RAZORREADY

MUHAMMAD HAFIZ LUKMAN BIN NURI ANSARIZAM

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

RAZORREADY

MUHAMMAD HAFIZ LUKMAN BIN NURI ANSARIZAM

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

RazorReady

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

STUDENT : _____ _____ Date : 1/9/2024
(MUHAMMAD HAFIZ LUKMAN BIN NURI ANSARIZAM)

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 : _____ Ummi Raba'ah ____ Date : 1/9/2024
(TS. DR. UMMI RABAAH BINTI HASHIM)

DEDICATION

This project report is dedicated to all the individuals who have supported and inspired me throughout this journey.

To my families and friends, who have been my pillars of strength, offering unwavering love, encouragement, and understanding. Your belief in me has been a constant source of motivation.

To my mentors and lectures, who have guided me with their wisdom, knowledge, and expertise. Your invaluable guidance has shaped my skills and pushed me to excel.

Lastly, to myself, for the determination, perseverance, and passion I have poured into this project. I have overcome challenges, learned from failures, and grown both personally and professionally.

This project report is a tribute to all of you. Thank you for being a part of this incredible journey and for being the driving force behind my achievements. Your support and belief in me have made all the difference.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to everyone who has contributed to the successful completion of this project. Without their support, guidance, and assistance, this endeavor would not have been possible.

First and foremost, I would like to thank our project supervisor, Ts. DR. UMMI RABAAH BINTI HASHIM, for their valuable guidance and constant support throughout the project. Their expertise, insights, and constructive feedback have been instrumental in shaping work and pushing me to achieve my best.

I would also like to express my appreciation to my friends and classmates who have supported me throughout the project. Their encouragement, discussions, and exchange of ideas have been invaluable in refining my concepts and approaches.

Furthermore, I would like to acknowledge the contribution of my families for their unwavering support, understanding, and encouragement throughout this project. Their belief in me has been a constant source of motivation, and I are truly grateful for their presence in our lives.

Lastly, I would like to thank all the resources, references, and organizations that have provided me with the necessary materials, data, and information to conduct my research and complete this project successfully.

I acknowledge the collective efforts of everyone involved in this project, and I extend my sincere gratitude to each and every individual who has played a part, directly or indirectly, in making this project a reality.

Thank you all for your unwavering support, guidance, and contributions.

ABSTRACT

RazorReady is a mobile application designed to revolutionize the barbershop experience by streamlining the appointment booking process, enhancing communication between barbers and clients, and providing a platform for discovering new styles and trends in grooming. The primary problems addressed include the inefficiency of traditional booking methods, poor communication channels, and the lack of a centralized platform for style inspiration. RazorReady offers a user-friendly interface that allows customers to easily schedule appointments, receive real-time notifications, and explore a curated gallery of haircut styles. For barbers, the app provides tools to manage schedules, track appointments, and maintain effective communication with clients. The research process involved analyzing current market needs, user behavior, and technological trends to develop a comprehensive solution that improves user experience, increases efficiency, enhances customer-barber relationships, and boosts business performance. The results obtained from initial testing indicate significant improvements in appointment management, customer satisfaction, and business operations. RazorReady is set to redefine the barbershop experience by leveraging technology to create a convenient, personalized, and inspiring platform for both customers and barbers.

ABSTRAK

RazorReady adalah aplikasi mudah alih yang direka untuk merevolusikan pengalaman di kedai gunting rambut dengan mempermudah proses tempahan janji temu, meningkatkan komunikasi antara tukang gunting rambut dan pelanggan, serta menyediakan platform untuk menemui gaya dan trend baru dalam dandanan. Masalah utama yang ditangani termasuk ketidakcekapan kaedah tempahan tradisional, saluran komunikasi yang lemah, dan kekurangan platform pusat untuk inspirasi gaya. RazorReady menawarkan antara muka mesra pengguna yang membolehkan pelanggan dengan mudah menjadualkan janji temu, menerima pemberitahuan masa nyata, dan meneroka galeri gaya rambut yang dikurasi. Bagi tukang gunting rambut, aplikasi ini menyediakan alat untuk mengurus jadual, menjejaki janji temu, dan mengekalkan komunikasi yang berkesan dengan pelanggan. Proses penyelidikan melibatkan analisis keperluan pasaran semasa, tingkah laku pengguna, dan trend teknologi untuk membangunkan penyelesaian komprehensif yang meningkatkan pengalaman pengguna, meningkatkan kecekapan, memperkukuh hubungan pelanggan-tukang gunting rambut, dan meningkatkan prestasi perniagaan. Hasil yang diperoleh daripada ujian awal menunjukkan peningkatan ketara dalam pengurusan janji temu, kepuasan pelanggan, dan operasi perniagaan. RazorReady bersedia untuk mentakrifkan semula pengalaman di kedai gunting rambut dengan memanfaatkan teknologi untuk mencipta platform yang mudah, diperibadikan, dan menginspirasi untuk pelanggan dan tukang gunting rambut.

TABLE OF CONTENTS

PAGE

DEDICATION ..... **III**

ACKNOWLEDGEMENTS ..... **IV**

ABSTRACT ..... **V**

ABSTRAK ..... **VI**

TABLE OF CONTENTS ..... **VII**

LIST OF TABLES ..... **XII**

LIST OF FIGURES ..... **XIII**

LIST OF ABBREVIATIONS ..... **XIV**

CHAPTER 1: INTRODUCTION **1**

1.1 Introduction 1

1.2 Problem Statement 2

1.3 Objectives 2

1.4 Scope 3

1.5 Project Significant 4

1.6 Expected Outcome 5

1.7 Conclusion 5

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY **6**

2.1 Introduction. 6

2.2 Facts and Finidng 7

2.2.1 Domain 7

2.2.2 Existing Application 8

2.2.3 Comparison between RazorReady and existing application 11

2.3 Project Methodology 12

2.4 Project Requirements 15

2.4.1 Software Requirements 15

2.4.2 Hardware Requirements 16

2.5 Gantt Chart 17

2.6 Conclusion 18

CHAPTER 3: ANALYSIS 19

3.1 Introduction 19

3.2 Problem Analysis 19

3.2.1 Flowchart 19

3.2.2 ERD Diagram 25

3.3 Requirement Analysis 26

3.3.1 Data Requirement 26

3.3.2 Funtional Requirement 29

3.3.2.1 User Registration and Login 29

3.3.2.2 Appointment Booking 30

3.3.2.3 Appointment Management 31

3.3.2.4 Barber Profile 33

3.3.3 Non-Functional Requirement 34

3.3.3.1 Usability 34

3.3.3.2 Reliability 35

3.3.3.3 Security 37

3.3.3.4 Performance 39

3.3.3.5 Scalability 41

3.3.3.6 Maintainability 42

3.3.4 Other Requirement 45

3.4 Conclusion. 48

CHAPTER 4: DESIGN 49

4.1 Introduction 49

4.2 High Level Design 49

4.2.1 System Architecture 50

4.2.2 User Interface Design 51

4.2.3 Database Design 56

4.2.3.1 Conceptual and Logical Database Design 57

4.3 Detailed Design 61

4.3.1 Sequence Diagram 61

4.3.2 UML Diagram 64

4.4 Conclusion 65

CHAPTER 5: IMPLEMENTATION 66

5.1 Introduction 66

5.2 Software Development Environtment Setup 66

5.3 Software Configuration 67

5.3.1 Managements. 67

5.3.1.1 Android Studio 68

5.3.1.2 Database Configuration. 68

5.3.2 Version Control Procedure 69

5.4 Implementation Status 70

5.5 Conclusion 71

CHAPTER 6: TESTING 72

6.1 Introduction 72

6.2 Test Plan 72

6.2.1 Test Organization 73

6.2.2 Test Environment Setup 74

6.2.2.1 Environment Setup 75

6.2.2.2 Application Software 75

6.2.2.3 System Software 76

6.2.2.4 System Hardware 76

6.2.3 Test Schedule 77

6.3 Test Strategy 78

6.3.1 Classes of Test 80

6.4 Test Design 81

6.4.1 Test Description 81

6.4.1.1 Test Description For Login Module 82

6.4.1.2 Test Description For Registration Module 83

Table 6.9 83

6.4.1.3 Test Description For Booking Appointment Module 84

6.4.1.4 Test Description For Appointment Booking Module 85

6.4.1.5 Test Description For Manage User Module 86

6.4.1.6 Test Description For Manage User Module 86

6.4.2 Test Data 87

6.4.2.1 Test Data For Login Module 88

6.5 Test Result and Analysis 99

6.6 Conclusion 102

CHAPTER 7: PROJECT CONCLUSION 103

7.1 Development On Strength And Weakness 103

7.1.1 Strength Of RazorReady 103

7.1.2 Weakness of RazorReady 103

7.2 Proposition For Improvement 104

7.3 Project Contribution 104

7.4 Conclusion 105

REFERENCES 106

LIST OF TABLES

Table	PAGE
Table 2.1 : Comparison Between Existing Project and RazorReady	11
Table 4.1 : User	59
Table 4.2 : Admin	59
Table 4.3 : Booking	60
Table 4.4 : Feedback	60
Table 4.5 : Appointment	60
Table 4.6 : Haircuts and Services	60
Table 5.1 : Implementation Status	70
Table 6.1 : Test Organization	73
Table 6.2 : Application Software	75
Table 6.3 : System Software	76
Table 6.4 : System Hardware	76
Table 6.5 : Test Schedule	77
Table 6.6 : Test Strategy	79
Table 6.7 : Classes of Test	80
Table 6.8 : Test Description For Login Module	82
Table 6.9 : Test Description For Register Module	83
Table 6.10 : Test Description For Booking Module	84
Table 6.11 : Test Description For View Booking Module	85
Table 6.12 : Test Description for Manage User Module	86
Table 6.13 : Test Description for Manage Appointment Module	86
Table 6.14: Test Result And Analysis For Login Module	99
Table 6.15 : Test Result And Analysis For Register Module	100
Table 6.16 : Test Result And Analysis For Booking Module	100
Table 6.17 : Test Result And Analysis For View Booking Module	101
Table 6.18 : Test Result And Analysis for Manage User Module	101
Table 6.19 : Test Result And Analysis for Manage Appointment Module	102

LIST OF FIGURES

Figure	PAGE
Figure 2.1 UI Design for Booksy	8
Figure 2.2 UI Design for Fresha	9
Figure 2.3 UI Design for Masters Pro	10
Figure 2.4 Agile Methodology	12
Figure 2.5 Gantt Chart	17
Figure 3.1 Main Flowchart	20
Figure 3.2 Booking Appointment Flowchart	21
Figure 3.3 List of User Booking Flowchart	22
Figure 3.4 List of User Flowchart	23
Figure 3.5 List Hairstyle Flowchart	24
Figure 3.6 ERD Diagram	25
Figure 3.7 Laptop Specification	45
Figure 3.8 Smartphone Software Specification	46
Figure 3.9 Android Studio	47
Figure 3.10 Firebase Console	47
Figure 4.1 System Architecture	50
Figure 4.2 Login Page	51
Figure 4.3 Register Page	52
Figure 4.4 Login Page Input	53
Figure 4.5 Booking Page	54
Figure 4.6 Booking List	55
Figure 4.7 ERD Diagram	57
Figure 4.8 Login User	61
Figure 4.9 Booking Appointment	62
Figure 4.10 History Booking	63
Figure 4.11 UML Diagram	64
Diagram 5.1 Deployment Diagram	67

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**

CHAPTER 1: INTRODUCTION

1.1 Introduction

In today's fast-paced world, convenience and efficiency are paramount. RazorReady is a cutting-edge mobile application designed to transform the way users book their haircuts, making the process seamless and hassle-free. This innovative app empowers users to schedule their barber appointments anytime, anywhere, directly from their smartphones.

RazorReady addresses a common pain point for both customers and barbers managing and coordinating appointment schedules. By leveraging advanced technology, RazorReady offers a user-friendly interface that simplifies the booking process, reduces wait times, and enhances the overall customer experience.

With RazorReady, users can browse available barbers, view their profiles, check real-time availability, and book appointments with just a few taps. The app also provides features such as appointment reminders, customizable booking options, and secure payment gateways, ensuring a smooth and efficient service from start to finish.

For barbers, RazorReady offers a powerful tool to manage their schedules, track appointments, and communicate with clients effortlessly. This not only helps in optimizing their workflow but also in building stronger relationships with their clientele.

In essence, RazorReady is more than just a booking app; it is a comprehensive solution designed to elevate the barbering experience for both customers and professionals. By integrating convenience, reliability, and innovation, RazorReady is set to become an indispensable tool in the modern grooming industry.

1.2 Problem Statement

In the traditional barbershop environment, customers often face significant challenges in booking appointments, including long wait times, lack of real-time availability information, and inefficient communication with barbers, which can lead to scheduling conflicts and dissatisfaction. Barbers, on the other hand, struggle with managing their schedules, tracking appointments, and maintaining effective communication with clients, which hampers their ability to optimize workflow and provide personalized services. Additionally, there is a lack of a centralized platform for customers to explore and be inspired by the latest haircut styles and grooming trends, further limiting their overall grooming experience. These issues highlight the need for a comprehensive solution that streamlines the appointment booking process, enhances communication, and provides a platform for style inspiration, thereby improving the overall efficiency and satisfaction for both customers and barbers.

1.3 Objectives

1. **Enhance Convenience**: RazorReady simplifies the appointment booking process for customers, allowing them to easily schedule their haircuts anytime, anywhere, with just a few taps.

2. **Facilitate Communication**: The app provides real-time notifications and updates regarding appointment status, ensuring both barbers and clients are informed of any changes promptly and efficiently.

3. **Inspire Creativity**: RazorReady aims to inspire creativity and exploration within the grooming community by offering a curated gallery of haircut styles and grooming trends, encouraging users to discover and try new looks.

1.4 Scope

1. Platform Development:

a) Design and develop a user-friendly mobile application that supports both iOS and Android platforms.

b) Implement robust backend infrastructure to handle user data, appointment scheduling.

c) Ensure the platform is secure, scalable, and capable of handling high traffic volumes.

2. User:

a) Develop a seamless user interface that allows customers to easily browse barbers, view profiles, check availability, and book appointments.

b) Provide a curated gallery of haircut styles and grooming trends to inspire creativity and exploration.

3. Barber:

a) Create a dedicated interface for barbers to manage their schedules, track appointments, and communicate with clients efficiently.

b) Provide tools for barbers to showcase their work, including uploading photos of haircut styles and receiving client reviews.

4. Admin Dashboard:

a) Develop an admin dashboard to oversee and manage the entire platform, including user and barber accounts, appointment data, and system performance.

b) Implement analytics and reporting features to monitor platform usage, identify trends.

c) Ensure the admin dashboard includes tools for handling customer support, resolving disputes, and maintaining the overall health of the platform.

1.5 Project Significant

1. Enhanced Customer Convenience:

   a) Streamlined Booking Process: By simplifying the appointment booking process, RazorReady saves customers time and effort, allowing them to schedule haircuts with ease and flexibility.

   b) Style Inspiration: The curated gallery of haircut styles and grooming trends provides customers with a wealth of inspiration, helping them discover and choose new looks with confidence.

2. Improved Barber Efficiency:

   a) Optimized Schedule Management: Barbers can efficiently manage their schedules, track appointments, and update availability in real-time, leading to better time management and reduced downtime.

   b) Showcasing Skills: The platform allows barbers to showcase their work, receive reviews, and build a reputable online presence, attracting more clients and enhancing their professional growth.

3. Business Growth and Performance:

   a) Increased Clientele: By providing a seamless and convenient booking experience, RazorReady can attract more customers to barbershops, leading to increased business and revenue.

   b) Customer Satisfaction: With improved service efficiency and personalized experiences, customer satisfaction is likely to increase, resulting in higher retention rates and positive word-of-mouth referrals.

4. Industry Innovation:

   a) Technological Advancement: RazorReady leverages advanced technology to modernize the traditional barbershop experience, setting a new standard for the industry.

   b) Competitive Edge: Barbershops that adopt RazorReady can gain a competitive advantage by offering a superior customer experience and staying ahead of market trends.

1.6 Expected Outcome

1. Improved User Experience: By offering a seamless and intuitive interface, RazorReady ensures that users can easily navigate the app, book appointments, and explore new styles with minimal effort.

2. Increased Efficiency: The app streamlines the appointment scheduling process, reducing wait times and optimizing the workflow for barbers. This allows for better time management and more efficient service delivery.

3. Enhanced Customer-Barber Relationships: Real-time notifications and updates facilitate better communication between barbers and clients, helping to build stronger, more personalized relationships.

4. Boosted Business Performance: By providing barbers with powerful tools to manage their schedules, track appointments, and communicate with clients, RazorReady helps barbershop businesses operate more smoothly and effectively, ultimately boosting their performance and customer satisfaction.

1.7 Conclusion

In conclusion, RazorReady is set to revolutionize the barbershop experience by addressing critical challenges such as inefficient appointment booking, poor communication, and lack of style inspiration through its innovative mobile application. By focusing on enhancing user experience, increasing efficiency, fostering better customer-barber relationships, and boosting business performance, RazorReady offers a comprehensive solution that benefits customers, barbers, and the grooming industry. The project's scope includes platform development, user and barber interfaces, and an admin dashboard, ensuring a holistic approach to solving the identified problems. With its user-friendly design, real-time notifications, and curated style gallery, RazorReady promises to deliver a seamless, personalized, and inspiring grooming journey, ultimately transforming the way barbershops operate and engage with their clients.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

The Literature Review and Project Methodology for RazorReady aim to provide a comprehensive understanding and systematic approach to developing an innovative mobile application for the grooming industry. The Literature Review explores existing research, technologies, and methodologies related to appointment booking systems, communication tools, and style discovery platforms, identifying gaps and opportunities for innovation. Insights from academic articles, industry reports, and case studies will inform the development of RazorReady. The Project Methodology outlines the phases of planning, design, development, testing, and deployment, adopting an agile framework for flexibility and iterative progress. Emphasizing user-centered design and robust backend development, the methodology ensures real-time data processing, secure transactions, and a seamless user experience. This structured approach aims to deliver a high-quality, efficient, and user-friendly application that meets the needs of customers, barbers, and the broader grooming industry.

2.2 Facts and Finidng

2.2.1 Domain

The barbershop industry is experiencing significant growth, driven by increasing consumer interest in grooming and personal care. Traditional barbershops often face inefficiencies due to reliance on walk-ins and phone bookings, leading to long wait times and scheduling conflicts. Malaysian consumers, particularly Millennials and Gen Z, show a strong preference for mobile apps that offer convenience and personalized experiences. Technological advancements in cloud computing and AI have enabled scalable, secure platforms that provide real-time notifications and personalized recommendations. While several mobile applications like Booksy and StyleSeat cater to salon and spa appointments globally, there is a gap in the market for barbershop-specific solutions in Malaysia. Compliance with data privacy regulations and payment security standards is essential. Efficient appointment management systems can increase barbershop revenue by optimizing booking slots and reducing no-shows, enhancing customer satisfaction and loyalty, and driving repeat business and positive word-of-mouth referrals.

2.2.2 Existing Application

a. Booksy

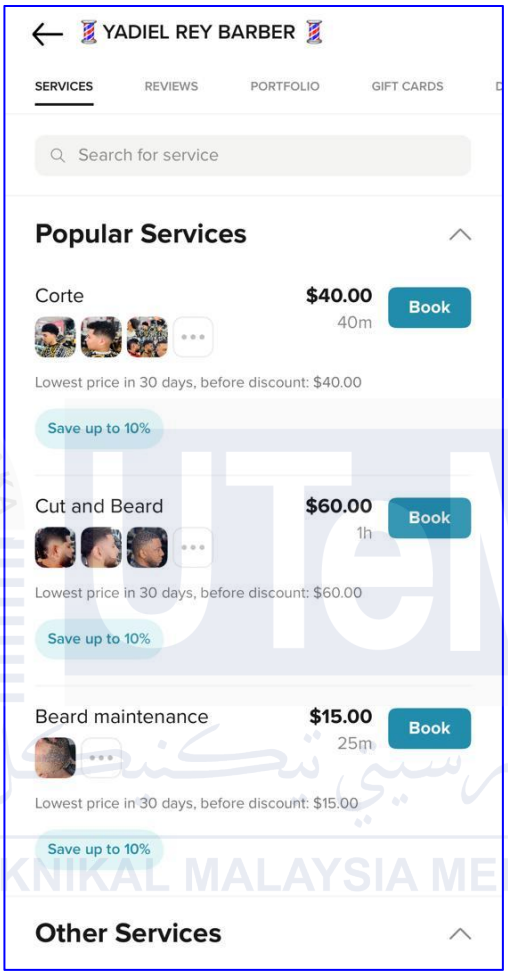

Figure 2.1 UI Design for Booksy

According to figure 2.1, it displays a booking interface for a barber named Yadiel Rey Barber, showcasing three popular services, a haircut for $40.00 taking 40 minutes, a cut and beard service for $60.00 taking 1 hour, and beard maintenance for $15.00 taking 25 minutes. Each service lists the lowest price in 30 days, mentions potential discounts up to 10%, and includes a booking button. Small thumbnail images illustrate examples of the services. The interface features tabs for "Services," "Reviews," "Portfolio," and "Gift Cards" at the top, with a search bar for specific services. The "Other Services" section is partially visible, indicating more available options. In support of this concept, RazorReady have been created to assist user with their everyday chores - patient management and appointments.

b. Fresha

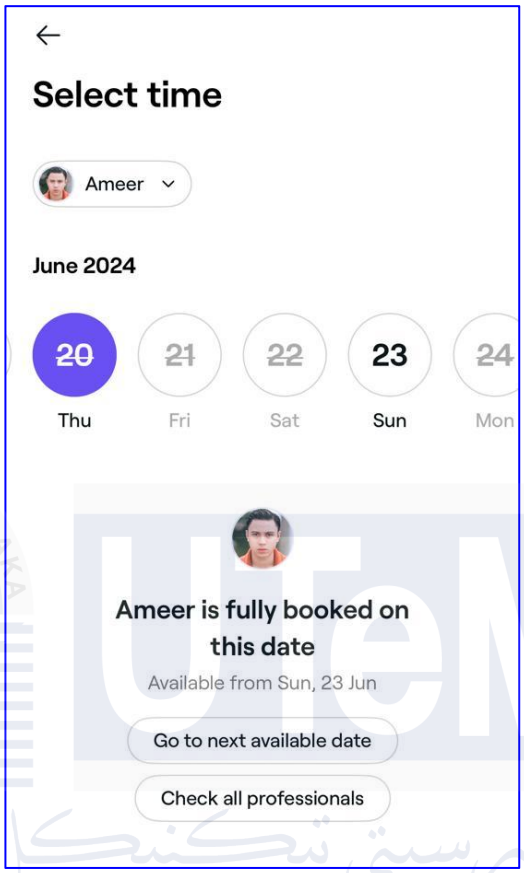

Figure 2.2 UI Design for Fresha

Figure 2.2 shows a booking interface for selecting an appointment time with a professional barber named Ameer. The date selection screen highlights June 20, 2024 (Thursday) in purple, indicating the current selection. Other dates from June 21 to June 24 are shown in a lighter shade, indicating availability to choose from. Underneath, there is a notice with a profile picture of Ameer, stating that "Ameer is fully booked on this date" and will be available from Sunday, June 23. There are two buttons below this message, "Go to next available date" and "Check all professionals," offering options to find the next available slot or to view other professionals for booking. Based on Fresha UI design, RazorReady provide the same method which is slot by slot for the booking section.

c. Masters Pro

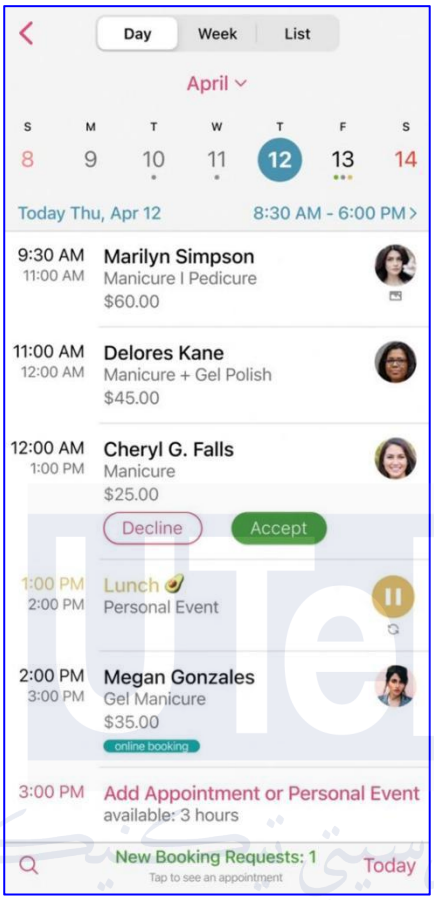

Figure 2.3 UI Design for Masters Pro

Figure 2.3 shows a scheduling interface for a professional on Thursday, April 12, listing appointments and personal events. The calendar highlights April 12, with working hours from 8:30 AM to 6:00 PM. Appointments include Marilyn Simpson at 9:30 AM for a manicure and pedicure ($60), Delores Kane at 11:00 AM for a manicure and gel polish ($45), Cheryl G. Falls at 12:00 PM for a manicure ($25) with options to accept or decline, and Megan Gonzales at 2:00 PM for a gel manicure ($35) booked online. A personal lunch event is scheduled from 1:00 PM to 2:00 PM. There is also an option to add more appointments or personal events, with a notification indicating one new booking request. This UI Design will be shown in admin page which is admin can see when the customer booked their haircuts and so on.

2.2.3 Comparison between RazorReady and existing application

Table 2.1 Comparison Between Existing Project and RazorReady

Feature	Booksy	Fresha	Masters Pro	RazorReady
Appointment Scheduling	Yes	Yes	Yes	Yes
Mobile App	Yes	Yes	Yes	Yes
Style Inspiration Gallery	Yes	Yes	No	Yes
Customer Reviews	Yes	Yes	Yes	Yes
Analytics and Reporting	Yes	No	No	Yes
Data Privacy Compliance	Yes	Yes	Yes	Yes
Communication Tools	Yes	Yes	Yes	Yes

2.3 Project Methodology

Developing a barbershop booking system mobile app requires a structured approach to ensure that the final product meets user needs and operates efficiently. This project methodology outlines a step-by-step process that covers all phases of the project, from initial planning to post-launch review. By following this methodology, the development team can systematically address each aspect of the project, ensuring a thorough and well-executed mobile application.

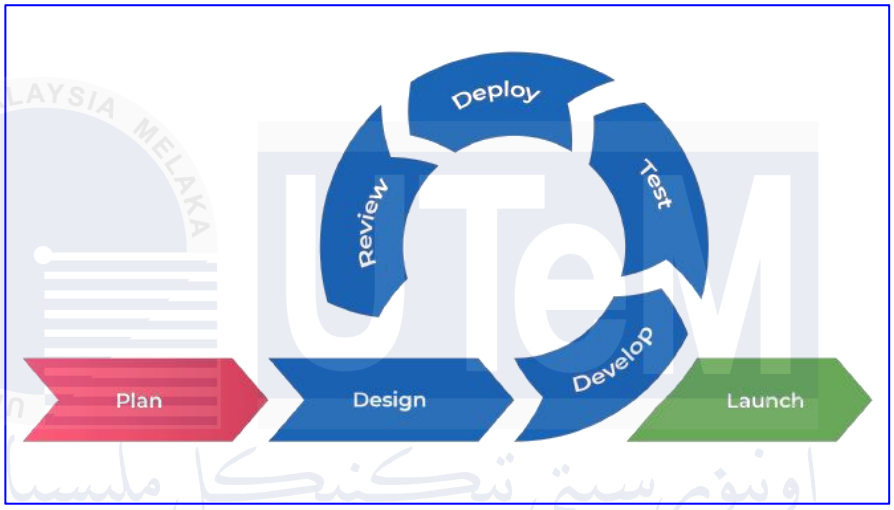

Figure 2.4 Agile Methodology

1. Plan

The planning phase is crucial as it sets the foundation for the entire project. In this stage, we will define clear goals and objectives for the barbershop booking system, such as enhancing customer convenience, streamlining appointment scheduling, and increasing customer engagement. Identifying stakeholders, including barbers, customers, and management, will help in understanding the diverse needs and expectations. Requirements gathering will involve detailed discussions with stakeholders to collect necessary specifications that the system must fulfill. Additionally, establishing a comprehensive project timeline with key milestones and deadlines will ensure that the project stays on track and progresses smoothly.

2. Design

The design phase focuses on creating a blueprint for the mobile app. This includes developing wireframes and mockups to visualize the user interface, ensuring an intuitive and user-friendly booking experience. Database design is another critical aspect, where we will plan the structure to efficiently store user information, booking details, and barber schedules. Defining the architecture of the app involves selecting appropriate frameworks, platforms, and technologies that align with the project's goals. Developing a prototype at this stage will help in demonstrating the basic functionality and design, allowing for early feedback and adjustments.

3. Develop

In the development phase, the actual coding and creation of the mobile app take place. Front-end development involves implementing the user interface based on the design specifications, ensuring that the app is responsive and functions well across various devices. Simultaneously, back-end development focuses on building the server-side logic, database interactions, and APIs required for the app's functionality. Integration of front-end and back-end components is crucial to ensure seamless data flow and overall functionality. This phase also includes implementing core features such as user registration, booking system, barber profiles, and notifications, bringing the app to life.

4. Test

Testing is a critical phase to ensure the quality and reliability of the app. It begins with unit testing, where individual components are tested to verify their correctness. Integration testing follows, examining the interactions between different components to identify any issues. User acceptance testing (UAT) involves real users testing the app and providing feedback, which is invaluable for identifying user experience issues and areas for improvement. Any bugs or issues discovered during testing will be addressed and fixed to ensure a smooth and error-free user experience.

5. Deploy

Deployment involves making the app available to users. This phase starts with deployment planning, which includes server setup, domain configuration, and app store submission. Setting up production environments and ensuring all configurations are in place is crucial for a smooth deployment. Once everything is ready, the app will be released to production environments and published on relevant app stores. Post-deployment, continuous monitoring will be conducted to ensure the app runs smoothly and to quickly address any issues that may arise.

6. Review

The review phase is about collecting feedback and analyzing performance post-launch. Gathering feedback from users and stakeholders will provide insights into the app's performance and usability. Performance metrics will be analyzed to identify any areas that need improvement. Based on this feedback and analysis, plans for future iterations will be made to continuously enhance the app's functionality and user experience.

7. Launch

The launch phase focuses on promoting the app and supporting its users. Marketing efforts will be undertaken to attract users to the app through various channels. Training sessions will be provided for barbers and staff to ensure they are comfortable using the system. Additionally, support channels will be set up to assist users with any issues or questions they may have, ensuring a smooth transition and positive user experience.

By following this comprehensive project methodology, the development of this project can systematically address each aspect of the project, ensuring the creation of a user-friendly, and efficient barbershop booking system mobile app.

2.4 Project Requirements

2.4.1 Software Requirements

1. Android Studio

Android Studio is the primary integrated development environment (IDE) for developing this mobile app. It offers comprehensive tools for building, testing, and debugging Android applications. Within Android Studio, the latest Android Software Development Kit (SDK) should be installed to access necessary libraries and tools required for app development. The built-in emulator in Android Studio will be utilized to test the app on different virtual devices, ensuring compatibility across various Android versions. Gradle, the build automation tool used within Android Studio, will manage dependencies and streamline the build process, ensuring smooth integration of all project components.

2. The Firebase Console

Firebase Console will play a crucial role in the backend services of the app. Firebase Authentication will handle secure user sign-up, sign-in, and authentication processes. To store and sync booking details, user profiles, and other relevant data in real-time, either Firebase Realtime Database or Firestore will be employed. Firebase Cloud Functions will run backend code in response to events triggered by Firebase features, adding flexibility and scalability to the app. Firebase Cloud Messaging (FCM) will be integrated to send notifications to users about upcoming appointments, promotions, and updates, enhancing user engagement. Additionally, Firebase Analytics will be implemented to track user interactions and gather insights, aiding in the continuous improvement of the app.

3. Canva

For design resources, Canva will be utilized to create visually appealing designs for the app's user interface. This includes designing icons, buttons, banners, and promotional graphics that ensure a seamless and attractive user experience. Canva will also be instrumental in developing a consistent brand identity for the app, including designing the logo, color scheme, and typography that align with the barbershop's branding. Furthermore, Canva will be used to create marketing materials such as social media posts, flyers, and advertisements to effectively promote the app to the target audienc

2.4.2 Hardware Requirements

1. Laptop

The development environment should be set up on a compatible operating system, such as Windows that supports Android Studio. Hardware requirements include a multi-core processor minimum Intel i5 or equivalent, at least 8 GB of RAM or 16 GB recommended for smooth performance, and sufficient storage space such as SSD recommended for storing project files, dependencies, and emulator images. Ensuring that the development environment meets these specifications will contribute to a more efficient development process.

2.5 Gantt Chart

Figure 2.5 below shows the Gantt Chart to deliver RazorReady.

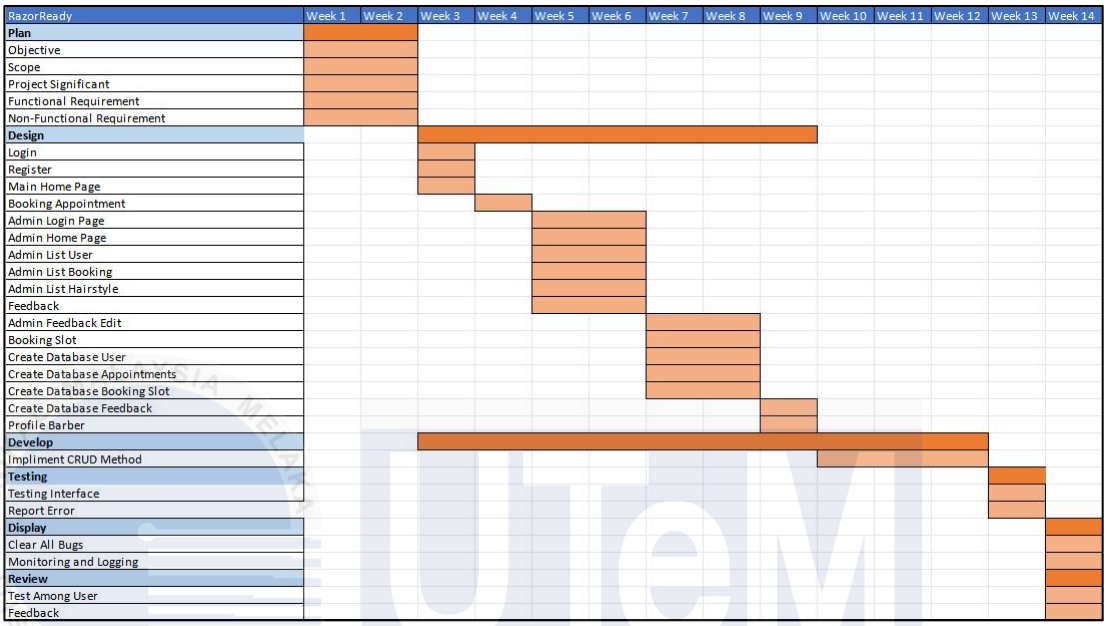

Figure 2.5 Gantt Chart

2.6 Conclusion

Developing a barbershop booking system mobile app requires a structured approach, involving clear phases of planning, designing, developing, testing, deploying, and reviewing. Utilizing Android Studio for development, Firebase Console for backend services, and Canva for design ensures a robust, user-friendly application. The detailed project timeline, represented in a Gantt chart, schedules specific tasks like proposal writing, project progress tracking, report writing, and evaluations, ensuring effective management of dependencies. By following this comprehensive methodology and using the specified tools, the development team can systematically create a high-quality app that enhances customer convenience, optimizes appointment scheduling, and improves overall engagement.

CHAPTER 3: ANALYSIS

3.1 Introduction

This chapter is providing the analysis of the application which consists of problem analysis, and requirement analysis. And for the requirement, there will be consisting of data requirement, functional requirement, non-functional requirement, and others requirement

3.2 Problem Analysis

This mobile application will be ease of use to everyone. Here we can see the flow of the application with illustrated diagrams.

3.2.1 Flowchart

As Figure 3.1 shows below, the flowchart depicts the process flow of a user management and booking system. It starts with the user choosing to register or log in. If a new user registers, they proceed to login as either a regular user or an admin. After logging in, verification is required. For regular users, they access the user homepage, where they can view profiles, contacts, barbers, haircut galleries, and other services. They can also book appointments and view their booking history, eventually logging out to end the session. For admins, after logging in and verification, they access the admin dashboard, where they can manage lift bookings, list users, barbers, hairstyles, and feedback, and reset slot bookings or delete feedback. The flow is designed to handle both user and admin functionalities efficiently.

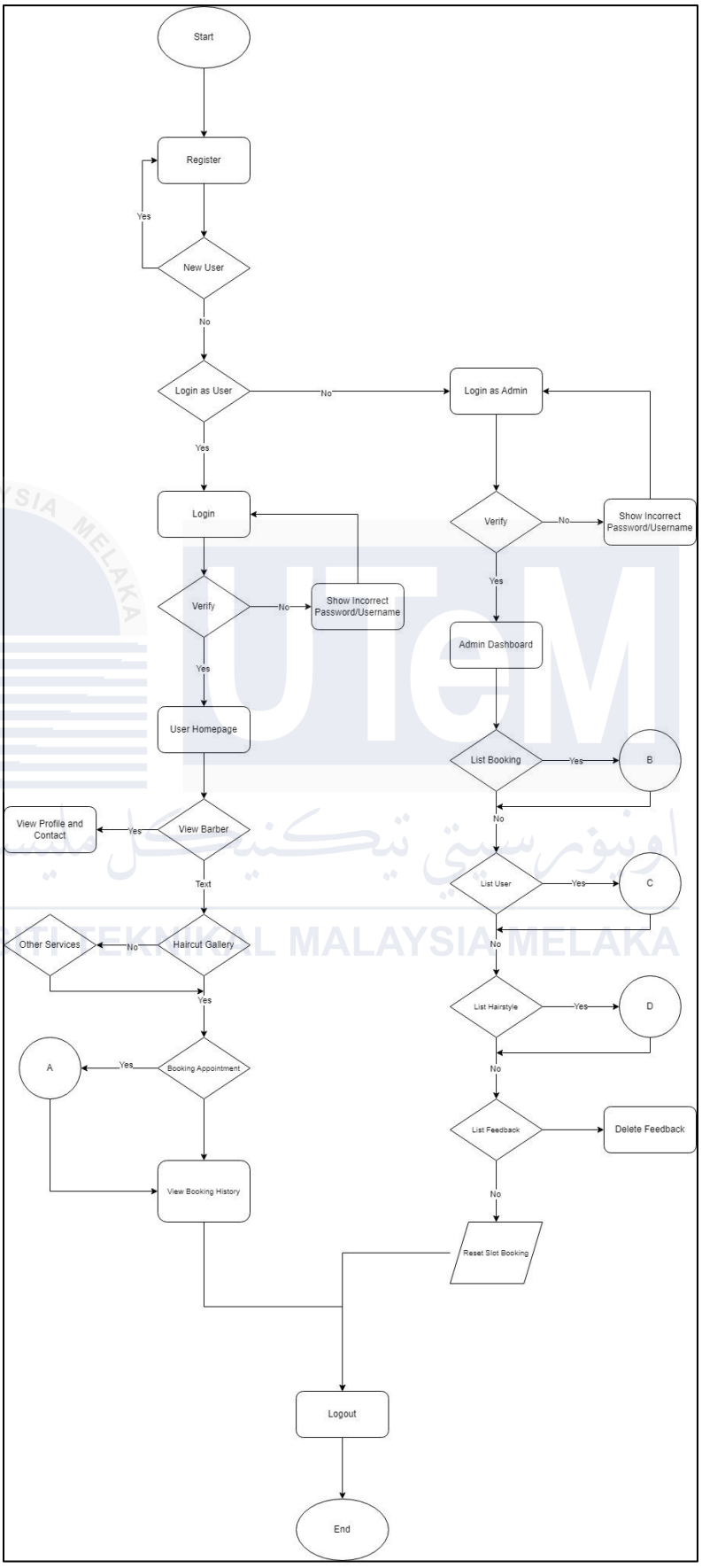

Figure 3.1 Main Flowchart

As Figure 3.2 shows below, the flowchart focuses on the appointment booking process within the system. Starting from point A, the user initiates by selecting the "Book an appointment" option. The user is then required to insert their booking details. The system checks for available slots. If no slots are available, the process loops back to the booking details step, allowing the user to modify or retry their request. If a slot is available, the system proceeds to confirm the booking as successful. Upon successful booking, the process ends. This flow ensures that the user can only complete a booking if a suitable slot is available, providing a seamless appointment scheduling experience.

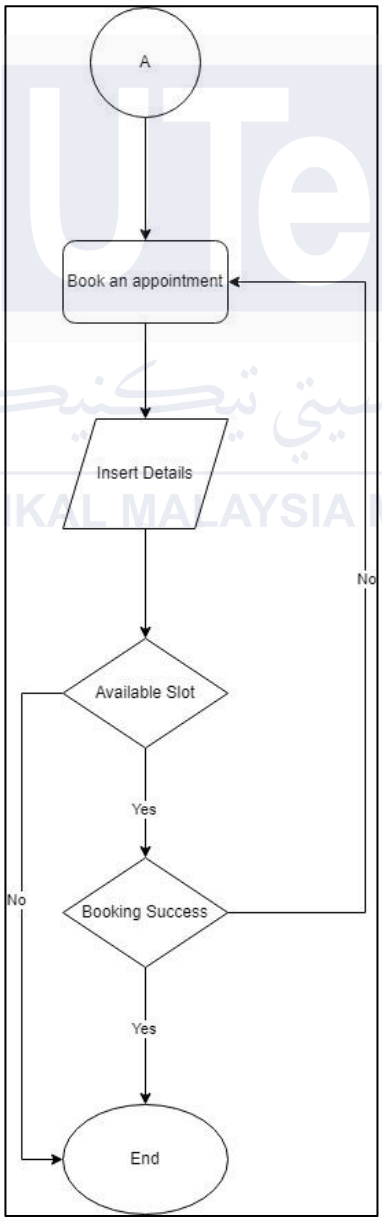

Figure 3.2 Booking Appointment Flowchart

As Figure 3.3 shows below, the flowchart details the process of managing existing appointments. Starting from point B, admin is first shown the details of users appointments. They then have the option to edit the booking details. If the admin chooses to delete the booking, they proceed to a confirmation step. If the confirmation is successful, the booking is deleted, and the process ends. If the admin does not confirm the deletion, they are returned to the appointment details step. This flow ensures that admin can view, edit, or delete their appointments while requiring confirmation for deletions to prevent accidental loss of bookings.

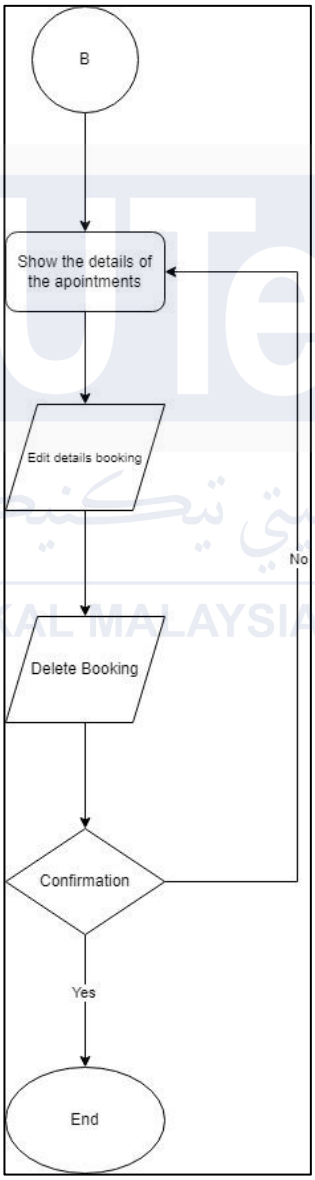

Figure 3.3 List of User Booking Flowchart

As Figure 3.4 shows below, the List User flowchart outlines the process for an admin managing user accounts within the application. Starting from point C, the admin is first shown the list of users in the application. The admin can then choose to edit user details. If the admin decides to delete a user, they proceed to a confirmation step. If the deletion is confirmed, the user is deleted, and the process ends. If the admin does not confirm the deletion, they are returned to the user list. This flow ensures that the admin can view, edit, or delete user accounts, with a confirmation step included to prevent accidental deletions.

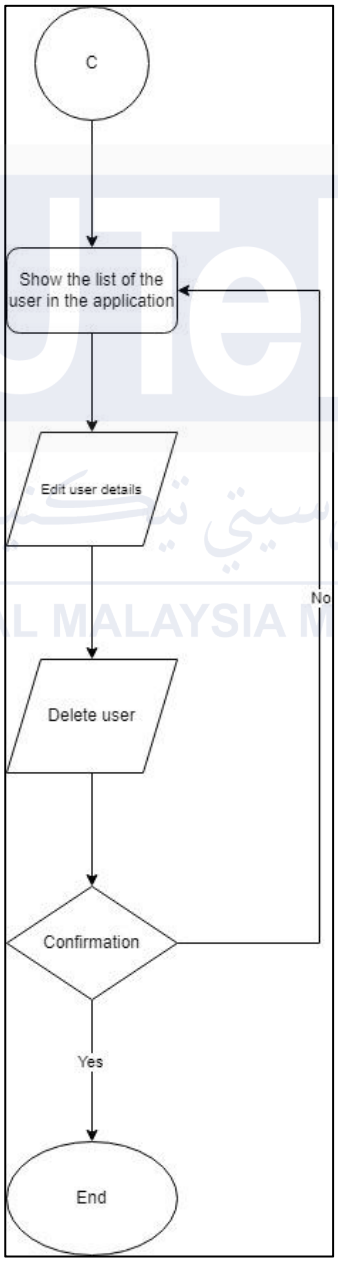

Figure 3.4 List of User Flowchart

As Figure 3.5 shows below, the List Hairstyle flowchart illustrates the process for an admin managing haircuts and other services within the application. Starting from point D, the admin is first shown the list of available haircuts and other services. The admin can choose to edit the details of these haircuts. Additionally, the admin has the option to add or delete hairstyles. Before finalizing any changes, the system requires confirmation. If the changes are confirmed, the process ends successfully. If the admin does not confirm the changes, they are returned to the list of haircuts and services. This flow ensures that the admin can manage service offerings efficiently, with a confirmation step to verify the modifications.

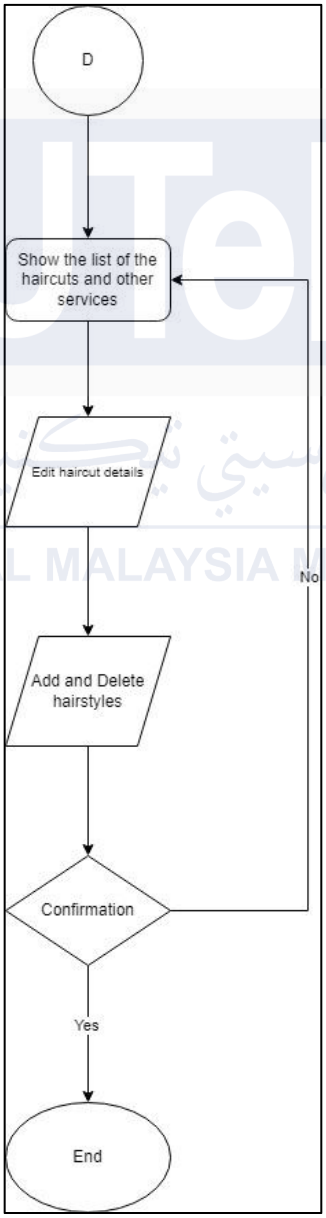

Figure 3.5 List Hairstyle Flowchart

3.2.2 ERD Diagram

The figure 3.6 below show an ERD diagram which is it illustrates a system for managing users, admins, appointments, haircuts, feedback, and bookings. The User entity has attributes like username, email, phone number, password, and role. The Admin entity includes similar attributes for admin-specific roles. Appointments link to users with details like phone number, barber, services, and time. Haircut and Services contain haircut ID, name, and price. Feedback includes feedback ID, content, email, and name. Bookings record slot and status information. Users can have multiple appointments, which are managed by admins, and admins can also handle bookings and feedback. The relationships ensure proper management of all entities within the system.

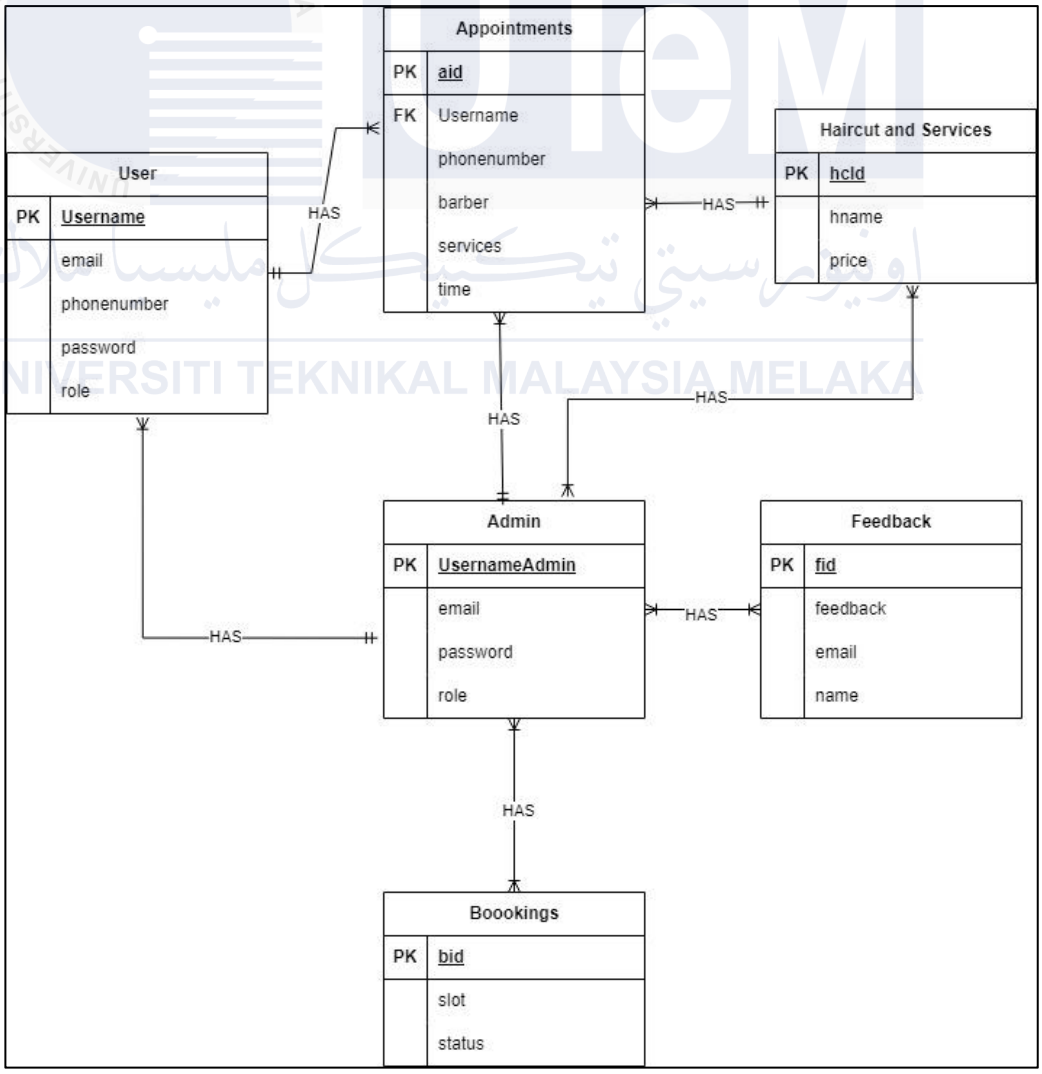

Figure 3.6 ERD Diagram

3.3 Requirement Analysis

3.3.1 Data Requirement

A barbershop booking system is an essential tool for modern barbershops to streamline their operations and enhance customer satisfaction. By digitizing the appointment booking process, barbershops can manage customer appointments more efficiently, reduce wait times, and provide better service. This system requires a detailed understanding of both data inputs and outputs to function effectively. In this analysis, we will outline the necessary data requirements for input and output to ensure the system meets the needs of customers, barbers, and administrators.

a. Input

User Registration Details The foundation of any booking system is user authentication, which requires detailed registration inputs. Users need to provide their username, email, phone number, password, and role (customer or admin). This information is crucial for creating a secure and personalized user experience. For instance, a unique username ensures that each user can have a distinct identity within the system, while the email and phone number are essential for communication and verification purposes. Passwords protect user accounts from unauthorized access, and the role helps in defining user permissions and access levels within the system.

To book an appointment, several pieces of information are required from the user. These include the customer’s username, phone number, selected barber, desired services, and preferred time slot. The username ensures that the booking is associated with the correct user account. The phone number allows for immediate communication if any changes or confirmations are needed. Selecting a barber and the desired services helps the system to allocate the appropriate professional and resources for the appointment. The preferred time slot ensures that the booking is made at a convenient time for both the customer and the barber, thereby optimizing the schedule and reducing wait times.

For the system to present accurate and comprehensive options to customers, it needs detailed input about the barbers and the services they offer. Each barber's

profile should include their name, availability, and expertise. This data helps customers choose the right professional for their needs. Additionally, the services offered, along with their prices, must be clearly defined. This includes various haircut styles, grooming services, and any additional offerings. By having this information readily available, the system can provide customers with all the necessary details to make informed decisions when booking their appointments.

b. Output

The system must provide several key outputs to ensure a seamless experience for customers. One of the primary outputs is the display of available time slots, which helps customers find a convenient booking time. After booking, the system should generate a confirmation that includes the appointment details such as the date, time, barber, and services selected. This confirmation can be sent via email or SMS to ensure the customer is well-informed. Additionally, the system should maintain and display a history of past appointments, allowing customers to keep track of their visits and services received.

Barbers need specific outputs from the system to manage their schedules and prepare for appointments. The system should generate a daily or weekly schedule that includes customer names, appointment times, and requested services. This information helps barbers organize their day and ensure they are prepared for each client. Additionally, the system should provide a detailed view of each appointment, including any special requests or notes made by the customer. By having this information readily accessible, barbers can deliver personalized and efficient services. Furthermore, notifications about upcoming appointments or any changes to the schedule should be sent to barbers to keep them informed and ready.

Administrators require comprehensive outputs to manage the barbershop effectively. The system should generate detailed reports on user activity, including new registrations, appointment bookings, and feedback submissions. This data helps admins monitor the system's usage and identify any areas that need improvement. Additionally, the system should provide a list of all appointments, which can be filtered by date, barber, or customer. This output is crucial for tracking the barbershop's performance and ensuring that all appointments are managed efficiently.

Feedback from customers should also be compiled and presented to admins, allowing them to address any issues and improve service quality. Notifications and alerts about system performance, user activities, and any potential issues should be part of the admin outputs to ensure smooth operations.

3.3.2 Funtional Requirement

Functional requirements are crucial for defining the specific behaviors, functions, and operations of a system. In the context of a barbershop booking system, these requirements outline the essential features and capabilities that the system must possess to meet the needs of its users, including customers, barbers, and administrators. By clearly specifying what the system should do, functional requirements ensure that the system's development aligns with the intended use cases and business objectives. This section will detail the core functional requirements necessary to build a comprehensive and efficient barbershop booking system, encompassing user management, appointment scheduling, service management, and administrative controls.

3.3.2.1 User Registration and Login

The user registration process is fundamental to the barbershop booking system, allowing new users to create accounts and access the system's features. When a user decides to register, the system must present a registration form that collects essential information such as username, email address, phone number, password, and role (either customer or admin). The username should be unique to ensure each user has a distinct identifier within the system. The email and phone number fields must be validated to ensure they follow proper formats and are unique to each user. Additionally, the password must meet security standards, such as a minimum length and a combination of letters, numbers, and special characters, to ensure account security. Once the user submits the registration form, the system should verify the data, store the user information securely in the database, and send a confirmation email or SMS to validate the account.

The login functionality is critical for providing access control to the barbershop booking system. Upon navigating to the login page, users must enter their registered username and password. The system should validate these credentials against the stored user data. If the credentials are correct, the system grants access to the user, directing them to their respective dashboard—either the customer homepage or the admin dashboard, based on

their role. If the credentials are incorrect, the system should display an error message and prompt the user to try again. Additionally, the system should implement a 'Forgot Password' feature, allowing users to reset their passwords by verifying their identity through their registered email or phone number. This functionality ensures that only authenticated users can access the system, protecting user data and maintaining system security.

Role-based access control (RBAC) is an essential aspect of the user registration and login process. When a user registers, they must specify their role, which determines the level of access they have within the system. Customers should have access to features such as viewing available services, booking appointments, and providing feedback. In contrast, admins should have broader access to manage users, appointments, services, and feedback. Upon successful login, the system should dynamically display the appropriate dashboard and functionalities based on the user's role. This ensures that users only have access to the features relevant to their needs and responsibilities, enhancing both security and user experience. By implementing robust RBAC, the system can effectively segregate functionalities and protect sensitive administrative controls from unauthorized access.

3.3.2.2 Appointment Booking

The appointment booking functionality is central to the barbershop booking system, enabling users to schedule their services conveniently. Upon accessing the appointment booking section, users should be presented with an intuitive interface where they can select a preferred date and time from a calendar view. The system must check for real-time availability of barbers and slots, ensuring no double bookings occur. Users should also be able to select their desired services from a predefined list, including various haircut styles, grooming services, and additional options. The system should then display the available barbers for the chosen services and times, allowing users to make an informed decision. Once all selections are made, the user can confirm the booking, and the system will store the appointment details in the database.

Users should have the ability to view, edit, and cancel their appointments through a dedicated section in their dashboard. The system must provide a comprehensive overview of all upcoming and past appointments, displaying details such as date, time, barber, and services booked. If a user needs to reschedule, they should be able to modify the date, time, or services, with the system checking for new availability and updating the booking accordingly. Cancellations should also be straightforward, with users able to cancel appointments and receive a cancellation confirmation. The system should handle all updates and cancellations in real-time, ensuring that the availability of barbers and slots is accurately reflected for other users. This functionality ensures flexibility and convenience, allowing users to manage their bookings according to their needs.

For the appointment booking system to function effectively, it must include robust management of barbers and services. Administrators should have access to a management interface where they can add, update, or remove barbers and services. Each barber's profile should include their name, available hours, and specializations, while services should be detailed with descriptions and pricing. The system should ensure that any changes made by admins are immediately reflected in the booking interface, maintaining accurate and up-to-date information for users. Additionally, the system should support setting limits on the number of appointments a barber can take in a day to prevent overbooking. By providing comprehensive management capabilities, the system can ensure that the appointment booking process remains smooth and efficient for both users and barbers.

3.3.2.3 Appointment Management

The appointment management functionality is crucial for both users and administrators to keep track of scheduled services. Users should be able to access a dedicated section in their dashboard where they can view all upcoming and past appointments. This section should display comprehensive details for each appointment, including the date, time, barber, and services booked. For administrators, the system should provide a similar interface but

with additional capabilities to view all appointments across the barbershop. Admins should be able to filter appointments by date, barber, or service type, providing a clear overview of the barbershop's schedule and helping in resource planning and allocation.

The system should offer users the flexibility to edit their appointments if needed. From their dashboard, users should be able to select an appointment and modify details such as the date, time, barber, or services requested. The system must check the availability of the new selections to prevent conflicts or overbookings. Once the changes are confirmed, the system should update the appointment details in the database and send a notification to both the user and the barber, informing them of the changes. For administrators, the ability to edit any user's appointment is essential for managing scheduling conflicts or accommodating special requests. Admins should have the capability to override certain restrictions if necessary, ensuring smooth operation and customer satisfaction.

Appointment cancellation is a critical feature that provides users with the flexibility to manage their schedules effectively. Users should be able to cancel appointments easily from their dashboard by selecting the specific appointment and confirming the cancellation. The system must update the database in real-time to reflect the cancellation and free up the slot for other users. A confirmation of the cancellation should be sent to the user and the barber via email or SMS. For administrators, the ability to cancel appointments is vital for handling situations where barbers become unavailable or other operational issues arise. Admins should be able to cancel appointments on behalf of users, providing a reason for the cancellation, which should be communicated to the affected parties. This functionality ensures that both users and administrators can effectively manage and adapt to changing schedules.

3.3.2.4 Barber Profile

The barbershop booking system must allow administrators to create and manage detailed profiles for each barber. When creating a barber profile, the system should prompt the admin to enter essential information, including the barber's name, contact information, availability, specialties, and any relevant experience or certifications. This information helps customers choose the right barber based on their needs and preferences. The system should also allow admins to update these profiles as necessary, ensuring that all information remains current. For instance, if a barber's working hours change or they acquire new skills, the profile can be easily updated to reflect these changes. This dynamic profile management ensures that the barbershop can maintain accurate and useful information for its clients.

For the appointment booking system to be effective, it must display barber profiles to users in a clear and informative manner. When users are selecting a barber for their appointment, they should be able to view detailed profiles that include the barber's name, photo, specialties, available services, and working hours. Additional information such as customer reviews and ratings can also be included to help users make informed decisions. This display should be accessible from the booking interface and any other relevant sections of the user dashboard. By providing comprehensive and accessible barber profiles, the system enhances the user experience, helping customers to find the best match for their needs.

Administrators must have the capability to manage and update the availability of each barber. This includes setting their working hours, days off, and any periods of leave or unavailability. The system should provide an intuitive interface for admins to input and adjust these schedules. Additionally, barbers should have the ability to request changes to their availability through the system, which admins can then approve or deny. The system must automatically update the booking calendar to reflect these changes, ensuring that customers only book appointments during times when the barber is available. This functionality is crucial for maintaining an

organized and accurate booking system, preventing scheduling conflicts and enhancing operational efficiency.

3.3.3 Non-Functional Requirement

Non-functional requirements are essential attributes that define the overall quality and user experience of a system. Unlike functional requirements, which specify what the system should do, non-functional requirements describe how the system should perform. These requirements encompass various aspects such as usability, reliability, performance, scalability, and security. In the context of a barbershop booking system, non-functional requirements ensure that the system is not only functional but also user-friendly, efficient, and robust. This section will focus on the non-functional requirements related to usability, detailing how the system should be designed to provide an intuitive and satisfactory user experience.

3.3.3.1 Usability

The barbershop booking system must have a user-friendly interface that is intuitive and easy to navigate. This means that both first-time users and returning customers should be able to find and use the features they need without extensive guidance. The design should be clean and simple, with clear labels and instructions for all interactive elements. Buttons, forms, and links should be easily identifiable and accessible, ensuring that users can perform actions such as booking an appointment, editing their profile, or providing feedback with minimal effort. By prioritizing usability in the interface design, the system can reduce user frustration and increase overall satisfaction.

Ensuring accessibility is a critical aspect of the system's usability. The barbershop booking system should be designed to be accessible to all users, including those with disabilities. This involves following web accessibility standards such as the Web Content Accessibility Guidelines (WCAG). Features like screen reader compatibility, keyboard navigation, and alternative text for images should be incorporated to assist users with visual, auditory, or motor impairments. Additionally, the system should support

multiple languages to cater to a diverse user base. By making the system accessible, the barbershop can ensure that all users have an equal opportunity to utilize the booking services effectively.

The system must feature a responsive design that provides a seamless user experience across a variety of devices and screen sizes. Whether accessed from a desktop computer, tablet, or smartphone, the system should adjust its layout and functionality to suit the device being used. This includes scaling text and images appropriately, reorganizing content to fit different screen dimensions, and ensuring that touch interactions work smoothly on mobile devices. A responsive design is essential for usability, as it allows users to access the system conveniently from any device, making it more likely that they will use the service regularly.

Incorporating user feedback mechanisms and help resources is vital for enhancing usability. The system should include options for users to provide feedback about their experience, report issues, and suggest improvements. This feedback can be collected through forms, surveys, or direct contact options. Additionally, a comprehensive help section or FAQ should be available, offering guidance on common tasks and troubleshooting steps. Providing these resources ensures that users can resolve their issues independently and feel supported while using the system. Regularly reviewing and acting on user feedback can also help the barbershop continually improve the system's usability over time.

3.3.3.2 Reliability

The barbershop booking system must ensure high availability to provide uninterrupted access for users at all times. This means the system should be operational 24/7, with minimal downtime for maintenance or updates. To achieve this, the system should be hosted on robust and reliable servers with failover mechanisms in place. These mechanisms include redundant server setups, automatic backup systems, and real-time monitoring to detect and address any issues promptly. High availability is critical as it

ensures users can book appointments, manage their profiles, and access services whenever they need, enhancing their trust and reliance on the system.

Maintaining data integrity is crucial for the reliability of the barbershop booking system. The system must ensure that all data entered, processed, and stored is accurate and consistent. This includes user information, appointment details, and service records. Data validation checks should be implemented to prevent incorrect or incomplete data entries. Additionally, transactional operations such as booking an appointment or updating user profiles should follow the ACID (Atomicity, Consistency, Isolation, Durability) principles to ensure that all changes are processed reliably and that the database remains in a consistent state. By upholding data integrity, the system can provide trustworthy and precise information to its users.

Effective error handling and recovery mechanisms are essential components of a reliable system. The barbershop booking system should be designed to detect, log, and respond to errors swiftly. When an error occurs, the system should provide clear and informative messages to the user, explaining the issue and suggesting possible actions to resolve it. For example, if a booking attempt fails due to a server error, the user should be informed and guided on how to retry or seek assistance. Additionally, the system should have automated recovery processes in place to restore normal operations quickly. This includes database recovery procedures, application restart capabilities, and data rollback mechanisms to prevent data loss and minimize disruption. By incorporating robust error handling and recovery features, the system can maintain continuous and reliable service for its users.

To ensure ongoing reliability, the barbershop booking system must implement continuous performance monitoring. This involves tracking key performance indicators (KPIs) such as response times, transaction success rates, and system load. Performance monitoring tools can detect potential bottlenecks and resource issues before they impact users. Regular performance assessments and stress testing should also be conducted to

identify and mitigate any vulnerabilities. By proactively monitoring and optimizing performance, the system can maintain a high level of reliability, ensuring that users experience smooth and efficient service even during peak usage times.

The system should be designed to handle varying loads and scale efficiently as demand grows. This means incorporating load balancing to distribute traffic evenly across servers and ensuring that the infrastructure can expand to accommodate an increasing number of users and transactions. Scalability planning includes both horizontal scaling (adding more servers) and vertical scaling (upgrading server capabilities). By preparing for scalability, the system can maintain reliability and performance as the barbershop grows, ensuring that all users receive consistent and dependable service regardless of the system load.

3.3.3.3 Security

The barbershop booking system must implement robust user authentication and authorization mechanisms to protect user accounts and sensitive data. Authentication should require users to provide a unique username and password combination, with options for multi-factor authentication (MFA) to add an extra layer of security. Passwords should be stored securely using hashing algorithms, and the system should enforce strong password policies to prevent weak or easily guessable passwords. Authorization controls should ensure that users can only access features and data appropriate to their roles (e.g., customers, barbers, admins). Role-based access control (RBAC) should be used to define and enforce these permissions, preventing unauthorized access to sensitive functionalities and data.

To safeguard the confidentiality and integrity of data, the system must employ encryption for data both in transit and at rest. Secure Sockets Layer (SSL) or Transport Layer Security (TLS) protocols should be used to encrypt data transmitted between the user's device and the server, preventing interception by unauthorized parties. Data stored in the system's databases,

including user information, appointment details, and payment information, should be encrypted using strong encryption standards like AES-256. This ensures that even if the data is accessed by unauthorized individuals, it remains unreadable and secure. Regular reviews and updates of encryption protocols should be conducted to maintain their effectiveness against emerging threats.

The security of the barbershop booking system must be maintained through regular security audits and vulnerability assessments. Security audits involve comprehensive reviews of the system's security policies, controls, and procedures to ensure compliance with industry standards and best practices. Vulnerability assessments should be conducted to identify and address potential security weaknesses in the system. This includes testing for common vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Automated tools and manual testing should be used to thoroughly assess the system's security posture. Any identified vulnerabilities must be promptly addressed through patches, updates, or configuration changes to mitigate risks.

A well-defined incident response plan is crucial for effectively managing and mitigating security incidents. The barbershop booking system should have procedures in place to detect, respond to, and recover from security breaches or attacks. This includes setting up monitoring systems to detect suspicious activities, such as multiple failed login attempts or unusual data access patterns. When a security incident is detected, the response plan should outline steps for containing the breach, eradicating the threat, and restoring affected systems to normal operation. Communication protocols should be established to inform affected users and relevant stakeholders about the incident and the measures being taken to address it. Post-incident analysis should be conducted to understand the root cause and prevent future occurrences.

The system must comply with relevant data protection regulations to ensure the privacy and security of user data. This includes adhering to laws

such as the General Data Protection Regulation (GDPR) for users in the European Union, and other local or international data protection laws that apply. Compliance involves implementing measures to protect personal data, providing users with rights over their data (such as access, correction, and deletion), and ensuring transparent data handling practices. Regular compliance reviews and updates should be conducted to align the system with evolving legal requirements, ensuring that the barbershop booking system remains legally compliant and trustworthy.

3.3.3.4 Performance

The barbershop booking system must ensure that all operations are performed with minimal delay, providing a smooth and efficient user experience. The system should aim for a response time of less than two seconds for most user interactions, such as loading pages, submitting forms, and fetching data. Quick response times are essential for maintaining user satisfaction, as slow performance can lead to frustration and abandonment. To achieve this, the system should utilize optimized database queries, efficient coding practices, and caching mechanisms to reduce load times and improve overall performance.

Scalability is a critical aspect of performance, ensuring that the system can handle an increasing number of users and transactions without degradation in performance. The barbershop booking system should be designed to scale both horizontally and vertically. Horizontal scaling involves adding more servers to distribute the load, while vertical scaling involves enhancing the capacity of existing servers. The system should incorporate load balancing to distribute traffic evenly across servers and avoid bottlenecks. Additionally, it should support cloud-based infrastructure, allowing for dynamic scaling based on demand. By planning for scalability, the system can maintain high performance even during peak usage periods.

Efficient resource utilization is vital for maintaining optimal performance. The system should be designed to make the best use of available CPU, memory, and network resources. This includes optimizing

code to reduce computational overhead, using efficient data structures and algorithms, and managing memory usage effectively to prevent leaks and overconsumption. The system should also implement asynchronous processing for tasks that do not require immediate user feedback, such as sending notifications or generating reports, to free up resources for real-time interactions. Regular monitoring and profiling of the system's resource usage can help identify and address inefficiencies, ensuring that resources are used effectively to support high performance

The system's throughput, or the number of transactions it can handle per second, is a key performance metric. The barbershop booking system should be able to process a high volume of bookings, profile updates, and other interactions concurrently without significant slowdowns. This requires efficient handling of concurrent requests and effective use of threading or asynchronous programming to manage multiple operations simultaneously. Throughput can be enhanced by optimizing database access, using connection pooling, and implementing parallel processing where appropriate. By maximizing throughput, the system can ensure that it remains responsive and reliable even under heavy load.

Continuous performance monitoring is essential to maintain and improve the system's performance over time. The barbershop booking system should implement real-time monitoring tools to track key performance indicators such as response times, error rates, and resource utilization. These tools can provide insights into the system's performance and help identify potential issues before they impact users. Regular performance testing, including load testing and stress testing, should be conducted to evaluate the system's behavior under various conditions and identify areas for improvement. Based on the results of these tests, the system can be optimized through code refactoring, database indexing, and other performance-enhancing techniques. By actively monitoring and optimizing performance, the system can deliver a consistently high-quality user experience.

3.3.3.5 Scalability

Scalability is a fundamental requirement for the barbershop booking system to ensure it can handle increased demand without performance degradation. The system should support both horizontal and vertical scaling. Horizontal scaling involves adding more servers to the system to distribute the load evenly. This can be achieved through load balancers that manage incoming traffic and distribute it across multiple servers, preventing any single server from becoming a bottleneck. Vertical scaling involves enhancing the capabilities of existing servers, such as adding more CPU, memory, or storage resources. This approach is essential for scenarios where increasing the power of individual servers can help meet demand. By supporting both scaling strategies, the system can adapt to varying loads and ensure consistent performance as user numbers and transaction volumes grow.

Load balancing is a critical component for ensuring the scalable operation of the barbershop booking system. Load balancers distribute incoming network traffic across multiple servers to ensure no single server is overwhelmed. This distribution helps in maintaining the performance and reliability of the system, especially during periods of high demand. The system should employ both hardware and software load balancers to manage traffic effectively. Techniques such as round-robin, least connections, and IP hash can be used to determine how traffic is distributed. Additionally, load balancers should have health check mechanisms to monitor the status of servers and redirect traffic away from any that are malfunctioning. By implementing robust load balancing, the system can handle large volumes of traffic and ensure a smooth user experience.

The database is often a critical bottleneck in scalable systems. The barbershop booking system should use a scalable database solution that can handle growing amounts of data and transactions efficiently. This includes using distributed databases or sharding techniques to spread the load across multiple database servers. Techniques such as database partitioning and replication can also help manage large datasets and improve read/write

performance. The system should also employ caching strategies, using technologies like Redis or Memcached, to reduce the load on the database by storing frequently accessed data in memory. By ensuring the database layer is scalable, the system can maintain fast and reliable data access as demand increases.

To maintain scalability, the barbershop booking system must undergo continuous performance testing. This involves regular load testing, stress testing, and capacity planning to identify potential scalability issues before they impact users. Performance testing tools can simulate varying levels of user load and transaction volumes to evaluate how the system performs under different conditions. The results of these tests can inform decisions about scaling infrastructure, optimizing code, and improving database performance. By continuously testing and optimizing for scalability, the system can ensure it remains responsive and reliable as user demand grows.

Adopting a microservices architecture can significantly enhance the scalability of the barbershop booking system. In a microservices architecture, the system is divided into small, independent services that can be developed, deployed, and scaled individually. This approach allows different parts of the system to scale according to their specific demands. For example, the booking service can be scaled independently of the user management service. Microservices also facilitate better fault isolation, meaning that issues in one service do not impact the entire system. By leveraging a microservices architecture, the system can achieve greater flexibility, maintainability, and scalability.

3.3.3.6 Maintainability

The barbershop booking system should be designed with a modular architecture to enhance maintainability. This involves breaking down the system into smaller, self-contained modules that handle specific functionalities, such as user management, appointment booking, payment processing, and feedback handling. Each module should have well-defined interfaces and be loosely coupled with other modules. This modular approach

allows developers to update or modify individual components without affecting the entire system. It also facilitates easier testing and debugging, as issues can be isolated within specific modules. By adopting a modular architecture, the system can be more easily maintained and scaled, ensuring long-term sustainability.

Maintaining high code quality is essential for the long-term maintainability of the system. The codebase should follow industry best practices, including adherence to coding standards, use of clear and consistent naming conventions, and implementation of comprehensive comments and documentation. Code reviews should be conducted regularly to ensure that new code meets these standards and to identify potential issues early. Additionally, thorough documentation should be maintained, including both inline comments within the code and external documentation covering the system's architecture, data models, APIs, and user guides. Good documentation ensures that new developers can quickly understand and contribute to the project, reducing the time and effort required for maintenance.

Automated testing is a critical component of maintainability, ensuring that the system remains reliable and functional as changes are made. The barbershop booking system should implement a comprehensive suite of automated tests, including unit tests, integration tests, and end-to-end tests. Unit tests should cover individual components and functions, ensuring they perform as expected in isolation. Integration tests should verify that different modules and services work together correctly, while end-to-end tests should simulate real user scenarios to validate the overall functionality of the system. Automated testing helps detect and fix issues early, reduces the risk of regressions, and ensures that the system remains stable and maintainable over time.

Implementing a CI/CD pipeline is essential for maintaining the system efficiently. A CI/CD pipeline automates the process of integrating code changes, running tests, and deploying updates to production. This automation

ensures that new features, bug fixes, and improvements can be delivered rapidly and reliably. The pipeline should include steps for building the application, running automated tests, performing static code analysis, and deploying to staging and production environments. By using CI/CD practices, the system can achieve faster release cycles, reduce manual intervention, and maintain a high level of quality and reliability. This approach also allows for quick rollback of changes if issues are detected, minimizing downtime and disruption.

Effective version control and change management practices are crucial for maintaining the system over time. The barbershop booking system should use a version control system (VCS) like Git to manage the source code. This allows for tracking changes, collaborating with multiple developers, and maintaining a history of code modifications. Branching and merging strategies should be employed to manage parallel development efforts and ensure that stable code is always present in the main branch. Change management processes should be in place to review, approve, and document changes to the system. This includes maintaining a change log, conducting impact analysis, and ensuring that changes are tested and validated before deployment. By following robust version control and change management practices, the system can be maintained systematically and predictably.

Continuous monitoring and logging are essential for identifying and addressing issues in a timely manner. The system should include comprehensive logging of all critical actions, errors, and performance metrics. Logs should be centralized and easily accessible for analysis. Additionally, real-time monitoring tools should be implemented to track system performance, detect anomalies, and alert administrators to potential issues. These tools can provide insights into the system's health and usage patterns, enabling proactive maintenance and quick resolution of problems. By maintaining effective monitoring and logging practices, the system can ensure high availability, performance, and reliability, contributing to overall maintainability.

3.3.4 Other Requirement

a. Laptop

Figure 3.7 shows application is developed in android studio software that installed in laptop.

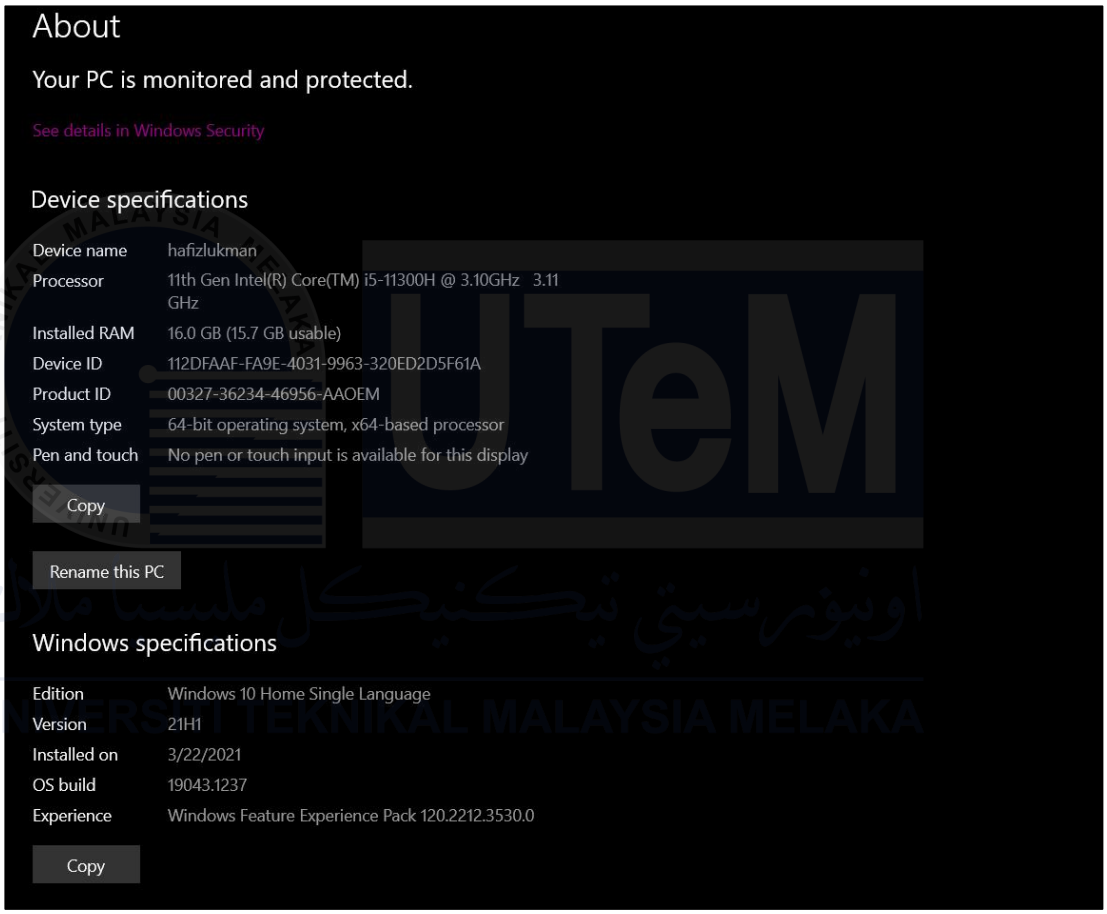

Figure 3.7 Laptop Specification

b. Smartphone

Figure 3.8 shows application should be visible in virtual phone in software.

Figure 3.8 Smartphone Software Specification

c. Android Studio

Figure 3.9 show that, to develop this application the coding will be done in this software

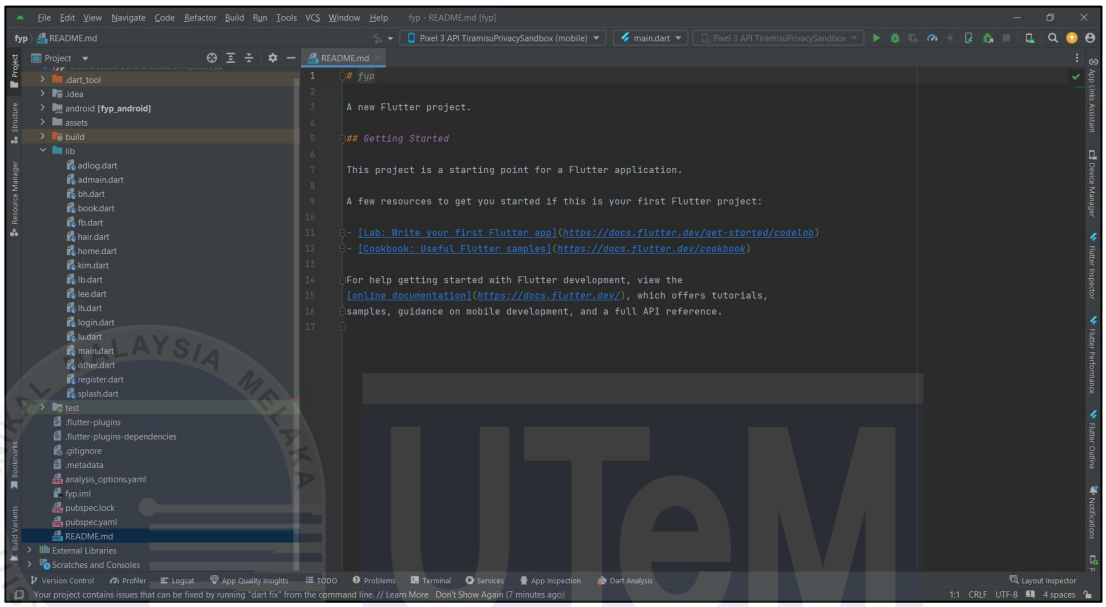

Figure 3.9 Android Studio

d. Firebase Console

Figure 3.10 shows that the system is using Firebase Console as datastore for the application.

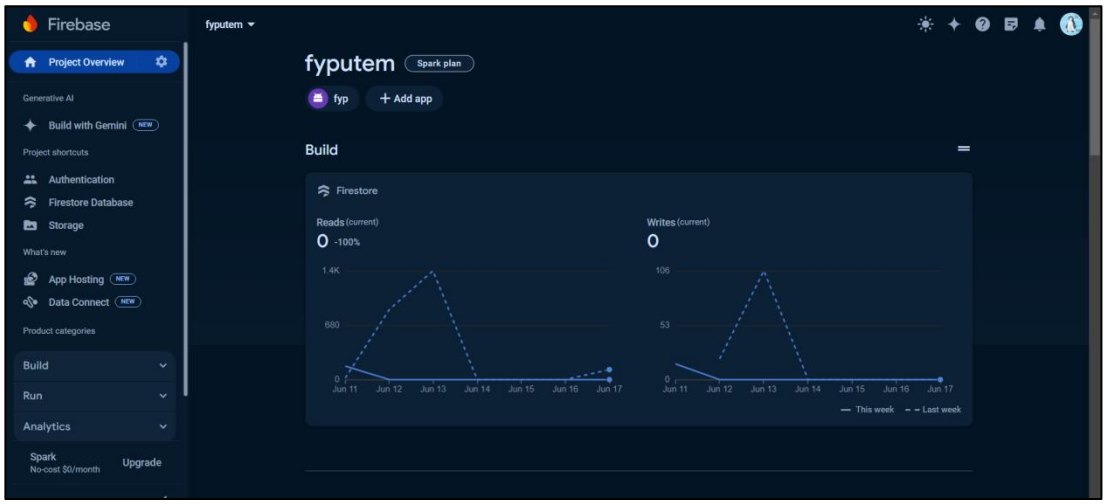

Figure 3.10 Firebase Console

3.4 Conclusion

In conclusion, the barbershop booking system must meet a comprehensive set of functional and non-functional requirements to ensure it operates effectively and provides a superior user experience. The functional requirements encompass user registration, login, appointment booking and management, as well as detailed barber profiles. The non-functional requirements address critical aspects such as usability, reliability, security, performance, scalability, and maintainability. Additionally, the system should be guided by well-defined flowcharts and an ERD diagram, which provide a clear blueprint for its architecture and processes. These visual tools help in understanding the system's workflow, ensuring that all components are seamlessly integrated and work together efficiently. By adhering to these requirements and utilizing the flowcharts and ERD, the system will be robust, secure, and scalable, capable of handling increasing demand and providing seamless service. Furthermore, with a focus on modular architecture, high code quality, automated testing, and effective monitoring, the system will be maintainable, ensuring long-term sustainability and adaptability to evolving needs.

CHAPTER 4: DESIGN

4.1 Introduction

The design phase is a critical stage in the development of the barbershop booking system, where the requirements and specifications are translated into a detailed blueprint for implementation. This phase involves creating the architectural framework, designing the database schema, developing user interface prototypes, and specifying the interactions between various system components. The goal is to ensure that the system is built on a solid foundation that supports its functionality, performance, and scalability requirements. Effective system design helps in identifying potential issues early, promotes better resource management, and ensures that the final product meets user expectations and business objectives. In this section, we will delve into the key design aspects of the barbershop booking system, including system architecture, database design, user interface design, and interaction diagrams, to provide a comprehensive plan for development and deployment.

4.2 High Level Design

RazorReady is a mobile application developed using Android Studio, designed to streamline the process of booking appointments for barbershop services. The system's high-level design comprises three core layers which is the Presentation Layer, which provides an intuitive user interface and user experience through activities and fragments for viewing barbers, selecting appointment slots, and managing profiles. Next, the Business Logic Layer, which handles core functionalities such as appointment scheduling, user authentication, and service catalog management to ensure seamless operations and secure transactions and the Data Layer, which manages data storage and retrieval using local databases and network modules for syncing with a remote server, ensuring data consistency and integrity. The application follows the MVVM architectural pattern, utilizing Android Jetpack components like LiveData and ViewModel for efficient data binding and lifecycle management.

4.2.1 System Architecture

Figure 4.1 shows a System Architecture for a barber booking system developed using Android Studio, catering to both users and administrators. Users can log in, register, make bookings, view history, check barber profiles, and give feedback. Administrators can log in, delete users and feedback, edit details, update booking slots, and delete bookings. The backend script handles Create, Read, Update, and Delete operations, communicating with the server to manage the database. Data from user and admin actions is sent to the server for processing and stored or retrieved from the database, ensuring synchronized and up-to-date information across the system.

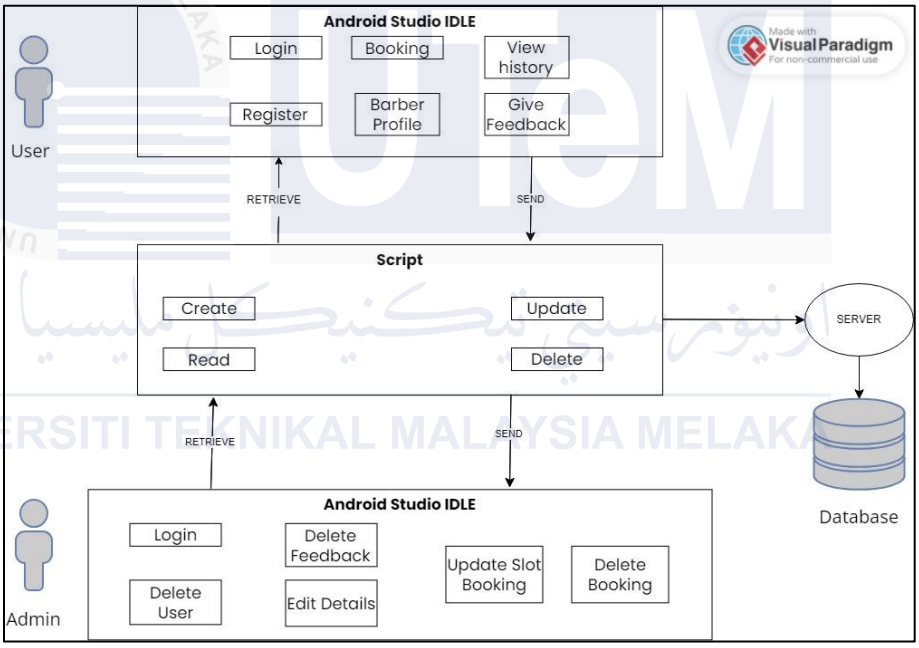

Figure 4.1 System Architecture

4.2.2 User Interface Design

The user interface design for the RazorReady features a cohesive and user-friendly layout across its login and registration screens. Based on the figure 4.2 and figure 4.3, on the login screen, a dark background sets the stage for the prominent display of the app's logo, "Razor Ready Barber," complete with a scissor icon. Below the logo, users are invited to enter their email and password in two vertically aligned input fields, followed by large, pink buttons labeled "Login" and "Register" for easy navigation. The registration screen maintains the dark theme and clean design, with a title "Register" at the top and three input fields for username, email, and password. This screen also includes two large, pink buttons labeled "Register" and "Back to Login," allowing users to create a new account or return to the login screen. Consistency in color scheme, typography, and button design across both screens ensures high visibility and a seamless experience for users.

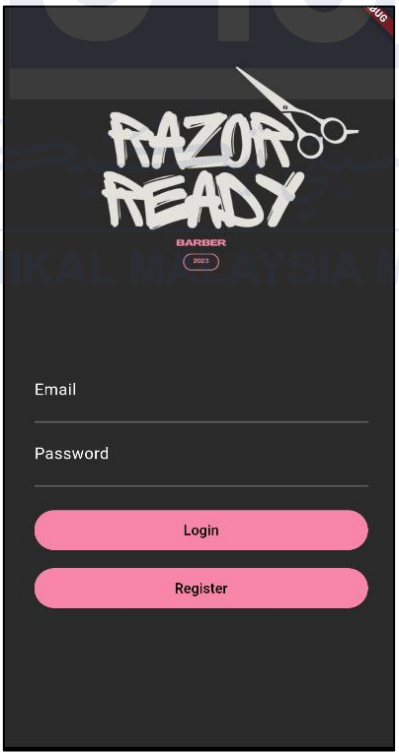

Figure 4.2 Login Page

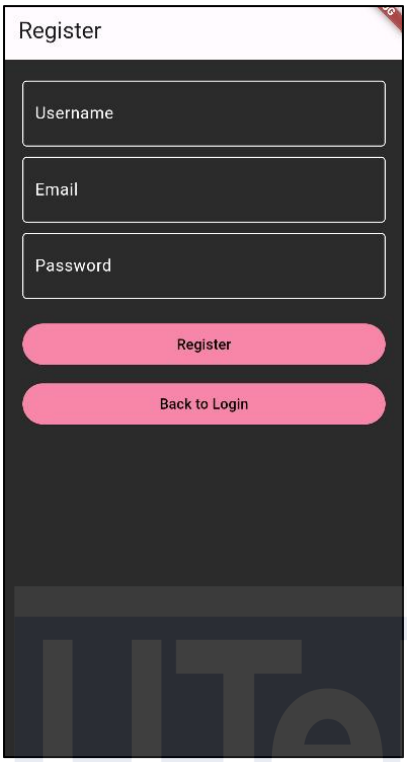

Figure 4.3 Register Page

(a) Input Design

Figure 4.4 shows the input design for the RazorReady app's, there are two text input fields for the user to enter their email and password. The email field is pre-filled with "amad@gmail.com," and the password field is obscured for security by using symbol “*”. Both input fields are clearly labeled and aligned vertically, providing a clean and organized appearance. Underneath these fields, two large, pink buttons labeled "Login" and "Register" are presented for easy interaction, guiding the user to either log into their account or navigate to the registration screen.

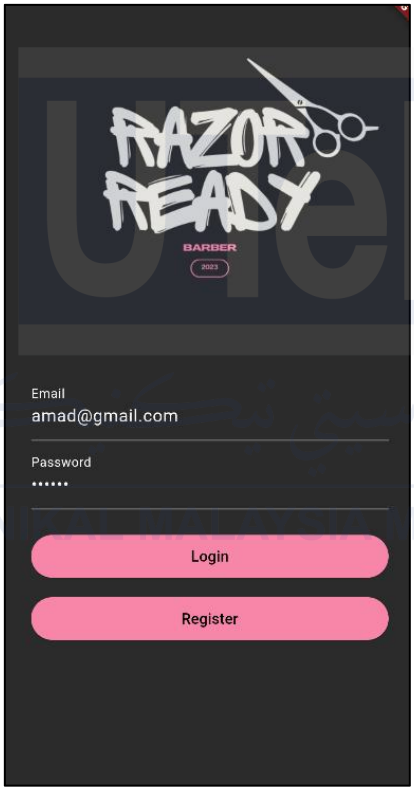

Figure 4.4 Login Page Input

According to Figure 4.5 below, a layout with a title and back arrow for easy navigation. Users can enter their phone number in a text field, select multiple services such as Haircut, Shave, Facial, Beard Trim, Hair Coloring, Hair Wash using checkboxes, and choose their preferred barber which is Barber Lee or Barber Kim via radio buttons accompanied by profile images. Available appointment times are displayed in a color-coded grid green for available, red for unavailable in one-hour increments, such as 12:00pm - 1:00pm. At the bottom, a prominent pink "Book Appointment" button allows users to finalize their booking easily. The design's use of contrasting colors, clear labels, and intuitive input elements ensures an efficient and seamless appointment scheduling process.

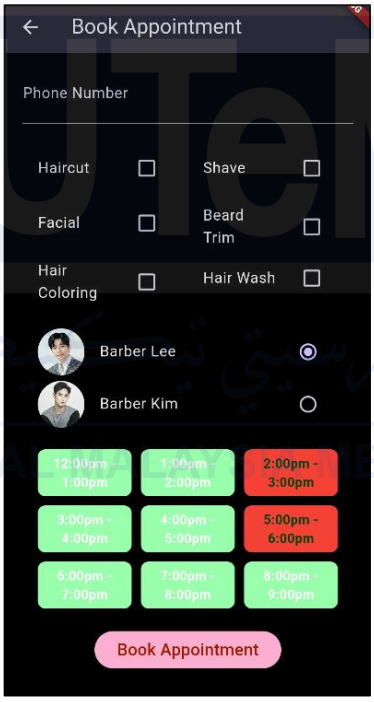

Figure 4.5 Booking Page

(b) Output Design

Figure 4.6 shows the "Booking List" for admin page features a clean and straightforward design, displaying a series of booking entries in a card-like format. Each entry contains the customer's name, the assigned barber, the services to be provided, and the appointment time. The entries are organized vertically, making it easy to scan through the list. A trash can icon is positioned to the right of each entry, indicating the option to delete the booking. At the top, a back arrow allows navigation to the previous page, and the page title "Booking List" is centered at the top, providing a clear indication of the page's purpose.

Figure 4.6 Booking List

4.2.3 Database Design

Database design refers to the process of creating a well-structured and organized database system that efficiently stores, manages, and retrieves data. It involves defining the logical and physical structure of the database, including tables, relationships, and constraints, based on the requirements and objectives of the system. The design process includes identifying entities, attributes, and relationships between them, and then translating them into a schema. This schema provides a blueprint for creating and maintaining the database. Effective database design aims to ensure data integrity, minimize redundancy, optimize performance, and facilitate easy data manipulation and retrieval. It requires careful consideration of factors such as data types, normalization, indexing, and security to create a robust and scalable database solution.

4.2.3.1 Conceptual and Logical Database Design

Conceptual database design for a barbershop booking app involves identifying and organizing the high-level entities and relationships essential to the application, such as Users, Barbers, Services, Appointments. It focuses on defining the entities and their attributes without detailing the specific implementation or database structures. Logical database design, on the other hand, translates the conceptual model into a more detailed schema, specifying tables, columns, primary and foreign keys, and constraints. For instance, it would detail how Users have unique IDs and profiles, Barbers offer specific Services with associated prices, and Appointments link Users to Barbers for specific times, ensuring efficient data storage and retrieval tailored to the app's functionality and performance requirements.

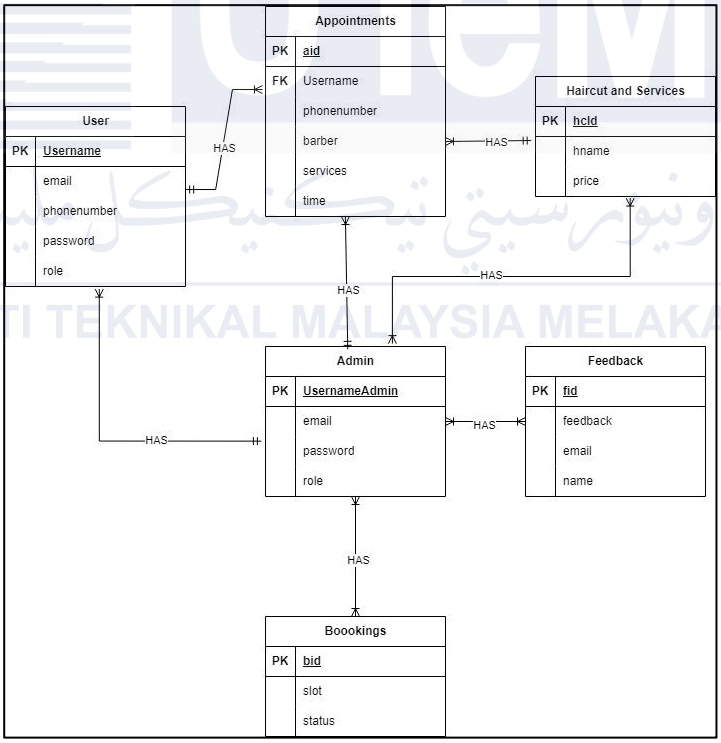

Figure 4.7 ERD Diagram

Figure 4.7 explain The provided Entity-Relationship Diagram (ERD) represents a conceptual and logical database design for a system that manages appointments, users, services, feedback, and bookings, possibly for a barbershop or salon. Each entity is clearly defined with its attributes and relationships to other entities, indicating the structure and rules governing the database.

In the conceptual design, the key entities identified are User, Admin, Appointments, Haircut and Services, Feedback, and Bookings. These entities represent the primary objects that need to be managed by the system. The User and Admin entities are central to the system, capturing the essential information for users and administrators, such as usernames, emails, passwords, and roles. The Appointments entity manages the scheduling details, while the Haircut and Services entity lists the available services and their prices. Feedback captures user reviews, and Bookings keep track of slots and their statuses.

In the logical design, each entity is translated into a table with specified primary keys (PK) and foreign keys (FK) to establish relationships. For instance, the User entity includes attributes like username, email, phone number, password, and role. The Appointments entity includes an appointment ID (aid), username, phone number, barber, services, and time. The Haircut and Services entity includes a haircut ID (hcid), haircut name, and price. Feedback includes a feedback ID (fid), feedback text, email, and name, while Bookings include a booking ID (bid), slot, and status.

The relationships between entities are crucial for maintaining data integrity and enabling efficient data retrieval. The ERD shows that a User can have multiple appointments (one-to-many relationship), and each appointment can reference a particular service from the Haircut and Services table. The Admin entity is related to the Appointments, Bookings, and Feedback tables, indicating that admins can manage appointments, bookings, and feedback. Feedback is linked to the user providing it, ensuring that each feedback entry is associated with a valid user.

Primary keys (PK) ensure that each record in a table is unique. For example, 'username' is the primary key in the User table, ensuring that no two users can have the same username. Foreign keys (FK) establish connections between tables, such as the 'username' in the Appointments table linking back to the 'username' in the User table. This ensures that appointments are always associated with valid users. Similarly, the 'hcid' in the Appointments table links to the 'hcid' in the Haircut and Services table, ensuring that appointments refer to valid services.

The design enforces data integrity through constraints such as primary keys, foreign keys, and the relationships between tables. These constraints ensure that the data remains consistent and valid across the system. For instance, an appointment cannot be created without a valid user and service, and feedback cannot be submitted without a corresponding user. These constraints help maintain the reliability and accuracy of the data, ensuring that the system functions as intended and supports the business processes effectively.

In summary, the ERD illustrates a well-structured database design that captures the key entities and their relationships in a system designed to manage users, appointments, services, feedback, and bookings. The use of primary and foreign keys, along with defined relationships, ensures data integrity and supports efficient data management.

Data Dictionary

Table 4.1 User

Field Name	Data Type	Field Lenght	Constraint	Description
username	boolean	20	primary	username
email	string	20	Not null	email
pasword	string	20	Not null	pasword
role	string	20	Not null	role

Table 4.2 Admin

Field Name	Data Type	Field Lenght	Constraint	Description
adusername	boolean	20	primary	adusername
adrole	string	20	Not null	adrole
ademail	string	20	Not null	ademail
adpassword	string	20	Not null	adpassword

Table 4.3 Booking

Field Name	Data Type	Field Lenght	Constraint	Description
bid	string	20	primary	bid
slot	string	20	Not null	slot
status	string	10	Not null	status

Table 4.4 Feedback

Field Name	Data Type	Field Lenght	Constraint	Description
fid	string	20	primary	fid
email	string	20	Not null	email
feedback	string	50	Not null	feedback
name	string	20	Not null	name

Table 4.5 Appointments

Field Name	Data Type	Field Lenght	Constraint	Description
aid	string	20	primary	aid
username	string	20	foreign	username
phone	string	10	Not null	phone
barber	string	20	Not null	barber
services	string	20	Not null	services
time	string	20	Not null	time

Table 4.6 Haircut and Services

Field Name	Data Type	Field Lenght	Constraint	Description
hcid	string	20	primary	hcid
hname	string	20	Not null	hname
price	int	3	Not null	price

4.3 Detailed Design

Here are the diagrams for detailed design which provide how the application is fit together and produce physical description of all part of the structure. A sequence diagram is shown in figure below how the system will proceed

4.3.1 Sequence Diagram

Figure 4.8 shows the process starts with the user initiating the login process by providing their email and password. This information is sent to the Login system component, which then forwards the credentials to the Validate component for verification. The Validate component checks the provided email and password against the records stored in the Database. If the email is found and the password matches, the validation is successful. The Validate component confirms the successful validation back to the Login component, which then returns a success message to the user. Finally, the user is presented with the home page.

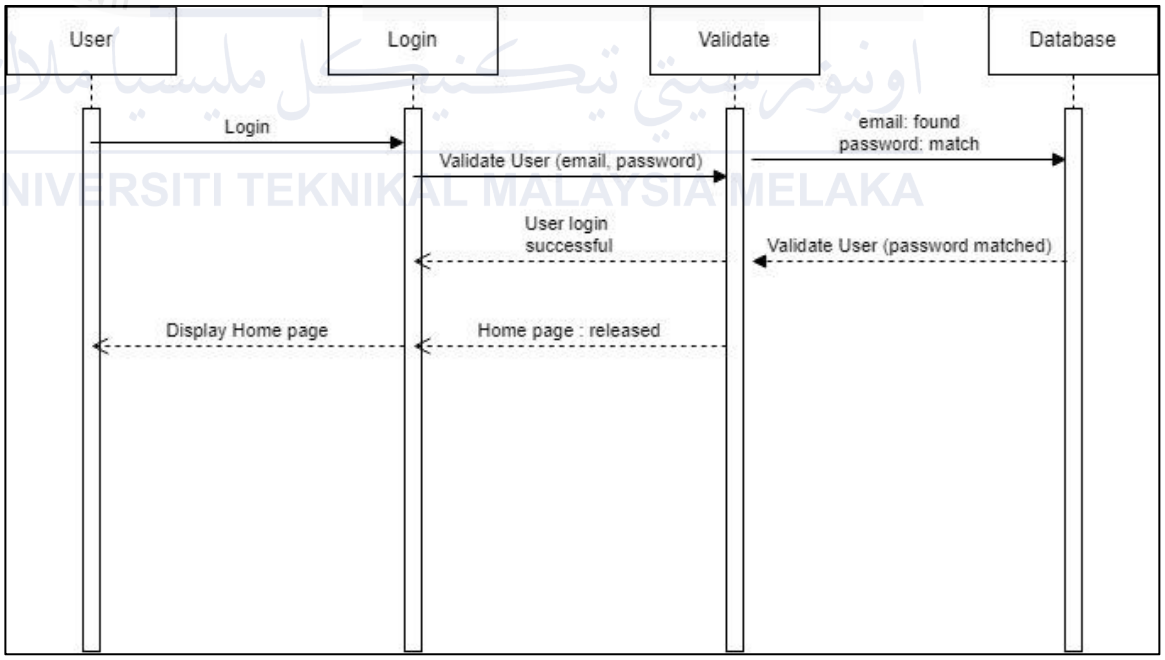

Figure 4.8 Login User

Figure 4.9 illustrates the sequence of actions involved in booking an appointment. The process begins with the user logging into the system and starting the booking process. The user can look up barber profiles, available hairstyles, and services through the Booking system component. The Booking component then checks for available free slots for booking. The Admin component sends the booking status back to the Booking component. Once a slot is confirmed, the Booking component confirms the booking with the user. The Admin component then sends the detailed booking information back to the user, and the user's booking history is updated accordingly.

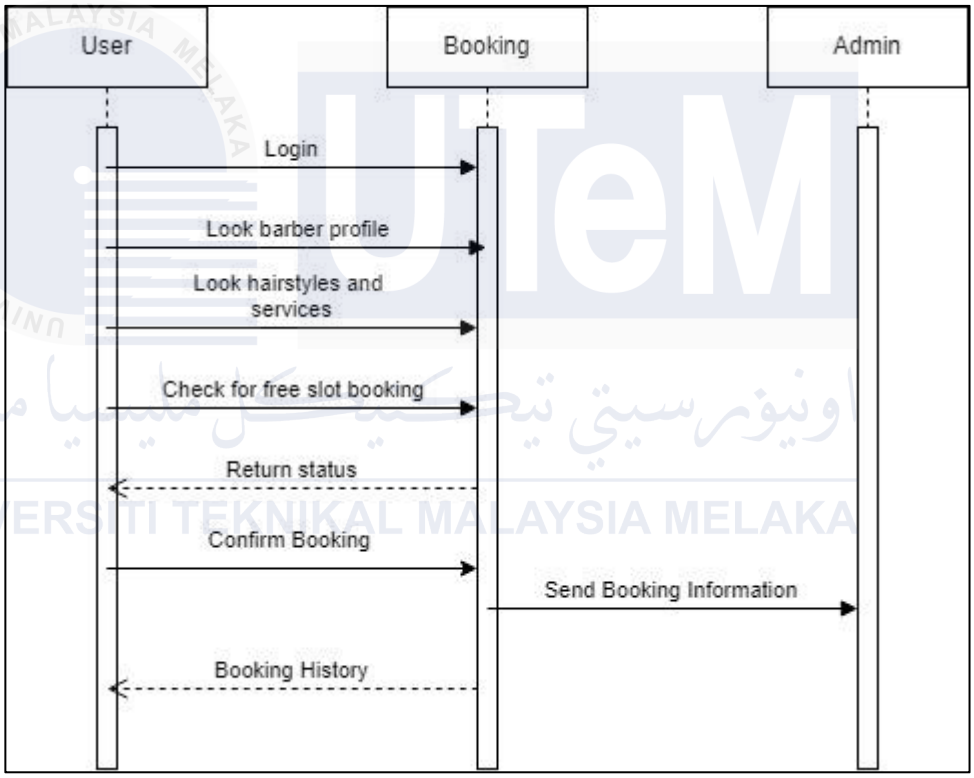

Figure 4.9 Booking Appointment

Figure 4.10 shows the user views their booking history. The user initiates a request to view the history by interacting with the Appointments component. The Appointments component then requests the list of appointments from the Database. The Database retrieves and provides the list of appointments back to the Appointments component. Finally, the user is presented with their appointment history, displaying all past appointments.

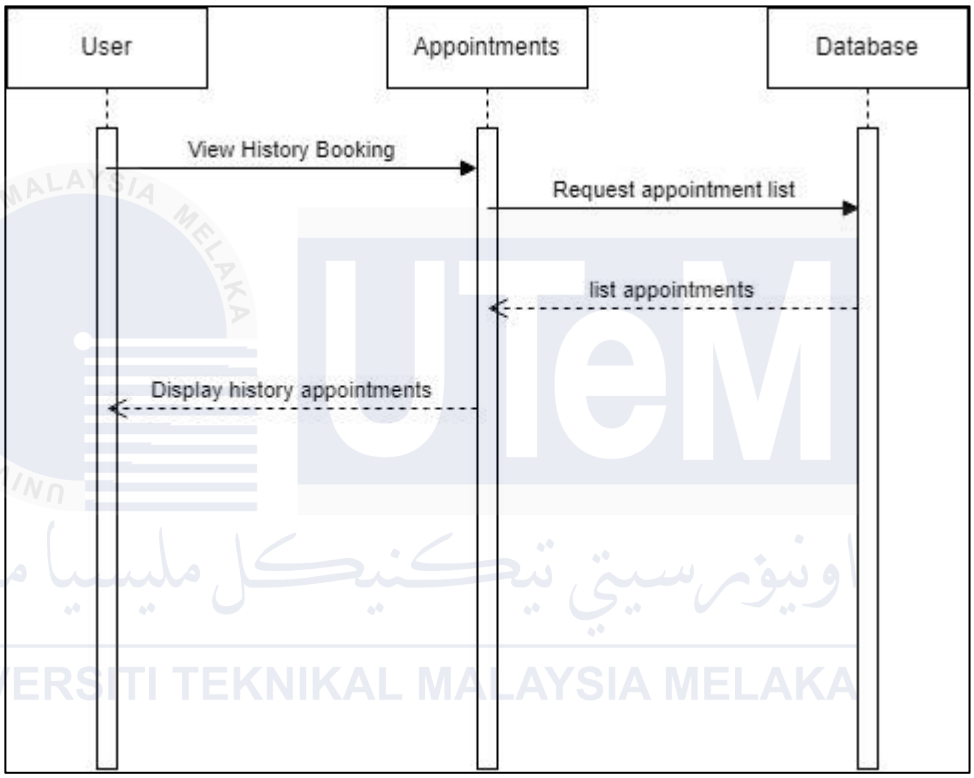

Figure 4.10 History Booking

4.3.2 UML Diagram

Figure 4.11 shows a UML diagram for RazorReady. This UML diagram contains several classes which is user, admin, feedback, hairstyleservices, appointments and booking.

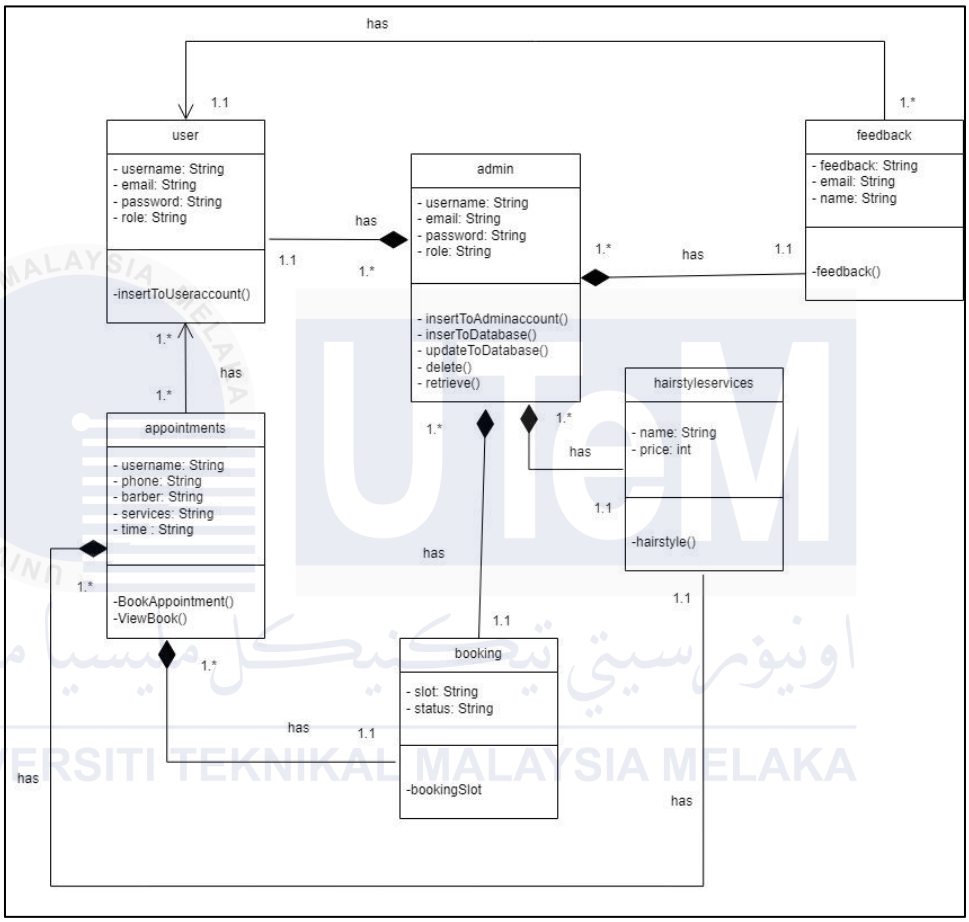

Figure 4.11 UML Diagram

4.4 Conclusion

This chapter explain all the design such as database design, system architecture, user interface design and etc. In conclusion, the high-level design of the Barbershop Booking System is a robust and scalable architecture that ensures efficient, secure, and user-friendly operation. By utilizing a layered structure comprising the Presentation, Business Logic, and Data layers along with the MVVM architectural pattern, the design promotes clear separation of concerns, maintainability, and enhanced user experience. The integration of Android Jetpack components ensures efficient data handling and lifecycle management, while third-party services for payments and notifications add valuable functionality. This design not only meets the current requirements but also lays a strong foundation for future development and scalability.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

The practical realisation of a plan or idea, comprising the translation of concepts into physical activities and consequences, is referred to as implementation. It includes the design, development, testing, and integration stages that lead to the deployment of a product, method, or solution. To guarantee that the final outcome matches with the initial goals and criteria, this process requires precise attention to detail, appropriate resource allocation, and continual review

The barber appointment mobile application system is designed to streamline the booking process for customers and barbers alike. It enables users to easily schedule appointments, select preferred services, and choose their favorite barbers through a user-friendly interface. The system also provides barbers with tools to manage their schedules, track bookings, and enhance customer interaction. By integrating essential features such as real-time availability, appointment reminders, and service customization, the application aims to improve the overall efficiency of the barbering business.

5.2 Software Development Environtment Setup

To develop the barber appointment mobile application system, Android Studio is the chosen integrated development environment (IDE). Begin by installing the latest version of Android Studio, which includes the Android SDK, emulators, and necessary build tools. Ensure that it got the latest Flutter and Dart SDKs installed. Additionally, configure Android device or emulator for testing, and set up version control using Git to manage codebase effectively. This setup provides a robust and efficient environment for developing, testing, and deploying the mobile application.

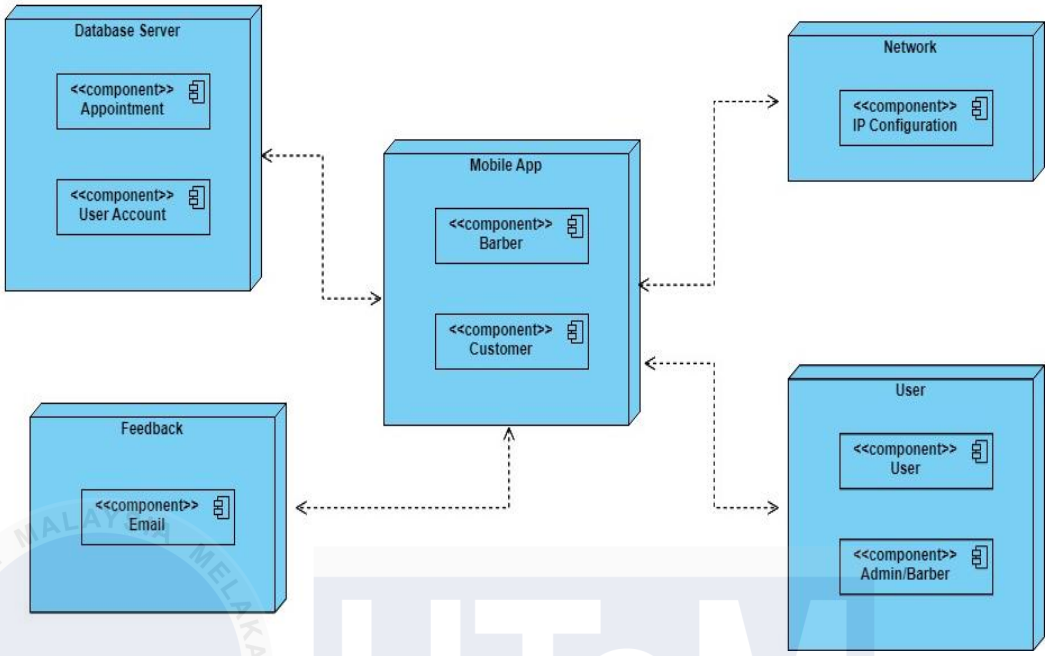

Diagram 5.1 Deployment Diagram

5.3 Software Configuration

SCM (Software Configuration Management) is a collection of practices and methods that allow for the systematic management of software artefacts, modifications, and versions across the software development lifecycle. By offering tools and procedures for version control, change tracking, documentation, and cooperation among development teams, SCM attempts to maintain the integrity, traceability, and control of software assets. It improves the overall quality and stability of software systems by enabling effective cooperation, minimizing disputes, and ensuring the availability of consistent and correct software deliverables.

5.3.1 Managements

This chapter will go through how the Dental Management Application is created and how configuration management is set up in this project. It is concerned with the software that must be put in the device prior to the development of the system.

5.3.1.1 Android Studio

Android Studio is a full-featured integrated development environment (IDE) for developing, testing, and delivering Android apps. It was created by Google and provides a feature-rich environment targeted to Android app development. Android Studio has an intuitive user interface for creating user interfaces, developing code, debugging, testing, and optimizing apps. It includes advanced features such as a virtual device emulator for testing, a layout editor for building UI designs, and a robust code editor with intelligent suggestions and real-time error checking. By integrating with the Android SDK, providing simple access to libraries and APIs, and supporting version control systems, Android Studio speeds the development process. It has become the preferred platform for Android developers, increasing productivity and enabling the development of high-quality and efficient Android applications.

5.3.1.2 Database Configuration

RazorReady utilizes Firebase Firestore as its primary database, providing a scalable and real-time cloud-based solution. Firestore's NoSQL structure allows for flexible and efficient data storage, enabling the system to manage collections such as users, barbers, appointments, and services. Each collection contains documents that store key-value pairs representing various attributes, like customer details, appointment times, and service options. The database configuration includes security rules to control data access and ensure that only authenticated users can read or modify data. Real-time synchronization allows the application to instantly reflect any changes across all connected clients, enhancing the user experience with up-to-date information on appointments and availability.

5.3.2 Version Control Procedure

Version control is a method of managing changes to software projects that allows for effective collaboration and tracking of code updates over time. It entails creating a repository that holds the project's codebase using version control technologies such as Git. Developers can then produce commits, which are snapshots of the code at various points and document changes done. These commits are organized into a history, which allows for simple code comparison, restoration, and branching. Version control promotes collaboration, aids in dispute resolution, and provides a disciplined approach to software development, improving code quality and project management. This are the steps for version control

i. Create an Android Studio Project

ii. Integrating Git – After setup, click VCS menu then click import into Version Control and select Create Git Repository.

iii. Integrating with GitHub – Navigate to File > New > Project from Version Control > GitHub.

iv. The Version Control Window will appear.

5.4 Implementation Status

Table below shows the process of the development status for each of the module

Table 5.1: Implementation Status

Module	Description	Duration To Complete	Date Complete
Registration	To ensure the user can register and the registered data successfully saved in database	1 Week	2/4/2024
Login	To ensure the user login and open the main page based user role	1 Week	9/4/2024
Splash Screen	The splash screen shows a company logo that appear around for a few seconds.	1 Week	9/4/2024
Main Page	The main page that a User will view when they successful login to a system.	1 Week	16/4/2024
Barber Profile	Page that show bio barber and details about barber.	1 Week	23/4/2024
Service and Hairstyle Page	Page that show hairstyle and other services that have been provided in the barbershop.	2 Week	6/5/2024
Book Appointment	To ensure user can book the appointment and can view the slot availability dates and session. To ensure user can choose the services their need.	5 Week	11/6/2024

Admin Page	Page for admin to check or update total sale, user account, and booking.	2 Week	11/6/2024
View Appointment	To ensure the user can view appointment detail after successfully make a booking.	1 Week	21/8/2024

5.5 Conclusion

In conclusion, RazorReady is a comprehensive solution that streamlines the booking and management process for both customers and barbers. Developed using Android Studio, the application leverages Firebase Firestore for real-time data management, ensuring efficient handling of appointments, user profiles, and services. With its flexible database setup and robust infrastructure, the system is poised to offer a seamless and user-friendly experience, enhancing the overall efficiency and customer satisfaction in the barbering business.

CHAPTER 6: TESTING

6.1 Introduction

Testing software documentation is an important step in the software development lifecycle since it assures the correctness, completeness, and usefulness of the numerous documents developed during the development process. This comprises test plans, test cases, user manuals, and other software project documentation. The primary purpose of testing documentation is to find and correct any contradictions, flaws, or inconsistencies in the documentation to verify that it corresponds to the software's real operation. This process entails examining, validating, and verifying the documentation against the needs and features of the product. Teams may improve communication, decrease misunderstandings, and support a smoother software development process by checking software documentation, resulting in a higher quality end product.

6.2 Test Plan

A test plan is a detailed document that specifies the testing strategy, scope, objectives, and technique for a software application or system. It acts as a road map for the whole testing process, including what needs to be tested, how it will be tested, the resources needed, and the testing activity calendar. The test plan also specifies the testing environment, tools, and success criteria, as well as the testing team's roles and duties. This document is critical in ensuring that testing activities are organized, methodical, and linked with project goals, assisting in the identification and management of risks, tracking progress, and ultimately delivering a high-quality software product.

6.2.1 Test Organization

The organized arrangement and administration of testing operations inside a software development project is referred to as test organization. It entails identifying the roles, responsibilities, procedures, and resources required to successfully plan, execute, and manage testing initiatives. A well-organized testing methodology ensures that testing objectives are explicit, that testing assignments are given to the appropriate team members, and that testing activities are in sync with the development process. Test organization also includes test documentation, test environments, and communication routes to let team members and stakeholders collaborate. This organization contributes to the delivery of a stable and high-quality software product by maintaining efficiency, transparency, and accountability during the testing process.

Table 6.1 below depicts the test organization table, which includes detailed information on the individuals engaged in the test stages.

Table 6.1: Test Organization

Test ID	Name	Role	Responsibility
T01	Muhammd Hafiz Lukman Bin Nuri Ansarizam	System Developer	Responsible for developing, executing, and testing integration and testing components.
		Software Tester	Act as both an employer and an employee in testing each module that has been designed.

		End User	Responsible for testing as an end user, who is a company or organization employee.

6.2.2 Test Environment Setup

A test environment is the infrastructure, hardware, software, and settings that are put up to allow software testing operations. It is a controlled and isolated environment that is intended to simulate the real-world conditions in which the codingr will work. Software is rigorously tested for functionality, performance, and compatibility across multiple platforms and settings in test environments. Creating environments for unit testing, integration testing, system testing, and user acceptability testing is part of this. A well-managed test environment is required for performing reliable and reproducible tests, discovering and resolving errors, and verifying that software performs successfully in various circumstances before it is released to production.

6.2.2.1 Environment Setup

The environment setup is a framework for managing the testing for this mobile application, which is to guarantee that all of the modules in this app work properly

6.2.2.2 Application Software

A software application is a computer coding or group of coding that performs certain activities, functions, or services for end users. Table 6.2 below shows the application that used in this system or application.

Table 6.2: Application Software

Application Software	Login
	Register
	Booking Appointment
	View Appointment
	Manage User
	Manage Appointment

6.2.2.3 System Software

System software is a type of computer software that is designed to manage and ease the functioning of a computer system and its physical components. Table 6.3 below shows the software that used in this system or application.

Table 6.3: System Software

System Software	Android Studio
	Firebase Console

6.2.2.4 System Hardware

The equipment required to construct this mobile application is known as system hardware. Table 6.4 below shows the hardware that used in this system or application.

Table 6.4: System Hardware

System Hardware	Laptop

6.2.3 Test Schedule

A test schedule is a precise calendar or plan that explains when certain testing activities inside a software development project will take place. It defines the beginning and finish dates for several testing phases, such as unit testing, integration testing, system testing, and user acceptability testing. A test schedule also allocates resources, defines dependencies, and establishes milestones to measure progress. It assists project managers and testing teams in ensuring that testing activities are aligned with the overall project timetable and that adequate time is allotted for each testing phase. A well-structured test plan is essential for project risk management, meeting deadlines, and delivering a high-quality software product on time.

Table 6.5 below shows the scheduled date to finish the testing for each module

Table 6.5: Test Schedule

Test Modul	**Start Date**	**End Date**	**Duration**
Login	22/8/2024	22/8/2024	1 Day
Register	22/8/2024	22/8/2024	1 Day
Booking Appointment	23/8/2024	23/8/2024	1 Day
View Appointment	24/8/2024	24/8/2024	1 Day
Manage User	24/8/2024	24/8/2024	1 Day

Manage Appointment	24/8/2024	24/8/2024	1 Day

6.3 Test Strategy

A test plan is a detailed document that describes the overall methodology, goals, and recommendations for testing activities inside a software development project. It is a high-level plan that defines the overall vision for how testing will be carried out to assure the delivery of a high-quality software product. A well-crafted test strategy will typically include information on the testing scope, testing objectives, entry and exit criteria, test levels (such as unit testing, integration testing, system testing, and acceptance testing), test environment requirements, resource allocation, test automation strategy, and risk assessment. It also includes stakeholder identification, roles and duties, and communication routes throughout the testing process.

Throughout the project's lifespan, the test strategy offers a framework for making crucial decisions about test design, execution, and assessment. It guarantees that testing is in accordance with project objectives, client needs, and industry standards. A clear and well-communicated test strategy helps teams to successfully plan, execute, and manage testing operations, improving the quality and dependability of the final software product while effectively managing risks and resources. Table 6.6 shows the description of each approach for testing strategy.

Table 6.6: Test Strategy

Approach	Description
White Box Testing	White box testing is a software testing method that involves examining the internal structure, logic, and code of an application. Testers with access to the source code use it to develop test cases that verify the correctness of specific code components, paths, and data flows within the application. This approach aims to uncover issues such as logic errors, boundary conditions, and control flow problems. White box testing is often used by developers during unit testing to ensure that each module or function performs as expected and that the overall product meets its design and functionality requirements.
Black Box Testing	Black box testing is a software testing technique where the tester evaluates an application's functionality without access to its internal code or structure. Acting as an external user, testers interact with the software by providing inputs and observing the outputs to determine whether the application functions correctly and meets its specified requirements. This approach focuses on the software's usability, functionality, and compatibility with various inputs and scenarios, rather than its internal

	workings. Black box testing is effective in identifying issues like incorrect behavior, user interface problems, and performance bottlenecks, helping to ensure that the software meets user expectations and performs as intended in real-world conditions.

6.3.1 Classes of Test

The many classes or degrees of testing are described in a test strategy to give an organized way to analyzing a software application. A test strategy's common kinds of tests may include as depicted in table 6.7:

Table 6.7: Classes of Test

Test Case ID	**Test Case**
Integration Testing	Focuses on testing how different software modules or components interact when combined. It ensures that they work together effectively as a system and communicate properly.
System Testing	This comprehensive testing stage assesses the entire system to confirm it meets the specified requirements and functions correctly. It evaluates the software's behavior, performance, and compatibility across various environments.

User Acceptance Testing (UAT)	End-users or stakeholders assess the software to determine if it meets their specific needs and business goals. This testing ensures that the software aligns with user expectations and is ready for production use.
Performance Testing	Measures the software's responsiveness, scalability, and stability under different load conditions. It includes stress testing, load testing, and scalability testing.

6.4 Test Design

Test design is the process of creating detailed test cases and scenarios based on the testing objectives and requirements outlined in the test strategy. This involves specifying the exact inputs, expected outputs, and testing conditions that will be used to thoroughly evaluate the software. The aim of test design is to ensure comprehensive coverage of the application's various functionalities, features, and use cases, identifying potential issues and verifying that the product meets its specifications. Effective test design is crucial for systematically and efficiently uncovering software defects, ensuring reliability and performance while adhering to project timelines and goals.

6.4.1 Test Description

A test description is a detailed document that outlines a specific test case or scenario in software testing. It includes a unique identifier, the test objective, preconditions, input data, expected results, and any specific steps or actions to be taken during test execution. This document provides a structured and comprehensive understanding of what is being tested, the purpose of the test, and how it will be conducted, enabling efficient and effective quality assurance processes to identify issues and ensure the product's reliability and performance.

6.4.1.1 Test Description For Login Module

Table 6.8 below shows a test description of unit testing for Login module.

Table 6.8: Test Description For Login Module

Test Case ID	**Test Case**	**Expected Result**
TCL01	To check the functionality login if enter Invalid username and password	An error message “Please enter correct username and password” must show.
TCL02	To check the functionality login if enter empty username	An error message “Please enter username”
TCL03	To check the functionality login if enter empty password”	An error message “Please enter password”
TCL04	To check the functionality login if enter valid username and password	Redirect the login page to user home page

6.4.1.2 Test Description For Registration Module

Table 6.9 below shows a test description of unit testing for Registration module.

Table 6.9: Test Description For Register Module

Test Case ID	**Test Case**	**Expected Result**
TCR01	To check the functionality registration if enter empty username	An error message “Please enter username”
TCR02	To check the functionality registration if enter empty email	An error message “Please enter email”
TCR03	To check the functionality registration if enter empty password	An error message “Please enter password”
TCR04	To check the functionality registration if enter valid data and equal password	Redirect to login page

6.4.1.3 Test Description For Booking Appointment Module

Table 6.10 below shows a test description of unit testing for Booking Appointment module.

Table 6.10: Test Description for Booking Module

Test Case ID	Test Case	Expected Result
TCB01	To check the functionality booking if no name entered.	An error message “Please enter name”
TCB02	To check the functionality booking if no phone number entered.	An error message “Please enter phone number”
TCB03	To check the functionality booking if no date selected.	An error message “Please select date”
TCB04	To check the functionality booking if no service selected.	An error message “Please select service”
TCB05	To check the functionality booking if no slot selected.	An error message “Please select slot”
TCB06	To check the functionality booking if enter valid data.	Redirect to view appointment page.

6.4.1.4 Test Description For Appointment Booking Module

Table 6.11 below shows a test description of unit testing for View Appointment module.

Table 6.11: Test Description for View Appointment Module

Test Case ID	**Test Case**	**Expected Result**
TCV01	To check the functionality on viewing appointment booking.	Display current appointment booking.
TCV02	To check the functionality on searching booking history by using phone number.	Display booking history.
TCV03	To check the functionality booking if no phone number entered.	An error message “Please enter phone number”

6.4.1.5 Test Description For Manage User Module

Table 6.12 below shows a test description of unit testing for Manage User module.

Table 6.12: Test Description for Manage User Module

Test Case ID	Test Case	Expected Result
TCU01	To check the functionality on Managing User in the System.	Account User have been deleted in the system.
TCU02	To check the functionality on displaying list of user.	Display List of User Account.

6.4.1.6 Test Description For Manage User Module

Table 6.13 below shows a test description of unit testing for Manage Booking module.

Table 6.13: Test Description for Manage Appointment Module

Test Case ID	Test Case	Expected Result
TCM01	To check the functionality on Managing Booking in the System.	Appointmennt have been deleted in the system.
TCM02	To check the functionality on displaying list of appointments.	Display List of Appointment.

6.4.2 Test Data

Test data is a specific collection of inputs, values, or situations that are purposefully selected and utilized to test a software coding. It is an important part of the software testing process since it helps analyze how the coding acts under different scenarios. To guarantee thorough testing coverage, test data contains both valid and incorrect inputs, as well as boundary values, edge situations, and common use scenarios. The purpose is to find and fix any flaws, faults, or unexpected behaviors in the coding so that it works correctly, efficiently, and securely in real-world scenarios. Thorough and successful software testing requires well-designed and diversified test data, which leads to greater software quality and dependability

6.4.2.1 Test Data For Login Module

System	RazorReady	Version	V1
Module	Login	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	22/8/2024

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result
TCL01	Verify to login	The username and password must be valid to login.	1) Enter invalid username 2) Enter invalid password 3) Click Login	Email: amadd@gmail.com Password: 1234567	An error message “Please enter correct email and password”.	An error message “Please enter correct username and password”.
TCL02	Verify to login	The email cannot	1) Enter empty	Email: -	An error message “Please enter	An error message “Please enter

		be empty.	email 2) Enter valid password 3) Click Login	Password: 1234567	email".	username".
TCL03	Verify to login	The password cannot be empty.	1) Enter valid username 2) Enter empty password 3) Click Login	Email: amadd@gmail.com Password: -	An error message "Please enter password".	An error message "Please enter password".
TCL04	Verify to login	Verify the email and password	1) Enter valid username 2) Enter valid	Email: amad@gmail.com	Redirect to Home Page	Redirect to Home Page

			password 3) Click Login	Password: 123456		

6.4.2.2 Test Data For Register Module

System	RazorReady	Version	V1
Module	Register	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	22/8/2024

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result
TCR01	Register for login	The username cannot be empty.	**1)** Enter empty usernname 2) Enter email **3)** Enter password	Username : - Email : amad@gmail.com Password: 123456	An error message “Please enter username”.	An error message “Please enter username”.
TCR02	Register for login	The email cannot be empty.	**1)** Enter usernname 2) Enter empty	Username : Amad	An error message “Please enter	An error message “Please enter

			email 3) Enter password	Email : - Password: 123456	email".	email".
TCR03	Register for login	The password cannot be empty.	**1)** Enter usernname 2) Enter email 3) Enter empty password	Username : Amad Email : amad@gmail.com Password: -	An error message "Please enter password".	An error message "Please enter password".
TCR04	Register for login	Verify Account	**1)** Enter usernname 2) Enter email 3) Enter password	Username : Amad Email : amad@gmail.com Password: 123456	Redirect to Login Page	Redirect to Login Page

6.4.2.3 Test Data For Booking Appointment Module

System	RazorReady	Version	V1
Module	Booking Appointment	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	22/8/2024

Test ID	**Test Scenario**	**Test Case**	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**
TCB01	Make an appointment	User need to select the date for an appointment	1) Did not select any date 2) Click Book	-	An error message “Please Select Date”.	An error message “Please Select Date”.
TCB02	Make an appointment	The name cannot be empty	**1)** Enter empty name 2) Click Book	Name : -	An error message “Please enter Name”.	An error message “Please enter Name”.

TCB03	Make an appointment	The phone number cannot be empty	1) Enter empty phone number 2) Click Book	Phone Number : -	An error message “Please enter Phone number”.	An error message “Please enter Phoen number”.
TCB04	Make an appointment	User need to select atleast one service.	1) Did not select at least one service 2) Click Book	-	An error message “Please at least one service”.	An error message “Please at least one service”.
TCB05	Make an appointment	User need to select available time slot	1) Did not select any time slot that available 2) Click Book	-	An error message “Please select time”.	An error message “Please select time”.

TCB06	Make an appointmment	Fill all the information needed in the appointment	1) Select date. 2) Enter the name 3) Enter the phone number 4) Choose barber 5) Select atleast one service 6) Select one yime slot that available 7) Click Book	Date : 28/8.2024 Name : Amad Phone Number : 0194876453 Barber : Kim Service : Haircut Time : 1:00 - 2:00	Redirect to View Appointment Page	Redirect to View Appointment Page

6.4.2.4 Test Data For View Appointment Module

System	RazorReady	Version	V1
Module	View Appointment	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	24/8/2024

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result
TCV01	Search Apppointment	The phone number cannot be empty	1) Enter empty phone number 2) Click Search	Phone number : -	An error message “Please enter Phone number”.	An error message “Please enter Phone number”.
TCV02	Search Apppointment	Enter phone number	1) Enter phone number 2) Click Search	Phone number : 0194876453	Show booking history	Show booking history

6.4.2.5 Test Data For Manage User Module

System	RazorReady	Version	V1
Module	Manage User	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	24/8/2024

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result
TCU01	Delete User Account from the system	Click delete button on the user account	1) Click delete icon	-	Account have been deleted from the system	Account have been deleted from the system

6.4.2.6 Test Data For Manage Booking Module

System	RazorReady	Version	V1
Module	Manage Booking	Revision	-
Process By	Muhammad Hafiz Lukman Bin Nuri Ansarizam	Date	24/8/2024

Test ID	Test Scenario	Test Case	Test Steps	Test Data	Expected Result	Actual Result
TCM01	Delete Appointment from the system	Click delete button on the appointment list	1) Click Delete icon	-	Appointment have been deleted from the system	Appointment have been deleted from the system

6.5 Test Result and Analysis

Test result and analysis refer to the study and evaluation of the outcomes from software testing activities. After executing test cases, the results are documented, highlighting any defects or inconsistencies identified during testing. The findings are then analyzed, with the testing team assessing the severity of the issues, identifying their root causes, and prioritizing them for resolution. This process helps project stakeholders determine whether the software meets its quality and functionality standards, and whether it is ready for release or needs further refinement. Effective test result analysis is crucial for addressing software defects promptly, enhancing overall product quality, and ultimately delivering a reliable and satisfactory user experience.

Table 6.14: Test Result And Analysis For Login Module

Module/Component	Result		
Test Case ID	Description	Pass	Fail
TCL01	The username and password must be valid to login.	/	
TCL02	The email cannot be empty.	/	
TCL03	The password cannot be empty.	/	
TCL04	Verify the email and password	/	

Table 6.15: Test Result And Analysis For Register Module

Module/Component	Result		
Test Case ID	Description	Pass	Fail
TCR01	The username cannot be empty.	/	
TCR02	The email cannot be empty.	/	
TCR03	The password cannot be empty.	/	
TCR04	Verify Account	/	

Table 6.16: Test Result And Analysis For Booking Module

Module/Component	Result		
Test Case ID	Description	Pass	Fail
TCL01	User need to select the date for an appointment	/	
TCL02	The name cannot be empty	/	
TCL03	The phone number cannot be empty	/	

TCL04	User need to select atleast one service.	/	
	User need to select available time slot	/	
	Fill all the information needed in the appointment	/	

Table 6.17: Test Result And Analysis For View Appointment Module

Module/Component		Result	
Test Case ID	Description	Pass	Fail
TCL01	The phone number cannot be empty	/	
TCL02	Enter phone number	/	

Table 6.18: Test Result And Analysis For Manage User Module

Module/Component	Result		
Test Case ID	Description	Pass	Fail
TCL01	Click delete button on the user account	/	

Table 6.19 : Test Result And Analysis For Manage Appointmnet Module

Module/Component	Result		
Test Case ID	Description	Pass	Fail
TCL01	Click delete button on the appointment list	/	

6.6 Conclusion

Testing techniques are essential in both documentation and development of a system. They form the foundation of quality assurance, ensuring that the system not only functions as intended but also meets the expectations of end-users. From unit testing to user acceptance testing, thorough testing helps identify and correct errors early in the development process, preventing costly issues after release. Additionally, testing provides valuable feedback for ongoing improvements, enhances the accuracy of documentation, and ultimately contributes to the deployment of a robust and reliable system that fulfills user needs and positively impacts the organization. Comprehensive testing is a vital part of the software development lifecycle.

CHAPTER 7: PROJECT CONCLUSION

7.1 Development On Strength And Weakness

RazorReady has reached its conclusion. Throughout the development process, all performances, testing results, and related information were meticulously documented. This data was later reviewed to analyze and identify the system's strengths and weaknesses before the handover to the initial customer.

7.1.1 Strength Of RazorReady

RazorReady is an innovative mobile-based system designed to assist barbers and clients in booking appointments. Clients can schedule their appointments by selecting available time slots and choosing their preferred date from the calendar. This system streamlines the booking process, making it easy and convenient for clients to find a suitable time that aligns with both their schedule and the barber's availability.

7.1.2 Weakness of RazorReady

However, during the observation process, a significant flaw was identified: the absence of a built-in payment method in the RazorReady system. While the application excels in allowing clients to book appointments and select barbershops based on location and availability, the lack of a payment integration presents a challenge.Without an integrated payment system, clients are unable to complete their transactions directly through the app, which could lead to inconvenience and potential loss of business. Clients might be forced to pay in person or use third-party payment platforms, adding unnecessary steps to the booking process. This absence also limits the app's ability to provide seamless, end-to-end service, which is crucial for enhancing user experience and ensuring customer satisfaction.

7.2 Proposition For Improvement

One of the most significant enhancements would be the integration of a secure and seamless payment system directly within the RazorReady app. This feature would allow clients to complete transactions without leaving the platform, streamlining the entire booking process. Implementing options such as credit/debit card payments, mobile wallets, or even a "pay later" feature would provide flexibility and convenience for users. Additionally, enabling automatic payment confirmations via email or in-app notifications would enhance user experience and trust.

To further improve user engagement, consider incorporating personalized features that cater to individual preferences. This could include offering recommendations based on previous bookings, allowing clients to save their favorite barbers or services, and sending personalized notifications or offers. Leveraging user data to customize the experience can make the app feel more tailored and relevant, encouraging repeat usage.

Introducing a loyalty program that rewards clients for frequent bookings or referrals could drive engagement and attract new users. Points could be earned for every appointment booked and redeemed for discounts or free services. Additionally, a referral program that offers incentives to both the referrer and the new user could help expand the app's user base organically.

7.3 Project Contribution

The development of RazorReady involved significant contributions across various stages, beginning with the initial conceptualization and design of the app, which focused on creating an intuitive and user-friendly interface for seamless appointment booking. Key functionalities were implemented, including a dynamic calendar system for selecting available time slots and secure user authentication protocols. Rigorous testing and quality assurance were conducted to ensure the app’s performance and reliability, with comprehensive documentation maintained throughout the process. User feedback played a crucial role in iterative development, leading to continuous refinement and enhancement of the app. The final stages included a thorough handover process to the initial customer, complete with training

and technical support, ensuring a successful transition from development to deployment.

7.4 Conclusion

The development of RazorReady has been a comprehensive and meticulously executed project, resulting in an innovative mobile application that effectively streamlines the appointment booking process for barbers and clients. From the initial conceptualization to the final handover, the project was marked by a focus on user experience, security, and functionality. Key features, such as a dynamic calendar system and secure authentication protocols, were implemented to meet the needs of the target audience. Despite the app's strengths, the absence of an integrated payment system was identified as a significant flaw, highlighting an area for future enhancement. The project also emphasized the importance of thorough documentation, user feedback, and iterative improvements, all of which contributed to the app’s success. In conclusion, RazorReady stands as a robust and user-friendly platform that has the potential to make a meaningful impact in the barbershop industry, with room for further refinement and expansion.

REFERENCES

1. Joe's Barbershop – Joe's Barbershop. (2024). https://joesbarbershop.com.my/

2. The Liffey Barbers. (2020). https://theliffeybarbers.ie/?gad_source=1&gclid=CjwKCAjw5qC2BhB8EiwAvqa41vzrWXKQhQQEsNrkWvxNW4l2dBayucKUoJzeUVuP4F-wm6DLBceC9xoCe1EQAvD_BwE

3. BT Barbershop The Sphere - 8 Jalan Kerinchi Unit LG-11A No. 1 Avenue 1 Bangsar South City - Kuala Lumpur | Fresha. (2024). Fresha - Instantly Book Salons and Spas Nearby. https://www.fresha.com/a/bt-barbershop-the-sphere-kuala-lumpur-8-jalan-kerinchi-rm7h75qa

4. Chop Barbershop. (2020). CHOP Barbershop. https://www.chopbarbershop.com/

5. The Barbershop Cuts & Cocktails. (2024). The Barbershop Cuts & Cocktails. The Barbershop | Cuts and Cocktails at the Cosmopolitan in Las Vegas, NV. https://www.thebarbershoplv.com/

6. Barbershop lounge. (2018). Barbershop Lounge. https://www.barbershoplounge.com/

7. Riva Salon. (2024, August 13). Men & Woman Salon Queens NY | Hair Salon for Men & Women Queens NY | Hair Experts Queens NY. Riva Salon -. https://rivasalon.com/?gad_source=1&gclid=CjwKCAjw5qC2BhB8EiwAvqa41vKU4F6f7DQtFKg2qG0JGKueWwcy7hvXDhw2iN9v8zEeL63Vx9TPTRoCCtAQAvD_BwE

8. Rudy's Barbershop | Superior hair products and services for everybody. (2024). Rudy's Barbershop. https://rudysbarbershop.com/?srsltid=AfmBOop8gXeEh8y3UmTVOIre385ymcUL-41sr-c5dgm5d3Qj0Zq1IKFI

9. The Spot Barbershop. (2024, August 14). The Spot Barbershop ® - Classic & Contemporary Haircuts. The Spot Barbershop®. https://thespotbarbershop.com/

10. Scout's Barbershop. (2024, February 12). Nashville Men & Women Barber shops | Scout's Barbershop | Hair Salons. https://scoutsbarbershop.com/