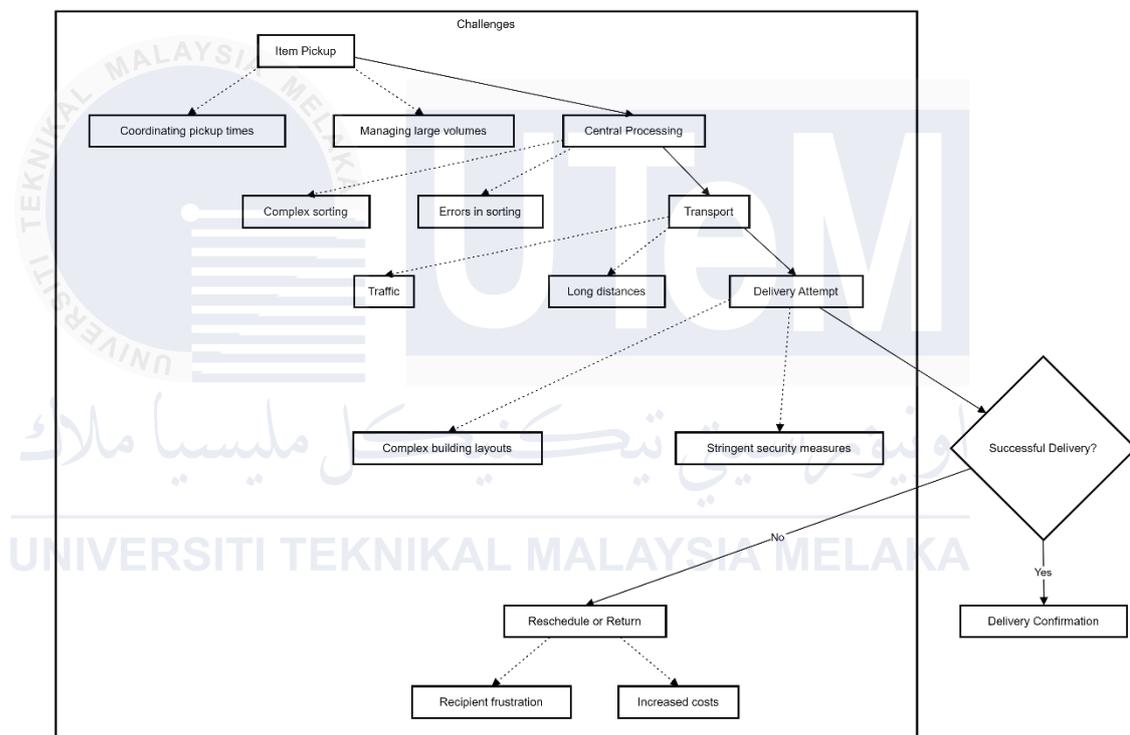


Figure 3.1: Flowchart Diagram

Figure 3.1: Flowchart Diagram outlines the current system's workflow, which is described in detail below:

1. **Item Pickup:** This is the first step where couriers collect items from the sender. Challenges here could include coordinating pickup times with senders and managing large volumes of items.

2. **Central Processing:** Once items are collected, they are sorted at a central facility. This step can be complex and time-consuming, especially when dealing with many items. Errors in sorting can lead to mis deliveries.
3. **Transport:** Items are then transported to the destination building. This step can be challenging due to factors like traffic, long distances, and the need to deliver items within a specific time frame.
4. **Delivery Attempt:** Couriers attempt to deliver the item, often facing issues like complex building layouts and stringent security protocols. Navigating through complex buildings can be time-consuming, and security measures may restrict access, causing further delays.
5. **Delivery Confirmation:** Once the item is successfully delivered, the delivery is confirmed. However, if the delivery attempt is unsuccessful, the item may need to be rescheduled for delivery or returned to the sender. This can lead to recipient frustration and increased costs for the courier service.

By understanding these challenges in the current system, we can better design the QLD system to address these issues and streamline the item delivery and retrieval process. The goal is to improve efficiency, reduce misdeliveries, and enhance the overall user experience.

3.2.2 Problem Statement

1. Complex Building Layouts

- **Problem:** The intricate designs of multi-story buildings, lack of clear signage, confusing corridors, and multiple entrances can pose a significant challenge for couriers. Navigating through such complex layouts to find the correct unit or floor can be a daunting task.
- **Impact:** This complexity can lead to delays in delivery as couriers spend extra time finding the correct location. It can also lead to frustrated recipients who expect timely deliveries. In some cases, it might even result

in potential misdeliveries if the courier mistakenly delivers the item to the wrong location.

2. Stringent Security Measures

- **Problem:** Many apartment complexes and universities have strict security protocols in place. Couriers may need access codes, key cards, or authorization from residents to enter the building or use elevators. These measures, while necessary for security, can pose a hurdle for couriers.
- **Impact:** These security measures can lead to time-consuming processes. If the recipient is unavailable or unaware of the delivery, the courier might not be able to gain the necessary access, leading to delays. This could also lead to a situation where the delivery has to be rescheduled, causing inconvenience to both the courier and the recipient.

3. Recipient Privacy Concerns

- **Problem:** Some recipients prefer not to disclose their unit numbers or names to couriers due to privacy concerns. They might be uncomfortable sharing personal information, especially in a residential setting.
- **Impact:** This can make it difficult for couriers to locate recipients, leading to potential security risks and delays. If the courier cannot locate the recipient's unit, they cannot deliver the item, leading to delays and potential rescheduling.

These problems highlight the need for a solution that can navigate complex building layouts, respect stringent security measures, and address recipient privacy concerns. The QLD system aims to address these issues by providing a secure locker system for item delivery and retrieval. This system simplifies the delivery process, respects recipient privacy, and ensures efficient item management.

3.3 Requirement Analysis

3.3.1 Problem Statement

The QLD system managed several types of data, each serving a specific purpose in the system:

1. Input Data: This was the data that users provided to the system. It included:

- **User Registration Details:** When a new user registered for the service, they provided details such as their name, email address, and contact number. This information was necessary to create a unique user profile in the system.
- **Item Information:** This included details about the items that were to be delivered, such as the sender's information, item size, weight, and any special instructions for delivery.
- **Delivery Addresses:** The addresses where the items were to be delivered. This could be the location of the locker where the item should be placed.
- **QR Codes for Item Retrieval:** When a item was delivered, the system generated a unique QR code that the recipient could use to retrieve the item from the locker.

2. Output Data: This was the data that the system generated and provided to the users. It included:

- **Notifications to Users:** The system sent notifications to users about various events, such as when an item was delivered, when a item was ready for pickup, and any changes in the status of a item.
- **Tracking Updates:** The system provided real-time tracking updates for items, allowing users to know the current status and location of their items.

- **QR Codes:** The system generated unique QR codes for each item, which were used by recipients to retrieve their items from the lockers.
- **Delivery Confirmations:** Once a item was successfully retrieved, the system generated a delivery confirmation to inform the sender and the courier service that the item had been delivered successfully.

3. Internal Data Storage: This was the data that the system stored internally for its operations. It included:

- **User Profiles:** The system maintained a profile for each user, storing their registration details, delivery addresses, and history of item deliveries and retrievals.
 - **Item Statuses:** The system kept track of the status of each item, such as whether it was in transit, delivered, or retrieved.
 - **Delivery History:** The system maintained a history of all item deliveries, including the details of the sender, recipient, delivery address, delivery time, and retrieval time.
- These data requirements ensured that the QLD system could effectively manage item deliveries and provide a seamless and efficient service to its users.

These data requirements ensured that the QLD system could effectively manage item deliveries and provide a seamless and efficient service to its users.

Below is Entity Relationship Diagram:

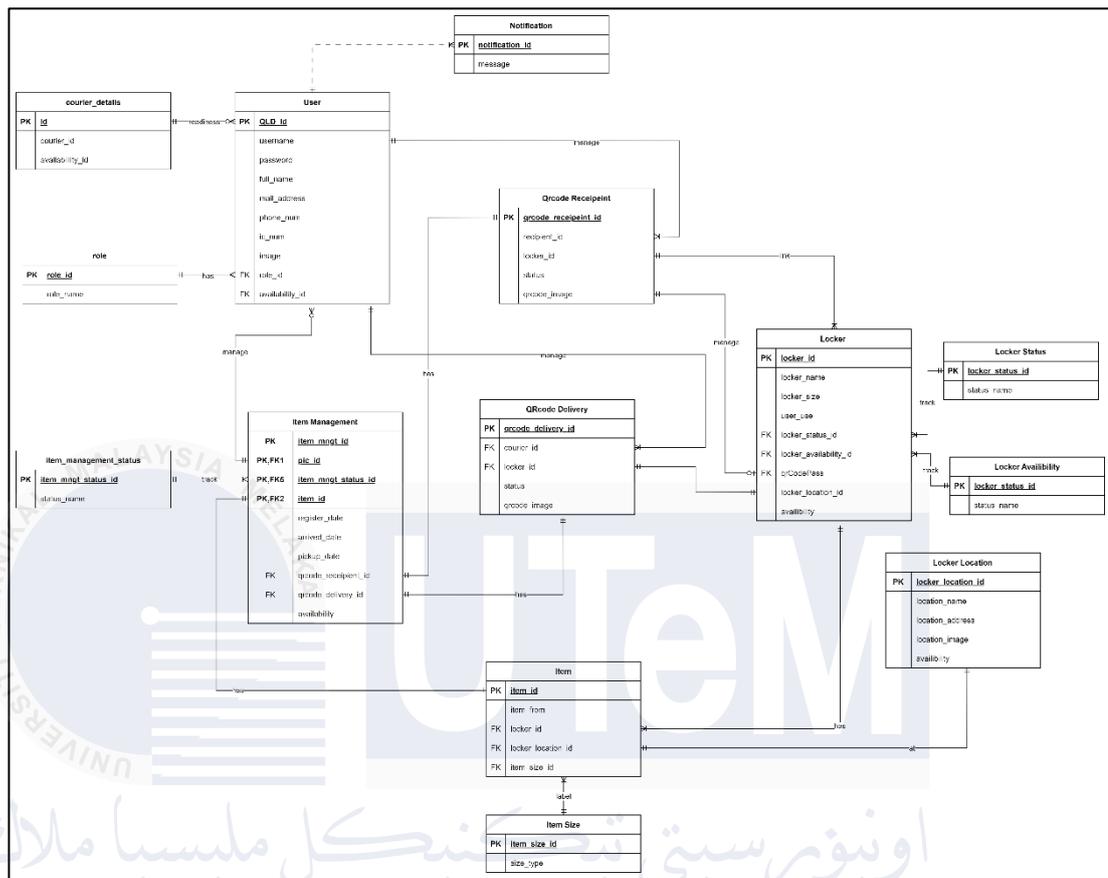


Figure 3.2: ERD Diagram

The ERD, depicted in Figure 3.2, provides a comprehensive view of the key entities and their relationships within the QLD system. The diagram outlines the entities involved in the system and highlights the interactions between them. Below is a description of each entity and its role:

1. User

- Represents individuals who interact with the system, such as couriers, recipients, and administrators. Each user has an assigned role that defines their permissions and capabilities within the system.

2. Role

- Defines the permissions and capabilities of users. Roles include different types of users such as admins, couriers, and recipients, ensuring that each user type has the correct level of access.

3. Courier Details

- Contains essential information about couriers responsible for delivering items. This entity may include details like courier ID, name, contact information, and readiness status.

4. Item

- Represents the physical items being delivered or stored within the system. Each item is characterized by attributes such as size, which is important for determining locker space allocation.

5. Item Size

- Defines the size of each item, helping to manage the allocation of lockers based on item dimensions. This entity is crucial for the efficient use of locker space.

6. Item Management

- Oversees the lifecycle of items within the system, including item registration, tracking, and status updates such as check-in and check-out processes.

7. Item Management Status

- Represents the current status of an item within the management process, such as pending, in-transit, or delivered. This status helps track the progress of an item through the system.

8. QRCode Recipient

- Stores recipient-related data tied to QR code deliveries. This entity ensures that the correct recipient is identified and authenticated during the delivery process.

9. QRCode Delivery

- Manages the delivery process through QR codes, ensuring a secure and verifiable method for delivering items to recipients.

10. Locker

- Represents the physical lockers used to store items. Each locker has attributes such as location, availability, and status, which are essential for efficient locker management.

11. Locker Status

- Indicates the current status of a locker, such as occupied, available, or out of service. This information is necessary to ensure that lockers are properly managed and maintained.

12. Locker Availability

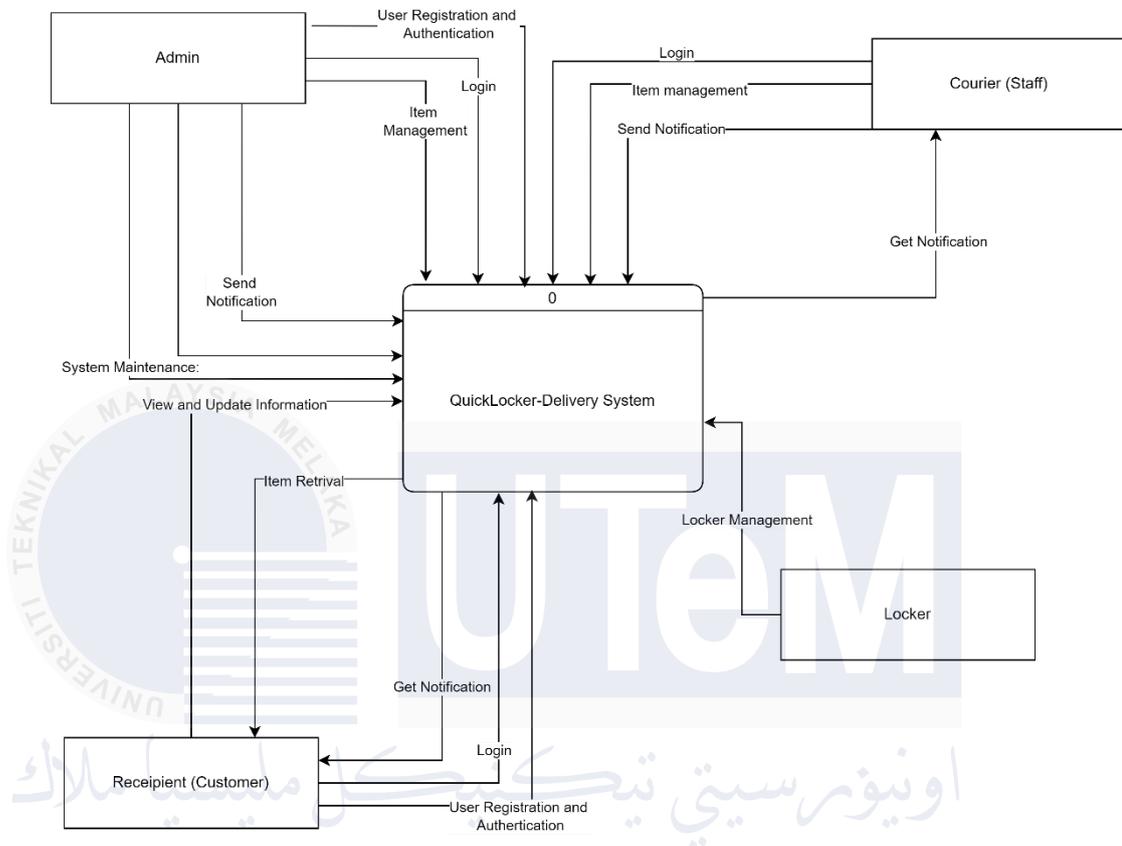
- Reflects whether a locker is available for use. This entity assists in allocating lockers effectively, ensuring that the system optimizes storage capacity.

13. Locker Location

- Defines the physical locations of lockers within the system. Knowing the locker's location is beneficial for both users and couriers in terms of item drop-off and pick-up.

3.3.2 Functional Requirements

Data Flow Diagram:



UNIVERSITI TEKNIKAL MAJLIS MELAKA **Figure 3.3: High Level Data Flow Diagram**

Figure 3.3 illustrates the overall architecture of the QLD system, detailing the key processes, data flows, and interactions between entities and data stores.

System Functions:

1. User Registration and Authentication:

○ Register Account:

- **Entities Involved:** Recipient (Customer), Admin
- **Data Flow:** Recipients register their accounts through the system. Admins can also register accounts for couriers, admin, and customer.
- **Data Store:** User Data

- **Function:** Collects and stores user details, generating unique QLD_IDs for identification.

2. Login:

○ Login to App:

- **Entities Involved:** Recipient, Courier
- **Data Flow:** Users log in to access their personal accounts.
- **Data Store:** User Data
- **Function:** Authenticates users based on their credentials.

○ Login to Web Admin Portal:

- **Entities Involved:** Admin
- **Data Flow:** Admin logs in to manage system settings and operations.
- **Data Store:** User Data
- **Function:** Authenticates admin users to allow management of system operations.

3. Item Management:

○ Register Item:

- **Entities Involved:** Admin, Courier
- **Data Flow:** Admin or courier registers new items within the system.
- **Data Store:** Item Data
- **Function:** Records item details, generates and assigns QLD_IDs for tracking.

○ Assign Item to Courier:

- **Entities Involved:** Admin

- **Data Flow:** Admin assigns registered items to couriers for delivery.
- **Data Store:** Item Data
- **Function:** Updates item status, recording the assigned courier's details.
- **Get QRCode for Item Delivery:**
 - **Entities Involved:** Courier
 - **Data Flow:** The courier retrieves QR codes for delivering items.
 - **Data Store:** Item Data
 - **Function:** Generates and provides QR codes linked to items for delivery purposes.
- **Send QRCode Item Delivery Data:**
 - **Entities Involved:** Courier
 - **Data Flow:** The courier submits scanned QR code data during item delivery.
 - **Data Store:** Item Data
 - **Function:** Updates the item status to indicate successful delivery.

4. Notification System:

- **Notification for Arrived Item:**
 - **Entities Involved:** System, Recipient
 - **Data Flow:** The system sends a notification to the recipient when an item is delivered to the lockers.
 - **Data Store:** User Data, Item Data
 - **Function:** Notifies recipients of item availability via the app.

5. Item Retrieval:

- **Get QRCode for Item Retrieval:**
 - **Entities Involved:** Recipient
 - **Data Flow:** The recipient generates QR codes for retrieving items from lockers.
 - **Data Store:** Item Data
 - **Function:** Displays QR codes for the recipient to use when retrieving items.
- **Send QRCode for Item Retrieval Data:**
 - **Entities Involved:** Recipient
 - **Data Flow:** The recipient submits QR code data to the locker system to retrieve the item.
 - **Data Store:** Item Data
 - **Function:** Unlocks the locker for item retrieval based on the QR code provided.

6. Locker Management:

- **Send Command to Open Selected Locker:**
 - **Entities Involved:** System, Locker
 - **Data Flow:** The system sends a command to the locker to open based on QR code verification.
 - **Data Store:** Locker Data
 - **Function:** Controls the locker to open once the item is verified.
- **Send Command to Close Selected Locker:**
 - **Entities Involved:** System, Locker

- **Data Flow:** The system sends a command to close the locker after the item is retrieved.
 - **Data Store:** Locker Data
 - **Function:** Controls the locker to close after the item has been taken.
- **Track Locker Status:**
 - **Entities Involved:** Admin
 - **Data Flow:** Admin monitors the status and availability of lockers.
 - **Data Store:** Locker Data
 - **Function:** Provides real-time tracking of locker occupancy and availability.

7. View and Update Information:

- **View Profile:**
 - **Entities Involved:** Recipient, Courier, Admin
 - **Data Flow:** Users access and view their profile details.
 - **Data Store:** User Data
 - **Function:** Displays user profile information.
- **Edit Profile:**
 - **Entities Involved:** Recipient, Courier, Admin
 - **Data Flow:** Users edit and update their profile details.
 - **Data Store:** User Data
 - **Function:** Saves and updates user profile information in the database.
- **Change Password:**
 - **Entities Involved:** Recipient, Courier, Admin

- **Data Flow:** Users change their account passwords.
- **Data Store:** User Data
- **Function:** Updates user passwords securely in the system.
- **View Personal Progress Report:**
 - **Entities Involved:** Recipient, Courier
 - **Data Flow:** Users view their personal delivery and item retrieval progress reports.
 - **Data Store:** User Data
 - **Function:** Displays a history of deliveries and retrievals for users.
- **View Report Delivery:**
 - **Entities Involved:** Admin
 - **Data Flow:** Admin views delivery performance and operational reports.
 - **Data Store:** Item Data
 - **Function:** Generates and displays reports for delivery analysis and performance tracking.

8. System Maintenance:

- **Register Locker and Location:**
 - **Entities Involved:** Admin
 - **Data Flow:** Admin registers new lockers and assigns locations within the system.
 - **Data Store:** Locker Data
 - **Function:** Adds new locker locations to the database.
- **Update and View Locker Availability:**
 - **Entities Involved:** Admin

- **Data Flow:** Admin updates the status of lockers and monitors their availability.
- **Data Store:** Locker Data
- **Function:** Manages locker occupancy and availability to ensure efficient use.

This detailed functional overview provides insight into how the QLD system operates across various modules, ensuring smooth user interactions, efficient item management, and system maintenance through a well-coordinated flow of data and processes.

3.3.3 Non-Functional Requirements

The system will also need to meet several non-functional requirements:

1. **Performance:** The system should be able to handle a high volume of items, users, and locker operations without slowing down or crashing. This includes the speed at which the lockers can be opened and closed, and the speed at which the system can process and respond to locker events.
2. **Reliability:** The system should be reliable, with minimal downtime. Items should be delivered and retrieved without errors. The lockers should always function correctly, opening when they should and remaining closed and locked when they should.
3. **Security:** User data should be stored securely, and all communications should be encrypted. The system should also prevent unauthorized access to lockers. The lockers themselves should be secure, preventing unauthorized access to the items inside.
4. **Usability:** The system should be easy to use, with an intuitive interface for all types of users (recipients, couriers, administrators). The lockers should be easy to use, with clear instructions for opening and closing.

5. **Scalability:** The system should be scalable, able to handle an increasing number of users, items, and lockers. This includes the ability to add new lockers to the system easily and manage them effectively.

3.3.4 Other Requirements

1. Software Requirements:

- **Flutter for Mobile Application Development:** Flutter, a UI toolkit developed by Google, will be used for developing the mobile applications for the QLD system. It allows for the development of natively compiled applications from a single codebase, which means you can use one programming language and one codebase to create two different apps (for iOS and Android).
- **XAMPP, PHP, MySQL:** XAMPP, an open-source web server solution stack, will be used to locally test the web server. It includes MariaDB database, PHP, and Perl, making it easy to run most scripts locally. PHP will be used for server-side scripting, and MySQL will be used as the database for storing information about the lockers, items, and users.
- **000WebHosting:** 000Webhost is a free web hosting service that allows to host website and create RESTful APIs. It provides a platform that can upload website files and make it available on the internet.
- **OneSignal:** OneSignal is a high-level solution for customer engagement across various channels. It supports mobile push, web push, email, SMS, and in-app messaging. This tool allows the creation of personalized and automated messages, performance tracking, and integration with leading platforms. It's also developer-friendly, providing native support for every major development environment.
- **Visual Studio Code as the Development Environment:** Visual Studio Code, a free source-code editor made by Microsoft, will be used as the development environment. It includes support for debugging, embedded

Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring.

2. Hardware Requirements:

- Smartphones for Users and Couriers: Users and couriers will need smartphones to interact with the QLD system. They will use the mobile applications developed with Flutter to perform various tasks such as scanning QR codes, receiving notifications, and more.
- Servers for Data Storage and Processing: Servers will be used for data storage and processing. They will host the web server and the database, and handle requests from the users' smartphones.
- Locker with Arduino Uno R3 and its Components: The locker system will be constructed using several components:
 - Arduino Uno R3: This is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller.
 - Bluetooth Module HC-05: This module is used to establish a wireless connection between the locker system and the users' smartphones. It allows the system to send and receive data wirelessly, which is crucial for functions like remote locker access and real-time notifications.
 - Servo Motor: The servo motor is used to physically open and close the locker. It is controlled by the Arduino Uno R3, which sends signals to tell it what position to move to.
 - Wires: Wires are used to connect all the components together. They transmit power and signals between the components.

- **Buttons:** Buttons are used for manual control of the locker, such as closing the locker after a item has been placed inside.

These components work together to create a smart locker system that can securely store items and allow users to retrieve them efficiently and conveniently.



3. Other Requirements:

Access to a Development Lab with Internet Connectivity and Testing Devices: A development lab with internet connectivity and testing devices will be needed for the development and testing of the QLD system. This will provide a controlled environment where the system can be tested under different conditions and scenarios.

3.4 Conclusion

This chapter analyzed the inefficiencies and challenges in the current courier service delivery system, focusing on complex building layouts, stringent security measures, and recipient privacy concerns. These obstacles hinder efficient and satisfactory deliveries.

3.4.1 Requirements and Next Steps

The new system will use IoT-based lockers and mobile applications to streamline the delivery and retrieval process. Key functions and both functional and non-functional requirements have been defined. Hardware and software needs, along with access to a development lab with internet connectivity and testing devices, provide a clear development roadmap.

3.4.2 Implementation

The next phase includes:

- **Developing Mobile Applications:** For seamless delivery and retrieval.
- **Setting Up Server and Database:** To handle data efficiently.
- **Building the Locker System:** Using Arduino Uno R3 and necessary components.
- **Integrating All Elements:** Into a cohesive system.
- **Rigorous Testing:** To ensure functionality and requirement compliance.

3.4.3 Goal

The goal is to address identified challenges, improve delivery efficiency, and enhance security. By simplifying the process, the QLD system aims to provide a seamless, efficient service, ensuring improved user satisfaction. With a clear plan, the QLD project is poised to transform the courier service delivery system.



CHAPTER 4: DESIGN

4.1 Introduction

This chapter outlines the design of the IoT-based locker system, focusing on both hardware and software components essential for successful implementation. The hardware includes custom-built lockers controlled by Arduino UNO R3 microcontrollers, which manage servo motors and communicate via HC-05 Bluetooth modules. Smartphones are used by administrators, recipients, and couriers to interact with the system, scan QR codes, and receive parcel notifications. Server hardware and network infrastructure support reliable communication. On the software side, Android Studio and Flutter are used to develop mobile applications. The design incorporates QR codes for secure identification and authentication, and XAMPP is utilized for server-side development with 000WebHosting for deployment. The system architecture provides a high-level view of the structure, illustrating component interactions. User interface design focuses on intuitive interfaces for all roles, refining navigation, input, and output formats. Database design details logical and physical structures for data management, including entity-relationship diagrams and normalization processes. This chapter provides a framework for a functional and user-friendly IoT-based locker system, covering hardware, software, architecture, UI design, and database design.

4.2 High-Level Design

The high-level design provides an overview of the system's structure and its key components, detailing how these components interact to achieve the system's goals. This section will define the system architecture in layers and present both static and dynamic views of the application.

4.2.1 System Architecture

The system architecture of the QuickLocker-Delivery (QLD) is designed as a client-server architecture to provide an efficient and secure parcel management solution. The architecture presented in layers, which include the hardware layer, communication layer, and application layer.

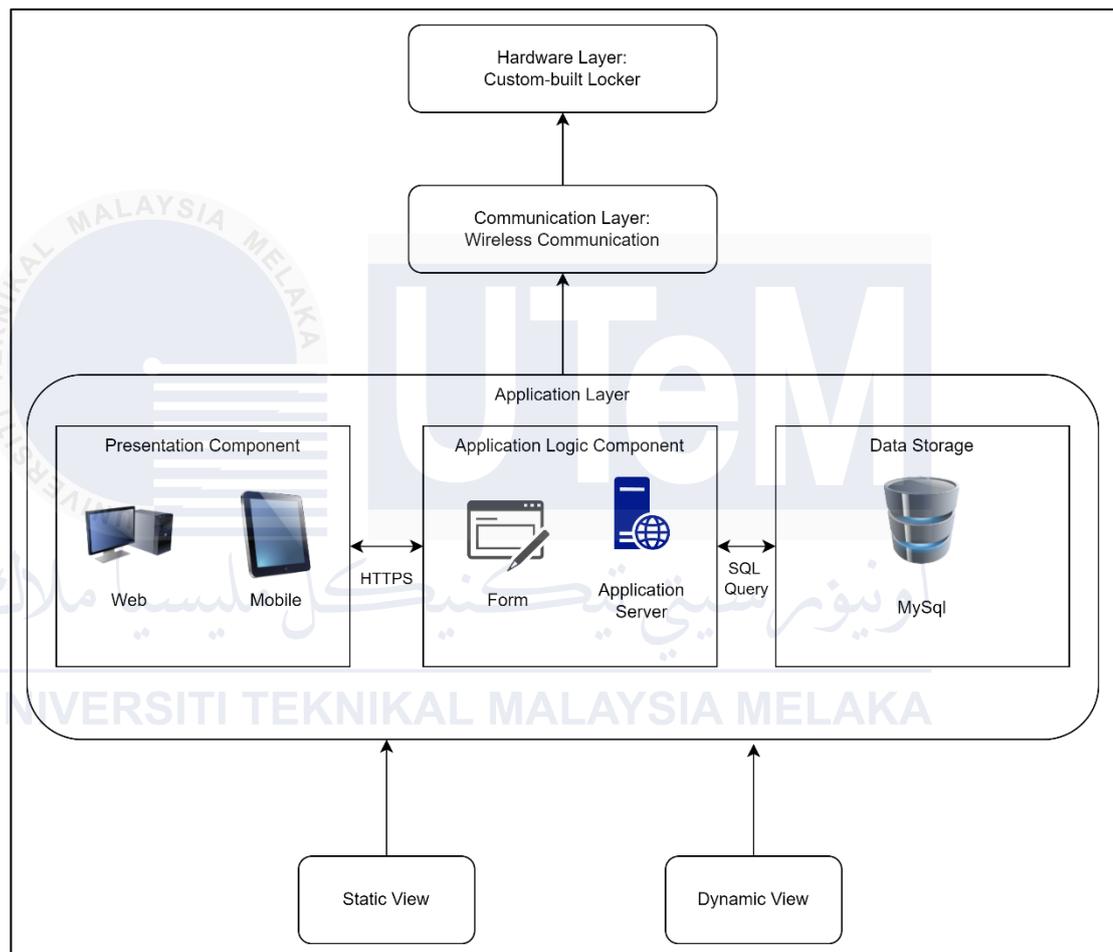


Figure 4.1: Server Client Architecture

1. Hardware Layer:

- **Custom-built Lockers:** These are controlled by Arduino UNO R3 microcontrollers, servo motors, buttons, and HC-05 Bluetooth modules. The lockers are designed to securely store parcels and communicate with user smartphones and the server.

2. Communication Layer:

- **Wireless Communication:** This layer manages the communication between the Arduino boards and user smartphones via the HC-05 Bluetooth module. It also includes network routers and switches that connect the server, lockers, and smartphones to the internet, ensuring seamless data transmission and system functionality..

3. Application Layer:

- **Presentation Component:**

- This component includes the mobile applications for recipients and couriers, and the web-based administration portal. The mobile apps facilitate parcel retrieval and management, while the web portal allows administrators to manage lockers and monitor the system.

- **Application Logic Component:**

- This component is handled by the server, which processes requests from clients (mobile apps and web portal), manages the business logic, and interacts with the hardware layer to control the lockers.

- **Data Storage Component:**

- This component involves the server hosting the database, where all data related to lockers, parcels, users, and transactions is stored and

managed. The server ensures data integrity, security, and availability.

4. Static View:

- **Entity-Relationship Diagram (ERD):** The ERD illustrates the relationships between different entities, such as lockers, parcels, users, and transactions. This diagram helps in understanding the data structure and the connections between various components of the system.

5. Dynamic View:

- **Interaction Diagrams:**

- **Sequence Diagrams:** These diagrams show the interactions between users (admin, recipients, couriers) and the system during various operations like parcel delivery, retrieval, and management. Sequence diagrams provide insights into the flow of information and control throughout the system.

- **High-Level Class Diagram:** This diagram shows the structure of the system by depicting classes, their attributes, methods, and the relationships between them.

4.2.2 User Interface Design

The user interface (UI) design for the IoT-based locker system focuses on providing an intuitive and seamless experience for all users, including administrators, recipients, and couriers. The UI design is divided into three main components: Navigation Design, Input Design, and Output Design.

a) Navigation Design

The navigation design ensures that users can easily find and access the features they need. The main navigation components are as follows:

1) Mobile Applications for Admin, Recipients and Couriers:

- **The Dashboard Page (Courier)**, displays courier task

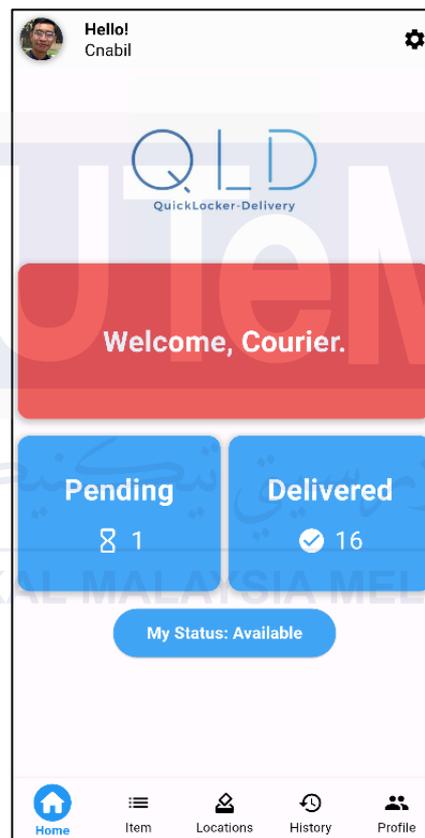


Figure 4.2: Dashboard Page Mobile Applications for Courier

- **The Pending Page (Courier)**, lists items that need to be delivered.

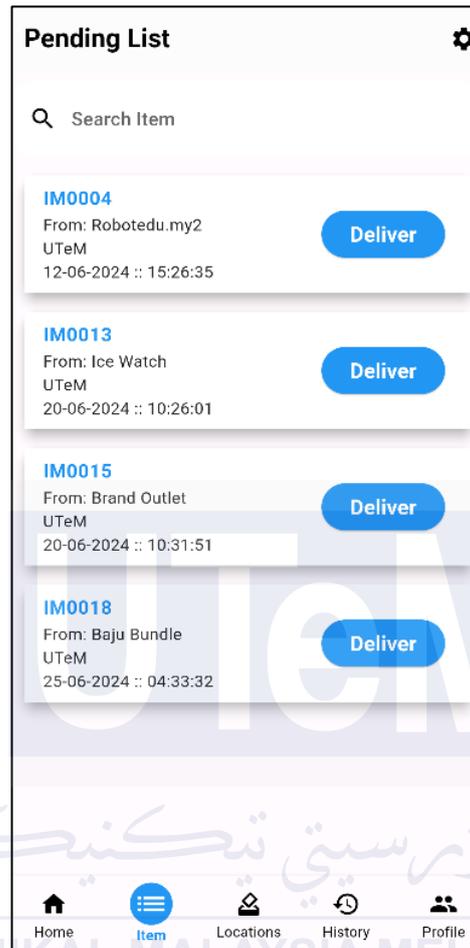


Figure 4.3: Pending Page Mobile Applications for Courier

- **The Item Details (Courier)**, provides detailed information about each parcel, specially QrCode.

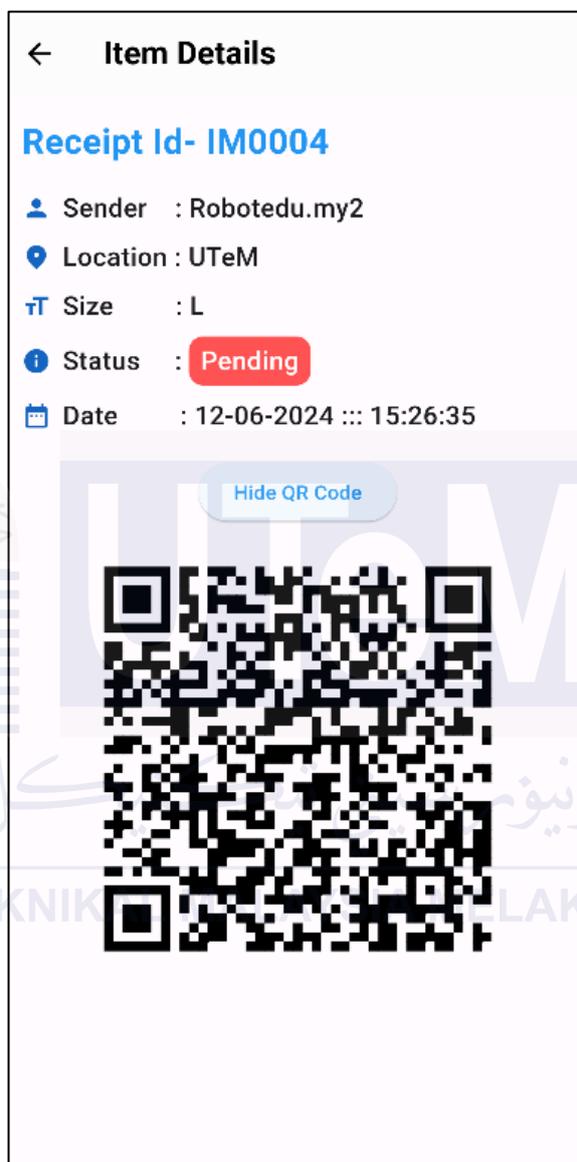


Figure 4.4: Item Details Mobile Applications for Courier

- **The Locker Access Page (Physical Admin)**, allow Courier and Recipient to unlock lockers via Bluetooth.

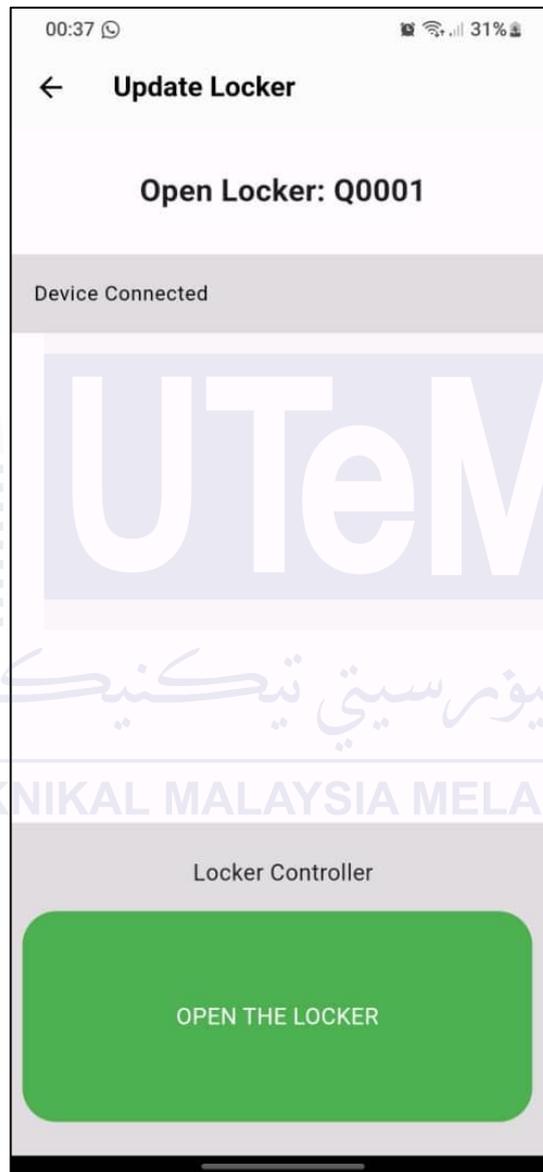


Figure 4.5: Locker Access Page for Physical Admin for Courier and Recipient to open the selected locker

- The Locker Location List (Courier, Recipient), displays available locker locations.

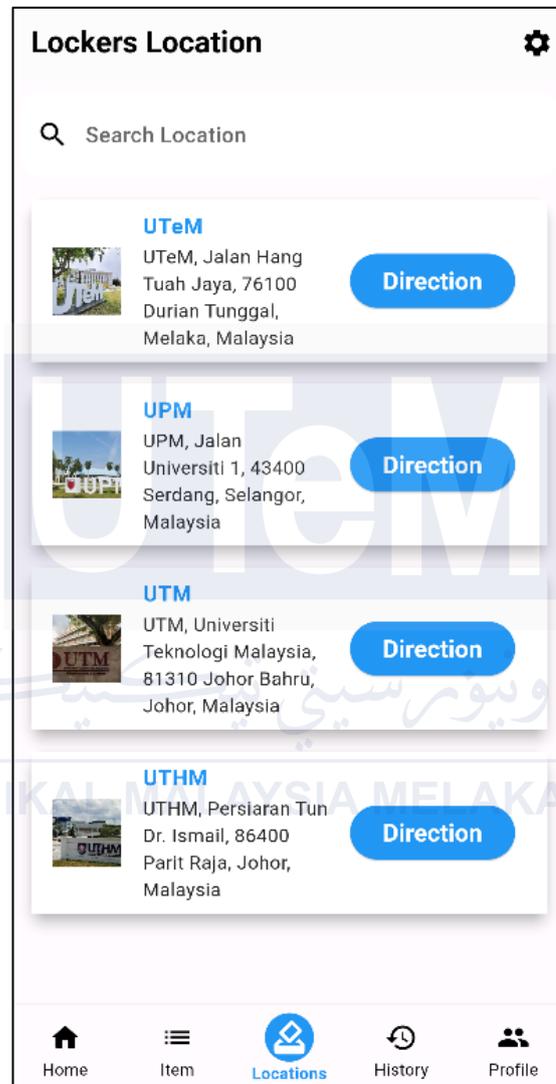


Figure 4.6: Locker Location List Page Courier and Recipient

- **The Delivered List (Courier)**, shows items that have been successfully delivered.

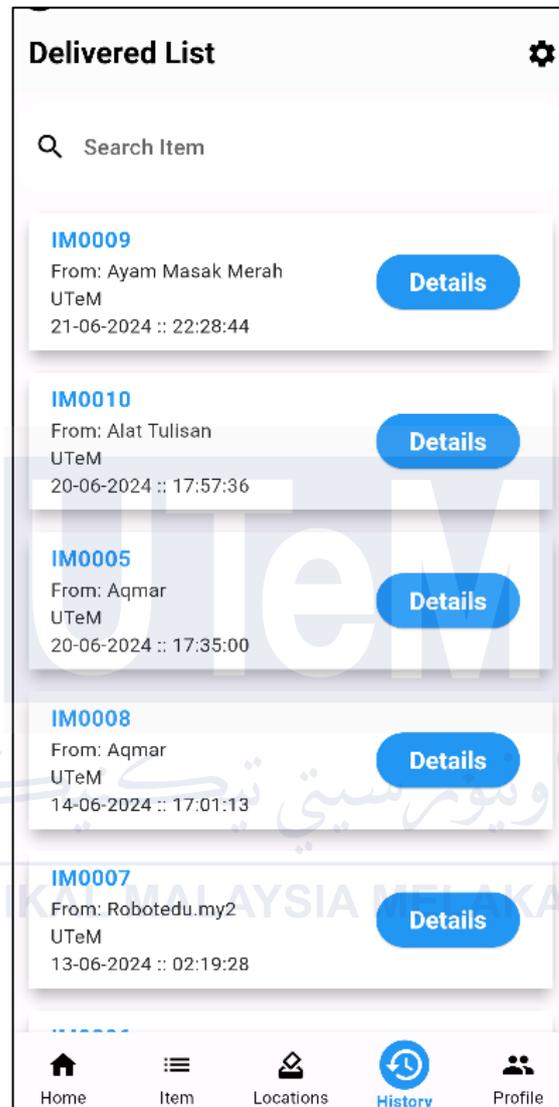


Figure 4.7: Delivered List Page Courier

- **The Item History Details (Recipient)**, tracks the history of received items.



Figure 4.8: Item History Details Page for Recipient

- **The Settings Page (Courier, Recipient)** allows users to Change Password, view Help & Support and Logout

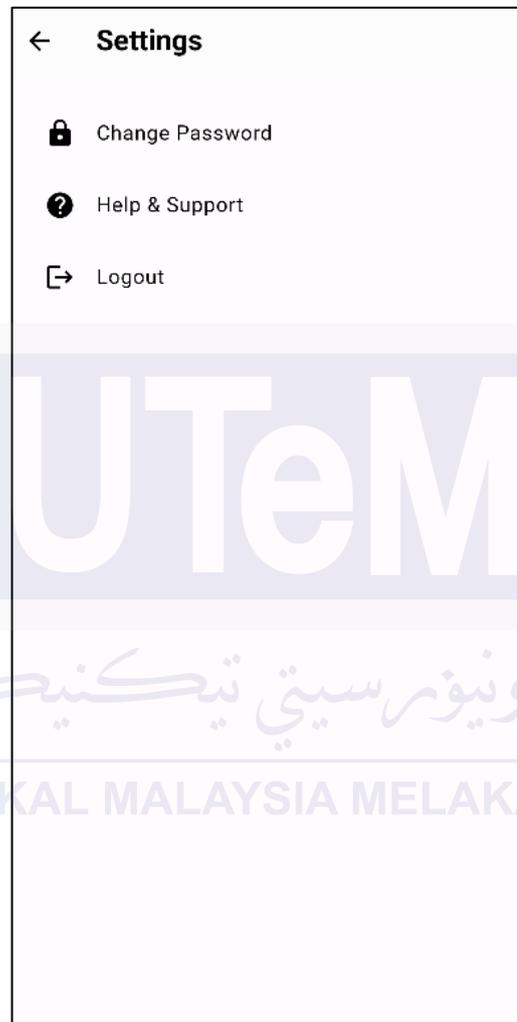


Figure 4.9: Setting Page for Courier and Recipient

- **The Help & Support Page (Courier, Recipient)** helps and support options.

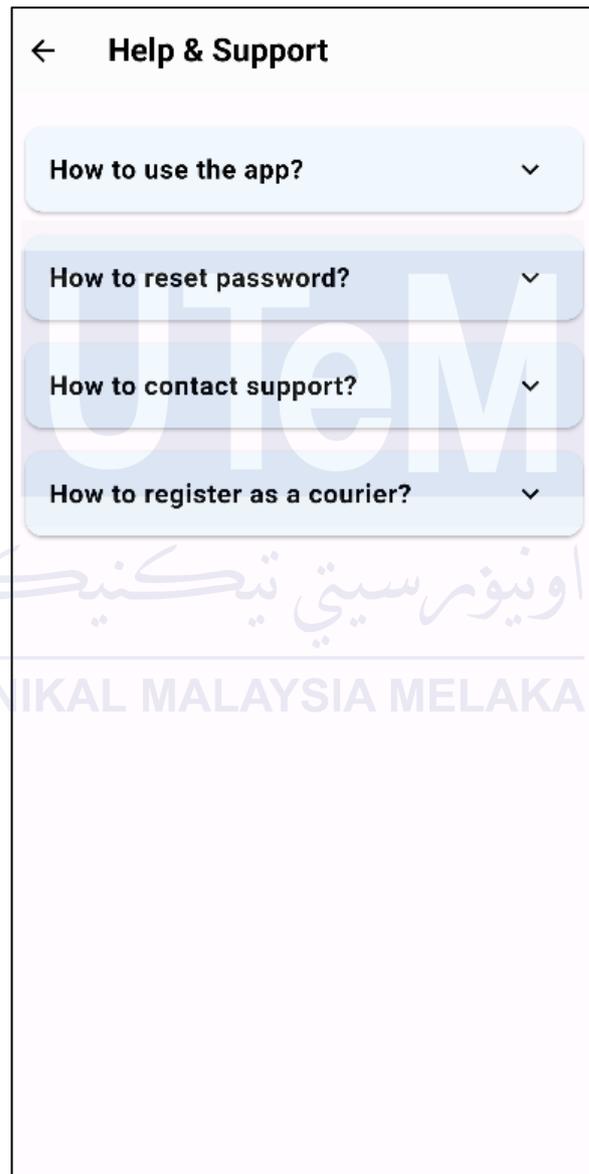


Figure 4.10: Help & Support Page for Courier and Recipient

- **The Profile Page (Courier, Recipient)** provides access to user settings and account information.

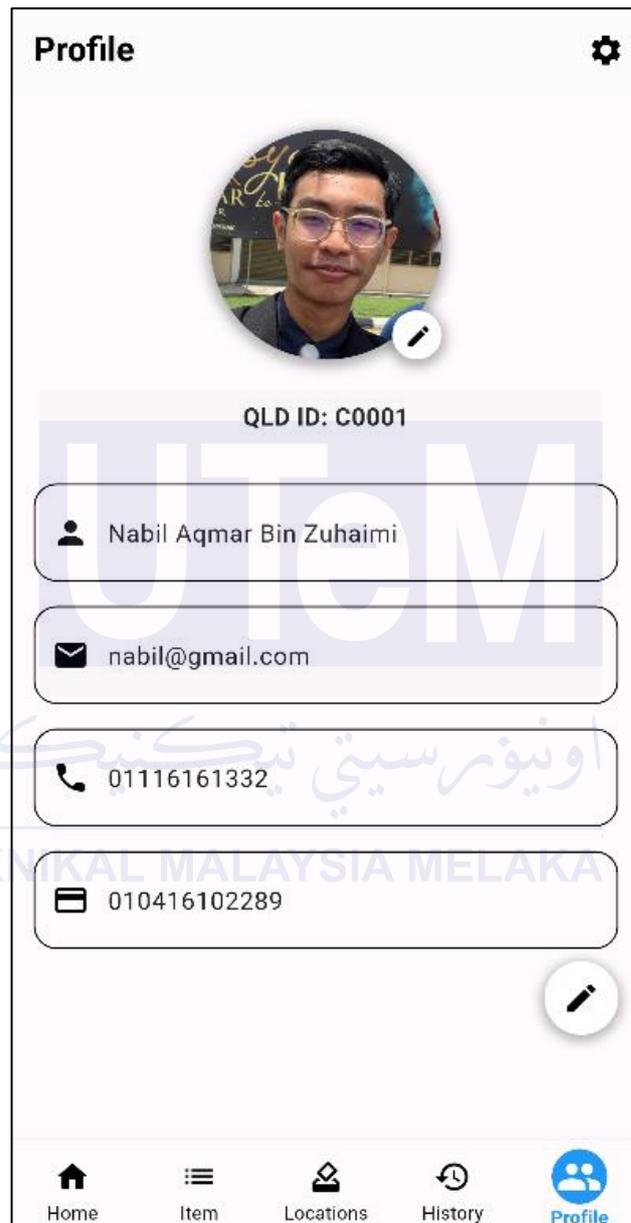


Figure 4.11: Profile Page for Courier and Recipient

- **The Location Page (Admin)** allows to select locker before Scan.

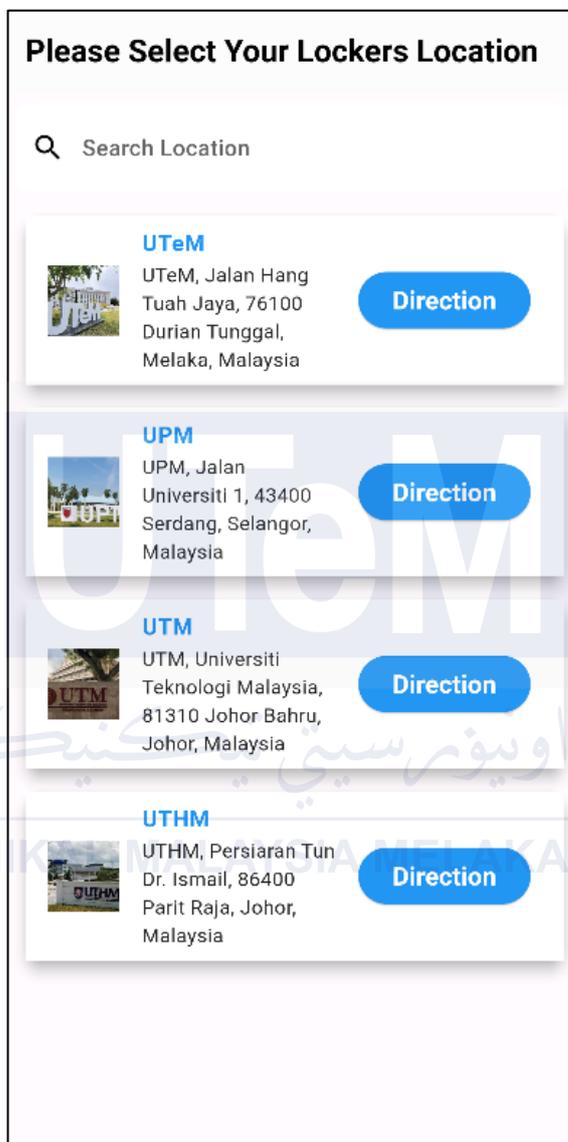


Figure 4.12: Location Page for Admin

- **The Locker Map Page (Courier and Recipient)** allows to select locker before Scan.

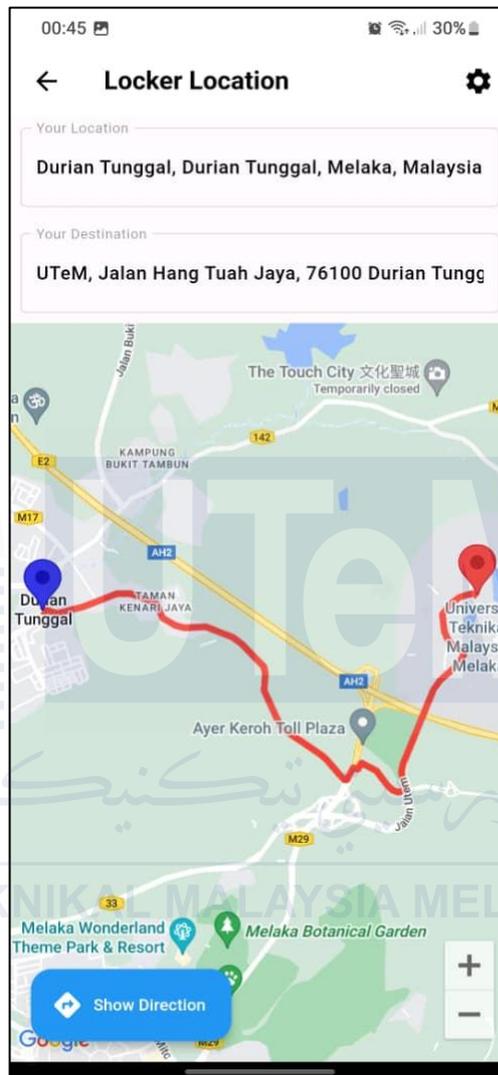


Figure 4.13: Locker Map Page Mobile App for Courier and Recipient

2) Web-Based Administration Portal:

- **The Dashboard (Admin)** provides an overview of system status, including locker availability and recent activity.

Item No.	Resit ID	Register Date	Courier Id	Due Count (Day)	Status
1	IM0004	2024-06-12 15:26:35	C0001	13	Pending
2	IM0012	2024-06-16 10:37:36	C0002	9	Pending
3	IM0013	2024-06-20 10:26:01	C0001	5	Pending
4	IM0015	2024-06-20 10:31:51	C0001	5	Pending
5	IM0018	2024-06-25 04:33:32	C0001	0	Pending

Figure 4.14: Dashboard Page for Admin

- **The Location List (Admin)** displays a list of all locker locations.

Item No.	Location Id	Location Name	Location Address	Image	Action
1	L0001	UTeM	UTeM, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia		
2	L0002	UPM	UPM, Jalan Universiti 1, 43400 Serdang, Selangor, Malaysia		
3	L0003	UTM	UTM, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia		

Figure 4.15: Location List Page for Admin

- **Figure 4.16 Locker List (Admin)** allows administrators to update and remove Locker.

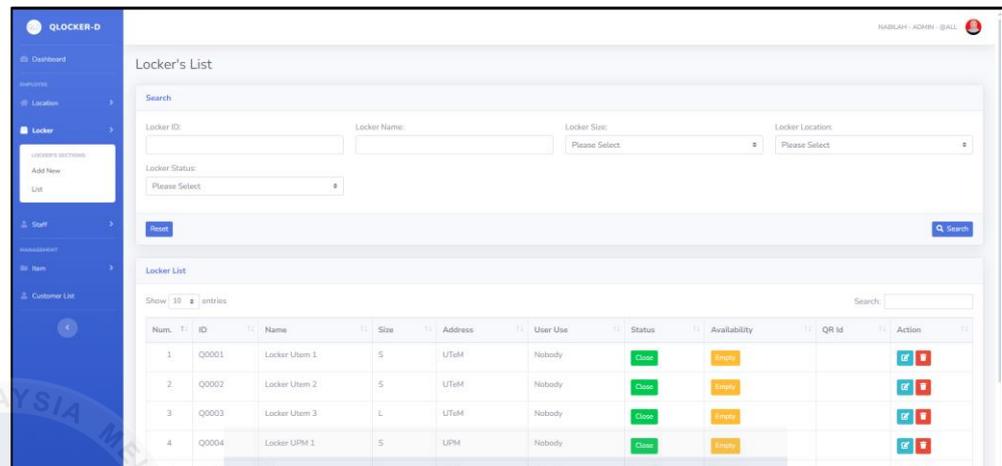


Figure 4.16: Locker List Page for Admin

- **Figure 4.17 Staff List (Admin)** manages courier and admin accounts.

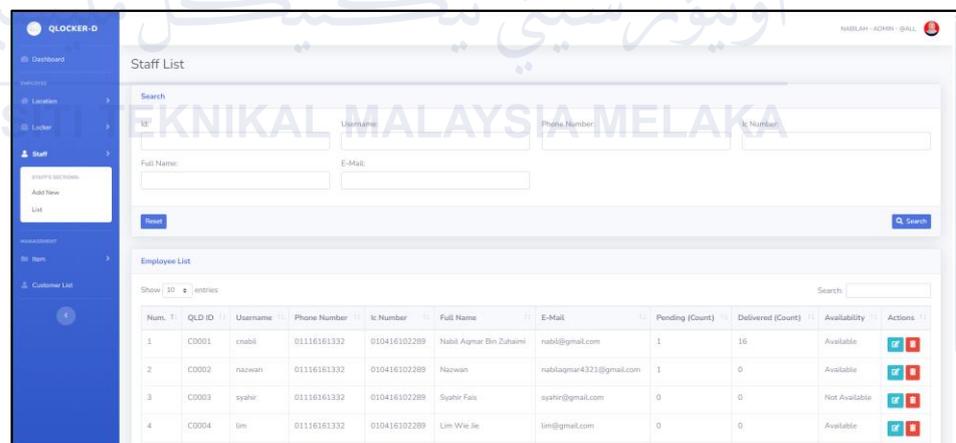
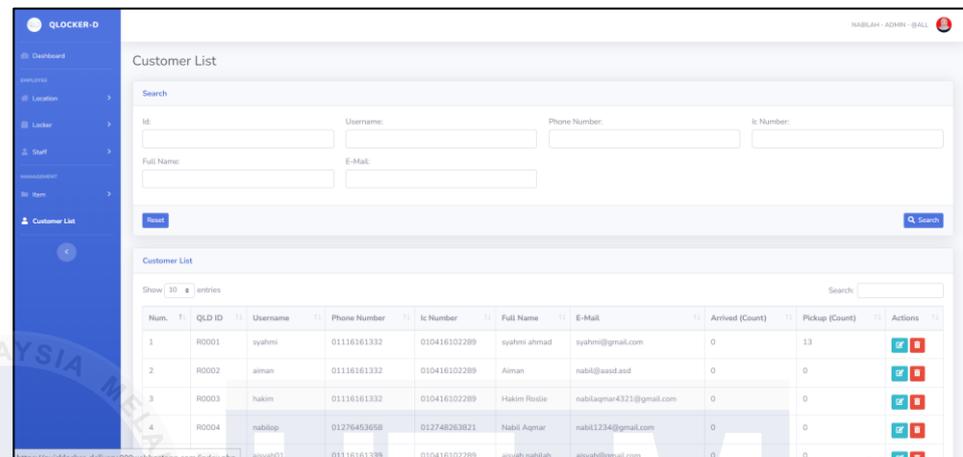


Figure 4.17: Staff List Page for Admin

- **Figure 4.18 Customer List (Admin)** manages recipient accounts.



Customer List

Search

ID: Username: Phone Number: IC Number:

Full Name: E-Mail:

Customer List

Show 10 entries

Num	QLD ID	Username	Phone Number	IC Number	Full Name	E-Mail	Arrived (Count)	Pickup (Count)	Actions
1	R0001	syahmi	01116161332	010416102289	Syahmi Ahmad	syahmi@gmail.com	0	13	<input type="checkbox"/> <input type="checkbox"/>
2	R0002	aiman	01116161332	010416102289	Aiman	nabil@asad.asd	0	0	<input type="checkbox"/> <input type="checkbox"/>
3	R0003	hakim	01116161332	010416102289	Hakim Roslie	nabilagmar4321@gmail.com	0	0	<input type="checkbox"/> <input type="checkbox"/>
4	R0004	nabiltop	01276453658	012748263921	Nabil Agmar	nabil1234@gmail.com	0	0	<input type="checkbox"/> <input type="checkbox"/>

Figure 4.18: Customer List Page for Admin

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

b) Input Design:

The input design focuses on the various types of data that users will need to enter the system, and the methods used to ensure data accuracy and consistency.

1) Mobile Applications:

- **The Login Page (Admin, Courier, Recipient)** users enter credentials to access the system and Reset Password

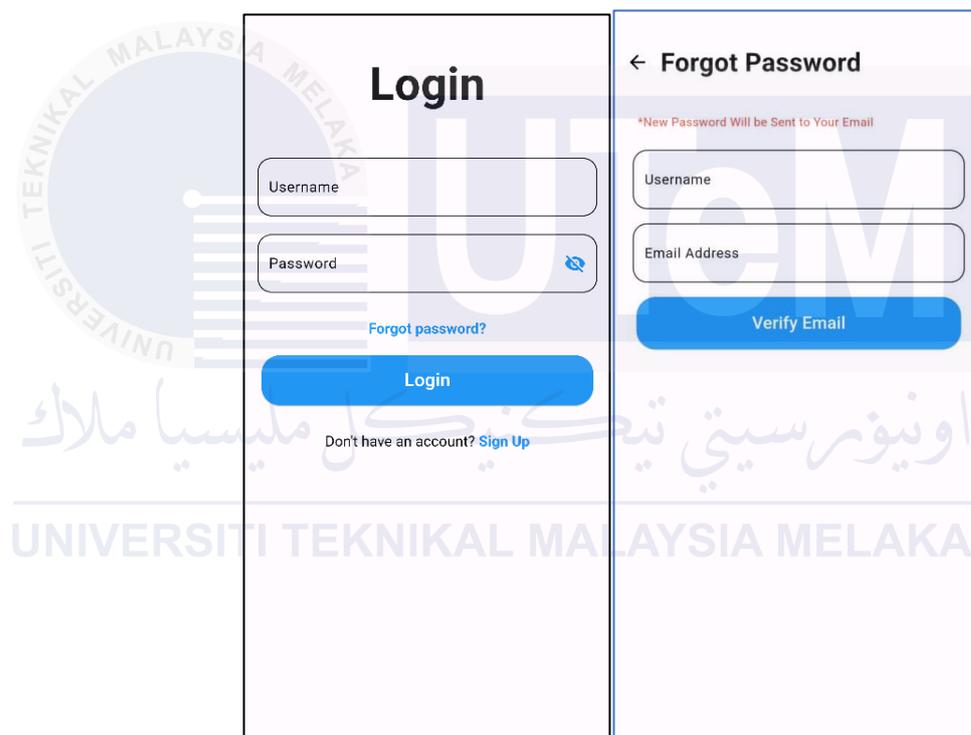


Figure 4.19: Login Page for Admin, Courier and Recipient and Forgot Password

- **Figure 4.20 Register Page (Recipient)** allows new recipients to create an account.

← **Sign Up**

Full Name

IC Number

Email Address

Phone Number

Username

Password

Confirm Password

Agree with terms and conditions

Please agree with terms and conditions

Register

Figure 4.20: Register Page for Recipient

- **Figure 4.21 QR Code Scanner Page (Admin)** used for parcel retrieval by scanning QR codes.



Figure 4.21: QrCode Scanner Page for Admin

- **Figure 4.22 Profile Page (Staff and Customer)**, shows users can update their personal information, such as name, email, and phone number.



Figure 4.22: Profile Page for Staff and Customer

2) Web-Based Administration Portal:

- **Figure 4.23 Add New Employee Page (Admin)** shows form fields include name, email, role (admin/courier), phone number, and status (active/inactive).

Figure 4.23: Add new Employee Page for Admin

- **Figure 4.24 Update Profile Staff Page (Admin)** allows admins to update staff profiles

Figure 4.24: Update Profile Staff Page for Admin

- **The Add New Location Page for Admin** show form fields include location ID, name, address, and details about the location.

Figure 4.25: Add New Location Page for Admin

- **The Add New Locker Page (Admin)** allows admins to add new lockers with details such as locker ID, location ID, and technical specifications.

Figure 4.26: Add New Locker Page for Admin

- **The Update Locker Page (Admin)** allows updates to locker information.

The screenshot displays the 'Update Locker' page within the QLOCKER-D admin dashboard. The page features a sidebar on the left with navigation options like Dashboard, Location, Locker, Staff, Item, and Customer List. The main content area contains a form titled 'Update Locker' with the following fields: Locker ID (text input with value 'Q0001'), Locker Name (text input with value 'Locker Utam 1'), Locker Size (dropdown menu with value 'S'), Locker Location (dropdown menu with value 'UTeM'), Locker Status (dropdown menu with value 'Close'), and Availability (dropdown menu with value 'Empty'). At the bottom of the form, there are 'Back' and 'Update' buttons. The footer of the page includes the text 'Copyright © QuickLocker Delivery 2024'.

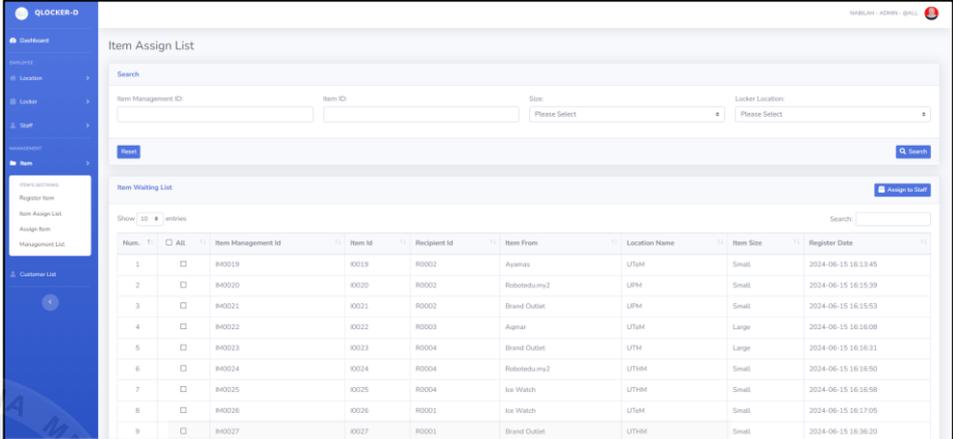
Figure 4.27: Update Locker Page for Admin

- **The Register Item Page (Admin)** shows form fields include recipient ID, locker ID, courier ID, and delivery details.

The screenshot displays the 'Register Item' page within the QLOCKER-D admin dashboard. The page features a sidebar on the left with navigation options like Dashboard, Location, Locker, Staff, Item, and Customer List. The main content area contains a form titled 'Register Item' with the following fields: Customer Id (e.g. R0001) (text input), Item From (Sender Address/Company) (text input), Locker Location (dropdown menu with value 'Please Select'), and Size (dropdown menu with value 'S'). At the bottom of the form, there are 'View Assignment List' and 'Register Item' buttons. The footer of the page includes the text 'Copyright © QuickLocker Delivery 2024'.

Figure 4.28: Register Item Page for Admin

- **The Item Assign List to Staff Page (Admin)** allows admin to assign items to couriers.



Item Assign List

Search

Item Management ID: Item ID: Size: Locker Location:

Item Waiting List

Show: 10 entries

Num.	<input type="checkbox"/>	Item Management Id	Item Id	Recipient Id	Item From	Location Name	Item Size	Register Date
1	<input type="checkbox"/>	IM0019	0019	R0002	Avanas	UTM	Small	2024-06-15 16:13:45
2	<input type="checkbox"/>	IM0020	0020	R0002	Robotek.my2	LPM	Small	2024-06-15 16:15:39
3	<input type="checkbox"/>	IM0021	0021	R0002	Brand Outlet	LPM	Small	2024-06-15 16:15:53
4	<input type="checkbox"/>	IM0022	0022	R0003	Agman	UTM	Large	2024-06-15 16:16:08
5	<input type="checkbox"/>	IM0023	0023	R0004	Brand Outlet	UTM	Large	2024-06-15 16:16:31
6	<input type="checkbox"/>	IM0024	0024	R0004	Robotek.my2	UTM	Small	2024-06-15 16:16:50
7	<input type="checkbox"/>	IM0025	0025	R0004	Ice Watch	UTM	Small	2024-06-15 16:16:58
8	<input type="checkbox"/>	IM0026	0026	R0001	Ice Watch	UTM	Small	2024-06-15 16:17:05
9	<input type="checkbox"/>	IM0027	0027	R0001	Brand Outlet	UTM	Small	2024-06-15 16:16:20

Figure 4.29: Item Assign List to Staff Page for Admin

- **Items Selected Assign to Staff Page (Admin):** Confirms assignments.

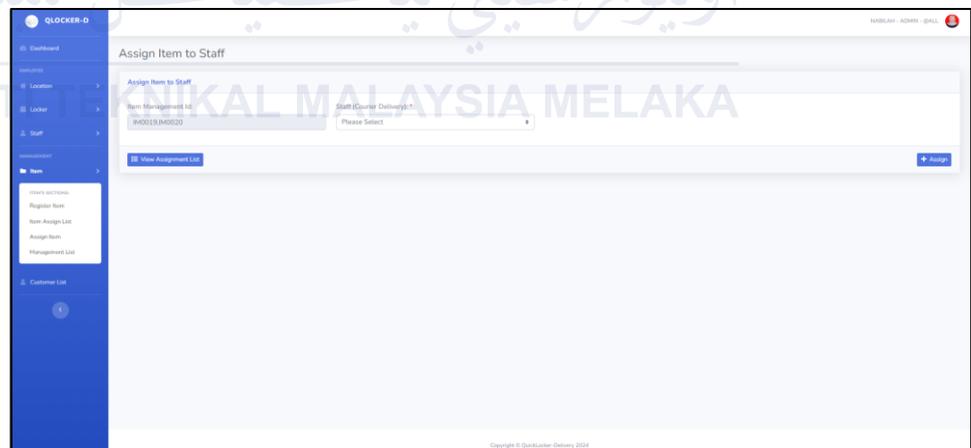


Figure 4.30: Items Selected Assign to Staff Page for Admin

- **Assign One Item to Staff Page (Admin):** Individual item assignment.

Figure 4.31: Assign One Item to Staff Page for Admin

- **The Item Details Page (Admin):** Provides detailed information about each item.

Figure 4.32: Item details Page for Admin

- **The Profile Page (Admin):** Manages admin account settings.

Figure 4.33: Profile Page for Admin

3) Validation Rules:

- Staff ID and Locker ID: Must be unique and follow the predefined format (e.g., Staff ID: C0001, Locker ID: Q0001).

- Parcel ID: Must match the format defined by the system (e.g. I0001).

- User Information: Email must be in a valid format; phone numbers must be numeric and of a specific length.

- Location Details: Address must be valid and detailed enough for accurate identification.

c) Output Design

The output design details the types of information that the system will generate for users, including reports and notifications.

1) Reports:

- **Figure Detail Reports:** Comprehensive reports on individual parcels, including delivery status, timestamps, and user interactions.

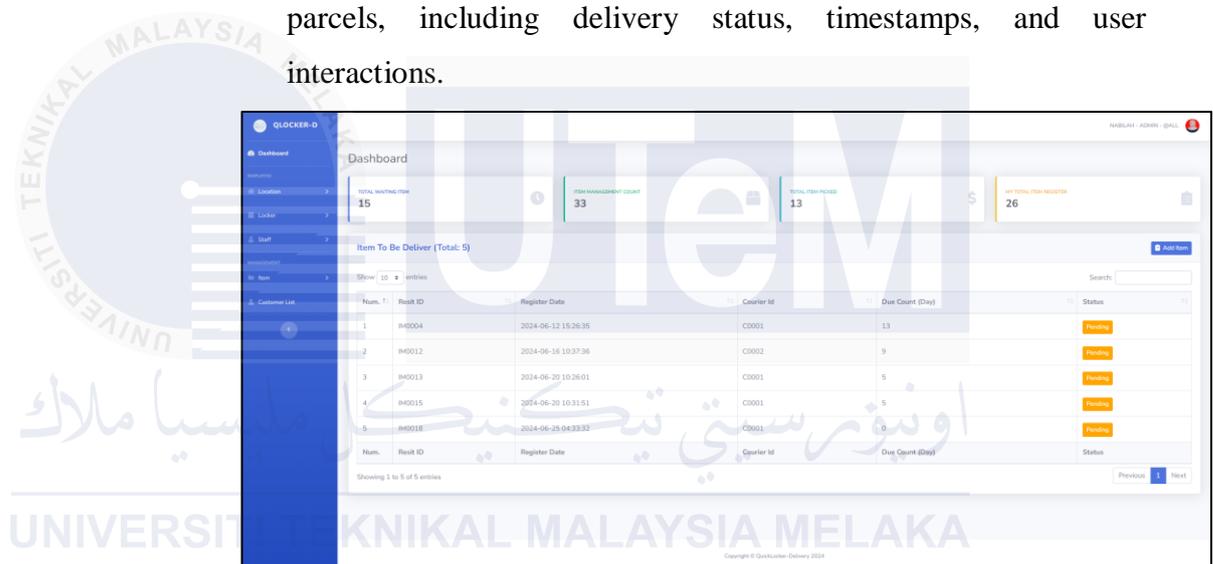


Figure 4.34: Report Page for Admin

- Figure belows show summary reports that Aggregated data on locker usage, number of deliveries per period, and user activity.

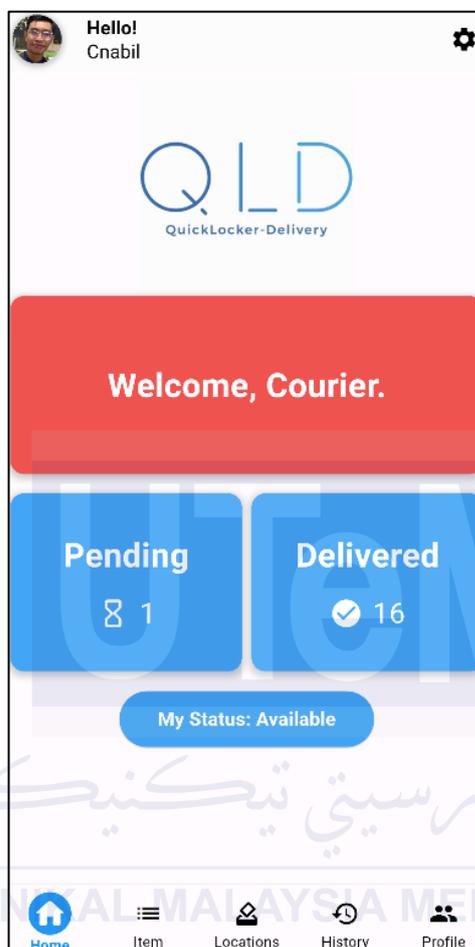


Figure 4.35: Report Page for Courier

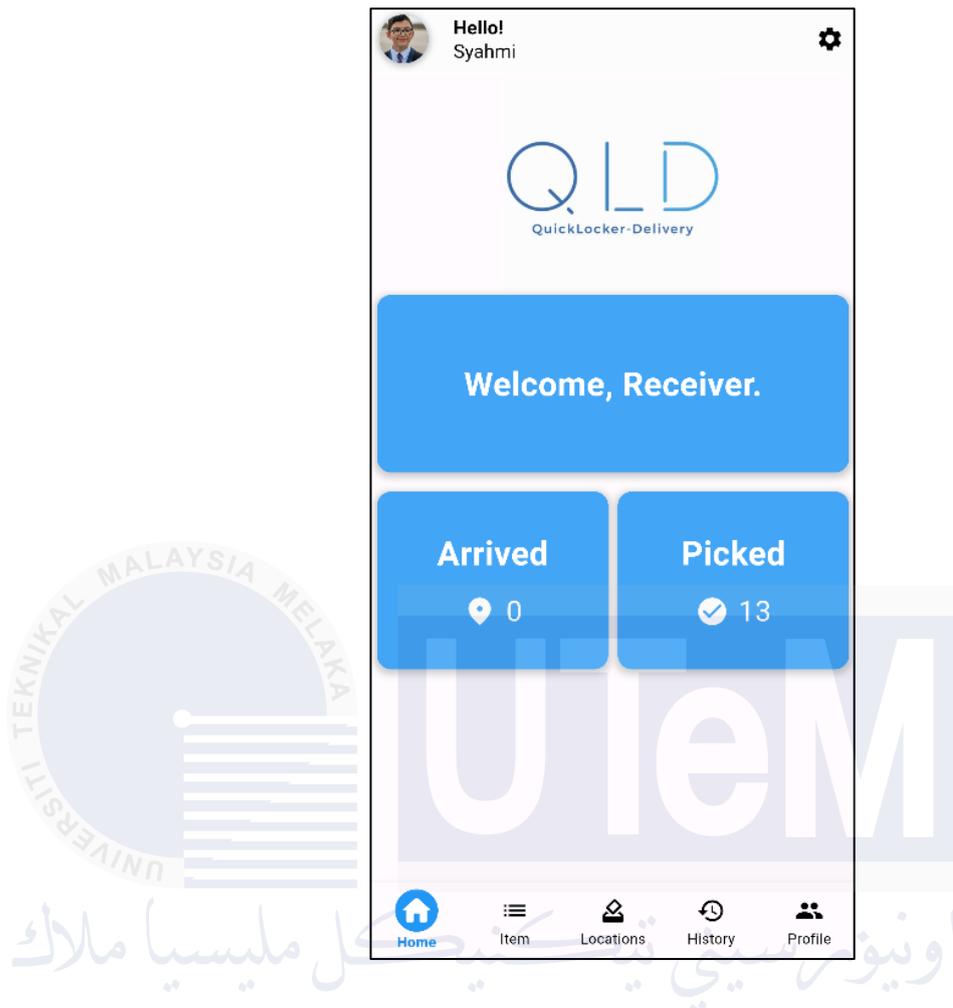


Figure 4.36: Report Page for Recipient

- The figure Management List Report Page for Admin and convert to PDF, can be filtered by entering start date and end date in the text fields.

Num.	Receipt Id	PIC Id	Recipient Id	Courier Id	Item Id	Item From	Locker Id	Register Date	Delivery Date	Pickup Date	Status	Action
1	IM0013	A0002	R0001	C0001	00013	Ice Watch	Q0001	2024-09-03 10:53:59	2024-09-02 21:56:52	2024-09-03 10:53:59	Arrived	[Icon]
2	IM0017	A0002	R0001	C0001	00017	Sevending Brand	Q0002	2024-09-03 10:52:30	2024-09-03 10:52:30	0000-00-00 00:00:00	Arrived	[Icon]

Num	Realt Id	PIC Id	Recipient Id	Courier Id	Item Id	Item From	Locker Id	Register Date	Delivery Date	Pickup Date	Status
1	IM0005	A0001	R0001	C0001	00005	Agmar	00003	2024-06-28 00:49:43	2024-06-28 00:49:43	2024-06-28 00:00:00	Success
2	IM0032	A0001	R0001	C0001	00032	Agmar	00001	2024-06-28 00:48:27	2024-06-28 00:46:30	2024-06-28 00:48:27	Success
3	IM0009	A0002	R0001	C0001	00009	Ayam Masak Merah		2024-06-28 00:39:55	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
4	IM0034	A0001	R0001	C0001	00034	POPULAR Bookstore		2024-06-26 12:05:13	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
5	IM0004	A0001	R0001	C0001	00004	Robotedu.my2	00002	2024-06-26 00:15:45	2024-06-12 23:26:53	2024-06-12 00:00:00	Success
6	IM0033	A0001	R0001		00033	Agmar		2024-06-16 19:43:59	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
7	IM0031	A0002	R0004		00031	POPULAR Bookstore		2024-06-16 00:38:13	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
8	IM0030	A0002	R0001		00030	POPULAR Bookstore		2024-06-16 00:37:47	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
9	IM0029	A0002	R0001		00029	POPULAR Bookstore		2024-06-16 00:37:38	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
10	IM0028	A0002	R0001		00028	POPULAR Bookstore		2024-06-16 00:37:23	0000-00-00 00:00:00	0000-00-00 00:00:00	Success
11	IM0027	A0002	R0001		00027	Brand Outlet		2024-06-16 00:36:20	0000-00-00 00:00:00	0000-00-00 00:00:00	Success

Figure 4.37: Management List Report Page for Admin and convert to PDF

2) Notifications:

- **Mobile Notifications:** Real-time alerts for recipients and couriers about parcel status changes, locker assignments, and system updates.

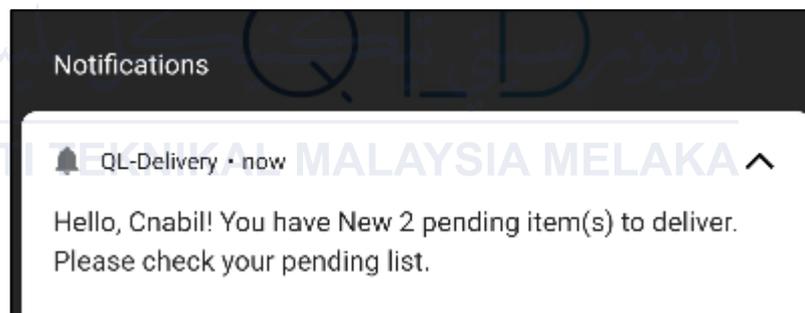


Figure 4.38: Notification Popup for Courier and Recipient

Conclusion

In conclusion, by refining these aspects of the UI design, the IoT-based locker system aims to provide a user-friendly and efficient interface that meets the needs of all users while ensuring accurate data entry and useful output generation.

4.2.3 Database Design

4.2.3.1 Conceptual and logical database design

The database design for the IoT-based locker system aims to provide a structured and efficient way to store and manage data related to users, lockers, parcels, and transactions. The Entity-Relationship Diagram (ERD) is used to model the logical structure of the database, capturing the relationships between different entities. This section introduces the logical data model (LDM), defines and constructs the ER diagrams, and provides a data dictionary along with normalization details.

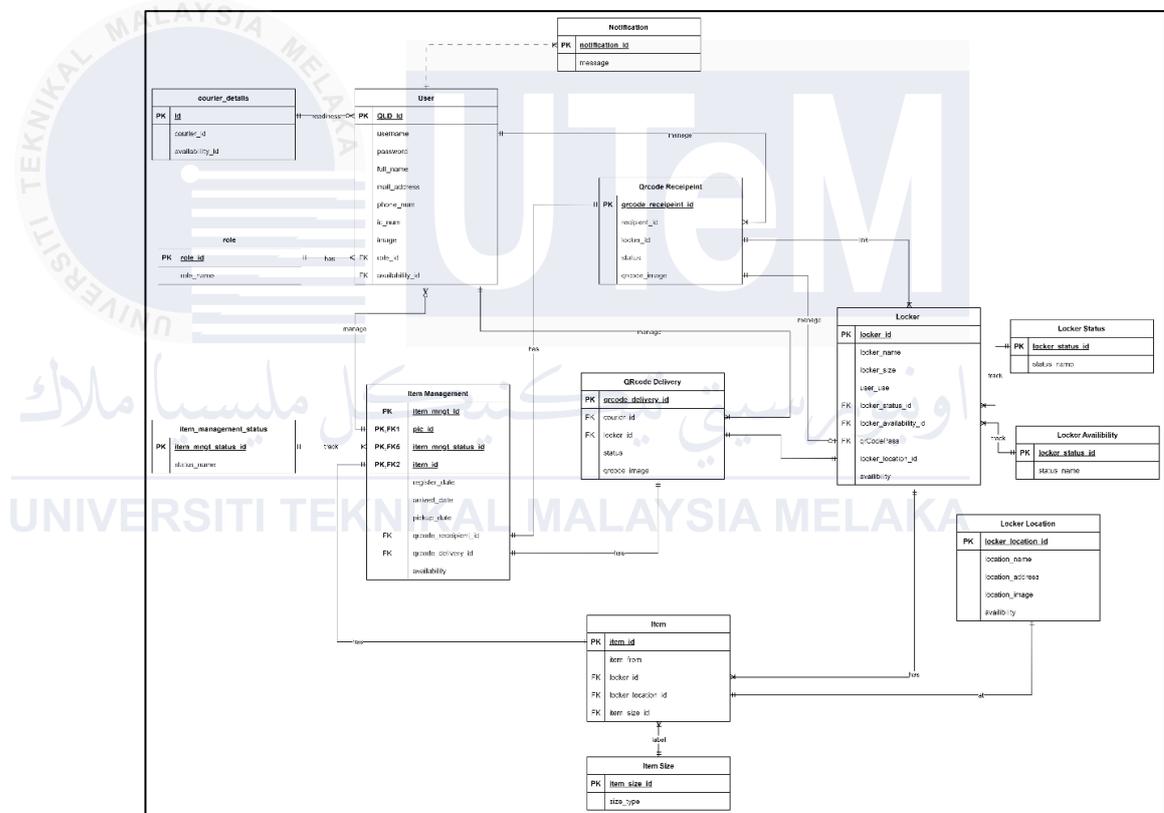


Figure 4.39: Entity Relationship Diagram (ERD)

Figure 4.40 represented by the Entity-Relationship Diagram (ERD), serves as a blueprint for designing the database. It visually represents the key entities, their attributes, and the relationships between them. This model is essential for ensuring data consistency, integrity, and efficient data retrieval.

a) ER Diagram Explanation:

The ERD provided captures the following key entities and their relationships:

1. User

- Attributes: user_id (PK), name, phone, email, role_id
- Relationships: Each user can be assigned one role, and a role can be associated with many users.

2. Role

- Attributes: role_id (PK), role_name
- Relationships: A role can be assigned to many users.

3. Locker

- Attributes: locker_id (PK), locker_location_id, locker_status_id, locker_availability_id
- Relationships: Each locker is located at a specific locker location and has a status and availability indicator.

4. Locker Location

- Attributes: locker_location_id (PK), location_name, address
- Relationships: A location can host multiple lockers.

5. Locker Status

- Attributes: locker_status_id (PK), status_name

- Relationships: A status can be assigned to multiple lockers.

6. Locker Availability

- Attributes: locker_availability_id (PK), availability_name
- Relationships: An availability status can be assigned to multiple lockers.

7. Item

- Attributes: item_id (PK), item_size_id, item_management_status_id
- Relationships: Each item has a size and a management status.

8. Item Size

- Attributes: item_size_id (PK), size_name
- Relationships: A size can be assigned to multiple items.

9. Item Management Status

- Attributes: item_management_status_id (PK), status_name
- Relationships: A management status can be assigned to multiple items.

10. Item Management:

- Attributes: item_management_id (PK), item_id, user_id
- Relationships: This entity represents the association between items and users (either couriers or recipients).

11. QRCode Delivery:

- Attributes: qr_code_id (PK), item_id, qr_code
- Relationships: Each QR code is associated with one item.

12. QRCode Recipient

- Attributes: qr_code_id (PK), user_id, qr_code
- Relationships: Each QR code is assigned to a recipient for item retrieval.

13. Courier Details

- Attributes: id (PK), courier_id, availability_id
- Relationships: Each courier detail is associated with a user. Foreign key: courier_id references users.qld_id.

b) Business Rules

- Uniqueness: User IDs, Locker IDs, and QR Codes must be unique.
- Roles: Users can have roles such as admin or courier, defined in the Role entity.
- Locker Management: Lockers are managed based on their location, status, and availability.
- Item Tracking: Items are tracked based on their size and management status, and each item is associated with a unique QR code for secure delivery and retrieval.
- Normalization: The database design adheres to normalization principles to eliminate redundancy and ensure data integrity.

c) Data Dictionary

Table 4.1: Courier_details

Attribute	Data Type	Description
id	int(11)	Primary key, auto-incremented
courier_id	varchar(100)	Foreign key, references users.qld_id
availability_id	varchar(100)	Availability identifier

Table 4.2: Item

Attribute	Data Type	Description
item_id	varchar(100)	Primary key
item_from	varchar(100)	Sender of the item
locker_location_id	varchar(100)	Foreign key, references locker_location.locker_location_id
locker_id	varchar(100)	Foreign key, references locker.locker_id (nullable)
item_size_id	int(11)	Foreign key, references item_size.item_size_id

Table 4.3: Item_management

Attribute	Data Type	Description
item_mngt_id	varchar(100)	Primary key
pic_id	varchar(100)	Foreign key, references users.qld_id
item_id	varchar(100)	Foreign key, references item.item_id
item_mngt_status_id	int(11)	Foreign key, references item_management_status.item_mngt_status_id
register_date	timestamp	Timestamp of registration, default current timestamp
arrived_date	datetime	Date and time of item arrival
pickup_date	datetime	Date and time of item pickup
qrcode_recipient_id	varchar(100)	Foreign key, references qrcode_recipient.qrcode_recipient_id
qrcode_delivery_id	varchar(100)	Foreign key, references qrcode_delivery.qrcode_delivery_id (nullable)
availability	int(11)	Availability status, 1 = Exist, 2 = Deleted

Table 4.4: Item_management_status

Attribute	Data Type	Description
item_mngt_status_id	int(11)	Primary key, auto-incremented
status_name	varchar(100)	Status name

Table 4.5: Item_size

Attribute	Data Type	Description
item_size_id	int(11)	Primary key
size_type	varchar(100)	Size type

Table 4.6: Locker

Attribute	Data Type	Description
locker_id	varchar(100)	Primary key
locker_name	varchar(100)	Name of the locker
locker_size	varchar(100)	Size of the locker
user_use	varchar(100)	User using the locker (nullable)
locker_location_id	varchar(100)	Foreign key, references locker_location.locker_location_id
locker_status_id	int(11)	Foreign key, references locker_status.locker_status_id
locker_availability_id	int(11)	Foreign key, references locker_availability.locker_availability_id
qrCodePass	varchar(100)	QR code pass, references qrcode_recipient.qrcode_recipient_id (nullable)
availability	int(11)	Availability status

Table 4.7: Locker_availability

Attribute	Data Type	Description
locker_availability_id	int(11)	Primary key, auto-incremented
availability	varchar(100)	Availability status

Table 4.8: Locker_location

Attribute	Data Type	Description
locker_location_id	varchar(100)	Primary key
location_name	varchar(100)	Name of the location
location_address	varchar(255)	Address of the location
location_image	mediumblob	Image of the location
availability	int(11)	Availability status

Table 4.9: Locker_status

Attribute	Data Type	Description
locker_status_id	int(11)	Primary key, auto-incremented
status_name	varchar(100)	Status name

Table 4.10: Qrcode_delivery

Attribute	Data Type	Description
qrcode_delivery_id	varchar(100)	Primary key
courier_id	varchar(100)	Foreign key, references users.qld_id
locker_id	varchar(100)	Foreign key, references locker.locker_id
status	varchar(100)	Status of the QR code
qrcode_image	blob	Image of the QR code

Table 4.11: Qrcode_recipient

Attribute	Data Type	Description
qrcode_recipient_id	varchar(100)	Primary key
recipient_id	varchar(100)	Foreign key, references users.qld_id
locker_id	varchar(100)	Foreign key, references locker.locker_id
status	varchar(100)	Status of the QR code
qrcode_image	blob	Image of the QR code

Table 4.12: Role

Attribute	Data Type	Description
role_id	int(11)	Primary key
role_name	varchar(100)	Name of the role

Table 4.13: Users

Attribute	Data Type	Description
qld_id	varchar(100)	Primary key
username	varchar(100)	Username
password	varchar(200)	Password
full_name	varchar(255)	Full name of the user
mail_address	varchar(100)	Mail address
phone_num	varchar(100)	Phone number
ic_num	varchar(100)	Identification card number
image	mediumblob	Image of the user
role_id	int(11)	Foreign key, references role.role_id
availability	int(11)	Availability status

Normalization:

The database design follows the principles of normalization to ensure that data is organized efficiently and redundancy is minimized. Key normalization steps include:

- First Normal Form (1NF): Ensuring that each table has a primary key and that each column contains atomic, indivisible values.
- Second Normal Form (2NF): Ensuring that all non-key attributes are fully functionally dependent on the primary key.
- Third Normal Form (3NF): Ensuring that all attributes are only dependent on the primary key, eliminating transitive dependencies.

By adhering to these principles, the database is designed to be robust, scalable, and maintainable, supporting the efficient operation of the IoT-based locker system.

4.3 Detailed Design

This section delves into the detailed design of the IoT-based locker system, emphasizing the logic and approach taken to satisfy the system requirements. The software design will be presented using Object-Oriented Analysis and Design (OOAD) principles, utilizing Unified Modeling Language (UML) for clarity and precision.

4.3.1 Software Design

The software design is structured around classes that encapsulate the system's functionality. Each class is defined with its responsibilities, attributes, and methods. Sequence diagrams will illustrate the interactions between objects to achieve specific tasks.

4.3.1.1 Sequence Diagram

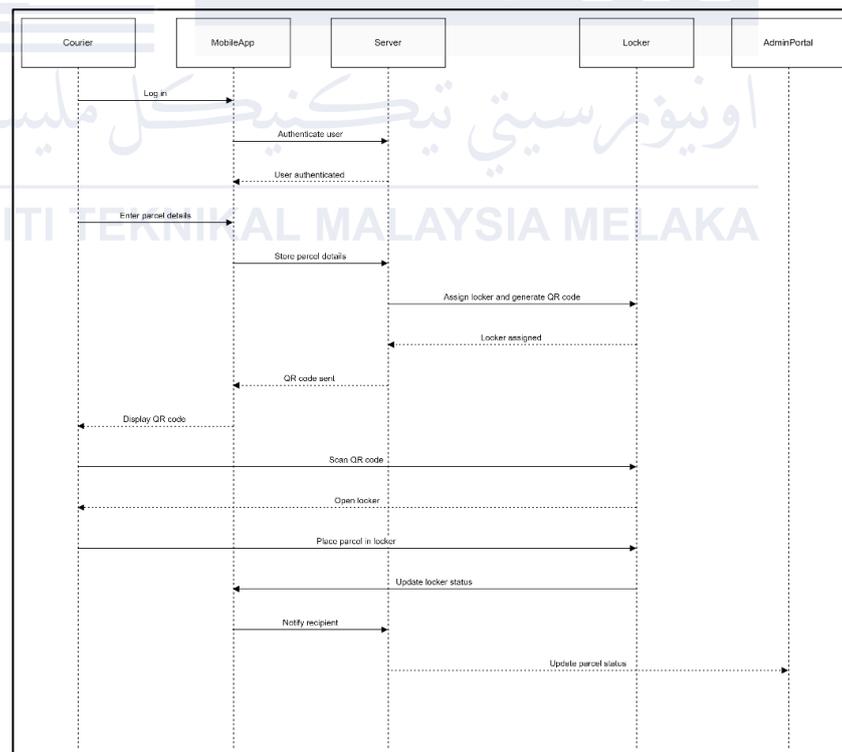


Figure 4.41: Sequence Diagram: Parcel Delivery Process

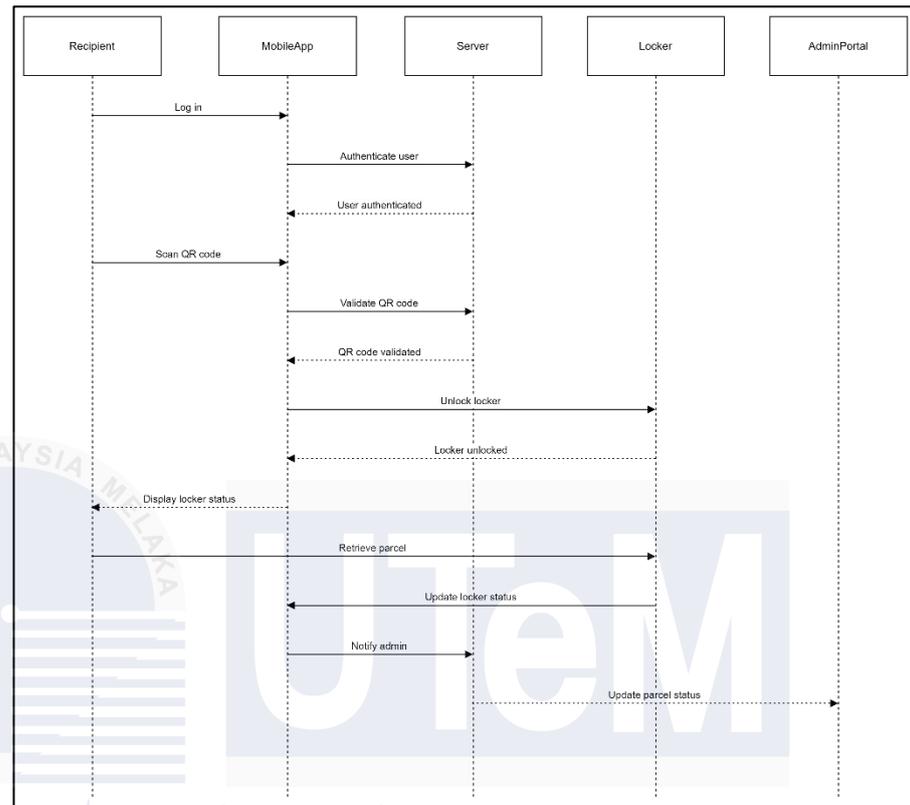


Figure 4.42: Sequence Diagram: Parcel Retrieval Process

4.3.1.2 State Diagram

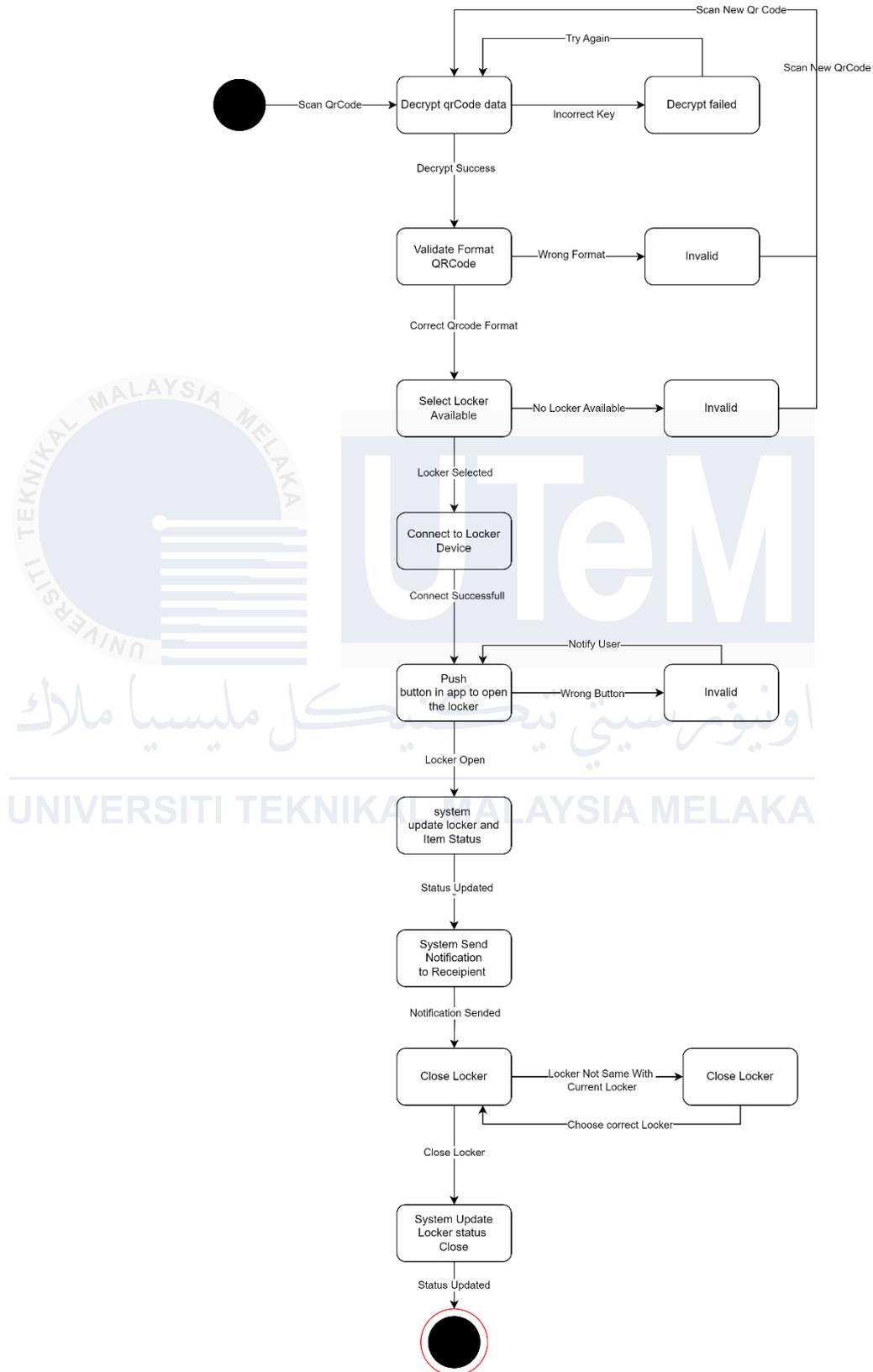


Figure 4.43: State Diagram: Open Locker Process for Courier and Recipient

4.3.1.3 UML Diagram

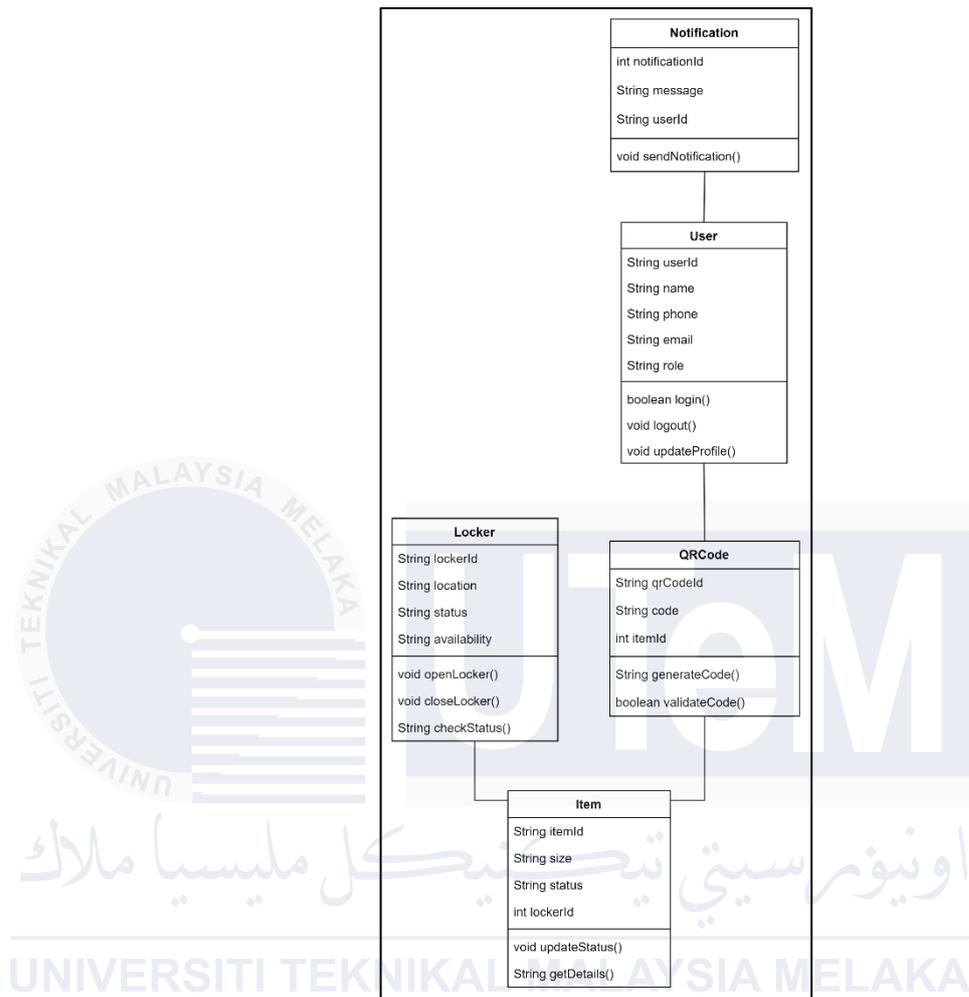


Figure 4.44: UML class diagram

4.3.2 Hardware Design

The hardware design of the IoT-based locker system involves integrating several key components to create a functional and efficient system. The primary components include an Arduino UNO R3 microcontroller, HC-05 Bluetooth module, servo motors, and a power supply. The schematic diagram provided illustrates the connections and interactions between these components.

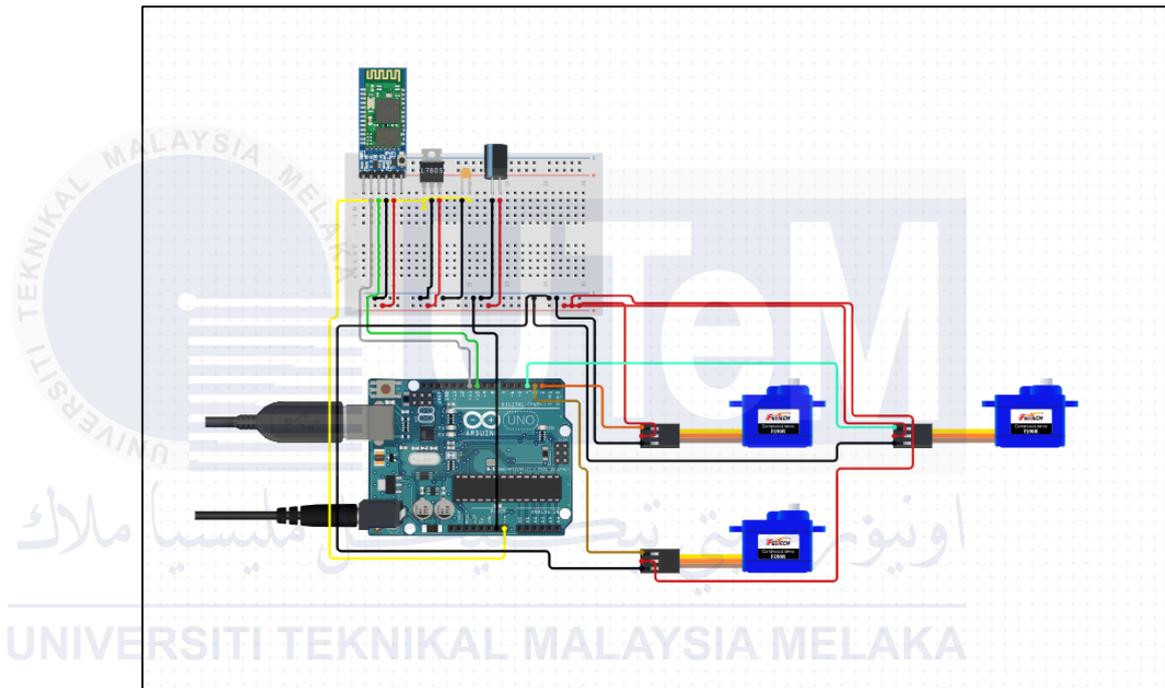


Figure 4.45: Hardware Diagram

4.3.2.1 Key Components and Connections:

1. Arduino UNO R3 Microcontroller:

- **Function:** Acts as the central processing unit of the system, controlling the servo motors and managing communication between components.
- **Connections:**
 - Connected to the power supply via the USB port.
 - Digital I/O pins connected to control the servo motors.
 - Serial pins connected to the HC-05 Bluetooth module for wireless communication.

2. HC-05 Bluetooth Module:

- **Function:** Facilitates wireless communication between the Arduino and the smartphones used by administrators, couriers, and recipients.
- **Connections:**
 - VCC and GND pins connected to the power supply rails on the breadboard.
 - TXD (transmit) pin connected to the RX (receive) pin of the Arduino.
 - RXD (receive) pin connected to the TX (transmit) pin of the Arduino.

3. Servo Motors:

- **Function:** Control the opening and closing of the lockers.
- **Connections:**
 - Power lines (red) connected to the positive rail of the breadboard.
 - Ground lines (black) connected to the negative rail of the breadboard.
 - Signal lines (yellow) connected to digital I/O pins on the Arduino.

4. Breadboard and Power Supply:

- **Function:** Distributes power to the components and provides a platform for connecting the circuit.
- **Connections:**
 - Positive and negative rails of the breadboard connected to the 5V and GND pins of the Arduino, respectively.
 - Components like the HC-05 Bluetooth module and servo motors connected to the breadboard for power and signal distribution.

4.3.2.2 Explanation of the Diagram:

In Figure 4.44: Hardware Diagram illustrates the integration of the Arduino UNO R3 with the HC-05 Bluetooth module and four servo motors. The Bluetooth module is used for wireless communication, allowing users to interact with the locker system via a mobile application. The servo motors are responsible for opening and closing the lockers based on signals received from the Arduino. The power is supplied through the USB port of the Arduino, which then distributes it to the other components via the breadboard.

The schematic ensures that all components are correctly powered and able to communicate effectively. The use of an Arduino UNO R3 provides a robust and flexible platform for controlling the locker system, while the HC-05 Bluetooth module enables seamless wireless interactions. The servo motors are connected in a way that allows them to be individually controlled, ensuring precise operation of each locker.

This hardware design forms the foundation of the IoT-based locker system, enabling it to function efficiently and reliably while providing a user-friendly experience.

4.3.3 Physical Database Design

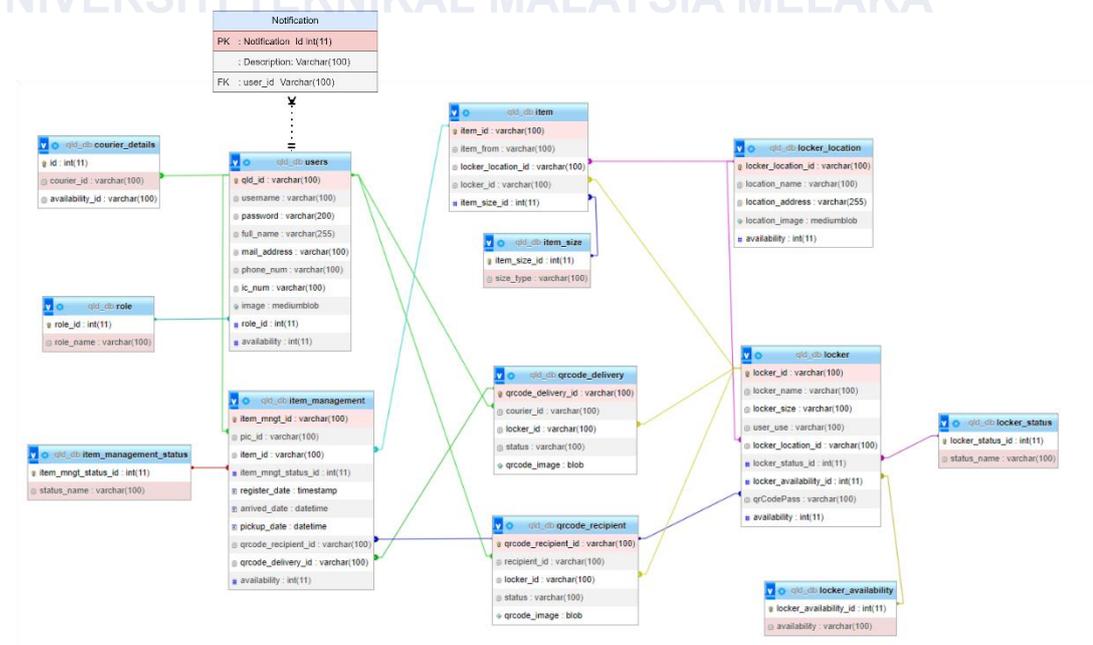


Figure 4.46: Physical ERD Diagram

Explanation of Constraints and Indexes for Figure 4.45:

- Primary Keys: Ensure each record in the table is unique.
- Foreign Keys: Ensure referential integrity by linking records between tables.
- Indexes: Speed up the retrieval of records and improve query performance.

- Auto Increment: Automatically generate unique values for the primary key columns where necessary.

Business Rules:

- Constraints: Ensure data integrity and consistency across tables.
- Validation: Enforced by defining appropriate data types and constraints in the table schema.

- File Organization: Using InnoDB as the storage engine for transactions, foreign keys, and row-level locking.

This design ensures the physical database structure is optimized for performance, data integrity, and maintainability.

4.4 Conclusion

In this chapter, the detailed software design of the QR Code and Smart Locker System was meticulously elaborated. Object-Oriented Analysis and Design (OOAD) methodologies were employed to articulate the system architecture, including the creation of UML class diagrams, sequence diagrams, and the transition from logical design to physical database design with DDL statements for table creation.

The chapter began by outlining the software requirements and continued with a thorough description of the system's logical design. This comprised defining classes, their attributes, and their relationships. Dynamic interactions between different system components were illustrated using sequence diagrams. Lastly, logical structures were translated into physical database designs, complete with DDL scripts to implement the database schema.

Key accomplishments in this chapter included:

- Detailed Software Design: Comprehensive details on each class, their attributes, methods, and relationships were provided, ensuring a clear understanding of the system's structure.
- Dynamic Interaction Modeling: Sequence diagrams were used to depict the interactions and flow of operations within the system.
- Physical Database Design: Logical designs were translated into physical database tables, including necessary constraints and relationships, ensuring data integrity and efficiency.

Next Activities:

- Implementation: With the design phase completed, the next step involves coding the system according to the specifications laid out in this chapter. This will include developing the backend logic, database operations, and user interface components.
- Testing: Post-implementation, rigorous testing (unit, integration, system, and user acceptance testing) will be conducted to ensure the system meets all requirements and performs as expected under various conditions.
- Deployment: Once testing is successfully completed, the system will be deployed in a live environment. This phase will involve setting up the necessary infrastructure and ensuring a smooth transition for all users.
- Maintenance and Evaluation: After deployment, ongoing maintenance will be required to address any issues that arise and to make continuous improvements based on user feedback and technological advancements.

By following this structured approach, a robust and efficient QR Code and Smart Locker System is expected to be delivered, meeting all specified requirements and providing a seamless user experience.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter provides an overview of the implementation phase of the QLD system. It describes the activities involved, the expected outcomes, and outlines the structure of the chapter.

5.2 Software Development Environment Setup

The development environment for the QLD system consists of various components to ensure seamless operation and integration. The setup includes client software, server software, hardware components, and network configuration.

1. Client Software



Figure 5.1: Flutter Logo

- **Mobile Applications:**
 - Developed using Flutter: Ensures cross-platform compatibility, primarily targeting Android.
 - Language: Dart.
 - Components:
 - Google Maps API: For location-based features.
 - File External: For local storage management.
 - Protocol: For communication protocols.

- Web Service Client: For server interaction and API requests.
- OneSignal: For push notifications.
- SQLite: Used for local data storage within the application, ensuring efficiency, offline functionality, and smooth user experience even without network connectivity.

2. Server Software



Figure 5.2: Xampp Logo

- **Web Server:**

- Developed using PHP: The server-side logic and web services are developed using PHP.
- Hosted on Apache Service (XAMPP): The web server is hosted using Apache, included in the XAMPP package for local development.
- Web Services: Handle client requests, providing APIs for the mobile applications and web applications to interact with the server.
- Database: MySQL Service is used for data storage and management. The database handles all data-related operations and stores information for the QLD system.

3. Hardware



Figure 5.3: Arduino Logo

- **IoT-Based Locker System:**

- Arduino Uno R3: For controlling the lockers.
- Bluetooth Module: For communication with mobile applications.
- Servo Motor: For locking/unlocking mechanisms

4. Network Setup

- **Communication Protocols:**

- Mobile Applications: Communicate with the server over HTTPS.
- IoT Locker System: Communicates with mobile applications via Bluetooth.

5. Notification Service



Figure 5.4: OneSignal Logo

- **OneSignal:** For sending notifications to mobile applications.



Figure 5.5: Firebase Logo

- **Firestore:** Used as the database for managing notifications.

6. Programming Language Used



Figure 5.6: PHP Logo

- **PHP:** For web application development.



Figure 5.7: Dart Logo

- **Dart:** For Flutter mobile application development.



Figure 5.8: C++ Logo

- **C++:** For Arduino Uno R3 programming.

5.2.1 Environment Architecture Diagram

Figure 5.1 visually represents the communication flow between the mobile applications, server, locker system, and the database. The components include tools and communication protocols in the mobile application, the web services and MySQL service in the server, and the Arduino with its components in the IoT locker system.

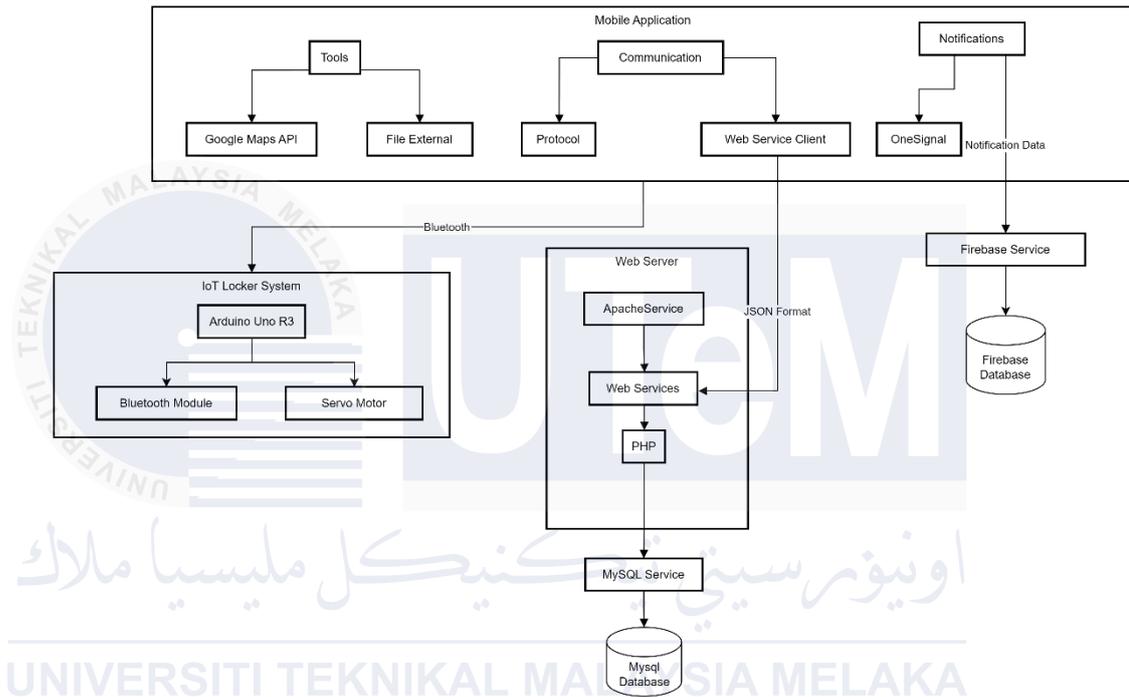


Figure 5.9: Environment Architecture Diagram

5.2.2 Environment Architecture Diagram Description

1. Development Machine:

- **Software:** Includes Android Studio IDE for Flutter and Visual Studio Code IDE for XAMPP for local server and database setup.
- **Purpose:** To write, test, and debug the application code before deployment.

2. XAMPP Server:

- **Components:** Apache server, MySQL database, and PHP.
- **Purpose:** Provides a local server environment to develop and test the backend services and database interactions.

3. Local MySQL Database:

- **Purpose:** To store and manage data during the development phase.

4. Flutter Development Environment:

- **Components:** Flutter SDK, Dart programming language, and supporting libraries.
- **Purpose:** To develop cross-platform mobile applications for both Android and iOS.

5. 000webhosting Server:

- **Purpose:** To host the web application and database for production. It provides a live environment where users can access the application.

6. Production MySQL Database:

- **Purpose:** To store and manage data in the live production environment.

7. Client Devices:

- **Components:** Mobile devices with the Flutter app installed.
- **Purpose:** End-users interact with the mobile application to manage and retrieve their parcels.

8. IoT Locker System:

- **Components:** Arduino Uno R3, Bluetooth Module, and Servo Motor.
- **Purpose:** To control the lockers, communicate with mobile applications, and manage locking/unlocking mechanisms.

9. Notification Service:

- **OneSignal:** For sending notifications to mobile applications.
- **Firebase:** Used as the database for managing notifications.

5.2.3 Summary

The development environment setup for the QLD system ensures a seamless transition from local development to production deployment. Using Flutter for the mobile application and XAMPP for local server setup provides a robust platform for development, while 000webhosting offers reliable hosting for the production environment. The architecture ensures efficient data management and smooth application performance across different stages of development and deployment.

5.3 Software Configuration Management

5.3.1 Configuration Environment Setup

Design and Setup of Configuration Management

In the QLD system, the configuration management is designed to ensure consistency and control throughout the software development lifecycle. The setup is aimed at facilitating seamless collaboration among developers, maintaining version control, and managing the deployment process effectively. Here is an overview of the configuration management setup:

1. Development Environment Configuration:

○ **Android Studio for Flutter Development:**

- **Purpose:** Used for developing and testing the mobile application.
- **Setup:** Configured with Flutter SDK and Dart language support. Includes necessary plugins and extensions to streamline mobile application development.

○ **Visual Studio Code for Web Development:**

- **Purpose:** Used for developing and testing the backend services.
- **Setup:** Configured with PHP, XAMPP for local server and database setup. Includes extensions for PHP development and database management.

○ **Arduino IDE:**

- **Purpose:** Used for developing and uploading code to the Arduino Uno R3.
- **Setup:** Configured with necessary libraries for Bluetooth communication and servo motor control.

2. Local Server and Database Configuration:

○ **XAMPP Server:**

- **Purpose:** Provides a local development environment for web services and database interactions.
- **Setup:** Includes Apache server, MySQL database, and PHP. Configured to mimic the production environment for accurate testing and debugging.
- **Local MySQL Database:**
 - **Purpose:** Used for storing and managing data during the development phase.
 - **Setup:** Database schemas and tables are created and maintained locally to ensure data consistency and integrity.
- **SQLite in Flutter:**
 - **Purpose:** SQLite is used within the mobile application to store data locally, enabling offline access and fast data retrieval.
 - **Setup:** The mobile application uses SQLite for handling various local data needs, ensuring efficiency and security when the app operates offline or with limited connectivity.

3. Production Environment Configuration:

- **000webhosting Server:**
 - **Purpose:** Used for hosting the web application and database in the production environment.
 - **Setup:** Configured with necessary web services and MySQL database to handle live user requests and data management.
- **Production MySQL Database:**
 - **Purpose:** To store and manage data in the live production environment.
 - **Setup:** Configured on the 000webhosting server to handle real-time data storage and retrieval.

4. Notification Service Configuration:

- **OneSignal:**
 - **Purpose:** For sending notifications to mobile applications.
 - **Setup:** Integrated with the mobile application to handle notification delivery.
- **Firebase:**
 - **Purpose:** Used as the database for managing notifications.
 - **Setup:** Configured to store notification data and handle interactions with OneSignal for real-time notifications.

5. Client Devices:

- **Mobile Devices:**
 - **Components:** Mobile devices with the Flutter app installed.
 - **Purpose:** End-users interact with the mobile application to manage and retrieve their parcels.

Software Tools for Configuration Control

1. Flutter and Android Studio:

- **Role:** Primary tools for mobile application development.
- **Configuration Control:** Uses built-in project management and versioning features to track changes and manage the development process.

2. Visual Studio Code:

- **Role:** Primary tool for backend service development.
- **Configuration Control:** Utilizes extensions and plugins for project management, code linting, and debugging. Ensures consistent code quality and adherence to standards.

3. XAMPP:

- **Role:** Provides a local server environment for development.
- **Configuration Control:** Manages configuration files and server settings to replicate the production environment.

4. 000webhosting:

- **Role:** Hosting platform for the production environment.
- **Configuration Control:** Manages deployment settings, domain configurations, and database management for live applications.

5. Arduino IDE:

- **Role:** Used for programming and uploading code to the Arduino Uno R3.
- **Configuration Control:** Ensures the code for the IoT locker system is versioned and consistent.

6. OneSignal and Firebase:

- **Role:** Tools for notification management.
- **Configuration Control:** Integrated with the mobile application and backend services to ensure seamless notification delivery and data management.

Summary

The configuration management setup in the QLD system is designed to ensure a smooth and controlled development process. By using Flutter with Android Studio, Visual Studio Code, XAMPP, 000webhosting, Arduino IDE, OneSignal, and Firebase, the project maintains consistency and control from development through to production, ensuring reliable and efficient software delivery.

5.3.2 Version Control Procedure

Version Control Management

Managing source code versions is crucial for maintaining the integrity and history of the codebase. Here is the procedure and control in place for managing the source code version:

1. Local Version Control:

- **Tool Used:** Git (locally within Android Studio and Visual Studio Code).
- **Procedure:**
 1. **Branching:** Developers create branches for new features, bug fixes, or any significant changes. This isolates their work from the main codebase.
 2. **Committing Changes:** Developers commit their changes to the local Git repository with meaningful commit messages that describe the changes made.
 3. **Merging:** Once a feature or fix is complete, it is merged back into the main branch after thorough testing to ensure stability.

2. Development and Production Environment Synchronization:

- **Procedure:**
 1. **Local Testing:** All changes are first tested locally on the development machine using XAMPP and the local MySQL database.
 2. **Staging Environment:** Before going live, the changes are deployed to a staging environment on 000webhosting. This mimics the production environment and allows for further testing.
 3. **Production Deployment:** After successful testing in the staging environment, the changes are deployed to the

production environment. Deployment scripts ensure that the deployment process is consistent and repeatable.

3. Backup and Recovery:

o Procedure:

1. Regular Backups: Regular backups of the source code and database are maintained to prevent data loss.
2. Recovery Plan: In case of a failure, the backup data is used to restore the system to the last stable state.

By following these procedures and using these tools, the configuration management and version control of the QLD system are maintained effectively, ensuring a stable and reliable development process.

5.4 Implementation Status

The current development status of the QLD system is as follows:

Table 5.1: Mobile Application Modules

Module	Description	Duration	Date Completed	Size of Software
User Authentication and Profile Module	Includes user login, registration, and profile management features using Flutter and Dart.	1 month	April 18, 2024	2 MB (approx.)
User Item List Pending and Arrived Module	Manages the list of pending and arrived items for the user.	1 month	April 18, 2024	3 MB (approx.)

User Item History List Delivered and Picked Module	Displays history of delivered and picked items.	1 month	April 18, 2024	3 MB (approx.)
Setting (Guideline/Change Password/Logout) Module	Provides settings for guidelines, changing password, and logout features.	0.5 months	May 3, 2024	2 MB (approx.)
User Report Module	Allows users to report issues or feedback.	0.5 months	May 3, 2024	3 MB (approx.)
Physical Admin Module	Admin features for managing physical locker interactions.	1 month	May 18, 2024	5 MB (approx.)
Forgot Password Module		0.5 month	August 15, 2024	1 MB (approx.)

Table 5.2: Website Portal Modules

Module	Description	Duration	Date Completed	Size of Software
User Authentication and Profile Module	Includes admin login, registration, and profile management features using PHP.	1 month	June 3, 2024	10 MB (approx.)
Location Module	Manages locations where lockers are deployed.	0.5 months	June 18, 2024	5 MB (approx.)
Locker Module	Manages the locker information and status.	0.5 months	June 18, 2024	5 MB (approx.)

Staff Module	Manages staff details and their access rights.	0.5 months	July 3, 2024	5 MB (approx.)
Customer Module	Manages customer details and their interactions with the system.	0.5 months	July 3, 2024	5 MB (approx.)
Item Management Module	Handles item management including adding, updating, and tracking items.	1 month	July 18, 2024	10 MB (approx.)
Report Module	Provides reporting features for the admin.	0.5 months	July 18, 2024	5 MB (approx.)
Convert Report to PDF Module	Convert Report item management to PDF file with filtering features	0.25 months	August 18, 2024	1 MB (approx.)

Table 5.3: IoT Locker System Module

Module	Description	Duration	Date Completed	Size of Software
IoT Locker System Module	Setup of IoT-based locker system using Arduino Uno R3, including programming the Bluetooth module for communication with the mobile application and controlling the servo motor for locking/unlocking.	1 month	June 3, 2024	10 KB (approx.)

Table 5.4: Notification Service Module

Module	Description	Duration	Date Completed	Size of Software
Notification Service Module	Integration of OneSignal for sending notifications and setting up Firebase for managing notification data.	0.5 months	June 18, 2024	5 MB (approx.)

Table 5.5: Database Module

Module	Description	Duration	Date Completed	Size of Software
Database Module	Setup and management of MySQL database for development and production, ensuring efficient data storage and retrieval.	0.5 months	June 16, 2024	Varies based on data volume

Table 5.6: Deployment and Testing Module

Module	Description	Duration	Date Completed	Size of Software
Deployment and Testing Module	Deployment to production server (000webhosting), extensive testing, and necessary adjustments.	0.5 months	July 16, 2024	Combined size of all deployed components

Table 5.7: Overall Project Timeline

Start Date	End Date
March 18, 2024	4 September, 2024

5.5 Dynamic Locker Code Management Description

The Dynamic Locker Code Management system is designed to generate and manage unique locker codes dynamically, ensuring the secure assignment and retrieval of locker contents. This system is particularly crucial in environments where lockers are shared among multiple users, such as in multi-building apartment complexes, office spaces, or public locker services.

Key Features:

1. Unique Locker Code Generation:

- The system generates unique locker IDs based on predefined rules, ensuring that no two lockers have the same code. This process involves the creation of sequentially incremented IDs, with specific characters reserved for particular locker locations.
- For example, the system ensures that the character 'Q' is exclusively reserved for lockers located in 'L0001', while other locations use different characters.

2. Dynamic Code Assignment:

- Locker codes are assigned dynamically as new lockers are added to the system. The dynamic nature of the code assignment allows for flexible locker management, accommodating the addition of new lockers without disrupting the existing code structure.
- The system avoids using static prefixes, opting instead for a randomized approach where codes are assigned in a way that balances security with usability.

3. Error Handling and Validation:

- The system includes robust error handling to prevent the generation of duplicate or invalid locker codes. Validation checks are in place to ensure that each locker code meets the required format and uniqueness criteria before it is assigned.
- In cases where the system detects potential conflicts, it automatically regenerates the locker code, ensuring seamless operation.

4. Scalability:

- The system is designed to scale the number of lockers and users, making it suitable for both small and large deployments. The use of dynamic code generation and assignment ensures that the system can handle an increasing number of lockers without sacrificing performance or security.

5. Integration with User Interface:

- The locker codes are integrated into the user interface, allowing for easy management and retrieval by the users. The interface provides a clear display of assigned locker codes, along with options for reassigning or updating codes as necessary.

Conclusion:

The Dynamic Locker Code Management system is a critical component of the locker management infrastructure, providing a secure, scalable, and efficient method for generating and managing locker codes. Its dynamic nature ensures flexibility and security, making it an ideal solution for various environments that require secure locker management.

5.6 IoT Locker Setup

The IoT locker system is designed to manage parcel deliveries efficiently. The setup involves foldable boxes acting as lockers, equipped with essential IoT components such as microcontrollers, sensors, and connectivity modules.



Figure 5.10: Foldable Locker Boxes

The locker system uses foldable boxes, as shown below, each labeled with a unique identifier (e.g., Q0001, Q0002, Q0003) for easy identification and management.

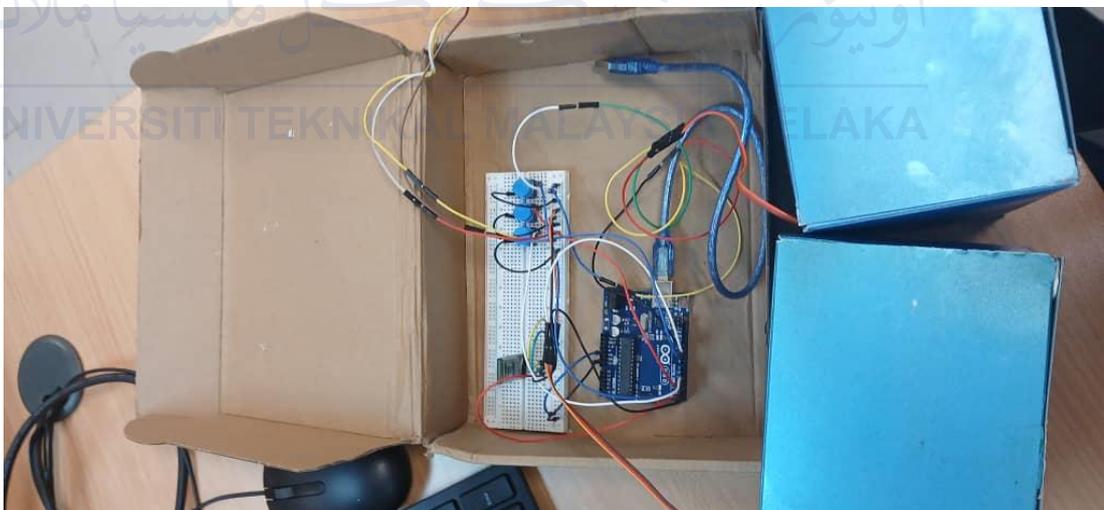


Figure 5.11: IoT Components Inside a Locker

The IoT components, including an Arduino Uno and a breadboard with various connections, are securely placed inside the locker boxes. This setup enables the remote operation of the lockers.



Figure 5.12: Example Parcel for Demonstration

An example parcel used in the demonstration of the IoT locker system is shown below. This parcel simulates the actual use case of the system in real-world scenarios.

In conclusion, this setup is crucial for ensuring that the lockers function correctly and that the IoT components are well-integrated, providing a seamless user experience.

5.7 Conclusion

This chapter has outlined the setup and progress of the QLD system implementation. The next steps involve completing the remaining development tasks, integrating all system components, and conducting comprehensive testing to ensure functionality and reliability.

CHAPTER 6: TESTING

6.1 Introduction

The testing phase is a crucial part of the software development lifecycle where the software application is evaluated to ensure it meets the specified requirements and functions correctly. The primary activities involved in this phase include the creation of test plans, designing test cases, executing tests, and analyzing test results. The testing strategy adopted for the QLD project involves a bottom-up approach combined with black-box testing methods to thoroughly assess the software. This chapter provides a detailed overview of the testing processes, including the test plan, test environment, test schedule, test strategy, test design, and test results.

6.2 Test Plan

The approach, scope, objectives, resources, and timetable for testing the QLD Project are described in detail in this section. The test plan guides the testing procedure throughout the Software Development Life Cycle (SDLC), acting as a roadmap to ensure systematic and organized testing efforts.

6.2.1 Test Organization

This section outlines the roles and responsibilities of various personnel involved in the testing process.

Table 6.1: Test Organization

Tester_ID	Name	Role	Responsibilities
T01	Nabil Aqmar Bin Zuhaimi	Software Developer	Developing the front-end and back-end of the application, execution and integration, and conducting tests.
T02	Ahmad Khairul Bin Nizam	Software Tester	Responsible for testing the functionality and non-functionality of the application modules.
T03	Nur Alya Binti Syamsuddin	End User	Responsible for testing the application from the end-user perspective.

6.2.2 Test Environment

The test environment is a controlled and configured setting that simulates real-world usage scenarios to ensure that the QLD Project application functions correctly and performs as expected before being released to end-users. The following sections detail the environment setup, application software, system software, and hardware configurations used in the testing process.

6.2.2.1 Environment Setup

The environment setup serves as a structured framework for managing the testing of the QLD Project application. This setup is crucial for verifying the correct operation of each module, ensuring that the system meets all functional requirements.

6.2.2.2 Application Software

Application software refers to the specific programs designed to perform user-centric tasks within the QLD Project application. Table 6.2 lists the various application modules used in the testing environment.

Table 6.2: Application Module

Application Module	Description	Platform
User Authentication	Verifies user login credentials.	Mobile/Web
User Registration	Handles new user sign-ups and validation.	Mobile/Web
User Information	Manages user profile details and updates.	Mobile/Web
Item Delivery	Manages the delivery process of items.	Mobile/Web
Item Delivery History	Displays history of delivered items.	Mobile
Google Maps API	Integrates location services within the app.	Mobile
QR Code Generator	Generates QR codes for item identification.	Mobile
Notification	Manages push notifications to users.	Mobile

Locker Functionality	Controls IoT locker system for secure item storage.	Mobile
Report	Generates reports on various application functions.	Mobile/Web
Locker Location	Manages and updates locker location information.	Web
Item Management Report Admin	Allows admins to view and manage item-related reports.	Web
Users Report Information	Manages reports related to user activities and details.	Web
Forgot Password	Allow users to request to reset password via email	Mobile
Convert report to PDF	Allow admin to convert item management report to PDF	Web
SQLite functionality	Allow user to use QR Code to open the locker without Wi-Fi connection	Mobile
Changeable Location in Locker Functionality	Allow Admin to change the location in locker code.	Mobile/Arduino

6.2.2.3 System Software

System software provides the underlying support needed for developing and running the QLD Project application. Table 6.3 lists the software tools utilized.

Table 6.3: System Software and purpose

System Software	Purpose
Android Studio (Flutter)	Integrated development environment (IDE) for Android.
XAMPP	Local server environment for testing PHP applications.
Visual Studio	IDE used for web development tasks.
MySQL	Database management system used for storing data online.
SQLite	Database management system used for storing local data in mobile.

6.2.2.4 System Hardware

The hardware setup includes the devices and components required for the development and testing of the QLD Project application. Table 6.4 lists the hardware configurations used.

Table 6.4: System Hardware Description

System Hardware	Description

MSI Thin GF63 10UC Laptop	Development and testing laptop.
Pixel_3a_API_34_extension_level_7_x86_64	Test emulator for android application.
Samsung Galaxy A13	Test mobile device for Android application.
Samsung Galaxy Tab A8	Microcontroller for IoT locker system integration.
Arduino Uno R3	Facilitates communication between the mobile app and Arduino.
Servo Motor	Controls unlocking mechanisms in the IoT system.
Buttons	Controls locking mechanisms in the IoT system.
USB Cables	Used for connecting and uploading code to Arduino.

6.2.3 Test Schedule

In the testing phase of the QLD Project, the test schedule is a comprehensive plan detailing the timing and order of testing tasks. It specifies who will oversee them, when specific testing tasks will be carried out, and how long each task is anticipated to take. This schedule is crucial for ensuring that testing activities are effectively planned, executed, and completed within the allocated timeframe. Below is the test schedule for the QLD Project:

Table 6.5: Test Schedule

No.	Testing Module	Start Date	End Date	Duration
1.	User Authentication Admin (Web)	01/8/2024	01/8/2024	1 Day
2.	User Registration Admin (Web)	01/08/2024	01/08/2024	1 Day
3.	User Profile (Web)	02/08/2024	02/08/2024	1 Day
4.	Item Delivery (Web)	02/08/2024	02/08/2024	1 Day
5.	Item Management Report (Web)	02/08/2024	02/08/2024	1 Day
6.	Locker Location (Web)	03/08/2024	03/08/2024	1 Day
7.	Locker Information (Web)	03/08/2024	03/08/2024	1 Day
8.	Users Report Information (Web)	04/08/2024	04/08/2024	1 Day
9.	Convert Report to PDF (Web)	04/08/2024	04/08/2024	1 Day
10.	User Login (Mobile)	04/08/2024	04/08/2024	1 Day
11.	User Registration (Mobile)	04/08/2024	04/08/2024	1 Day
12.	User Profile (Mobile)	05/08/2024	05/08/2024	1 Day

13.	Item Delivery (Mobile)	06/08/2024	06/08/2024	1 Day
14.	Item Delivery History (Mobile)	06/08/2024	06/08/2024	1 Day
15.	Google Map API (Mobile)	06/8/2024	06/8/2024	1 Day
16.	QR Code Generator (Mobile)	06/8/2024	06/8/2024	1 Day
17.	Notification (Mobile)	06/8/2024	06/8/2024	1 Day
18.	Locker Functionality (Mobile with Arduino)	07/8/2024	07/8/2024	1 Day
19.	Item Delivery Report (Mobile)	08/8/2024	08/8/2024	1 Day
20.	Forgot Password (Mobile)	08/8/2024	08/8/2024	1 Day
21.	SQLite functionality	09/8/2024	09/8/2024	1 Day
22.	Changeable Location in Locker Functionality	09/8/2024	09/8/2024	1 Day

6.3 Test Strategy

The Test Strategy for the QLD Project is a comprehensive document in software testing that precisely outlines the specific approach and testing goals for the application. This test strategy addresses various questions, including what is intended to be accomplished and how it will be achieved. The testing phase will involve two

testing: dynamic testing and user acceptance testing (UAT), conducted through questionnaires with the end users.

6.3.1 Dynamic Testing

Dynamic testing is a software testing technique that involves evaluating the behavior of the QLD Project application while it is running. In this case, the focus is on assessing the application's functionality and performance in a real-time, operational environment. For the QLD Project, only **Bottom-Up Testing** and **Black Box Testing** are utilized.

- **Bottom-Up Testing:** This approach involves testing the system from the lowest levels up to the higher levels of the application. Individual modules or components are tested first, and once they pass, they are integrated to form larger subsystems, which are then tested collectively. This approach ensures that all components function correctly before moving up to test more complex functionalities.
- **Black Box Testing:** This is a technique that does not require knowledge of the internal structure or code of the system being tested. The tester provides inputs to the system and observes the outputs, focusing on the functionality of the system based on its specifications. Black Box Testing helps identify issues like incorrect or missing functionalities, usability problems, and unexpected behavior without needing insight into the underlying code.

This strategy ensures the QLD Project meets its functional requirements and provides a seamless user experience.

6.3.2 System Usability Scale (SUS)

The System Usability Scale (SUS) is a widely recognized tool used to assess the usability of software applications. In the context of the QLD Project, SUS will serve as the primary method for evaluating the application's user-friendliness and

overall user experience. SUS provides a quantitative measurement of how intuitive and accessible the software is for users based on their actual interactions with it.

During the SUS evaluation, users—including admin users, staff couriers, and end customers—will engage with the QLD system and complete a structured questionnaire designed to capture their perceptions of usability. The questionnaire contains a set of statements that users will rate based on their experience with the software, covering aspects such as ease of use, system complexity, and confidence in navigating the system. This feedback is crucial in determining whether the system meets usability standards before its official release.

The goal of SUS is to ensure that the software not only functions as intended but also delivers a positive and seamless user experience. This method provides valuable insights into the usability of the software and highlights areas that may require improvement. Based on SUS feedback, the development team can make final adjustments to enhance the user interface, streamline workflows, or simplify interactions.

The results of the SUS evaluation will be aggregated into a score that reflects the overall usability of the system. A high SUS score indicates that the QLD system is easy to use and meets user expectations, while a lower score may prompt further refinements. The SUS method ensures that the software is not only technically sound but also user-friendly, improving the likelihood of a successful deployment and adoption by its intended audience.

6.3.3 Analysis of Issues in University Courier Service Delivery and Evaluation of the QuickLocker-Delivery Project.

The objective of this report is to analyze the current state of the manual courier delivery system at the university and evaluate the proposed QuickLocker-Delivery (QLD) system. To achieve this, a survey was conducted among university

stakeholders, gathering data on their experiences with the current delivery system and their opinions on the proposed QLD project.

The survey consisted of several key questions designed to assess user satisfaction with the existing manual delivery process, identify the primary challenges faced, and evaluate the perceived benefits and potential improvements offered by the QLD system. The detailed questionnaire and the comprehensive results from the survey are provided in **Appendix B**. This appendix includes the full list of survey questions, the distribution of responses, and graphical representations of the data collected.

By analyzing this data, the report aims to provide a clear understanding of the current issues within the university's courier service and determine the most effective features of the proposed QuickLocker-Delivery system for improving service efficiency and user satisfaction.

6.4 Test Design

Test design is the process of developing test cases that will be used to validate the functionality of the QLD Project's software system. It is an essential step in the software testing process, as it ensures that the tests are thorough and effective in identifying defects.

6.4.1 Test Description

The test description section is used to verify that the system functions as intended. Each test description includes a unique identifier, a description, and the expected outcome. Testing will be conducted by Nabil Aqmar, Ahmad Khairul, and Nur Alya.

6.4.1.1 Web Testing Module

6.4.1.1.1 User Login (Web)

Table 6.6 provides an overview of the test cases for the User Login module on the web platform. These test cases verify the login functionality, including correct and incorrect input handling, as well as role-based access for the Admin.

Table 6.6: User Login (Web)

Test Case ID	Description	Expected Result
TW01_01	To check the functionality of login when the user enters the correct username and password.	Successful login with valid credentials; appropriate error message with invalid credentials.
TW01_02	To check the functionality of login when the user enters an incorrect username and password.	An error message "Wrong Username or Password" will be displayed.
TW01_03	To check the login functionality when the Username field is empty.	An error message "Username cannot be empty" will be displayed.
TW01_04	To check the login functionality when the Password field is empty.	An error message "Password cannot be empty" will be displayed.
TW01_05	To check the login role for Admin.	Successful navigation to the next page in the application according to user role.

6.4.1.1.2 User Registration (Web)

Table 6.7 provides an overview of the test cases for the User Registration module on the web platform.

Table 6.7: User Registration (Web)

Test Case ID	Description	Expected Result
TW02_01	To check the functionality of user registration on the "Add New Employee" page with valid data.	Users can create a new account, and a message "Signup Success!" will be displayed.
TW02_02	To check the registration functionality when any required field is empty.	An error message "Input Field cannot be empty" will be displayed.
TW02_03	To check the registration functionality when the input format is incorrect (e.g., wrong email format).	An error message "Input Format (input error field name placed here) is wrong, please use (guided format here)" will appear.
TW02_04	To check the registration functionality when the entered Username already exists.	An error message "Username Exist!" will be displayed.
TW02_05	To check the validity of the selected role in the web interface.	The selected role in the web is paired correctly with the registered user by the admin.

TW02_06	To check the registration functionality when the password and confirm password fields do not match.	An error message "Password does not match" will be displayed.
TW02_07	To check the registration functionality when an unsupported image format is uploaded.	An error message "Invalid image format. Please upload a valid image file" will be displayed.
TW02_08	To check the registration functionality when the entered email already exists in the system.	An error message "Email already exists!" will be displayed.
TW02_09	To check the registration functionality when the phone number format is invalid (e.g., contains alphabets).	An error message "Invalid phone number format" will be displayed below the input field.
TW02_10	To check the registration functionality when the entered password is shorter than the required minimum length.	An error message "Password is too short, please enter at least 8 characters" will be displayed below the input field.
TW02_11	To check the registration functionality when no image is uploaded.	An error message "Image cannot be empty" will be displayed.

TW02_12	To check the successful registration with valid data and correct role assignment.	Users can create a new account, role assignment is successful, and a message "Signup Success!" will be displayed.
---------	---	---

6.4.1.1.3 User Information (Web)

Table 6.8 provides an overview of the test cases for the User Information module on the web platform.

Table 6.8: User Information (Web)

Test Case ID	Description	Expected Result
TW03_01	To check if the user can update their profile information such as name, contact number, address, and password.	Profile information is successfully updated, and a confirmation message is displayed.
TW03_02	To check if the user can view their profile information correctly.	Profile information is correctly displayed in the profile section.
TW03_03	To check if the system prevents updating the profile when required fields are left empty.	The system should display an error message indicating that required fields cannot be left empty.

TW03_04	To check if the system prevents updating the password when the new password and confirmation do not match.	The system should display an error message indicating that the new password and confirmation do not match.
TW03_05	To check if the system validates the email format during profile update.	The system should display an error message indicating an invalid email format.
TW03_06	To check if the user can update their profile without uploading an image.	Profile information is successfully updated without requiring an image, and a confirmation message is displayed.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

6.4.1.1.4 Item Delivery (Web)

Table 6.9 provides an overview of the test cases for the Item Delivery module on the web platform.

Table 6.9: Item Delivery (Web)

Test Case ID	Description	Expected Result
TW04_01	To check if the admin can register an item.	Item is successfully registered.

TW04_02	To check if the admin can assign an item to a staff courier for delivery.	Item is successfully assigned to the staff courier.
TW04_03	To check if the admin can filter items by Item Management ID.	Items are correctly filtered and displayed based on the Item Management ID.
TW04_04	To check if the admin can filter items by Item ID.	Items are correctly filtered and displayed based on the Item ID.
TW04_05	To check if the admin can filter items by Locker Location.	Items are correctly filtered and displayed based on the Locker Location.
TW04_06	To check if the admin can filter items by Size.	Items are correctly filtered and displayed based on the Size.
TW04_07	To check if the admin can reset the filter criteria.	All filters are cleared, and the full list of items is displayed.

6.4.1.1.5 Item Management Report Admin (Web)

Table 6.10 provides an overview of the test cases for the Item Management Report Admin module on the web platform.

Table 6.10: Item Management Report Admin (Web)

Test Case ID	Description	Expected Result
TW05_01	To check if the admin can view the report on the dashboard.	The correct report is displayed on the dashboard based on the admin's data and access rights.
TW05_02	To check if the admin can view, search, update and delete the report on the item management page.	Admin can view, search, update and delete the report on the item management page successfully.

6.4.1.1.6 Locker Location (Web)

Table 6.11 provides an overview of the test cases for the Locker Location module on the web platform.

Table 6.11: Locker Location (Web)

Test Case ID	Description	Expected Result
TW06_01	To check if a new locker location registration can be registered.	A new locker location registration can be registered successfully.

TW06_02	To check if locker location information can be searched, viewed, updated and deleted.	The correct locker location information is displayed and can be updated and deleted.
---------	---	--

6.4.1.1.7 Locker Information (Web)

Table 6.12 provides an overview of the test cases for the Locker Information module on the web platform.

Table 6.12: Locker Information (Web)

Test Case ID	Description	Expected Result
TW07_01	To check if a new locker registration can be registered.	A new locker registration can be registered successfully.
TW07_02	To check if locker information can be searched, viewed, updated, and deleted.	The correct locker information is displayed and can be searched, updated, and deleted.
TW07_03	To check if locker status can be changed.	The locker status can be successfully updated, and the changes are reflected in the locker list.
TW07_04	To check if the locker availability can be changed.	The locker availability is successfully updated (e.g., Available/Unavailable),

		and the changes are reflected.
TW7_05	To check if invalid input (e.g., leaving required fields empty) is handled.	The system displays error messages for required fields that are left empty, preventing invalid input submission.
TW07_06	To check if the search functionality returns correct locker details.	Locker search returns accurate results based on entered criteria (Locker ID, Name, Size, Location, Status).
TW07_07	To check if the locker list is correctly paginated and shows correct entries.	The locker list is paginated correctly, displaying the correct number of entries per page as per user selection.
TW07_08	Verify that the locker ID is correctly paired with the location ID.	The locker ID is successfully paired with the correct location ID.

6.4.1.1.8 Users Report Information (Web)

Table 6.13 provides an overview of the test cases for the Users Report Information module on the web platform.

Table 6.13: Users Report Information (Web)

Test Case ID	Description	Expected Result
TW08_01	To check if user Staff Courier information can be viewed, searched, updated, and deleted.	The correct user information is displayed and can be searched, updated, and deleted.
TW08_02	To check if user Customer (Receiver) information can be viewed, searched, updated, and deleted.	The correct user information is displayed and can be searched, updated, and deleted.

6.4.1.1.9 Convert Report to PDF (Web)

Table 6.14 provides an overview of the test cases for the Convert Report to PDF module on the web platform.

Table 6.14: Convert Report to PDF (Web)

Test Case ID	Description	Expected Result
TW09_01	Verify that the user can successfully convert the item management report into a PDF file using the	The item management report is converted to a PDF and downloaded without errors. The PDF file should contain all the information

	"Generate Report PDF" button.	displayed in the "Item Management List" table.
TW09_02	Verify that the generated PDF report accurately reflects the filtered data when a search query or filter is applied to the "Item Management List".	The generated PDF should contain only the filtered data as per the applied search filters and match the data displayed in the web interface.
TW09_03	Verify that the generated PDF is properly formatted and readable on various PDF readers.	The PDF report should be readable, with proper alignment, formatting, and no data truncation across different PDF readers (e.g., Adobe Acrobat, browser PDF viewers, etc.).
TW09_04	Verify that the generated PDF report is stored with a meaningful filename that includes the date and time of generation.	The downloaded PDF should have a filename (Item_Management_Report_YYYY/MM/DD.pdf), corresponding to the date and time the report was generated.
TW09_05	Verify that large datasets in the "Item Management List" table are correctly handled and exported to the PDF.	The generated PDF should include all records, even when there are many entries, without any performance issues or data omissions.

6.4.1.2 Mobile Testing Module

6.4.1.2.1 User Login (Mobile)

Table 6.15 provides an overview of the test cases for the User Login module on the mobile platform.

Table 6.15: User Login (Mobile)

Test Case ID	Description	Expected Result
TM01_01	To check the functionality of login when the user enters the correct username and password.	Successful login with valid credentials; appropriate error message with invalid credentials.
TM01_02	To check the functionality of login when the user enters an incorrect username and password.	An error message "Wrong Username or Password" will be displayed.
TM01_03	To check the login functionality when the Username field is empty.	An error message "Username cannot be empty" will be displayed.
TM01_04	To check the login functionality when the Password field is empty.	An error message "Password cannot be empty" will be displayed.
TM01_05	To check the login role for Admin, Staff Courier, Customer Receiver.	Successful navigation to the next page in the

		application according to user role.
--	--	-------------------------------------

6.4.1.2.2 User Registration (Mobile)

Table 6.16 provides an overview of the test cases for the User Registration module on the mobile platform.

Table 6.16: User Registration (Mobile)

Test Case ID	Description	Expected Result
TM02_01	To check the functionality of the user registration on the signup page.	Users can create a new account, a message "Signup Success!" will be displayed.
TM02_02	To check the registration functionality when any field is empty.	An error message "Input Field cannot be empty" will be displayed.
TM02_03	To check the registration functionality when the input format is wrong.	An error message "Input Format (input error field name placed here) is wrong please use (guided format here)" will be displayed below the input field.

TM02_04	To check functionality when the Username entered is not available.	An error message "Username Exist!" will be displayed.
TM02_05	To check functionality when the user did not tick the terms and conditions checkbox.	An error message "Accept the terms and conditions checkbox" will be displayed below checkbox.
TM02_06	To check the functionality of sending a welcome email after successful registration.	A welcome email is sent to the user's registered email address immediately after successful registration.

6.4.1.2.3 User Profile (Mobile)

Table 6.17 provides an overview of the test cases for the User Profile module on the mobile platform.

Table 6.17: User Profile (Mobile)

Test Case ID	Description	Expected Result
TM03_01	To check if the user can update their profile information such as name, contact number, and email address.	Profile information is successfully updated, and a confirmation message is displayed.

TM03_02	To check if the user can view their profile information correctly.	Profile information is correctly displayed in the profile section.
TM03_03	To check if the user receives an error when attempting to update the QLD ID.	The QLD ID field should be non-editable, and no changes should be allowed.
TM03_04	To check if the user can update their profile picture.	Profile picture is successfully updated, and a confirmation message is displayed.

6.4.1.2.4 Item Delivery (Mobile)

Table 6.18 provides an overview of the test cases for the Item Delivery module on the mobile platform.

Table 6.18: Item Delivery (Mobile)

Test Case ID	Description	Expected Result
TM04_01	To check if the staff courier can receive the assigned item from the admin using QLD ID.	Staff courier successfully receives the assigned item and sees it in the pending list.
TM04_02	To check if the receiver can view the delivered item list after the staff courier completes the delivery.	Delivered item list is correctly displayed for the receiver.

TM04_03	To check if the staff courier can search for a specific item in the pending list.	The search function works correctly, displaying the specific item in the pending list.
TM04_04	To check if the staff courier can attempt to receive an item that is not assigned.	The item does not appear in the pending list, and the staff courier cannot receive it.

6.4.1.2.5 Item Delivery History (Mobile)

Table 6.19 provides an overview of the test cases for the Item Delivery History module on the mobile platform.

Table 6.19: Item Delivery History (Mobile)

Test Case ID	Description	Expected Result
TM05_01	To check if the staff courier and receiver can view and the item delivery history.	Item delivery history is correctly displayed based on user data.

6.4.1.2.6 Google Map API (Mobile)

Table 6.20 provides an overview of the test cases for the Item Google Map API module on the mobile platform.

Table 6.20: Google Map API (Mobile)

Test Case ID	Description	Expected Result
TM06_01	To check if the direction map displays correctly based on the locker location chosen by the user.	The correct direction map is displayed based on the chosen locker location.

6.4.1.2.7 QR Code Generator (Mobile)

Table 6.21 provides an overview of the test cases for the Item QR Code Generator module on the mobile platform.

Table 6.21: QR Code Generator (Mobile)

Test Case ID	Description	Expected Result
TM07_01	To check if QR codes are generated based on item data such as itemMngtId, itemSize, roleId, and lockerLocationId.	QR codes are generated correctly, and the associated item data is embedded accurately in the QR code.
TM07_02	To check if the generated QR codes can be scanned and the item data such as itemMngtId, itemSize, roleId, and lockerLocationId are retrieved correctly.	QR codes are scanned correctly, and the data is accurately retrieved and matches the item data

		associated with the QR code.
TM07_03	To check if the QR code data is encrypted before generating the QR code based on item data.	QR code data is encrypted correctly, ensuring the item data is securely encoded within the QR code.
TM07_04	To check if encrypted QR code data can be decrypted successfully and display the original item data such as itemMngtId, itemSize, roleId, and lockerLocationId.	QR code data is decrypted successfully, and the original item data is revealed and displayed correctly.

6.4.1.2.8 Notification (Mobile)

Table 6.22 provides an overview of the test cases for the Notification module on the mobile platform.

Table 6.22: Notification (Mobile)

Test Case ID	Description	Expected Result
TM08_01	To check if a notification will be sent to the staff courier after the admin assigns an item to the staff courier for delivery.	The staff courier receives a notification indicating that a new item has been assigned for delivery.
TM08_02	To check if a notification will be sent to the receiver after the staff courier delivers the item.	The receiver receives a notification indicating that their item has been

		delivered and is ready for pickup.
TM08_03	To check if a reminder notification is sent to the receiver after they open the locker and pick up the item.	The receiver receives a notification reminding them to close the locker after picking up the item.

6.4.1.2.9 Locker Functionality (Mobile with Arduino)

Table 6.23 provides an overview of the test cases for the Locker Functionality module on the mobile platform.

Table 6.23: Locker Functionality (Mobile with Arduino)

Test Case ID	Description	Expected Result
TM09_01	To check if the locker opens correctly using a valid QR code.	The locker opens correctly, locker status is updated, and a notification is sent to the recipient.
TM09_02	To check system behavior when an invalid QR code is scanned.	An "Invalid" error is displayed, and the user is prompted to scan a new QR code.
TM09_03	To check if the system handles decryption failure during QR code scanning.	A "Decrypt failed" error is displayed, and the user

		is prompted to scan a new QR code.
TM09_04	To validate QR code format and handle errors when the format is incorrect.	An "Invalid" error is displayed.
TM09_05	To check system behavior when no lockers are available for selection.	An "Invalid" error is displayed when no lockers are available.
TM09_06	To verify successful connection to the locker device after selecting a locker.	The locker device connects successfully.
TM09_07	To check system response when the locker device fails to connect.	An "Invalid" error is displayed when the locker device connection fails.
TM09_08	To check if the system handles incorrect button press for opening the locker.	An "Invalid" error is displayed when the wrong button is pressed.
TM09_09	To verify locker opens correctly when the correct button is pressed.	The locker opens correctly after the correct button is pressed.
TM09_10	To verify if the system updates locker and item status after the locker is opened.	Locker and item status are updated successfully after the locker is opened.

TM09_11	To check if the system sends a notification to the recipient after the locker is opened.	A notification is sent successfully to the recipient after the locker is opened.
TM09_12	To check if the locker closes correctly when the correct locker is selected.	The locker closes correctly, and the locker status is updated in the system.
TM09_13	To verify system behavior when the wrong locker is selected for closing.	An "Invalid" error is displayed, and the user is prompted to choose the correct locker.
TM09_14	To verify if the system updates locker status correctly after the locker is closed.	Locker status is updated successfully in the system after the locker is closed.
TM09_15	To check if Wi-Fi connection is working correctly with Arduino setup.	The system verifies that the Wi-Fi is connected successfully, and a success message is displayed. In case of failure, an error message is shown indicating Wi-Fi connection issues.

6.4.1.2.10 Item Delivery Report (Mobile)

Table 6.24 provides an overview of the test cases for the Item Delivery Report module on the mobile platform.

Table 6.24: Item Delivery Report (Mobile)

Test Case ID	Description	Expected Result
TM10_01	To check if the staff courier can view the report on the dashboard.	The correct report is displayed on the dashboard based on the staff courier's data and responsibilities.
TM10_02	To check if the receiver can view the report on the dashboard.	The correct report is displayed on the dashboard based on the receiver's data and interactions.

6.4.1.2.11 Forgot Password (Mobile)

Table 6.25 provides an overview of the test cases for the Item Delivery Report module on the mobile platform.

Table 6.25: Forgot Password (Mobile)

Test Case ID	Description	Expected Result
TM11_01	Verify that the user can request a password reset by providing a valid	The system verifies the username and email address. If valid, a new

	username and email address.	password is generated and sent to the email address. The user sees a confirmation message indicating that the password has been sent.
TM11_02	Verify that the system displays an error when the user provides an invalid email or username.	The system rejects the input and displays an error message indicating invalid username or email. No password reset email is sent.
TM11_03	Verify that the user can reset their password by clicking on a link in the email, which uses a token for verification.	The system sends a password reset link with a unique token to the user's email. When the user clicks the link, the system verifies the token and allows the user to reset their password.
TM11_04	Verify that a new random password is generated and sent to the user's email after resetting the password via the token.	After the user resets their password via the link, the system generates a new random password and sends it to the user's email. A confirmation message is displayed.
TM11_05	Verify that the token in the password reset link	The system invalidates the token if it is used after its expiration time and

	expires after a certain period.	displays an error message indicating that the link has expired. The user must request a new password reset link.
--	---------------------------------	--

6.4.1.2.12 SQLite Functionality (Mobile)

Table 6.26 provides an overview of the test cases for the SQLite Functionality module on the mobile platform.

Table 6.26: SQLite Functionality (Mobile)

Test Case ID	Description	Expected Result
TM12_01	Verify that when the user (Staff Courier/Receiver) has no internet connection, it uses SQLite data.	The system retrieves and displays data from SQLite, ensuring offline functionality.
TM12_02	Verify that after the user scans the QR code, they cannot scan it again while offline.	The system prevents duplicate scanning of the same QR code in offline mode.
TM12_03	Verify that the data is updated in SQLite when changes occur in the MySQL database.	SQLite data is synced and updated correctly from MySQL when the connection is restored.
TM12_04	Verify that SQLite stores multiple locker items	SQLite successfully manages multiple locker

	correctly and updates item status accordingly.	items and updates the item statuses accurately.
--	--	---

6.4.1.2.13 Changeable Location in Locker Functionality (Mobile with Arduino)

Table 6.27 provides an overview of the test cases for the Changeable Location in Locker Functionality module on the mobile platform.

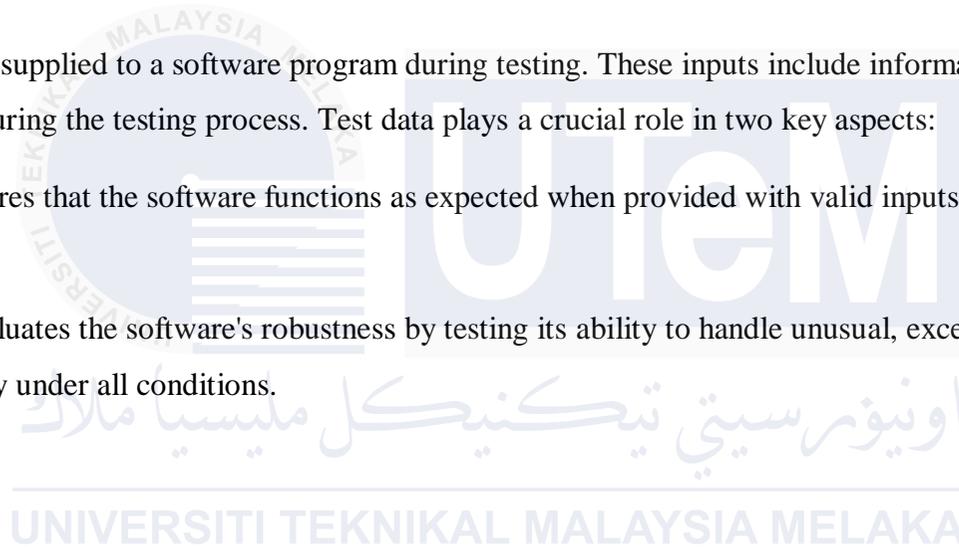
Table 6.27: Changeable Location in Locker Functionality (Mobile with Arduino)

Test Case ID	Description	Expected Result
TM13_01	To verify that when the first character of the Locker ID is changed in the Arduino code, it updates the location of the locker in the system. A physical admin will scan the QR code via the mobile app to validate the new location.	The system correctly updates the location of the locker when the first character of the Locker ID changes, and the mobile app reflects the updated location when the physical admin scans the QR code.

6.4.2 Test Data

Test data comprises the inputs supplied to a software program during testing. These inputs include information that either impacts the software's operation or is affected by it during the testing process. Test data plays a crucial role in two key aspects:

1. **Positive Testing:** Ensures that the software functions as expected when provided with valid inputs, confirming that the desired outcomes are achieved.
2. **Negative Testing:** Evaluates the software's robustness by testing its ability to handle unusual, exceptional, or unexpected inputs, ensuring it behaves appropriately under all conditions.



6.4.2.1 Web Testing Data

6.4.2.1.1 Test Data for User Login Admin (Web)

System: QuickLocker-Delivery
 Module/Unit: User Login Admin (Web)
 Processed By: Nur Alya Binti Syamsuddin

Version: v1
 Revision: -
 Date: 01/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW01_01	Login with valid credentials	<ol style="list-style-type: none"> Navigate to login page Enter valid username and password Click login button 	Username: admin123 Password: password	Successful login with valid credentials; user is redirected to the admin dashboard.
TW01_02	Login with incorrect	<ol style="list-style-type: none"> Navigate to login page Enter invalid username and password 	Username: wronguser Password: wrongpass	An error message "Wrong Username or Password" will be displayed.

	username and password	3. Click login button		
TW01_03	Login with empty Username field	<ol style="list-style-type: none"> 1. Navigate to login page 2. Leave the Username field empty 3. Enter valid password 4. Click login button 	<p>Username: (empty)</p> <p>Password: password</p>	An error message "Username cannot be empty" will be displayed.
TW01_04	Login with empty Password field	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter valid username 3. Leave the Password field empty 4. Click login button 	<p>Username: admin123</p> <p>Password: (empty)</p>	An error message "Password cannot be empty" will be displayed.
TW01_05	Check role-based navigation for Admin	<ol style="list-style-type: none"> 1. Navigate to login page 2. Enter valid admin credentials 	<p>Username: admin123</p> <p>Password: password</p>	Successful login as Admin; user is navigated to the admin-specific section of the application.

		3. Click login button		
--	--	-----------------------	--	--

6.4.2.1.2 Test Data for User Registration Admin (Web)

System: QuickLocker-Delivery

Module/Unit: User Registration Admin (Web)

Processed By: Nur Alya Binti Syamsuddin

Version: v1

Revision: -

Date: 01/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW02_01	Successful User Registration	<ol style="list-style-type: none"> 1. Login as admin. 2. Navigate to "Add New Employee" page 3. Enter valid data for all fields 4. Upload an image 5. Select a role 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p>	Users can create a new account, a message "Signup Success!" will be displayed.

		6. Click "Add New" button	Full Name: John Doe Role: Staff Image: validImage.png	
TW02_02	Registration with Empty Fields	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Leave one or more required fields empty 3. Click "Add New" button 	Username: (empty) Password: password123 Confirm Password: password123 Phone Number: 0123456789 IC Number: 123456789012 E-mail: john.doe@example.com Full Name: John Doe Role: Staff Image: validImage.png	An error message "Input Field cannot be empty" will be displayed.

TW02_03	Registration with Wrong Input Format	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter invalid data in any field (e.g., incorrect email format) 3. Click "Add New" button 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: johndoe@example (invalid email format)</p> <p>Full Name: John Doe</p> <p>Role: Staff</p> <p>Image: validImage.png</p>	An error message "Input Format (input error field name placed here) is wrong, please use (guided format here)" will be displayed below the input field.
TW02_04	Registration with Username Already Taken	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter a username that already exists in the system 3. Click "Add New" button 	<p>Username: existinguser</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p>	An error message "Username Exist!" will be displayed.

			<p>IC Number: 123456789012</p> <p>E-mail: existinguser@example.com</p> <p>Full Name: Existing User</p> <p>Role: Staff</p> <p>Image: validImage.png</p>	
TW02_05	<p>Role Selection Validity</p>	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Fill in valid data for all fields 3. Select the desired role from the dropdown (e.g., Admin, Staff) 4. Click "Add New" button 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Admin</p> <p>Image: validImage.png</p>	<p>The selected role in the web interface is correctly paired with the registered user, and the user is assigned the appropriate permissions.</p>

TW02_06	Registration with Password Mismatch	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter valid data for all fields except mismatched passwords 3. Click "Add New" button 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password456 (mismatch)</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Staff</p> <p>Image: validImage.png</p>	An error message "Password does not match" will be displayed.
TW02_07	Registration with Invalid Image File Format	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter valid data for all fields 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p>	An error message "Invalid image format. Please upload a valid image file" will be displayed.

		<p>3. Upload an image in an unsupported format (e.g., .txt)</p> <p>4. Click "Add New" button</p>	<p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Staff</p> <p>Image: invalidImage.txt</p>	
TW02_08	Registration with Duplicate Email	<p>1. Navigate to "Add New Employee" page</p> <p>2. Enter valid data for all fields but use an email that already exists in the system</p> <p>3. Click "Add New" button</p>	<p>Username: john.doe2</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com (duplicate)</p> <p>Full Name: John Doe</p> <p>Role: Staff</p>	An error message "Email already exists!" will be displayed.

			Image: validImage.png	
TW02_09	Registration with Invalid Phone Number	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter valid data for all fields except an invalid phone number format (e.g., alphabets) 3. Click "Add New" button 	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: abc123 (invalid format)</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Staff</p> <p>Image: validImage.png</p>	An error message "Invalid phone number format" will be displayed below the input field.
TW02_10	Registration with Short Password	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter valid data for all fields except a password that 	<p>Username: john.doe</p> <p>Password: pass (too short)</p> <p>Confirm Password: pass</p>	An error message "Password is too short, please enter at least 8 characters" will be displayed below the input field.

		<p>is shorter than the required length (e.g., less than 8 characters)</p> <p>3. Click "Add New" button</p>	<p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Staff</p> <p>Image: validImage.png</p>	
TW02_11	<p>Registration without Image Upload</p>	<p>1. Navigate to "Add New Employee" page</p> <p>2. Enter valid data for all fields but do not upload any image</p> <p>3. Click "Add New" button</p>	<p>Username: john.doe</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.doe@example.com</p> <p>Full Name: John Doe</p> <p>Role: Staff</p>	<p>An error message "Image cannot be empty" will be displayed.</p>

			Image: (none uploaded)	
TW02_12	Successful Registration with Role Assignment	<ol style="list-style-type: none"> 1. Navigate to "Add New Employee" page 2. Enter valid data for all fields 3. Select a valid role (e.g., Admin or Staff) 4. Click "Add New" button 	<p>Username: john.admin</p> <p>Password: password123</p> <p>Confirm Password: password123</p> <p>Phone Number: 0123456789</p> <p>IC Number: 123456789012</p> <p>E-mail: john.admin@example.com</p> <p>Full Name: John Admin</p> <p>Role: Admin</p> <p>Image: validImage.png</p>	Users can create a new account, role assignment is successful, and a message "Signup Success!" will be displayed.

6.4.2.1.3 Test Data for User Profile (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: User Profile (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 02/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW03_01	Successful Profile Update	<ol style="list-style-type: none"> Navigate to the "Profile" page. Update profile fields (name, phone number, email, IC number, role, and image). Click "Update" button. 	Full Name: Nabil Aqmar Phone Number: 01116161332 Email: nabil@gmail.com IC Number: 010416102289 Image: newImage.png	Profile information is successfully updated, and a confirmation message "Profile Updated!" is displayed.
TW03_02	View Profile Information Correctly	<ol style="list-style-type: none"> Navigate to the "Profile" page. View the displayed user profile information (name, 	No specific test data (user views the data as presented on the page).	Profile information is correctly displayed as shown in the fields (Full Name, Phone, IC Number, etc.).

		phone number, IC number, role, email, and image).		
TW03_03	Profile Update with Empty Required Fields	<ol style="list-style-type: none"> 1. Navigate to the "Profile" page. 2. Leave one or more required fields empty (e.g., Full Name). 3. Click "Update" button. 	Full Name: (empty) Phone Number: 01116161332 Email: nabil@gmail.com IC Number: 010416102289 Image: image.png	An error message "Input Field cannot be empty" is displayed.
TW03_04	Password Change with Mismatched New Passwords	<ol style="list-style-type: none"> 1. Navigate to the "Change Password" section. 2. Enter old password. 3. Enter new password and confirm password (with mismatch). 4. Click "Update". 	Old Password: oldPass123 New Password: password123 Confirm New Password: password456	An error message "Password does not match" is displayed.
TW03_05	Profile Update with Invalid Email Format	<ol style="list-style-type: none"> 1. Navigate to the "Profile" page. 2. Enter an invalid email 	Full Name: Nabil Aqmar Phone Number: 01116161332 Email: invalidEmail	An error message "Invalid Email Format" is displayed.

		format. 3. Click "Update" button.	IC Number: 010416102289 Image: image.png	
TW03_06	Profile Update without Uploading an Image	1. Navigate to the "Profile" page. 2. Enter valid data but do not upload any image. 3. Click "Update" button.	Full Name: Nabil Aqmar Phone Number: 01116161332 Email: nabil@gmail.com IC Number: 010416102289 Image: (none uploaded)	An error message "Image cannot be empty" is displayed.

6.4.2.1.4 Test Data for Item Delivery (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Item Delivery (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 02/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW04_01	Register an item.	<ol style="list-style-type: none"> Navigate to the "Register Item" page. Enter the required details: Customer ID, Item From, Locker Location, Size. Click the "Register" button. 	Customer ID: R0001 Item From: Alat Tulisan Locker Location: UTeM Size: Small (S)	Item is successfully registered, and a confirmation message is displayed.
TW04_02	Assign item to a staff courier for delivery.	<ol style="list-style-type: none"> Navigate to the "Item Assign List" page. Select the item by checking the checkbox. 	Item Management ID: IM0010 Staff ID: C0001 (selected on the next screen after checking the item)	Item is successfully assigned to the staff courier, and confirmation is displayed.

		<ol style="list-style-type: none"> Click the "Assign to Staff" button. Select a staff member to assign. 		
TW04_03	Filter items by Item Management ID.	<ol style="list-style-type: none"> Navigate to the "Item Assign List" page. Enter the Item Management ID in the search field. Click "Search". 	Search Criteria: Item Management ID = IM0010	The item with ID IM0010 is correctly filtered and displayed.
TW04_04	Filter items by Item ID.	<ol style="list-style-type: none"> Navigate to the "Item Assign List" page. Enter the Item ID in the search field. Click "Search". 	Search Criteria: Item ID = I0010	The item with ID I0010 is correctly filtered and displayed.
TW04_05	Filter items by Locker Location.	<ol style="list-style-type: none"> Navigate to the "Item Assign List" page. Select the Locker 	Search Criteria: Locker Location = UTeM	Items with the Locker Location 'UTeM' are correctly filtered and displayed.

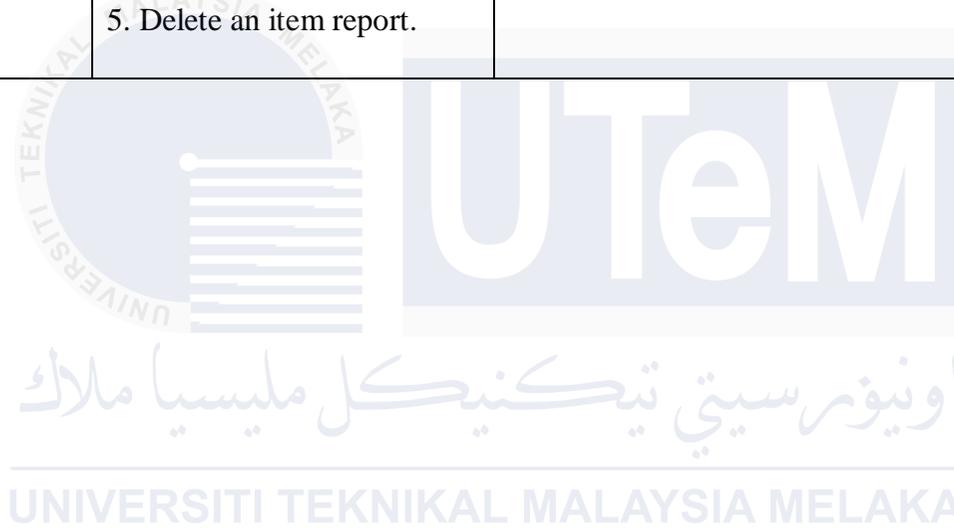
		Location from the dropdown. 3. Click "Search".		
TW04_06	Filter items by Size.	1. Navigate to the "Item Assign List" page. 2. Select the Size from the dropdown. 3. Click "Search".	Search Criteria: Size = Small (S)	Items with the size 'Small (S)' are correctly filtered and displayed.
TW04_07	Reset the filter criteria.	1. Navigate to the "Item Assign List" page. 2. Apply any filter. 3. Click the "Reset" button.	Action: Click the "Reset" button after applying a filter.	All filters are cleared, and the full list of items is displayed.

6.4.2.1.5 Test Data for Item Management Report (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Item Management Report (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 02/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW05_01	View the report on the dashboard.	<ol style="list-style-type: none"> 1. Login as Admin. 2. Navigate to the Dashboard. 3. Verify the Item Delivery Report section. 	Admin ID: A0001	The correct report is displayed on the dashboard based on the admin's data and access rights.
TW05_02	View, search, update, and delete the report on the item management page.	<ol style="list-style-type: none"> 1. Login as Admin. 2. Navigate to the "Item Management" page. 3. Search for an item using available filters (e.g., Item ID, Status). 	Admin ID: A0001 Search Criteria: Item ID = IM0005, Status = "Arrived" Update Data: Change Status to "Picked"	Admin can view, search, update, and delete the report on the item management page successfully.

		4. Update an item report (e.g., change status). 5. Delete an item report.		
--	--	---	--	--



6.4.2.1.6 Test Data for Locker Location (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Locker Location (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 03/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW06_01	Successful Locker Location Registration	<ol style="list-style-type: none"> Navigate to "Add New" under "Location's Sections". Fill in the locker location details (Name, Address, Image). Click the "Register" button. 	Location Name: UMS Location Address: UMS, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia Image: ums.png	A new locker location is successfully registered and appears in the "Location's List".
TW06_02	Search, View, Update, and Delete Locker Location Information	<ol style="list-style-type: none"> Navigate to "Location's List". Search for a specific location using "Location ID", "Name", or "Address". 	Search by Location ID: L0001 Update: Change Location Address to "Updated Address". Delete: Remove location "L0003".	The correct location information is displayed, can be updated successfully with a confirmation, and can be deleted.

		<ol style="list-style-type: none"> Click "Edit" to update. Click "Delete". 		
TW06_03	Location Search Functionality with Empty Search Fields	<ol style="list-style-type: none"> Navigate to "Location's List". Leave the search fields empty. Click the "Search" button. 	Search Fields Left Empty	All location records in the system are displayed in the list.
TW06_04	Location Update with Empty Required Fields	<ol style="list-style-type: none"> Navigate to "Location's List". Select a location to update. Remove required field data (e.g., Location Name or ID). Click "Update". 	Location Name: (empty) Location Address: UTeM, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia	An error message is displayed, indicating "Field cannot be empty".
TW06_05	Delete Location Confirmation	<ol style="list-style-type: none"> Navigate to "Location's List". Click the "Delete" button 	Select Location ID: L0002	A confirmation dialog is displayed, asking the user to confirm the deletion. Upon

		on a selected location. 3. Confirm the deletion action.		confirmation, the location is deleted.
--	--	---	--	---



6.4.2.1.7 Test Data for Locker Information (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Locker Information (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 03/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW07_01	Successful Locker Registration	<ol style="list-style-type: none"> Navigate to "Add New Locker". Fill in the locker details (Locker Name, Size, Location). Click "Register" button. 	Locker Name: Locker UMS 1 Locker Size: S Locker Location: UMS	A new locker is successfully registered and appears in the "Locker's List".
TW07_02	Search, View, Update, and Delete Locker Information	<ol style="list-style-type: none"> Navigate to "Locker's List". Search for a specific locker using "Locker ID", "Name", "Size", "Location", or "Status". 	Search by Locker ID: Q0001 Update: Change Locker Size to "M". Delete: Remove Locker "Q0004".	The correct locker information is displayed, can be updated successfully with a confirmation, and can be deleted.

		<ol style="list-style-type: none"> Click "Edit" to update. Click "Delete". 		
TW07_03	Locker Search Functionality with Empty Search Fields	<ol style="list-style-type: none"> Navigate to "Locker's List". Leave the search fields empty. Click the "Search" button. 	Search Fields Left Empty	All locker records in the system are displayed in the list.
TW07_04	Locker Update with Empty Required Fields	<ol style="list-style-type: none"> Navigate to "Locker's List". Select a locker to update. Remove required field data (e.g., Locker Name or Size). Click "Update". 	Locker Name: (empty) Locker Size: M Locker Location: UTeM	An error message is displayed, indicating "Field cannot be empty".
TW07_05	Locker Status Update	<ol style="list-style-type: none"> Navigate to "Locker's List". Select a locker. Update the locker status 	Locker ID: Q0002 Status: Available	The locker status is updated successfully and reflected in the locker list.

		(e.g., Available/Unavailable). 4. Save the changes.		
TW07_06	Delete Locker Confirmation	<ol style="list-style-type: none"> 1. Navigate to "Locker's List". 2. Click the "Delete" button on a selected locker. 3. Confirm the deletion action. 	Select Locker ID: Q0003	A confirmation dialog is displayed, asking the user to confirm the deletion. Upon confirmation, the locker is deleted successfully.
TW07_07	Pagination of Locker List	<ol style="list-style-type: none"> 1. Navigate to "Locker's List". 2. Set the number of entries to display (e.g., 10 entries per page). 3. Verify pagination. 	Entries per page: 10	The locker list is paginated correctly, showing the correct number of entries per page as per the user selection.
TW07_08	Verify that the locker ID is correctly paired	<ol style="list-style-type: none"> 1. Navigate to "Add New Locker". 2. Fill in locker details 	Locker Name: Locker UMS 3 Locker Size: M Locker Location: UMS	The locker ID is successfully generated and paired with the

	with the location ID	(Locker Name, Size, Location). 3. Register the locker. 4. Verify the pairing of Locker ID with Location ID.		correct location ID during registration.
--	----------------------	---	--	--

6.4.2.1.8 Test Data for Users Report Information (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Users Report Information (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 04/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW08_01	View, search, update, and delete Staff Courier information	<ol style="list-style-type: none"> 1. Navigate to the "Staff List" page. 2. Enter the Staff Courier ID in the search field. 3. Click "Search". 4. Select a staff member from the list and click "Edit". 5. Update details and click "Save". 6. Select a staff member and click "Delete". 	<p>Staff ID: C0001</p> <p>Search Criteria: Username = cnabil</p> <p>Update Data: Change phone number to 01111613456</p>	<ol style="list-style-type: none"> 1. The staff information for Staff ID C0001 is correctly displayed and searchable. 2. The staff information is successfully updated. 3. The staff member is successfully deleted.

TW08_02	View, search, update, and delete Customer (Receiver) information	<ol style="list-style-type: none"> 1. Navigate to the "Customer List" page. 2. Enter the Customer ID in the search field. 3. Click "Search". 4. Select a customer from the list and click "Edit". 5. Update details and click "Save". 6. Select a customer and click "Delete". 	<p>Customer ID: R0001</p> <p>Search Criteria: Full Name = Alya</p> <p>Update Data: Change Full Name to Alya Ameraa</p>	<ol style="list-style-type: none"> 1. The customer information for Customer ID R0001 is correctly displayed and searchable. 2. The customer information is successfully updated. 3. The customer is successfully deleted.
---------	--	--	---	--

6.4.2.1.9 Test Data for Convert Report to PDF (Web)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Convert Report to PDF (Web) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 04/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TW09_01	Verify successful PDF generation using the "Generate Report PDF" button.	<ol style="list-style-type: none"> Navigate to the "Item Management List" page. Click the "Generate Report PDF" button. 	Item Management List with several entries.	The item management report is converted to a PDF and downloaded without errors. The PDF should contain all the information displayed in the "Item Management List".
TW09_02	Verify that filtered data is reflected in the generated PDF report.	<ol style="list-style-type: none"> Navigate to the "Item Management List" page. Apply filters to fields (e.g., Status: Pending). Click the "Search" button. 	Filter by: - Status: Pending - Date Range: 06/16/2024 to 06/30/2024	The generated PDF contains only the filtered data as per the applied search filters and matches the data displayed on the web interface.

		4. Click "Generate Report PDF".		
TW09_03	Verify that the generated PDF report is properly formatted and readable on various PDF readers.	<ol style="list-style-type: none"> 1. Generate a report using the "Generate Report PDF" button. 2. Open the PDF using different PDF readers. 	Item Management List with several entries.	The PDF report is readable, properly aligned, and formatted without any truncation, across various PDF readers (e.g., Adobe Acrobat, browser PDF viewers).
TW09_04	Verify that the generated PDF report is stored with a meaningful filename that includes the date and time of generation.	<ol style="list-style-type: none"> 1. Navigate to the "Item Management List" page. 2. Click the "Generate Report PDF" button. 3. Check the name of the downloaded PDF. 	Current date: 29/08/2024	The downloaded PDF should have a filename like Item_Management_Report_29/08/2024.pdf, corresponding to the date the report was generated.

TW09_05	Verify that large datasets are handled and exported correctly in the PDF.	<ol style="list-style-type: none"> 1. Navigate to the "Item Management List" page. 2. Ensure a large data set is present (e.g., 10,000 records). 3. Click "Generate Report PDF". 	Dataset: 30 records	The generated PDF includes all records, even with large datasets, without performance issues or missing data.
---------	---	---	---------------------	---

6.4.2.2 Mobile Testing Data

6.4.2.2.1 Test Data for User Login (Mobile)

System: QuickLocker-Delivery
 Module/Unit: User Login (Mobile)
 Processed By: Nur Alya Binti Syamsuddin

Version: v1
 Revision: -
 Date: 04/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM01_01	To check the functionality of login when the user enters the correct username and password.	<ol style="list-style-type: none"> 1. Launch the mobile application. 2. Navigate to the login screen. 3. Enter valid_username in the Username field. 4. Enter valid_password in the Password field. 	<p>Username: valid_username</p> <p>Password: valid_password</p>	Successful login with valid credentials; appropriate error message with invalid credentials.

		5. Click on the "Login" button.		
TM01_02	To check the functionality of login when the user enters an incorrect username and password.	<ol style="list-style-type: none"> 1. Launch the mobile application. 2. Navigate to the login screen. 3. Enter invalid_username in the Username field. 4. Enter invalid_password in the Password field. 5. Click on the "Login" button. 	Username: invalid_username Password: invalid_password	An error message "Wrong Username or Password" will be displayed.
TM01_03	To check the login functionality when the Username field is empty.	<ol style="list-style-type: none"> 1. Launch the mobile application. 2. Navigate to the login screen. 3. Leave the Username field empty. 	Username: (empty) Password: valid_password	An error message "Username cannot be empty" will be displayed.

		<p>4. Enter valid_password in the Password field.</p> <p>5. Click on the "Login" button.</p>		
TM01_04	To check the login functionality when the Password field is empty.	<p>1. Launch the mobile application.</p> <p>2. Navigate to the login screen.</p> <p>3. Enter valid_username in the Username field.</p> <p>4. Leave the Password field empty.</p> <p>5. Click on the "Login" button.</p>	<p>Username: valid_username</p> <p>Password: (empty)</p>	An error message "Password cannot be empty" will be displayed.
TM01_05	To check the login role for Admin, Staff Courier,	<p>1. Launch the mobile application.</p> <p>2. Navigate to the login screen.</p>	<p>Admin:</p> <p>Username: admin_username</p> <p>Password: admin_password</p> <p>Staff Courier:</p>	Successful navigation to the next page in the application according to user role (Admin, Staff Courier, Customer).

	Customer Receiver.	<p>3. Enter the appropriate username and password according to the role (Admin, Staff Courier, Customer Receiver).</p> <p>4. Click on the "Login" button.</p>	<p>Username: staff_username</p> <p>Password: staff_password</p> <p>Customer Receiver:</p> <p>Username: customer_username</p> <p>Password: customer_password</p>	
--	--------------------	---	--	--

6.4.2.2.2 Test Data for User Registration (Mobile)

System: QuickLocker-Delivery Version: v1
Module/Unit: User Registration (Mobile) Revision: -
Processed By: Nur Alya Binti Syamsuddin Date: 05/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM02_01	To check the functionality of user registration on the signup page.	<ol style="list-style-type: none">Navigate to the signup page.Enter valid details in all required fields including User Full Name, IC Number, Email, Phone Number, Username, Password, Confirm Password, etc.Tick the terms and conditions checkbox.	<ul style="list-style-type: none">User Full Name: John DoeIC Number: 010416102289Email: nabilaqmar4321@gmail.comPhone Number: 0123456789Username: johndoePassword: Password123!Confirm Password: Password123!Terms and Conditions: Ticked	Users can create a new account, and a message "Signup Success!" will be displayed.

		4. Click on the "Signup" button.		
TM02_02	To check the registration functionality when any field is empty.	<ol style="list-style-type: none"> 1. Navigate to the signup page. 2. Leave one or more fields empty (e.g., Email). 3. Fill in other required fields including User Full Name, IC Number, Phone Number, Username, Password, Confirm Password, etc. 4. Click on the "Signup" button. 	<ul style="list-style-type: none"> - User Full Name: John Doe - IC Number: 010416102289 - Email: (Leave blank) - Phone Number: 0123456789 - Username: johndoe - Password: Password123! - Confirm Password: Password123! 	An error message "Input Field cannot be empty" will be displayed.
TM02_03	To check the registration functionality	<ol style="list-style-type: none"> 1. Navigate to the signup page. 2. Enter an invalid format in one or more fields (e.g., 	<ul style="list-style-type: none"> - User Full Name: John Doe - IC Number: 010416-10-2289 (Invalid format) - Email: 	An error message "Input Format (IC Number) is wrong, please use 010416102289 without '-'" will

	when the input format is wrong.	<p>invalid IC Number format).</p> <p>3. Fill in other required fields including User Full Name, Email, Phone Number, Username, Password, Confirm Password, etc.</p> <p>4. Click on the "Signup" button.</p>	<p>nabilaqmar4321@gmail.com</p> <ul style="list-style-type: none"> - Phone Number: 0123456789 - Username: johndoe - Password: Password123! - Confirm Password: Password123! 	be displayed below the input field.
TM02_04	To check functionality when the Username entered is not available.	<p>1. Navigate to the signup page.</p> <p>2. Enter a Username that already exists.</p> <p>3. Fill in other required fields including User Full Name, IC Number, Email, Phone Number, Password, Confirm Password, etc.</p>	<ul style="list-style-type: none"> - User Full Name: John Doe - IC Number: 010416102289 - Email: nabilaqmar4321@gmail.com - Phone Number: 0123456789 - Username: existinguser - Password: Password123! - Confirm Password: Password123! 	An error message "Username Exist!" will be displayed.

		4. Click on the "Signup" button.		
TM02_05	To check functionality when the user did not tick the terms and conditions checkbox.	<ol style="list-style-type: none"> 1. Navigate to the signup page. 2. Enter valid details in all required fields including User Full Name, IC Number, Email, Phone Number, Username, Password, Confirm Password, etc. 3. Leave the terms and conditions checkbox unticked. 4. Click on the "Signup" button. 	<ul style="list-style-type: none"> - User Full Name: John Doe - IC Number: 010416102289 - Email: nabilaqmar4321@gmail.com - Phone Number: 0123456789 - Username: johndoe - Password: Password123! - Confirm Password: Password123! - Terms and Conditions: Unticked 	An error message "Accept the terms and conditions checkbox" will be displayed below the checkbox.
TM02_06	To check the functionality of sending a	<ol style="list-style-type: none"> 1. Navigate to the signup page. 2. Enter valid details in all 	<ul style="list-style-type: none"> - User Full Name: John Doe - IC Number: 010416102289 - Email: 	A welcome email is sent to the user's registered email address

	welcome email after successful registration.	required fields including User Full Name, IC Number, Email, Phone Number, Username, Password, Confirm Password, etc. 3. Tick the terms and conditions checkbox. 4. Click on the "Signup" button.	nabilaqmar4321@gmail.com - Phone Number: 0123456789 - Username: johndoe - Password: Password123! - Confirm Password: Password123! - Terms and Conditions: Ticked	immediately after successful registration.
--	--	--	---	--

6.4.2.2.3 Test Data for User Profile (Mobile)

System: QuickLocker-Delivery
Module/Unit: User Profile (Mobile)
Processed By: Nur Alya Binti Syamsuddin

Version: v1
Revision: -
Date: 05/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM03_01	Update profile information.	<ol style="list-style-type: none">1. Navigate to the Profile section.2. Click the edit icon.3. Update the name, contact number, or email address fields.4. Save the changes.	<ul style="list-style-type: none">- Name: <i>Nabil Aqmar Bin Zuhaimi</i>- Contact Number: <i>01116161332</i>- Email Address: <i>nabil@gmail.com</i>- IC: <i>010416102289</i>	Profile information is successfully updated, and a confirmation message is displayed.
TM03_02	View profile information.	<ol style="list-style-type: none">1. Navigate to the Profile section.	<ul style="list-style-type: none">- QLD ID: <i>C0001</i>- Name: <i>Nabil Aqmar Bin Zuhaimi</i>- Contact Number: <i>01116161332</i>	Profile information is correctly displayed in the profile section.

			<p>- IC Number: <i>010416102289</i></p> <p>- Email Address: <i>nabil@gmail.com</i></p>	
TM03_03	Attempt to update the QLD ID.	<ol style="list-style-type: none"> 1. Navigate to the Profile section. 2. Click the edit icon. 3. Attempt to change the QLD ID field. 	- QLD ID: <i>C0001</i>	The QLD ID field should be non-editable, and no changes should be allowed.
TM03_04	Update profile picture.	<ol style="list-style-type: none"> 1. Navigate to the Profile section. 2. Click the profile picture. 3. Select a new profile picture from the gallery or camera. 4. Save the changes. 	- New Profile Picture: <i>testprofile.jpg</i>	Profile picture is successfully updated, and a confirmation message is displayed.

6.4.2.2.4 Test Data for Item Delivery (Mobile)

System: QuickLocker-Delivery
Module/Unit: Item Delivery (Mobile)
Processed By: Nur Alya Binti Syamsuddin

Version: v1
Revision: -
Date: 06/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM04_01	Staff courier receives the assigned item from the admin.	<ol style="list-style-type: none">Admin assigns the item to a staff courier using QLD ID.Staff courier logs into the mobile app.Staff courier navigates to the "Item" tab.Staff courier checks the pending list for the assigned item.	<p>Item ID: IM0004 Staff Courier QLD ID: C0001</p>	Staff courier successfully receives the assigned item and sees it in the pending list.

TM04_02	Receiver checks delivered item list after delivery.	<ol style="list-style-type: none"> 1. Staff courier completes the delivery by clicking the "Deliver" button. 2. Receiver logs into the mobile app. 3. Receiver navigates to the "History" tab to view delivered items. 	<p>Item ID: IM0004</p> <p>Receiver QLD ID: C0001</p>	Delivered item list is correctly displayed for the receiver.
TM04_03	Staff courier searches for a specific item in the pending list.	<ol style="list-style-type: none"> 1. Staff courier logs into the mobile app. 2. Staff courier navigates to the "Item" tab. 3. Staff courier uses the search bar to find a specific item. 	Search Query: "IM0013"	The search function works correctly, displaying the specific item "IM0013" in the pending list.
TM04_04	Staff courier attempts to receive an item	<ol style="list-style-type: none"> 1. Staff courier logs into the mobile app. 2. Staff courier navigates to 	Item ID: IM0018	The item does not appear in the pending list, and the staff courier cannot receive it.

	that is not assigned.	the "Item" tab. 3. Staff courier checks the pending list for unassigned items.		
--	-----------------------	---	--	--



6.4.2.2.5 Test Data for Item Delivery History (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Item Delivery History (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 06/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM05_01	Admin, staff courier, and receiver view the item delivery history.	<ol style="list-style-type: none"> 1. User logs into the mobile app. 2. User navigates to the "History" tab. 3. User views the item delivery history. 	<p>Staff Courier: Staff Courier QLD ID: C0001</p> <p>Receiver: Receiver QLD ID: C0002</p>	Item delivery history is correctly displayed based on the user's role and associated data.

6.4.2.2.6 Test Data for Google Map API (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Google Map API (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 06/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM06_01	User checks if the direction map displays correctly based on the chosen locker location.	<ol style="list-style-type: none"> 1. User logs into the mobile app. 2. User navigates to the "Locations" tab. 3. User selects a locker location. 4. User views the direction map. 	<p>User Location: Durian Tunggal, Melaka, Malaysia</p> <p>Locker Location: UTeM, Jalan Hang Tuah Jaya, 76100 Durian Tunggal</p>	The correct direction map is displayed with the route from "Durian Tunggal, Melaka, Malaysia" to "UTeM, Jalan Hang Tuah Jaya, 76100 Durian Tunggal".

6.4.2.2.7 Test Data for QR Code Generator (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: QR Code Generator (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 06/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM07_01	Generate QR code based on item data.	<ol style="list-style-type: none"> 1. Navigate to the "QR Code Generator" page. 2. Enter item details (e.g., Item Management ID, Size, Role ID, Locker Location ID). 3. Click "Generate QR Code". 	Item Data: qrCode: QCD0001 itemMngtId: IM0001 itemSize: S roleId: 2 lockerLocationId: L0001	QR code is generated correctly, and data such as itemMngtId, itemSize, roleId, and lockerLocationId are embedded accurately.

TM07_02	Scan QR code and retrieve item data.	<ol style="list-style-type: none"> 1. Open the QR Code Scanner. 2. Scan the generated QR code. 3. Check if the retrieved data matches the input. 	QR Code: QCD0001	Scanned QR code data is accurately retrieved and matches the associated item data (e.g., IM0001, S, roleId: 2, lockerLocationId: L0001).
TM07_03	Encrypt QR code data.	<ol style="list-style-type: none"> 1. Navigate to the "QR Code Generator" page. 2. Enter item details. 3. Click "Encrypt Data". 4. Generate QR code. 	Item Data: qrCode: QCD0001 itemMngtId: IM0001 itemSize: S roleId: 2 lockerLocationId: L0001	QR code data is encrypted correctly, ensuring the information is securely encoded.
TM07_04	Decrypt QR code data.	<ol style="list-style-type: none"> 1. Scan the encrypted QR code. 2. Use the decryption tool. 3. Retrieve and view the decrypted data. 	Encrypted QR Code: QCD0001	QR code data is decrypted successfully, revealing the original item data (e.g., IM0001, S, roleId: 2, lockerLocationId: L0001).

6.4.2.2.8 Test Data for Notification (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Notification (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 07/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM08_01	Notification Sent to Staff Courier After Assignment	<ol style="list-style-type: none"> Admin assigns an item to a staff courier for delivery. System sends a notification to the staff courier. 	Item Assigned: IM0001 Staff Courier: C0001	The staff courier (C0001) receives a notification that item IM0001 has been assigned for delivery.
TM08_02	Notification Sent to Receiver After Item Delivery	<ol style="list-style-type: none"> Staff courier marks an item as delivered. System sends a notification to the receiver. 	Item Delivered: IM0002 Receiver: R0001 Message: Hello, Syahmi! You have item(s) arrived. Please check your arrived list.	The receiver (R0001) receives a notification that item IM0002 has been delivered and is ready for pickup.

TM08_03	Receiver gets reminder to close the locker after pickup	<ol style="list-style-type: none"> Receiver opens the locker and picks up the item. Wait for the reminder notification to close the locker. 	Locker Opened: L0001 Item: IM0002 Receiver: R0001 Message: Hello, Syahmi! Don't forget to close the locker!	The receiver (C0001) receives a notification reminding them to close the locker after picking up the item.
---------	---	---	--	--

6.4.2.2.9 Test Data for Locker Functionality (Mobile with Arduino)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Locker Functionality (Mobile with Arduino) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 07/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM09_01	Open Locker using QR Code (Valid)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 4. Select available locker 5. Connect to locker device 6. Push button in app to open locker 7. Update locker and item status in system 	QR Code Data:	<ol style="list-style-type: none"> 1. Locker opens correctly 2. Locker status is updated 3. Notification is sent to recipient

		8. System sends notification to recipient		
TM09_02	Open Locker using QR Code (Invalid QR Code)	<ol style="list-style-type: none"> 1. Scan QR Code 2. QR Code fails validation 	Locker ID: Q0001	<ol style="list-style-type: none"> 1. "Invalid" error is displayed 2. Prompt user to scan a new QR Code
TM09_03	Open Locker using QR Code (Decryption Failure)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 	Incorrect Decryption Key	<ol style="list-style-type: none"> 1. "Decrypt failed" error is displayed 2. Prompt user to scan a new QR Code
TM09_04	Validate QR Code Format (Invalid)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 	Wrong QR Code Format	1. "Invalid Format" error is displayed
TM09_05	Select Available Locker (No Lockers Available)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 4. Select available locker 	No Locker Available	1. "Invalid" error is displayed

TM09_06	Connect to Locker Device (Successful)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR 	Locker Selected	1. Locker device connected successfully
TM09_07	Connect to Locker Device (Failure)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 4. Select available locker 5. Connect to locker device 	Locker Device Connection Failure	1. "Device Not Found" error is displayed
TM09_08	Open Locker (Wrong Button)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 4. Select available locker 5. Connect to locker device 	Button Locker ID: Q0003	1. "Wrong locker to close" error is displayed

		6. Push wrong button to open locker		
TM09_09	Open Locker (Correct Button)	<ol style="list-style-type: none"> 1. Scan QR Code 2. Decrypt QR Code data 3. Validate QR Code format 4. Select available locker 5. Connect to locker device 6. Push correct button to open locker 	Button Locker ID: Q0001	1. Locker opens correctly
TM09_10	Update Locker and Item Status (After Opening)	<ol style="list-style-type: none"> 1. Locker opens 2. System updates locker and item status 	Locker ID: Q0001	1. Locker and item status updated successfully
TM09_11	Send Notification to Recipient (After Opening)	<ol style="list-style-type: none"> 1. Locker opens 2. System updates locker and item status 3. System sends notification to recipient 	Locker Opened	1. Notification sent successfully

TM09_12	Close Locker (Correct Locker)	<ol style="list-style-type: none"> 1. Select the correct locker 2. Push button in app to close locker 3. Validate locker status 	Locker ID: Q0001	<ol style="list-style-type: none"> 1. Locker closes correctly 2. Locker status is updated in the system
TM09_13	Close Locker (Incorrect Locker)	<ol style="list-style-type: none"> 1. Select an incorrect locker 2. Push button in app to close locker 3. Validate locker status 	Locker ID: Q0003	<ol style="list-style-type: none"> 1. "Invalid" error is displayed 2. Prompt user to choose the correct locker
TM09_14	Update Locker Status (After Closing)	<ol style="list-style-type: none"> 1. Locker closes 2. System updates locker status 	Locker ID: Q0001	<ol style="list-style-type: none"> 1. Locker status updated successfully
TM09_15	Verify Wi-Fi connectivity using the ESP8266 module with SoftwareSerial	<ol style="list-style-type: none"> 1. Power on the locker system. 2. Initialize the ESP8266 Wi-Fi module with SoftwareSerial. 	Wi-Fi Module: ESP8266 SoftwareSerial: In use Wi-Fi Credentials: SSID: "DT121_2.4GHz" Password: "CcDT1217626"	Wi-Fi successfully connects to the network. If the pin on Arduino Uno R3 damaging the wires to connect the ESP8266,

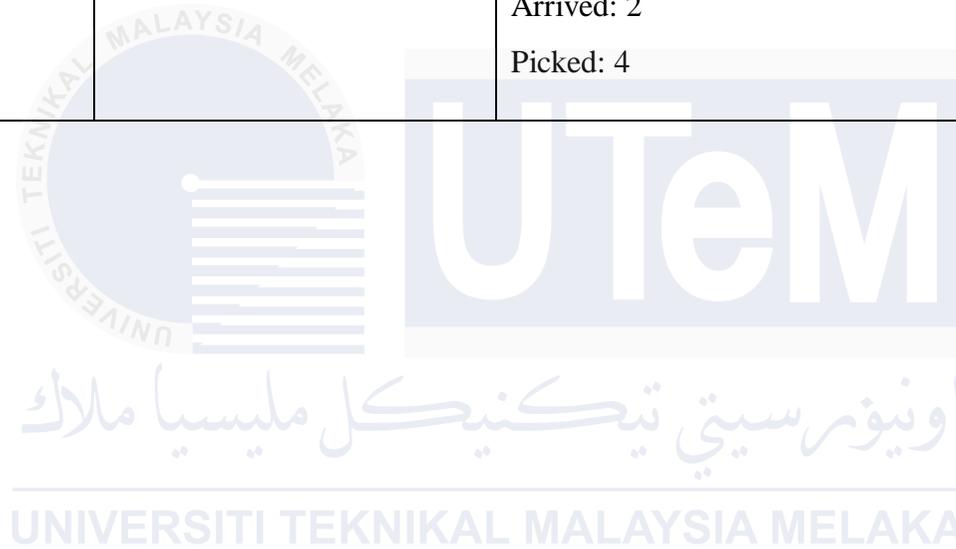
		<p>3. Enter valid Wi-Fi credentials (SSID and Password).</p> <p>4. Attempt to connect the Wi-Fi module to the network.</p>		connection fails, and an " Failed to Connect" error is displayed.
--	--	--	--	---

6.4.2.2.10 Test Data for Item Delivery Report (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Item Delivery Report (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 08/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM10_03	View Report on Dashboard (Staff Courier)	<ol style="list-style-type: none"> 1. Login as Staff Courier 2. Navigate to the Dashboard 3. View the report section 	User Role: Staff Courier Staff Courier Data: Valid QLD_ID Staff Courier: C0001 Pending: 2 Delivered: 3	The correct report is displayed based on the staff courier's data and responsibilities.
TM10_04	View Report on Dashboard (Receiver)	<ol style="list-style-type: none"> 1. Login as Receiver 2. Navigate to the Dashboard 	User Role: Receiver Receiver Data: Valid QLD_ID Receiver: R0001	The correct report is displayed based on the receiver's data and interactions.

		3. View the report section	Arrived: 2 Picked: 4	
--	--	----------------------------	-------------------------	--



6.4.2.2.11 Test Data for Forgot Password (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Forgot Password (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 08/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM11_01	Request password reset with valid username and email address.	<ol style="list-style-type: none"> 1. Open the Forgot Password page. 2. Enter a valid username. 3. Enter a valid email address associated with the username. 4. Click on the "Submit" button. 	Username: valid_user Email: user@example.com	The system verifies the username and email address. If valid, a new password is generated and sent to the email address. The user sees a confirmation message indicating that the password has been sent.
TM11_02	Attempt password reset with invalid	<ol style="list-style-type: none"> 1. Open the Forgot Password page. 2. Enter an invalid username 	Username: invalid_user Email: invalid@example.com	The system rejects the input and displays an error message indicating an invalid username or

	username or email address.	or email address. 3. Click on the "Submit" button.		email. No password reset email is sent.
TM11_03	Reset password using a token link sent to email.	<ol style="list-style-type: none"> 1. Request a password reset (as in TM11_01). 2. Check email for the reset link. 3. Click on the link in the email. 4. Enter a new password and confirm it. 5. Click "Reset Password". 	<p>Username: valid_user Email: user@example.com New Password: new_password123 Confirm Password: new_password123</p>	The system sends a password reset link with a unique token to the user's email. When the user clicks the link, the system verifies the token and allows the user to reset their password.
TM11_04	Generate and send a new random password after token-based reset.	1. Perform a password reset using the token link (as in TM11_03).	<p>Username: valid_user Email: user@example.com</p>	After the user resets their password via the link, the system generates a new random password and sends it to the user's email. A confirmation message is displayed.

TM11_05	Validate token expiration for password reset link.	<ol style="list-style-type: none"> 1. Request a password reset (as in TM11_01). 2. Do not click on the link for a specified time period (let the token expire). 3. Attempt to reset the password by clicking the link after expiration. 	Username: valid_user Email: user@example.com	The system invalidates the token if it is used after its expiration time and displays an error message indicating that the link has expired. The user must request a new password reset link.
---------	--	--	---	---

6.4.2.2.12 Test Data for SQLite Functionality (Mobile)

System: QuickLocker-Delivery Version: v1
 Module/Unit: SQLite Functionality (Mobile) Revision: -
 Processed By: Nur Alya Binti Syamsuddin Date: 09/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM12_01	Verify offline functionality using SQLite data	<ol style="list-style-type: none"> 1. Disconnect from the internet. 2. Perform a data retrieval task (e.g., view locker items). 	Role: Staff Courier/Receiver	The system retrieves and displays data from SQLite, ensuring offline functionality.
TM12_02	Prevent duplicate QR code scanning in offline mode	<ol style="list-style-type: none"> 1. Disconnect from the internet. 2. Scan a QR code. 3. Attempt to scan the same QR code again while offline. 	QR Code Data From ID: IM0049	The system prevents duplicate scanning of the same QR code in offline mode.

TM12_03	Sync and update SQLite data from MySQL when the connection is restored	<ol style="list-style-type: none"> 1. Modify data in the MySQL database (e.g., change locker status). 2. Reconnect to the internet. 3. Sync the data from MySQL to the mobile device. 	Locker ID: Q0001 Item Status: Delivered	SQLite data is synced and updated correctly from MySQL after the connection is restored.
TM12_04	Store multiple locker items and update status in SQLite	<ol style="list-style-type: none"> 1. Store multiple locker items in SQLite (e.g., deposit multiple parcels). 2. Update the status of each item via the mobile app in offline mode. 	Locker ID: Q0002 Item 1 Status: Delivered Item 2 Status: Arrived	SQLite successfully manages multiple locker items and updates the item statuses accurately based on offline status changes.

6.4.2.2.13 Test Data for Changeable Location in Locker Functionality (Mobile with Arduino)

System: QuickLocker-Delivery Version: v1
 Module/Unit: Changeable Location in Locker Functionality (Mobile with Arduino) Revision: -
 Processed By: Nabil Aqmar bin Zuhaimi Date: 08/8/2024

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results
TM13_01	Verify locker location update in the system when the first character of the Locker ID changes	<ol style="list-style-type: none"> 1. Change the first character of the Locker ID in the Arduino code (e.g., from "Q" to "L"). 2. Scan the QR code via the mobile app. 3. Verify the updated location in the system. 	<p>Original Locker ID: Q0001</p> <p>New Locker ID: L0001</p> <p>Location: UMS</p>	The system correctly updates the location of the locker when the first character of the Locker ID changes, and the mobile app reflects the updated location when the QR code is scanned.

6.5 System Usability Scale

The System Usability Scale (SUS) for the QLD project is aimed at assessing the overall usability of the application through a standardized survey. A total of 30 respondents, including admin users, staff couriers, and end customers, will be asked to complete the SUS questionnaire distributed via Google Forms. Respondents will rate their experience with the application based on factors such as ease of use, navigation, functionality, and user satisfaction.

Over a one-week period, participants will engage with key features of the application such as user authentication, item delivery, QR code scanning, and notifications. The responses will be analyzed to calculate a usability score ranging from 0 to 100. A score of 68 or above will indicate acceptable usability, with higher scores representing better user experiences. The feedback obtained will guide any final adjustments needed before the official release of the application.

6.5.1 Questionnaires for System Usability Scale

Table 6.28: Questionnaires for System Usability Scale

Question No.	Question	Description	Purpose
1	I think that I would like to use this system frequently.	Evaluates the user's desire or willingness to repeatedly use the system.	Satisfaction
2	I found the system unnecessarily complex.	Investigates whether users perceive the system as overly complicated.	Usability

3	I thought the system was easy to use.	Determines if the user perceives the system as user-friendly and straightforward.	Usability
4	I think that I would need technical support to be able to use this system.	Gauges the user's confidence in using the system without external help or training.	Performance
5	I found the various functions in this system were well integrated.	Assesses whether the user feels that different features of the system work together seamlessly.	Usability
6	I thought there was too much inconsistency in this system.	Identifies any inconsistencies in design or behavior that could confuse users.	Usability
7	I would imagine that most people would learn to use this system very quickly.	Assesses the user's belief in the ease of learning and mastering the system.	Usability
8	I found the system very cumbersome to use.	Measures how much the user finds the system difficult or inconvenient to operate.	Satisfaction

9	I felt very confident using the system.	Evaluates the user's confidence and comfort while using the system.	Performance
---	---	---	-------------

6.6 Test Results and Analysis

6.6.1 Test Result for Dynamic Testing

The software testing process must include both test results and analysis. They entail assessing the results of the testing outcomes and interpreting the data gathered to learn more about the functionality and quality of the tested program. In conclusion, test findings and analysis are vital to the software development lifecycle because they reveal information about the software's functionality, dependability, and conformance to specifications. These efforts assist in producing a more reliable product for end users while also advancing the quality of software over time.

6.6.1.1 Test Result for Website

6.6.1.1.1 Test Result and Analysis for User Login Admin (Web)

Table 6.29 shows the Test Result and Analysis for User Login Admin (Web).

Table 6.29: Test Result and Analysis for User Login Admin (Web)

Test Case ID	Actual Result	Pass	Fail
TW01_01	User successfully logged in with valid credentials and was redirected to the admin dashboard.	✓	

TW01_02	System displayed the error message "Wrong Username or Password" after entering invalid credentials.	✓	
TW01_03	System displayed the error message "Username cannot be empty" after leaving the username field empty and entering a valid password.	✓	
TW01_04	System displayed the error message "Password cannot be empty" after leaving the password field empty and entering a valid username.	✓	
TW01_05	User successfully logged in as Admin and was navigated to the admin-specific section of the application.	✓	

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

6.6.1.1.2 Test Result and Analysis for User Registration Admin (Web)

Table 6.30 shows the Test Result and Analysis for User Registration Admin (Web).

Table 6.30: User Registration Admin (Web)

Test Case ID	Actual Result	Pass	Fail
TW02_01	User successfully registered with all valid inputs. "Signup Success!" message displayed.	✓	
TW02_02	System displayed the error message "Input Field cannot be empty" when	✓	

	one or more required fields were left empty.		
TW02_03	System displayed the error message "Input Format (input error field name placed here) is wrong, please use (guided format here)" for incorrect email format.	✓	
TW02_04	System displayed the error message "Username Exist!" when a duplicate username was used.	✓	
TW02_05	The system correctly assigned the selected role, and the user received the appropriate permissions.	✓	
TW02_06	System displayed the error message "Password does not match" when passwords were mismatched.	✓	
TW02_07	System displayed the error message "Invalid image format. Please upload a valid image file" when an unsupported image format was uploaded.	✓	
TW02_08	System displayed the error message "Email already exists!" when a duplicate email was used.	✓	

TW02_09	System displayed the error message "Invalid phone number format" for an incorrect phone number format.	✓	
TW02_10	System displayed the error message "Password is too short, please enter at least 8 characters" for a short password.	✓	
TW02_11	System displayed the error message "Image cannot be empty" when no image was uploaded.	✓	
TW02_12	User successfully registered with role assignment, and a "Signup Success!" message was displayed.	✓	

6.6.1.1.3 Test Result and Analysis for User Profile (Web)

Table 6.31 shows the Test Result and Analysis for User Profile (Web).

Table 6.31: Test Result and Analysis for User Profile (Web)

Test Case ID	Actual Result	Pass	Fail
TW03_01	Profile information was successfully updated with all valid inputs, and a confirmation message "Profile Updated!" was displayed.	✓	
TW03_02	Profile information was correctly displayed on the "Profile" page,	✓	

	showing accurate details for Full Name, Phone Number, IC Number, Role, Email, and Image.		
TW03_03	System displayed the error message "Input Field cannot be empty" when a required field (Full Name) was left empty during the profile update.	✓	
TW03_04	System displayed the error message "Password does not match" when new passwords were mismatched during the password change.	✓	
TW03_05	System displayed the error message "Invalid Email Format" when an incorrect email format was entered during the profile update.	✓	
TW03_06	System displayed the error message "Image cannot be empty" when no image was uploaded during the profile update.	✓	

6.6.1.1.4 Test Result and Analysis for Item Delivery (Web)

Table 6.32 shows the Test Result and Analysis for Item Delivery (Web).

Table 6.32: Test Result and Analysis for Item Delivery (Web)

Test Case ID	Actual Result	Pass	Fail

TW04_01	Item was successfully registered with the provided details, and a confirmation message was displayed.	✓	
TW04_02	Item was successfully assigned to the staff courier, and a confirmation message was displayed after selection.	✓	
TW04_03	The item with Item Management ID "IM0010" was correctly filtered and displayed on the "Item Assign List" page.	✓	
TW04_04	The item with Item ID "I0010" was correctly filtered and displayed on the "Item Assign List" page.	✓	
TW04_05	Items with the Locker Location "UTeM" were correctly filtered and displayed on the "Item Assign List" page.	✓	
TW04_06	Items with the size "Small (S)" were correctly filtered and displayed on the "Item Assign List" page.	✓	
TW04_07	All filters were successfully cleared, and the full list of items was displayed after clicking the "Reset" button.	✓	

6.6.1.1.5 Test Result and Analysis for Item Management Report (Web)

Table 6.33 shows the Test Result and Analysis for Item Management Report (Web).

Table 6.33: Test Result and Analysis for Item Management Report (Web)

Test Case ID	Actual Result	Pass	Fail
TW05_01	The correct report was displayed on the dashboard according to the admin's data and access rights.	✓	
TW05_02	Admin was able to successfully view, search, update (change status), and delete the item report on the item management page.	✓	

6.6.1.1.6 Test Result and Analysis for Locker Location (Web)

Table 6.34 shows the Test Result and Analysis for Locker Location (Web).

Table 6.34: Test Result and Analysis for Locker Location (Web)

Test Case ID	Actual Result	Pass	Fail
TW06_01	A new locker location was successfully registered, and it appeared in the "Location's List".	✓	
TW06_02	The correct locker location information was displayed, updated successfully, and deleted as required.	✓	

TW06_03	All locker location records were displayed when the search fields were left empty.	✓	
TW06_04	An error message "Field cannot be empty" was displayed when trying to update the location with an empty required field.	✓	
TW06_05	A confirmation dialog was displayed for deletion, and upon confirmation, the location was successfully deleted.	✓	

6.6.1.1.7 Test Result and Analysis for Locker Information (Web)

Table 6.35 shows the Test Result and Analysis for Locker Information (Web).

Table 6.35: Test Result and Analysis for Locker Information (Web)

Test Case ID	Actual Result	Pass	Fail
TW07_01	A new locker was successfully registered, and it appeared in the "Locker's List".	✓	
TW07_02	The correct locker information was displayed, updated successfully (size changed), and the locker was deleted as required.	✓	

TW07_03	All locker records were displayed when the search fields were left empty.	✓	
TW07_04	An error message "Field cannot be empty" was displayed when trying to update the locker with an empty required field.	✓	
TW07_05	The locker status was successfully updated to "Available" and reflected correctly in the locker list.	✓	
TW07_06	A confirmation dialog was displayed for deletion, and upon confirmation, the locker was successfully deleted.	✓	
TW07_07	The locker list was paginated correctly, showing 10 entries per page as per the user selection.	✓	

6.6.1.1.8 Test Result and Analysis for Users Report Information (Web)

Table 6.36 shows the Test Result and Analysis for Users Report Information (Web).

Table 6.36: Test Result and Analysis for Users Report Information (Web)

Test Case ID	Actual Result	Pass	Fail
TW08_01	1. The staff information for Staff ID C0001 was correctly displayed and searchable. 2. The staff information (phone	✓	

	number) was successfully updated to 01111613456. 3. The staff member was successfully deleted from the system.		
TW08_02	1. The customer information for Customer ID R0001 was correctly displayed and searchable. 2. The customer information (Full Name) was successfully updated to Alya Ameraa. 3. The customer was successfully deleted from the system.	✓	

6.6.1.1.9 Test Result and Analysis for Convert Report to PDF (Web)

Table 6.37 shows the Test Result and Analysis for Convert Report to PDF (Web).

Table 6.37: Convert Report to PDF (Web)

Test Case ID	Actual Result	Pass	Fail
TW09_01	The item management report was successfully converted to a PDF and downloaded without errors. The PDF contained all the information displayed in the "Item Management List".	✓	
TW09_02	The generated PDF contained only the filtered data as per the applied search	✓	

	filters and matched the data displayed on the web interface.		
TW09_03	The PDF report was readable, properly aligned, and formatted across various PDF readers without any truncation.	✓	
TW09_04	The downloaded PDF had a filename like Item_Management_Report_20240829.pdf, corresponding to the date the report was generated.	✓	
TW09_05	The generated PDF included all records, even with large datasets, without performance issues or missing data.	✓	

6.6.1.2 Test Result for Mobile

6.6.1.2.1 Test Result and Analysis for User Login (Mobile)

Table 6.38 shows the Test Result and Analysis for User Login (Mobile).

Table 6.38: User Login (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM01_01	User successfully logged in with valid credentials and was redirected to the correct dashboard based on the role.	✓	
TM01_02	The system displayed the error message "Wrong Username or Password" after entering invalid credentials.	✓	

TM01_03	The system displayed the error message "Username cannot be empty" after leaving the username field empty and entering a valid password.	✓	
TM01_04	The system displayed the error message "Password cannot be empty" after leaving the password field empty and entering a valid username.	✓	
TM01_05	User successfully logged in as Admin, Staff Courier, and Customer Receiver, and was navigated to the appropriate section of the application according to the role.	✓	

6.6.1.2.2 Test Result and Analysis for User Registration (Mobile)

Table 6.39 shows the Test Result and Analysis for User Registration (Mobile).

Table 6.39: Test Result and Analysis for User Registration (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM02_01	User successfully created a new account, and the message "Signup Success!" was displayed.	✓	
TM02_02	System displayed the error message "Input Field cannot be empty" when the email field was left blank.	✓	
TM02_03	System displayed the error message "Input Format (IC Number) is wrong, please use	✓	

	010416102289 without ' - ' " when an invalid IC Number format was entered.		
TM02_04	System displayed the error message "Username Exist!" when trying to register with a username that already exists.	✓	
TM02_05	System displayed the error message "Accept the terms and conditions checkbox" when the terms and conditions checkbox was not ticked.	✓	
TM02_06	A welcome email was successfully sent to the user's registered email address immediately after successful registration.	✓	

6.6.1.2.3 Test Result and Analysis for User Profile (Mobile)

Table 6.40 shows the Test Result and Analysis for User Profile (Mobile).

Table 6.40: Test Result and Analysis for User Profile (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM03_01	Profile information was successfully updated with the new name, contact number, and email address, and a confirmation message was displayed.	✓	
TM03_02	All profile information (QLD ID, name, contact number, IC number, and email address) was correctly displayed in the profile section.	✓	
TM03_03	The QLD ID field was non-editable, and no changes were allowed as expected.	✓	
TM03_04	The profile picture was successfully updated with the selected image, and a confirmation message was displayed.	✓	

6.6.1.2.4 Test Result and Analysis for Item Delivery (Mobile)

Table 6.41 shows the Test Result and Analysis for Item Delivery (Mobile).

Table 6.41: Test Result and Analysis for Item Delivery (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM04_01	Staff courier successfully received the assigned item (Item ID: IM0004) and saw it in the pending list.	✓	
TM04_02	The receiver's delivered item list was correctly displayed after the delivery of Item ID: IM0004.	✓	
TM04_03	The search function worked correctly, displaying item "IM0013" in the pending list.	✓	
TM04_04	The item (Item ID: IM0018) was not displayed in the pending list, as it was not assigned to the staff courier.	✓	

6.6.1.2.5 Test Result and Analysis for Item Delivery History (Mobile)

Table 6.42 shows the Test Result and Analysis for Item Delivery History (Mobile).

Table 6.42: Item Delivery History (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM05_01	The item delivery history was correctly displayed for Staff Courier (C0001), and Receiver (R0002) based on their roles.	✓	

6.6.1.2.6 Test Result and Analysis for Google Map API (Mobile)

Table 6.43 shows the Test Result and Analysis for Google Map API (Mobile).

Table 6.43: Test Result and Analysis for Google Map API (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM06_01	The correct direction map was displayed from "Durian Tunggal, Melaka, Malaysia" to "UTeM, Jalan Hang Tuah Jaya, 76100 Durian Tunggal".	✓	

6.6.1.2.7 Test Result and Analysis for QR Code Generator (Mobile)

Table 6.44 shows the Test Result and Analysis for QR Code Generator (Mobile).

Table 6.44: Test Result and Analysis for QR Code Generator (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM07_01	The QR code was generated correctly with itemMngtId: IM0001, itemSize: S, roleId: 2, and lockerLocationId: L0001 accurately embedded in the code.	✓	
TM07_02	The scanned QR code data was retrieved accurately and matched the item data: IM0001, itemSize: S, roleId: 2, lockerLocationId: L0001.	✓	
TM07_03	QR code data was encrypted correctly, ensuring the information was securely encoded.	✓	
TM07_04	QR code data was decrypted successfully, and the original item data (IM0001, itemSize: S, roleId: 2, lockerLocationId: L0001) was revealed.	✓	

6.6.1.2.8 Test Result and Analysis for Notification (Mobile)

Table 6.45 shows the Test Result and Analysis for Notification (Mobile).

Table 6.45: Test Result and Analysis for Notification (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM08_01	The staff courier (C0001) received a notification that item IM0001 was assigned for delivery.	✓	
TM08_02	The receiver (R0001) received a notification with the message "Hello, Syahmi! You have item(s) arrived. Please check your arrived list."	✓	
TM08_03	The receiver (R0001) received a reminder notification with the message "Hello, Syahmi! Don't forget to close the locker!" after picking up item IM0002 from locker L0001.	✓	

6.6.1.2.9 Test Result and Analysis for Locker Functionality (Mobile with Arduino)

Table 6.46 shows the Test Result and Analysis for Locker Functionality (Mobile with Arduino).

Table 6.46: Test Result and Analysis for Locker Functionality (Mobile with Arduino)

Test Case ID	Actual Result	Pass	Fail
TM09_01	Locker opened successfully, locker status updated, and recipient received a notification	✓	
TM09_02	"Invalid" error displayed, user prompted to scan a new QR code	✓	
TM09_03	"Decrypt failed" error displayed, user prompted to scan a new QR code	✓	
TM09_04	"Invalid Format" error displayed after validating the QR code format	✓	
TM09_05	"Invalid" error displayed, no lockers available	✓	
TM09_06	Locker device connected successfully after QR code validation	✓	
TM09_07	"Device Not Found" error displayed after failed connection to locker device	✓	
TM09_08	"Wrong locker to close" error displayed after pushing wrong button	✓	

TM09_09	Correct locker opened successfully after pushing the correct button	✓	
TM09_10	Locker and item status updated successfully after opening the locker	✓	
TM09_11	Notification sent successfully to recipient after locker opened	✓	
TM09_12	Locker closed correctly, status updated in the system	✓	
TM09_13	"Invalid" error displayed after selecting incorrect locker to close	✓	
TM09_14	Locker status updated successfully after locker closed	✓	
TM09_15	Wi-Fi successfully connects to the network. If the pin on Arduino Uno R3 damaging the wires to connect the ESP8266, connection fails, and an "Failed to Connect" error is displayed.		✓

6.6.1.2.10 Test Result and Analysis for Item Delivery Report (Mobile)

Table 6.47 shows the Test Result and Analysis for Item Delivery Report (Mobile).

Table 6.47: Test Result and Analysis for Item Delivery Report (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM10_03	The correct report is displayed based on the staff courier's data and responsibilities	✓	
TM10_04	The correct report is displayed based on the receiver's data and interactions	✓	

6.6.1.2.11 Test Result and Analysis for Forgot Password (Mobile)

Table 6.48 shows the Test Result and Analysis for Forgot Password (Mobile).

Table 6.48: Test Result and Analysis for Forgot Password (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM11_01	The system verifies the username and email address. If valid, a new password is generated and sent to the email address. The user sees a confirmation message indicating that the password has been sent.	✓	
TM11_02	The system rejects the input and displays an error message indicating an invalid username or email. No password reset email is sent.	✓	

TM11_03	The system sends a password reset link with a unique token to the user's email. When the user clicks the link, the system verifies the token and allows the user to reset their password.	✓	
TM11_04	After the user resets their password via the link, the system generates a new random password and sends it to the user's email. A confirmation message is displayed.		
TM11_05	The system invalidates the token if it is used after its expiration time and displays an error message indicating that the link has expired. The user must request a new password reset link.		

6.6.1.2.12 Test Result and Analysis for SQLite Functionality (Mobile)

Table 6.49 shows the Test Result and Analysis for Item Delivery Report (Mobile).

Table 6.49: Test Result and Analysis for SQLite Functionality (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM12_01	The system successfully retrieved and displayed data from SQLite when there was no internet connection.	✓	
TM12_02	The system prevented the duplicate scanning of the same QR code in offline mode.	✓	

TM12_03	SQLite data was synced and updated correctly from MySQL after the connection was restored.	✓	
TM12_04	Multiple locker items were stored and managed correctly in SQLite, and item statuses were updated accurately.	✓	

6.6.1.2.13 Test Result and Analysis for Changeable Location in Locker Functionality (Mobile with Arduino)

Table 6.50 shows the Test Result and Analysis for Changeable Location in Locker Functionality (Mobile with Arduino).

Table 6.50: Test Result and Analysis for Item Delivery Report (Mobile)

Test Case ID	Actual Result	Pass	Fail
TM13_01	The system correctly updated the locker location when the first character of the Locker ID changed, and the mobile app reflected the updated location when the QR code was scanned.	✓	

6.6.2 Summary of Recorded Test Case

Table 6.51: Summary of Recorded Test Case

Module	Test Case ID	Total Success
User Login Admin (Web)	TW01_01 – TW01_05	5
User Registration Admin (Web)	TW02_01 – TW02_12	12
User Profile (Web)	TW03_01 – TW03_06	6
Item Delivery (Web)	TW04_01 – TW04_07	7
Item Management Report (Web)	TW05_01 – TW05_02	2
Locker Location (Web)	TW06_01 – TW06_05	5
Locker Information (Web)	TW07_01 – TW07_07	7
Users Report Information (Web)	TW08_01 – TW08_02	2
Convert Report to PDF (Web)	TW09_01 – TW09_05	5
User Login (Mobile)	TM01_01 – TM01_05	5
User Registration (Mobile)	TM02_01 – TM02_06	6
User Profile (Mobile)	TM03_01 – TM03_04	4

Item Delivery (Mobile)	TM04_01 – TM04_04	4
Item Delivery History (Mobile)	TM05_01	1
Google Map API (Mobile)	TM06_01	1
QR Code Generator (Mobile)	TM07_01 – TM07_04	4
Notification (Mobile)	TM08_01 – TM08_03	3
Locker Functionality (Mobile)	TM09_01 – TM09_15	14
Item Delivery Report (Mobile)	TM10_03 – TM10_04	2
Forgot Password (Mobile)	TM11_01 – TM11_05	5
SQLite Functionality	TM12_01 – TM12_05	5
Changeable Location in Locker Functionality (Mobile with Arduino)	TM13_01	1
Total		105

Table 6.51 summarizes the test cases for the QLD project, documenting 22 modules with 104/105 successful tests. However, the TM09_15 test case for Locker Functionality (Mobile) failed due to technical limitations and configuration challenges with the Arduino Uno.

The failure occurred because the Arduino Uno can only support one communication module at a time—either Bluetooth or Wi-Fi. Initially, the Wi-Fi module needed to connect to specific pins (e.g., Rx and Tx), but these pins were already occupied by the Bluetooth module. To work around this issue, SoftwareSerial was used to connect the ESP8266 Wi-Fi module to alternative pins.

Despite this workaround, there was a problem with controlling the 5.0V power supply without a voltage regulator (such as the AMS1117-3.3 or LD1117-3.3) to reduce the voltage to 3.3V, which is necessary for the safe operation of the ESP8266 module. This oversight led to damage in the wiring due to the unregulated voltage, ultimately causing the Wi-Fi connection to fail during the test.

6.6.3 User Usability Testing Result and Analysis

6.6.3.1 User Usability Testing Result

Table 6.52: User Usability Testing Result

Question	Frequency:				
	1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree				
	1	2	3	4	5
I think that I would like to use this system frequently.	1	0	0	10	3
I found the system unnecessarily complex.	1	11	2	0	0

I thought the system was easy to use.	0	1	0	7	6
I think that I would need technical support to be able to use this system.	0	9	3	2	0
I found the various functions in this system were well integrated.	0	0	0	11	3
I thought there was too much inconsistency in this system.	2	12	0	0	0
I would imagine that most people would learn to use this system very quickly.	0	0	2	10	2
I found the system very cumbersome to use.	5	9	0	0	0
I felt very confident using the system.	0	0	1	12	1
I needed to learn a lot of things before I could get going with this system.	1	10	2	1	0

6.6.3.2 User Usability Testing Analysis and Result

The usability of the QLD system was evaluated using the System Usability Scale (SUS) questionnaire. The questionnaire was distributed to 14 participants, including Admin Users, Staff Couriers, and Receivers, who were asked to rate their experience based on 10 specific statements related to the usability of the system. Each

question used a five-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree."

6.6.3.2.1 Calculate User Usability Testing

Below are the results and the calculations for each question, broken down into positive and negative questions.

Positive Questions:

- **Q1: I think that I would like to use this system frequently.**

$$\circ (1 \times 0) + (0 \times 1) + (0 \times 2) + (10 \times 3) + (3 \times 4) = 42$$

- **Q3: I thought the system was easy to use.**

$$\circ (0 \times 0) + (1 \times 1) + (0 \times 2) + (7 \times 3) + (6 \times 4) = 56$$

- **Q5: I found the various functions in this system were well integrated.**

$$\circ (0 \times 0) + (0 \times 1) + (0 \times 2) + (11 \times 3) + (3 \times 4) = 45$$

- **Q7: I would imagine that most people would learn to use this system very quickly.**

$$\circ (0 \times 0) + (0 \times 1) + (2 \times 2) + (10 \times 3) + (2 \times 4) = 46$$

- **Q9: I felt very confident using the system.**

$$\circ (0 \times 0) + (0 \times 1) + (1 \times 2) + (12 \times 3) + (1 \times 4) = 44$$

Negative Questions:

- **Q2: I found the system unnecessarily complex.**

$$\circ (1 \times 4) + (11 \times 3) + (2 \times 2) + (0 \times 1) + (0 \times 0) = 46$$

- **Q4: I think that I would need technical support to be able to use this system.**

$$\circ (0 \times 4) + (9 \times 3) + (3 \times 2) + (2 \times 1) + (0 \times 0) = 41$$

- **Q6: I thought there was too much inconsistency in this system.**

- $(2 \times 4) + (12 \times 3) + (0 \times 2) + (0 \times 1) + (0 \times 0) = 42$
- **Q8: I found the system very cumbersome to use.**
 - $(5 \times 4) + (9 \times 3) + (0 \times 2) + (0 \times 1) + (0 \times 0) = 71$
- **Q10: I needed to learn a lot of things before I could get going with this system.**
 - $(1 \times 4) + (10 \times 3) + (2 \times 2) + (1 \times 1) + (0 \times 0) = 43$

Finally, the System Usability Scale (SUS) score was calculated using the following method:

1. **Sum of all positive questions (Q1, Q3, Q5, Q7, Q9):**

- $42 + 56 + 45 + 46 + 44 = 233$

2. **Sum of all negative questions (Q2, Q4, Q6, Q8, Q10):**

- $46 + 41 + 42 + 71 + 43 = 243$

3. **Total Score (Positive + Negative):**

- **476**

4. **Average Score per Question:**

- $476 / 10 = 47.6$

5. **Overall, SUS Score:**

- SUS score formula: Overall SUS score = 47.6×2.5
- Result: **119.0**

6. **Average SUS Score per Participant:**

- The overall SUS score of **119.0** was divided by the number of participants (14):
 - **85.0**

Interpretation of the SUS Score

The average SUS score across all participants is **85.0** out of 100. Based on SUS score interpretation standards, this score falls within the "**Excellent**" range, indicating that the QLD system has excellent usability performance.

Table 6.53: Interpretation of the SUS Score

SUS Score	Grade	Rating
> 80.3	A	Excellent
68 – 80.3	B	Good
68	C	Okay (Average)
51 – 68	D	Poor
< 51	F	Awful

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

6.6.4 Analyze on Issues in College Courier Service Delivery and Opinions on the QLD Project

6.6.4.1 Current Manual Delivery System at UTEM

The current manual delivery system at UTEM presents several challenges, as revealed by survey respondents. Satisfaction levels with the existing process are varied, with a majority of the 31 respondents rating their satisfaction as moderate: 3.2% rated it as 1, 19.4% as 2, 32.3% as 3, 25.8% as 4, and 19.4% as 5. These ratings suggest a general dissatisfaction, primarily falling within the lower to middle satisfaction range. The survey also highlighted key challenges faced by users of the current system: 67.7% of respondents reported delayed deliveries, 41.9% faced missed deliveries, all respondents (100%) indicated difficulties in locating parcels, pointing to a significant logistical issue, and 77.4% expressed concerns about the security of parcels. These findings underscore the inefficiencies and inadequacies of the current manual delivery system, demonstrating a clear need for improvement.

6.6.4.2 Proposed QuickLocker-Delivery (QLD) System

In response to these issues, the QuickLocker-Delivery (QLD) system is proposed as a solution. Survey feedback suggests several beneficial features of the QLD system, including QR code access, which was deemed beneficial by all 31 respondents (100%), indicating a strong preference for secure and easy parcel retrieval methods. Additionally, 61.3% valued real-time tracking for increased visibility and reduced uncertainty regarding parcel locations. Automated notifications were appreciated by 71% of respondents for keeping them informed about parcel status, while 93.5% highlighted secure lockers as a crucial feature, addressing the security concerns raised with the current system. Moreover, 77.4% found the user-friendly mobile app to enhance their overall experience. Notably, 100% of the respondents believe that the QLD system would improve the delivery process at UTEM, indicating unanimous support for its implementation.

6.6.4.3 Conclusion and Recommendations

In conclusion, the analysis clearly identifies several significant challenges associated with the current manual delivery system at UTEM, such as delays, missed deliveries, difficulty locating parcels, and security concerns. The survey results show a strong preference for the QuickLocker-Delivery (QLD) system, which offers features that directly address these issues. It is recommended that UTEM proceed with the implementation of the QLD system, given unanimous support and the identified benefits. Additionally, further communication with stakeholders and training sessions on using the new features, especially QR code access and the mobile app, should be conducted to ensure a smooth transition. Continuous monitoring of the system's performance and regular feedback collection will be essential to ensure ongoing improvements and to address any new challenges that may arise. By adopting the QLD system, UTEM has the potential to significantly enhance the efficiency, security, and user satisfaction associated with parcel delivery on campus.

6.7 Conclusion

This chapter summarized the testing phase of the QLD Project, including the test plan, test environment, test schedule, test strategy, test design, and test results. The next activities involve final adjustments based on test results and preparing for the deployment of the application.

CHAPTER 7: PROJECT CONCLUSION

7.1 Observation on Weaknesses and Strengths

The QLD project exhibits a mix of strengths and weaknesses. A key weakness is the absence of a PHP framework like Laravel for the website, which hampers the scalability and maintainability of the web application. This omission makes the development process more tedious and less structured than it could be. Additionally, the locker system, while operational, faced issues with Wi-Fi connectivity, which prevented the system from effectively scanning existing items in the locker and retrieving or sending data to the MySQL database.

On the other hand, the project demonstrates significant strengths. The overall system flow is smooth and intuitive, offering users an easy-to-use interface that enhances their experience. The use of Flutter for mobile development is another strength, providing cross-platform compatibility and efficient performance. The system also incorporates robust security measures, ensuring user data safety while maintaining efficient delivery and notification management. This balance between usability and security has been well-received by users. Feedback from users has been largely positive, highlighting the seamless integration of IoT technology in managing the smart locker functionality and the overall ease of navigation.

7.2 Propositions for Improvement

To enhance the QLD system, several improvements are proposed. Firstly, migrating the web application to the Laravel framework would significantly improve its structure, security, and scalability. Laravel's built-in tools for authentication, database management, and routing would simplify future maintenance and expansion. Secondly, adding Wi-Fi features to the locker system is crucial, enabling the lockers to effectively communicate with the server for real-time data retrieval and updates. This enhancement would improve system performance, particularly in managing

dynamic locker assignments and prioritizing deliveries. Lastly, improving the UI/UX design for both web and mobile platforms would increase user satisfaction by making the system more visually appealing and functional.

7.3 Project Contribution

The QLD project contributes significantly to the university, faculty, and individuals involved. For the university, it offers a practical solution to courier service challenges, serving as a model for similar systems in other departments or institutions. Faculty members benefit from the project's demonstration of applied knowledge in IoT, web programming, and mobile app development, reinforcing the real-world application of theoretical concepts. On a personal level, the project has enhanced my skills in Flutter development, IoT integration, and web application security, providing valuable hands-on experience in problem-solving and project management. The user manual for the system can be found in Appendix C.

7.4 Conclusion

In conclusion, the QLD system successfully meets the objectives set at the project's outset. It effectively addresses issues in university courier services, provides a functional IoT-based locker system using QR code technology, and demonstrates its effectiveness through a user-friendly mobile application. The integration of IoT and mobile technology has significantly improved security, efficiency, and user experience in courier service management. While there are areas for improvement, particularly in the web application's back-end framework and the locker algorithm, the system proves to be a valuable tool for enhancing university logistics. Further refinement based on the provided propositions will ensure the system's scalability and robustness for broader implementation.

REFERENCES

- Patil, P. (n.d.). Agile Software Development Methods Comparison. LinkedIn. Retrieved from <https://www.linkedin.com/pulse/agile-software-development-methods-comparison-prachi-patil/>
- Flutter Documentation (n.d.). Flutter Documentation. Flutter Dev. Retrieved from <https://flutter.dev/docs>
- XAMPP Documentation (n.d.). XAMPP Documentation. Apache Friends. Retrieved from <https://www.apachefriends.org/docs/>
- Arduino Documentation (n.d.). Arduino Documentation. Arduino. Retrieved from <https://www.arduino.cc/en/Guide>
- OneSignal Documentation (n.d.). OneSignal Documentation. OneSignal. Retrieved from <https://documentation.onesignal.com/>
- Firebase Documentation (n.d.). Firebase Documentation. Firebase. Retrieved from <https://firebase.google.com/docs>
- MySQL Documentation (n.d.). MySQL Documentation. Oracle. Retrieved from <https://dev.mysql.com/doc/>
- Software Testing Fundamentals (n.d.). Software Testing Fundamentals: Bottom-Up Integration Testing. Software Testing Fundamentals. Retrieved from <https://softwaretestingfundamentals.com/integration-testing/>
- ISTQB Foundation Level Syllabus (n.d.). International Software Testing Qualifications Board (ISTQB) Foundation Level Syllabus: Black-Box Testing. ISTQB. Retrieved from <https://www.istqb.org/>

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The Art of Software Testing* (3rd ed.). John Wiley & Sons.

UI/UX Trend (n.d.). SUS Calculator. UI/UX Trend. Retrieved from <https://uiuxtrend.com/sus-calculator/>

UI/UX Trend (n.d.). Measuring System Usability Scale (SUS). UI/UX Trend. Retrieved from <https://uiuxtrend.com/measuring-system-usability-scale-sus/>



APPENDICES

Appendix A: System Usability Scale (SUS) Questionnaire and Result Data

QLD
QuickLocker-Delivery

QuickLocker-Delivery System Usability Scale

I am conducting a usability assessment for the QuickLocker-Delivery system. This survey aims to gather feedback on your experience using the system, which is designed to enhance parcel delivery services in both multi-level apartment buildings and company facilities. Your insights will help us evaluate the system's ease of use, effectiveness, and overall user satisfaction. The results will be used to refine the system to better meet your needs. Thank you for your participation.

nabilaqmar01@gmail.com [Switch account](#)

Not shared

[Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix A

QLD
QuickLocker-Delivery

QuickLocker-Delivery System Usability Scale

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Participant Information

Name *

Your answer

Email Address *

Your answer

Age *

Under 18

18 - 35

35 - 45

Above 45

Participating Roles *

Administrator

Staff Courier

Receiver

[Back](#) [Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix A

QLD
QuickLocker-Delivery

QuickLocker-Delivery System Usability Scale

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

System Usability Testing

Please rate your agreement with the following statements based on your experience using the QuickLocker-Delivery system.
For each statement, choose a score from 1 to 5, where:

- 1 = Strongly Disagree
- 2 = Disagree
- 3 = Neutral
- 4 = Agree
- 5 = Strongly Agree

1. I think that I would like to use this system frequently. *

1 2 3 4 5

Strongly Disagree Strongly Agree

Appendix A

2. I found the system unnecessarily complex. *

1 2 3 4 5

Strongly Disagree Strongly Agree

3. I thought the system was easy to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

4. I think that I would need technical support to be able to use this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

5. I found the various functions in this system were well integrated. *

1 2 3 4 5

Strongly Disagree Strongly Agree

6. I thought there was too much inconsistency in this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

Appendix A

7. I would imagine that most people would learn to use this system very quickly. *

1 2 3 4 5

Strongly Disagree Strongly Agree

8. I found the system very cumbersome to use. *

1 2 3 4 5

Strongly Disagree Strongly Agree

9. I felt very confident using the system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

10. I needed to learn a lot of things before I could get going with this system. *

1 2 3 4 5

Strongly Disagree Strongly Agree

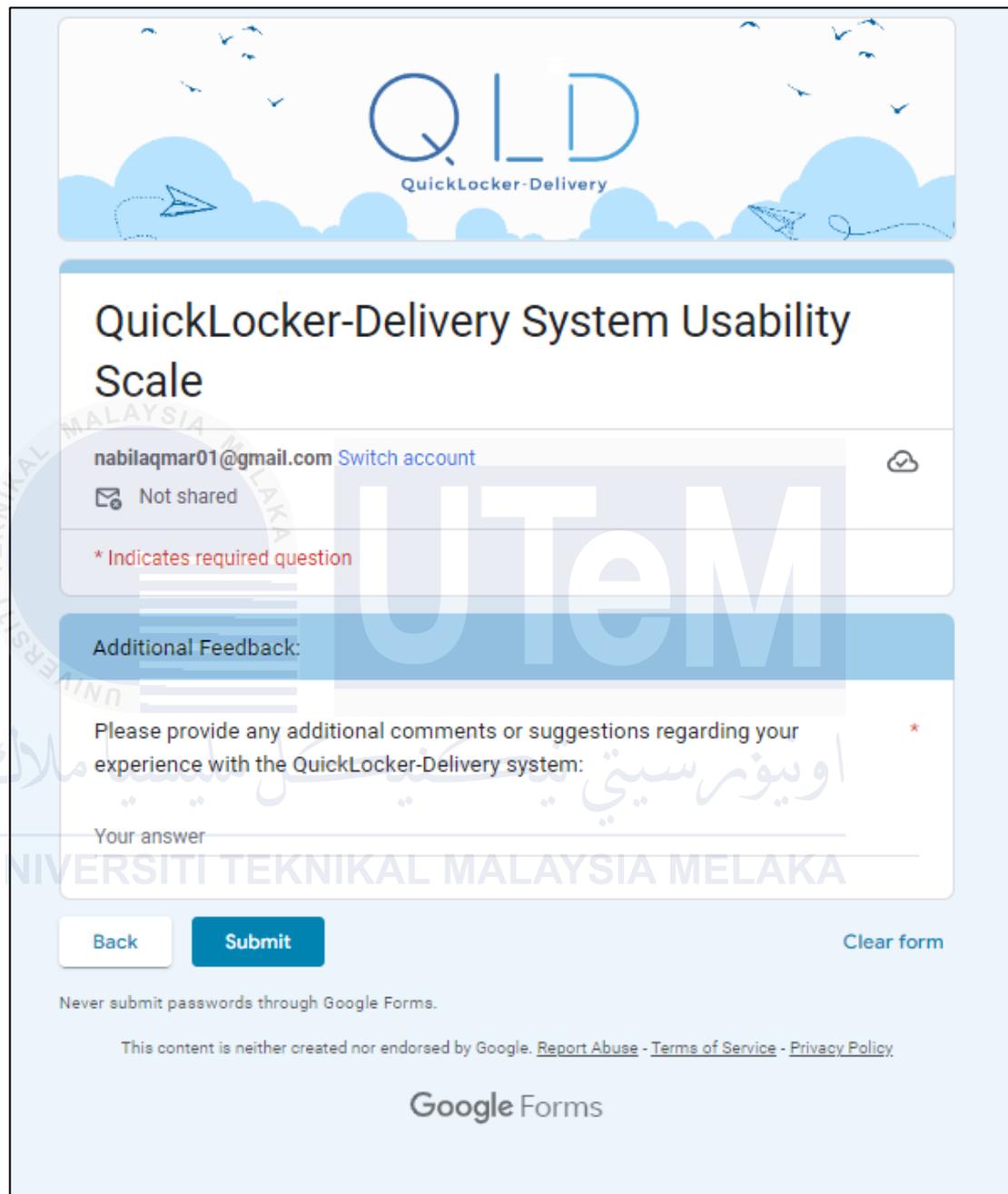
[Back](#) [Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix A



The image shows a Google Forms survey interface. At the top, there is a header with the logo 'QLD QuickLocker-Delivery' and a decorative background of clouds and birds. The main title of the survey is 'QuickLocker-Delivery System Usability Scale'. Below the title, the user's email address 'nabilaqmar01@gmail.com' is displayed, along with a 'Switch account' link and a 'Not shared' status. A red asterisk indicates a required question. The question text is 'Please provide any additional comments or suggestions regarding your experience with the QuickLocker-Delivery system:'. Below the question, there is a text input field with the placeholder 'Your answer'. At the bottom of the form, there are three buttons: 'Back', 'Submit', and 'Clear form'. The footer contains the text 'Never submit passwords through Google Forms.' and 'This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy'. The Google Forms logo is at the very bottom.

QLD
QuickLocker-Delivery

QuickLocker-Delivery System Usability Scale

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Additional Feedback:

Please provide any additional comments or suggestions regarding your experience with the QuickLocker-Delivery system: *

Your answer

[Back](#) [Submit](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix A

Questions Responses **14** Settings

14 responses [Link to Sheets](#) ⋮

Accepting responses

Summary Question Individual

Participant Information

Name

14 responses

Nabil Aqmar

Muhammad Aiman

Muhammad Syahir

Suhail Azmi

Hafiz Luqman

Lim Wei jie

Muhammad Fikri

Noor Lela Binti Hassim

Dines Kumar

Email Address

14 responses

nabilaqmar01@gmail.com

aimanhakeem@gmail.com

syahir21@gmail.com

suhailAzmi46@gmail.com

haf1zlq@gmail.com

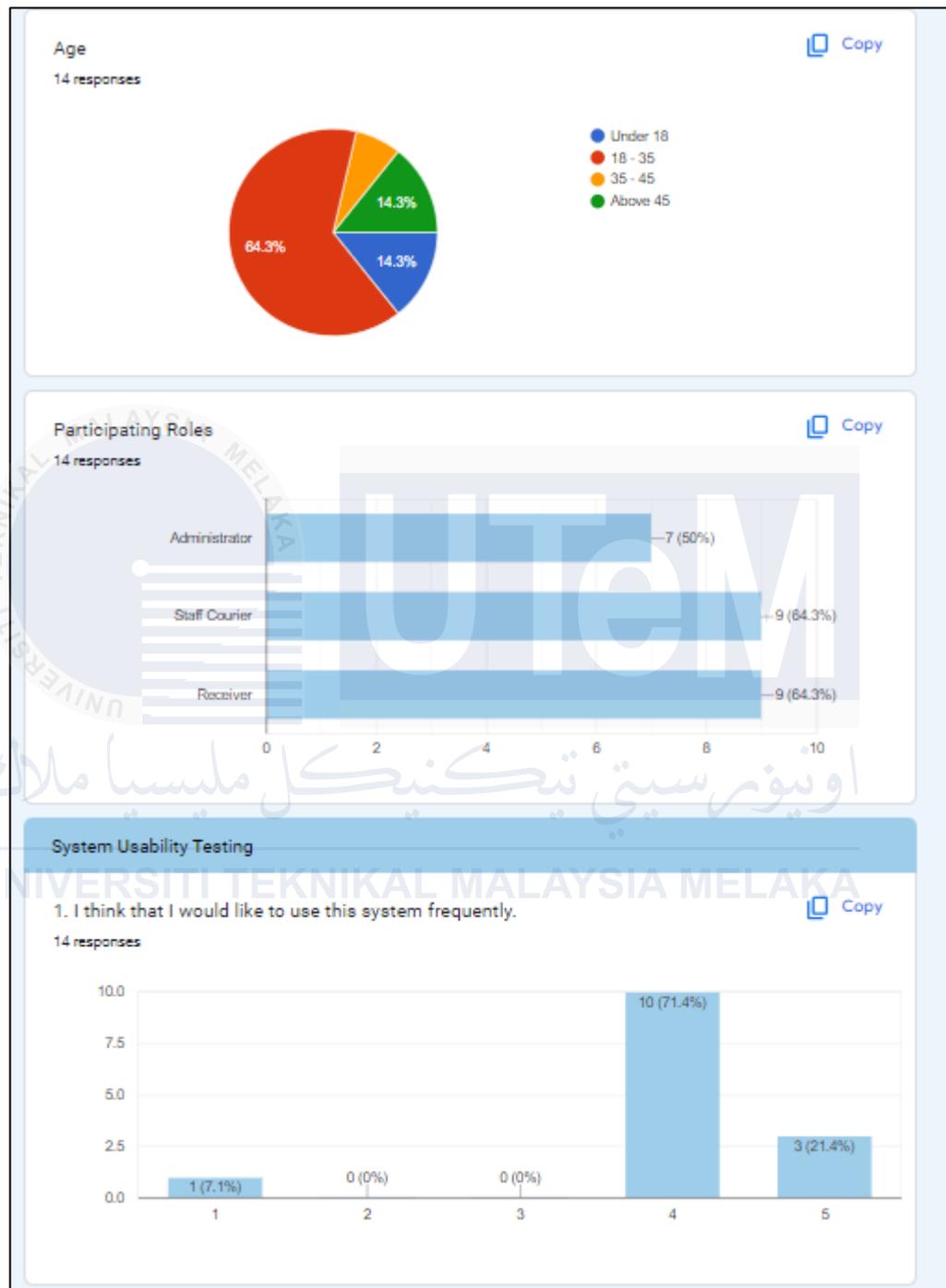
lim764@gmail.com

fikrif4dzil01@gmail.com

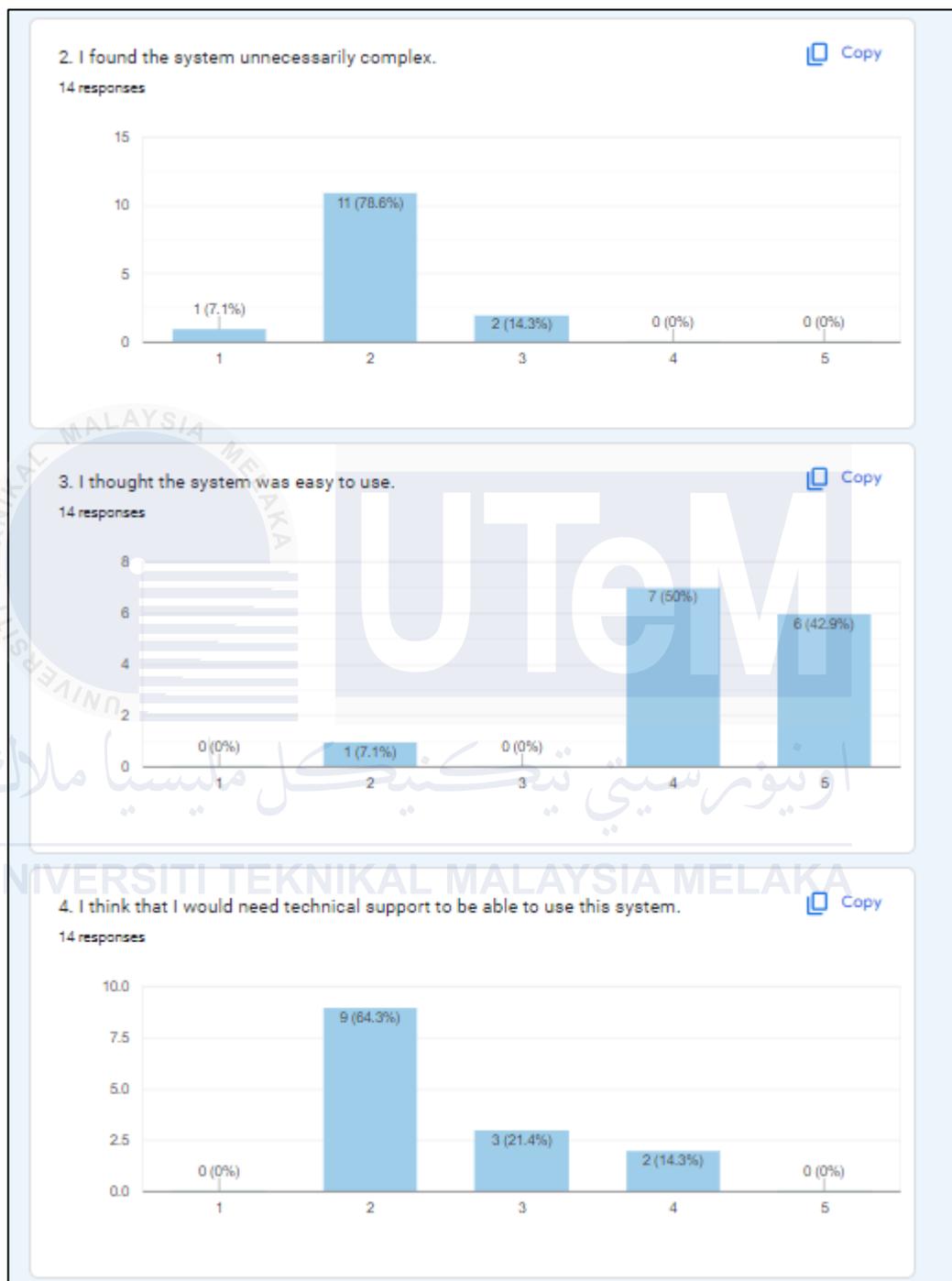
noorlela71@gmail.com

dinesk648@gmail.com

Appendix A



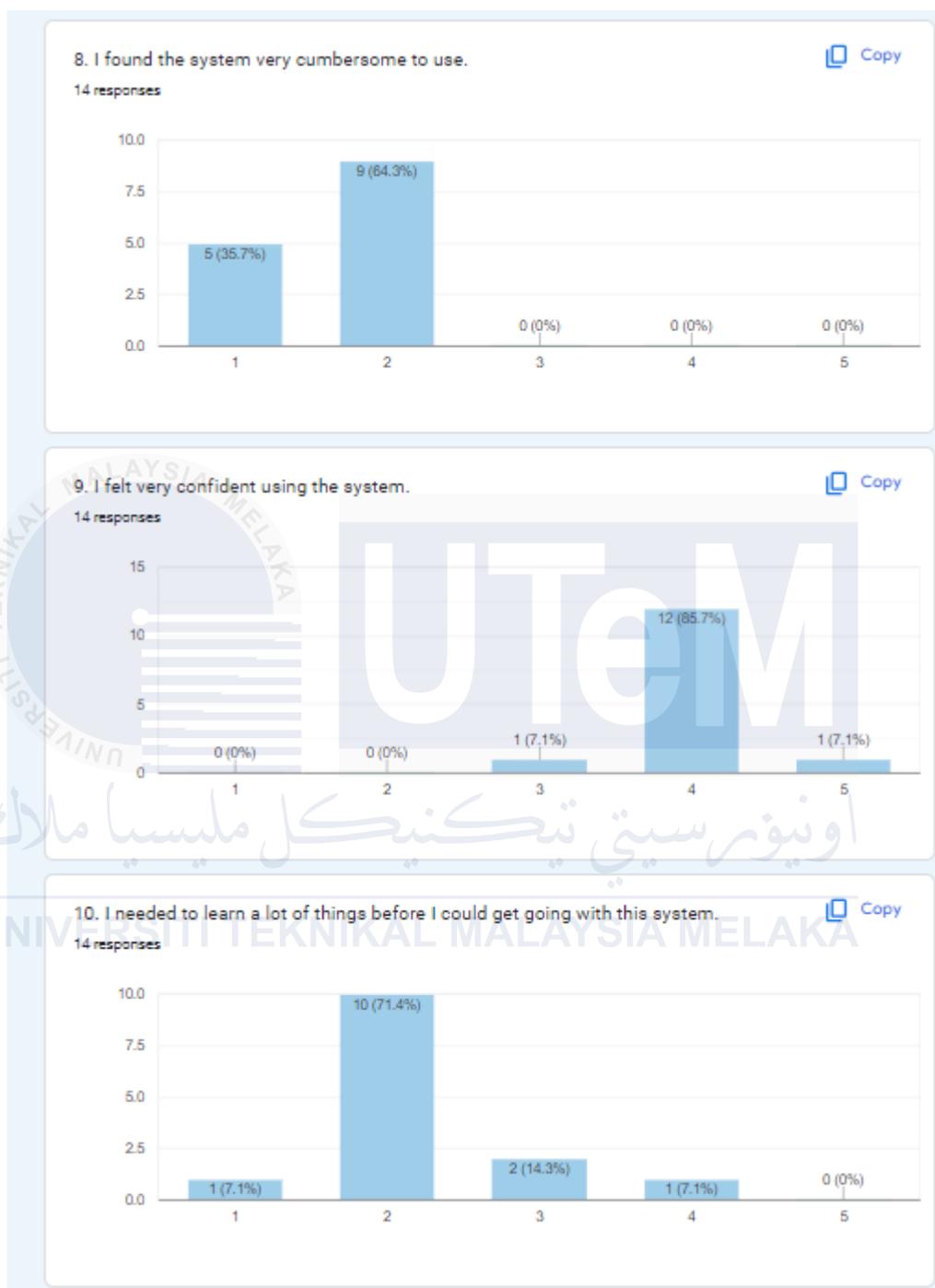
Appendix A



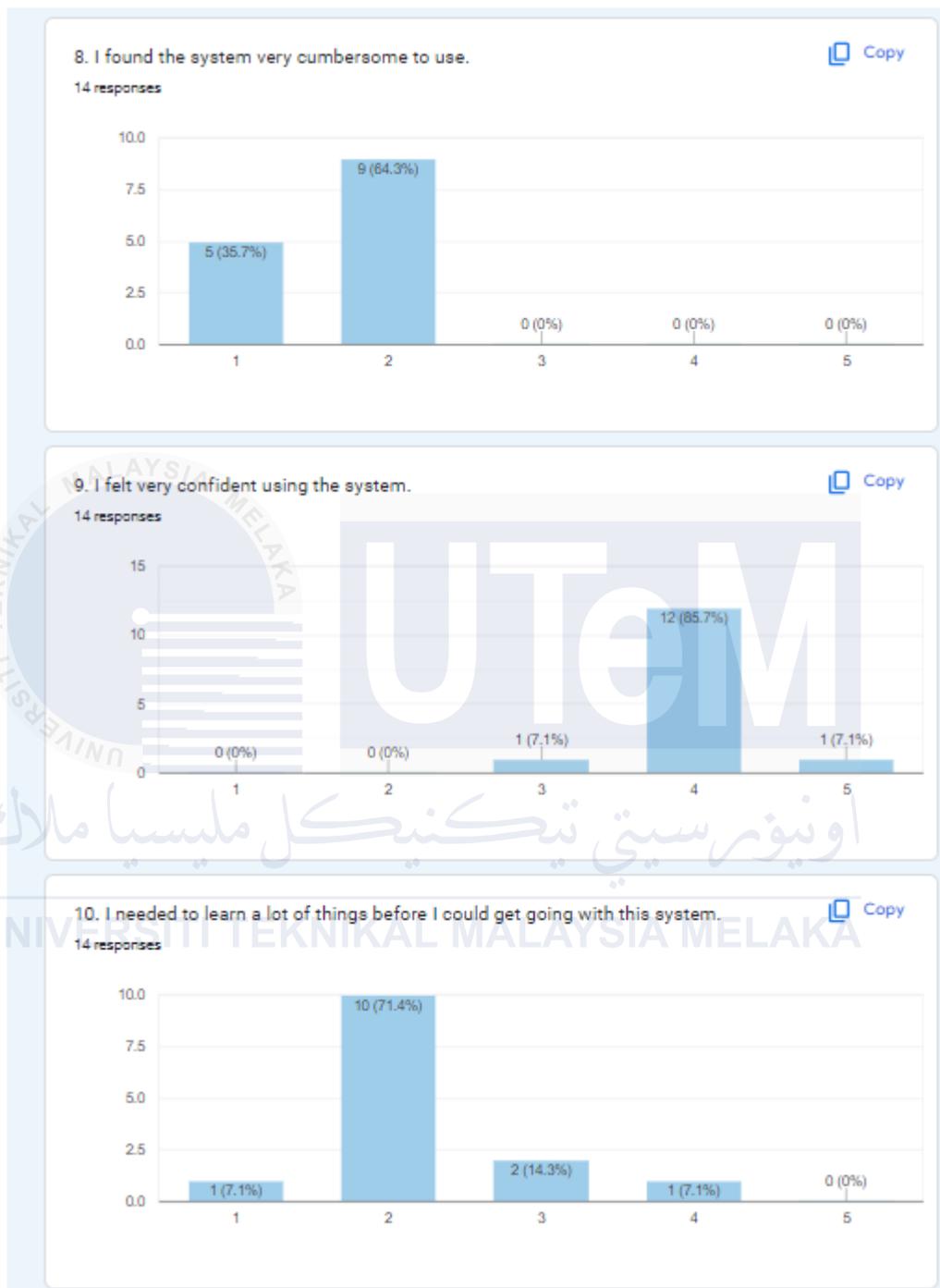
Appendix A



Appendix A



Appendix A



Appendix A

Additional Feedback:

Please provide any additional comments or suggestions regarding your experience with the QuickLocker-Delivery system:

14 responses

-
- Okay
- Well structured. Make some bar graft for report
- Need to improve locker
- Physical Admin kurang menarik

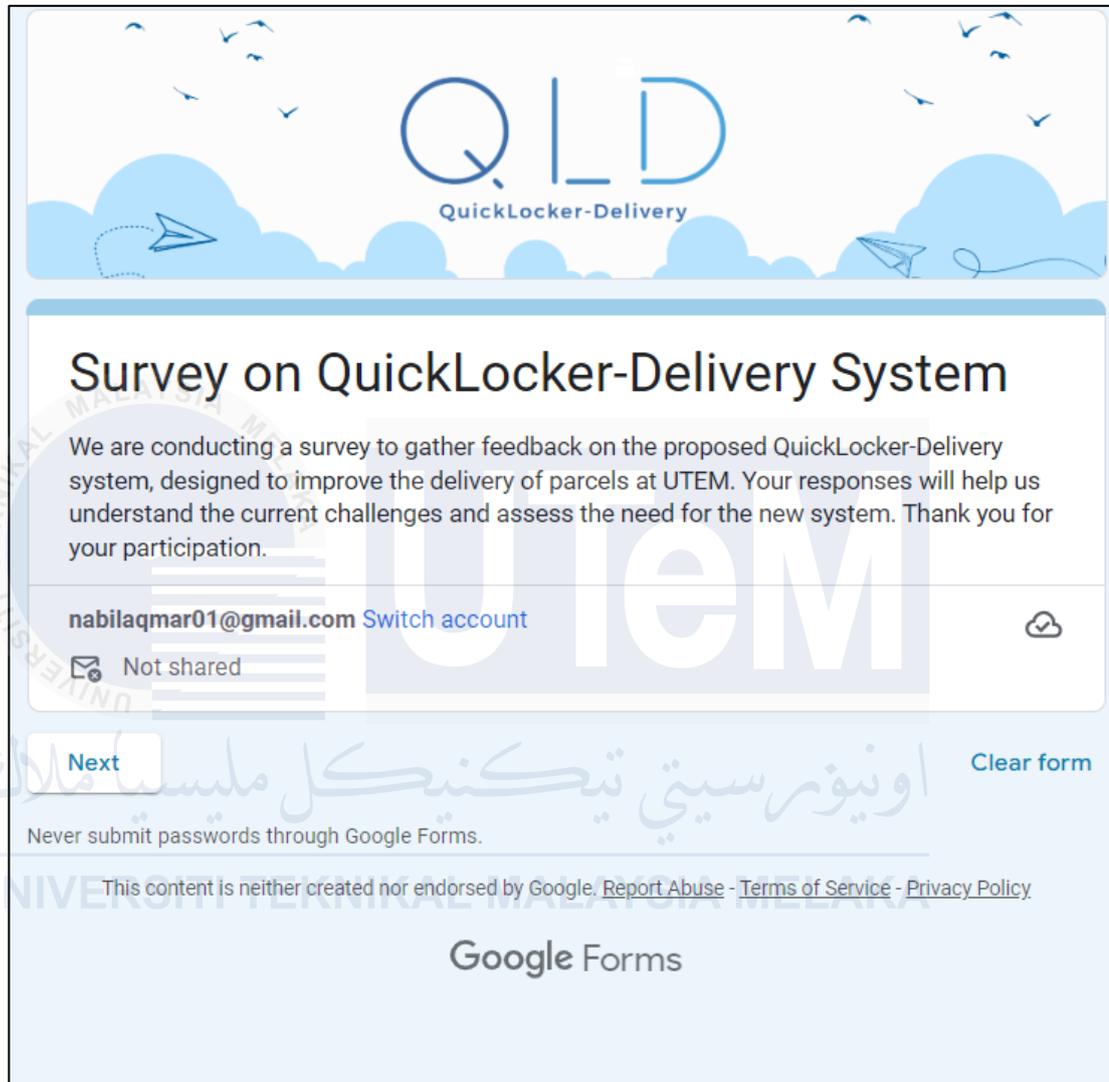


UTeM

اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Appendix B: Questionnaire on Issues in College Courier Service Delivery and Opinions on the QLD Project, and Results Data



The image shows a screenshot of a Google Forms survey. At the top, there is a header with the logo 'QLD QuickLocker-Delivery' and a decorative background of blue clouds and paper airplanes. The main title of the survey is 'Survey on QuickLocker-Delivery System'. Below the title, the text reads: 'We are conducting a survey to gather feedback on the proposed QuickLocker-Delivery system, designed to improve the delivery of parcels at UTEM. Your responses will help us understand the current challenges and assess the need for the new system. Thank you for your participation.' The form is associated with the email address 'nabilaqmar01@gmail.com' and has a 'Switch account' link. There is a 'Not shared' status indicator. At the bottom of the form, there are 'Next' and 'Clear form' buttons. A warning message states: 'Never submit passwords through Google Forms.' Below this, there is a disclaimer: 'This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#).' The Google Forms logo is visible at the very bottom.

QLD
QuickLocker-Delivery

Survey on QuickLocker-Delivery System

We are conducting a survey to gather feedback on the proposed QuickLocker-Delivery system, designed to improve the delivery of parcels at UTEM. Your responses will help us understand the current challenges and assess the need for the new system. Thank you for your participation.

nabilaqmar01@gmail.com [Switch account](#)

Not shared

[Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix B

QLD
QuickLocker-Delivery

Survey on QuickLocker-Delivery System

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Demographic Information

Full Name *
Your answer

Email Address *
Your answer

Age *

Under 18

18-24

25-34

35-44

45-54

Gender *

Male

Female

Role at UTEM *

Student

Staff

Administrator

[Back](#) [Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix B: Issues in College Courier Service Delivery Questionnaire and Data

QLD
QuickLocker-Delivery

Survey on QuickLocker-Delivery System

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Current Delivery Process at UTEM

How often do you receive parcels at UTEM? *

Daily
 Weekly
 Monthly
 Rarely
 Never

Rate your satisfaction with the current manual delivery process. *

1 2 3 4 5

Very Dissatisfied Very Satisfied

What challenges have you faced with the current delivery system? *

Delayed deliveries
 Missed deliveries
 Difficulty locating parcels
 Security concerns
 Other: _____

[Back](#) [Next](#) [Clear form](#)

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

Appendix B



Survey on QuickLocker-Delivery System

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Proposed QuickLocker-Delivery System (QLD)

Admin:

University admins effortlessly manage the system through a web-based portal, ensuring seamless operations and real-time monitoring of locker assignments and system settings.

Staff (Courier):

John, a diligent courier, uses the QLD Courier Mobile App to securely log in, assign a locker, and generate a QR code for Sarah's eagerly awaited parcel.

Student (Recipient):

Sarah, notified via her QLD Recipient Mobile App, swiftly scans the QR code at the locker and retrieves her parcel with ease. This process not only enhances security through secure authentication and encrypted communication but also significantly reduces delivery delays.

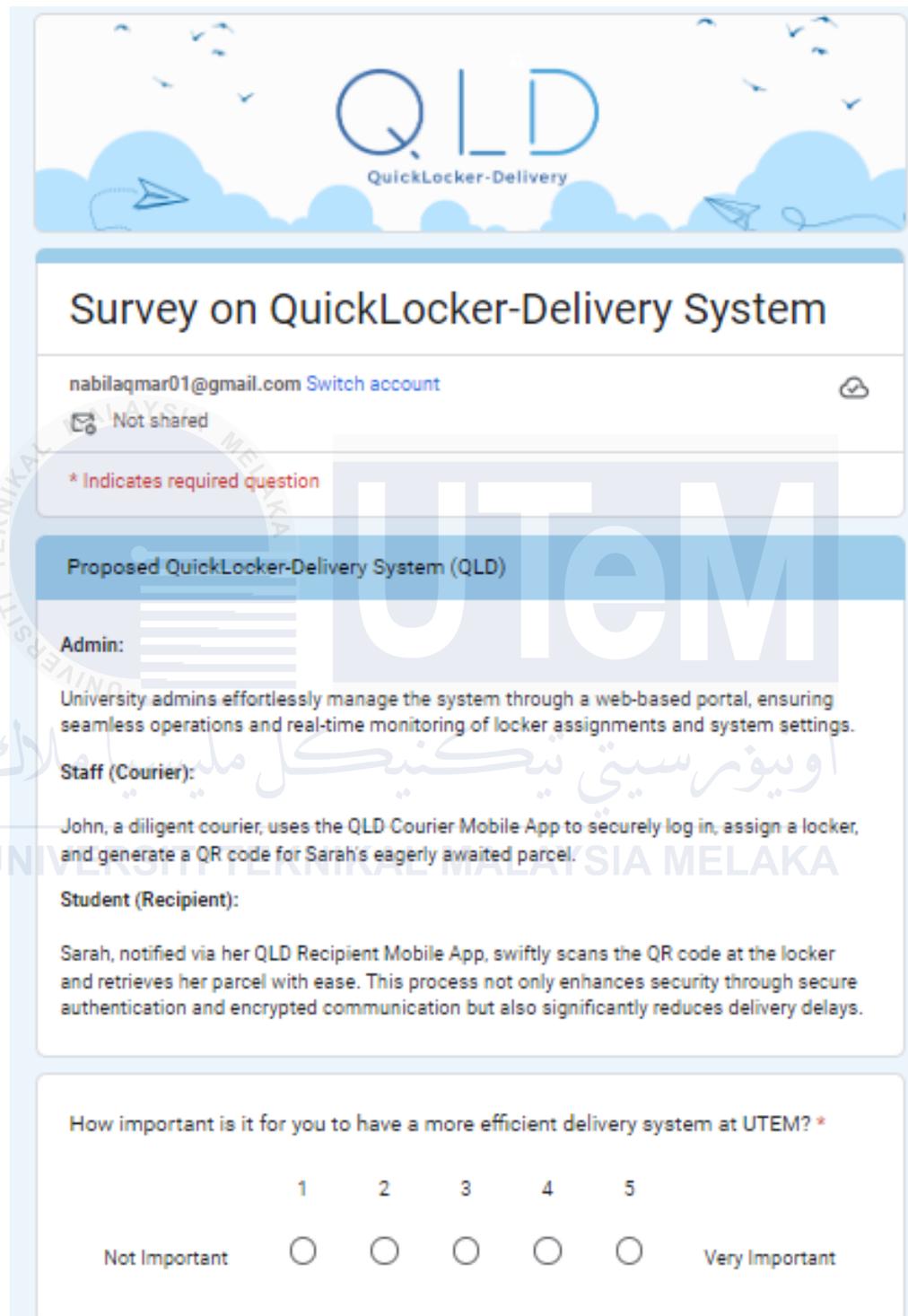
How important is it for you to have a more efficient delivery system at UTEM? *

	1	2	3	4	5	
Not Important	<input type="radio"/>	Very Important				

Which features of the QuickLocker-Delivery system do you find most beneficial? *

- QR code access
- Real-time tracking
- Automated notifications
- Secure lockers
- User-friendly mobile app

Appendix B



QLD
QuickLocker-Delivery

Survey on QuickLocker-Delivery System

nabilaqmar01@gmail.com [Switch account](#)

Not shared

* Indicates required question

Proposed QuickLocker-Delivery System (QLD)

Admin:

University admins effortlessly manage the system through a web-based portal, ensuring seamless operations and real-time monitoring of locker assignments and system settings.

Staff (Courier):

John, a diligent courier, uses the QLD Courier Mobile App to securely log in, assign a locker, and generate a QR code for Sarah's eagerly awaited parcel.

Student (Recipient):

Sarah, notified via her QLD Recipient Mobile App, swiftly scans the QR code at the locker and retrieves her parcel with ease. This process not only enhances security through secure authentication and encrypted communication but also significantly reduces delivery delays.

How important is it for you to have a more efficient delivery system at UTEM? *

1 2 3 4 5

Not Important Very Important

Appendix B

How likely are you to use the QuickLocker-Delivery system if implemented? *

1 2 3 4 5

Very Unlikely Very Likely

Do you think the QuickLocker-Delivery system will improve the delivery process at *
UTEM?

Yes
 No

Additional Comments

Your answer

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Google Forms

Appendix B

31 responses [Link to Sheets](#) ⋮

Accepting responses

Summary Question Individual

Demographic Information

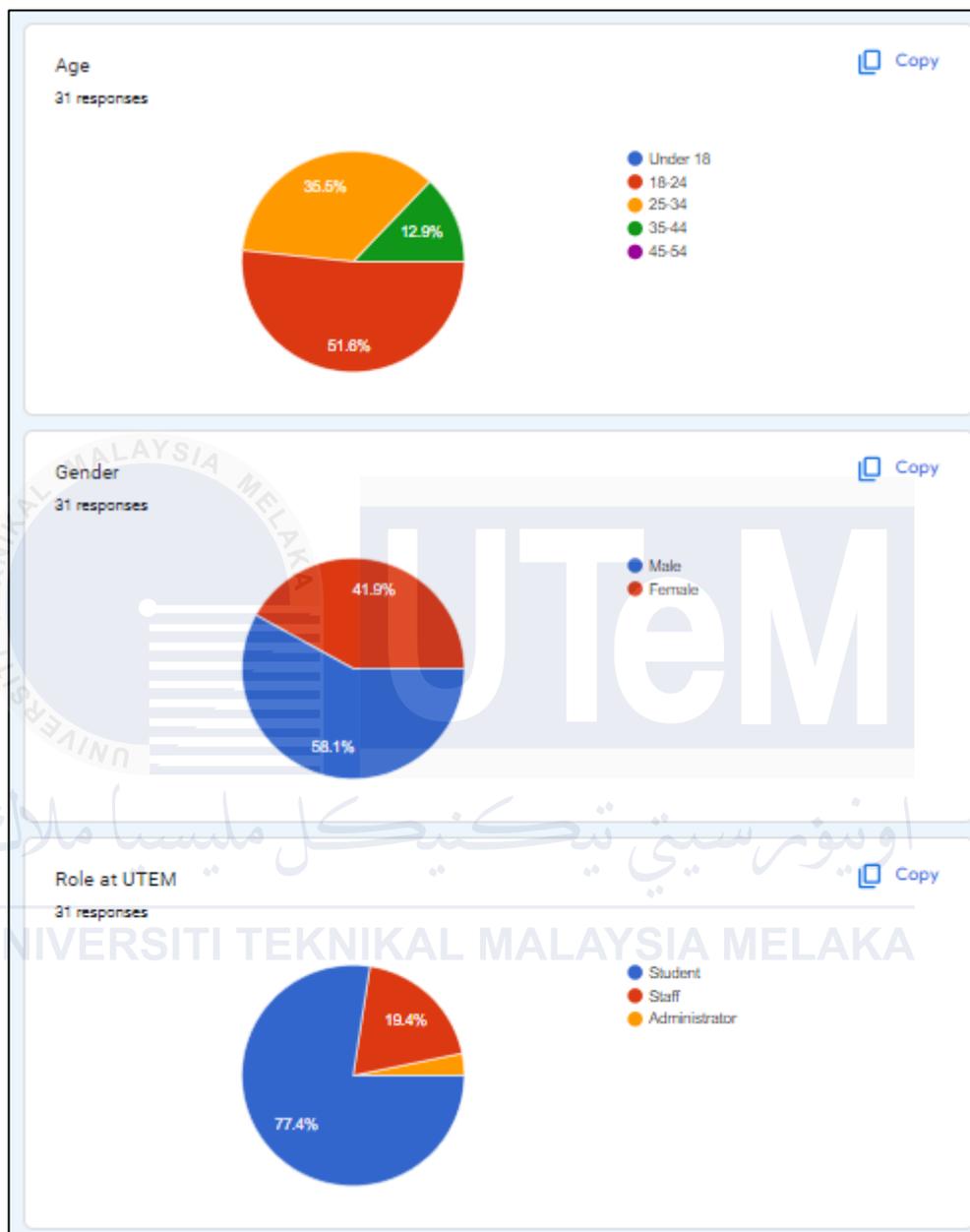
Full Name
31 responses

Taqi bin Abyad
Jane Rusli
Nor Mastura binti Farhan Midali
Rachel Yow Kat Gui
Janaky Pillai a/p Mahathir Ganesan
Noor Syazryana Najmi
Mohd Nik Aminuddin bin Syed Miskan
Nadarajan Thevrandran
Hee Siau Choo

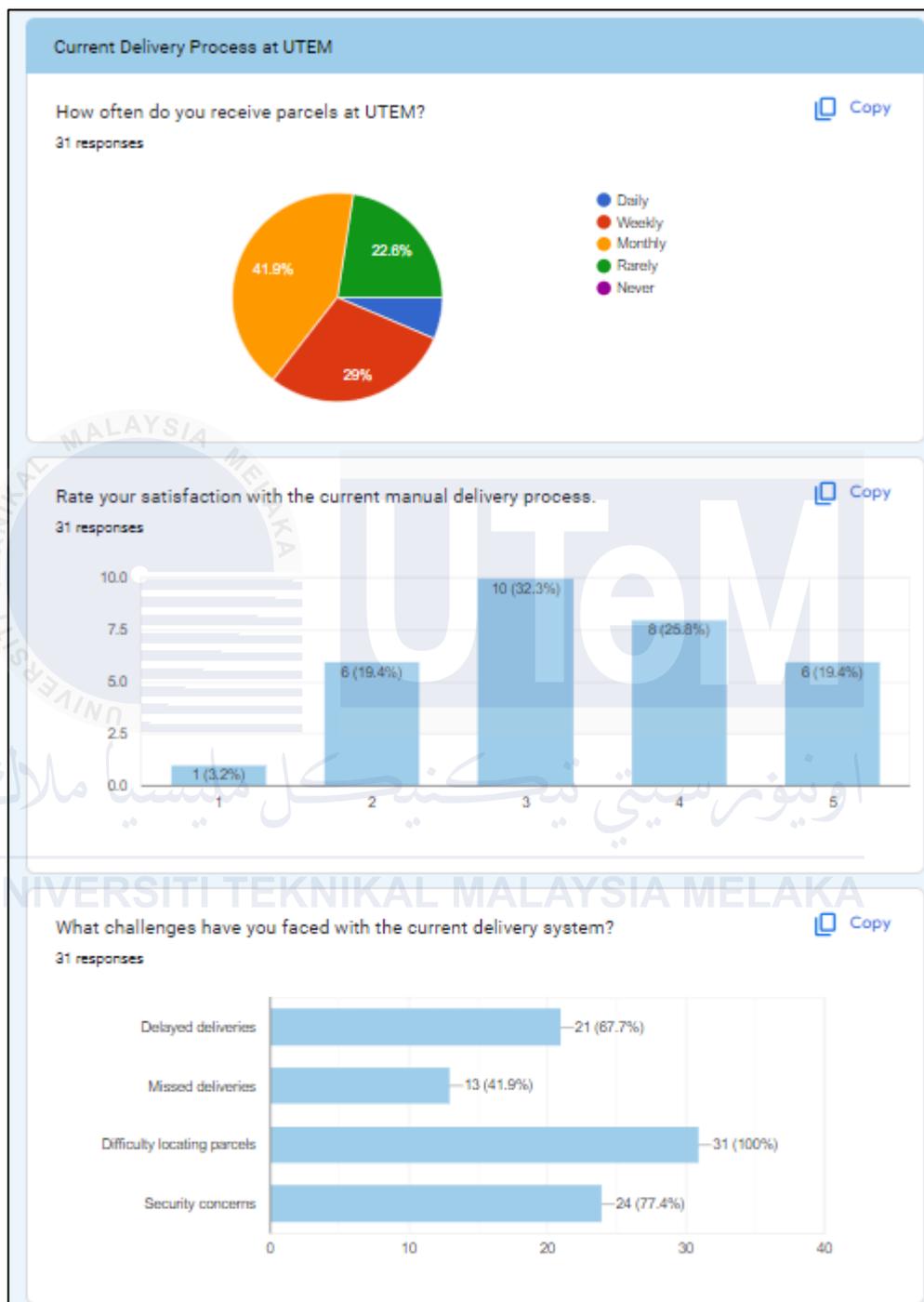
Email Address
31 responses

nabilaqmar01@gmail.com
Azhan@gmail.com
damiahmadazhar@gmail.com
ahmadnazwan73@gmail.com
B032210438@student.utm.edu.my
Muhaajir@gmail.com
Waatiq@gmail.com
Shaamil@gmail.com
Kaarim@gmail.com

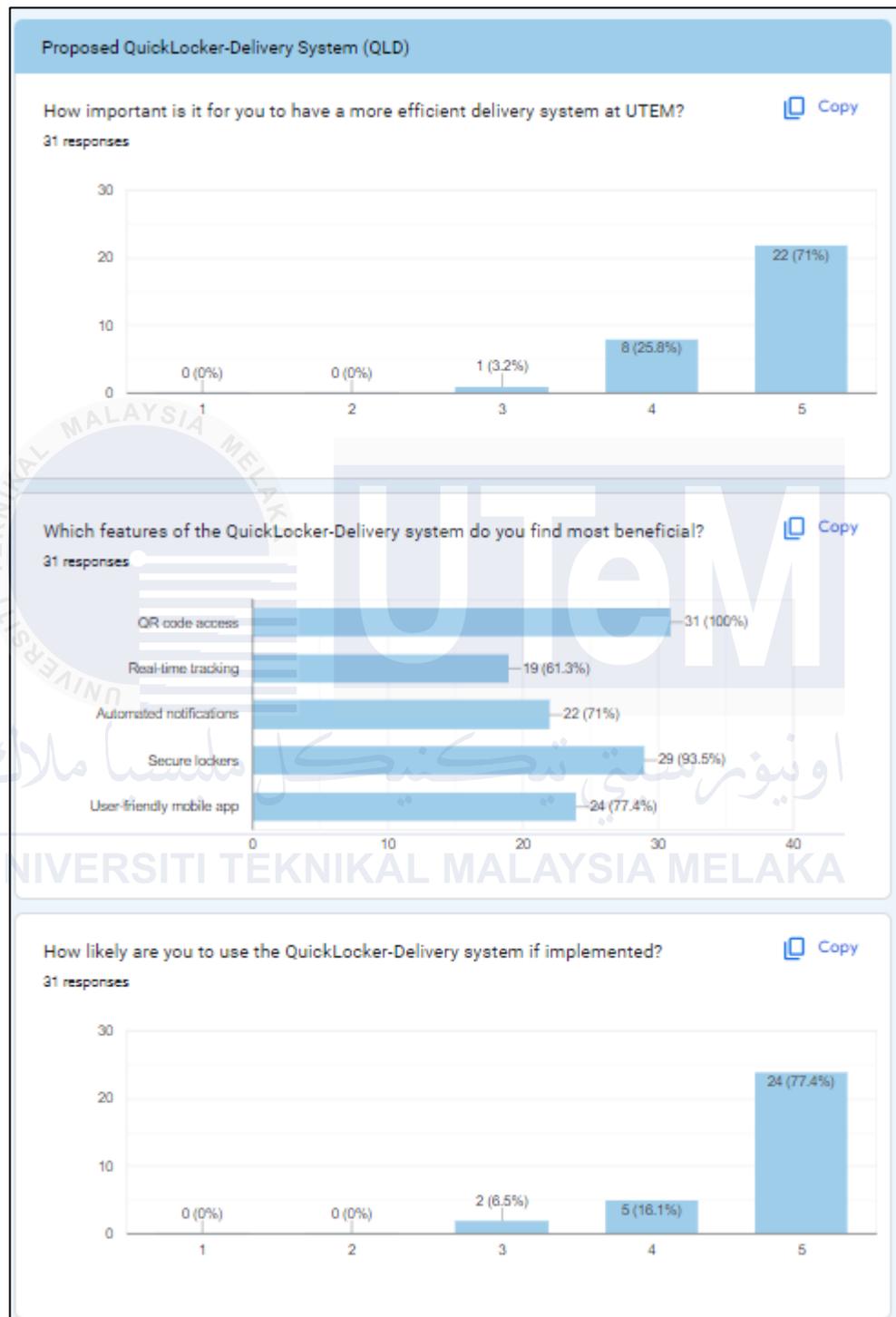
Appendix B



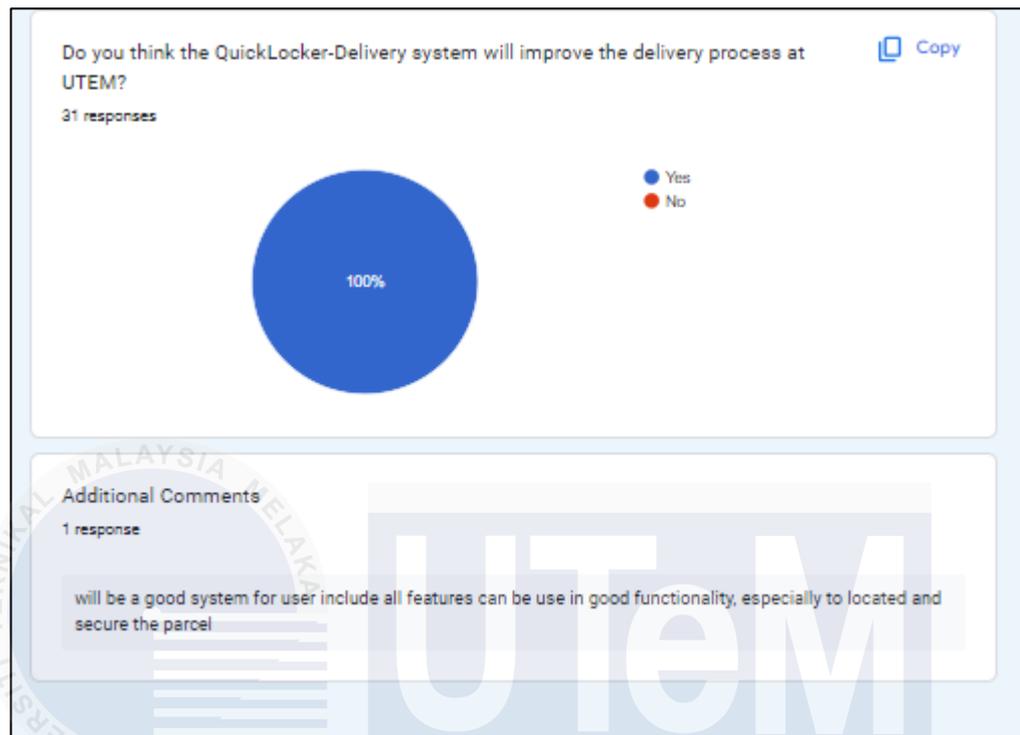
Appendix B



Appendix B



Appendix B



Appendix C: User Guideline

1. Overview

QLD is an innovative locker system designed to streamline parcel delivery and retrieval. This guide provides instructions for recipients, couriers, and administrators to effectively use the system.

2. For Recipients (Customers)

2.1 Registering for QLD

1. Download the App: Download the QLD app from the Google Play Store.
2. Create an Account: Open the app and sign up using your email and personal details.
3. Obtain QLD_ID: After registration, you will receive a unique QLD_ID, which you will use for all parcel deliveries.

2.2 Using Your QLD_ID

1. Shopping Online: When purchasing items online (e.g., on Shopee), enter your delivery address and include your QLD_ID in the address details.
 - Example: John Doe, QLD_ID: R0004, Locker Location: Campus Locker 1
2. Parcel Delivery: The seller will ship your parcel to the specified locker location.

2.3 Parcel Retrieval

1. Notification: You will receive a notification via the QLD app when your parcel is delivered to the locker.
2. Generate QR Code: Open the app, navigate to your parcel details, and generate a QR code for retrieval.
3. Retrieve Parcel: Go to the locker location, scan the QR code at the locker to open it, and collect your parcel.

3. For Couriers

3.1 Registering as a Courier

1. Visit Administration Office: Go to the administration office to apply for a courier position.
2. Provide Required Information: Fill out the necessary forms and provide any required identification or documentation.
3. Receive Credentials: Once approved, you will receive login credentials for the QLD Courier app.

3.2 Using the Courier App

1. Download the App: Download the QLD Courier app from the Google Play Store.
2. Log In: Use the credentials provided by the administration office to log in.

3.3 Delivering Parcels

1. Receive Parcel Details: Obtain parcel details and the recipient's QLD_ID from the drop-off department.

2. **Generate QR Code:** In the app, input the parcel details and generate a QR code for locker assignment.
3. **Scan QR Code:** At the locker location, use the QR scanner on the locker to open an available locker.
4. **Place Parcel in Locker:** Place the parcel inside the locker and close it.
5. **Notify Recipient:** The system will automatically notify the recipient that their parcel is ready for pickup.

4. For Administrators

4.1 Accessing the Web-Based Administration Portal

1. **Log In:** Access the administration portal via your web browser using secure credentials.
2. **Manage Lockers:** Assign lockers to locations, monitor locker status, and manage occupancy.
3. **Courier Accounts:** Create and manage courier accounts, providing them with the necessary credentials.
4. **System Settings:** Adjust system settings, including security protocols and notification preferences.

4.2 Managing Deliveries

1. **Register Incoming Parcels:** Enter details of incoming parcels and assign them to courier staff.
2. **Monitor Deliveries:** Track the status of parcels and ensure timely delivery to the lockers.

3. Analyze Reports: Generate and review real-time occupancy overviews and analysis reports to optimize the system.

This user guide aims to help all users navigate the QLD system effortlessly, ensuring a smooth and efficient parcel delivery and retrieval process.

