UTEM XPRESS

MUHAMMAD AKMAL BIN MOHD SABRI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UTEM XPRESS

MUHAMMAD AKMAL BIN MOHD SABRI

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

2021

DECLARATION

I hereby declare that this project report entitled

UTEM XPRESS

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : MUHAMMAD AKMAL BIN MOHD SABRI Date : 02/09/2021

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 : TS. MASHANUM BINTI OSMAN Date : 02/09/2021

DEDICATION

I would like to thank my supervisor, Ts. Mashanum Binti Osman, for the patient guidance encouragement and advice she has provided throughout my time as her student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. I would also like to thank all the members of faculty at Universiti Teknikal Malaysia Melaka who helped me in this project development. I must express my gratitude to my family members, for their continued support and encouragement. Also, to my all-fellow friends. I was continually amazed by their willingness to proofread countless pages of reports, who experienced all the ups and downs of my project. Completing this report would have been more difficult were it not for the support and friendship provided by the other members. I am indebted to them for their help. Finally, I would like to thank the Faculty of Information Technology and Communication, for giving me the opportunity to do this precious project.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Ts. Mashanum Binti Osman, my supervisor for supporting me from the beginning of this project until its completion. She gave me alternatives to constraints that came. She also believes that I can finish this project successfully. I would also like to thank to both of my parents for understanding my situations and giving ideas to build this system. They are also the reason why I keep moving and pushing myself so that my project is not just a regular system. Finally, thank you to my friends and acquaintances that guided me in programming and documentation. I have learned many things from their helps.

ABSTRACT

UTeM Xpress is an alternative to make it easier for the UTeM community to send and buy goods effortlessly. A common problem among students is the difficulty of buying or sending their goods to a place because they do not have their own vehicle. A special system that targets students and university staff as their main users in realizing this idea. The idea is to make students and university staff as their main customers for those who have business handling their goods. Students are also the reason for the emergence of this idea to help them generate side income. The solution is to allow users create orders through the website through their personal devices. They also need to include the address of the sender and recipient of the goods as well as the additional services required. Students who are registered as shipping partners are responsible for handling the order as specified. They earn revenue from the price of the order that has been made and paid by the user. Students need to pick up the goods at the sender’s address then proceed with delivery to the recipient’s address. Order fees can be collected either at the sender's or recipient's address. We expect that these ideas and solutions will be able to have a positive impact on the university community in managing their day-to-day affairs.

ABSTRAK

UTeM Xpress adalah alternatif untuk memudahkan komuniti UTeM menghantar dan membeli barang dengan mudah. Masalah yang sering berlaku di kalangan pelajar adalah kesukaran membeli atau menghantar barang ke tempat kerana mereka tidak memiliki kenderaan sendiri. Sistem khas yang mensasarkan pelajar dan kakitangan universiti sebagai pengguna utama mereka dalam merealisasikan idea ini. Ideanya adalah untuk menjadikan pelajar dan kakitangan universiti sebagai pelanggan utama bagi mereka yang mempunyai perniagaan mengendalikan barang mereka. Pelajar juga menjadi sebab timbulnya idea ini untuk membantu mereka menjana pendapatan sampingan. Penyelesaiannya adalah dengan membolehkan pengguna membuat pesanan melalui laman web melalui peranti peribadi mereka. Mereka juga perlu menyertakan alamat pengirim dan penerima barang serta perkhidmatan tambahan yang diperlukan. Pelajar yang didaftarkan sebagai rakan kiriman bertanggungjawab mengendalikan pesanan seperti yang ditentukan. Mereka memperoleh pendapatan dari harga pesanan yang telah dibuat dan dibayar oleh pengguna. Pelajar perlu mengambil barang di alamat pengirim kemudian meneruskan penghantaran ke alamat penerima. Bayaran pesanan dapat dikumpulkan sama ada di alamat pengirim atau penerima. Kami menjangkakan bahawa idea dan penyelesaian ini akan dapat memberi kesan positif kepada komuniti universiti dalam menguruskan urusan harian mereka.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
DFD	**-**	**Data Flow Diagram**
URL	**-**	**Uniform Resource Locators**
PWA	**-**	**Progressive Web Application**

LIST OF ATTACHMENTS

CHAPTER 1: INTRODUCTION

1.1 Introduction

The title of this proposed project is UTeM Xpress which is a web-based application system that facilitates the university community to manage their order deliveries on daily basis. The background of selected area for this project is logistic. The idea to do this project was inspired by an actual application that a company has run and operates just around the Klang Valley. The company created an application to make it easier for residents to manage the delivery or purchase of their goods on the same day and it is limited to the Klang Valley area only. I expect that if this idea can be realized in the university area, there would be a lot of benefits to be gained and used together in the university community. This system has three types of target users, namely users, runners, and admins. They are the main users of this system while only students can register as runners to fulfill orders that have been placed by users. One of the main purposes of developing this system is to make it easier for users to deliver and buy goods even if they do not have their own vehicles and also able to generate side income for students. This system allows users to place orders online on the website that has been provided. They need to fill in the addresses of the sender and recipient as well as the additional services they need. While students who are registered as runners are responsible for delivering orders that have been made by users. The runners must pick up the goods at the pickup address then forward to the drop-off address and payment for the order can be made by the users either at the pickup or drop-off address. With this initiative, students can also generate their side income within the university grounds.

1.2 Problem Statement(s)

- Students find it difficult to send and buy goods from outside the university because they do not have their own vehicles.
- The management of the residential college does not allow year 1 and 2 students to bring private vehicles.
- Constraints of students to generate side income in the university area.

1.3 Objective

- To provide delivery services outside and within the university grounds.
- To provide income-generating opportunities to university students.
- To propose a platform that allows the purchase of goods and delivery of goods for those who do not own a vehicle.

Scope

Target User: UTeM's Staff & Student

Platform: Web-based

Modules: Student & Staff:

1. Register as a (customer) & Login

2. Create order

3. View order record

4. Rate and review runner

Modules: Only Student:

1. Register as a (runner) & Login

2. Accept order

3. Deliver order

4. View past order record

5. Tracks earning

Modules: Only Staff:

1. Register as a (admin) & Login

2. Manage User

3. Manage Runner

4. Manage record

5. Manage Admin

6. View system statistic data

1.4 Project Significance

- Why need to build this system?

The needs to build this system was to allow staffs and students to manage their purchasing and sending goods inside and outside the universities area. With this system, students are no longer having to spend more on cost, energy, and time just to manage their goods.

- Benefits to be earned.

The university communities can easily manage their goods even do not own vehicle and let the runner help them to deliver their item to the desired location. Students also can generate side income as they were able to collect the order fee from the users.

- Who will benefit from this and how?

This system was developed specifically for university communities. This will benefit staff and student by allowing them to manage their daily orders and let the shipping partner deliver their order while helping to support the student to collect order fee as their daily wages.

1.5 Expected Output

My expectation from the development of this project can help facilitate the delivery of goods to university residents who have time and vehicle constraints. Apart from that, it is hoped that university students use this service fully and carefully to facilitate their daily affairs as well as generate side income for them. With this system, students no longer must worry about sending goods inside and outside the university area because this system has provided the best platform to meet the needs of all UTeM communities. The solution for this project is to provide web applications that almost the entire community has their own personal devices to facilitate the process of ordering and delivering goods.

1.6 Conclusion

Hopefully, these ideas and system that are going to be developed will brings a lot of benefits and positive impact towards university communities in managing their goods. We do hope, this system able to fulfill the request from both parties (staff and student) without having some serious issues in future. In the next chapter, we will provide a literature review and project methodology used for this project development.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

The facts and findings will be discussed in depth in this chapter of the Literature Review to reveal the necessary concepts, theories, and knowledges about the system and system domain. The analysis of existing products and systems that provide similar functionalities is discussed in the following section. Comparisons between existing products and this project form the basis of the analysis. Following that, the chosen methodology will be introduced, along with all the activities associated with it. Following that, in section 2.4, you will find a list of all the software and hardware requirements. The available existing system figures are provided in figures 2.1, 2.2, 2.3. The project methodology diagram is provided in figure 2.4 and project milestones in figure 2.5. The project schedule will be illustrated, and milestones will be listed in the final section.

2.2 Facts and Finding

The domains related to this project are web application, application of information technology in logistics and deliveries. The explanation is included in the following sections.

2.2.1 Domain

The domain related to this project development is Logistic. A good example of IT driven competitiveness can be found in the logistics industry. Logistics information systems are used in every big company. This system helps companies improve operational efficiency by tracking resources from when they are first obtained, such as raw materials, to their point of consumption. Companies leverage these systems to gain end-to-end visibility of their products or raw materials. Logistics information systems also help companies track internal information within a company by providing reports on inventory costs that determine how much more inventory needs to be purchased (Ballou, 2013). This feature allows companies to be more efficient by providing optimal lot sizes and lead times.

2.2.2 Existing System

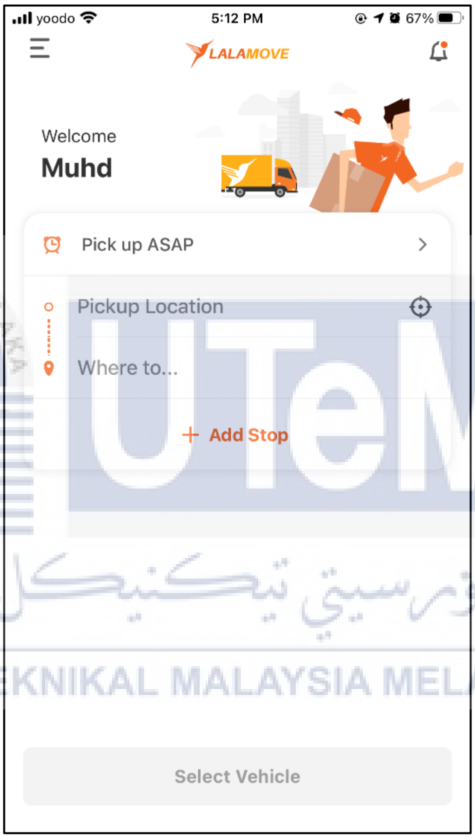
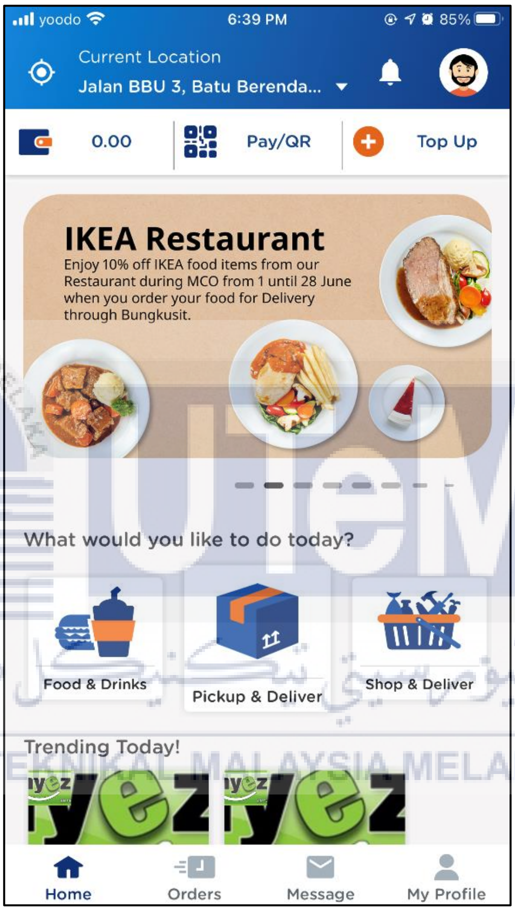


Figure 2.1: Lalamove Application (https://www.lalamove.com/en-my/, 2021)

The first current system that we are referring is Lalamove application which is also same-day delivery application that only operates in Klang valley. The first screen interface displayed were the location interface that requires user to input the pickup and drop-off location before proceeding with vehicle selection. Then, the users are required to choose their desired mode of transportation to deliver their order. Each vehicle contains different additional services for user to choose if they need it. After that, the user will be displayed more details page to get the extra details regarding the order from user. For example, the system display textbox for the user to enter the order remarks. The system also able to display order records that has been made by the user and categorize it based on its status such as on-going, completed, and cancelled. The system enable user to add runner to their favorite list and prioritize the order just for the favorite runner. The system able to display information like profile, runner profile, and order details.



Enjoy 10% off IKEA food items from our Restaurant during MCO from 1 until 28 June when you order your food for Delivery through Bungkusit.

What would you like to do today?

Figure 2.2: Bungkusit Application (https://bungkusit.com.my/, 2021)

The next current system that we are referring is Bungkusit application which is also same-day delivery application that can operates in and outside from Klang valley. The first screen interface displayed were the home interface displayed and contain a lot of features that allow user to explore it. This app categorizes type of order into three type which is Food & Drinks, Pickup & Deliver, Shop & Deliver. After the user has determine the type of orders that he/she would like to choose then the app will display the vehicle selection interface. The user needs to select the vehicle type, set the pickup and delivery location, preferred time of delivery either for now or later, and upload optional photo to be reference by the runners. The application also can view past order record made by the users, message with runners, profile details, and order details.

Table 2.1: Comparison Existing System

Vendor	**Lalamove**	**Bungkusit**
Commission percentage taken for each order	20.5%	15%
Popularity	Well-know	Still ongoing
User interface	Easy to understand	Hard to understand
Area covered	Klang Valley only	All over Malaysia.
Order type	• On-demand • Scheduled for later	• On-demand • Scheduled for later
Order category	• General	• Food & Drinks • Pickup & Deliver • Shop & Deliver
Payment method	Online payment & Cash & E-wallet	Cash & Credit Points
Transport mode	• Motorcycle, • Car • 4 x 4 Pickup • 7-ft Van • 9-ft Van • 1-ton lorry • 3-ton lorry	• Motorcycle • Car • Pickup • Van • 1-ton lorry • 3-ton lorry

Motorcycle calculation	**Load Limit:** 2-10 kg **Size Limit (L x W x H):** 36 cm x 36 cm x 36 cm **Fare:** 0-5 km: RM 5.00 5-10 km: RM 1.00/km >10 km: RM 0.80/km **Additional Stops:** RM 1.00/ each stop **Return Trip:** RM 4.00 **Buy4U (max: RM100):** RM 4.00	**Base Fare:** **Parcel 0-5km:** RM5.00 **Food Purchase 0-5km:** RM7.00 **Order From Anywhere 0-5km:** Rm7.00 **Grocery Purchase 0-5km:** RM10.00 **Size Limit (L × W × H):** 50 cm × 35 cm × 25 cm **Weight Limit:** 10kg **Additional 1km Fare:** **Parcel:** RM0.90 **Food Purchase:** RM1.00 **Order from anywhere:** RM1.00 **Grocery Purchase:** RM1.00
Car calculation	**Load Limit:** 40 kg **Size Limit (L x W x H):** 50 cm x 50 cm x 50 cm **Fare:** 0-5 km: RM 8.00 6-15 km: RM 1.00/km >16 km: RM 1.50/km **Additional Stops:** RM 2.00/ each stop **Return Trip:** RM 4.00	**Base Fare:** **Parcel 0-5km:** RM8.00 **Order From Anywhere 0-5km:** Rm15.00 **Grocery Purchase 0-5km:** RM17.00 **Size Limit (L × W × H):** 50 cm × 50 cm × 50 cm **Weight Limit:** 40kg

	Buy4U (max: RM100): RM 4.00 **Door-to-Door:** RM 5.00	<u>**Additional 1km Fare:**</u> **Parcel:** RM1.30 **Order from anywhere:** RM1.10 **Grocery Purchase:** RM1.00

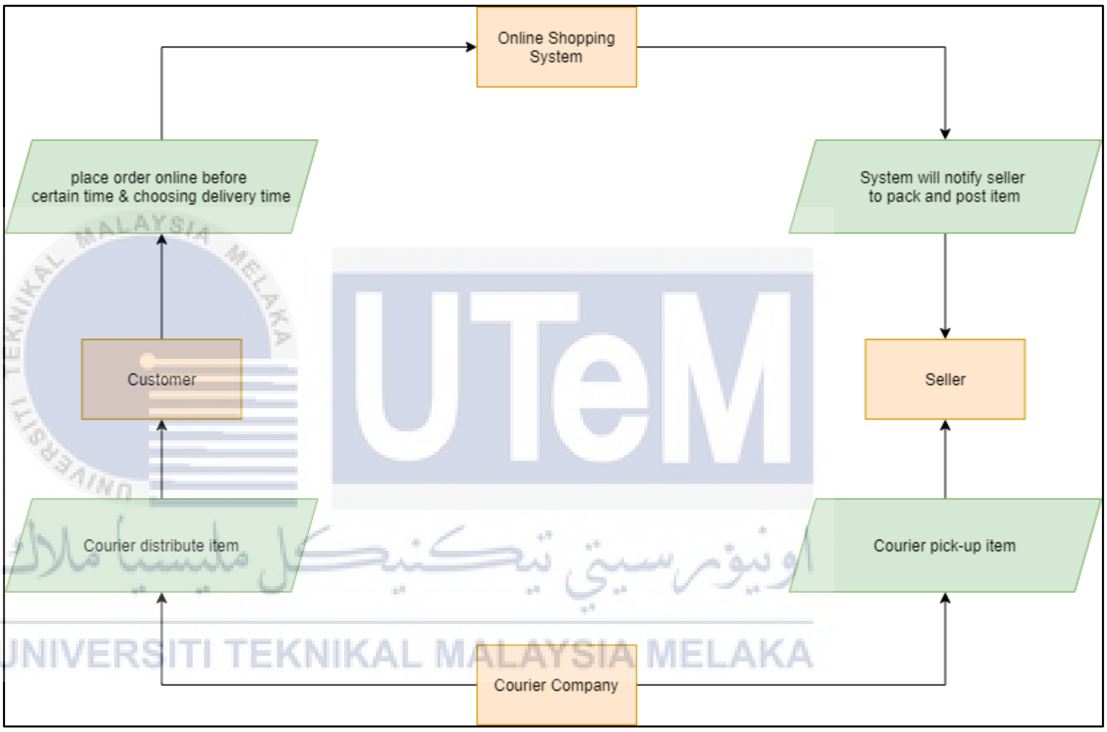

Online Shopping System
place order online before certain time & choosing delivery time
System will notify seller to pack and post item
Customer
Seller
Courier distribute item
Courier pick-up item
Courier Company

Figure 2.3: Existing System Diagram

For current system operating method is explained in the figure 1 below. Nowadays the people need to browse the e-commerce website to purchase item and wait for a couple days to receive the ordered item. There is 3 type of delivery time, the first one is deferred delivery which take time in 3-5 days and this use case is when the customer want to buy something and do not mind waiting for item to received. Second is next-day delivery which take time in 1 day to received and this use case is when the customer wants to purchase the item and want to receive the item as quickly as possible otherwise, they will just buy it in physical store and the last type of delivery is same-day delivery which usually took 1-12 hours to received. This use case is when the customer is in urgent situation and do not have time to check out other stores but need it today.

2.2.3 Technique

This section will define several types of technique that have been initiated to gather wide and necessary information on this project development which including:

1. Documentation review

   Analysis process is carried out by reviewing the existing system documentation. This process is to collect existing function and features so we can add something different for this project development. We also can learn something new from existing system while doing this technique.

2. Observation

   This technique uses to watch the program in operation to gather information about what happens day-to-day.

3. Case studies

   Used to depict experiences, processes, or practices with a view to developing understanding through examination and cross comparisons.

2.3 Project Methodology

The methodology for software development has a great variety of approaches. We call it as models of system development life cycle (SDLC). The point where the model or framework that will guide the development process is chosen is a central strategic aspect. The wrong life cycle can delay the project and affect client satisfaction. For this project, we use the Spiral Model. The Spiral Model is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process. Each phase of spiral model in software engineering begins with a design goal and ends with the client reviewing the progress. The spiral model in software engineering was first mentioned by Barry Boehm in his 1986 paper. The development process in Spiral model in SDLC, starts with a small set of requirements and goes through each development phase for those set of requirements. The software engineering team adds functionality for the additional requirement in every-increasing spirals until the application is ready for the production phase. The below figure very well explains Spiral Model:

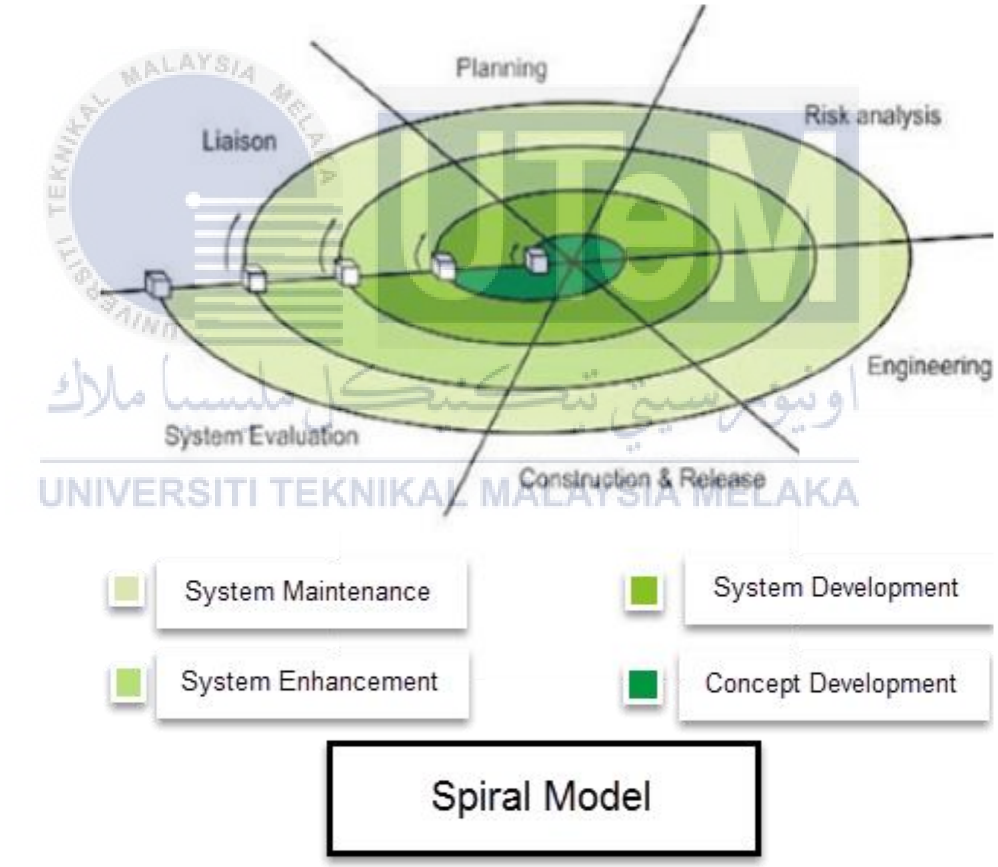


Figure 2.4: Spiral Model Lifecycle

Planning: The first phase includes estimating the cost, schedule, and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer.

Risk Analysis : Identification of potential risk is done while risk mitigation strategy is planned and finalized.

Engineering: It includes testing, coding, and deploying software at the customer site

Evaluation: Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun.

2.4 Project Requirements

This section lists out the software and hardware requirements related to the development of this project.

2.4.1 Software Requirement

- PHPStorm as a text editor for code
- Xampp as a web and database server
- GitHub as a code hosting platform for version control and collaboration

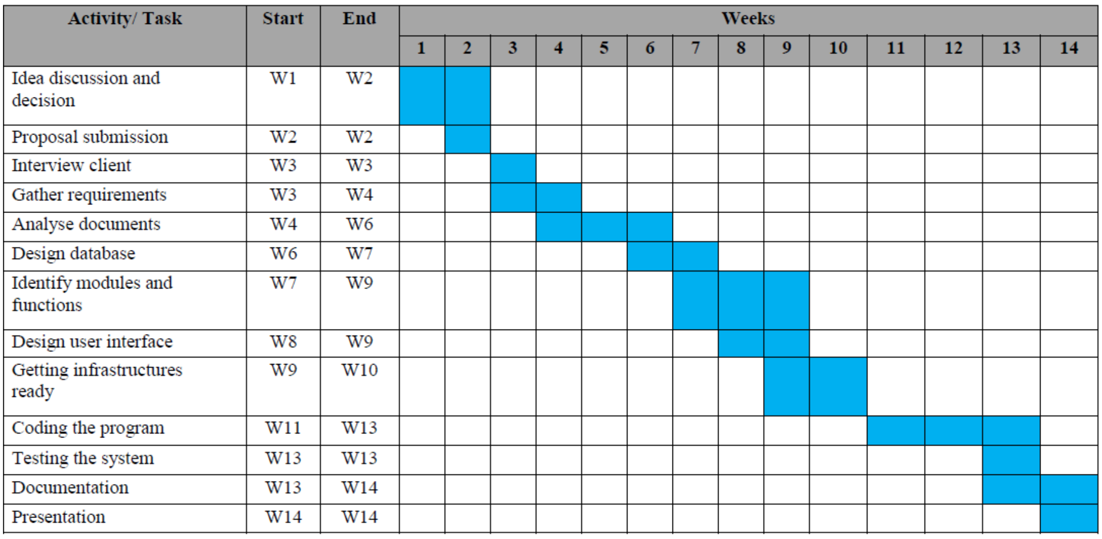
2.4.2 Hardware Requirement

- Laptop

2.4.3 Other Requirement

- None

2.5 Project Schedule and Milestones



Activity/ Task	Start	End	Weeks													
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
Idea discussion and decision	W1	W2	■	■												
Proposal submission	W2	W2		■												
Interview client	W3	W3			■											
Gather requirements	W3	W4			■	■										
Analyse documents	W4	W6				■	■	■								
Design database	W6	W7						■	■							
Identify modules and functions	W7	W9							■	■	■					
Design user interface	W8	W9								■	■					
Getting infrastructures ready	W9	W10									■	■				
Coding the program	W11	W13											■	■	■	
Testing the system	W13	W13													■	
Documentation	W13	W14													■	■
Presentation	W14	W14														■

Figure 2.5: Project Milestones

2.6 Conclusion

To develop a quality software, we must clear what we need to develop it before starting it. We need to prepare all the requirement first then setup a plan to arrange our development followed by each phase that we have identified based on type of software development life cycle that we have chosen for instance this project use Spiral model. This will help us manage the effort that need to be put more in a certain crucial phase and will prevent us from being exceeding the development timeframe that has been provided. A proper planning will reduce error or mistake made by the developer during development phase and providing extra time to correction for any bugs found.

CHAPTER 3: ANALYSIS

3.1 Introduction

In the third chapter, the analysis phase is to discuss the collected and processed information from existing system. It is the initial outline of the software is about to be developed. The existing system and the proposed system are being examined in this chapter. The weakness of existing of the system are observed to ensure the improvement can be made for the proposed system. In this phase, it will be divided into two stages which are the analysis of the existing system and the proposed system to be developed. Figure 3.1 shows the existing system flow, figure 3.2 display level 0 data flow diagram, and figure 3.3 display level 1 data flow diagram. All the table are included in section 3.3.1.1 data dictionary, 3.3.2 functional requirements and 3.3.3 non-functional requirements.

3.2 Problem Analysis

A problem analysis was carried out prior to designing architecture process to investigate the issue or failure that may arise from the current system implementation. In the existing system implementation, universities are prohibiting students that are from year 1 and 2 from using their own personal vehicles in campus area and while only year 3 student can use their own personal vehicles to go inside and outside campus area. This implementation give rise to some misunderstanding from the students itself because the year 3 student is not allowed to stay in the campus area hence, they are allowed to use personal vehicle but then the question arise is who are going to use the parking space provided in the campus aside from staff. The student argues that those students living on campus are also eligible to use the parking space provided.

Due to the rules set by the management, we find out that the student was having difficulties to send or buy goods from the desired places as they do not have personal vehicles. The existing system provides bus transportation to go to local supermarkets to buy their necessities, but it is quite limited to the movement of students in terms of time and distance. They need to obey the time and location that has been set by the management and route for the bus. The student also allowed to use private car sharing application like Grab Car to go to their desired place, but this initiative does cost for the transportation fee towards student.

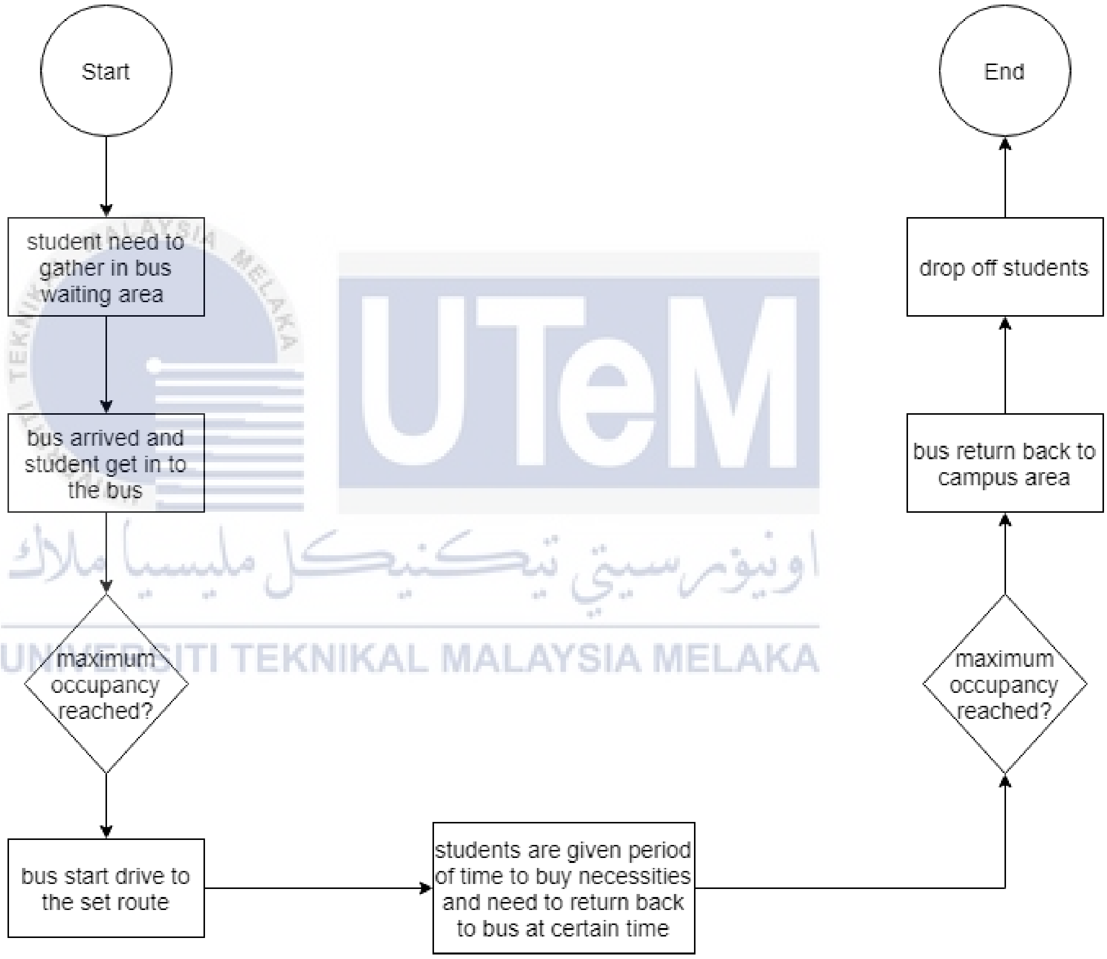

Start
student need to gather in bus waiting area
bus arrived and student get in to the bus
maximum occupancy reached?
bus start drive to the set route
students are given period of time to buy necessities and need to return back to bus at certain time
maximum occupancy reached?
bus return back to campus area
drop off students
End

Figure 3.1: Existing System Flow

Figure 3 shows the process for students that use the bus transportation provided by the management to buy their necessities. Students need to wait earlier before the bus arrive at the bus station and get into the bus and wait for the other student to get into the bus. If the bus were full of students, the other student needs to wait for other bus to arrive. After the bus has reach max occupancy, the driver start drive based on the route that has been set by management. Student were given period to buy their needs and require returning to bus at a certain time. If the student were late to return to the bus, then they need to wait for another bus. After the student has return and max occupancy reached then the bus driver will return to campus area to drop off students.

3.3 Requirement Analysis

This section covers elaboration on analysis of data requirements, functional requirements, and non-functional requirements of this UTeM Xpress system.

3.3.1 Data Requirement

After analyzing the process flow of current system. The next important part is to study the data requirement. Data requirements discuss what kind of data should the system input and output and what data should the system store internally. Based on the table 1 below, we list all the data candidate that we require to store in the system database.

3.3.1.1 Data Dictionary

This will section describes the field name, data type, description, required, remarks, default value, primary and foreign key for each column in every table contain in the database.

Table 3.1: Additional Service Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
service_id	int(11)	PK	Service’s Id	Yes	Auto increment
service_name	varchar(255)	PK	Service’s Name	Yes	

Table 3.2: Admins Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
admin_email	varchar(255)	PK	Admin's email	Yes	
admin_nric	varchar(255)		Admin's identification number	Yes	
admin_name	varchar(255)		Admin's full name	Yes	
admin_status	varchar(255)		Admin's status	No	Default: available
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.3: Roles table

Column Name	Data Type	PK/FK	Description	Required	Remarks
role_id	int(11)	PK	Role's id	Yes	Auto increment
role_name	varchar(255)		Role's name	Yes	
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.4: Staffs Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
staff_email	varchar(255)	PK	Staff's email	Yes	
staff_nric	varchar(255)		Staff's identification number	Yes	
staff_name	varchar(255)		Staff's full name	Yes	
staff_designation	varchar(255)		Staff's designation	Yes	

staff_faculty	varchar(255)		Staff's faculty	Yes	
staff_division	varchar(255)		Staff's division	Yes	
staff_status	varchar(255)		Staff's status	Yes	Default: available
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.5: Student Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
student_email	varchar(255)	PK	Student's email	Yes	
student_nric	varchar(255)		Student's identification number	Yes	
student_name	varchar(255)		Student's full name	Yes	
student_faculty	varchar(255)		Student's faculty	Yes	
student_status	varchar(255)		Student's status	Yes	Default: available
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.6: Vehicle Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
vehicle_id	int(11)	PK	Vehicle's id	Yes	Auto increment
vehicle_type	varchar(255)		Vehicle's type	Yes	
vehicle_weight_capacity	varchar(255)		Vehicle's max weight capacity	Yes	
vehicle_size_capacity	varchar(255)		Vehicle's max size capacity	Yes	

Table 3.7: Order Table

column name	data type	PK/FK	Description	Required	Remarks
order_id	int(11)	PK	Order's id	Yes	Auto increment
vehicle_id	int(11)	FK	Vehicle's id	Yes	
id	int(11)	FK	User's id	Yes	
runner_id	int(11)	FK	Runner's id	Yes	
order_pickup_ location	varchar(255)		Pickup's location	Yes	
order_dropoff_ location	varchar(255)		Dropoff's location	Yes	
reciever_name	varchar(255)		Receiver's name	Yes	
receiver_phone	varchar(255)		Receiver's telephone number	Yes	
user_review	varchar(255)		User's review	No	
user_rating	varchar(255)		User's rating	No	
runner_rating	varchar(255)		Runner's rating	No	
order_fee	decimal(7,2)		Order's fee	Yes	
favourite	int(11)		Order's favourite by user	No	
order_remarks	varchar(255)		Order's remark	Yes	
order_status	varchar(255)		Order's status	Yes	Default: available
order_date	date		Order's date	Yes	
order_time	time		Order's time	Yes	
order_type	varchar(255)		Order's type	Yes	
pickup_location_ latitude	varchar(255)		Pickup's location latitude	Yes	
pickup_location_ longitude	varchar(255)		Pickup's location longitude	Yes	
dropoff_location_ latitude	varchar(255)		Dropoff's location latitude	Yes	

dropoff_location_longitude	varchar(255)		Dropoff's location longitude	Yes	

Table 3.8: Order Service Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
order_service_id	int(11)	PK	Order's service id	Yes	Auto increment
order_id	int(11)		Order's id	Yes	
service_id	int(11)		Service's id	Yes	

Table 3.9: Runner Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
runner_id	int(11)	PK	Runner's id	Yes	Auto increment
user_id	int(11)		User' id	Yes	
runner_license_picture_front	varchar(255)		Runner's license image front	No	
runner_license_picture_back	varchar(255)		Runner's license image back	No	
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.10: Runner Vehicle Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
vehicle_id	int(11)	PK	Vehicle's id	Yes	Auto increment
runner_id	int(11)		Runner's id	Yes	
vehicle_type	varchar(255)		Vehicle type	No	
vehicle_picture	varchar(255)		Vehicle image	No	
vehicle_number_plate_picture	varchar(255)		Vehicle number plate image		
vehicle_roadtax_picture	varchar(255)		Vehicle's roadtax image		
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.11: User Roles Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
role_id	int(11)	PK	Role's id	Yes	
user_id	int(11)	PK/FK	User' id	Yes	
staff_email	varchar(255)	FK	Staff's email	No	
student_email	varchar(255)	FK	Student's email	No	
admin_email	varchar(255)	FK	Admin's email	No	
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

Table 3.12: User Table

Column Name	Data Type	PK/FK	Description	Required	Remarks
id	int(11)	PK	User’s id	Yes	
email_verified_at	timestamp		Date of email verification	Yes	
email	varchar(255)		User’s email	No	
password	varchar(255)		User’s password	No	
two_factor_secret	text			No	
two_factor_recovery_codes	text			No	
name	varchar(255)		User’s name	Yes	
user_phone	varchar(255)		User’s phone	Yes	Unique
user_picture	varchar(255)		User’s picture	No	
user_nric_picture front	varchar(255)		User’s identification front image	No	
user_nric_picture_back	varchar(255)		User’s identification back image	No	
user_status	varchar(255)		User’s status		Default: available
remember_token	varchar(100)			No	
created_at	timestamp		Created date	No	
updated_at	timestamp		Updated date	No	

3.3.2 Functional Requirement

Functional requirement describes what the system should do also with its features and functions. Generally, functional requirements show the system behavior under certain conditions. Functional requirements are important for the development of a software product. Table below shows the functional requirements that are implemented into this system development. Each functional requirement state comes with its description.

Table 3.13: Functional Requirement Table

No.	Module	Description
1.	User management	This system shall support basic user profile function such as register, login & logout and update profile.
2.	Order management	This system shall enable system users to create order and set order details based on desired pickup and drop off location, mode of transportation, additional services, remarks, receiver name and phone number, date, and time of the order.
3.	Job management	This system also shall enable runner to view available order based on their registered type of vehicle and sort order based on the order type.
4.	Rate and Review management	Upon completion of order and status has been changed to completed, the system must display a form of review and rating towards order experience by the user and runner.
5.	Vehicle Management	The system must prompt runner to set up their runner profile and vehicles information details before start taking

		customer's orders. The required information including front and back image of driving license, vehicle type, vehicle image, plate number image and road tax image.
6.	Report Management	This system shall be able to generate reports based on the data stored in the database. Also, able to generate statistic information for example number of users, runners, orders.
7.	Runner management	This system shall support basic user profile function such as register, login & logout and update profile for the runner.
8	Admin management	This system shall enable admins to oversee the entire system's current state and also provide privileges set of action to manage the system.

3.3.2.1 Data Flow Diagram

This section describes the scope and boundaries of a system with Data Flow Diagrams (DFD). It contains context diagram, Level 0, Level 1 of DFD.

(a) Context Diagram DFD

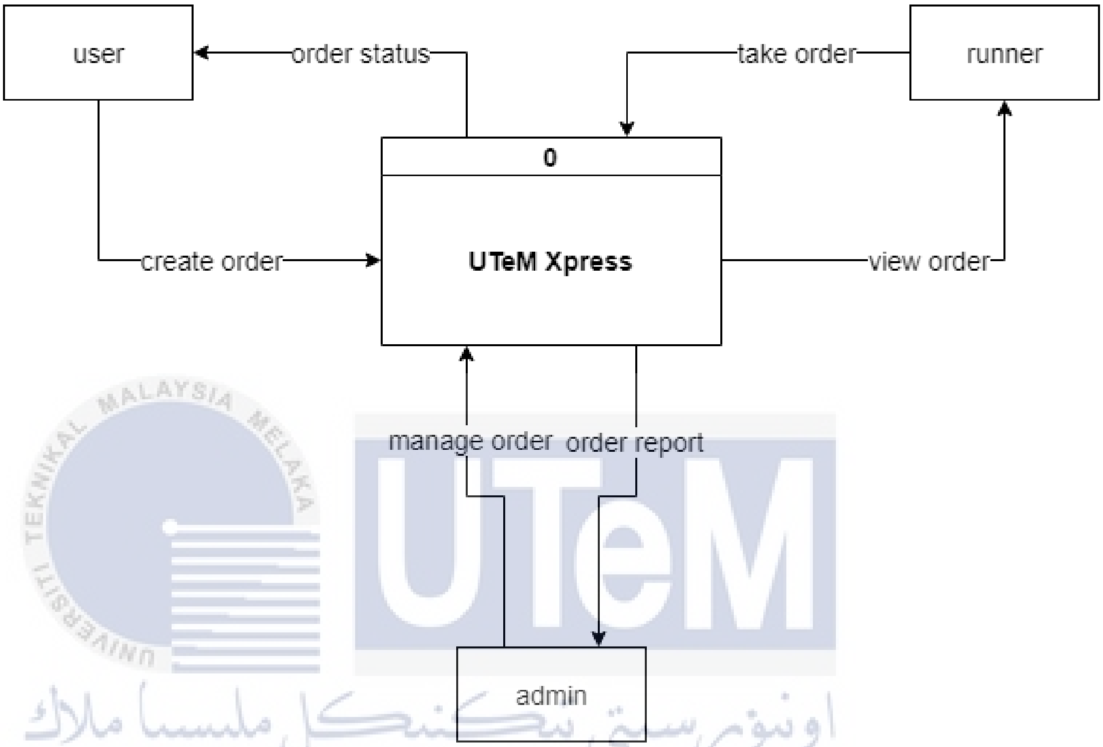


Figure 3.2: Context Diagram DFD

UTeM Xpress system is developed to ease the process of university communities to purchase goods and to deliver things from day-to-day activities. The end user of this system is user who is referring to student and staff of university, the runner which only student allowed to register and admin as to management daily reports for making sure the system run smoothly and maintaining business continuity. Each user is provided with different set of activities on each side. The user can create an order and get updated by the system of the current order status until its completion. The runner can view the order created by the users based on its registered vehicle and take the order and can start delivering the orders. For the admin side, they are provided with management report information of the whole and able to see the report.

(b) Level 0 DFD

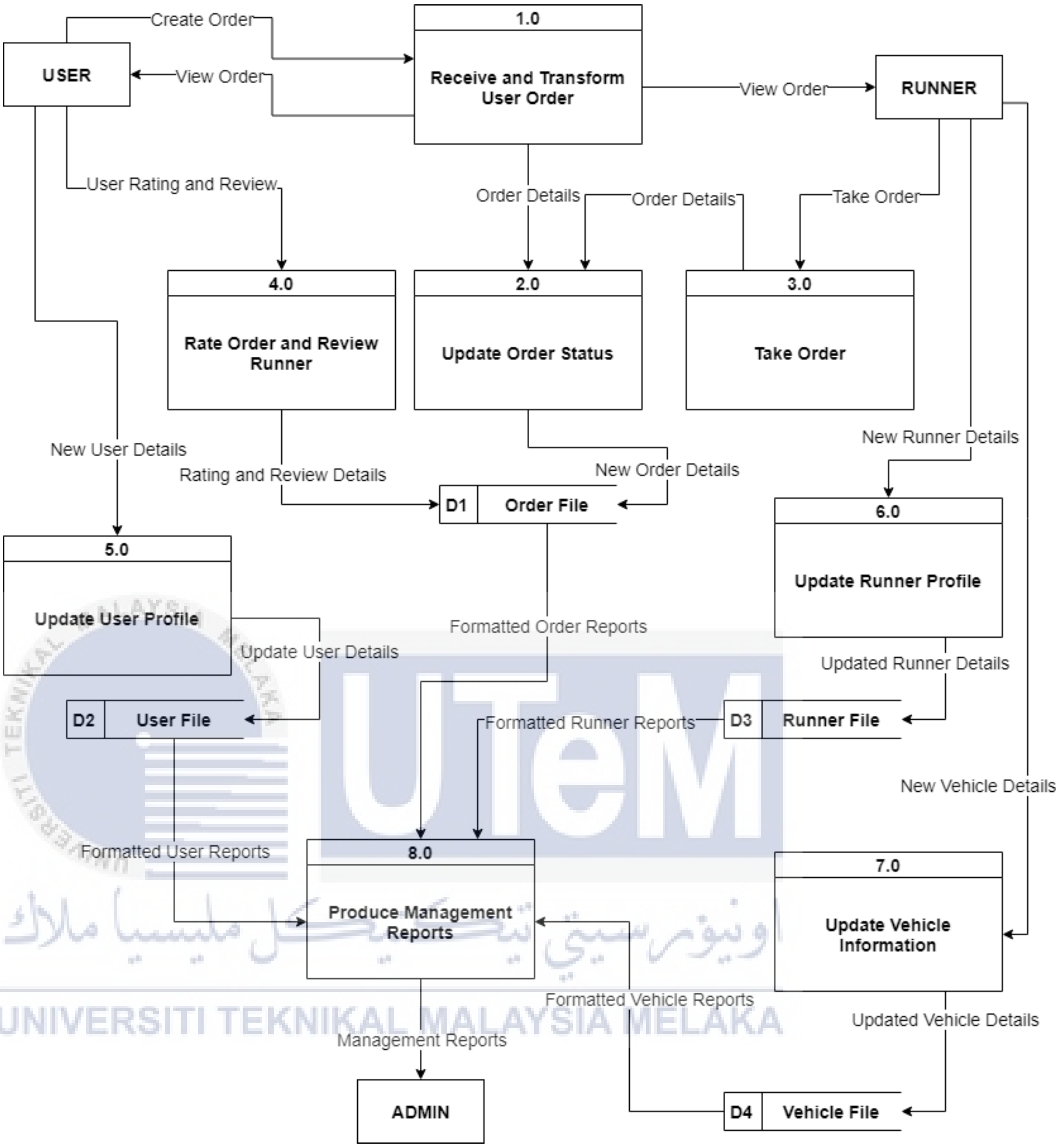


Figure 3.3: Level 0 DFD

3.3.3 Non-functional Requirement

Table 3.14: Non-functional Requirement Table

No.	Module	Description
1.	Performance & Scalability	The system shall load the web page within accepted range amount of time. Also, the system needs to return the result request by the user within 2 seconds. The system also can be able to support high load volume of usage and concurrent users at the same time and easy to scale the system if the number of users getting higher.
2.	Portability and Compatibility	This system shall be able to run across various type of operating systems, browsers, and devices without having any conflict.
3.	Reliability, Availability and Maintainability	The system must be trusted in managing and keeping user's credentials and privacy from disclosure. The system must be available for access by users whenever and wherever users are. For a certain period, the system can be temporarily rested to make way for repair work.
4.	Security	The system must be able to maintain the security of user data from being invaded by irresponsible parties or computer viruses. The system must ensure that only authorized users can access the system. The system must store the password in hashing format

		so that the database administrator cannot find out either.
5.	Usability	The system must use features that are easily understood by the user in the design of the system interface. The interface theme of this system must be the same as the other interfaces so as not to confuse the user. The use of font size, and colors must be clear so that it is easy for users to read

3.3.4 Others Requirement

Other requirement required by this system in term of hardware is a laptop with minimum 4Gb of ram to develop the system and internet connection at least 30Mbps to do searching information and testing. In term of software, it will require text editor called 'Php Storm' to write code, 'Xampp' as localhost web server to run the system and MySQL as database to store system data.

3.4 Conclusion

This chapter presents the explanation of all the requirements collected through analysis processes. The requirements that have been discussed in detailed are functional requirements, non-functional requirements, and data requirements. All the analyzed requirements will then be transferred to the design stage to produce all the needed designs, including database, software as well as hardware design which will be widely discussed in the next chapter.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter begins with the explanation of high-level design which consists of system architecture and user interface designs. In the user interface design section, it covers navigation design, input design and output design. Next, the high-level design describes database design in conceptual and logical designs which will be used in the development of this project. The next part of this chapter explains detailed design which covers software design and physical database design. Software design describes details of each function provided in the system.

4.2 High-Level Design

This section describes high level design in details, including system architecture, user interface design and database design.

4.2.1 System Architecture

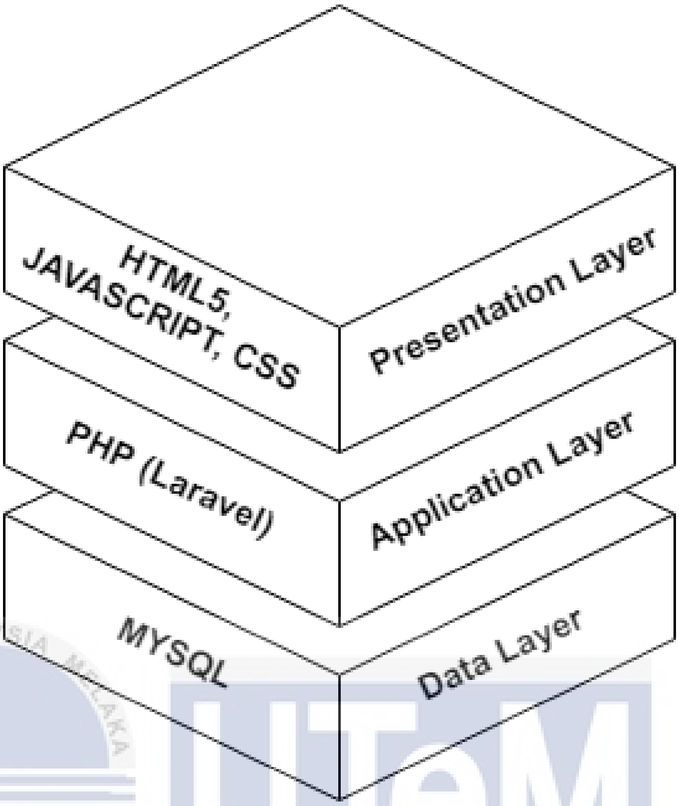


Figure 4.1: Project Architecture

UTeM Xpress was built using a three-tier system design. It is a client-server architecture that includes capabilities for display, application processing, and data administration. Each function is split into three levels and physically segregated.

Three-tier web application development is commonly used in popular websites. The presentation layer is the application's highest level. The user interface is presented at this layer as a means of communication with the user. Its primary responsibility is to analyze user requests and transform the results into a format that the user can comprehend. This system's presentation tier is the user's web browser, which is used to access it. The front-end platform for displaying the data supplied from the server is the browser.

Business logic tier, application layer, and middle layer are all terms used to describe the logic tier. As seen in Figure (figure’s number), this layer sits between the display and data tiers. The functioning of an application is controlled by this layer, which processes instructions, does calculations, and makes logical judgments and assessments.

Between the data tier and the presentation layer, it is also in charge of data access and processing. The application server, XAMPP server, is the system's intermediate layer, and it performs content processing using the Laravel Framework.

The data tier is the lowest layer, and it contains data persistence methods. This is the place where data from a database or file system is saved and extracted. This layer sends the data from the database to the logic tier for thorough processing, and then displays the results to the user. MySQL is the system's backend database, which includes both data sets and database management tools for administering and accessing the data stored within.

4.2.2 User Interface Design

This section covers all the UTeM Xpress's user interface designs, including navigation, input, and output design. A screenshot image will be included with each user interface design.

4.2.2.1 Navigation Design

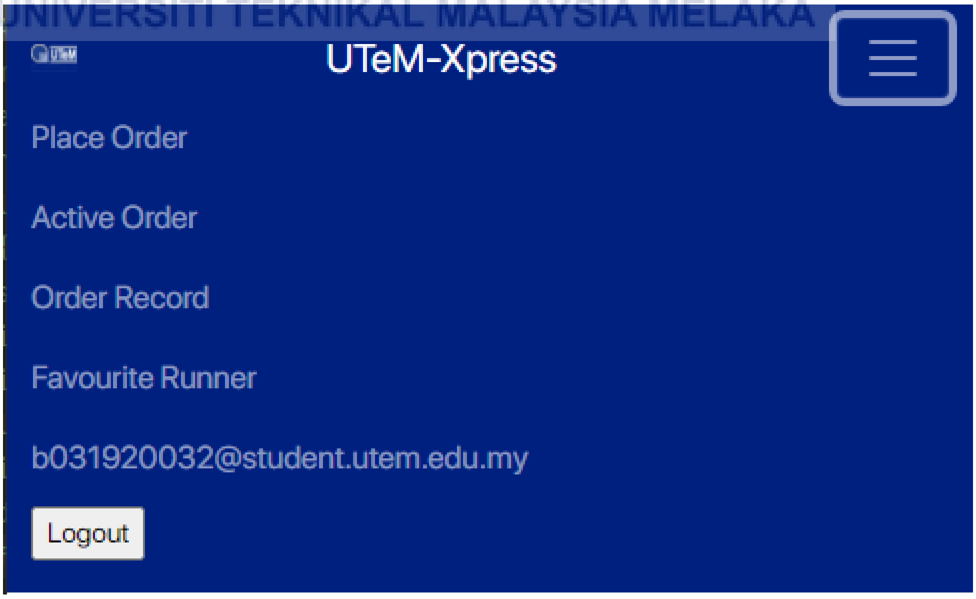


Figure 4.2: User Navigation

The navigation menu bar of this system is seen in the diagram above. Every webpage has a top-level navigation menu. The user may use this to access all of the modules' functionalities. The navigation path to each function is depicted in the structural diagram shown above.

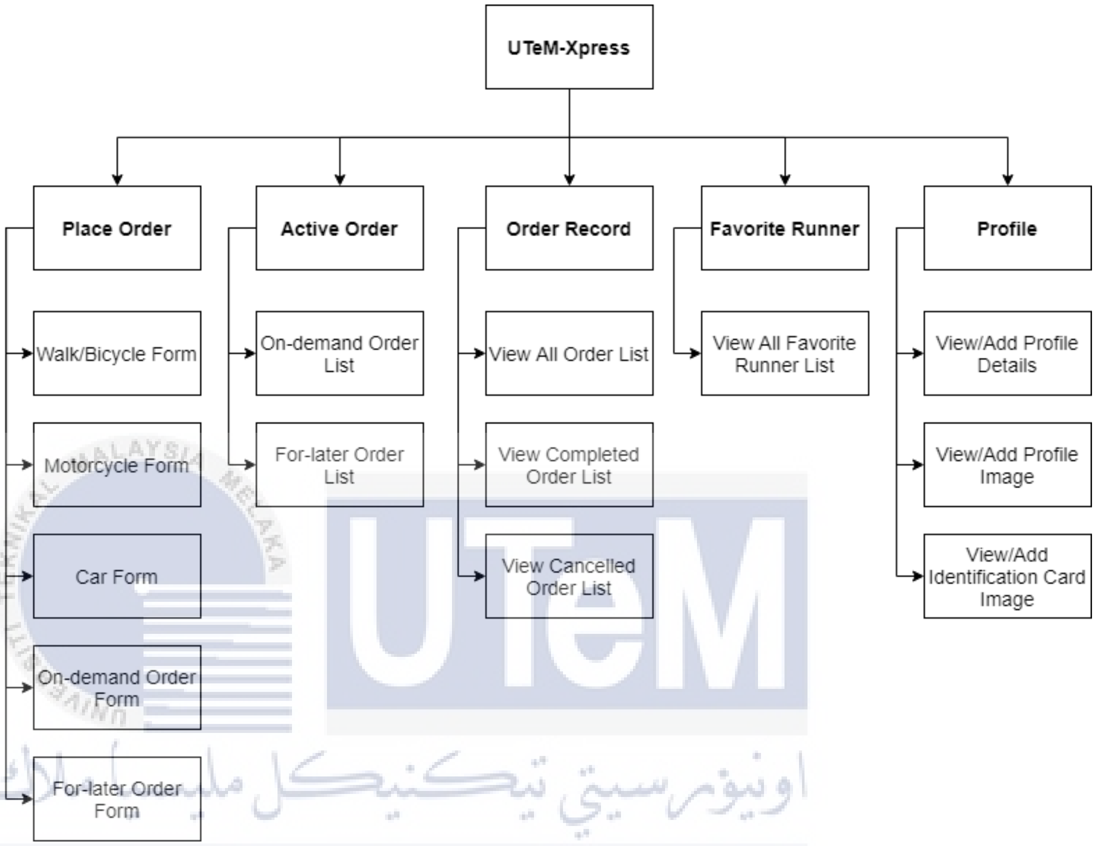
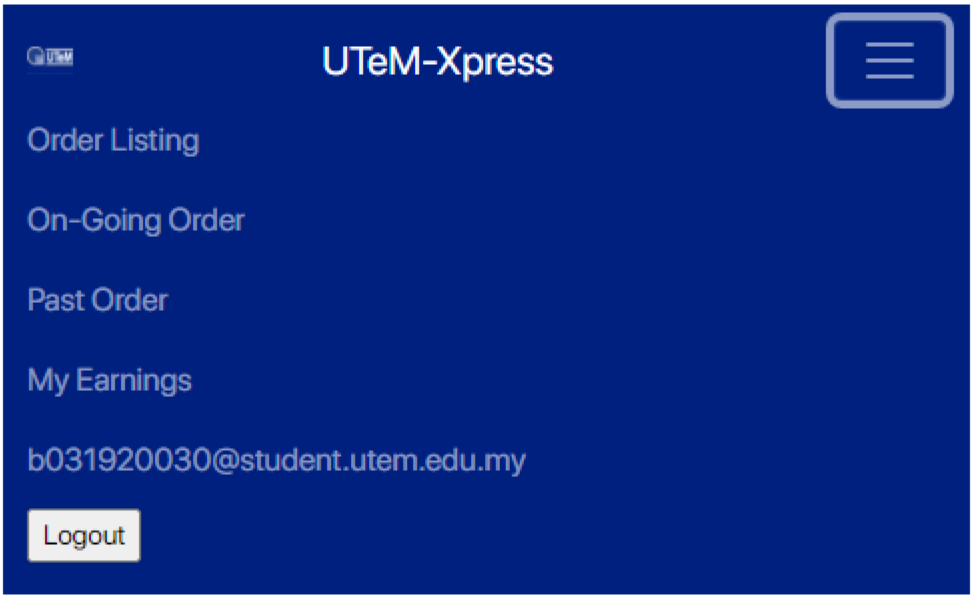


Figure 4.3: User Navigation Breakdown



UTeM-Xpress

Order Listing

On-Going Order

Past Order

My Earnings

b031920030@student.utem.edu.my

Logout

Figure 4.4: Runner Navigation

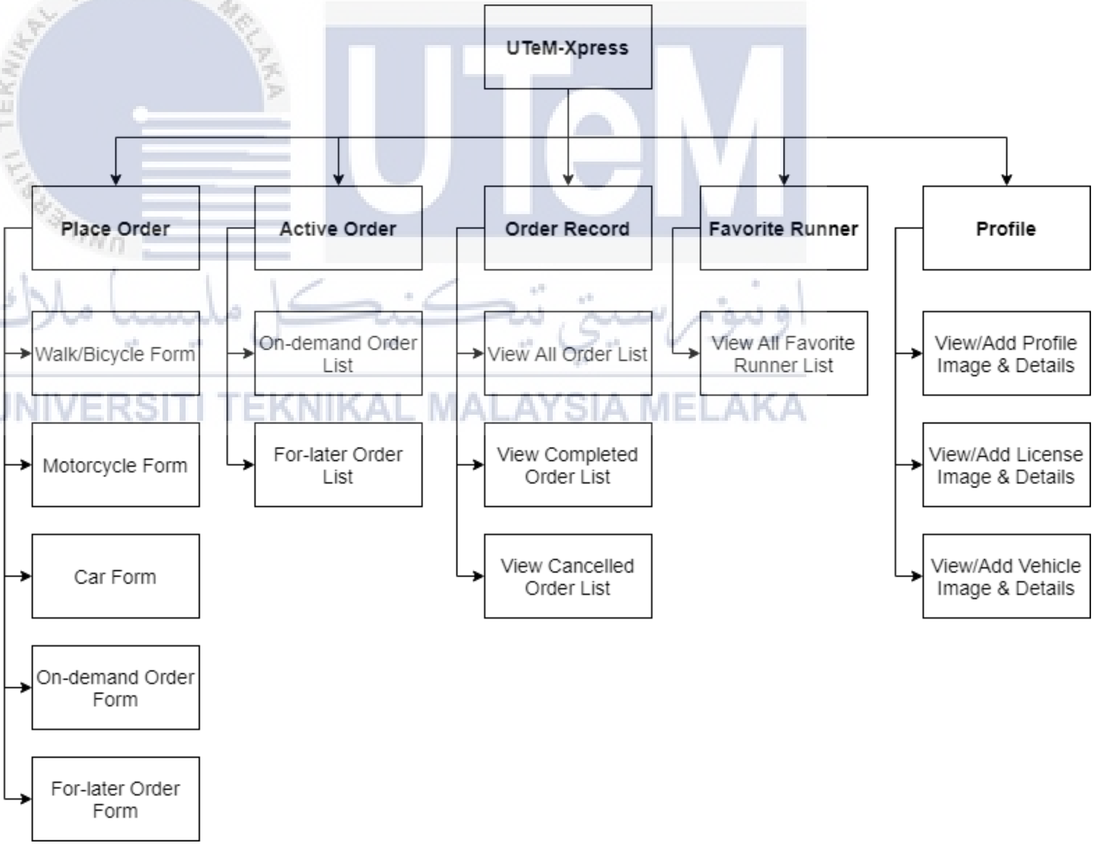


Figure 4.5: Runner Navigation Breakdown

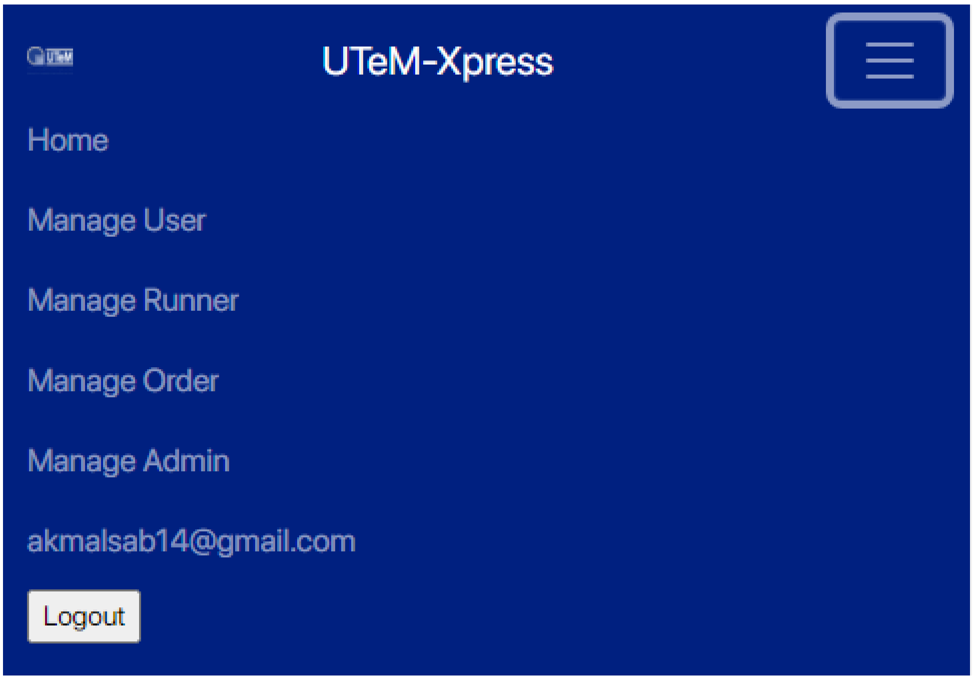


Figure 4.6: Admin Navigation

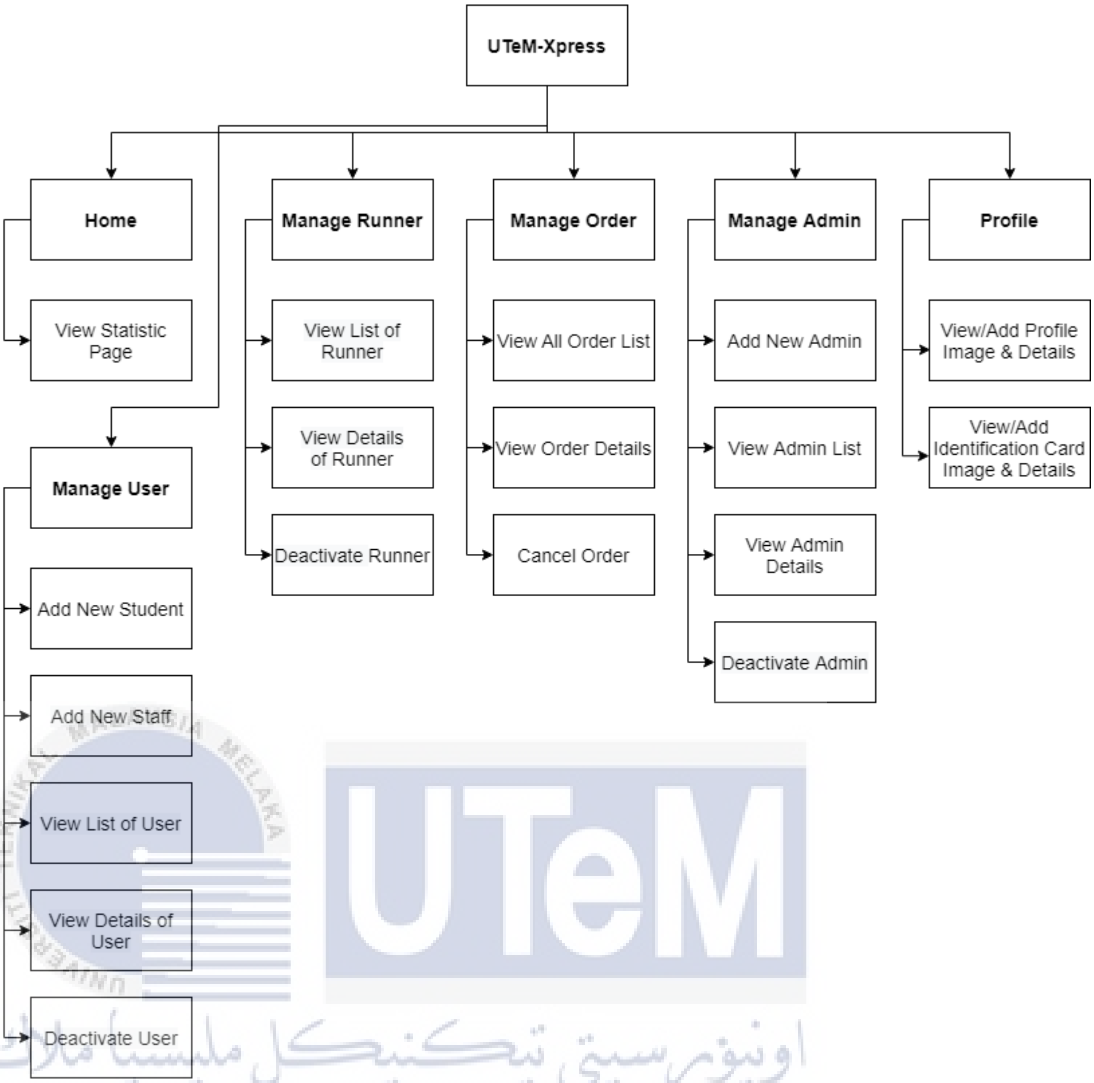
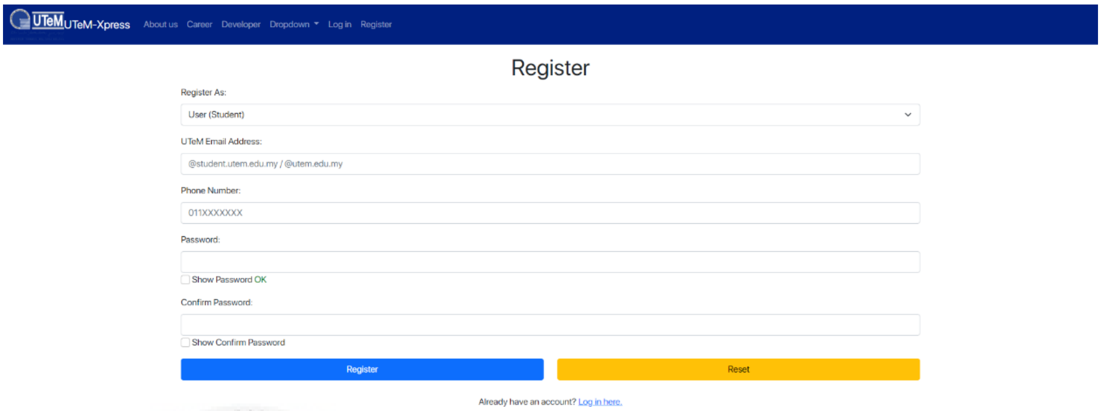


Figure 4.7: Admin Navigation Breakdown

4.2.2.2 Input Design

This section contains all the interface designs of the screen that enable user to input data into this system.



Register

Register As:
User (Student)

UTeM Email Address:
@student.utem.edu.my / @utem.edu.my

Phone Number:
011XXXXXXX

Password:

□ Show Password OK

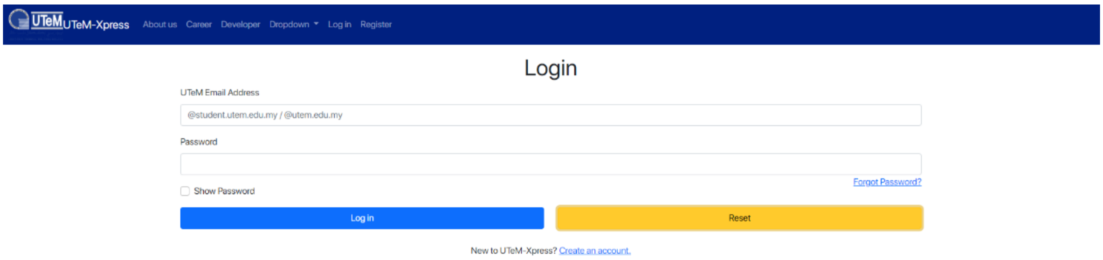
Confirm Password:

□ Show Confirm Password

Already have an account? Log in here.

Figure 4.8: Registration Form Screen

Figure 4.8 shows the input screen for user to register into the system. The system's user consists of customer which referring to student and staff of UTeM. Second is Runner which only for student and lastly is admin which can be only register by staff. First, user need to determine they want to register as what role, next user need to enter his or her university's email, personal phone number and lastly password and confirm password. Password must be 8 character and above for security purposes.



Login

UTeM Email Address
@student.utem.edu.my / @utem.edu.my

Password

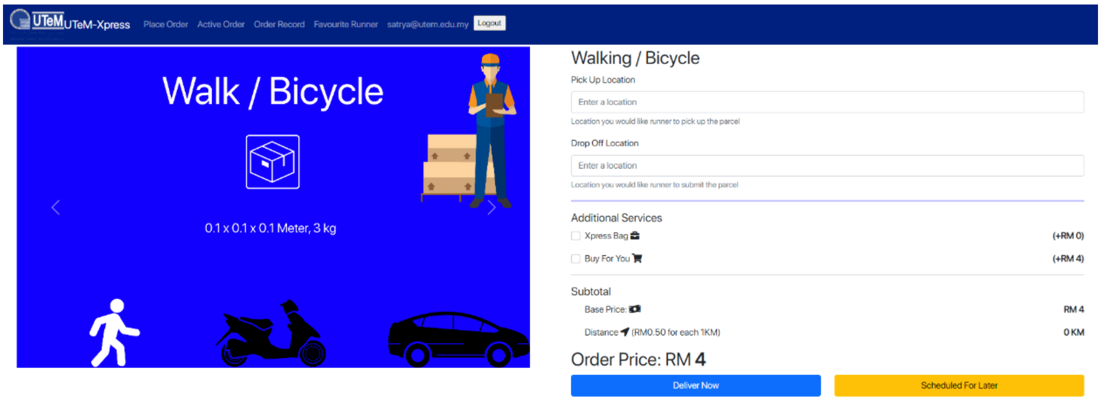
Forgot Password?

□ Show Password

New to UTeM-Xpress? Create an account.

Figure 4.9: Login Form Screen

Figure 4.9 above shows the login page where users need to enter their credentials which is university's email and password to logon into system. If they do not have account yet they can click 'Create an account' link below the login and reset button.



Walking / Bicycle

Pick Up Location

Enter a location

Location you would like runner to pick up the parcel

Drop Off Location

Enter a location

Location you would like runner to submit the parcel

Additional Services

□ Xpress Bag	(+RM 0)
□ Buy For You	(+RM 4)

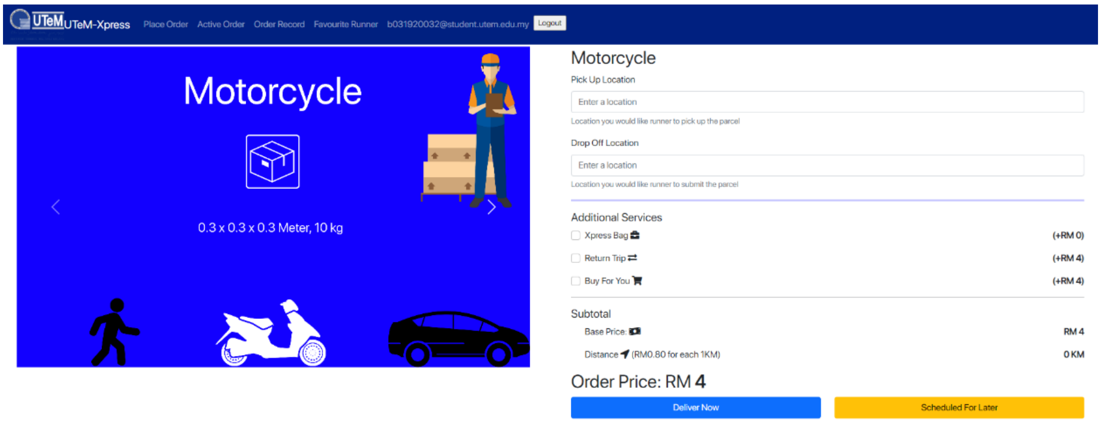
Subtotal

Base Price:	RM 4
Distance (RM0.50 for each 1KM)	0 KM

Order Price: RM **4**

Figure 4.10: Walk Order Form

Figure 4.10 above show the order form specifically for walking or bicycle as mode of transportation. The user was informed the acceptable capacity of item that has been set for this kind of transportation in the poster provided on the left side of the screen. Users need to input the desired pickup and drop off location for the runner to deliver those items. They are also able to select additional services provided for the current mode of transportation. Services for ‘Xpress Bag’ mean that the users require for the runner to bring the delivery bag to put the item. Services ‘Buy for You’ mean that the users need the runner to buy their desired item from pickup location that has been set and they need to pay the exact amount of purchase goods price to the runner back. The order fee details were also displayed just to make the user confirmed with the details entered before proceeding to next step. There are two buttons provided on the below which is ‘Deliver Now’ mean that the order is on-demand and the user need the runner to deliver the order right now while the second button is ‘Scheduled for Later’ mean that the user want to set the order earlier but want the runner to deliver the order on certain date and time.



Motorcycle

Pick Up Location

Enter a location

Location you would like runner to pick up the parcel

Drop Off Location

Enter a location

Location you would like runner to submit the parcel

Additional Services

□ Xpress Bag	(+RM 0)
□ Return Trip ⇄	(+RM 4)
□ Buy For You	(+RM 4)

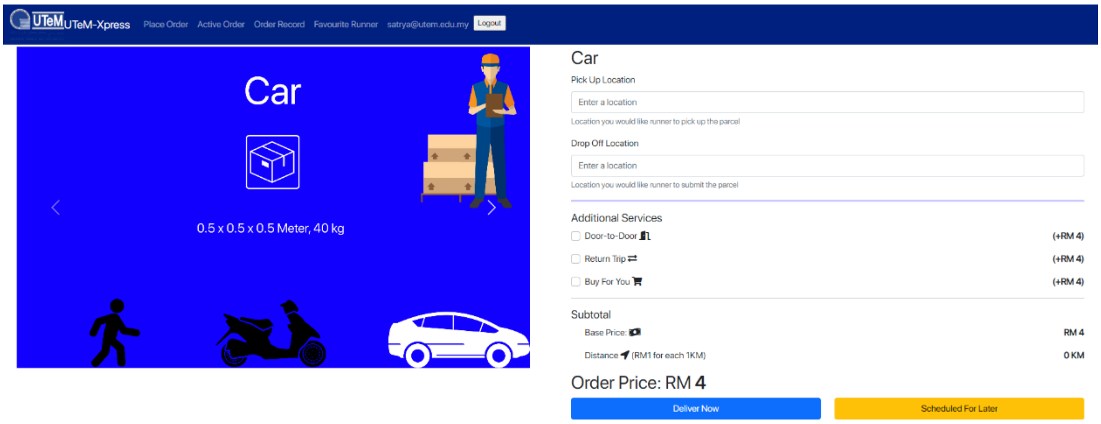
Subtotal

Base Price:	RM 4
Distance (RM0.80 for each 1KM)	0 KM

Order Price: RM **4**

Figure 4.11: Motorcycle Order Form

Figure 4.11 above show the order form specifically for motorcycle as mode of transportation. The user was informed the acceptable capacity of item that has been set for this kind of transportation in the poster provided on the left side of the screen. Users need to input the desired pickup and drop off location for the runner to deliver those items. They are also able to select additional services provided for the current mode of transportation. Services for 'Xpress Bag' mean that the users require for the runner to bring the delivery bag to put the item. Services 'Buy for You' mean that the users need the runner to buy their desired item from pickup location that has been set and they need to pay the exact amount of purchase goods price to the runner back. Services for 'Return Trip' mean that the user need the runner pick up the item at the pickup location then proceed to drop off location then return to the pickup location to deliver the item. The order fee details were also displayed just to make the user confirmed with the details entered before proceeding to next step. There are two buttons provided on the below which is 'Deliver Now' mean that the order is on-demand and the user need the runner to deliver the order right now while the second button is 'Scheduled for Later' mean that the user want to set the order earlier but want the runner to deliver the order on certain date and time.



Car

Pick Up Location

Enter a location

Location you would like runner to pick up the parcel

Drop Off Location

Enter a location

Location you would like runner to submit the parcel

Additional Services

□ Door-to-Door	(+RM 4)
□ Return Trip ⇄	(+RM 4)
□ Buy For You 🛒	(+RM 4)

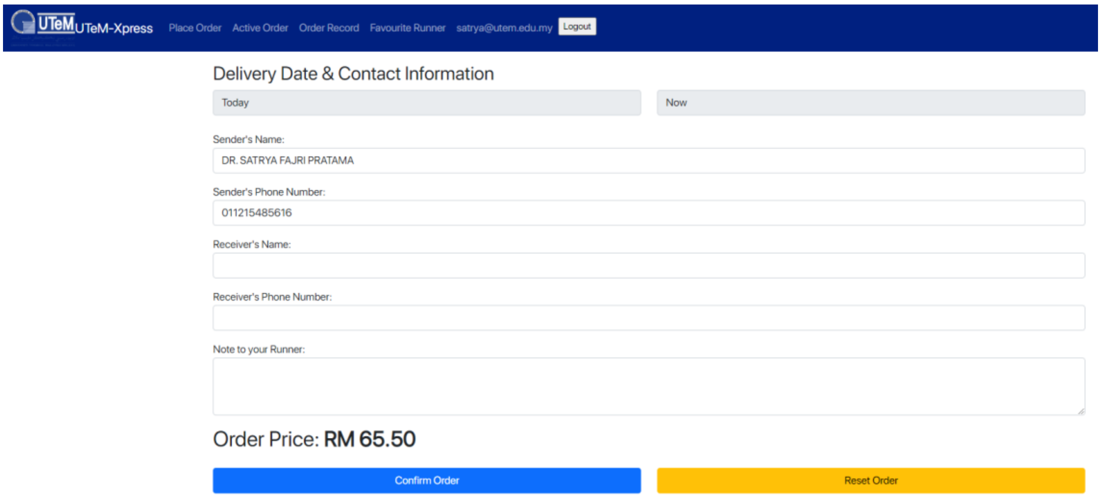
Subtotal

Base Price:	RM 4
Distance (RM1 for each 1KM)	0 KM

Order Price: RM **4**

Figure 4.12: Car Order Form

Figure 4.12 above show the order form specifically for car as mode of transportation. The user was informed the acceptable capacity of item that has been set for this kind of transportation in the poster provided on the left side of the screen. Users need to input the desired pickup and drop off location for the runner to deliver those items. They are also able to select additional services provided for the current mode of transportation. Services for ‘Door to Door’ mean that the users need the runner to pick up the item and deliver the item in front of the door of location that has been set. Services ‘Buy for You’ mean that the users need the runner to buy their desired item from pickup location that has been set and they need to pay the exact amount of purchase goods price to the runner back. Services for ‘Return Trip’ mean that the user need the runner pick up the item at the pickup location then proceed to drop off location then return to the pickup location to deliver the item. The order fee details were also displayed just to make the user confirmed with the details entered before proceeding to next step. There are two buttons provided on the below which is ‘Deliver Now’ mean that the order is on-demand and the user need the runner to deliver the order right now while the second button is ‘Scheduled for Later’ mean that the user want to set the order earlier but want the runner to deliver the order on certain date and time.



Delivery Date & Contact Information

Today | Now

Sender's Name: DR. SATRYA FAJRI PRATAMA

Sender's Phone Number: 011215485616

Receiver's Name:

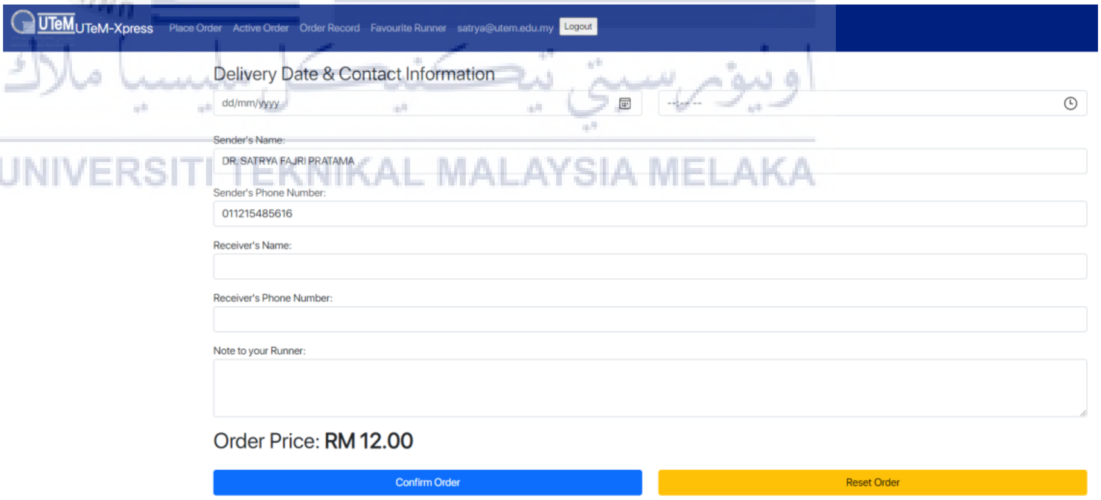
Receiver's Phone Number:

Note to your Runner:

Order Price: **RM 65.50**

Figure 4.13: Order Now Form

Figure 4.13 above show the ‘Order Now’ page which mean the user need the runner to deliver the order right now. The date and time automatically have been set to current date and time. The sender’s name and phone number also automatically set by using the user’s details. Users only need to enter the receiver’s name and phone number, also the extra remarks for runner references.



Delivery Date & Contact Information

dd/mm/yyyy | --:-- --

Sender's Name: DR. SATRYA FAJRI PRATAMA

Sender's Phone Number: 011215485616

Receiver's Name:

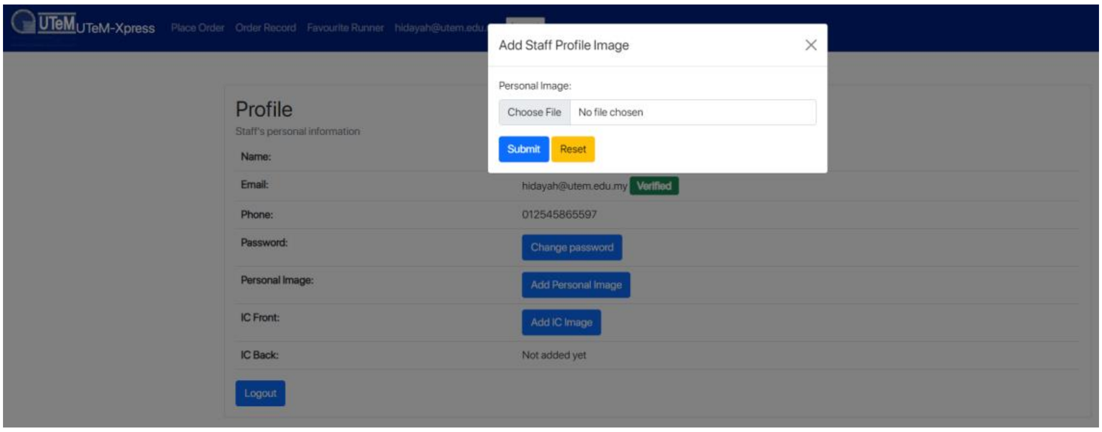
Receiver's Phone Number:

Note to your Runner:

Order Price: **RM 12.00**

Figure 4.14: Schedule for Later Form

Figure 4.14 above show the ‘Schedule for Later’ page which mean the user need the runner to deliver the order on certain date and time. The date and time need to be entered by the user. The sender’s name and phone number also automatically set by using the user's details. Users only need to enter the receiver's name and phone number, also the extra remarks for runner references.



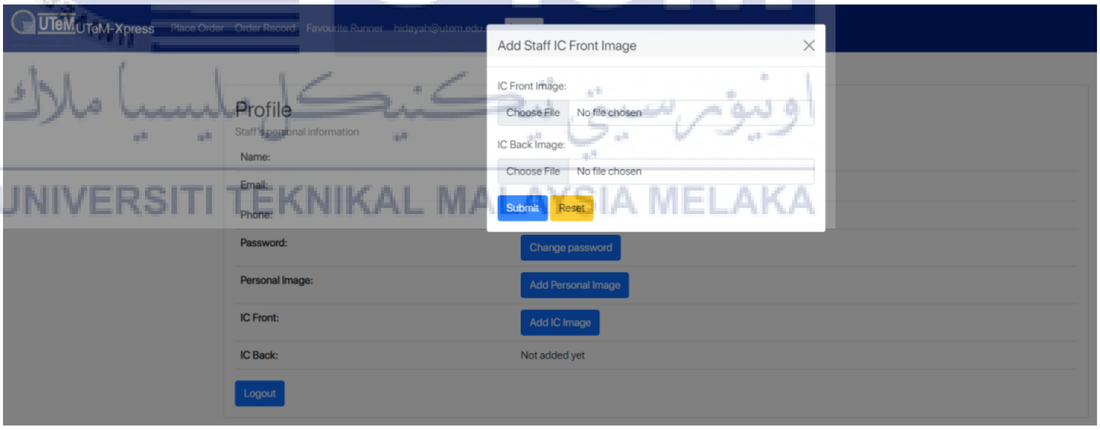
Personal Image: No file chosen

Profile
Staff's personal information

Name:
Email: hidayah@utem.edu.my
Phone: 012545865597
Password:
Personal Image:
IC Front:
IC Back: Not added yet

Figure 4.15: Add Staff Image Form

Figure 4.15 above show the input screen for the staff to enter their personal image. The user needs to click 'Choose File' button and browse for the image then click submits. A picture will be uploaded into the system.



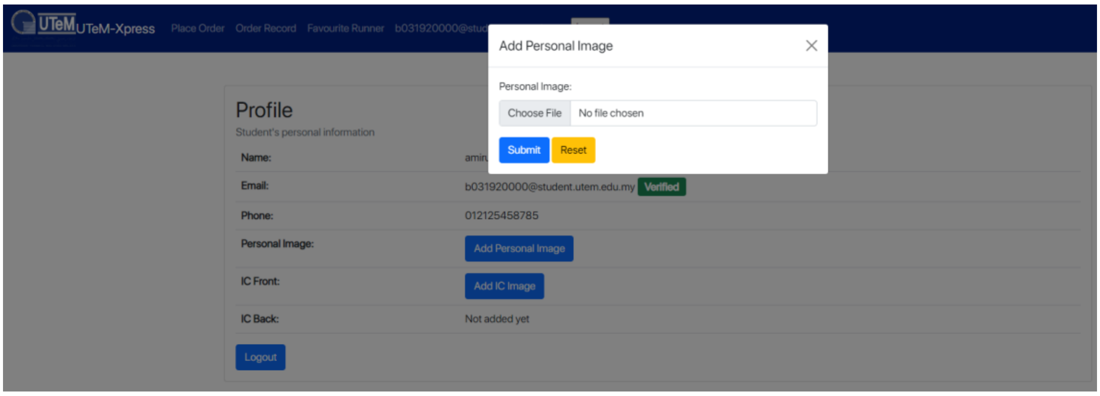
IC Front Image: No file chosen
IC Back Image: No file chosen

Profile
Staff's personal information

Name:
Email:
Phone:
Password:
Personal Image:
IC Front:
IC Back: Not added yet

Figure 4.16: Add IC Image Form

Figure 4.16 above show the input screen for the staff to enter their identification card image. The user needs to click 'Choose File' button and browse for the image then click submits. A picture will be uploaded into the system.



Add Personal Image

Personal Image: No file chosen

Profile

Student's personal information

Name: amir

Email: b031920000@student.utem.edu.my Verified

Phone: 012125458785

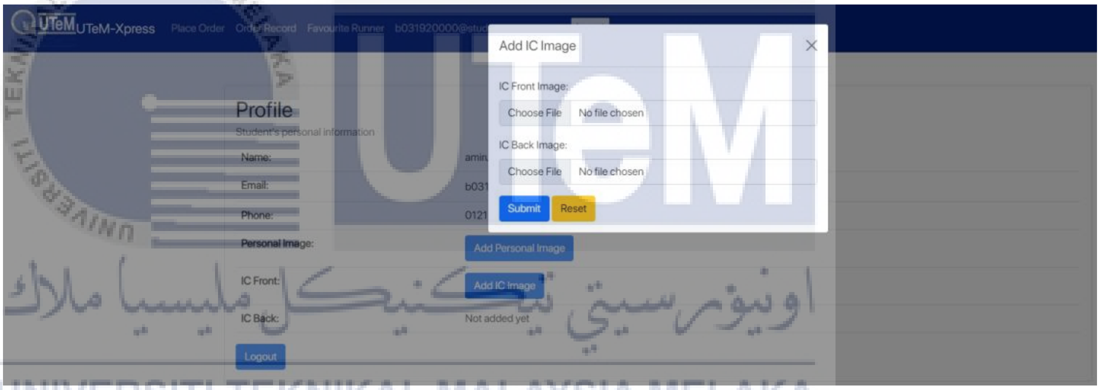
Personal Image:

IC Front:

IC Back: Not added yet

Figure 4.17: Add Personal Image Form

Figure 4.17 above show the input screen for the student to enter their personal image. The user needs to click ‘Choose File’ button and browse for the image then click submits. A picture will be uploaded into the system.



Add IC Image

IC Front Image: No file chosen

IC Back Image: No file chosen

Profile

Student's personal information

Name: amir

Email: b031

Phone: 0121

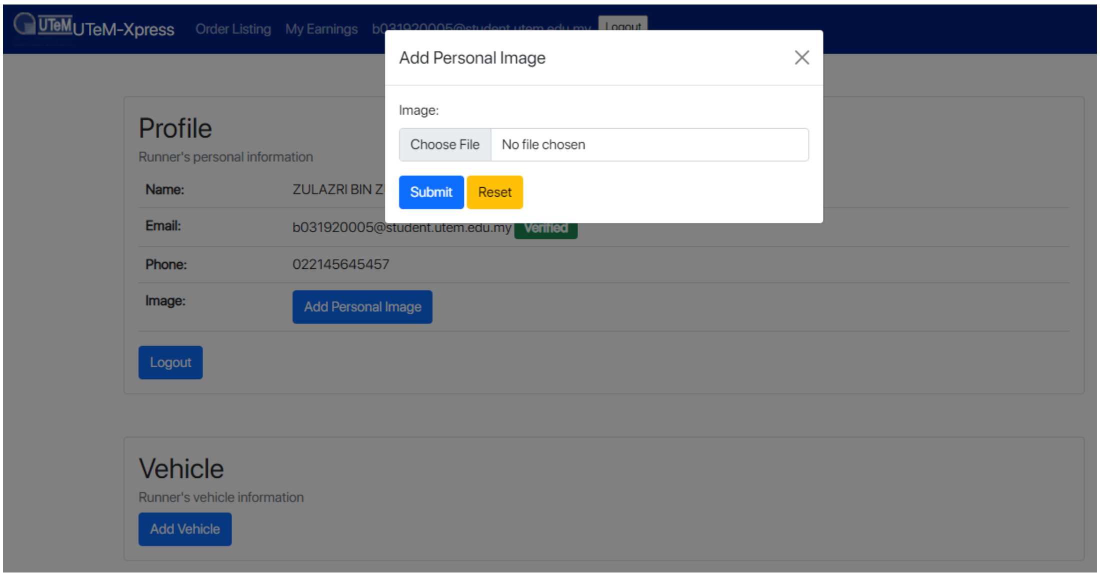
Personal Image:

IC Front:

IC Back: Not added yet

Figure 4.18: Add IC Personal Image Form

Figure 4.18 above show the input screen for the student to enter their identification card image. The user needs to click ‘Choose File’ button and browse for the image then click submits. A picture will be uploaded into the system.



Add Personal Image

Image:

Choose File No file chosen

Submit Reset

Profile

Runner's personal information

Name: ZULAZRI BIN Z

Email: b031920005@student.utem.edu.my Verified

Phone: 022145645457

Image: Add Personal Image

Logout

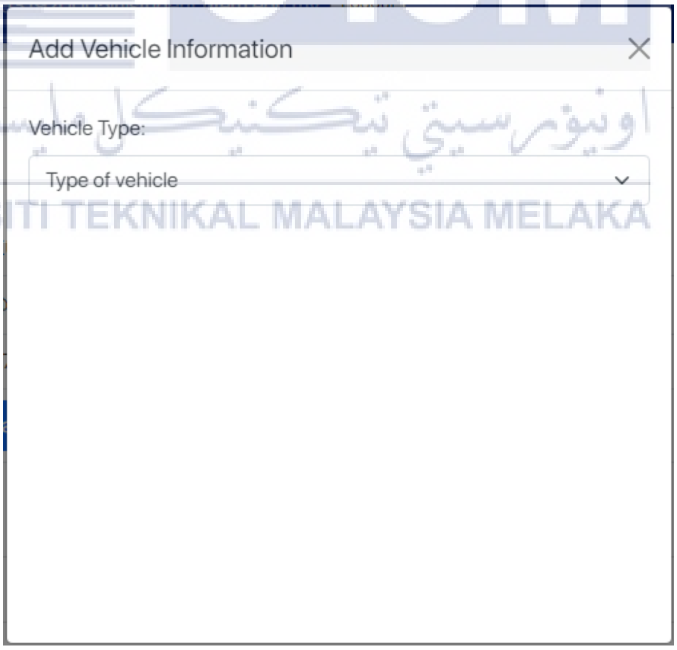
Vehicle

Runner's vehicle information

Add Vehicle

Figure 4.19: Add Runner Personal Image Form

Figure 4.19 above show the input screen for the runner to enter their personal image. The user needs to click 'Choose File' button and browse for the image then click submits. A picture will be uploaded into the system.



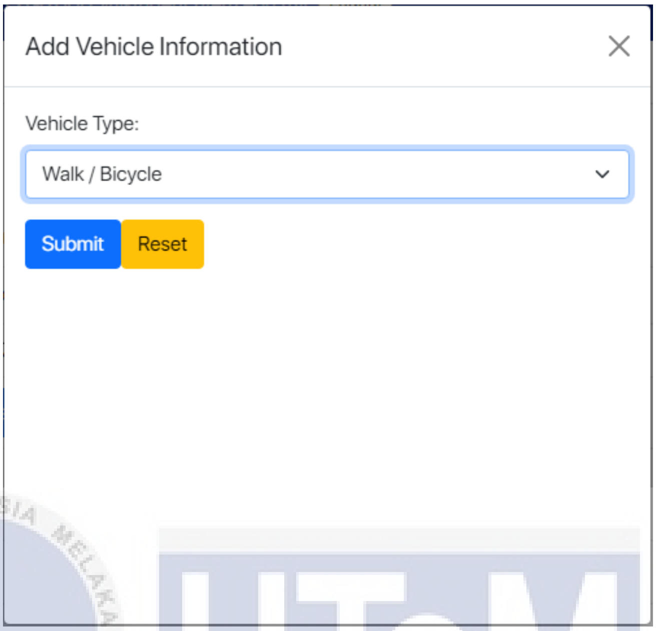
Add Vehicle Information

Vehicle Type:

Type of vehicle

Figure 4.20: Add Vehicle Information Form

Figure 4.20 above show the input screen for the runner to enter their vehicle information. The user needs to select type of vehicle provided in the dropdown list. The option provided is ‘Walk/Bicycle’, ’Motorcycle’, and ‘Car’.

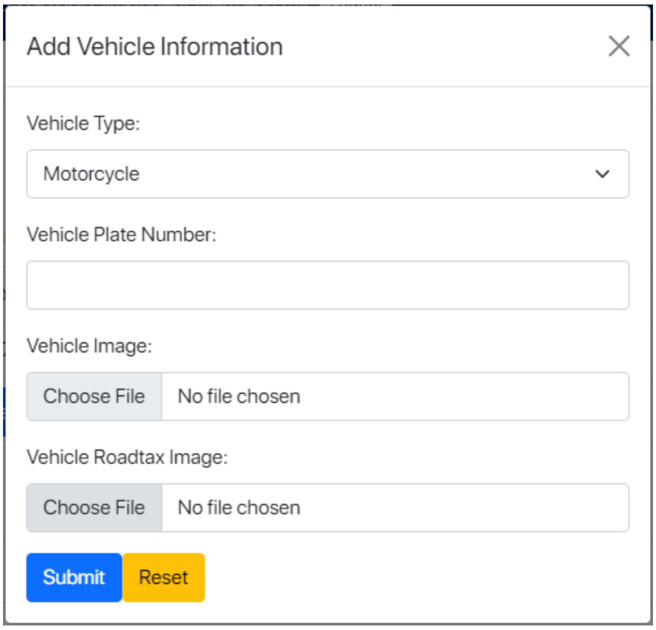


Add Vehicle Information

Vehicle Type: Walk / Bicycle

Figure 4.21: Walk Vehicle Form

Figure 4.21 above shows that if the runner selects the ‘Walk / Bicycle’ as their mode of transportation then the system does not need any other input, the user only needs to click submit button to confirm.



Add Vehicle Information

Vehicle Type: Motorcycle

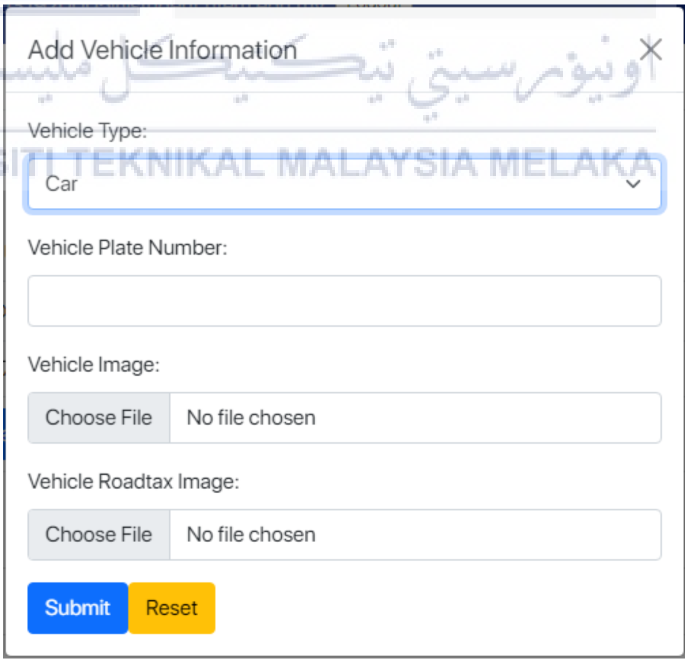
Vehicle Plate Number:

Vehicle Image: Choose File No file chosen

Vehicle Roadtax Image: Choose File No file chosen

Figure 4.22: Motorcycle Vehicle Form

Figure 4.22 above shows that if the runner selects the 'Motorcycle' as their mode of transportation then the system will show the input for the vehicle's plate number, image, and road-tax image, then user needs to click submit button to confirm.



Add Vehicle Information

Vehicle Type: Car

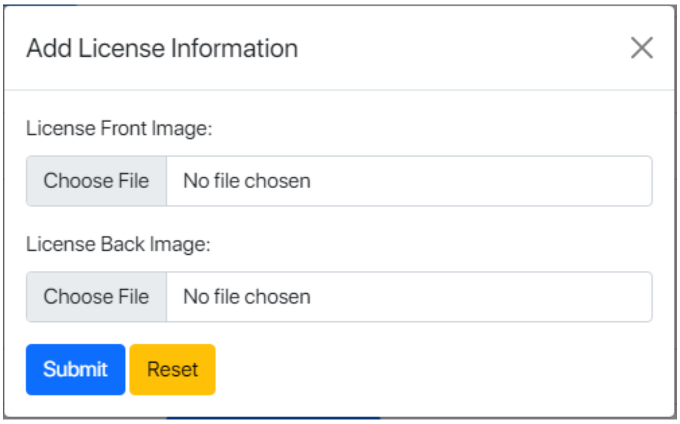
Vehicle Plate Number:

Vehicle Image: Choose File No file chosen

Vehicle Roadtax Image: Choose File No file chosen

Figure 4.23: Car Vehicle Form

Figure 4.23 above shows that if the runner selects the 'Car' as their mode of transportation then the system will show the input for the vehicle's plate number, image, and road-tax image, then user needs to click submit button to confirm.



Add License Information

License Front Image:

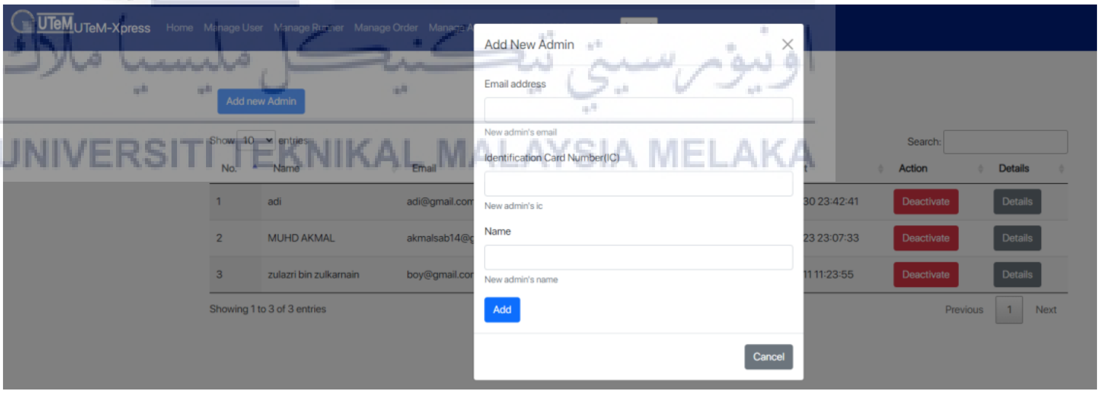
Choose File No file chosen

License Back Image:

Choose File No file chosen

Figure 4.24: Runner Add License Image Form

Figure 4.24 above show the input screen for the runner to enter their license image. The user needs to click 'Choose File' button and browse for the image then click submits. A picture will be uploaded into the system.



Show 10 entries

Search:

No.	Name	Email		Action	Details
1	adi	adi@gmail.com	30 23:42:41	Deactivate	Details
2	MUHD AKMAL	akmalsab14@g	23 23:07:33	Deactivate	Details
3	zulazri bin zulkarnain	boy@gmail.cor	11 11:23:55	Deactivate	Details

Showing 1 to 3 of 3 entries

Add New Admin

Email address

New admin's email

Identification Card Number(IC)

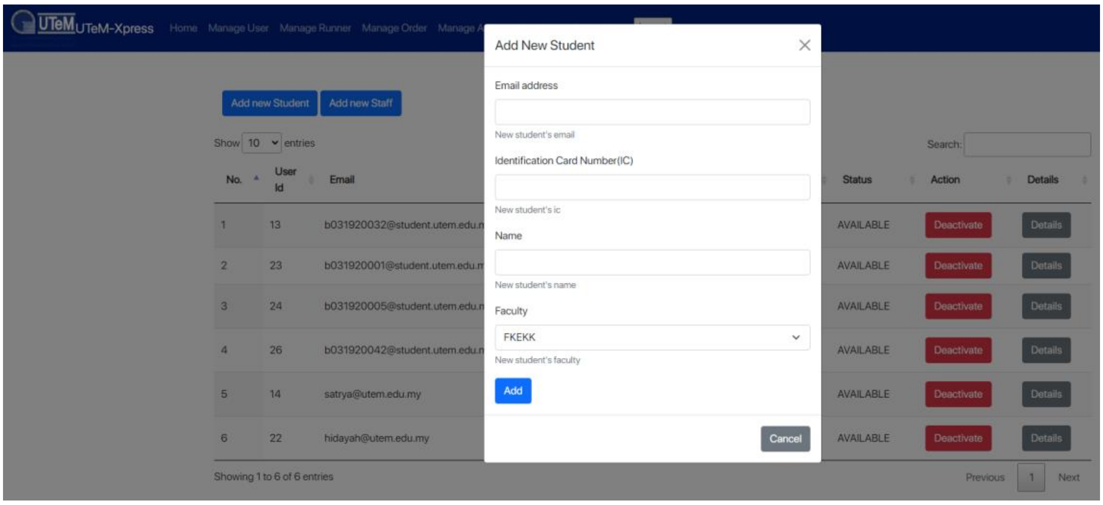
New admin's ic

Name

New admin's name

Figure 4.25: Add New Admin Form

Figure 4.25 above shows the input screen for admin to create new admin. The admin is required to enter email address, identification number, and name for the new admin then click add to register new admin.



Add New Student

Email address

New student's email

Identification Card Number(IC)

New student's ic

Name

New student's name

Faculty

FKEKK

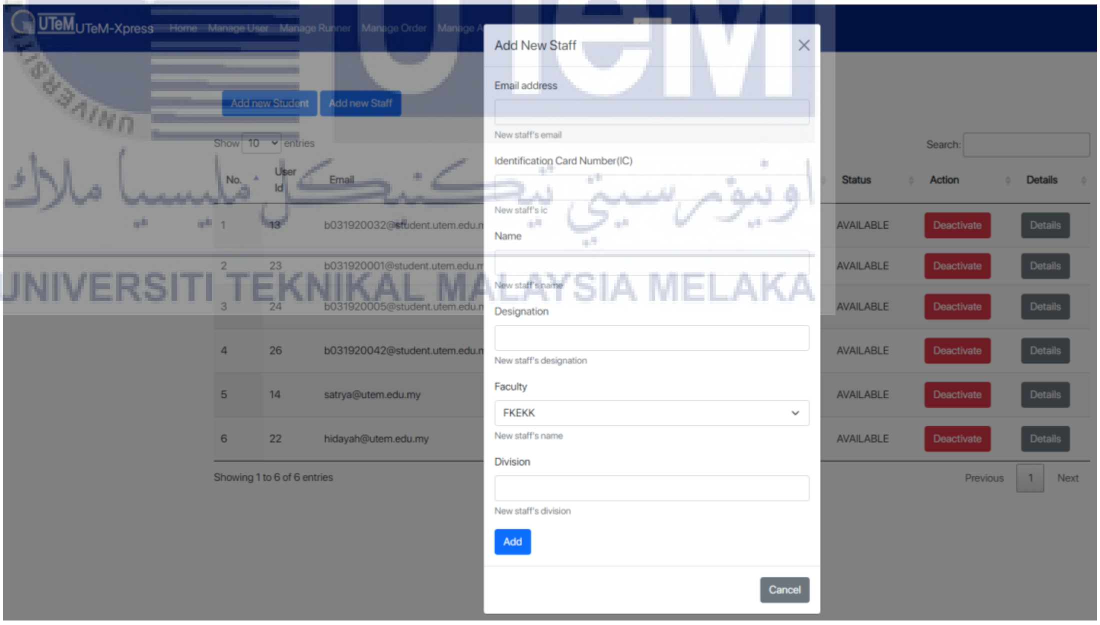
New student's faculty

No.	User Id	Email	Status	Action	Details
1	13	b031920032@student.utem.edu.m	AVAILABLE	Deactivate	Details
2	23	b031920001@student.utem.edu.m	AVAILABLE	Deactivate	Details
3	24	b031920005@student.utem.edu.m	AVAILABLE	Deactivate	Details
4	26	b031920042@student.utem.edu.m	AVAILABLE	Deactivate	Details
5	14	satrya@utem.edu.my	AVAILABLE	Deactivate	Details
6	22	hidayah@utem.edu.my	AVAILABLE	Deactivate	Details

Showing 1 to 6 of 6 entries

Figure 4.26: Add New Student Form

Figure 4.26 above shows the input screen for admin to create new student. The admin is required to enter email address, identification number, name and faculty for the new student then click add to register new student.



Add New Staff

Email address

New staff's email

Identification Card Number(IC)

New staff's ic

Name

New staff's name

Designation

New staff's designation

Faculty

FKEKK

New staff's name

Division

New staff's division

No.	User Id	Email	Status	Action	Details
1	13	b031920032@student.utem.edu.m	AVAILABLE	Deactivate	Details
2	23	b031920001@student.utem.edu.m	AVAILABLE	Deactivate	Details
3	24	b031920005@student.utem.edu.m	AVAILABLE	Deactivate	Details
4	26	b031920042@student.utem.edu.m	AVAILABLE	Deactivate	Details
5	14	satrya@utem.edu.my	AVAILABLE	Deactivate	Details
6	22	hidayah@utem.edu.my	AVAILABLE	Deactivate	Details

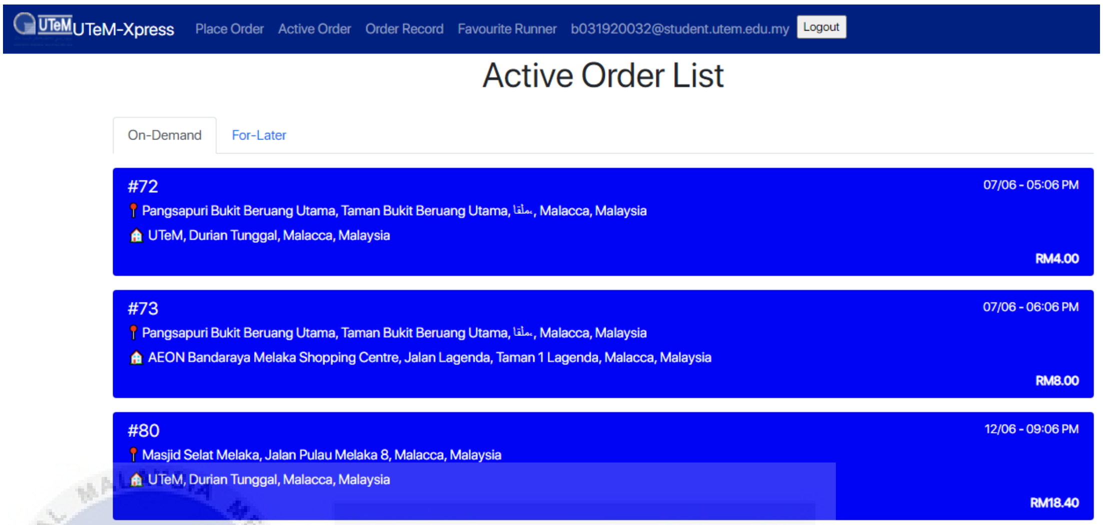
Showing 1 to 6 of 6 entries

Figure 4.27: Add New Staff Form

Figure 4.27 above shows the input screen for admin to create new staff. The admin is required to enter email address, identification number, name, designation, faculty, and division for the new staff then click add to register new staff.

4.2.2.3 Output Design

This section contains all the interface designs of the screen that display output from this system.



UTeM UTeM-Xpress Place Order Active Order Order Record Favourite Runner b031920032@student.utem.edu.my Logout

Active Order List

On-Demand For-Later

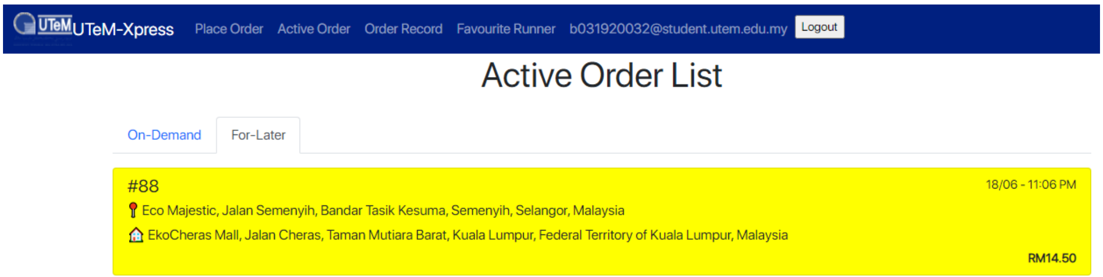
#72 07/06 - 05:06 PM
Pangsapuri Bukit Beruang Utama, Taman Bukit Beruang Utama, ملاكا, Malacca, Malaysia
UTeM, Durian Tunggal, Malacca, Malaysia
RM4.00

#73 07/06 - 06:06 PM
Pangsapuri Bukit Beruang Utama, Taman Bukit Beruang Utama, ملاكا, Malacca, Malaysia
AEON Bandaraya Melaka Shopping Centre, Jalan Lagenda, Taman 1 Lagenda, Malacca, Malaysia
RM8.00

#80 12/06 - 09:06 PM
Masjid Selat Melaka, Jalan Pulau Melaka 8, Malacca, Malaysia
UTeM, Durian Tunggal, Malacca, Malaysia
RM18.40

Figure 4.28: Active On-Demand Order List Screen

Figure 4.28 above show the active order list screen. This page is listing all the order that created by the users that have not completed yet. It displays the list of active order for 'on-demand' category for the user. Users need to click the order list to view the order details.



UTeM UTeM-Xpress Place Order Active Order Order Record Favourite Runner b031920032@student.utem.edu.my Logout

Active Order List

On-Demand For-Later

#88 18/06 - 11:06 PM
Eco Majestic, Jalan Semenyih, Bandar Tasik Kesuma, Semenyih, Selangor, Malaysia
EkoCheras Mall, Jalan Cheras, Taman Mutiara Barat, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM14.50

Figure 4.29: Active For-Later Order List Screen

Figure 4.29 above show the active order list screen. This page is listing all the order that created by the users that have not completed yet. It displays the list of active order for 'for-later' category for the user. Users need to click the order list to view the order details.

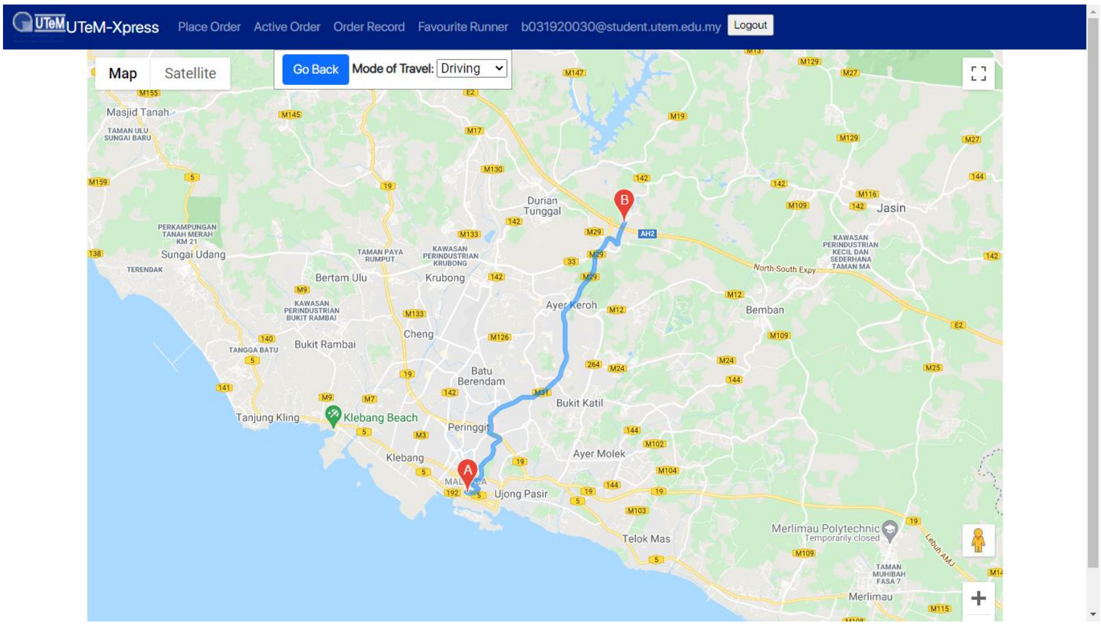
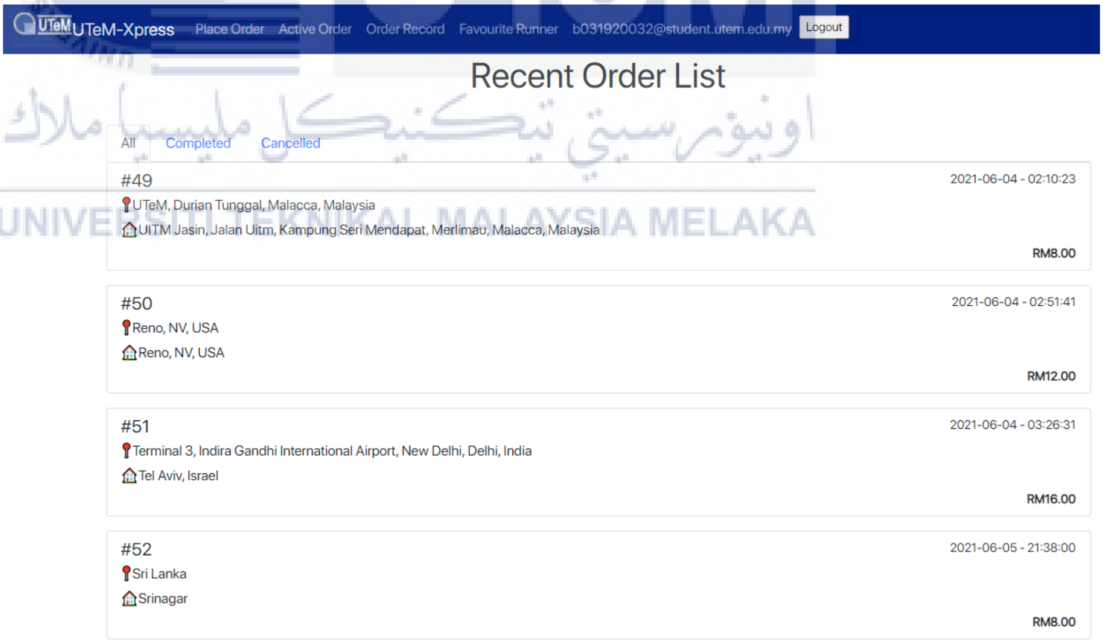


Figure 4.30: Map Screen

Figure 4.30 above show the map screen for the order. It displays straight two points of locations which referring to pick-up location and drop-off location.


UTeM-Xpress Place Order Active Order Order Record Favourite Runner b031920032@student.utem.edu.my Logout

Recent Order List

All Completed Cancelled

#49 2021-06-04 - 02:10:23
UTeM, Durian Tunggal, Malacca, Malaysia
UiTM Jasin, Jalan Uitm, Kampung Seri Mendapat, Merlimau, Malacca, Malaysia
RM8.00

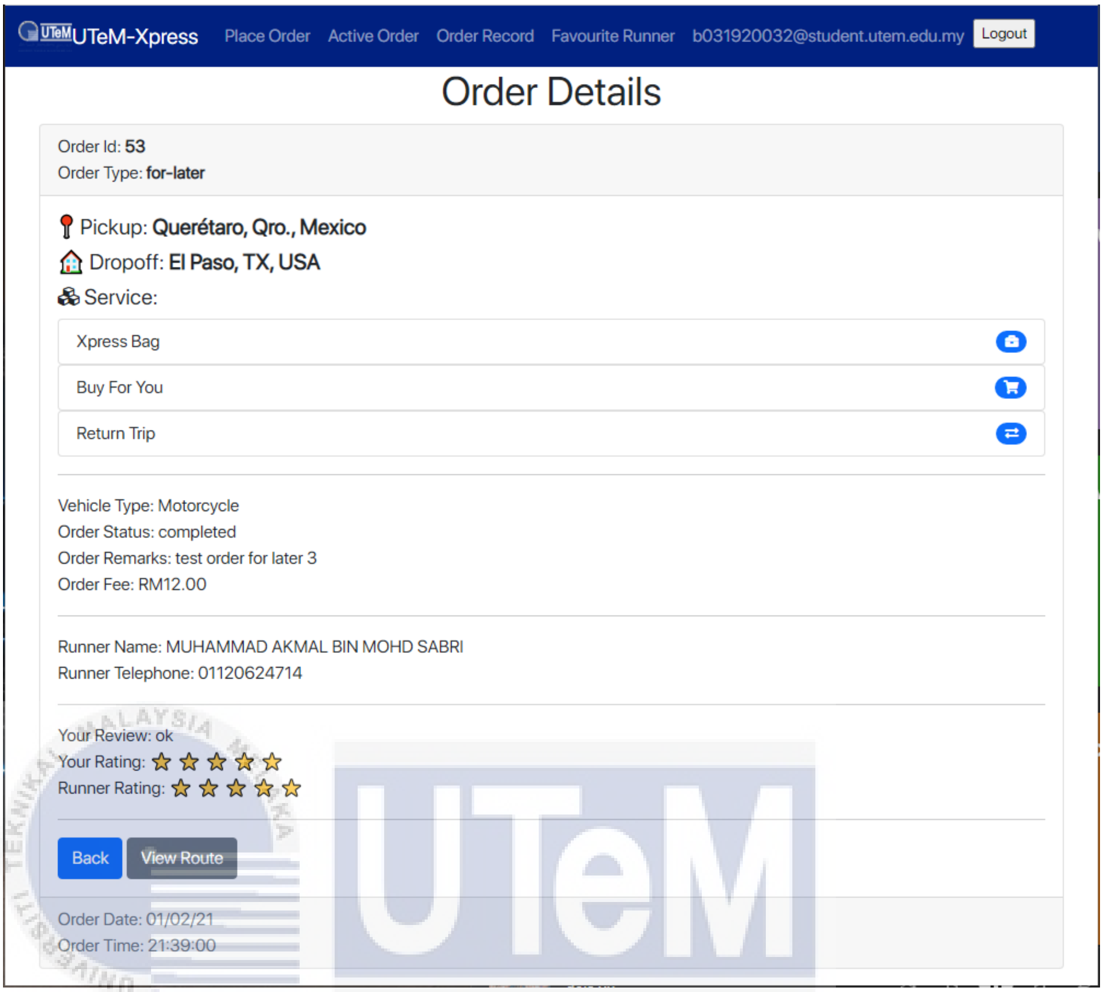
#50 2021-06-04 - 02:51:41
Reno, NV, USA
Reno, NV, USA
RM12.00

#51 2021-06-04 - 03:26:31
Terminal 3, Indira Gandhi International Airport, New Delhi, Delhi, India
Tel Aviv, Israel
RM16.00

#52 2021-06-05 - 21:38:00
Sri Lanka
Srinagar
RM8.00

Figure 4.31: Recent Order List Screen

Figure 4.31 above show the recent order list screen. It displays list of orders and categorize it based on its status which is complete, cancelled and all for both. Users need to click the order list to view the order details.



Order Details

Order Id: 53
Order Type: for-later

Pickup: Querétaro, Qro., Mexico
Dropoff: El Paso, TX, USA
Service:

Xpress Bag
Buy For You
Return Trip

Vehicle Type: Motorcycle
Order Status: completed
Order Remarks: test order for later 3
Order Fee: RM12.00

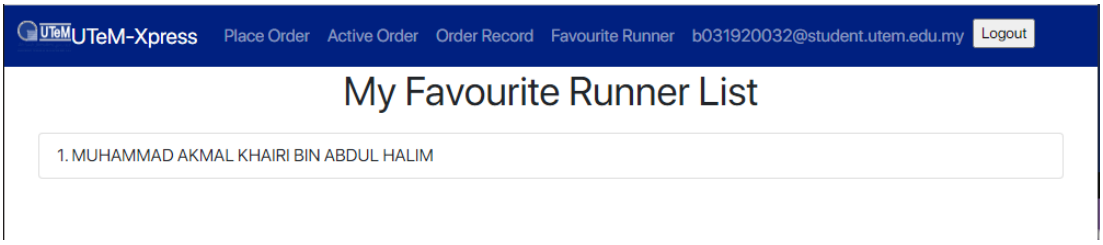
Runner Name: MUHAMMAD AKMAL BIN MOHD SABRI
Runner Telephone: 01120624714

Your Review: ok
Your Rating: ★★★★★
Runner Rating: ★★★★★

Order Date: 01/02/21
Order Time: 21:39:00

Figure 4.32: Order Details Screen

Figure 4.32 above show the order details screen. It displays the order id, type, pickup location, drop off location, selected additional services, vehicle type, order status, remarks, fee, runner’s name, runner’s telephone number, user’s review, user’s rating, runner’s rating, date, and time.

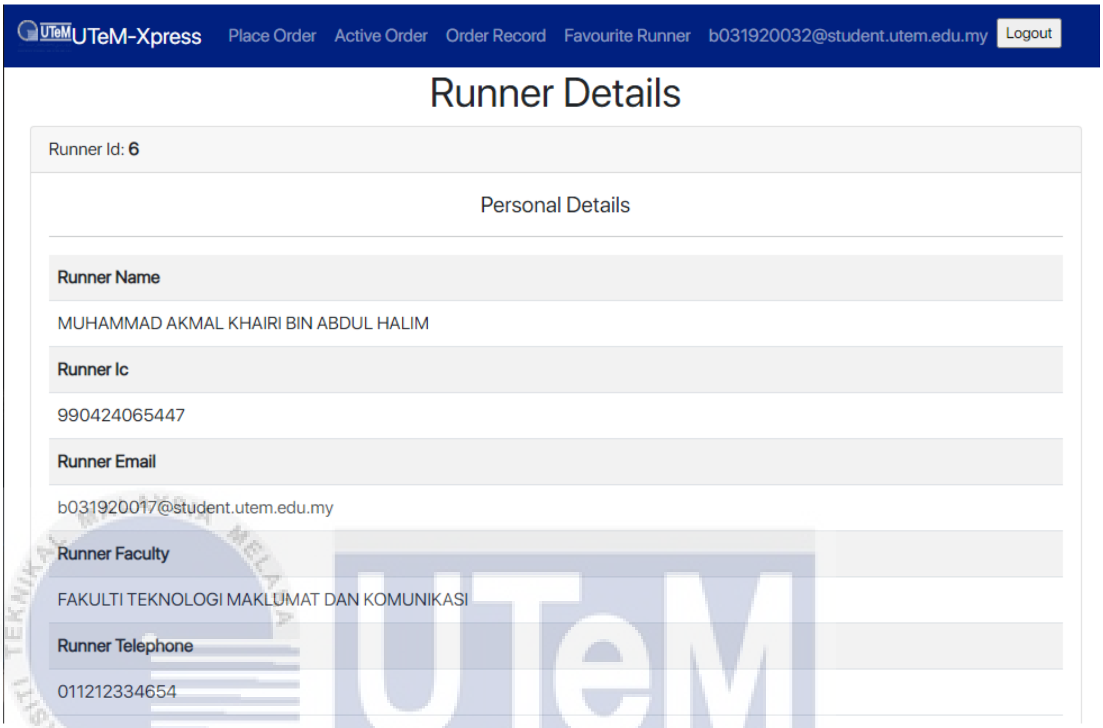


My Favourite Runner List

1. MUHAMMAD AKMAL KHAIRI BIN ABDUL HALIM

Figure 4.33: Favourite Runner List Screen

Figure 4.33 above show the favorite runner list screen, it displays list of runners that user have add to favorite. User only need to click the runner list to view the runner details.



Runner Details

Runner Id: **6**

Personal Details

Runner Name

MUHAMMAD AKMAL KHAIRI BIN ABDUL HALIM

Runner Ic

990424065447

Runner Email

b031920017@student.utem.edu.my

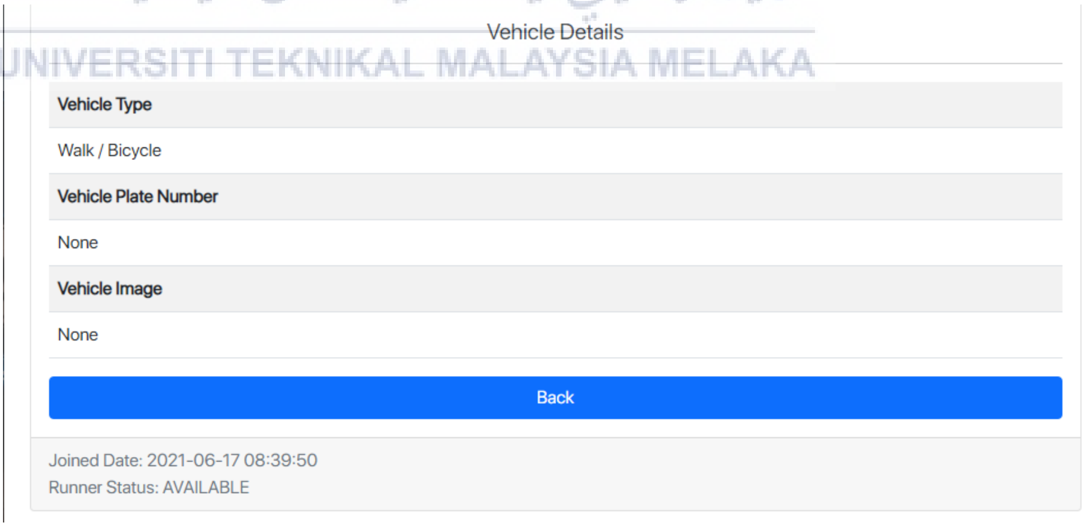
Runner Faculty

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

Runner Telephone

011212334654

Figure 4.34: Runner Details Screen



Vehicle Details

Vehicle Type

Walk / Bicycle

Vehicle Plate Number

None

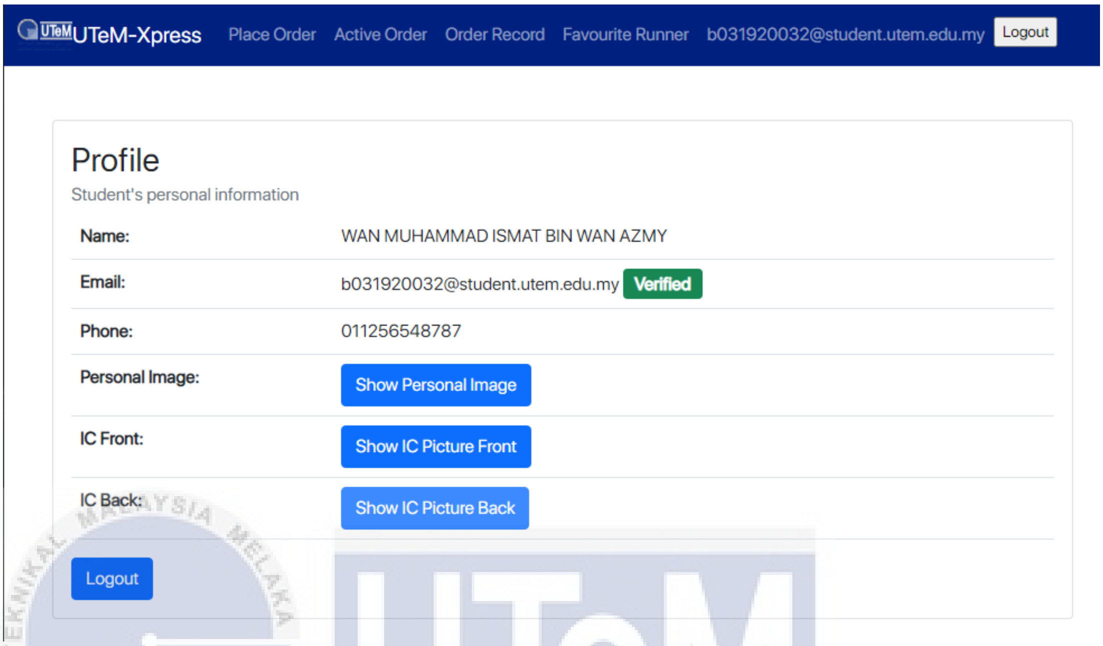
Vehicle Image

None

Joined Date: 2021-06-17 08:39:50
Runner Status: AVAILABLE

Figure 4.35: Runner Details Screen 2

Figure 4.34 & 4.35 above show the runner details screen. It displays runner information including name, identification card number, email, faculty, telephone, and vehicle details consist of type, plate number, and image.



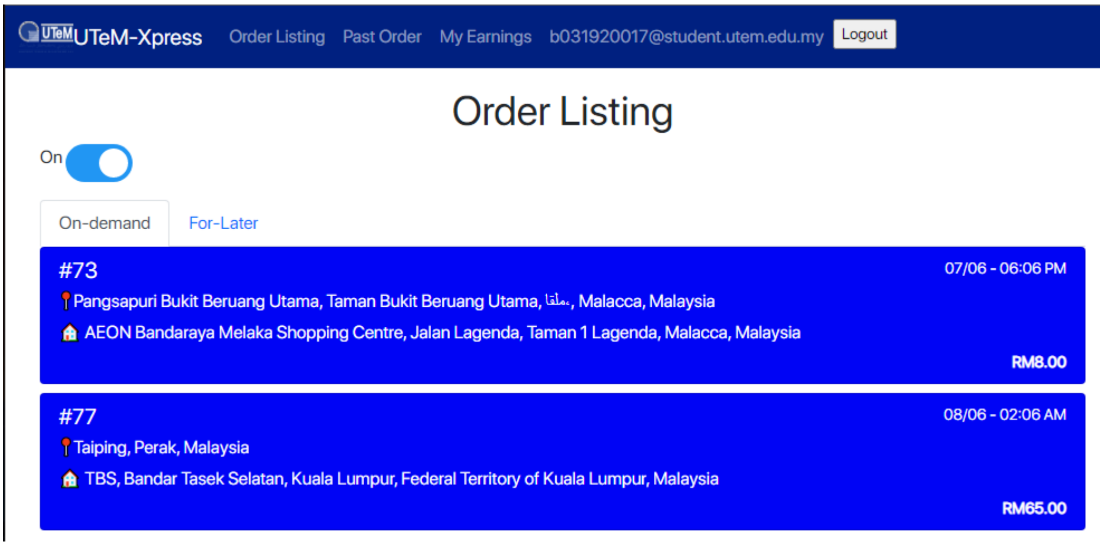
Profile

Student's personal information

Name:	WAN MUHAMMAD ISMAT BIN WAN AZMY
Email:	b031920032@student.utem.edu.my Verified
Phone:	011256548787
Personal Image:	Show Personal Image
IC Front:	Show IC Picture Front
IC Back:	Show IC Picture Back

Figure 4.36: Profile Screen

Figure 4.36 above show the profile screen for the users. It displays user's information such as name, email, phone, image, front, and back image of identification card. User only required to click the show image button to view the image.



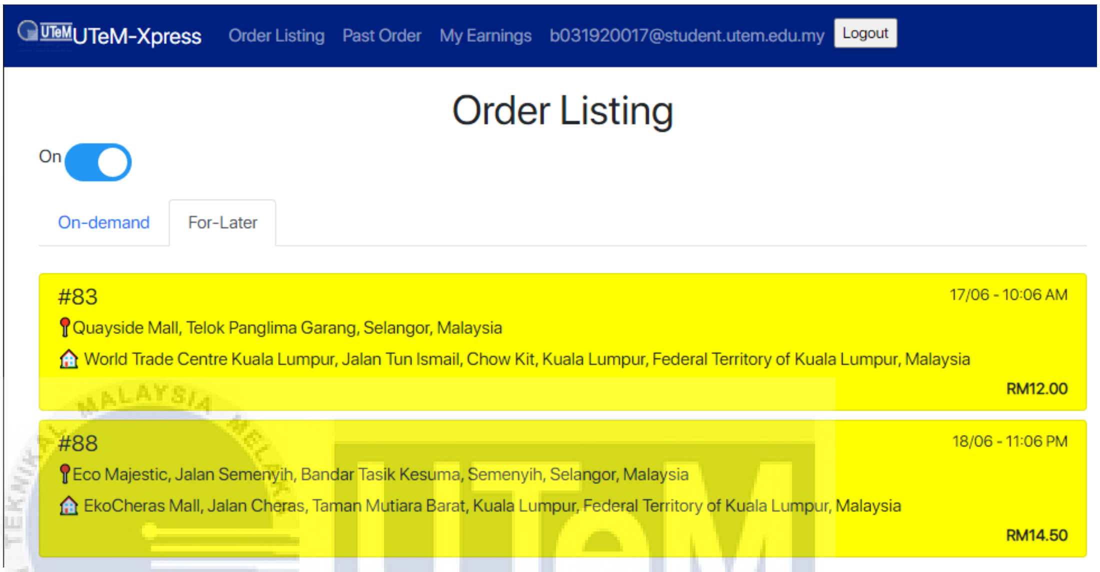
Order Listing

#73 07/06 - 06:06 PM
📍Pangsapuri Bukit Beruang Utama, Taman Bukit Beruang Utama, ملاك, Malacca, Malaysia
🏠 AEON Bandaraya Melaka Shopping Centre, Jalan Lagenda, Taman 1 Lagenda, Malacca, Malaysia
RM8.00

#77 08/06 - 02:06 AM
📍Taiping, Perak, Malaysia
🏠 TBS, Bandar Tasek Selatan, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM65.00

Figure 4.37: Order Listing Screen for Runner

Figure 4.37 above show the order listing screen for the runner. It displays all the order created by the user with the status of ‘on-demand’. This screen will only list the order that request type of vehicle based on the type of vehicle registered by the runner. If the runner register vehicle as ‘car’ then the screen will list only order that request for car as mode of transportation.



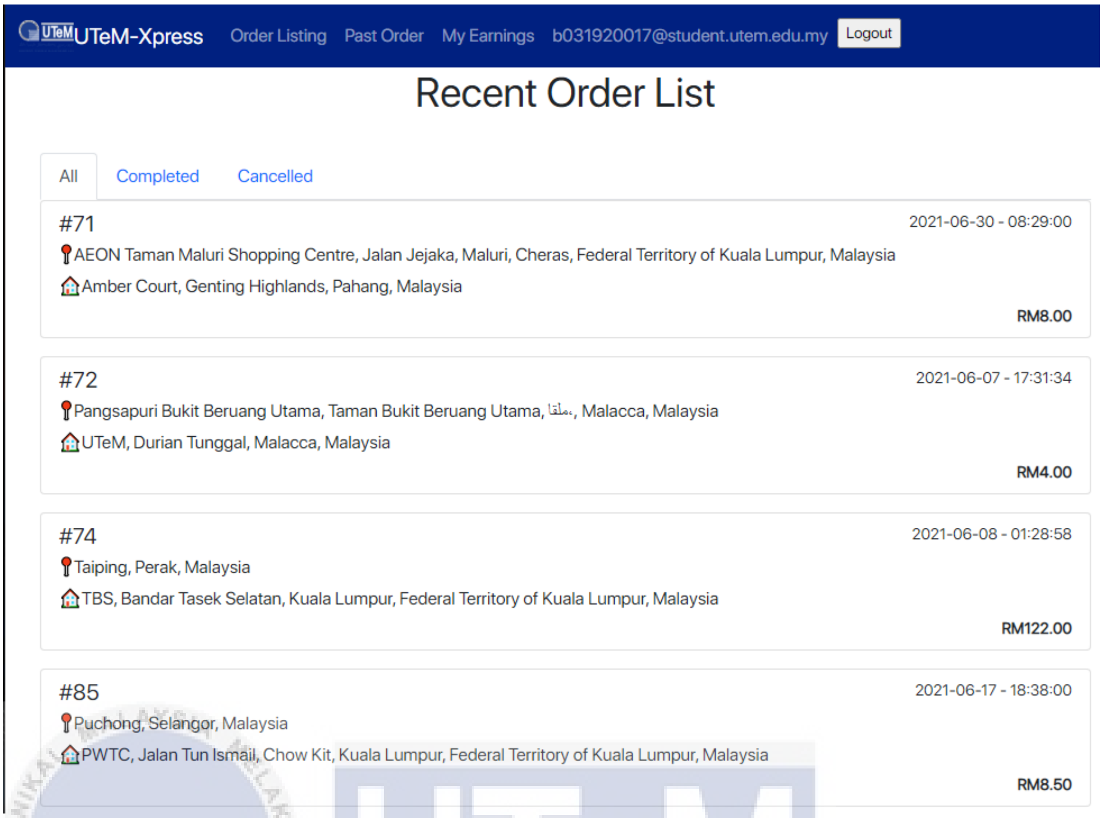
Order Listing

#83 17/06 - 10:06 AM
📍Quayside Mall, Telok Panglima Garang, Selangor, Malaysia
🏠 World Trade Centre Kuala Lumpur, Jalan Tun Ismail, Chow Kit, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM12.00

#88 18/06 - 11:06 PM
📍Eco Majestic, Jalan Semenyih, Bandar Tasik Kesuma, Semenyih, Selangor, Malaysia
🏠 EkoCheras Mall, Jalan Cheras, Taman Mutiara Barat, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM14.50

Figure 4.38: Order Listing Screen for Runner

Figure 4.38 above show the order listing screen for the runner. It displays all the order created by the user with the status of ‘for-later’.



Recent Order List

#71 2021-06-30 - 08:29:00
📍AEON Taman Maluri Shopping Centre, Jalan Jejaka, Maluri, Cheras, Federal Territory of Kuala Lumpur, Malaysia
🏠Amber Court, Genting Highlands, Pahang, Malaysia
RM8.00

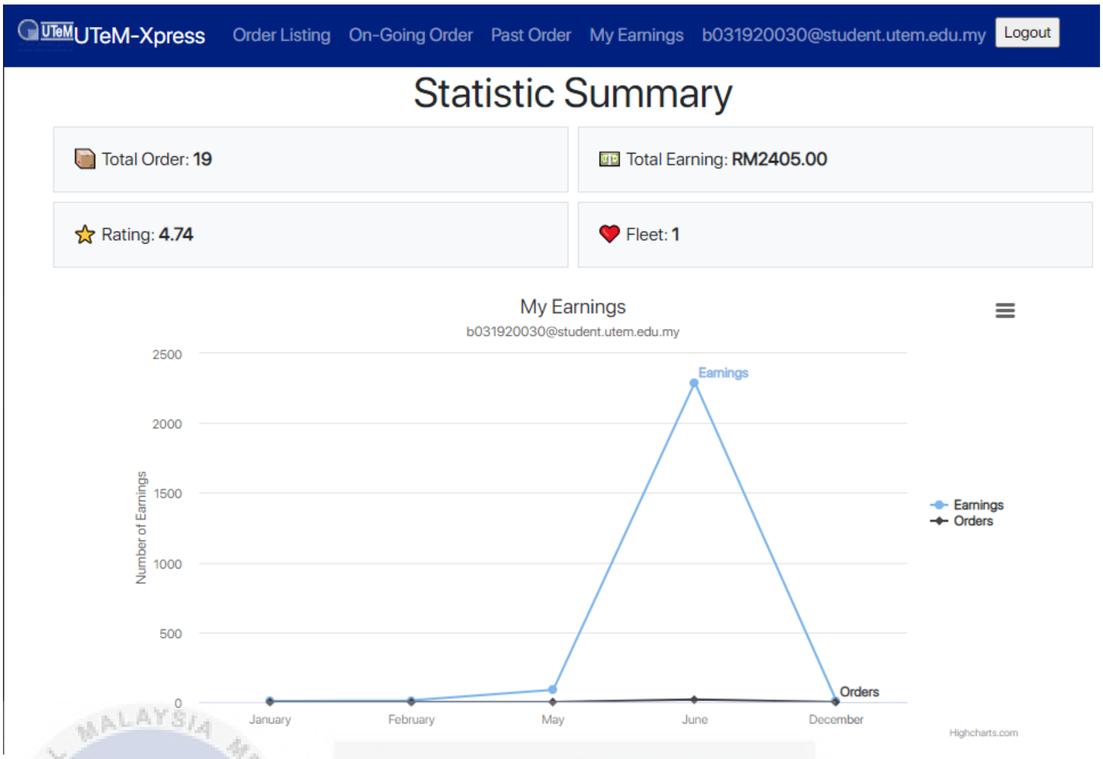
#72 2021-06-07 - 17:31:34
📍Pangsapuri Bukit Beruang Utama, Taman Bukit Beruang Utama, ملقا, Malacca, Malaysia
🏠UTeM, Durian Tunggal, Malacca, Malaysia
RM4.00

#74 2021-06-08 - 01:28:58
📍Taiping, Perak, Malaysia
🏠TBS, Bandar Tasek Selatan, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM122.00

#85 2021-06-17 - 18:38:00
📍Puchong, Selangor, Malaysia
🏠PWTC, Jalan Tun Ismail, Chow Kit, Kuala Lumpur, Federal Territory of Kuala Lumpur, Malaysia
RM8.50

Figure 4.39: Recent Order List Screen

Figure 4.39 above show the recent order list screen. It displays all the order that the runner have took and categorize it based on its status which is completed, cancelled and all. The runner only needs to click the order list to view the order details.



Statistic Summary

Total Order: **19**

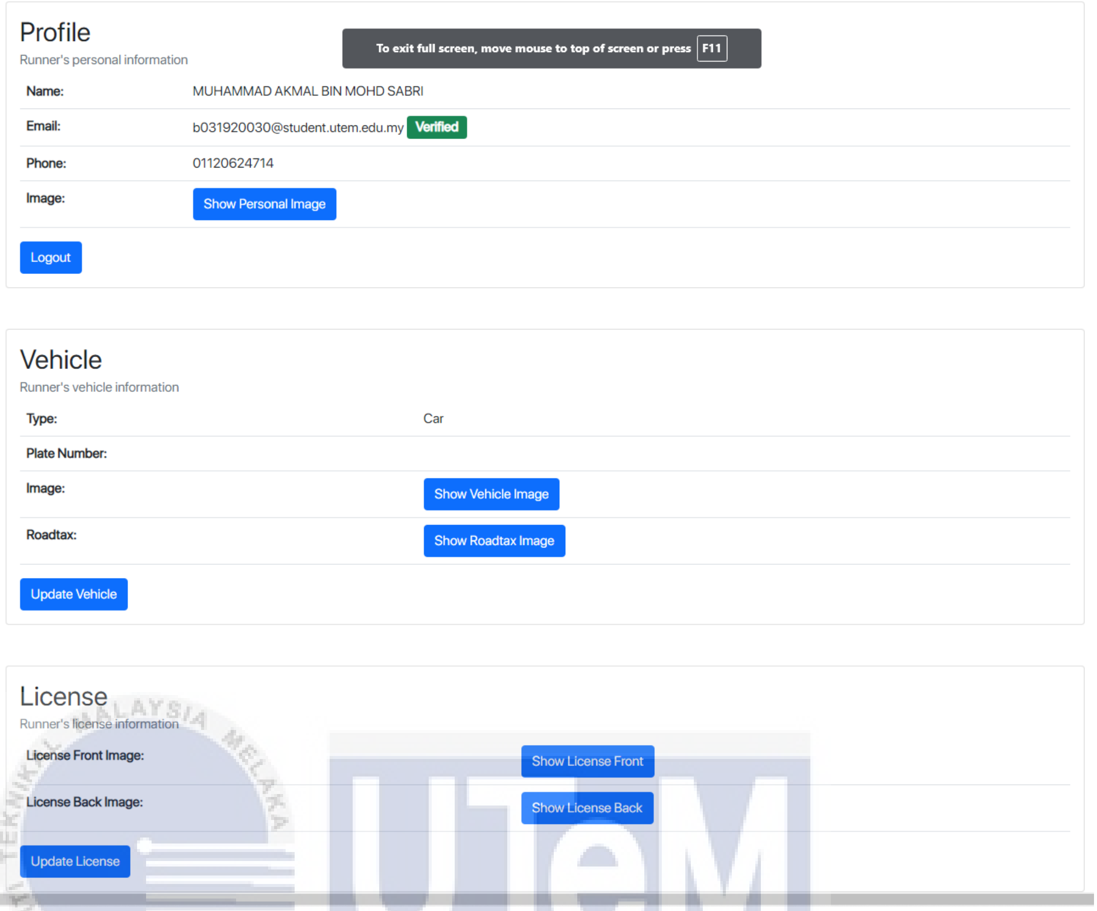
Total Earning: **RM2405.00**

Rating: **4.74**

Fleet: **1**

Figure 4.40: Earning Screen

Figure 4.40 above show the earning screen for the runner. It displays the total order taken by the runner, total earning, average rating, total favorite by users and on the below provided a graph that show number of earning and order received for each month.



Profile

Runner's personal information

Name:	MUHAMMAD AKMAL BIN MOHD SABRI
Email:	b031920030@student.utem.edu.my Verified
Phone:	01120624714
Image:	Show Personal Image

Logout

To exit full screen, move mouse to top of screen or press F11

Vehicle

Runner's vehicle information

Type:	Car
Plate Number:	
Image:	Show Vehicle Image
Roadtax:	Show Roadtax Image

Update Vehicle

License

Runner's license information

License Front Image:	Show License Front
License Back Image:	Show License Back

Update License

Figure 4.41: Profile Screen for Runner

Figure 4.41 above show the profile screen for the runner. It displays three section which is profile, vehicle, and license section. Profile sections display runner’s name, email, phone, and image. Vehicle sections display vehicle type, plate number, image, and road-tax image. License sections display front and back image of runner’s license.

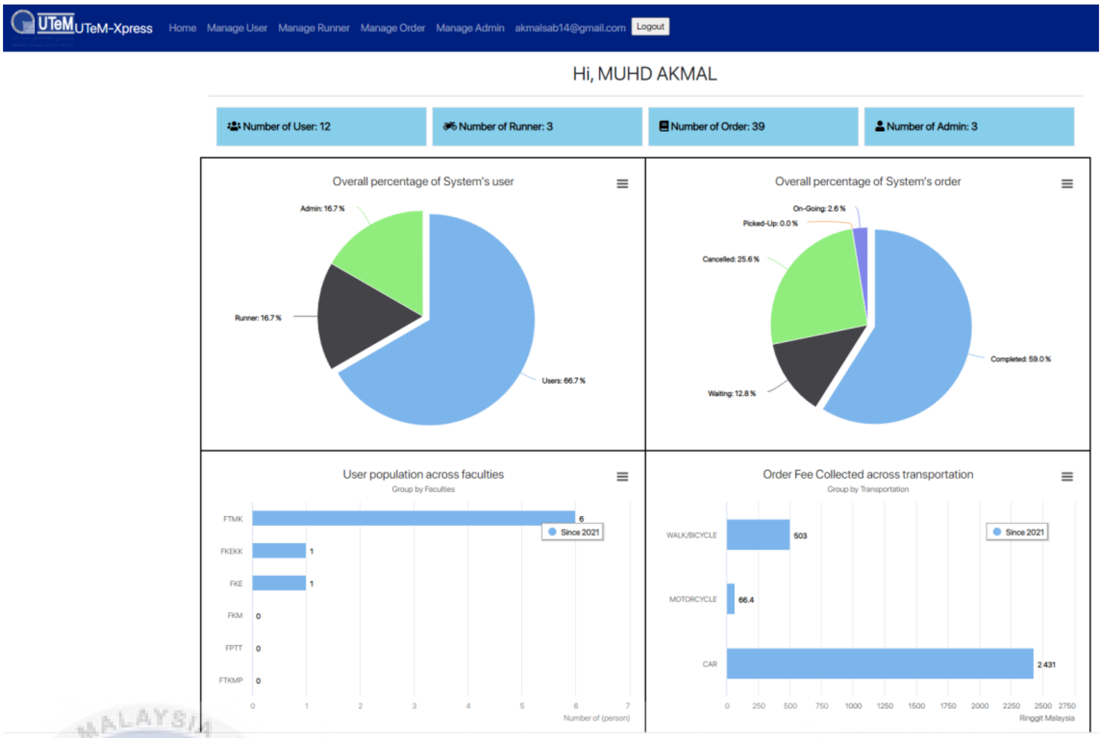


Figure 4.42: Admin Home Screen

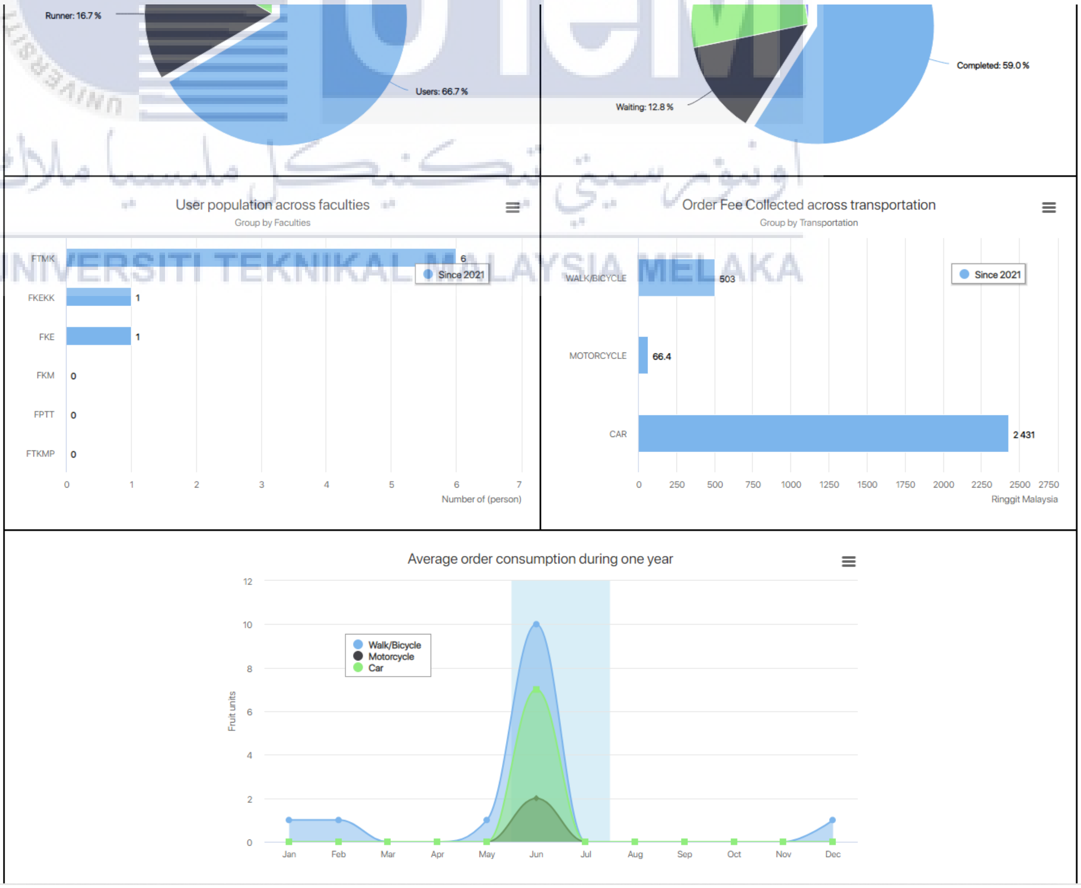
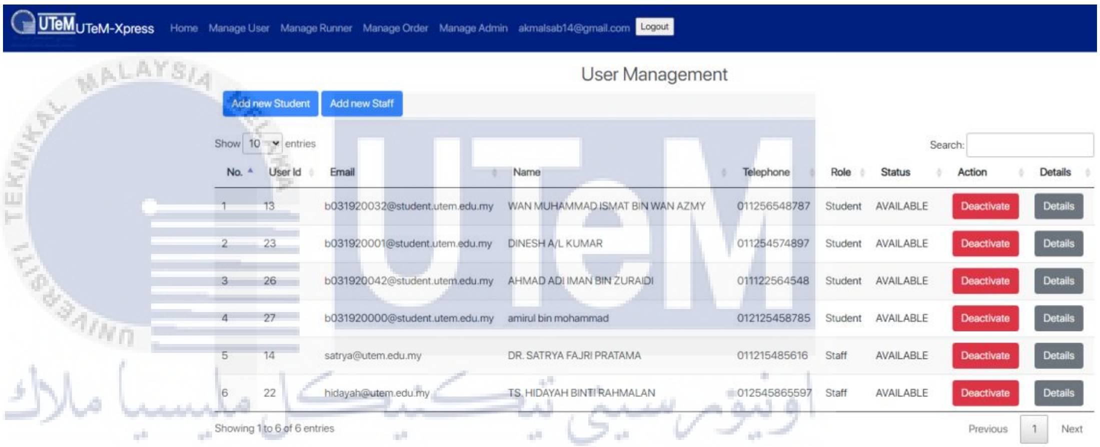


Figure 4.43: Admin Home Screen 2

Figure 4.42 & 4.43 above show the home screen for admins. It displays the statistic of the overall progress in the system. There are 6 graphs provided in the screen. The first graph displays the percentage of system's user based on user category which is user, runner, and admin. The second graph display the percentage of system's order based on order category, which is waiting, on-going, picked-up, cancelled, completed. The third graph displays the number of users across faculties. The fourth graph displays the number of amount collected based on transportation type which is walk/bicycle, motorcycle, and car. Last graphs display the number of orders created for each mode of transportation for each month.



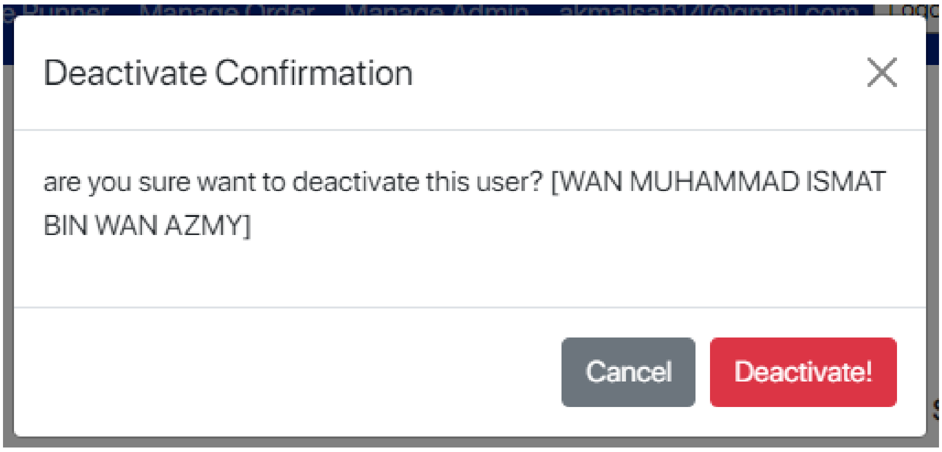
User Management

No.	User Id	Email	Name	Telephone	Role	Status	Action	Details
1	13	b031920032@student.utem.edu.my	WAN MUHAMMAD ISMAT BIN WAN AZMY	011256548787	Student	AVAILABLE	Deactivate	Details
2	23	b031920001@student.utem.edu.my	DINESH A/L KUMAR	011254574897	Student	AVAILABLE	Deactivate	Details
3	26	b031920042@student.utem.edu.my	AHMAD ADI IMAN BIN ZURAIDI	01122564548	Student	AVAILABLE	Deactivate	Details
4	27	b031920000@student.utem.edu.my	amirul bin mohammad	012125458785	Student	AVAILABLE	Deactivate	Details
5	14	satrya@utem.edu.my	DR. SATRYA FAJRI PRATAMA	011215485616	Staff	AVAILABLE	Deactivate	Details
6	22	hidayah@utem.edu.my	TS. HIDAYAH BINTI RAHMALAN	012545865597	Staff	AVAILABLE	Deactivate	Details

Showing 1 to 6 of 6 entries

Figure 4.44: User Management Screen

Figure 4.44 above show the user management screen for admin. It displays a list of users in the system and viewed in table format. Admins can deactivate users and reactivate back. Admins also can view the user's details. Admins can register new student and staff into the system. Admin can do searching for specific details and sorting based on each column value.

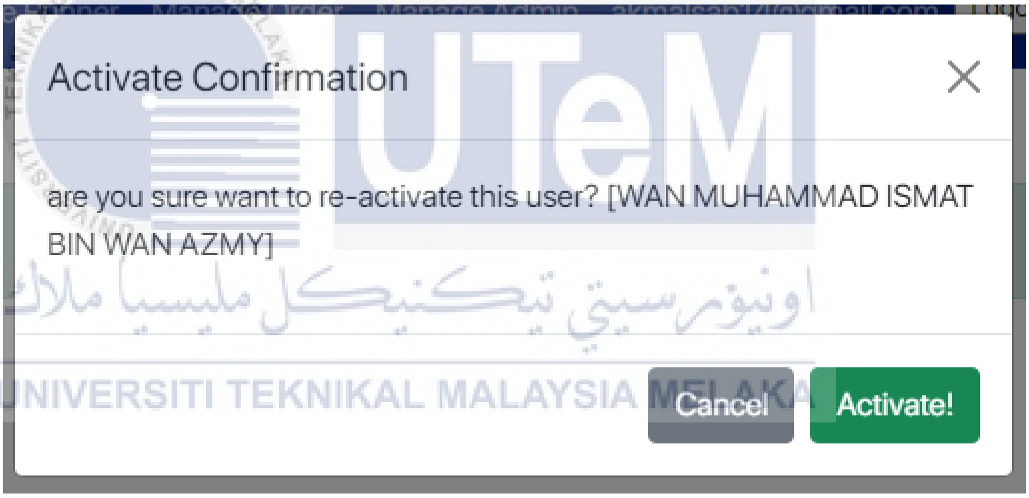


Deactivate Confirmation

are you sure want to deactivate this user? [WAN MUHAMMAD ISMAT BIN WAN AZMY]

Figure 4.45: Deactivate Confirmation Screen

Figure above 4.45 show a pop-up message screen to ask admin's confirmation before proceeding with deactivation of users.

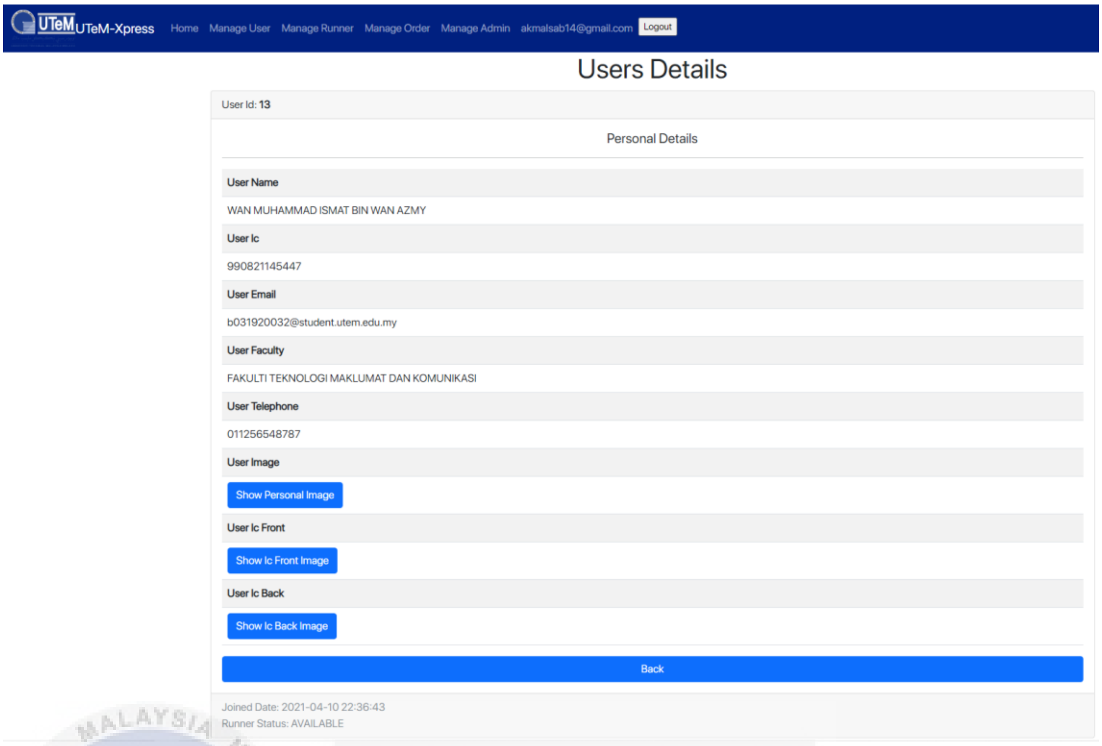


Activate Confirmation

are you sure want to re-activate this user? [WAN MUHAMMAD ISMAT BIN WAN AZMY]

Figure 4.46: Activate Confirmation Screen

Figure above 4.46 show a pop-up message screen to ask admin's confirmation before proceeding with reactivation of users.



Users Details

User Id: **13**

Personal Details

User Name

WAN MUHAMMAD ISMAT BIN WAN AZMY

User Ic

990821145447

User Email

b031920032@student.utem.edu.my

User Faculty

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

User Telephone

011256548787

User Image

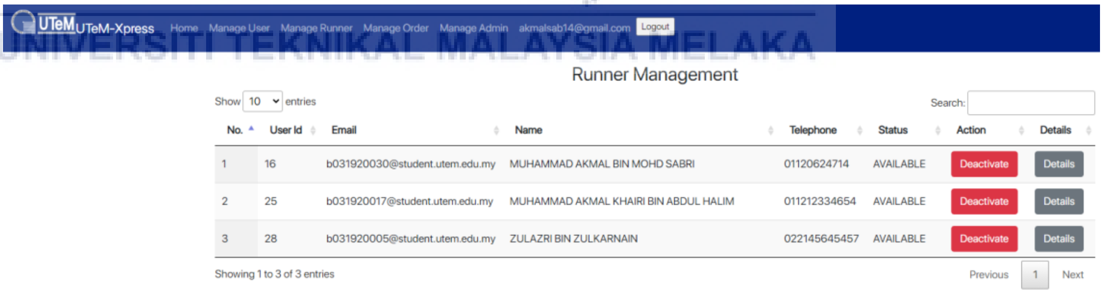
User Ic Front

User Ic Back

Joined Date: 2021-04-10 22:36:43
Runner Status: AVAILABLE

Figure 4.47: User's Detail Screen

Figure 4.47 above show user’s detail screen. It displays the information related to the selected user including user’s name, identification number, email, faculty, telephone, image, front, and back image of identification card.



Runner Management

Show 10 entries

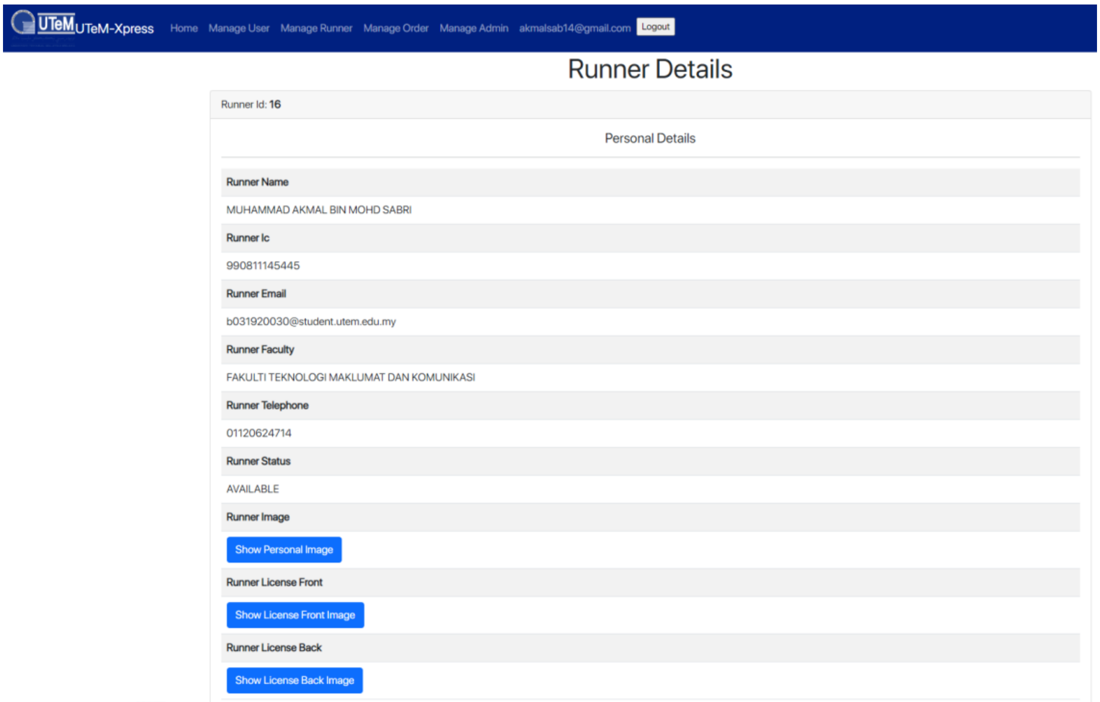
Search:

No.	User Id	Email	Name	Telephone	Status	Action	Details
1	16	b031920030@student.utem.edu.my	MUHAMMAD AKMAL BIN MOHD SABRI	01120624714	AVAILABLE	Deactivate	Details
2	25	b031920017@student.utem.edu.my	MUHAMMAD AKMAL KHAIRI BIN ABDUL HALIM	011212334654	AVAILABLE	Deactivate	Details
3	28	b031920005@student.utem.edu.my	ZULAZRI BIN ZULKARNAIN	022145645457	AVAILABLE	Deactivate	Details

Showing 1 to 3 of 3 entries

Figure 4.48: Runner Management Screen

Figure 4.48 above show the runner management screen for admin. It displays a list of runners in the system and viewed in table format. Admins can deactivate runner and reactivate back. Admins also can view the runner’s details. Admin can do searching for specific details and sorting based on each column value.



Runner Details

Runner Id: **16**

Personal Details

Runner Name

MUHAMMAD AKMAL BIN MOHD SABRI

Runner Ic

990811145445

Runner Email

b031920030@student.utem.edu.my

Runner Faculty

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

Runner Telephone

01120624714

Runner Status

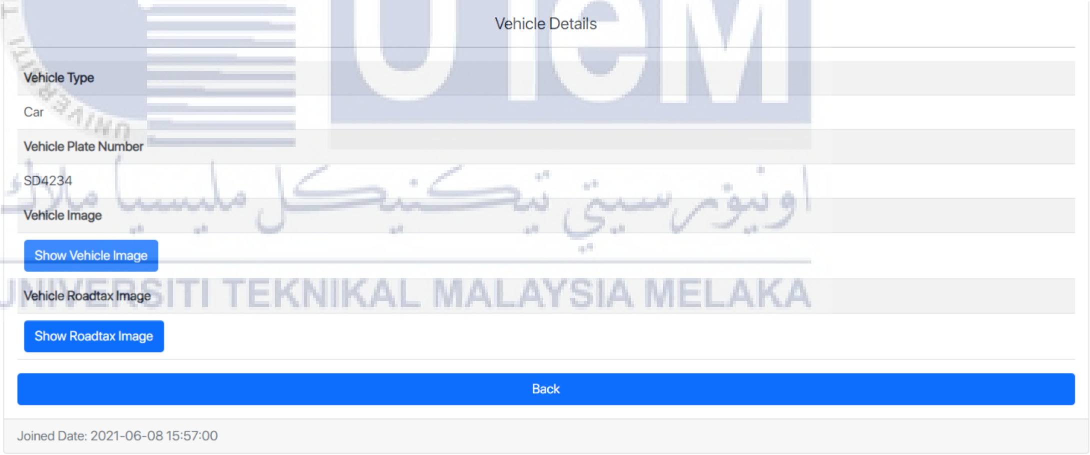
AVAILABLE

Runner Image

Runner License Front

Runner License Back

Figure 4.49: Runner Detail Screen



Vehicle Details

Vehicle Type

Car

Vehicle Plate Number

SD4234

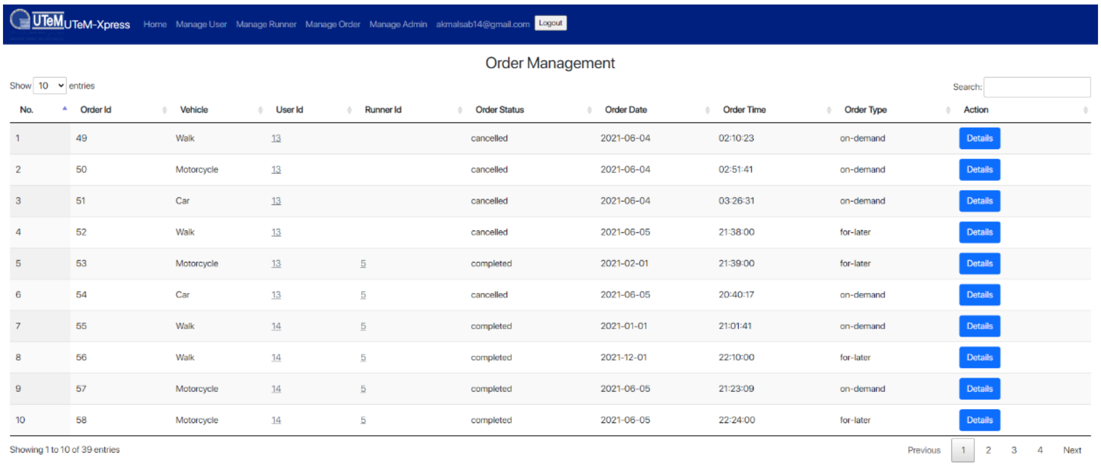
Vehicle Image

Vehicle Roadtax Image

Joined Date: 2021-06-08 15:57:00

Figure 4.50: Runner Detail Screen 2

Figure 4.49 & 4.50 above show runner’s detail screen. It displays the information related to the selected runner including runner’s name, identification number, email, faculty, telephone, status, image, front, and back image of license. In vehicle details, it displays the vehicle type, plate number, image, and road-tax image.



Order Management

Show 10 entries

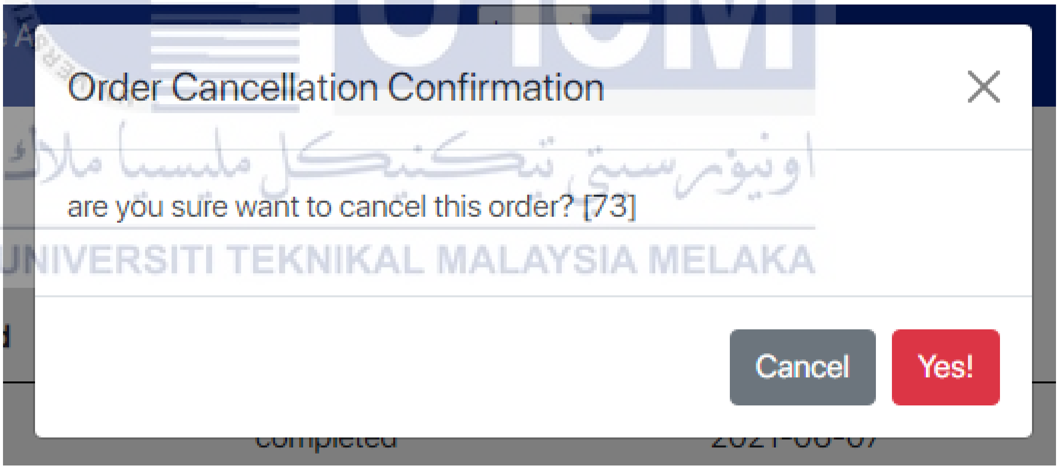
Search:

No.	Order Id	Vehicle	User Id	Runner Id	Order Status	Order Date	Order Time	Order Type	Action
1	49	Walk	13		cancelled	2021-06-04	02:10:23	on-demand	Details
2	50	Motorcycle	13		cancelled	2021-06-04	02:51:41	on-demand	Details
3	51	Car	13		cancelled	2021-06-04	03:26:31	on-demand	Details
4	52	Walk	13		cancelled	2021-06-05	21:38:00	for-later	Details
5	53	Motorcycle	13	5	completed	2021-02-01	21:39:00	for-later	Details
6	54	Car	13	5	cancelled	2021-06-05	20:40:17	on-demand	Details
7	55	Walk	14	5	completed	2021-01-01	21:01:41	on-demand	Details
8	56	Walk	14	5	completed	2021-12-01	22:10:00	for-later	Details
9	57	Motorcycle	14	5	completed	2021-06-05	21:23:09	on-demand	Details
10	58	Motorcycle	14	5	completed	2021-06-05	22:24:00	for-later	Details

Showing 1 to 10 of 39 entries

Figure 4.51: Order Management Screen

Figure 4.51 above show the order management screen for admins. It displays a list of orders in the system and viewed in table format. Admins can cancel an order. Admins also can view the order's details. Admin can do searching for specific details and sorting based on each column value.

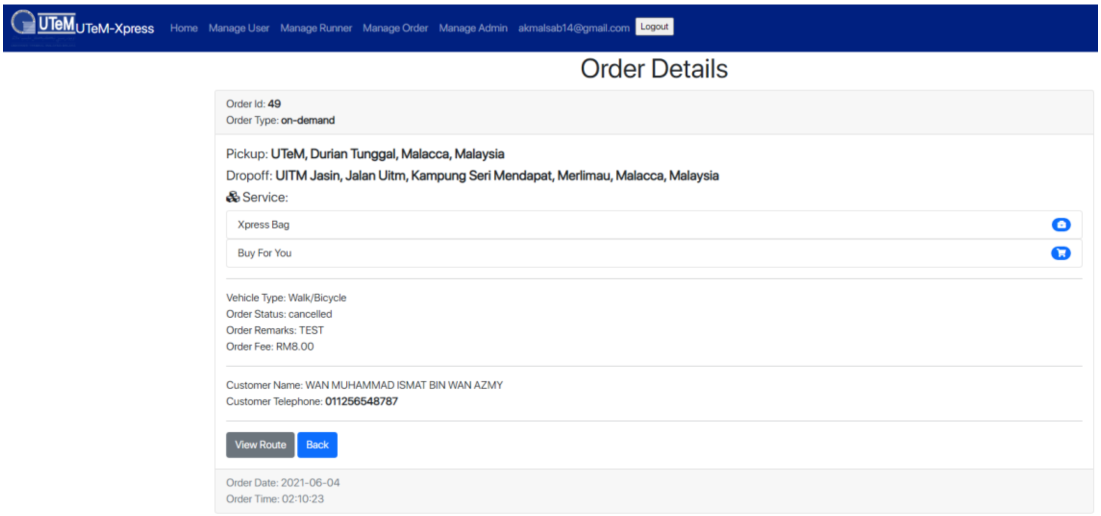


Order Cancellation Confirmation

are you sure want to cancel this order? [73]

Figure 4.52: Order Cancellation Screen

Figure above 4.52 show a pop-up message screen to ask admin's confirmation before proceeding with cancellation of order.



Order Details

Order Id: **49**
Order Type: **on-demand**

Pickup: **UTeM, Durian Tunggal, Malacca, Malaysia**
Dropoff: **UITM Jasin, Jalan Uitm, Kampung Seri Mendapat, Merlimau, Malacca, Malaysia**
Service:

Xpress Bag
Buy For You

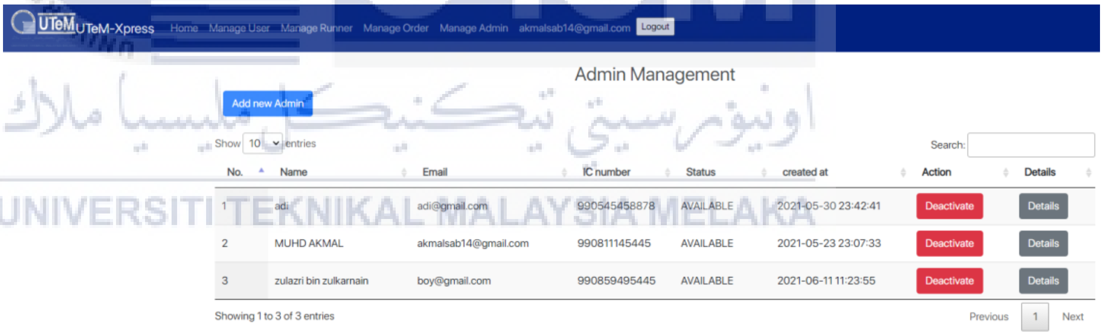
Vehicle Type: Walk/Bicycle
Order Status: cancelled
Order Remarks: TEST
Order Fee: RM8.00

Customer Name: WAN MUHAMMAD ISMAT BIN WAN AZMY
Customer Telephone: **011256548787**

Order Date: 2021-06-04
Order Time: 02:10:23

Figure 4.53: Order's Detail Screen

Figure 4.53 above show order’s detail screen. It displays the information related to the selected order including id, type, pickup location, drop off location, selected additional services, vehicle type, status, remarks, fee, customer’s name, and telephone. Admins also can view the order route.



Admin Management

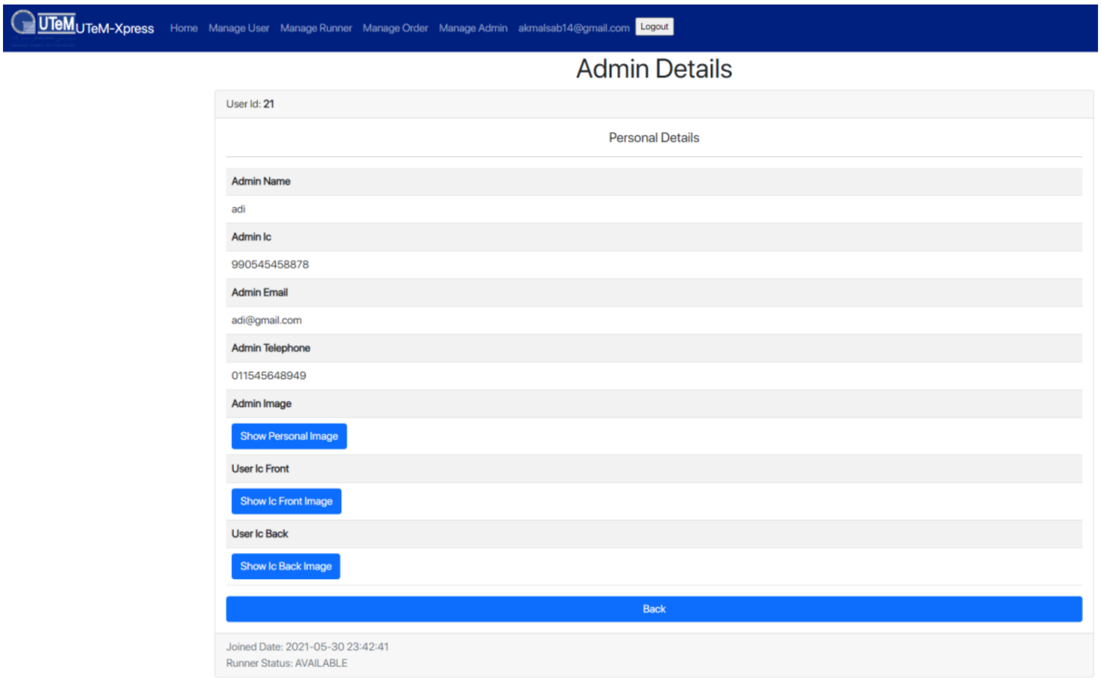
Show 10 entries

No.	Name	Email	IC number	Status	created at	Action	Details
1	adi	adi@gmail.com	990545458878	AVAILABLE	2021-05-30 23:42:41	Deactivate	Details
2	MUHD AKMAL	akmalsab14@gmail.com	990811145445	AVAILABLE	2021-05-23 23:07:33	Deactivate	Details
3	zulazri bin zulkarnain	boy@gmail.com	990859495445	AVAILABLE	2021-06-11 11:23:55	Deactivate	Details

Showing 1 to 3 of 3 entries

Figure 4.54: Admin Management Screen

Figure 4.54 above show the admin management screen for admins. It displays a list of admins in the system and viewed in table format. Only super admin can deactivate an admin. Admins also can view the other admin’s details. Admin can do searching for specific details and sorting based on each column value.



Admin Details

User Id: **21**

Personal Details

Admin Name

adi

Admin Ic

990545458878

Admin Email

adi@gmail.com

Admin Telephone

011545648949

Admin Image

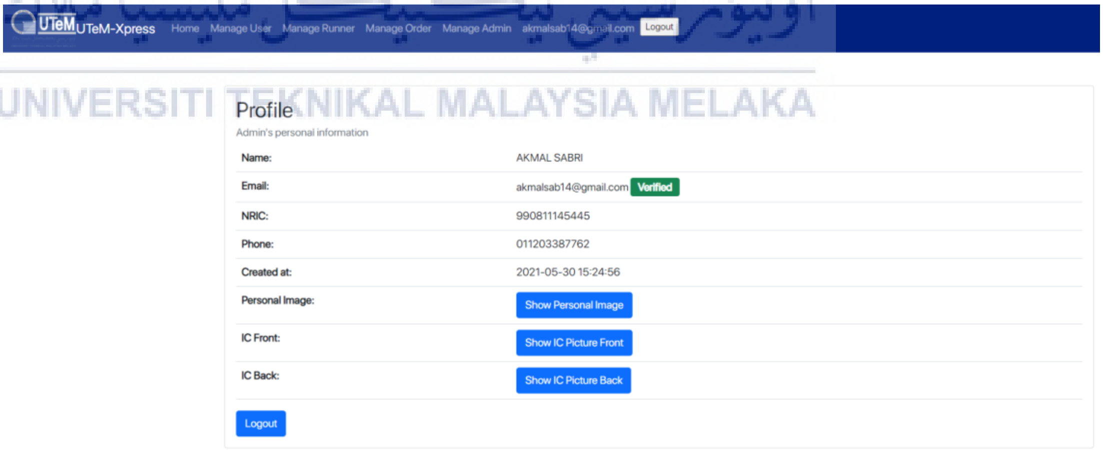
User Ic Front

User Ic Back

Joined Date: 2021-05-30 23:42:41
Runner Status: AVAILABLE

Figure 4.55: Admin's Detail Screen

Figure 4.55 above show admin’s detail screen. It displays the information related to the selected admin including user’s name, identification number, email, telephone, image, front, and back image of identification card.



Profile

Admin's personal information

Name:	AKMAL SABRI
Email:	akmalsab14@gmail.com Verified
NRIC:	990811145445
Phone:	011203387762
Created at:	2021-05-30 15:24:56
Personal Image:	Show Personal Image
IC Front:	Show IC Picture Front
IC Back:	Show IC Picture Back

Figure 4.56: Admin's Profile Screen

Figure 4.56 above show admin’s profile screen. It displays the information related to the admin including user’s name, email, identification number, telephone, creation date, image, front, and back image of identification card.

4.2.3 Database Design

This section contains the conceptual and logical database designs of UTeM Xpress system.

4.2.3.1 Conceptual and Logical Database Design

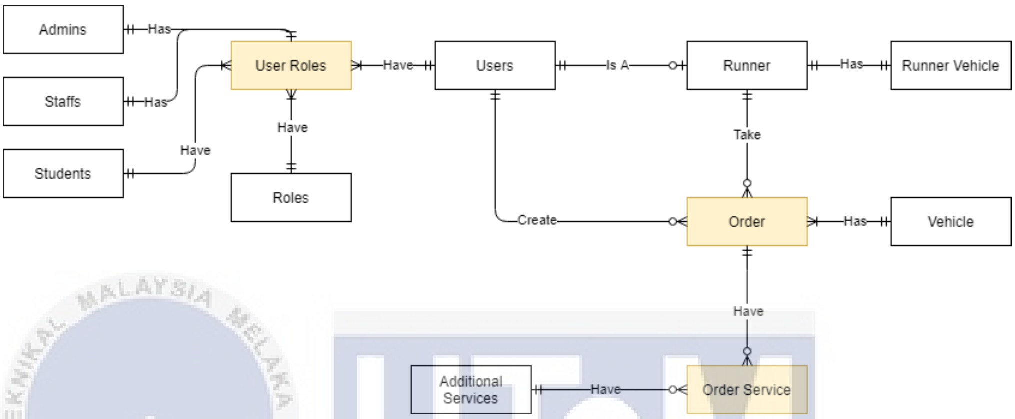


Figure 4.57: Conceptual Database Design

Conceptual Database Design is displayed in the figure 4.57 above. The database of UTeM Xpress system consist of 12 entities. They are admins, students, staff, roles, user roles, users, runner, runner vehicle, order, vehicle, order service, additional services. The description of each entity is explained below.

Admins entity contains the information related to admin’s personal information such as email, identification number, name, and status.

Staffs entity contains the information related to staff’s personal information such as email, identification number, name, faculty, division, designation and status.

Students entity contains the information related to student’s personal information such as email, identification number, name, faculty, and status.

Roles entity contain the information of system’s role exist in the system such as user, runner, and admin.

User Roles entity contain the information type of roles that the user has registered.

Users entity contain the information of users such as email and password for credential, name, telephone number, personal image, identification card image, status.

Runner entity contain the information of users that registered as runner, this entity store information like license image.

Runner Vehicle entity contain the information of vehicle for each runner such as vehicle type, image, plate number image, road tax image.

Order entity contain the information related to the orders created by the users. The information stored was date, time, pickup and drop off location, selected additional services, customer's name, runner's name, fee, vehicle type, status, remarks, customer's review, customer's rating, and runner's rating.

Vehicle entity contain the information of limit capacity for weight and size in acceptable range for each vehicle type.

Order Services entity contain the information of additional services that are bind to each order.

Additional Services entity contain the information of each available additional services offered by the system. This information such as id and name of services.

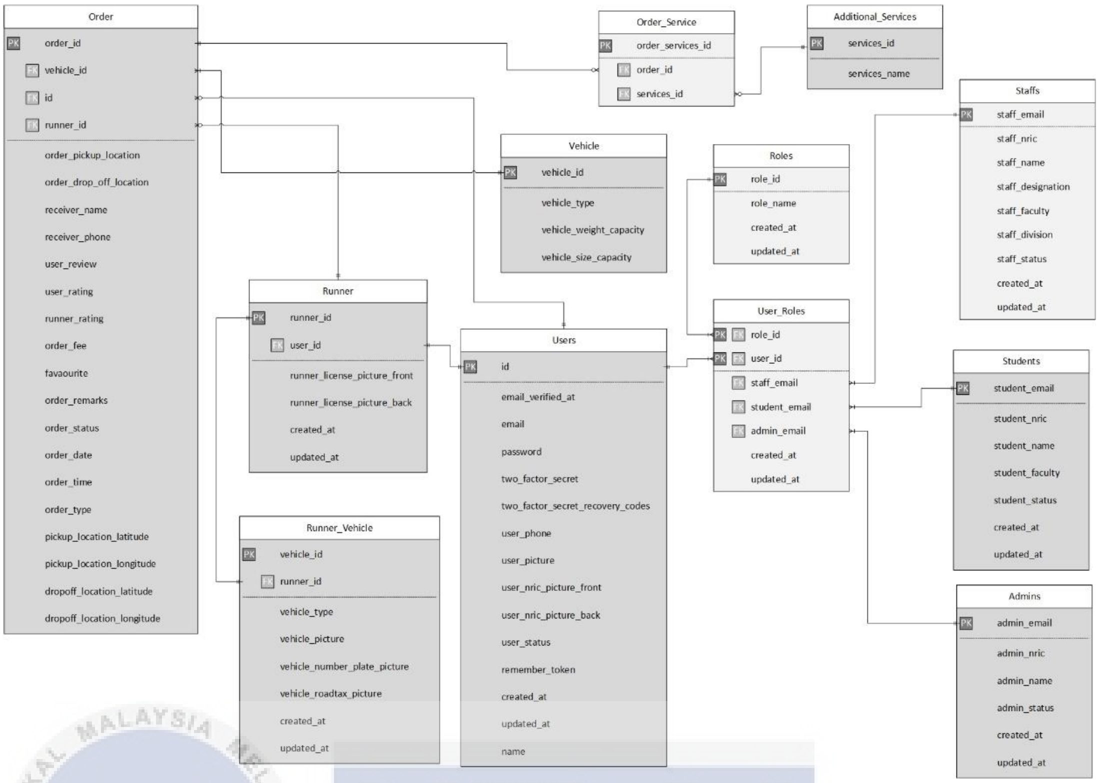


Figure 4.58: Logical Database Design

Figure 4.58 above show the logical database design for UTeM Xpress system. Consist of 12 tables and the business rule are explained as below:

Business rule:

1. Admin has only one user roles.
2. Each user roles belongs to one and only admin.
3. Students may have many users' roles.
4. Each user roles belongs to one and only student.
5. Staff has only one user roles.
6. Each user roles belongs to one and only staff.
7. Each user roles belongs to one and only role.
8. A role belongs to many user roles.
9. Each user roles belongs to one and only users.
10. A users may have one and many user roles.
11. A users may or may not become a runner.
12. Each runner belongs to one and only user.

13. Each runner has one and only runner vehicle.
14. Each runner vehicle belongs to one and only runner.
15. User may or may not create one or many orders.
16. Each order belongs to one and only user.
17. Runner may or may not take one or many orders.
18. Each order belongs to one and only runner.
19. Each order has one and only vehicle.
20. Each vehicle belongs to one or many orders.
21. Each order may or may not have zero or many order services.
22. Each order services belong to one and only order.
23. An additional service may or may not belongs to order service.
24. Each order service belongs to one and only additional service.

4.3 Detailed Design

This section elaborates project detailed designs, including physical database design and software design.

4.3.1 Software Design

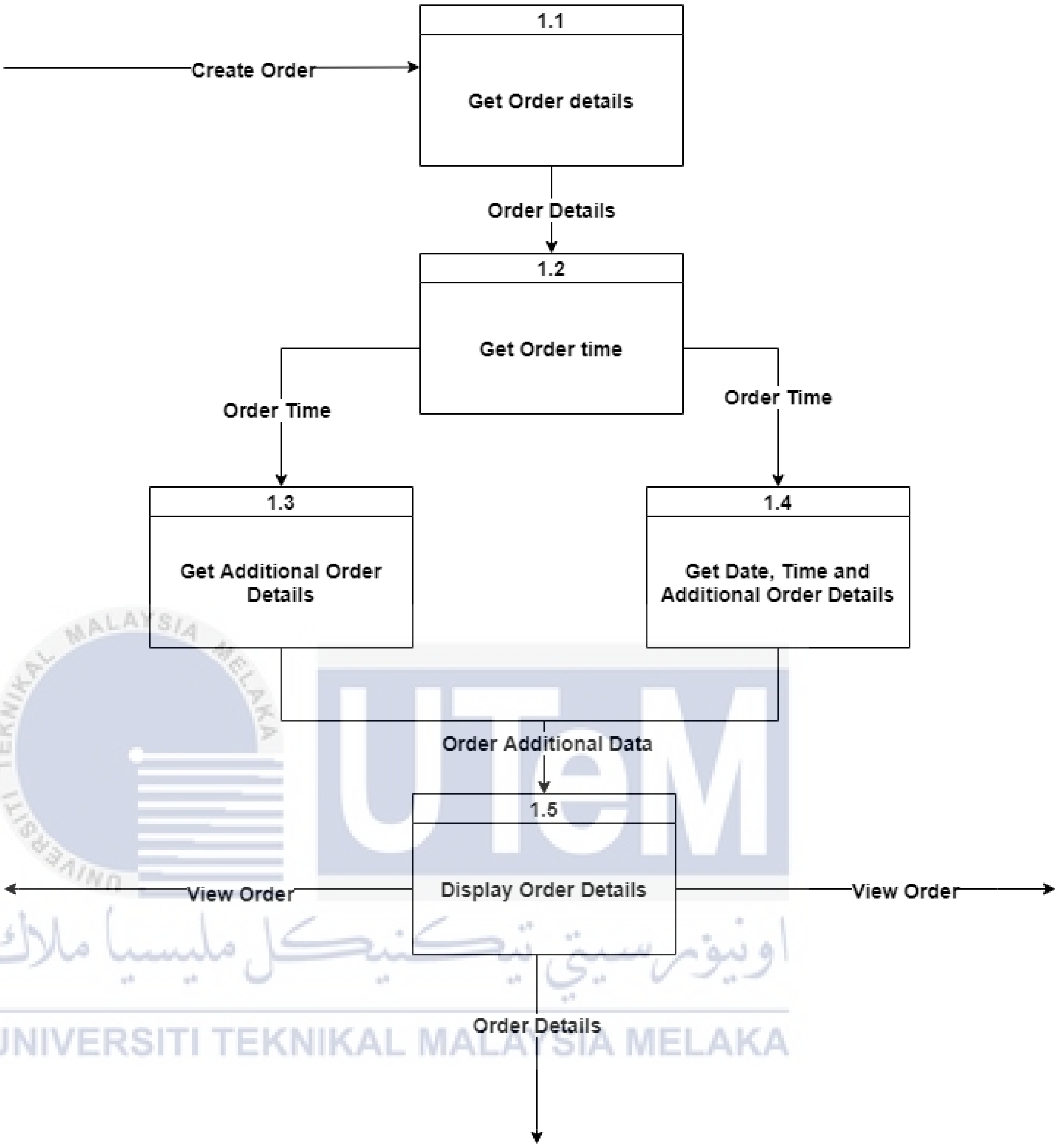


Figure 4.59: Process 1.0 Level 1

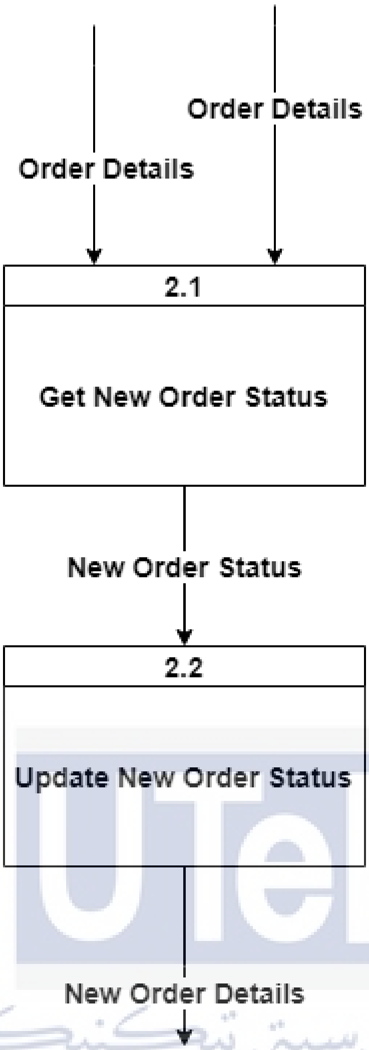


Figure 4.60: Process 2.0 Level 1

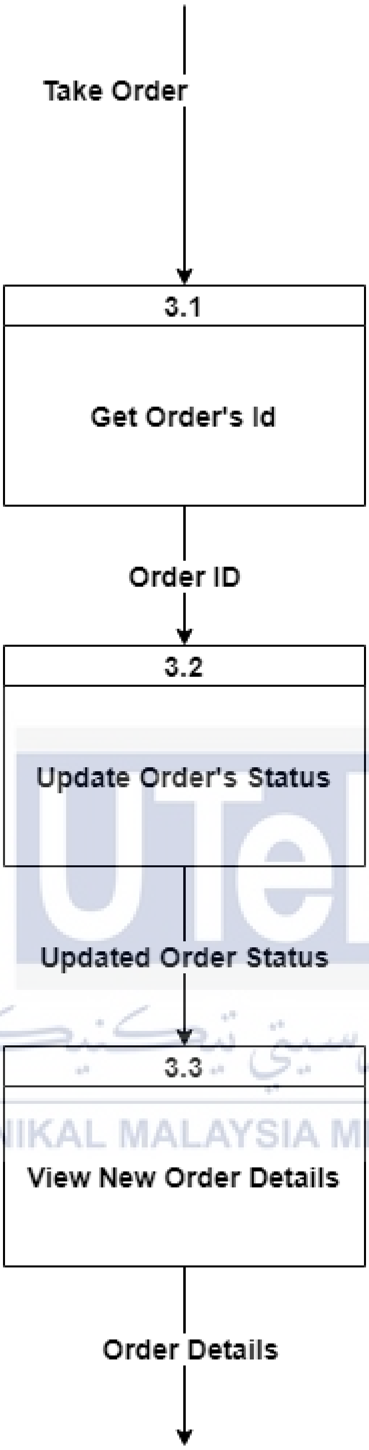


Figure 4.61: Process 3.0 Level 1

4.3.2 Physical Design

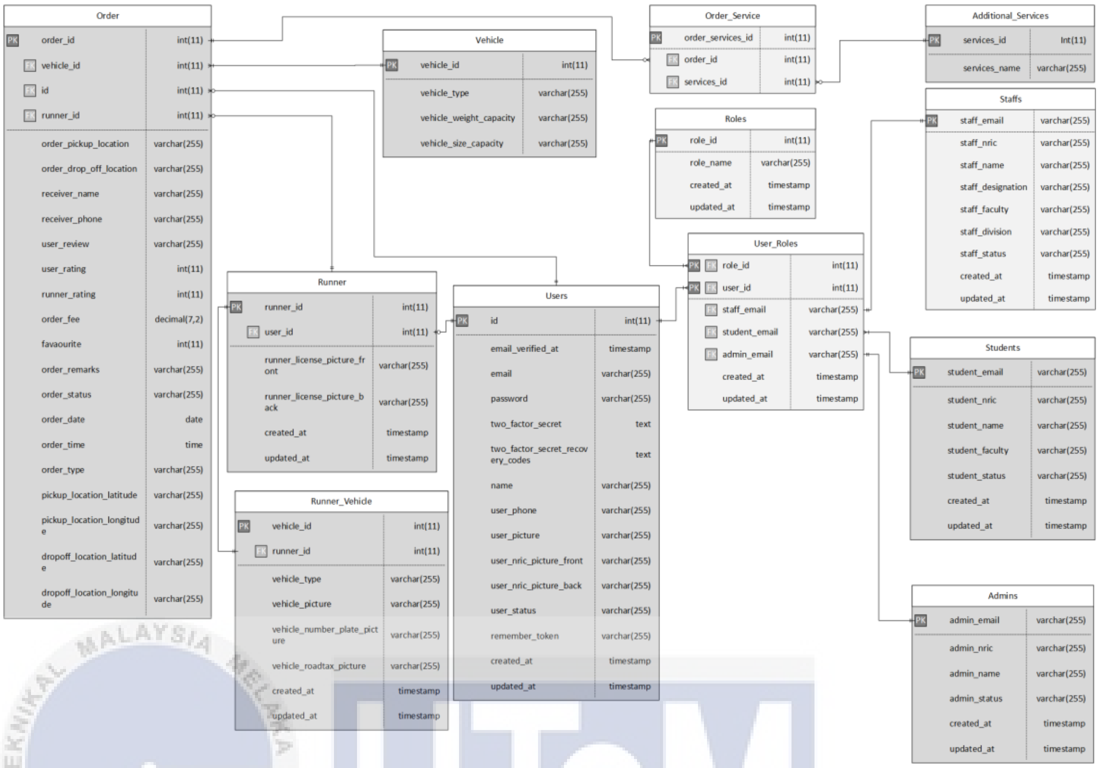

Order
PK order_id int(11)
FK vehicle_id int(11)
FK id int(11)
FK runner_id int(11)
order_pickup_location varchar(255)
order_drop_off_location varchar(255)
receiver_name varchar(255)
receiver_phone varchar(255)
user_review varchar(255)
user_rating int(11)
runner_rating int(11)
order_fee decimal(7,2)
favaourite int(11)
order_remarks varchar(255)
order_status varchar(255)
order_date date
order_time time
order_type varchar(255)
pickup_location_latitude varchar(255)
pickup_location_longitude varchar(255)
dropoff_location_latitude varchar(255)
dropoff_location_longitude varchar(255)
Vehicle
PK vehicle_id int(11)
vehicle_type varchar(255)
vehicle_weight_capacity varchar(255)
vehicle_size_capacity varchar(255)
Order_Service
PK order_services_id int(11)
FK order_id int(11)
FK services_id int(11)
Additional_Services
PK services_id Int(11)
services_name varchar(255)
Roles
PK role_id int(11)
role_name varchar(255)
created_at timestamp
updated_at timestamp
Staffs
PK staff_email varchar(255)
staff_nric varchar(255)
staff_name varchar(255)
staff_designation varchar(255)
staff_faculty varchar(255)
staff_division varchar(255)
staff_status varchar(255)
created_at timestamp
updated_at timestamp
User_Roles
PK FK role_id int(11)
PK FK user_id int(11)
FK staff_email varchar(255)
FK student_email varchar(255)
FK admin_email varchar(255)
created_at timestamp
updated_at timestamp
Runner
PK runner_id int(11)
FK user_id int(11)
runner_license_picture_front varchar(255)
runner_license_picture_back varchar(255)
created_at timestamp
updated_at timestamp
Users
PK id int(11)
email_verified_at timestamp
email varchar(255)
password varchar(255)
two_factor_secret text
two_factor_secret_recovery_codes text
name varchar(255)
user_phone varchar(255)
user_picture varchar(255)
user_nric_picture_front varchar(255)
user_nric_picture_back varchar(255)
user_status varchar(255)
remember_token varchar(255)
created_at timestamp
updated_at timestamp
Students
PK student_email varchar(255)
student_nric varchar(255)
student_name varchar(255)
student_faculty varchar(255)
student_status varchar(255)
created_at timestamp
updated_at timestamp
Runner_Vehicle
PK vehicle_id int(11)
FK runner_id int(11)
vehicle_type varchar(255)
vehicle_picture varchar(255)
vehicle_number_plate_picture varchar(255)
vehicle_roadtax_picture varchar(255)
created_at timestamp
updated_at timestamp
Admins
PK admin_email varchar(255)
admin_nric varchar(255)
admin_name varchar(255)
admin_status varchar(255)
created_at timestamp
updated_at timestamp

Figure 4.62: Physical Database Design

Physical Database Design is shown in figure 4.62 displays the data type of all columns in each entity. There are several data types used in constructing the MySQL database, which is consist of integer, varchar, decimal, text, date, time, timestamp.

4.4 Conclusion

This chapter explains and describes several designs used to aid the development of this system. All the diagrams are the artifacts of the design stage in the Spiral model. The design diagrams which including system architecture design, user interface design, database design and software design are the blueprint for the next stage of the project, implementation phase. Implementation phase transfers and converts the data shown on the design works into a usable component, module, or prototype.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter discusses about the implementation of the UTeM Xpress. It covers the software development environment setup, software configuration management and configuration environment setup.

5.2 Software Development Environment Setup

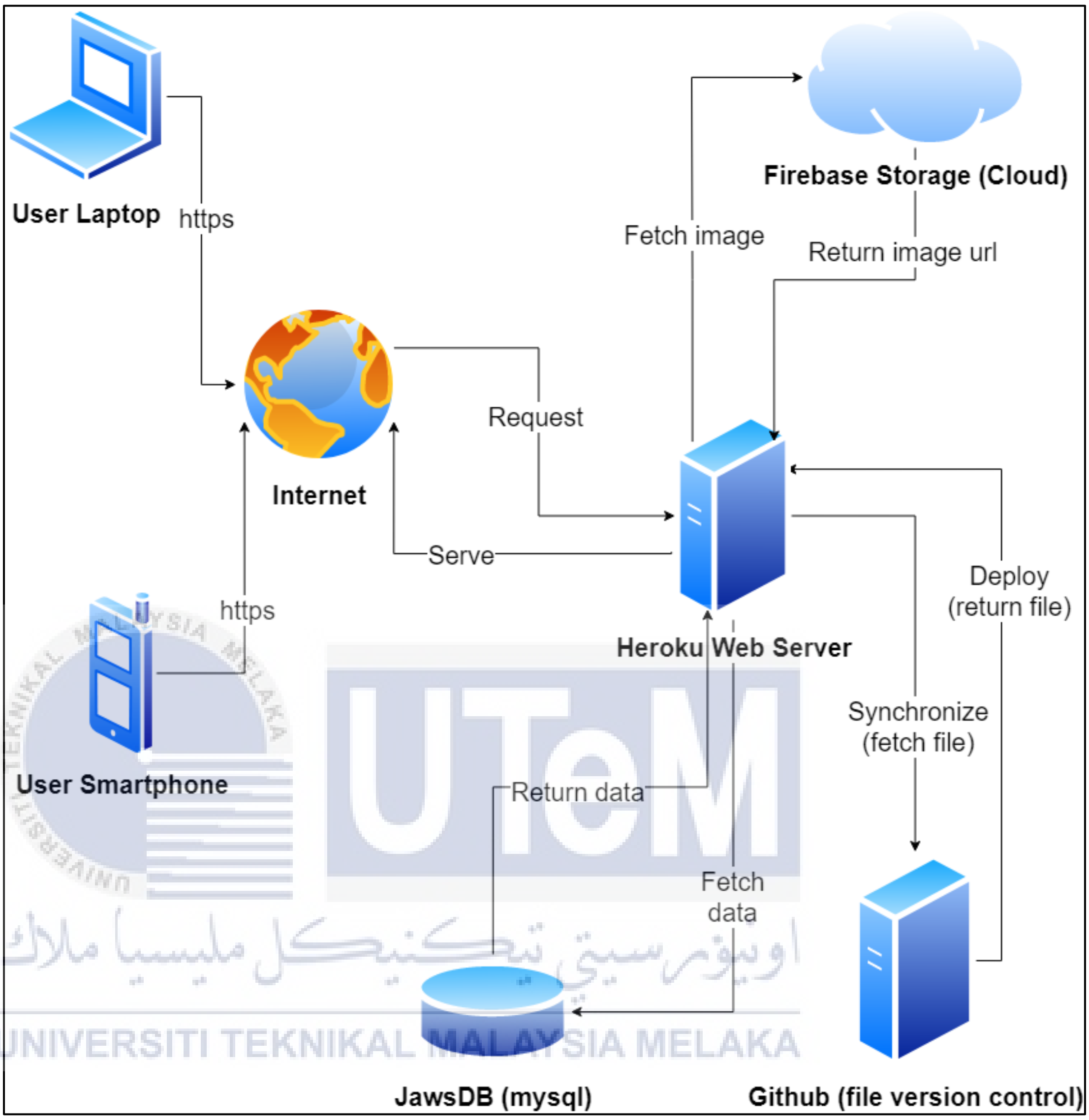


Figure 5.1: Deployment Diagram

Figure 5.1 is the deployment diagram of the UTeM Xpress system. Based on the diagram above, the system can be accessed by user's personal devices like mobile and laptop. When the user goes online and search the URL of the system on the browser, then it will send request to the server, then it will return the information back to the user so they can display the system in their personal device. The system was deployed on Heroku which is a cloud platform as a service. All the file systems were stored in the GitHub to manage file versioning control. Every change made to each file will be synchronized again by Heroku for each push command made by the developer. This will be making sure that only the latest file systems were deployed on the server.

For the database part, Jaws DB were used as the MySQL cloud database to store and persist the crucial information required by the system. For storage, Firebase storage is used as cloud storage to store the image uploaded by the user.

Table 5.1: Configuration Software

No	Properties	Value
Heroku (Server)		
1.	Website URL	https://utemxpress.herokuapp.com/
JawsDB (MySQL)		
2.	Host	ohunm00fjsjs1uzy.cbetxkdyhwsb.us-east-1.rds.amazonaws.com
3.	Username	rv4sqct3kuhcm7mf
4.	Password	xj7bv3vsydrjqsra
5.	Port	3306
6.	Database	ch4m20xn8k35dd32
GitHub		
7.	Repositories URL	https://github.com/AkmalSab/UtemXpress.git
Firebase Storage		
8.		https://console.firebase.google.com/project/utemxpress/storage/utemxpress.appspot.com/files

5.3 Software Configuration Management

This section explains the configuration environment setup and version control procedures.

5.3.1 Configuration Environment Setup

The configuration environment of this system requires Heroku as cloud service, JawsDB as a cloud database and Firebase Storage as a cloud storage. The following part covers the setup process of each service required to make this system live on the internet.

5.3.1.1 Setup process of Heroku

Step 1: Go to https://www.heroku.com/ and sign up for an account.

Step 2: Click New button and click 'create new app'

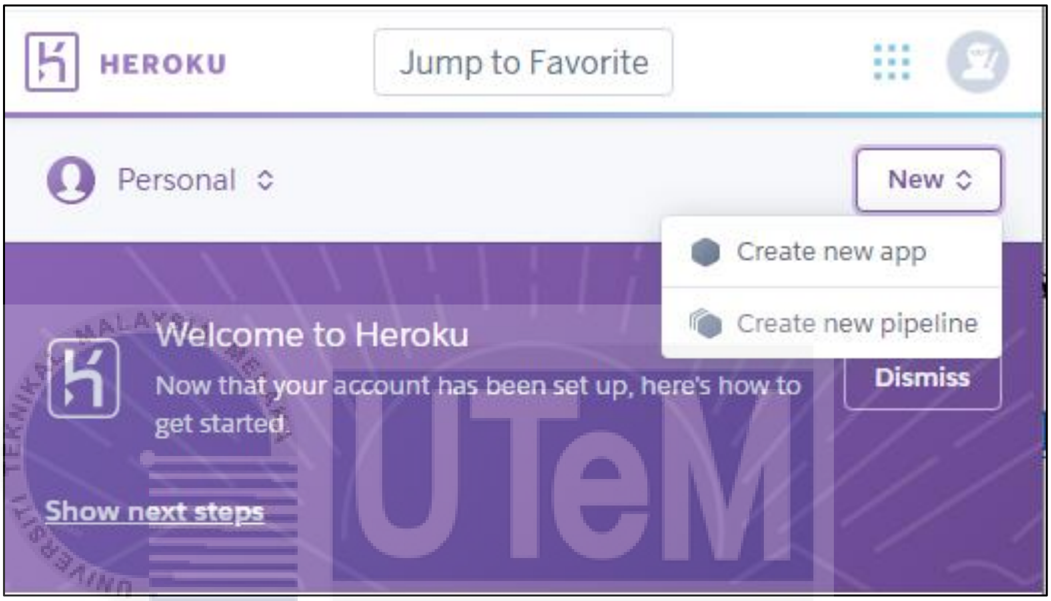
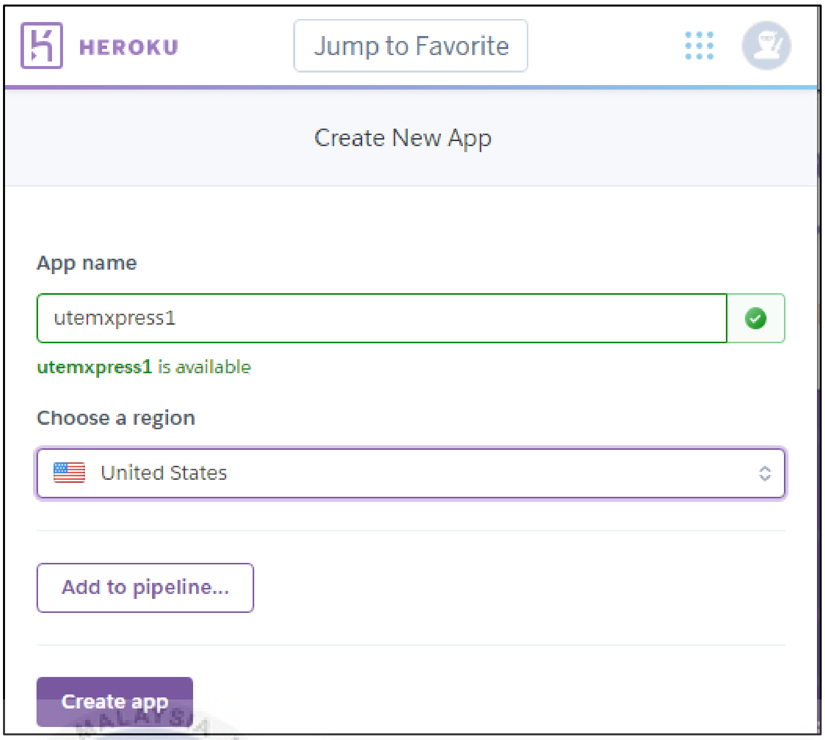


Figure 5.2: Heroku Website

Step 3: Enter the app name and choose region.



Create New App

App name

utemxpress1

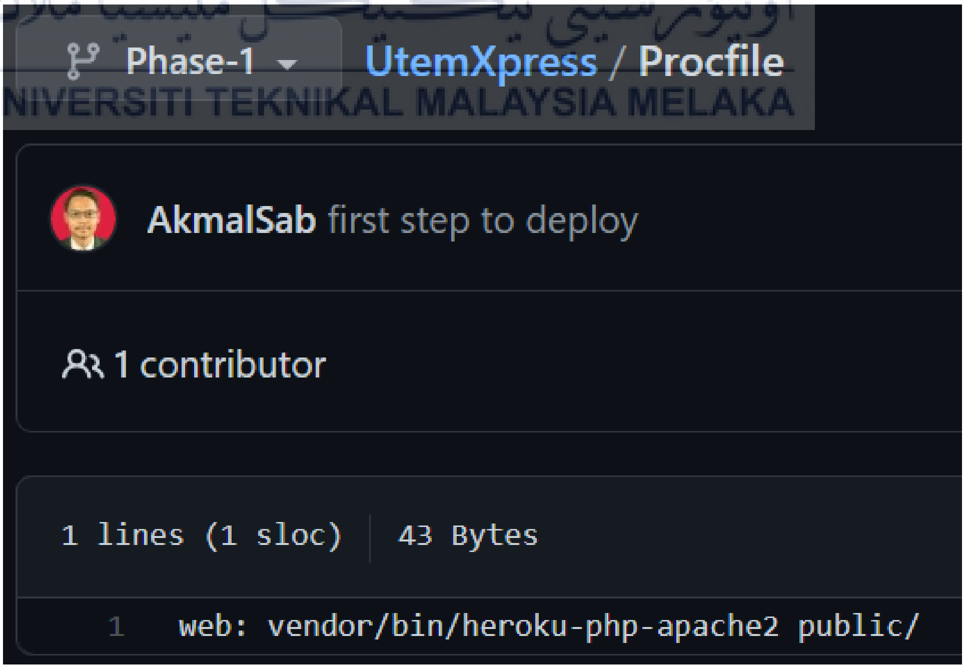
utemxpress1 is available

Choose a region

United States

Figure 5.3: Heroku Create New App Form

Step 4: In your project directory, create a new file name it 'Procfile' and save it without any file extension type. Commit this changes and push to your repositories.



Phase-1 UtemXpress / Procfile

AkmalSab first step to deploy

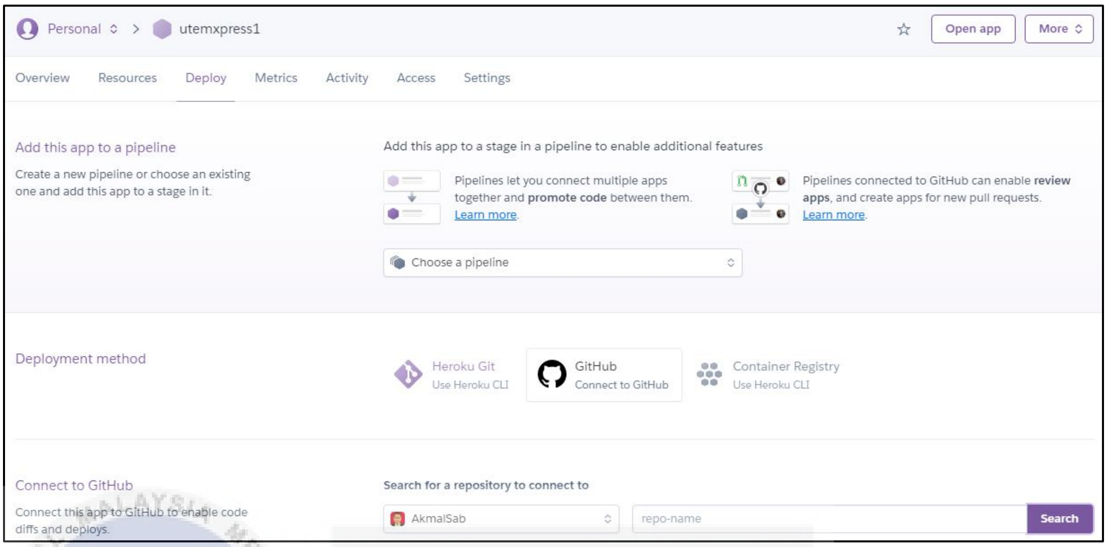
1 contributor

1 lines (1 sloc) | 43 Bytes

```
web: vendor/bin/heroku-php-apache2 public/
```

Figure 5.4: Procfile File

Step 5: Go to ‘Deploy’ tab, in the ‘Deployment method’ section, click GitHub then authorize Heroku to connect your GitHub account. After that, choose your project directories.



Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features

Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

Deployment method

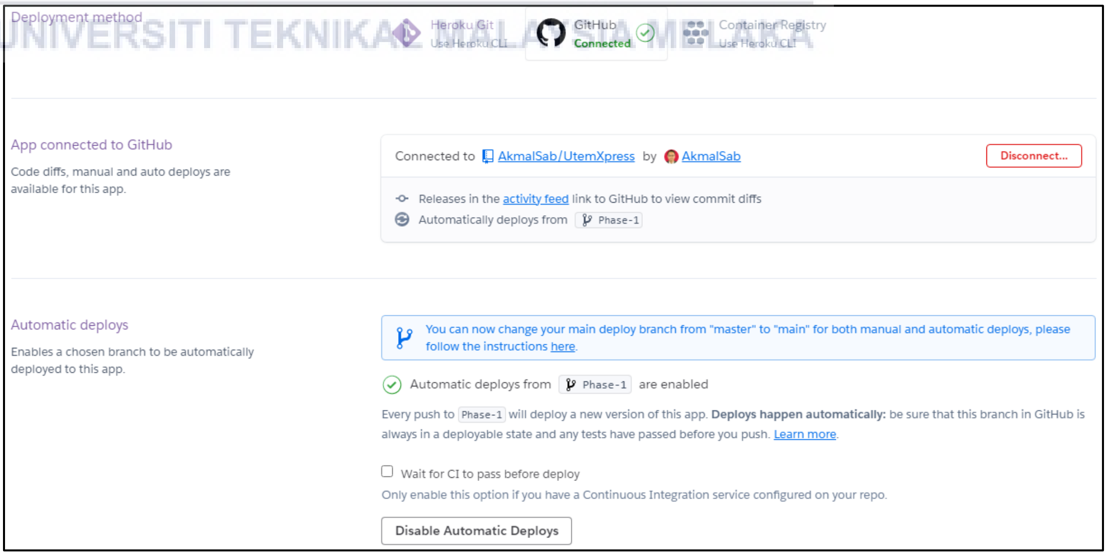
Connect to GitHub

Connect this app to GitHub to enable code diffs and deploys.

Search for a repository to connect to

Figure 5.5: Heroku Deploy Page

Step 6: Once everything is complete, your ‘deploy’ tab should be look like this.



Deployment method

App connected to GitHub

Code diffs, manual and auto deploys are available for this app.

Connected to AkmalSab/UtemXpress by AkmalSab

Releases in the activity feed link to GitHub to view commit diffs

Automatically deploys from Phase-1

Automatic deploys

Enables a chosen branch to be automatically deployed to this app.

You can now change your main deploy branch from "master" to "main" for both manual and automatic deploys, please follow the instructions here.

✓ Automatic deploys from Phase-1 are enabled

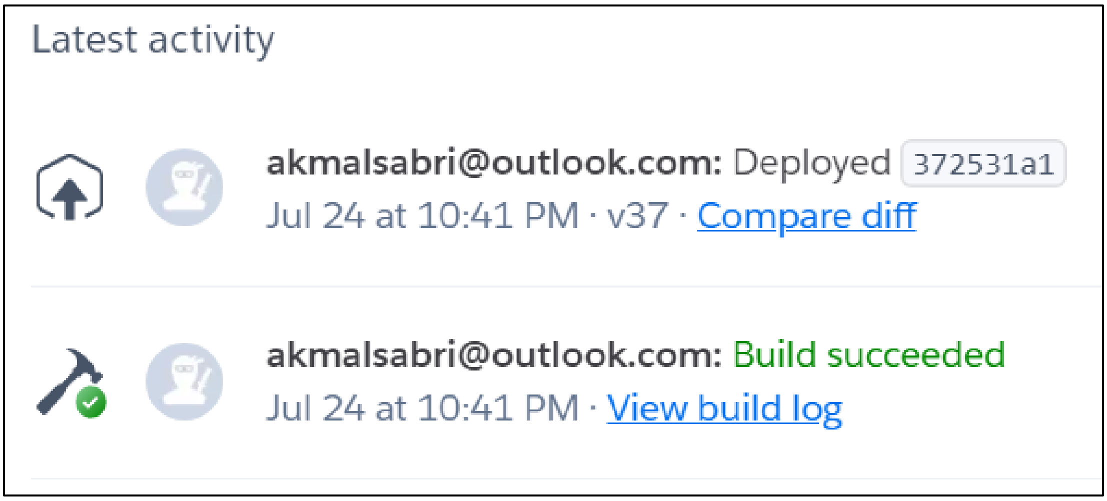
Every push to Phase-1 will deploy a new version of this app. **Deploys happen automatically:** be sure that this branch in GitHub is always in a deployable state and any tests have passed before you push. Learn more.

□ Wait for CI to pass before deploy

Only enable this option if you have a Continuous Integration service configured on your repo.

Figure 5.6: Fetch from GitHub Repositories

Step 7: Every changes you push to your GitHub repositories. Heroku will synchronize fetching all the latest file and rebuild all the project and re-deployed again.



Latest activity

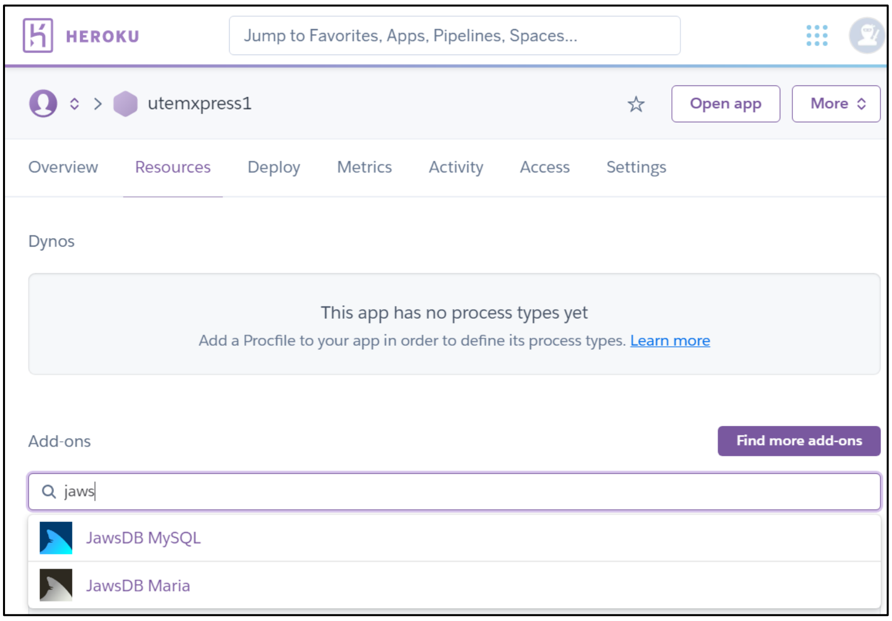
akmalsabri@outlook.com: Deployed 372531a1
Jul 24 at 10:41 PM · v37 · Compare diff

akmalsabri@outlook.com: Build succeeded
Jul 24 at 10:41 PM · View build log

Figure 5.7: Deploy Complete

5.3.1.2 Setup process of JawsDB

Step 1: Go to ‘Resources’ tab and type ‘JawsDB’ in the input text under the ‘Add-ons’ section. Click the ‘JawsDB MySQL’ option.



Dynos

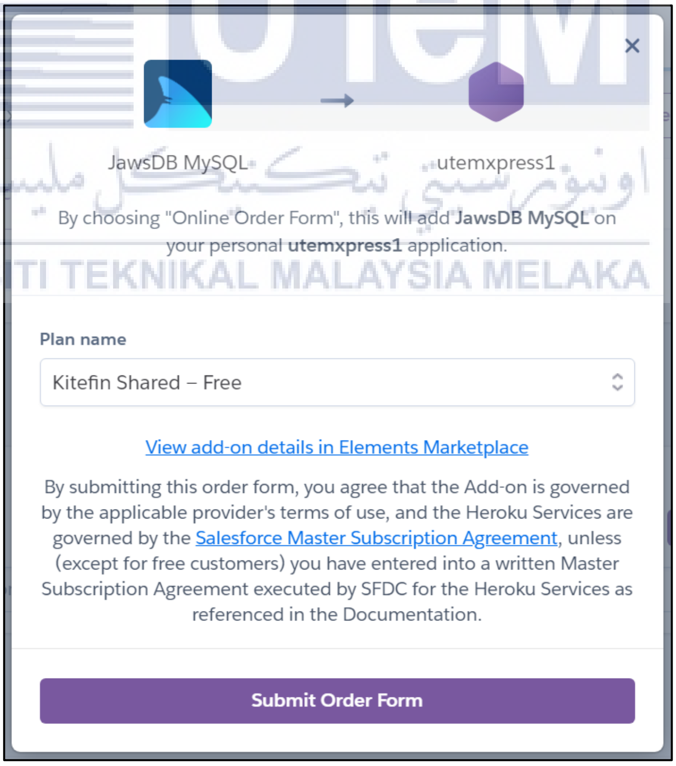
This app has no process types yet

Add a Procfile to your app in order to define its process types. Learn more

Add-ons

Figure 5.8:Resources Tab Page

Step 2: Click 'Submit Order Form' button.



By choosing "Online Order Form", this will add **JawsDB MySQL** on your personal **utemxpress1** application.

Plan name: Kitefin Shared – Free

View add-on details in Elements Marketplace

By submitting this order form, you agree that the Add-on is governed by the applicable provider's terms of use, and the Heroku Services are governed by the Salesforce Master Subscription Agreement, unless (except for free customers) you have entered into a written Master Subscription Agreement executed by SFDC for the Heroku Services as referenced in the Documentation.

Figure 5.9: Install Jaws DB

Step 3: On the ‘Overview’ tab, it will show JawsDB under the ‘Installed add-ons’ section. Then click the JawsDB MySQL and it will redirect you to connection information page.

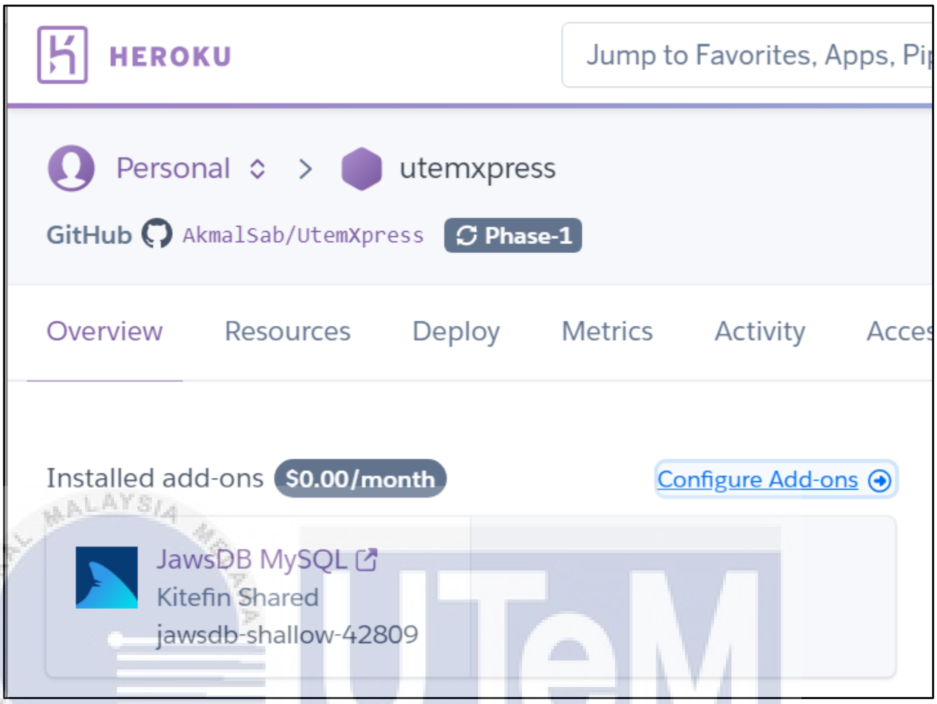
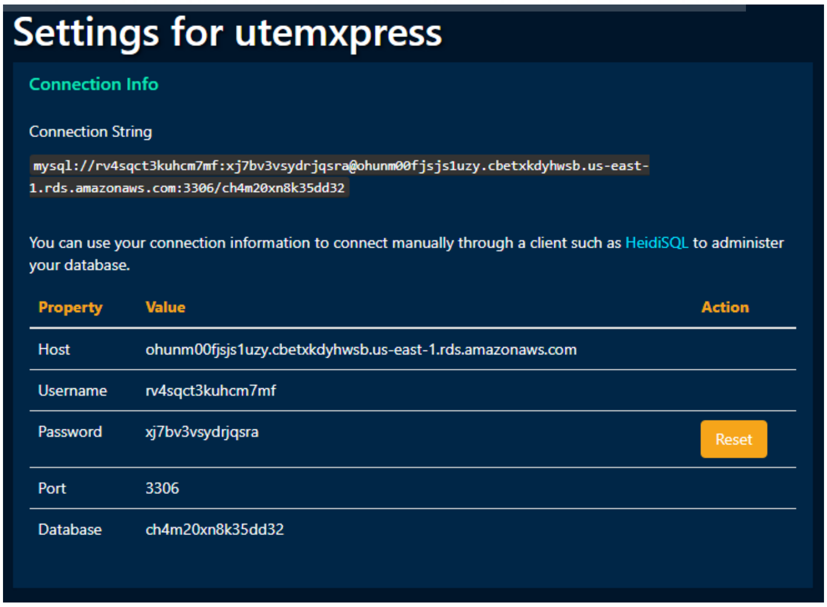


Figure 5.10: Heroku Overview Page

Step 4: JawsDB’s setting page for utemxpress project.



Settings for utemxpress

Connection Info

Connection String

```
mysql://rv4sqct3kuhcm7mf:xj7bv3vsydrjqsra@ohunm00fjsjs1uzy.cbetxkdyhwsb.us-east-
1.rds.amazonaws.com:3306/ch4m20xn8k35dd32
```

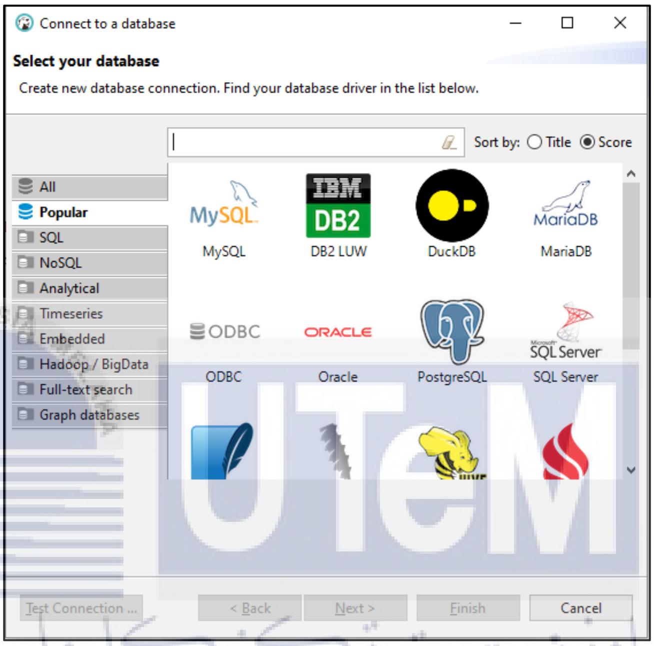
You can use your connection information to connect manually through a client such as HeidiSQL to administer your database.

Property	Value	Action
Host	ohunm00fjsjs1uzy.cbetxkdyhwsb.us-east-1.rds.amazonaws.com	
Username	rv4sqct3kuhcm7mf	
Password	xj7bv3vsydrjqsra	Reset
Port	3306	
Database	ch4m20xn8k35dd32	

Figure 5.11: Jaws DB Setting

Step 5: Install a SQL client software application and a database administration tool like ‘DBeaver’. Then enter the JawsDB connection information to connect with

Step 6: Click ‘Database’ tab and choose ‘new database connection’ option. The ‘connect to a database’ window will open and choose MySQL then click next.



Connect to a database

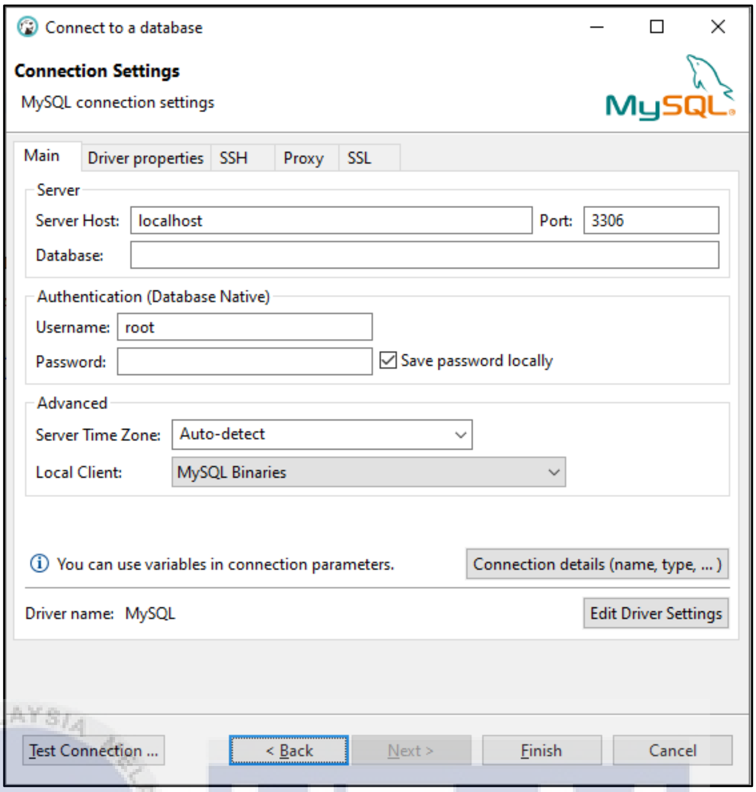
Select your database

Create new database connection. Find your database driver in the list below.

Sort by: ○ Title ● Score

Figure 5.12: Choose Database Type

Step 7: Enter the connection information provided by JawsDB into the window’s form then click finish. You will be automatically suggested to install connection drive for MySQL connection.



Server Host: localhost Port: 3306

Database:

Username: root

Password: ☑ Save password locally

Server Time Zone: Auto-detect

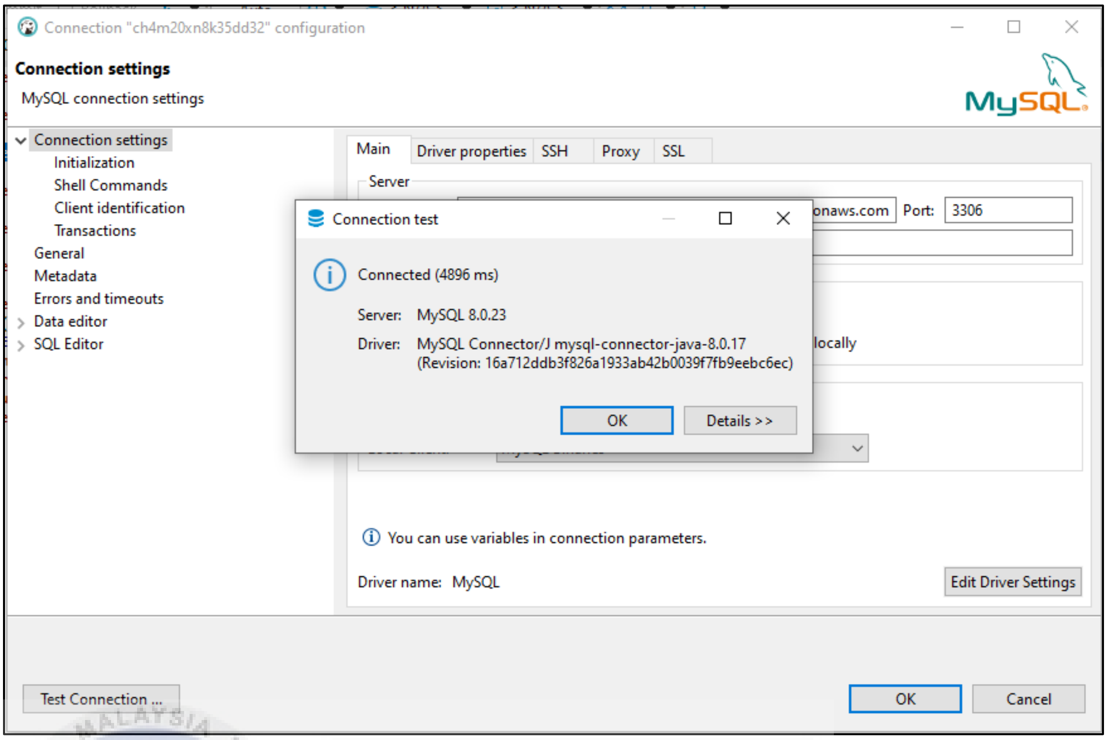
Local Client: MySQL Binaries

You can use variables in connection parameters.

Driver name: MySQL

Figure 5.13: Connection Settings

Step 8: After the connection has been made successfully, open a new sql script in DBeaver by pressing ctrl+j and then run all the sql command of the entire database that you were developed during localhost development.



Port: 3306

Connected (4896 ms)

Server: MySQL 8.0.23

Driver: MySQL Connector/J mysql-connector-java-8.0.17 (Revision: 16a712ddb3f826a1933ab42b0039f7fb9eebc6ec)

You can use variables in connection parameters.

Driver name: MySQL

Figure 5.14: Test Database Connection

5.3.1.3 Setup process of Firebase Storage

Step 1: Go to https://firebase.google.com/ and 'click go to console' button.

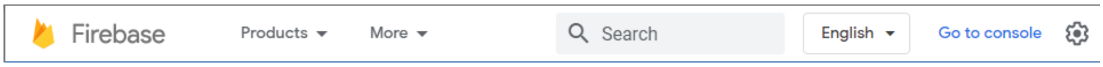


Figure 5.15: Firebase Header

Step 2: Click 'Add Project' button

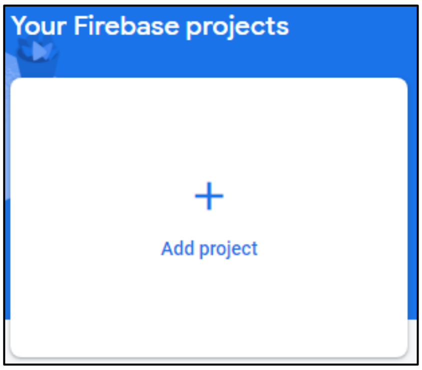
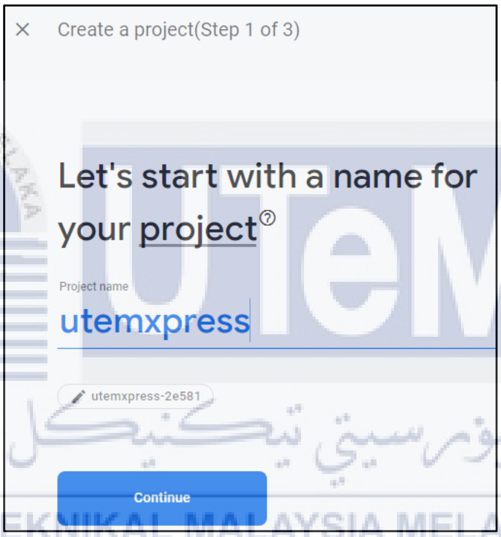


Figure 5.16: Add Project Firebase

Step 3: Enter the project name then click ‘continue’ button.



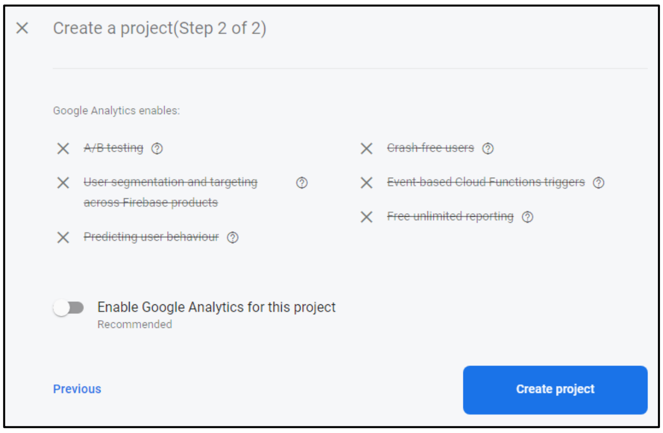
Let's start with a name for your project

Project name: utemxpress

utemxpress-2e581

Figure 5.17: Enter Project Name

Step 4: Uncheck the ‘enable Google analytic for this project’ and click ‘Create project’ button



Create a project(Step 2 of 2)

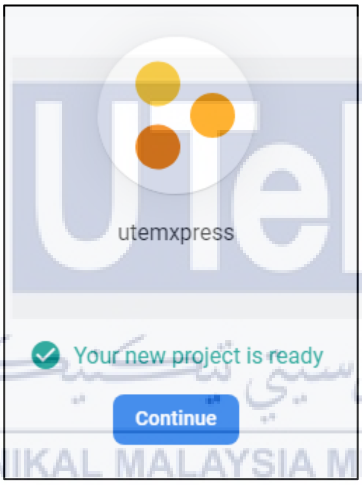
Google Analytics enables:

- ~~A/B testing~~
- ~~User segmentation and targeting across Firebase products~~
- ~~Predicting user behaviour~~
- ~~Crash-free users~~
- ~~Event-based Cloud Functions triggers~~
- ~~Free unlimited reporting~~

Enable Google Analytics for this project
Recommended

Figure 5.18: Project Services

Step 5: Click ‘continue’ button.



utemxpress

Your new project is ready

Figure 5.19: Project Creation Complete

Step 6: On the left-side menu panel, click ‘Storage’ menu.

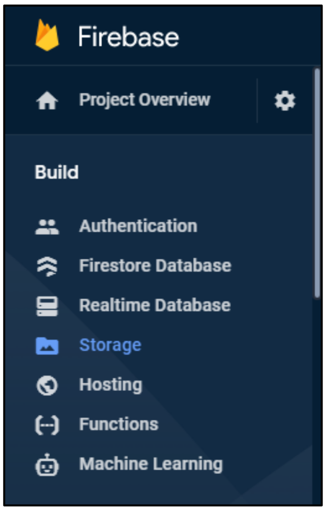
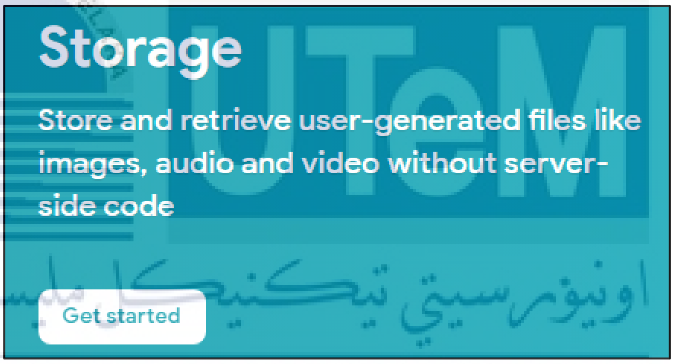


Figure 5.20: Firebase Menu

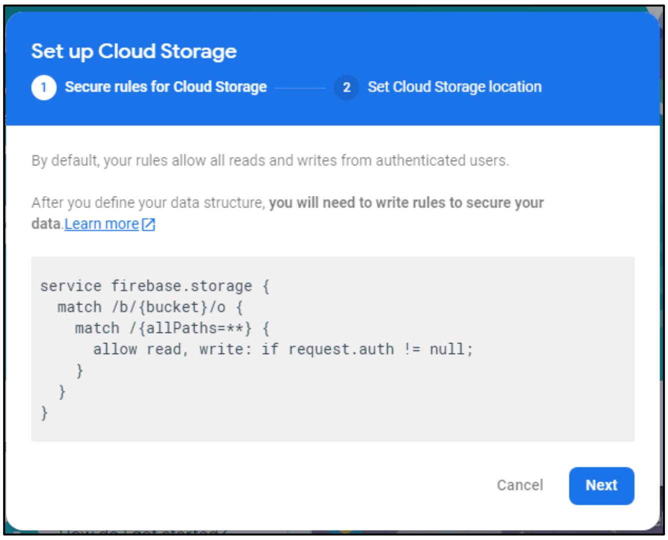
Step 7: Click ‘Get started’ button.



Store and retrieve user-generated files like images, audio and video without server-side code

Figure 5.21: Firebase Storage

Step 8: Click ‘Next’ button.



Set up Cloud Storage
1 Secure rules for Cloud Storage
2 Set Cloud Storage location

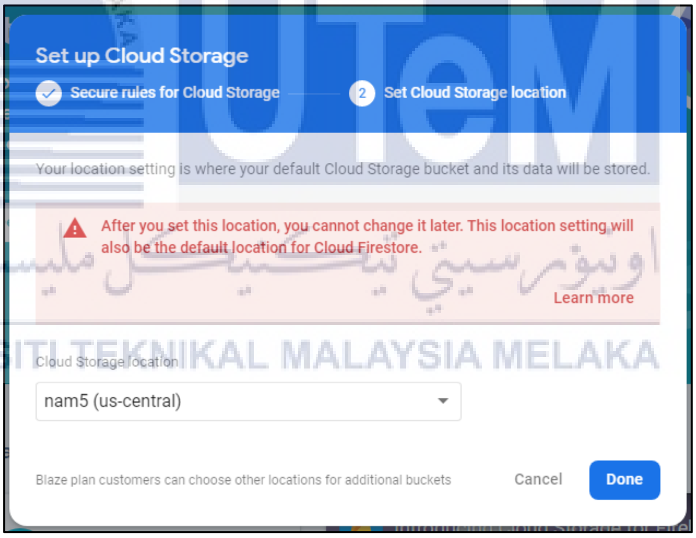
By default, your rules allow all reads and writes from authenticated users.

After you define your data structure, **you will need to write rules to secure your data** Learn more

```
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write: if request.auth != null;
    }
  }
}
```

Figure 5.22: Firebase Storage Rules

Click ‘Done’ button.



Your location setting is where your default Cloud Storage bucket and its data will be stored.

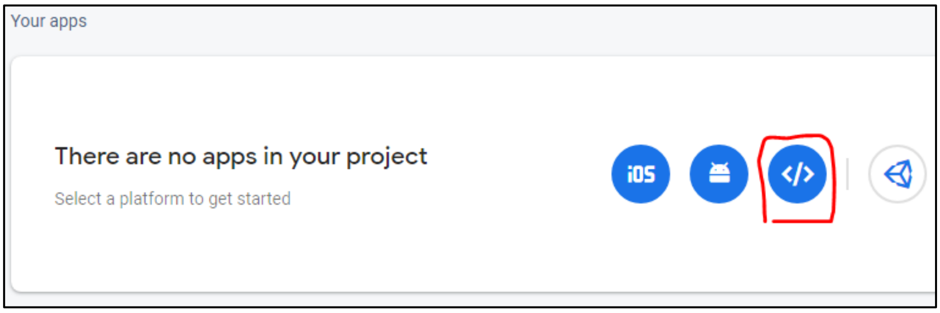
After you set this location, you cannot change it later. This location setting will also be the default location for Cloud Firestore.

Cloud Storage location: nam5 (us-central)

Blaze plan customers can choose other locations for additional buckets

Figure 5.23: Firebase Storage Location

Step 9: Click the 'setting' icon next to project overview. In 'General' tab, scroll down to 'Your apps' section and click the icon below.



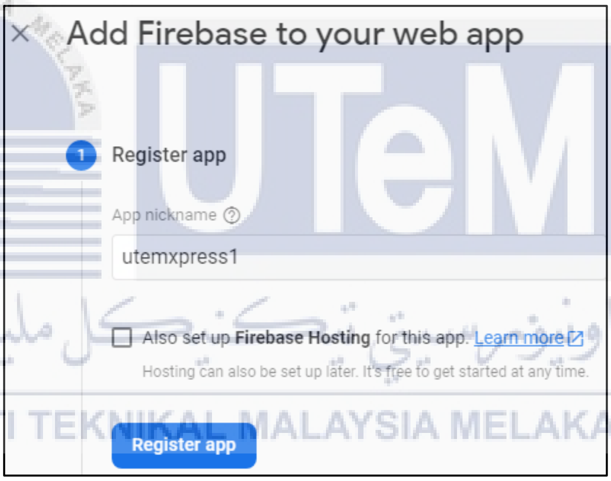
Your apps

There are no apps in your project

Select a platform to get started

Figure 5.24: Add App to Project

Step 10: Enter your app name in the text input then click 'Register app' button.



Add Firebase to your web app

1 Register app

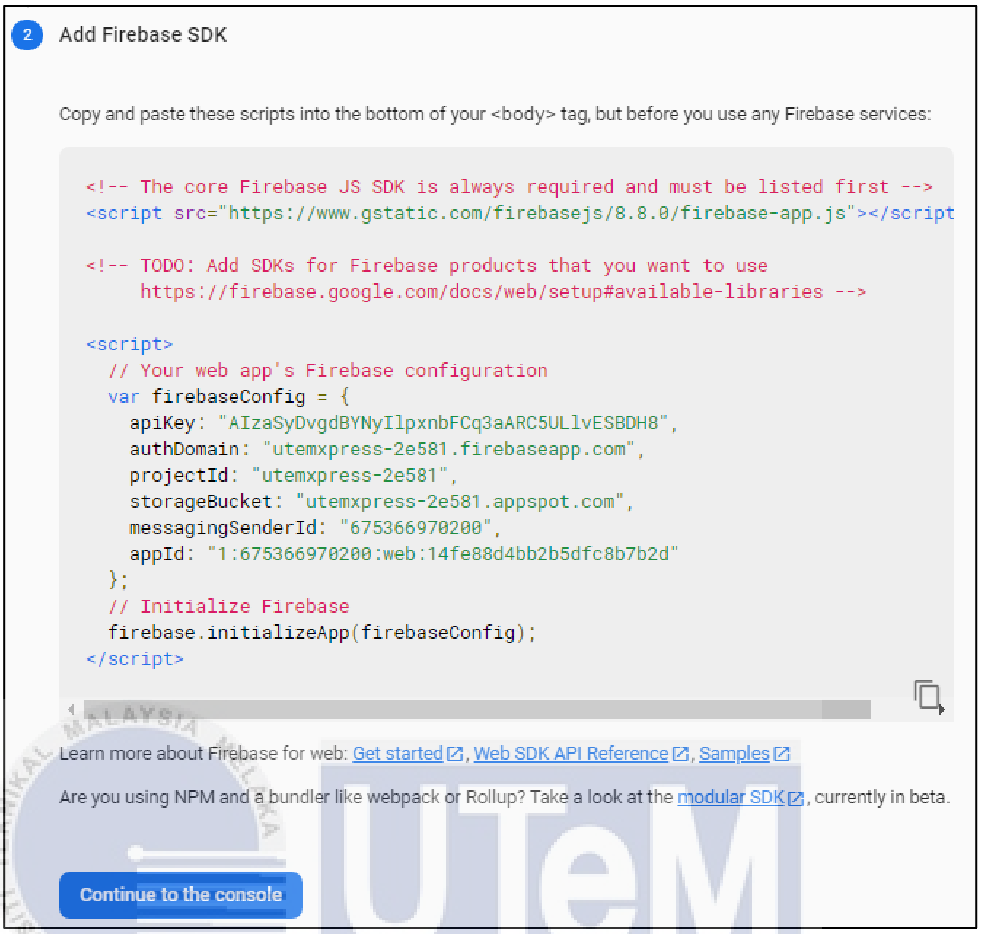
App nickname: utemxpress1

□ Also set up **Firebase Hosting** for this app. Learn more

Hosting can also be set up later. It's free to get started at any time.

Figure 5.25: Firebase App Name

Then copy the Firebase SDK code below and paste it in your project file. Then go to continue to the console.



2 Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

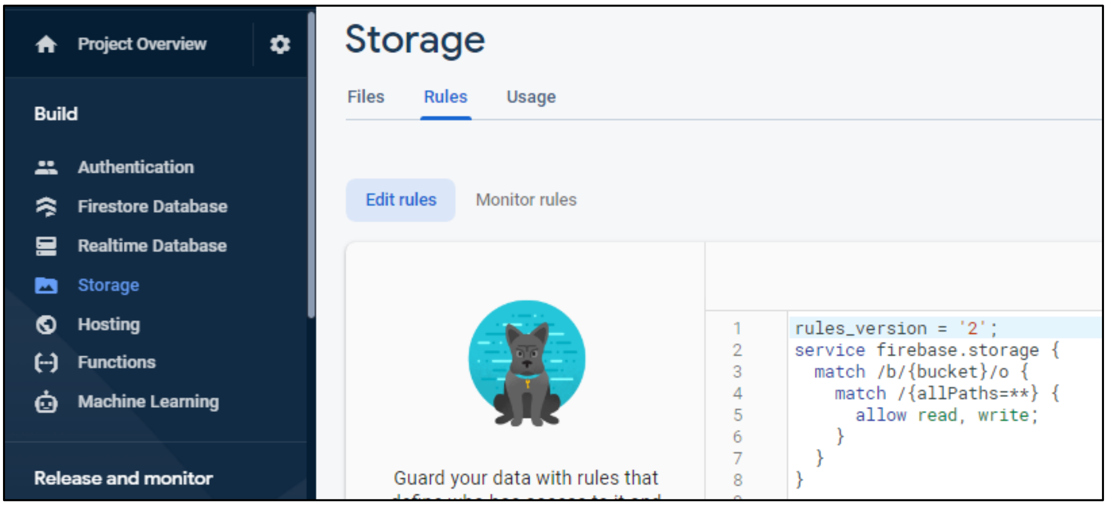
```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/8.8.0/firebase-app.js"></script

<!-- TODO: Add SDKs for Firebase products that you want to use
     https://firebase.google.com/docs/web/setup#available-libraries -->

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyDvgdBYNyIlpxnbFCq3aARC5ULlvESBDH8",
    authDomain: "utemxpress-2e581.firebaseapp.com",
    projectId: "utemxpress-2e581",
    storageBucket: "utemxpress-2e581.appspot.com",
    messagingSenderId: "675366970200",
    appId: "1:675366970200:web:14fe88d4bb2b5dfc8b7b2d"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: Get started, Web SDK API Reference, Samples

Are you using NPM and a bundler like webpack or Rollup? Take a look at the modular SDK, currently in beta.

Figure 5.26: Firebase App SDK

Step 11: Go back to Firebase Storage, click ‘Rules’ tab and edit your rules same as in the picture below.



```
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /{allPaths=**} {
      allow read, write;
    }
  }
}
```

Figure 5.27: Firebase Storage Rules

5.3.2 Version Control Procedure

The version control of this system is implemented with GitHub. GitHub is a web-based Git repository hosting service or website. GitHub provides the Git functionalities such as version control and source code management with its added features. Besides providing the version control mechanism for the development of the code, it is also an efficient mean to store file changes and maintain file integrity.

5.3.2.1 GitHub

Step 1: Go to https://github.com/ , create an account if you not registered yet.

Step 2: Click the + logo and select 'New repository' option.

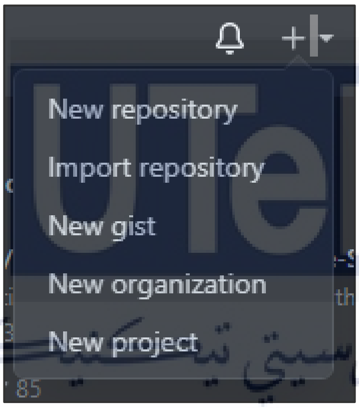
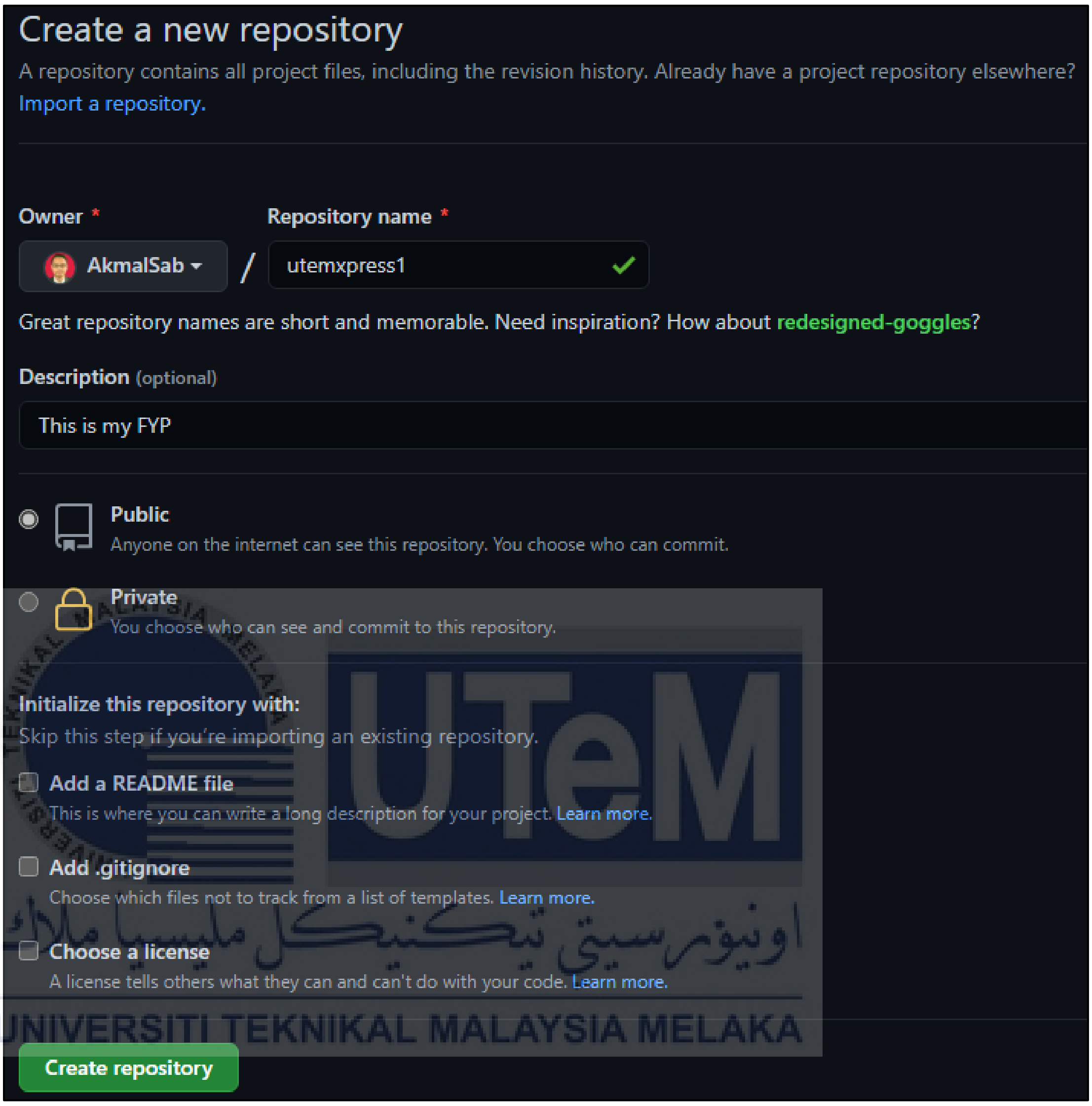


Figure 5.28: Create New GitHub Repositories

Step 3: Enter the repository name and description then click ‘Create repository’ button.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * AkmalSab / Repository name * utemxpress1 ✓

Great repository names are short and memorable. Need inspiration? How about redesigned-goggles?

Description (optional)

This is my FYP

● Public
Anyone on the internet can see this repository. You choose who can commit.

○ Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

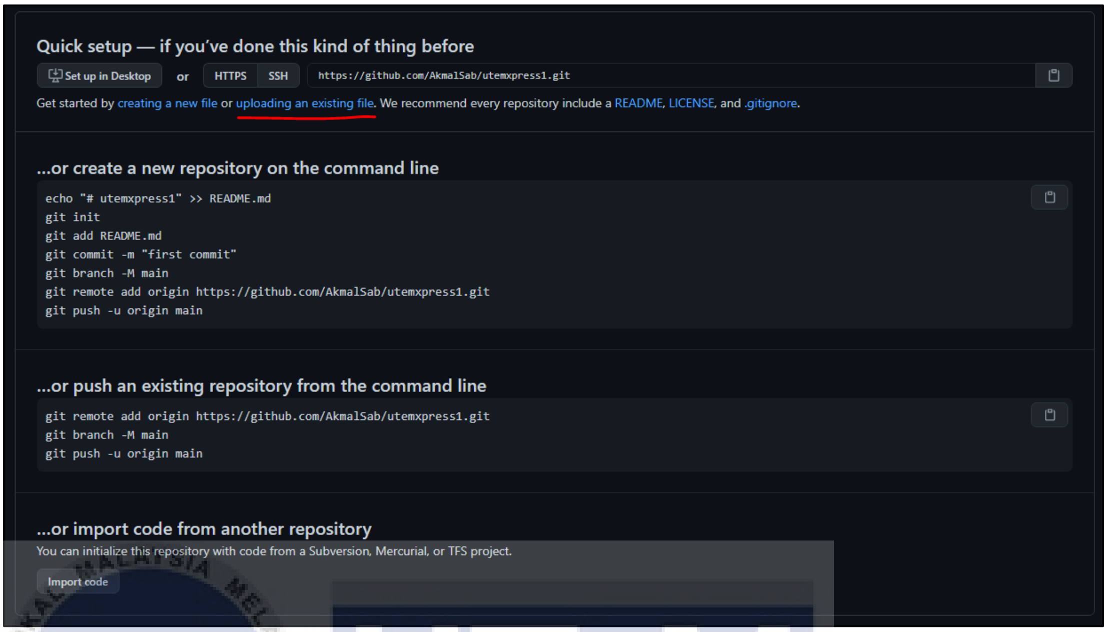
□ Add a README file
This is where you can write a long description for your project. Learn more.

□ Add .gitignore
Choose which files not to track from a list of templates. Learn more.

□ Choose a license
A license tells others what they can and can't do with your code. Learn more.

Figure 5.29: Repositories Details

Step 4: Click ‘upload an existing file’ link and drag your working folder into GitHub and click ‘commit’ button.



Quick setup — if you’ve done this kind of thing before

```
https://github.com/AkmalSab/utemxpress1.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# utemxpress1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/AkmalSab/utemxpress1.git
git push -u origin main
```

...or push an existing repository from the command line

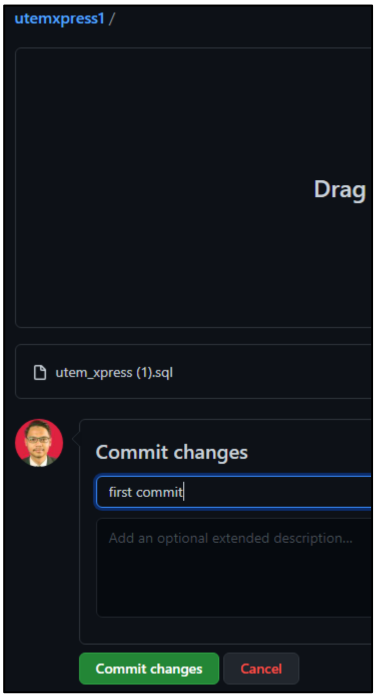
```
git remote add origin https://github.com/AkmalSab/utemxpress1.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

Figure 5.30: GitHub New Repository

Drag the project file in the area and write the commit message for example “first commit” then click ‘commit changes’ button.



first commit

Figure 5.31: First Commit

This is how its look like when we upload the file into our repository.

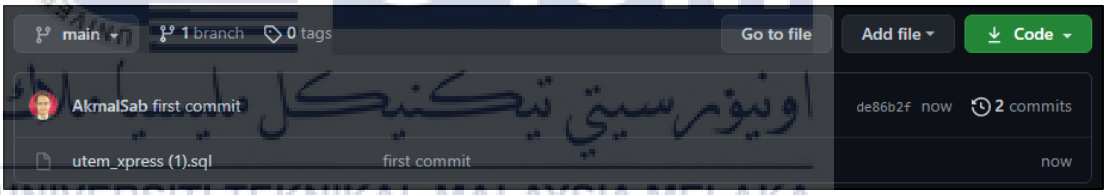
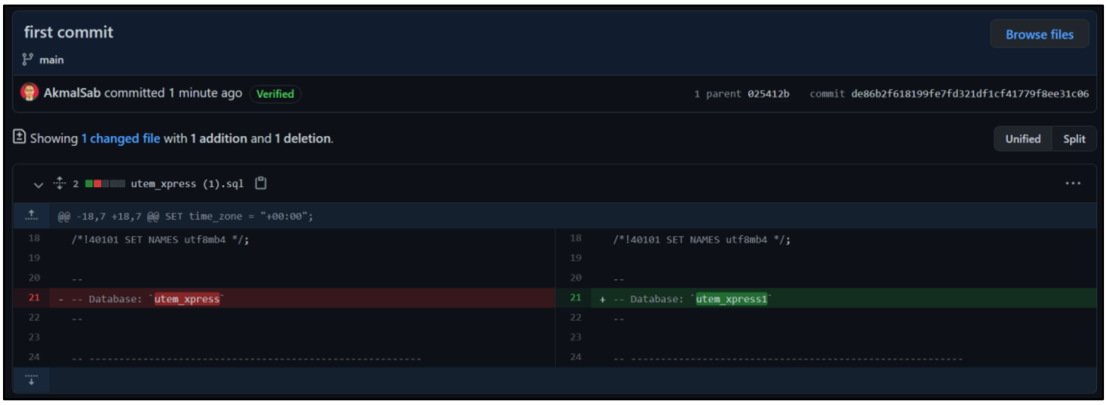


Figure 5.32: Committed Repository

Step 5: Make changes to the same file and re-upload again into GitHub repository. GitHub can detect any changes were made to every file by comparing the old commit with the new one. As you can see below the Database's name were change from utem_xpress to utem_xpress1. The red color is old file, and the green is the new file.



first commit

main

AkmalSab committed 1 minute ago Verified

1 parent 025412b commit de86b2f618199fe7fd321df1cf41779f8ee31c06

Showing 1 changed file with 1 addition and 1 deletion.

```
2  utem_xpress (1).sql

@@ -18,7 +18,7 @@ SET time_zone = "+00:00";
  /*!40101 SET NAMES utf8mb4 */;          /*!40101 SET NAMES utf8mb4 */;

  --                                       --
- -- Database: `utem_xpress`             + -- Database: `utem_xpress1`
  --                                       --

  -- --------------------------------      -- --------------------------------
```

Figure 5.33: Comparison Files

5.4 Implementation Status

This section describes the progress of the development status for each of the module in the system. The implementation status of each of the main module are presented in table 5.2 below.

Table 5.2: Module Date Completion

No.	Module	Description	Complete Duration	Date Completed
		User management		
1.	Login	Verify and validate user with credentials provided.	5 days	26/03/2021
2.	Register	Register new user into system and validate user from UTeM's email.	5 days	31/03/2021
3.	Add profile personal image	Allow user's add profile personal image.	3 days	03/04/2021

4.	Add Front IC Image	Allow users to add front IC image.	3 days	06/04/2021
5.	Add Back IC Image	Allow users to add back IC image.	3 days	09/04/2021
6.	Logout	Allow users to logout from system.	1 days	10/04/2021
Order management				
7.	Create order on demand	Allow user to create an order for on-demand	5 days	19/04/2021
8.	Create order for later	Allow user to create an order for later time and date	5 days	24/04/2021
9.	Order history record	Display all order records based on all type of order categories.	3 days	27/04/2021
10.	View order route	Display map route from pickup up to drop off location.	7 days	04/05/2021
11.	Cancel order	Allow users to cancel active order.	1 days	05/05/2021
12.	Active order On-Demand	Display all active current on demand order made by users.	2 days	07/05/2021
13.	Active For-Later	Display all active current for later order made by users.	2 days	09/05/2021
14.	Add runner to Favorite	Add runner into users favorite list.	3 days	12/05/2021
Job management				
15.	Order Listing	Display waiting On-demand & For-later order	5 days	17/05/2021

16.	Past order record	Display past order record based on order status	1 days	18/05/2021
Rate and Review management				
17.	Rate order and review runner	Allow users and runner to rate the order.	5 days	23/05/2021
Vehicle Management				
18.	Add vehicle information	Allow runner to add vehicle and road tax image.	5 days	28/05/2021
19.	Add license information	Allow runner to add license image	5 days	02/06/2021
Report Management				
20.	Earnings	Display earnings earn by runner.	7 days	08/06/2021
21.	Statistic	Display statistic of overall information system's current state	7 days	14/06/2021
Runner management				
22.	Update Vehicle information	Allow runner to alter vehicle information	4 days	18/06/2021
23.	Update License information	Allow runner to alter license information	4 days	22/06/2021
Admin management				
24.	Activate & deactivate user	Allow admin to change users, runners & other admins status.	4 days	14/04/2021
25.	Add new user	Allow admin to add new student, staff, and admin into system.	2 days	16/04/2021

26.	Cancel Order	Allow admin to cancel order on behalf of user.	1 days	17/04/2021

5.5 Conclusion

This chapter covers software development environment setup at the first part of this chapter and software configuration management in the following part which elaborates on configuration environment setup and version control procedures. It also provides overview of implementation status of every module in UTeM Xpress.

CHAPTER 6: TESTING

6.1 Introduction

This chapter reveals the details of the test plan which contains the description of test organization, test environment and test schedule. It also describes the test strategy in term of tests classes. The chapter is the followed by the test design consists of test description and test data. Then, the chapter presents the test results and analysis based on the testing activities

6.2 Test Plan

This section explains the personnel involved in the test organization. Then, test environment and test schedule are also documented in the following sections.

6.2.1 Test Organization

According to Figure 6.1, it shows the test organization which consists of one testing manager as the group leader and three testers

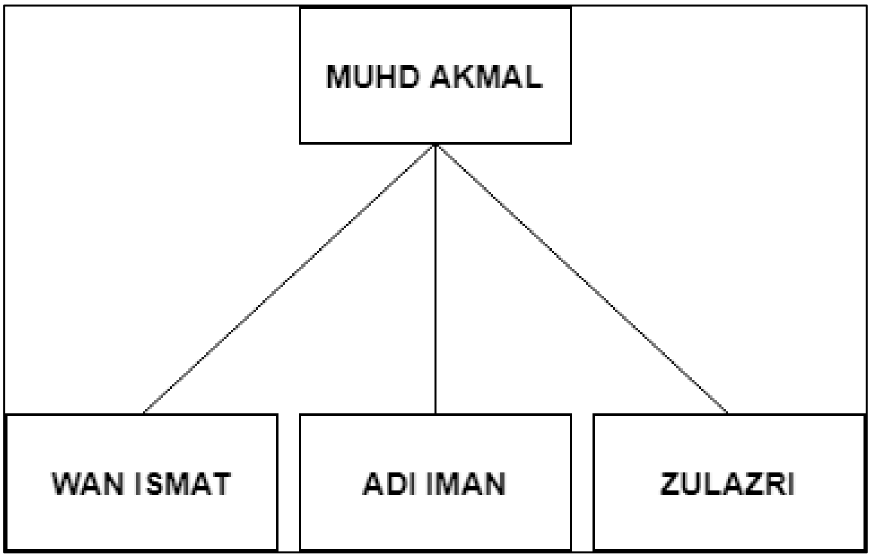


Figure 6.1: Test Organization

In the testing phase of this project, supervisor is appointed as the testing manager and developer acts as tester 1 who prepares test cases, test data and test environment. In the functionality test, the testing manager supervises all the testing to evaluate the output correctness. While in the user acceptance test, supervisor acts as the user to carry out all the necessary tests according to the test cases as well as few suggested test cases to detect more defects or bugs.

6.2.2 Test Environment

All the testing including unit testing, integration testing and system testing are done in the development environment. The personal computer hardware specification and firmware configurations thus are same as the one used in development environment. It is defined chapter 2.4.2. All the testing is completed in the LAN based environment with connection to internet. Server and database configurations are done according to the chapter 5.3.1.1

6.2.3 Test Schedule

According to the test plan, all the testing activities are scheduled to be completed within seven days from, 19th July 2021, and 26th July 2021. Table 6.1 shows the test schedule which displays the date to carry out test cases of each main module and sub-module of the system. In the first day, all the testing activities consists of carrying out all the test cases must be done at least once, some test cases should be done at least two times to clarify the discovered bugs. All the test cases that are failed in the first day should be repeated on the second testing day once the fix has been done, which also known as regression testing.

Table 6.1: Testing Schedule

No.	Module	Duration	Schedule
	User management		
1.	Login	1 day	1/08/2021
2.	Register	2 days	2/08/2021
3.	Add profile personal image	1 day	4/08/2021
4.	Add Front IC Image	1 day	5/08/2021
5.	Add Back IC Image	1 day	6/08/2021
6.	Logout	1 day	7/08/2021
	Order management		
7.	Create order on demand	3 days	8/08/2021
8.	Create order for later	3 days	11/08/2021
9.	Order history record	1 day	14/08/2021
10.	View order route	1 day	15/08/2021
11.	Cancel order	1 day	16/08/2021
12.	Active order On-Demand	1 day	17/08/2021
13.	Active For-Later	1 day	18/08/2021
14.	Add runner to Favorite	1 day	19/08/2021
	Job management		
15.	Order Listing	1 day	20/08/2021
16.	Past order record	1 day	21/08/2021
	Rate and Review management		
17.	Rate order and review runner	1 days	22/08/2021
	Vehicle Management		
18.	Add vehicle information	3 days	25/08/2021
19.	Add license information	3 days	28/08/2021
	Report Management		

20.	Earnings	1 days	29/08/2021
21.	Statistic	1 days	30/08/2021
Runner management			
22.	Update Vehicle information	3 days	31/06/2021
23.	Update License information	3 days	3/07/2021
Admin management			
24.	Activate & deactivate user	1 days	6/07/2021
25.	Add new user	1 days	7/07/2021
26.	Cancel Order	1 days	8/07/2021

6.3 Test Strategy

This section discusses the test strategy implemented in the testing phase. A test strategy refers to combines different kinds of test case design methods to form a properly structured series of steps which ensure the success of software testing. Test strategy is commonly constructed by testing professionals, like testing specialist, software engineer and project managers. The selected test strategy is bottom-up approach and black-box classes of tests. Bottom-up testing begins with the lowest level component testing in the testing of submodule that are not yet integrated and followed by the main module testing. Then, the testing phase proceeds with integration and system testing. Unit testing is carried out at the lowest level. It examines the smallest unit of the system, which can be a module or component. The unit of each unit testing is smallest module which can be the smallest set of lines of code. Unit testing is required by test strategy because it can contribute to the system testing. Unit testing is commonly treated as a white box test class. Integration testing is carried out after more than one tested unit are integrated to form a larger structure. Integration testing is generally done on the interfaces that are between the components and modules. If the quality attribute of components cannot be assessed with proper ways, then integration testing is done on the larger structure that is being combined by more than one component or modules. System testing aims to run the end-to-end quality test of the system. This testing is done according to the requirement specifications and functional requirements of the system. System testing also assures non-function quality attributes are being checked, including performance, reliability, and usability (Chauhan & Singh, 2014). Black-box testing is the testing technique used by the testers to inspect the functionalities without knowing the how are the internal workings of the tested part (Khan & Khan, 2014). The primary purpose of black-box testing is to test how well the specified requirements for the system are fulfilled by the system. Black box testing does not require testers to have complete knowledge or understanding of the internal logical structure of the system as testers only carry out testing on the important aspects of the system. The black-box testing examines whether all inputs are correctly accepted, and outputs are accurately produced (Chauhan & Singh, 2014).

6.3.1 Classes of tests

Functionality test is carried out to evaluate the whether the specified system functionalities are working in desired way and produce correct output after a set of proper input is accepted. All the functionality related test cases are explained in following section, Chapter 6.4.

6.4 Test Design

This section describes the test case of every module in test description. It also presents the test data used to run the test cases.

6.4.1 Test Description

Test case identification, test cases and expected result for each module are designed and documented in this section.

6.4.2 Use Case (Black Box Testing)

A Use Case in Testing is a brief description of a particular use of the software application by an actor or user. Use cases are made on the basis of user actions and the response of the software application to those user actions. It is widely used in developing test cases at system or acceptance level.

6.4.2.1 Login use case

Table 6.2: Login Use Case

Main Success Scenario	**Step**	**Description**
A: Actor S: System	1	A: Enter UTeM email and password
	2	S: Validate credentials
	3	S: Redirect to home
Extensions	1a	**Credentials empty** S: Display 'The email field is required' message. S: Display 'The password field is required' message.
	2a	**UTeM email not exist** S: Display 'User not found, if you are runner please tick 'I am runner' checkbox below' message
	2b	**Password is incorrect** S: Display 'This credential is incorrect' message
	2c	**Runner checkbox unchecked** S: Display 'User not found, if you are runner please tick 'I am runner' checkbox below' message

6.4.2.2 Register use case

Table 6.3: Register Use Case

Main Success Scenario	**Step**	**Description**
A: Actor S: System	1	A: Enter registration details
	2	S: Validate form data
	3	S: Redirect to verify email page
Extensions	1a	**Registration details empty** S: Display 'Please fill out this field' message.
	1b	**Password and confirm password does not match** S: Disabled submit button and display 'Confirm Password are not matched with password' message
	1c	**Password character length less than 8** S: Display 'Password length must be at least 8 characters' message
	2a	**User already exist** S: Display 'Student with the same role already exist !' message
	2b	**Phone number already in use** S: Display 'Phone number already been used !' message

6.4.2.3 Add profile personal image use case

Table 6.4: Add Profile Personal Image Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click add Personal Image button.
	2	A: Click choose file and upload image file type.
	3	S: Check file type.
	4	S: Redirect back to profile and display 'Personal image has been added' success message.
Extensions	3a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message

6.4.2.4 Add front IC image use case

Table 6.5: Add Front Ic Image Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click add IC Image button.
	2	A: Click choose file for IC front image and upload image file type.
	3	S: Check file type for IC front image.
	4	A: Click choose file for IC back image and upload image file type.
	5	S: Check file type for IC back image.
	6	S: Redirect back to profile and display 'Identification Card image has been added' success message.
Extensions	3a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message.
	5a	**File is non-image type**

		S: Display ‘Alert Files uploaded in non-picture format’ message.

6.4.2.5 Rate order and review runner use case

Table 6.6: Rate Order and Review Runner Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Choose stars to rate.
	2	A: Enter review.
	3	A: Click Submit button.
	4	S: Verify rate and review
Extensions	4a	**Rating is empty** S: Display ‘Please do not leave the rate empty’ error message.

6.4.2.6 Add vehicle information use case

Table 6.7: Add Vehicle Information Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click add vehicle button.
	2	A: Choose vehicle type,
	3	A: Enter vehicle plate number
	4	A: Click choose file for vehicle image and upload image file type.
	5	A: Click choose file for vehicle road tax image and upload image file type.
	6	A: Click submit button.
	7	S: Verify form data

	8	S: Redirect back to profile and display 'Vehicle information has been uploaded successfully' message.
Extensions	4a	<u>**File is non-image type**</u> S: Display 'Alert Files uploaded in non-picture format' message.
	5a	<u>**File is non-image type**</u> S: Display 'Alert Files uploaded in non-picture format' message.

6.4.2.7 Add license information use case

Table 6.8: Add License Information Use Case

Main Success Scenario	**Step**	**Description**
A: Actor S: System	1	A: Click add license button.
	2	A: Click choose file for license front image and upload image file type.
	3	A: Click choose file for license back image and upload image file type.
	4	A: Click submit button.
	5	S: Verify form data
	6	S: Redirect back to profile and display 'License image has been uploaded successfully' message.
Extensions	2a	<u>**File is non-image type**</u> S: Display 'Alert Files uploaded in non-picture format' message.
	3a	<u>**File is non-image type**</u> S: Display 'Alert Files uploaded in non-picture format' message.

6.4.2.8 Update vehicle information use case

Table 6.9: Update Vehicle Information Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click update vehicle button.
	2	A: Choose vehicle type,
	3	A: Enter vehicle plate number
	4	A: Click choose file for vehicle image and upload image file type.
	5	A: Click choose file for vehicle road tax image and upload image file type.
	6	A: Click submit button.
	7	S: Verify form data
	8	S: Redirect back to profile and display 'Vehicle information has been updated successfully' message.
Extensions	4a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message.
	5a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message.

6.4.2.9 Update license information use case

Table 6.10: Update License Information Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click update license button.
	2	A: Click choose file for license front image and upload image file type.
	3	A: Click choose file for license back image and upload image file type.
	4	A: Click submit button.
	5	S: Verify form data
	6	S: Redirect back to profile and display 'License image has been updated successfully' message.
Extensions	2a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message.
	3a	**File is non-image type** S: Display 'Alert Files uploaded in non-picture format' message.

6.4.2.10 Add new user use case

Table 6.11: Add New User Use Case

Main Success Scenario	Step	Description
A: Actor S: System	1	A: Click add new student button.
	2	A: Enter new user details
	3	A: Click add button
	4	S: Verify form data
	5	S: Redirect back to manage user page and display 'New student added' message.
Extensions	4a	<u>**Student already exist!**</u> S: Display 'Student already exist!' error message.

6.4.3 Unit Testing

Unit testing is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected. Unit Testing is done during the development (coding phase) of an application by the developers. Unit Tests isolate a section of code and verify its correctness. A unit may be an individual function, method, procedure, module, or object (see Appendix A).

No.	Unit Testing Name	Pages
1.	Login Test Case	131
2.	Register Test Case	134
3.	Add Profile Personal Image Test Case	137
4.	Add Front IC Image Test Case	138
5.	Add Back IC Image Test Case	140
6.	Logout Test Case	141
7.	Create Order on Demand Test Case	142
8.	Create Order for Later Test Case	145
9.	Order History Record Test Case	149

10.	View Order Route Test Case	150
11.	Cancel Order Test Case	152
12.	Active Order on Demand Test Case	152
13.	Active Order for Later Test Case	153
14.	Add Runner to Favorite Test Case	154
15.	Order Listing Test Case	154
16.	Past Order Record Test Case	155
17.	Rate Order and Review Runner Test Case	155
18.	Add Vehicle Information Test Case	157
19.	Add License Information Test Case	160
20.	Earnings Test Case	162
21.	Statistic Test Case	163
22.	Update Vehicle Information Test Case	164
23.	Update License Information Test Case	168
24.	Activate & Deactivate User Test Case	170
25.	Add New User Test Case	171

6.4.4 Integration Testing (Big Bang Approach)

Integration testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Big Bang Testing is an Integration testing approach in which all the components or modules are integrated together at once and then tested as a unit. This combined set of components is considered as an entity while testing. If all of the components in the unit are not completed, the integration process will not execute.

6.4.4.1 Integration Testing: Login Module

Table 6.12: Login Module

Test Case ID	Objective	Description	Expected Result
IT1	Check the interface link between the Login and create order module.	Enter login credentials and click on the login button	To be directed to create order form
IT2	Check the interface link between the Login and Register module	Click Create an account link	To be navigated to registration form
IT3	Check the interface link between the Login and Order Listing module (for runner)	Enter login credentials, check the I am runner checkbox and click on the login button	To be directed to order listing page.

6.4.4.2 Integration Testing: Register Module

Table 6.13: Register Module

Test Case ID	Objective	Description	Expected Result
IT4	Check the interface link between the Register and Verify Email module.	Enter registration form details and click Register button	To be redirected to email verification page.

6.4.4.3 Integration Testing: Create Order on Demand Module

Table 6.14: Create Order on Demand Module

Test Case ID	Objective	Description	Expected Result
IT5	Check the interface link between the Create Order on Demand and View Order Route module.	Enter order on demand form details, click Deliver Now button, enter details for Delivery Date & Contact Information form, click Confirm Order button and click View Route button.	To be redirected to order details page and a route map is displayed.
IT6	Check the interface link between the Create Order on Demand and Cancel Order module.	Enter order on demand form details, click Deliver Now button, enter details for Delivery Date & Contact Information form, click Confirm Order button and click Cancel Order button.	To be redirected to order details page and status of order is changed to 'Cancelled'
IT7	Check the interface link between the Create Order on Demand and Active order on Demand module.	Enter order on demand form details, click Deliver Now button, enter details for Delivery Date & Contact Information form, click Confirm Order button, click Active Order tab, and click On-Demand tab.	To be redirected to active order page and list of active order on demand listed under On-Demand Tab.

6.4.4.4 Integration Testing: Create Order for Later Module

Table 6.15: Create Order for Later Module

Test Case ID	Objective	Description	Expected Result
IT8	Check the interface link between the Create Order for Later and View Order Route module.	Enter order on demand form details, click Scheduled for Later button, enter details for Delivery Date & Contact Information form, click Confirm Order button and click View Route button.	To be redirected to order details page and a route map is displayed.
IT9	Check the interface link between the Create Order for Later and Cancel Order module.	Enter order on demand form details, click Scheduled for Later button, enter details for Delivery Date & Contact Information form, click Confirm Order button and click Cancel Order button.	To be redirected to order details page and status of order is changed to ‘Cancelled’
IT10	Check the interface link between the Create Order for Later and Active order for Later module.	Enter order on demand form details, click Scheduled for Later button, enter details for Delivery Date & Contact Information form, click Confirm Order button, click Active Order tab, and click For-Later tab.	To be redirected to active order page and list of active order for later listed under For-Later Tab.

6.4.4.5 Integration Testing: Order History Record Module

Table 6.16: Order History Record Module

Test Case ID	Objective	Description	Expected Result
IT11	Check the interface link between the Order History Record and Order Details module.	Click Order Record tab on the navbar and select each tab of order status and click on the one of the orders in the list.	To be redirected to order details page.

6.4.4.6 Integration Testing: Add runner to Favorite Module

Table 6.17: Add Runner to Favorite Module

Test Case ID	Objective	Description	Expected Result
IT12	Check the interface link between the Add runner to Favorite and Order Details module.	Once the order status on the order details page is set to completed, click Add to Favorite button.	To be redirected back to order detail page with success message 'runner has added to favorite'.

6.4.4.7 Integration Testing: Past Order Record

Table 6.18: Past Order Record

Test Case ID	Objective	Description	Expected Result
IT13	Check the interface link between the Past	Click Past Order tab on the navbar and select each tab of	To be redirected to

	Order Record and Order Details module.	order status and click on the one of the orders in the list.	order details page.

6.4.4.8 Integration Testing: Rate order and Review runner

Table 6.19: Rate Order and Review Runner

Test Case ID	Objective	Description	Expected Result
IT14	Check the interface link between the Rate order and Review runner and Order Details module.	Once the order status has change to completed, enter the rating form, and click Submit button	To be redirected back to Order Details page and Order has been rated successfully message is displayed.

6.5 Test Result and Analysis

According to the result of the test cases in appendix A, more than 91% of the test cases have obtained the pass result and the rest have failed in the testing. Users have been involved actively throughout the testing process by completing all the test cases with the testers. There are some fail results found in some test cases that are suggested by the users and solutions will be sought to improve the test results. Overall, user is satisfied with the developed project as most of the test cases are achieving positive results which indicates the specified requirements are properly implemented and fulfilled in this project.

6.5.1 Unit Testing Result (First Testing)

Table 6.20: Unit Testing Result

No.	Test Case ID	Result (Pass/Fail)	Remarks if Fail	Tested By
1.	UML1	Pass	-	Ismat
2.	UML2	Pass	-	Ismat
3.	UML3	Pass	-	Ismat
4.	UML4	Pass	-	Ismat
5.	UML5	Pass	-	Ismat
6.	UMR1	Pass	-	Ismat
7.	UMR2	Pass	-	Ismat
8.	UMR3	Pass	-	Ismat
9.	UMR4	Pass	-	Ismat
10.	UMR5	Pass	-	Ismat
11.	UMR6	Pass	-	Ismat
12.	UMP1	Pass	-	Ismat
13.	UMP2	Pass	-	Ismat
14.	UMP3	Fail	Personal image can be uploaded even non-image file type.	Ismat
15.	UMF1	Pass	-	Ismat
16.	UMF2	Pass	-	Ismat
17.	UMF3	Fail	Front IC image can be uploaded even non-image file type.	Ismat
18.	UMB1	Pass	-	Ismat
19.	UMB2	Pass	-	Ismat
20.	UMB3	Fail	Back IC image can be uploaded even non-image file type.	Ismat
21.	UMO1	Pass	-	Ismat
22.	OMD1	Pass	-	Ismat
23.	OMD2	Pass	-	Adi
24.	OMD3	Pass	-	Adi

25.	OMF1	Pass	-	Adi
26.	OMF2	Pass	-	Adi
27.	OMF3	Pass	-	Adi
28.	OMF4	Pass	-	Adi
29.	OMH1	Pass	-	Adi
30.	OMR1	Pass	-	Adi
31.	OMC1	Pass	-	Adi
32.	OMA1	Pass	-	Adi
33.	OMB1	Pass	-	Adi
34.	OMT1	Pass	-	Adi
35.	JMO1	Pass	-	Adi
36.	JMP1	Pass	-	Adi
37.	RRR1	Pass	-	Adi
38.	RRR2	Pass	-	Adi
39.	VMV1	Pass	-	Adi
40.	VMV2	Pass	-	Adi
41.	VMV3	Pass	-	Adi
42.	VMV4	Fail	Vehicle and road tax image can be uploaded even non-image file type.	Adi
43.	VMV5	Pass	-	Adi
44.	VML1	Pass	-	Adi
45.	VML2	Pass	-	Zulazri
46.	VML3	Pass	License image can be uploaded even non-image file type.	Zulazri
47.	RME1	Pass	-	Zulazri
48.	RMS1	Pass	-	Zulazri
49.	RMU1	Pass	-	Zulazri
50.	RMU2	Pass	-	Zulazri
51.	RMU3	Pass	-	Zulazri
52.	RMU4	Fail	Update Vehicle and road tax image can be uploaded	Zulazri

			even non-image file type.	
53.	RMU5	Pass	-	Zulazri
54.	RML1	Pass	-	Zulazri
55.	RML2	Pass	-	Zulazri
56.	RML3	Fail	Update License image can be uploaded even non-image file type.	Zulazri
57.	AMA1	Pass	-	Zulazri
58.	AMA2	Pass	-	Zulazri
59.	AMA3	Pass	-	Zulazri
60.	AMA4	Pass	-	Zulazri
61.	AMA5	Pass	-	Zulazri
62.	AMA6	Pass	-	Zulazri
63.	AMU1	Pass	-	Zulazri
64.	AMU2	Pass	-	Zulazri
65.	AMU3	Pass	-	Zulazri
66.	AMU4	Pass	-	Zulazri
67.	AMU5	Pass	-	Zulazri
68.	AMU6	Pass	-	Zulazri

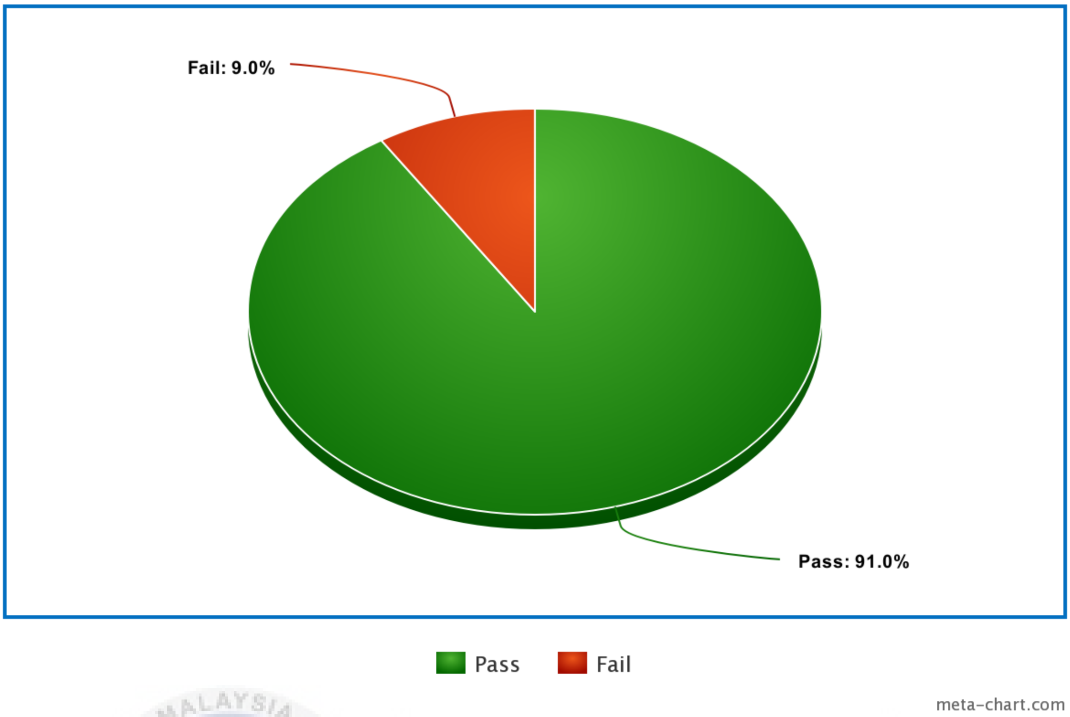


Figure 6.2: Unit Testing Result Chart

In the first phase of the test implemented, the results found that there were several unit tests that failed because of image-type file validation errors. The percentage for failed tests is 9% while for those who succeed it is 91%.

6.5.2 Unit Testing (Second Testing)

Table 6.21: Unit Testing

No.	Test Case ID	Result (Pass/Fail)	Remarks if Fail	Tested By
14.	UMP3	Pass	-	Ismat
17.	UMF3	Pass	-	Ismat
20.	UMB3	Pass	-	Ismat
42.	VMV4	Pass	-	Adi
52.	RMU4	Pass	-	Zulazri
56.	RML3	Pass	-	Zulazri

6.5.3 Integration Testing Result (First Testing)

According to the result of the integration testing below, 100% of the test cases have obtained the pass result and none have failed in the testing even for the first try.

Table 6.22: Integration Testing

No.	Test Case ID	Result (Pass/Fail)	Remarks if Fail	Tested By
1.	IT1	Pass	-	Muhd Akmal
2.	IT2	Pass	-	Muhd Akmal
3.	IT3	Pass	-	Muhd Akmal
4.	IT4	Pass	-	Muhd Akmal
5.	IT5	Pass	-	Muhd Akmal
6.	IT6	Pass	-	Muhd Akmal
7.	IT7	Pass	-	Muhd Akmal
8.	IT8	Pass	-	Muhd Akmal
9.	IT9	Pass	-	Muhd Akmal
10.	IT10	Pass	-	Muhd Akmal
11.	IT11	Pass	-	Muhd Akmal
12.	IT12	Pass	-	Muhd Akmal
13.	IT13	Pass	-	Muhd Akmal
14.	IT14	Pass	-	Muhd Akmal

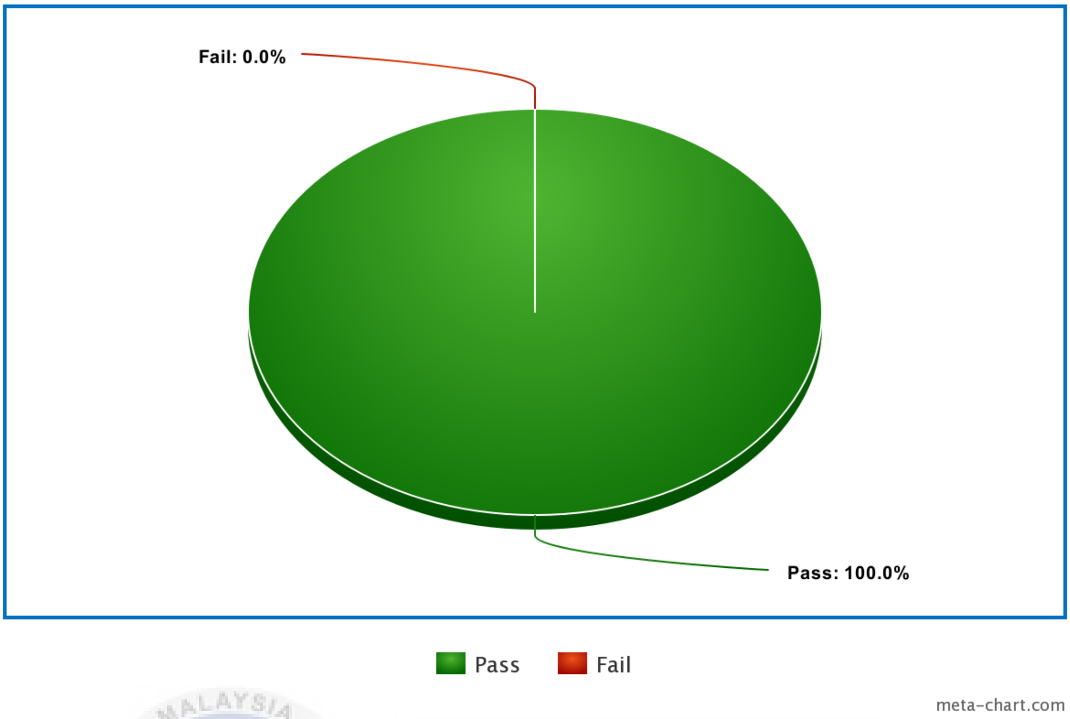


Figure 6.3: Integration Testing Result Chart

In the second phase of the test was implemented, the results found that all the test units were successful as a result of the correction on the failed test units during the first phase was carried out.

6.6 Conclusion

This chapter describes and discusses the details of the testing phases and testing that are run to evaluate and validate the UTeM Xpress System functionality. Next chapter presents the conclusion of this project, it will cover the potential improvements and contribution of this project.

CHAPTER 7: PROJECT CONCLUSION

7.1 Observation on Weakness and Strengths

Throughout the testing phase run on the UTeM Xpress system, we have identified some of bugs that contained among the system modules and the correction has been made towards the error found in system. From the testing phase, some of the users also pay attention to every detail of the system and each of them have their own opinion towards the weakness and strengths from this system.

Among the weakness that we have gather from users' opinion is tracking features which enable customer to track runner current location movement from maps. Currently, this feature is not available as it requires a lot of effort and exploration to make this thing work. Users are believing that with this feature contain in the system it would make the customers easier to estimate the arrival time of runner hence ease the user to standby receive the parcel.

Next, is regarding web security concerns. We do have implemented a security practice during the implementation of the website to prevent the system hacked or attacked by the intruders. For example, we do encrypt our system user's password to prevent abuse from database admin and any intruders. Users are likely to use Two-Factor authentication to provide a better layer of security for the system.

Third is the system is limited only to university organizations. The main target users for this system are only students and university staff but users are confident that if the system is completed for public use, it is better to share this initiative with the public.

In terms of strengths, the system's user interface is a responsive design across a wide range of screen sizes. This capability is very useful for every type of device that the user uses. The interface of the website will be determined based on the width of the screen device so that users can access the system easily on a computer or even from a mobile phone.

Next up is the ordering experience is a quick and easy ordering process. The ordering process is easy. On the main page, users are instructed to select the vehicle type then need to enter the pick-up and drop-off locations. After that, enter the contact details, time, and date for the runner to take the order. The process is so simple and neat to make it easier for users to place orders.

Finally, the system is integrated with SMS functionality. This SMS function is used to provide information to users about the latest status with respect to profile information or orders. Each time a new user registers an account, the system will send an SMS message reminding the user to verify their email first before accessing the system. Also, when the runner has finished taking the order from the pickup address the system will send an SMS to inform the recipient to be ready that the runner will arrive soon and when the runner has finished delivering the order the system will also send a message to the recipient stating that the order has been successfully delivered.

7.2 Propositions for Improvement

After observing the system testing process, we have identified several suggestions for improvements that can be implemented to this system.

One of them is, progressive web application (PWA) for smartphone support. Today, the world has advanced in line with smartphone technology and the average person has a personal smartphone to carry out their daily affairs. With the PWA, users can install this website as an application on their phone's home page. This PWA is the same and works like a native application for an operating system for the smartphone, but it is still a website that is displayed as an application so that users can feel like it is a real application. The advantage of this is that users now no longer have to open the website and type in the website address to access the system anymore. All they have to do to access the system is with one click on the application icon on the home page of their smartphone.

The next suggestion is to implement a runner tracking feature. As discussed in the system weakness section, this feature is particularly useful for making it easier for senders and receivers to estimate the runner’s arrival time to their destination. This function is more effective for those living in high -rise house blocks.

Finally, the reorganization of the system interface and user experience to attract users to continue using this application in the future. Improvements need to be made to the system interface to keep the design always up to date. From the aspect of user experience, the process of creating orders can be further simplified so as not to confuse users, especially those who are older or have no experience in the field of information technology.

7.3 Project Contribution

The main contribution of this project is to the community of Universiti Teknikal Malaysia Melaka by providing a same-day delivery application called UTeM Xpress.

The first contribution is based on the objective we want to achieve is to implement the delivery of goods outside and inside the university area. This is intended to help students who do not have or are not allowed to bring a private vehicle into the campus area. These students must find it difficult for them to buy or deliver their goods from one location to another. In fact, they have to take a bus or use a third-party service which will also incur high costs to pay for this service.

Next, provide a platform to allow students to generate money while within the university campus area. This can obviously help students supplement their pocket money for daily expenses on campus. The system only opens opportunities for students to become runners who will deliver orders placed by customers. Money from ordered fares will be the main source of student pocket money. Students do not necessarily need to have a vehicle because they can also be runners without a vehicle.

Finally, the reduction in the number of movements within the university area during the pandemic season. With this system, users only need to order through their computer or smartphone from home or residential college. Only runners move to pick up and deliver customer orders and this method can clearly reduce the spread of epidemics among each other. Runners can also send goods without contact with the customer by placing the goods in front of the consumer’s door if the consumer agrees to do the receipt without contact.

7.4 Conclusion

This project has met all the specifications of the conditions stated in the objectives section. The project is considered successful and further developments and improvements should be made in the future. I am so thankful that finally the development of this project has been completed. This system is important to use in the university community because it simplifies the process of placing an order and strengthens the relationship between students and lecturers at the educational level. I hope this project can go a step further and can be used in industry to simplify their work process

REFERENCES

L. (n.d.). *Fastest Delivery Service and Instant Courier | Lalamove Malaysia*. Lalamove. Retrieved August 26, 2021, from https://www.lalamove.com/en-my/

Bungkusit. (n.d.). *Bungkusit*. Malaysia’s Fastest On-Demand Delivery. Retrieved August 26, 2021, from https://bungkusit.com.my/

TutorialsPoint. (n.d.). *SDLC - Spiral Model*. SDLC - Spiral Model. Retrieved August 26, 2021, from https://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm

MrSpeedy. (n.d.). *Same Day Delivery in Klang Valley | MrSpeedy MY*. Fastest Same Day Delivery in Klang Valley. Retrieved August 26, 2021, from https://mrspeedy.my/express-delivery/same-day-delivery-malaysia

Nation, E. (2019, April 29). What is the Same-Day Delivery in E-commerce? ECN | E-Commerce Nation. https://www.ecommerce-nation.com/same-day-delivery-ecommerce/#:%7E:text=What%20is%20same%20day%20delivery%3F,-The%20same%20day&text=It%20is%20defined%20as%20the,morning%20of%20the%20next%20day.

Hamilton, T. (2021, August 16). *How to Write Test Cases: Sample Template with Examples*. How to Write Test Cases: Sample Template with Examples. https://www.guru99.com/test-case.html

Hamilton, T. (2021b, August 16). *Integration Testing: What is, Types, Top Down & Bottom Up Example*. Integration Testing: What Is, Types, Top Down & Bottom Up Example. https://www.guru99.com/integration-testing.html

Hamilton, T. (2021a, August 14). *What is BLACK Box Testing? Techniques, Example & Types*. What Is BLACK Box Testing? Techniques, Example & Types. https://www.guru99.com/black-box-testing.html

APPENDICES

APPENDIX A

Login Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
UML1	Check user login with valid credentials.	User already registered an account.	1. Go to site https://utemxpress.herokuapp.com/login 2. Enter UTeM email address. 3. Enter password. 4. Click login button.	UTeM email address: **B031920030@student.utem.edu.my** password: **123456789**	User should be redirected to home page.
UML2	Check user login with invalid credentials.	User already registered an account or not yet.	1. Go to site https://utemxpress.herokuapp.com/login 2. Enter wrong UTeM email address. 3. Enter wrong password. 4. Click login button.	UTeM email address: **B031920099@student.utem.edu.my** password: **123456789abcd123**	The system should display error message "User not found, if you are runner

					please tick 'I am runner' checkbox below”
UML3	Check user login with empty credentials.	-	1. Go to site https://utemxpress.herokuapp.com/login 2. Leave the UTeM email address empty. 3. Leave the password empty. Click login button.	UTeM email address: - password: -	Error message “The email field is required.” is displayed under email address input Error message “The password field is required.” is displayed under

					password input.
UML4	Check user login with role runner and check the ‘I am runner’ checkbox.	User already registered an account with runner role.	1. Go to site https://utemxpress.herokuapp.com/login 2. Enter UTeM email address 3. Enter password 4. Check the ‘I am runner’ checkbox 5. Click Login	UTeM email address: **B031920030@student.utem.edu.my** password: **123456789**	The user will be redirected to runner home page.
UML5	Check user login with role runner and uncheck the ‘I am runner’ checkbox.	User already registered an account with runner role.	1. Go to site https://utemxpress.herokuapp.com/login 2. Enter UTeM email address 3. Enter password 4. Uncheck the ‘I am runner’ checkbox 5. Click Login	UTeM email address: **B031920030@student.utem.edu.my** password: **123456789**	Error message “User not found, if you are runner please tick 'I am runner' checkbox below” will be displayed

					under email address input.

Register Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
UMR1	Check user registration with non-existing data.	User is not registered yet.	1. Go to site http://localhost:8000/register 2. Choose role. 3. Enter UTeM email address. 4. Enter phone number. 5. Enter password. 6. Enter confirm password. 7. Click register.	Role: **student** UTeM email address: **B031920099@student.utem.edu.my** Phone number: **01125458495** Password: **123456789** Confirm password: **123456789**	The system will validate the users input and send email verification link to user's email.

UMR2	Verify email address.	User has been register but not verify email yet.	1. Go to email 2. Check inbox with title 'Verify Email Address'. 3. Click Verify Email Address button.	-	User will be redirected to home page.
UMR3	Check user registration with existing data.	User is already registered.	1. Go to site http://localhost:8000/register 2. Choose exist role. 3. Enter existing UTeM email address. 4. Enter existing phone number. 5. Enter password. 6. Enter confirm password. 7. Click register button.	Role: **student** UTeM email address: **B031920032@student.utem.edu.my** Phone number: **01120624714** Password: **123456789** Confirm password: **123456789**	Error message "Student with the same role already exist !" will be displayed under email address input.
UMR4	Check user registration with non-existing data	User is not register yet.	1. Go to site http://localhost:8000/register 2. Choose role. 3. Enter UTeM email address.	Role: **student** UTeM email address: **B031920017@student.utem.edu.my**	Error message "Phone number already been used !" will be displayed

	but with existing phone number.		4. Enter existing phone number. 5. Enter password. 6. Enter confirm password. 7. Click register button.	Phone number: **01120624714** Password: **123456789** Confirm password: **123456789**	under phone number input.
UMR5	Check user registration with empty input.	-	1. Go to site http://localhost:8000/register 2. let role be default option. 3. Leave the UTeM email address empty. 4. Leave the phone number empty. 5. Leave the password empty. 6. Leave the confirm password empty. 7. Click register button.	Role: **student** UTeM email address: - Phone number: - Password: - Confirm password: -	"please fill out the field" error message will be shown below the empty input.
UMR6	Check user registration	User is not register yet.	1. Go to site http://localhost:8000/register	Role: **student**	Error message "Confirm Password

	with confirm password value does not match with password values.		2. let role be default option. 3. enter UTeM email address. 4. enter the phone number. 1. enter the password. 2. enter the confirm password values different with password values. 3. Click register button.	UTeM email address: **B031920099@student.utem.edu.my** Phone number: **01120624714** Password: **123456789** Confirm password: **123456789abcd123**	are not matched with password" will be displayed under confirm password input and the register button will be disabled.

Add Profile Personal Image Test Case

Test Case ID	**Test Scenario**	**Pre-requisite**	**Test Steps**	**Test Data**	**Expected Result**
UMP1	Upload personal image with image file type.	User has not upload personal image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add Personal Image button 3. Click choose file. 4. Choose image file type (png/jpg).	Personal image: **Image file type (png/jpg/jpeg)**	"Personal image has been added" message displayed, and the uploaded image will be displayed.

			5. Click submit. 6. Click Show Personal Image button.		
UMP2	Upload personal image with empty input.	User has not upload personal image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add Personal Image button 3. Click submit.	Personal image: -	Submit button will be disabled
UMP3	Upload personal image with not image file type.	User has not upload personal image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add Personal Image button 3. Click choose file. 4. Choose non image file type (mp3/mp4).	Personal image: **Non-Image file type (mp3/mp4/pdf)**	"Alert Files uploaded in non-picture format" will be displayed.

Add Front IC Image Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
UMF1	Upload front IC image with	User has not upload front	1. Go to site https://utemxpress.herokuapp.com/staff/profile	IC front image:	"Personal image has been added" message

	image file type.	IC image yet.	2. Click Add IC Image button 3. Click choose file. 4. Choose image file type (png/jpg). 5. Click submit. 6. Click Show Personal Image button.	**Image file type (png/jpg/jpeg)**	displayed, and the uploaded image will be displayed.
UMF2	Upload front IC image with empty input	User has not upload front IC image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add IC Image button 3. Click submit.	IC front image: -	Submit button will be disabled
UMF3	Upload front IC image with not image file type.	User has not upload front IC image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add IC Image button 3. Click choose file. 4. Choose non image file type (mp3/mp4).	IC front image: **Non-Image file type (mp3/mp4/pdf)**	"Alert Files uploaded in non-picture format" will be displayed.

Add Back IC Image Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
UMB1	Upload back IC image with image file type.	User has not upload back IC image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add IC Image button 3. Click choose file. 4. Choose image file type (png/jpg). 5. Click submit. 6. Click Show Personal Image button.	IC back image: **Image file type (png/jpg/jpeg)**	"Personal image has been added" message displayed, and the uploaded image will be displayed.
UMB2	Upload back IC image with empty input.	User has not upload back IC image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add IC Image button. 3. Click submit.	IC back image: -	Submit button will be disabled
UMB3	Upload back IC image with not image file type.	User has not upload back IC image yet.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click Add IC Image button 3. Click choose file.	IC back image: **Non-Image file type (mp3/mp4/pdf)**	"Alert Files uploaded in non-picture format" will be displayed.

			1. Choose non image file type (mp3/mp4). 2. Click submit.		

Logout Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
UMO1	Logout from active session.	User already logged in.	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Click logout button.	-	User will be redirected to index page.

Create Order on Demand Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMD1	Create on demand order.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Enter pick up location. 4. Enter drop off location. 5. Choose any additional services. 6. Click deliver now button. 7. Enter receiver's name. 8. Enter receiver's phone number. 9. Enter note to your runner. 10. Click confirm order.	Pick up location: **Any location** Drop off location: **Any location** Additional services: **Any service** Receiver's name: **Any name**	After step 6 user will be redirected to Delivery Date & Contact Information page. After step 10 user will be redirected to order details page.

				Receiver's phone number: **Any phone number** Note to your runner: **Any note**	
OMD2	Crete on demand order with empty pickup and drop-off input.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Leave the pickup location empty. 4. Leave the drop off location empty. 5. Leave the additional services empty. 6. Click deliver now button.	Pick up location: - Drop off location: - Additional services: -	"please fill out this field" message is displayed on empty input.

OMD3	Create on demand order with empty receiver’s name and phone input.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Enter pick up location. 4. Enter drop off location. 5. Choose any additional services. 6. Click deliver now button. 7. Leave the receiver’s name empty. 8. Leave the receiver’s phone number empty. 9. Leave the Enter note to your runner empty. 10. Click confirm order.	Pick up location: **Any location** Drop off location: **Any location** Additional services: **Any service** Receiver’s name: - Receiver’s phone number: -	After step 6 user will be redirected to order receiver’s detail page. After step 10 user will be redirected to order details page.

				Note to your runner: -	

Create Order for Later Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMF1	Create for later order.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Enter pick up location. 4. Enter drop off location. 5. Choose any additional services. 6. Click scheduled for later button. 7. Enter date 8. Enter time 9. Enter receiver’s name.	Pick up location: **Any location** Drop off location: **Any location** Additional services: **Any services** Date:	After step 6 user will be redirected to Delivery Date & Contact Information page. After step 12 user will be redirected to order details page.

			10. Enter receiver's phone number. 11. Enter note to your runner. 12. Click confirm order.	**Any date** Time: **Any time** Receiver's name: **Any name** Receiver's phone number: **Any phone number** Note to your runner: **Any note**	
OMF2	Create for later order with empty pickup and drop-off input.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Leave the pickup location empty. 4. Leave the drop off location empty.	Pick up location: - Drop off location:	"please fill out this field" message is displayed on empty input.

			5. Leave the additional services empty. 6. Click schedule for later button.	- Additional services: -	
OMF3	Create for later order with empty date, time, receiver's name, and phone input.	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Enter pick up location. 4. Enter drop off location. 5. Choose any additional services. 6. Click deliver now button. 7. Leave the date empty. 8. Leave the time empty. 9. Leave the receiver's name empty. 10. Leave the receiver's phone number empty. 11. Leave the Enter note to your runner empty. 12. Click confirm order.	Pick up location: **Any location** Drop off location: **Any location** Additional services: **Any services** Date: - Time: - Receiver's name:	After step 6 user will be redirected to order receiver's detail page. On step 12 confirm order button will be disabled.

				- Receiver's phone number: - Note to your runner: -	
OMF4	Create for later order with time that out from range '7AM – 8PM'	User is already logged in,	1. Go to site https://utemxpress.herokuapp.com/staff/profile 2. Choose any type of vehicle. 3. Enter pick up location. 4. Enter drop off location. 5. Choose any additional services. 6. Click deliver now button. 7. Enter date. 8. Enter time.	Pick up location: **Any location** Drop off location: **Any location** Additional services: **Any services** Date: **Any date**	After step 8 the confirm order button will be disabled if the time is out of range.

				Time: **TIME < 7AM or TIME > 8PM**	

Order History Record Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMH1	View order history record	User already logged in.	1. Go to site /student/showOrderRecord 2. Click all tab and click the order list. 3. Click completed tab and click the order list. 4. Click cancelled tab and click the order list.	-	All order history will be display based on its category for each tab and the order details page will be displayed when clicked the order list.

			5. Click walk tab and click the order list. 6. Click motorcycle tab and click the order list. 7. Click car tab and click the order list.		

View Order Route Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMR1	View order route	User already logged in.	1. Go to site /student/showOrderRecord. 2. Click all tab, click the order list, and click view route button. 3. Click completed tab, click the order list, and click view route button. 4. Click cancelled tab, click the order list, and click view route button.	-	User will be redirected to route page and google map will be displayed.

			5. Click walk tab, click the order list, and click view route button. 6. Click motorcycle tab, click the order list, and click view route button. 7. Click car tab, click the order list, and click view route button.		

Cancel Order Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMC1	Cancel an order	Order has been created earlier and status is still waiting.	1. Go to site student/activeOrder. 2. Click an order in the list in On-Demand / For-Later tab. 3. Click cancel order button.	-	User will be redirected back to order details page and “Order has successfully cancelled!” will be displayed.

Active Order on Demand Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMA1	View active on demand order	Order on-Demand has been created earlier by user.	1. Go to site student/activeOrder. 2. Click On-Demand tab. 3. Click an order in the list	-	The user will be redirected to order details page that have waiting status.

Active Order for Later Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMB1	View active for laer order	Order for-Later has been created earlier by user.	1. Go to site student/activeOrder. 2. Click For-Later tab. 3. Click an order in the list	-	The user will be redirected to order details page that have waiting status.

Add Runner to Favorite Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
OMT1	Add runner to favorite list.	Order has been created and status is set to completed.	1. Go to site student/home. 2. Create an order 3. Let the runner update the order status to completed. 4. Click add runner to favorite button.	-	The user will be redirected to order details page and "Runner has successfully added to favorite!" message will be displayed

Order Listing Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
JMO1	View order listing from runner side.	Runner already logged in and turn the button to on.	1. Go to site /runner/home. 2. Click radio button to on order listing (initial state is off).	-	The system will display list of waiting order in on-demand and for later tab.

Past Order Record Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
JMP1	View completed order taken by runner.	Runners are logged in and have completed previous orders.	1. Go to site /runner/completedOrder. 2. Click All tab. 3. Click Completed tab. 4. Click Cancelled tab.	-	The system will display list of order on each tab.

Rate Order and Review Runner Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
RRR1	Check Rate order and review runner.	Order have been created earlier and completed by the runner.	1. Go to site student/home. 2. Create an order.	Rating: **5 stars** Review: **“ok”**	The user will be redirected to order details page and “Order has successfully rated!” message will be displayed

			3. Let the runner update the order status to completed. 4. Select star rating. 5. Enter review input.		
RRR2	Check Rate order and review runner with empty input	Order have been created earlier and completed by the runner.	1. Go to site student/home. 2. Create an order. 3. Let the runner update the order status to completed. 4. Let the star rating empty. 5. Let the review input empty.	Rating: - Review: -	“please fill out this field” message is displayed on review input.

Add Vehicle Information Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
VMV1	Check add vehicle information (Walk / Bicycle)	Runner already registered an account.	1. Go to /runner/profile. 2. Click add vehicle button. 3. Choose vehicle type. 4. Click submit button.	vehicle type: **Walk / Bicycle**	"Vehicle information has been uploaded successfully" message will be displayed.
VMV2	Check add vehicle information (Motorcycle)	Runner already registered an account.	1. Go to /runner/profile. 2. Click add vehicle button. 3. Choose vehicle type. 4. Enter vehicle plate number. 5. Enter vehicle image	vehicle type: **Motorcycle** vehicle plate number: **any plate numbers** vehicle image: **any vehicle image**	"Vehicle information has been uploaded successfully" message will be displayed.

			6. Enter road tax image 7. Click submit button.	road tax image: **any road tax image**	
VMV3	Check add vehicle information (Car)	Runner already registered an account.	1. Go to /runner/profile. 2. Click add vehicle button. 3. Choose vehicle type. 4. Enter vehicle plate number. 5. Enter vehicle image 6. Enter road tax image 7. Click submit	vehicle type: **Motorcycle** vehicle plate number: **any plate numbers** vehicle image: **any vehicle image** road tax image: **any road tax image**	"Vehicle information has been uploaded successfully" message will be displayed.
VMV4	Check add vehicle information (Motorcycle / Car) with non-image file type	Runner already registered an account.	1. Go to /runner/profile. 2. Click add vehicle button.	vehicle type: **Motorcycle** vehicle plate number:	"Alert Files uploaded in non-picture format" will be displayed.

			3. Choose vehicle type. 4. Enter vehicle plate number. 5. Enter vehicle image 6. Enter road tax image 7. Click submit button.	**any plate numbers** vehicle image: **any vehicle non-image type** road tax image: **any road tax non-image type**	
VMV5	Check add vehicle information (Motorcycle / Car) with empty input	Runner already registered an account.	1. Go to /runner/profile. 2. Click add vehicle button. 3. Choose vehicle type. 4. Let the vehicle plate number empty. 5. Let the vehicle image empty.	vehicle type: **Motorcycle** vehicle plate number: - vehicle image: - road tax image: -	The submit button will be disabled.

			6. Let the road tax image empty. 7. Click submit button		

Add License Information Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
VML1	Check add license information.	Runner already registered an account and enter vehicle details.	1. Go to /runner/profile. 2. Click add license button. 3. Enter license front image. 4. Enter license back image. 5. Click submit button.	license front image: **any license image** license back image: **any license image**	"License image has been uploaded successfully" message will be displayed.

VML2	Check add license information with empty input.	Runner already registered an account and enter vehicle details.	1. Go to /runner/profile. 2. Click add license button. 3. Enter license front image. 4. Enter license back image. 5. Click submit button.	license front image: - license back image: -	Submit button will be disabled.
VML3	Check add license information with non-image type input.	Runner already registered an account and enter vehicle details.	1. Go to /runner/profile. 2. Click add license button. 3. Enter license front image. 4. Enter license back image.	license front image: **any license non-image type** license back image:	"Alert Files uploaded in non-picture format" will be displayed.

			5. Click submit button.	**any license non-image type**	

Earnings Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
RME1	Check runner earnings.	Runner already register an account and have completed previous order.	1. Go to site /runner/earning.	-	Graph diagram that depicts the number of earnings and order taken by the runner is displayed.

Statistic Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
RMS1	Check runner statistic.	Runner already register an account and have completed previous order.	1. Go to site /runner/earning.	-	The system will display the summary of earnings earn by the runner including total order, total earning, fleets, and rating.

Update Vehicle Information Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
RMU1	Check update vehicle information (Walk / Bicycle)	User already have already entered the vehicle information.	1. Go to /runner/profile. 2. Click update vehicle button. 3. Choose vehicle type. 4. Click submit button.	vehicle type: **Walk / Bicycle**	"Vehicle information has been updated successfully" message will be displayed.
RMU2	Check update vehicle information (Motorcycle)	User already have already entered the vehicle information.	1. Go to /runner/profile. 2. Click update vehicle button. 3. Choose vehicle type.	vehicle type: **Motorcycle** vehicle plate number: **any plate numbers** vehicle image:	"Vehicle information has been updated successfully" message will be displayed.

			4. Enter vehicle plate number. 5. Enter vehicle image 6. Enter road tax image 7. Click submit button.	**any vehicle image** road tax image: **any road tax image**	
RMU3	Check update vehicle information (Car)	User already have already entered the vehicle information.	1. Go to /runner/profile. 2. Click update vehicle button. 3. Choose vehicle type. 4. Enter vehicle plate number. 5. Enter vehicle image. 6. Enter road tax image.	vehicle type: **Motorcycle** vehicle plate number: **any plate numbers** vehicle image: **any vehicle image** road tax image: **any road tax image**	“Vehicle information has been updated successfully” message will be displayed.

			7. Click submit button.		
RMU4	Check update vehicle information (Motorcycle / Car) with non-image file type	User already have already entered the vehicle information.	1. Go to /runner/profile. 2. Click update vehicle button. 3. Choose vehicle type. 4. Enter vehicle plate number. 5. Enter vehicle image. 6. Enter road tax image. 7. Click submit button.	vehicle type: **Motorcycle** vehicle plate number: **any plate numbers** vehicle image: **any vehicle non-image type** road tax image: **any road tax non-image type**	“Alert Files uploaded in non-picture format” will be displayed.
RMU5	Check update vehicle information (Motorcycle / Car) with empty input	User already have already entered the vehicle information.	1. Go to /runner/profile.	vehicle type: **Motorcycle**	The submit button will be disabled.

			2. Click update vehicle button. 3. Choose vehicle type. 4. Let the vehicle plate number empty. 5. Let the vehicle image empty. 6. Let the road tax image empty. 7. Click submit button.	vehicle plate number: - vehicle image: - road tax image: -	

Update License Information Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
RML1	Check update license information.	User already have already entered the license information.	1. to /runner/profile. 2. Click update license button. 3. Enter license front image. 4. Enter license back image. 5. Click submit button.	license front image: **any license image** license back image: **any license image**	"License image has been updated successfully" message will be displayed.
RML2	Check update license information with empty input.	User already have already entered the license information.	1. Go to /runner/profile. 2. Click update license button. 3. Enter license front image.	license front image: - license back image: -	Submit button will be disabled.

			4. Enter license back image. 5. Click submit button		
RML3	Check update license information with non-image type input.	User already have already entered the license information.	1. Go to /runner/profile. 2. Click update license button. 3. Enter license front image. 4. Enter license back image. 5. Click submit button.	license front image: **any license non-image type** license back image: **any license non-image type**	"Alert Files uploaded in non-picture format" will be displayed.

Activate & Deactivate User Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
AMA1	Check activates user.	User must be already existing, registered and status must be unavailable.	1. Go to site /admin/manageUser. 2. Click activate button. 3. Click activate! button.	-	"User has successfully re-activated!" message is displayed and user status change to AVAILABLE
AMA2	Check activates runner.	User must be already existing, registered and status must be unavailable.	1. Go to site /admin/ manageRunner. 2. Click activate button. 3. Click activate! button.	-	"User has successfully re-activated!" message is displayed and user status change to AVAILABLE
AMA3	Check activates admin.	User must be already existing, registered and status must be unavailable.	1. Go to site /admin/ manageAdmin. 2. Click activate button. 3. Click activate! button.	-	"User has successfully re-activated!" message is displayed and user status change to AVAILABLE
AMA4	Check deactivates user.	User must be already existing, registered and status must be available.	1. Go to site /admin/manageUser.Click 2. deactivate button. 3. Click deactivate! button.	-	"User has successfully de-activated!" message is displayed and user status change to UNAVAILABLE

AMA5	Check deactivates runner.	User must be already existing, registered and status must be available.	1. Go to site /admin/ manageRunner. 2. Click deactivate button. 3. Click deactivate! button.	-	"User has successfully de-activated!" message is displayed and user status change to UNAVAILABLE
AMA6	Check deactivates admin.	User must be already existing, registered and status must be available.	1. Go to site /admin/ manageAdmin. 2. Click deactivate button. 3. Click deactivate! button.	-	"User has successfully de-activated!" message is displayed and user status change to UNAVAILABLE

Add New User Test Case

Test Case ID	Test Scenario	Pre-requisite	Test Steps	Test Data	Expected Result
AMU1	Check adds new user (Student).	User information must be existing and not register yet.	1. Go to site /admin/manageUser. 2. Click add new student button. 3. Enter email address.	Email address: **Matric_no@student.utem.edu.my** IC: **Any 12 numbers** Name:	"New student added" message is displayed.

			4. Enter IC. 5. Enter name. 6. Enter faculty. 7. Click add button.	**Any name** Faculty: **Any faculty**	
AMU2	Check add new user (Staff).	User information must be existing and not register yet.	1. Go to site /admin/manageUser. 2. Click add new staff button. 3. Enter email address. 4. Enter IC. 5. Enter name. 6. Enter designation. 7. Enter faculty. 8. Enter division. 9. Click add button.	Email address: **First_name@utem.edu.my** IC: **Any 12 numbers** Name: **Any name** Designation: **Any designation** Faculty: **Any faculty** Division: **Any division**	“New staff added” message is displayed.
AMU3	Check adds new user (Admin).	User information must be	1. Go to site /admin/manageAdmin.	Email address: **Any email address** IC:	“New admin added” message is displayed, and

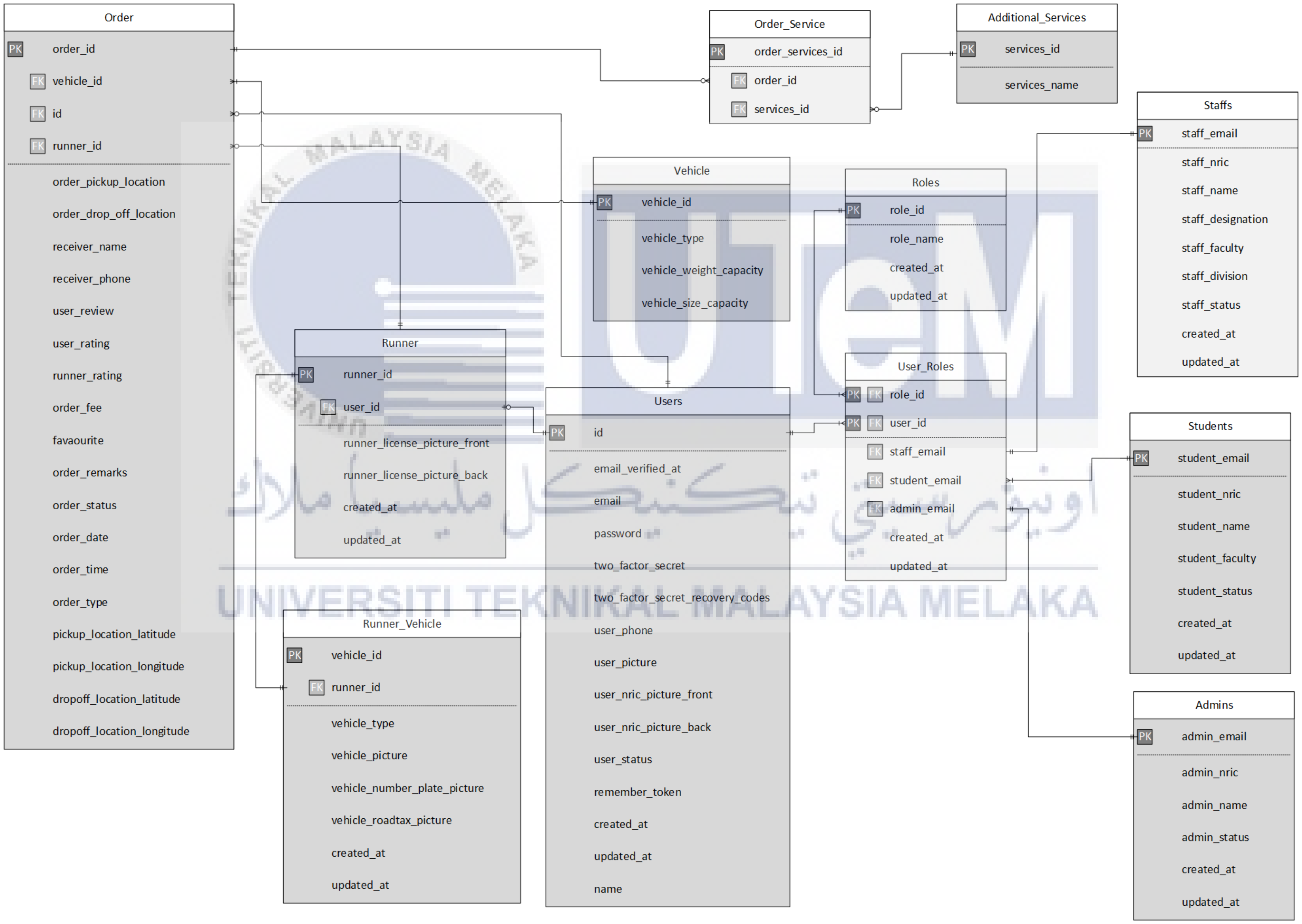
		existing and not register yet.	2. Click add new admin button. 3. Enter email address. 4. Enter IC. 5. Enter name. 6. Click add button.	**Any 12 numbers** Name: **Any name**	admin status set to UNREGISTERED.
AMU4	Check adds new user (Student) with empty input.	User information must be existing and not register yet.	1. Go to site /admin/manageUser. 2. Click add new student button. 3. Let the email address empty. 4. Let the IC empty. 5. Let the name empty. 6. Let the faculty empty. 7. Click add button.	-	“Please fill out this field” message is displayed on empty input.
AMU5	Check adds new user	User information must be	1. Go to site /admin/manageUser.	-	“Please fill out this field” message is displayed on empty input.

	(Staff) with empty input.	existing and not register yet.	2. Click add new staff button. 3. Let the email address empty. 4. Let the IC empty. 5. Let the name empty. 6. Let the designation empty. 7. Let the faculty empty. 8. Let the division empty. 9. Click add button.		
AMU6	Check adds new user (Admin) with empty input.	User information must be existing and not register yet.	1. Go to site /admin/manageAdmin. 2. Click add new admin button. 3. Let the email address empty. 4. Let the IC empty. 5. Let the name empty.	-	“Please fill out this field” message is displayed on empty input.

			6. Click add button.		

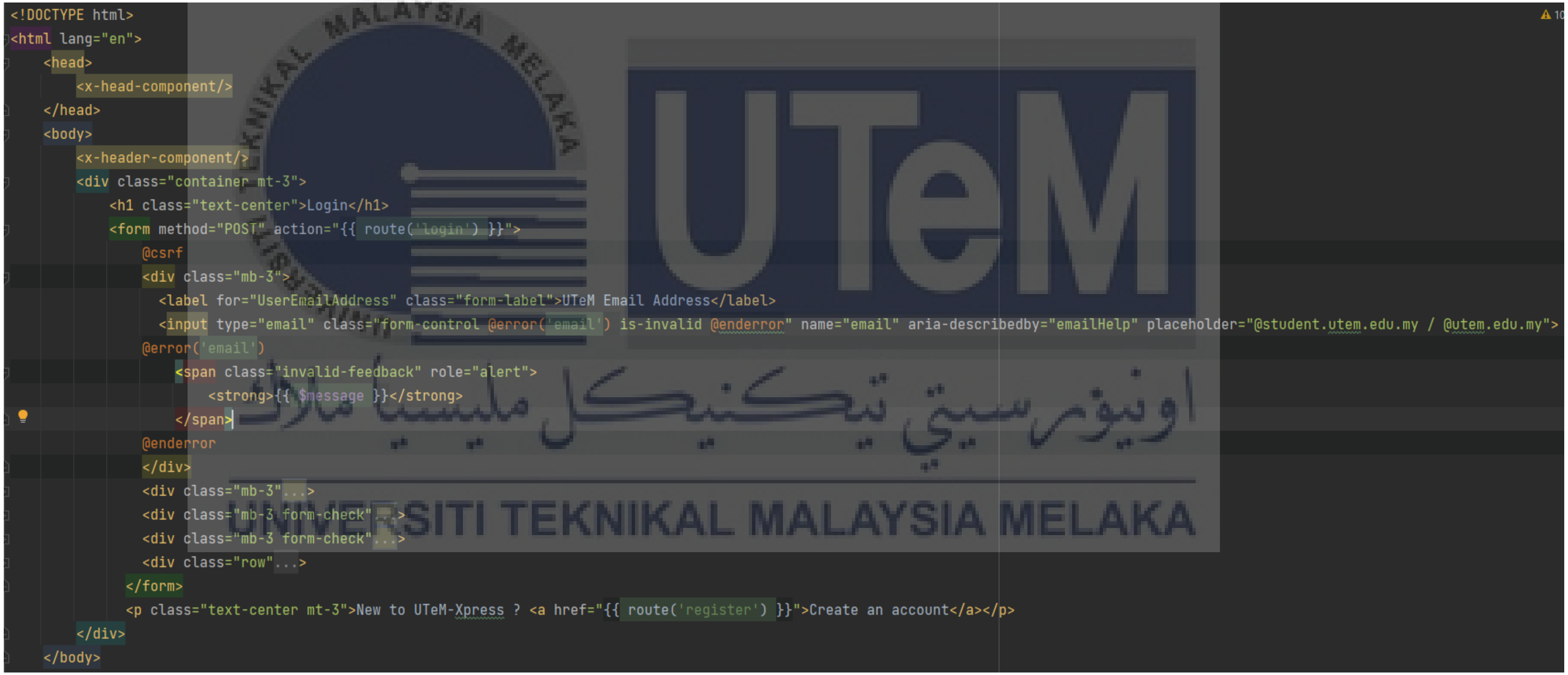
APPENDIX B

Entity Relationship Diagram


Order
PK order_id
FK vehicle_id
FK id
FK runner_id
order_pickup_location
order_drop_off_location
receiver_name
receiver_phone
user_review
user_rating
runner_rating
order_fee
favaourite
order_remarks
order_status
order_date
order_time
order_type
pickup_location_latitude
pickup_location_longitude
dropoff_location_latitude
dropoff_location_longitude
Order_Service
PK order_services_id
FK order_id
FK services_id
Additional_Services
PK services_id
services_name
Staffs
PK staff_email
staff_nric
staff_name
staff_designation
staff_faculty
staff_division
staff_status
created_at
updated_at
Vehicle
PK vehicle_id
vehicle_type
vehicle_weight_capacity
vehicle_size_capacity
Roles
PK role_id
role_name
created_at
updated_at
Runner
PK runner_id
FK user_id
runner_license_picture_front
runner_license_picture_back
created_at
updated_at
User_Roles
PK FK role_id
PK FK user_id
FK staff_email
FK student_email
FK admin_email
created_at
updated_at
Users
PK id
email_verified_at
email
password
two_factor_secret
two_factor_secret_recovery_codes
user_phone
user_picture
user_nric_picture_front
user_nric_picture_back
user_status
remember_token
created_at
updated_at
name
Students
PK student_email
student_nric
student_name
student_faculty
student_status
created_at
updated_at
Runner_Vehicle
PK vehicle_id
FK runner_id
vehicle_type
vehicle_picture
vehicle_number_plate_picture
vehicle_roadtax_picture
created_at
updated_at
Admins
PK admin_email
admin_nric
admin_name
admin_status
created_at
updated_at

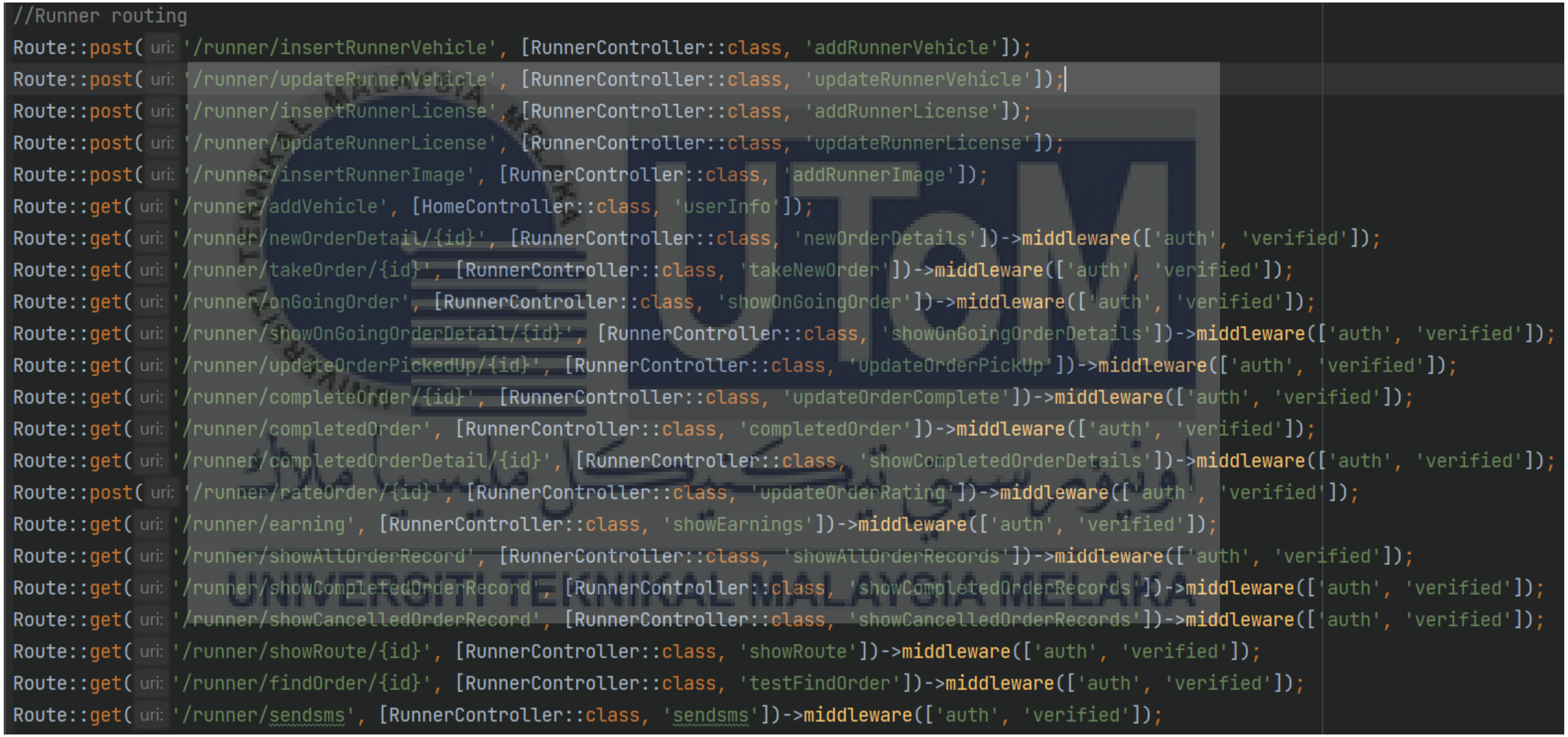
APPENDIX C

Login HTML



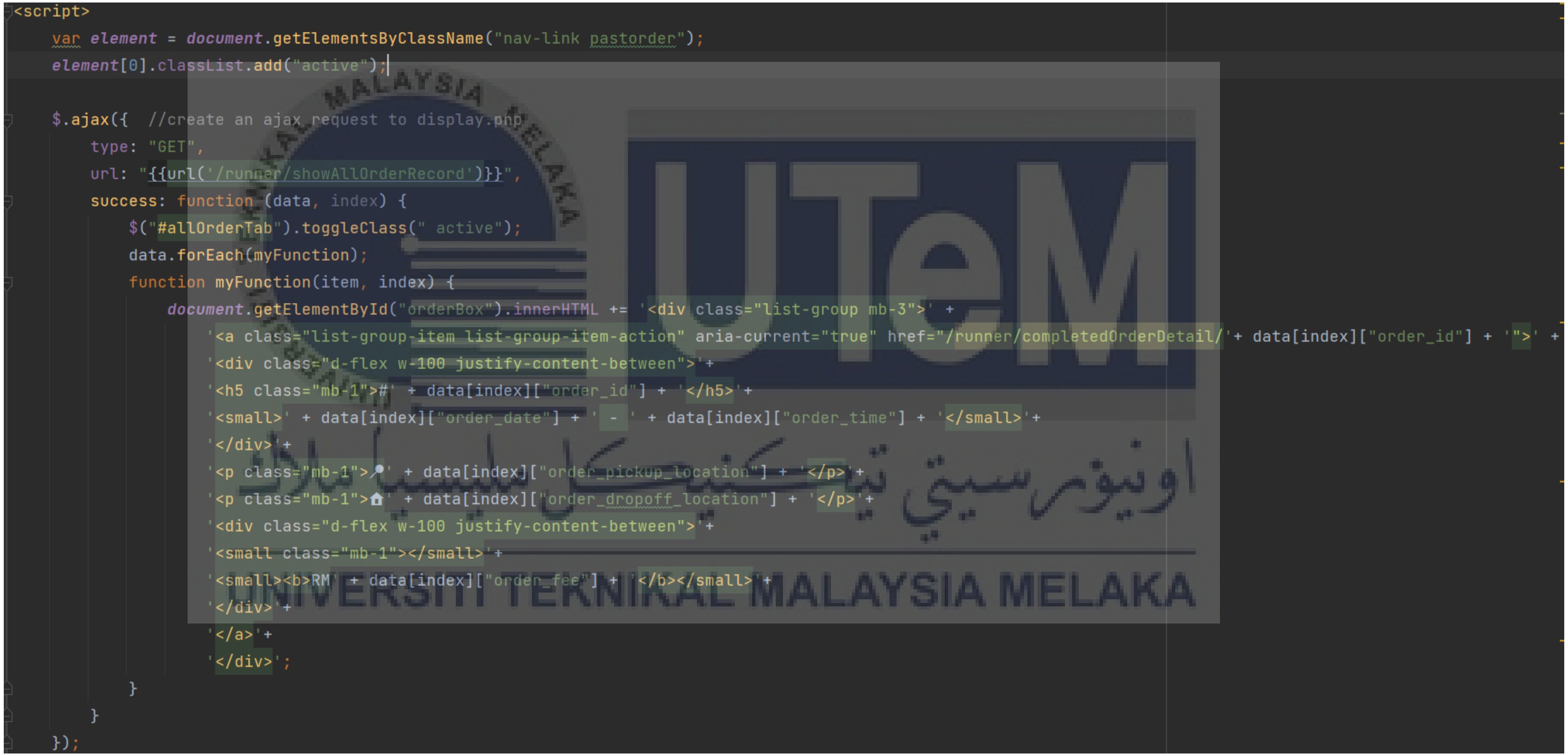
```
<!DOCTYPE html>
<html lang="en">
    <head>
        <x-head-component/>
    </head>
    <body>
        <x-header-component/>
        <div class="container mt-3">
            <h1 class="text-center">Login</h1>
            <form method="POST" action="{{ route('login') }}">
                @csrf
                <div class="mb-3">
                  <label for="UserEmailAddress" class="form-label">UTeM Email Address</label>
                  <input type="email" class="form-control @error('email') is-invalid @enderror" name="email" aria-describedby="emailHelp" placeholder="@student.utem.edu.my / @utem.edu.my">
                @error('email')
                    <span class="invalid-feedback" role="alert">
                        <strong>{{ $message }}</strong>
                    </span>
                @enderror
                </div>
                <div class="mb-3"...>
                <div class="mb-3 form-check"...>
                <div class="mb-3 form-check"...>
                <div class="row"...>
              </form>
              <p class="text-center mt-3">New to UTeM-Xpress ? <a href="{{ route('register') }}">Create an account</a></p>
        </div>
    </body>
```

Web routing



```
//Runner routing
Route::post( uri: '/runner/insertRunnerVehicle', [RunnerController::class, 'addRunnerVehicle']);
Route::post( uri: '/runner/updateRunnerVehicle', [RunnerController::class, 'updateRunnerVehicle']);
Route::post( uri: '/runner/insertRunnerLicense', [RunnerController::class, 'addRunnerLicense']);
Route::post( uri: '/runner/updateRunnerLicense', [RunnerController::class, 'updateRunnerLicense']);
Route::post( uri: '/runner/insertRunnerImage', [RunnerController::class, 'addRunnerImage']);
Route::get( uri: '/runner/addVehicle', [HomeController::class, 'userInfo']);
Route::get( uri: '/runner/newOrderDetail/{id}', [RunnerController::class, 'newOrderDetails'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/takeOrder/{id}', [RunnerController::class, 'takeNewOrder'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/onGoingOrder', [RunnerController::class, 'showOnGoingOrder'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/showOnGoingOrderDetail/{id}', [RunnerController::class, 'showOnGoingOrderDetails'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/updateOrderPickedUp/{id}', [RunnerController::class, 'updateOrderPickUp'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/completeOrder/{id}', [RunnerController::class, 'updateOrderComplete'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/completedOrder', [RunnerController::class, 'completedOrder'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/completedOrderDetail/{id}', [RunnerController::class, 'showCompletedOrderDetails'])->middleware(['auth', 'verified']);
Route::post( uri: '/runner/rateOrder/{id}', [RunnerController::class, 'updateOrderRating'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/earning', [RunnerController::class, 'showEarnings'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/showAllOrderRecord', [RunnerController::class, 'showAllOrderRecords'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/showCompletedOrderRecord', [RunnerController::class, 'showCompletedOrderRecords'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/showCancelledOrderRecord', [RunnerController::class, 'showCancelledOrderRecords'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/showRoute/{id}', [RunnerController::class, 'showRoute'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/findOrder/{id}', [RunnerController::class, 'testFindOrder'])->middleware(['auth', 'verified']);
Route::get( uri: '/runner/sendsms', [RunnerController::class, 'sendsms'])->middleware(['auth', 'verified']);
```

AJAX Get Request Code



```
<script>
    var element = document.getElementsByClassName("nav-link pastorder");
    element[0].classList.add("active");

    $.ajax({  //create an ajax request to display.php
        type: "GET",
        url: "{{url('/runner/showAllOrderRecord')}}",
        success: function (data, index) {
            $("#allOrderTab").toggleClass(" active");
            data.forEach(myFunction);
            function myFunction(item, index) {
                document.getElementById("orderBox").innerHTML += '<div class="list-group mb-3">' +
                    '<a class="list-group-item list-group-item-action" aria-current="true" href="/runner/completedOrderDetail/'+ data[index]["order_id"] + '">' +
                    '<div class="d-flex w-100 justify-content-between">'+
                    '<h5 class="mb-1">#' + data[index]["order_id"] + '</h5>'+
                    '<small>' + data[index]["order_date"] + ' - ' + data[index]["order_time"] + '</small>'+
                    '</div>'+
                    '<p class="mb-1">📍' + data[index]["order_pickup_location"] + '</p>'+
                    '<p class="mb-1">🏠' + data[index]["order_dropoff_location"] + '</p>'+
                    '<div class="d-flex w-100 justify-content-between">'+
                    '<small class="mb-1"></small>'+
                    '<small><b>RM' + data[index]["order_fee"] + '</b></small>'+
                    '</div>'+
                    '</a>'+
                    '</div>';
            }
        }
    });
```