

# My Salon

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Problem Statement . . . . .	2
<b>2</b>	<b>Feasibility Study</b>	<b>3</b>
2.1	Technical Feasibility . . . . .	3
2.2	Operational Feasibility . . . . .	3
2.3	Economical Feasibility . . . . .	4
2.4	Scheduling Feasibility: . . . . .	5
<b>3</b>	<b>Requirement Analysis and Specifications</b>	<b>6</b>
3.1	Stakeholder Identification: . . . . .	6
3.2	Functional Requirements Gathering: . . . . .	6
3.3	Non-Functional Requirements Elicitation: . . . . .	6
3.4	Specifications . . . . .	7
<b>4</b>	<b>Planning</b>	<b>9</b>
<b>5</b>	<b>Design and Implementation</b>	<b>10</b>
5.1	Use Case Diagram . . . . .	10
5.2	Dataflow Diagram . . . . .	12
5.3	Sequence Diagram . . . . .	13
<b>6</b>	<b>Implementation</b>	<b>14</b>
6.1	Technologies Uses . . . . .	14
6.2	Coding . . . . .	14
6.3	Some Screenshots of My Salon . . . . .	15
<b>7</b>	<b>Testing</b>	<b>17</b>
7.1	Unit Testing . . . . .	17
7.2	White-Box Testing . . . . .	17
7.3	Black-Box Testing . . . . .	17
7.4	Security Testing . . . . .	17
7.5	Integration Testing . . . . .	17
<b>8</b>	<b>Conclusion</b>	<b>18</b>

# **1 Introduction**

## **1.1 Problem Statement**

The XYZ Barbershop, a popular local establishment, is seeking a comprehensive software solution to streamline its operations, enhance customer experience, and reduce wait times. The current manual appointment system is causing inefficiencies, leading to longer wait times for customers and challenges in managing employee schedules and customer information.

The primary challenge faced by the XYZ Barbershop is the lack of an automated system to manage appointments, monitor real-time wait times, and maintain customer profiles and preferences. The absence of a structured software solution results in reduced operational efficiency, leading to dissatisfied customers and potential revenue loss. The barbershop aims to address these issues by developing a tailored software solution using the Waterfall model.

## 2 Feasibility Study

The feasibility study aims to assess the viability of developing a comprehensive barbershop management software solution for XYZ Barbershop using the Waterfall model. This study evaluates the technical, operational, economic, and scheduling aspects of the project to determine whether the proposed software solution aligns with the goals and resources of the barbershop.

### 2.1 Technical Feasibility

**Hardware Compatibility:** The software solution does not demand any specialized or high-end hardware. The existing mobile and devices in use at the barbershop are well-equipped to support the software's functionalities. Additionally, the software can operate on multiple devices, including computers, tablets, and smartphones, ensuring flexibility and accessibility for both staff and customers.

**Software Compatibility:** The software solution is developed using widely used programming languages and frameworks, such as Flutter for the user interface. These technologies are well-known to the development team and are compatible with various operating systems and browsers. This compatibility ensures that the software can be accessed and used seamlessly across different platforms.

**Integration Possibilities:** The software solution is designed with integration in mind. It can easily integrate with existing systems used by the barbershop, such as payment gateways for online payments and customer relationship management (CRM) systems for managing customer information. The flexibility to integrate enhances the software's utility and enables a streamlined workflow without disrupting established processes.

**Scalability Considerations:** The software solution is designed to accommodate future growth. The underlying architecture is modular, allowing for the addition of new features and functionalities as the barbershop's needs evolve. The system can be scaled to handle a higher volume of appointments and customer data without compromising performance or user experience.

**Security Measures:** The software solution places a strong emphasis on data security. Different encryption protocols will be implemented to safeguard customer information, appointment details, and payment data. Regular security audits and updates will ensure that the software remains resilient against emerging threats.

### 2.2 Operational Feasibility

**User Adoption:** Operational feasibility assesses the readiness of XYZ Barbershop's staff and stakeholders to adopt the new barbershop management software solution. The transition from manual processes to the software-driven approach involves evaluating the staff's familiarity with technology and their willingness to embrace change.

**Technology Familiarity:** The barbers and staff at XYZ Barbershop are already accustomed to using technology in their daily operations. They utilize digital tools for appointment

scheduling, communication, and online marketing. This familiarity with technology suggests a high likelihood of smooth user adoption of the new software solution.

**User Training:** To ensure successful user adoption, a comprehensive training plan will be implemented. The development team will provide detailed user documentation and conduct training sessions tailored to the needs of the barbers and staff. This approach aims to build confidence and proficiency in using the software, making the transition seamless.

**Change Management:** Change management strategies will be employed to address any potential resistance to adopting the software. Clear communication about the benefits of the new solution, opportunities for feedback, and active involvement of staff in the implementation process will foster a positive attitude toward change.

**Efficiency Enhancement:** The software solution's user-friendly interface and streamlined functionalities are designed to enhance operational efficiency. The staff will benefit from automated appointment scheduling, reduced administrative tasks, and real-time information availability. This efficiency improvement directly aligns with the operational goals of XYZ Barbershop.

**Customer Experience:** Operational feasibility also takes into account the impact of the software solution on customer experience. The ability to book appointments online, receive reminders, and access real-time wait time information enhances customer satisfaction. A positive customer experience contributes to repeat business and positive word-of-mouth referrals.

### 2.3 Economical Feasibility

**Cost Analysis:** The economic feasibility of developing the barbershop management software solution involves a comprehensive cost analysis that includes both initial development expenses and ongoing maintenance costs. The estimated costs are carefully evaluated in comparison to the projected benefits to determine the software's financial viability.

**Development Costs:** The primary costs associated with software development include personnel expenses, software tools and licenses, and potential outsourcing costs. A qualified in-house development team is available, ensuring that personnel expenses are within the budget. Necessary software tools and licenses are already accessible, minimizing additional expenses. The decision to develop in-house reduces outsourcing costs and enhances cost control.

**Deployment Costs:** Deployment costs include server hosting and potential user training expenses. The barbershop has existing server resources that can be utilized for hosting the software, resulting in minimal hosting costs. User training costs are manageable, as the development team is equipped to provide comprehensive training materials and sessions.

**Return on Investment (ROI):** The economic feasibility assessment emphasizes the software's potential for a positive ROI. By optimizing operational processes and enhancing customer experience, the software is projected to contribute to increased revenue and customer retention. The development and deployment costs are expected to be recouped through improved efficiency and enhanced business performance over time.

**Cost-Benefit Analysis:** A thorough cost-benefit analysis considers the initial investment against the anticipated long-term benefits. The costs associated with software development, deployment, and ongoing maintenance are weighed against the projected revenue increase, operational savings, and improved customer satisfaction. The analysis indicates a favorable ratio of benefits to costs, reinforcing the economic feasibility of the software solution.

## 2.4 Scheduling Feasibility:

**Timeframe and Milestones:** Scheduling feasibility evaluates the timeline for developing and deploying the barbershop management software solution using the Waterfall model. The structured approach of the Waterfall model allows for a well-defined schedule with clear milestones for each phase of the project.

**Development Phases:** The Waterfall model divides the project into distinct phases, such as requirements gathering, system design, implementation, testing, deployment, and validation. Each phase has a predetermined timeframe, ensuring a systematic progression.

**Resource Availability:** The availability of resources, including the development team's expertise and the barbershop's infrastructure, contributes to the scheduling feasibility.

**Development Team:** A skilled in-house development team is available to dedicate their expertise to the project. This resource availability ensures that development progresses efficiently and is not hindered by outsourcing delays.

**Infrastructure:** The barbershop's existing infrastructure, including hardware and software resources, is readily available for software deployment. This eliminates the need for significant setup time or resource procurement.

**Risks and Contingency:** While the Waterfall model's structured approach minimizes uncertainties, potential risks such as unexpected technical challenges or scope adjustments are addressed through contingency plans.

### 3 Requirement Analysis and Specifications

Requirement analysis is a crucial phase that involves identifying, documenting, and refining the functional and non-functional requirements of the barbershop management software solution. This phase lays the foundation for the software's design and development by ensuring that all stakeholder needs and expectations are captured accurately.

#### 3.1 Stakeholder Identification:

Identify the primary stakeholders involved in the software project. These stakeholders include barbers, customers, and the development team. Each group's input is valuable in understanding their specific needs and expectations from the software solution.

#### 3.2 Functional Requirements Gathering:

Engage in discussions, interviews, and workshops with stakeholders to identify functional requirements. These requirements define the software's core features and capabilities. Key functional requirements include:

- **Giving Serial:** Customers should be able to schedule appointments online, selecting services and preferred barbers.
- **Wait Time Tracking:** Real-time wait times should be displayed on screens in the barbershop and accessible through a customer app.
- **Remove Serial:** The user and Barber both can remove the serial.
- **Open/Close Shop:** Barber opens his Shop at any time but closes when finishes the serial.
- **Search Shop:** Users search shop in location base or specific shop name base. And he can see details.
- **Payment:** There is an advance payment system. Users can pay in advance.
- **Set Serial Limit:** Barber can set the limitation of the serial as he wishes. People can't request over the limit.
- **User Profiles:** The software should store customer information, appointment history, and preferences for tailored services.
- **Notifications:** Automated reminders should be sent to customers before their appointments.

#### 3.3 Non-Functional Requirements Elicitation:

Non-functional requirements capture the qualities that the software should possess. Engage with stakeholders to identify these requirements, which encompass performance, security, usability, and more. Non-functional requirements include:

- **Usability:** The software should have an intuitive interface for both staff and customers.
- **Performance:** The software should load quickly and respond promptly to user interactions, even during peak usage.

- **Security:** Customer data and payment information should be encrypted and protected from unauthorized access.
- **Scalability:** The software should handle increased usage and data volume as the barbershop grows.

### 3.4 Specifications

The specifications phase translates the identified requirements from the requirement analysis into a detailed technical document that serves as a blueprint for the development of the barbershop management software solution. This phase outlines how each feature will be implemented, the design of user interfaces, the database schema, and integration points with existing systems.

**User Interface Design:** The specifications document outlines the design of the software's user interface (UI). This includes detailed mockups, wireframes, and visual representations of each screen and interaction. The UI design ensures that the software is visually appealing, intuitive to use, and aligned with the branding of XYZ Barbershop.

**Layout and Components:** Define the layout of each screen, including the arrangement of buttons, forms, and navigation elements. Detail the components that will be present on each screen and their functionalities.

**Color Scheme and Visual Elements:** Specify the color palette, typography, icons, and other visual elements that will be used throughout the software's user interface.

**Interaction Flow:** Describe how users will navigate through the software, including the sequence of screens and the actions they can take on each screen.

**Database Schema:** The specifications document outlines the structure of the software's database, which stores customer information, appointment records, employee details, and more. The database schema defines the relationships between different data entities and ensures efficient data management.

**Tables and Fields:** Define the tables that will be present in the database and the fields within each table. For example, the "Customers" table might include fields like name, contact information, and preferences.

**Relationships:** Specify how different tables are related to each other. For instance, the relationship between customers and their appointment history.

**Integration Points:** The specifications document details how the software solution will integrate with existing systems used by XYZ Barbershop. Integration ensures that data flows seamlessly between the software and other tools.

**API Integration:** If integrating with third-party systems, specify the APIs (Application Programming Interfaces) that will be used for data exchange.

**Data Mapping:** Describe how data will be mapped and synchronized between the software and existing systems.

**Error Handling and Validation:** The specifications phase also addresses error handling and data validation. Detail how the software will handle errors, such as incorrect user inputs,

and how users will be informed of errors.

**Input Validation:** Specify the rules and checks that will be in place to ensure that users provide valid and accurate data.

**Error Messages:** Describe the error messages that will be displayed to users when errors occur, along with suggestions for corrective actions.

## 4 Planning

The planning indicates what needs to be done, which resources must be utilized, and when the project is due. In short, it's a timetable that outlines start and end dates and milestones that must be met for the project to be completed on time. Project planning encompasses the following actions:

The task list and required time for this project are given below:

- Task 1: Split project into tasks and estimate time and resources required to complete each task.
  - Estimated Time: TBD
- Task 2: Organize tasks concurrently for optimal workforce utilization.
  - Estimated Time: TBD
- Task 3: Depending on project manager's skill and experience.
  - Estimated Time: TBD

<b>Task Name</b>	<b>Start Date</b>	<b>End Date</b>
<b>i) Analysis</b>	<b>01-07-2023</b>	<b>30-09-2023</b>
- Requirement Elicitation	01-07-2023	15-09-2023
- Feasibility Study	15-07-2023	15-08-2023
- System Planning and Scheduling	15-08-2023	30-09-2023
<b>ii) Front End</b>	<b>01-10-2023</b>	<b>15-11-2023</b>
- User Interface Design	01-10-2023	15-10-2023
- Front End Development	15-10-2023	30-10-2023
- UI Testing	01-11-2023	15-11-2023
<b>iii) Back End</b>	<b>01-12-2023</b>	<b>15-01-2024</b>
- Database Design	01-12-2023	15-12-2023
- Server-side Logic	15-12-2023	01-01-2024
- Integration Testing	01-01-2024	15-01-2024

## 5 Design and Implementation

### 5.1 Use Case Diagram

A use case diagram in software engineering and systems analysis depicts the interactions between a system and its external actors (users or other systems) based on various use cases. It provides a high-level view of the system's functionalities and how users or external systems interact with it. Here's a short description of the main components:

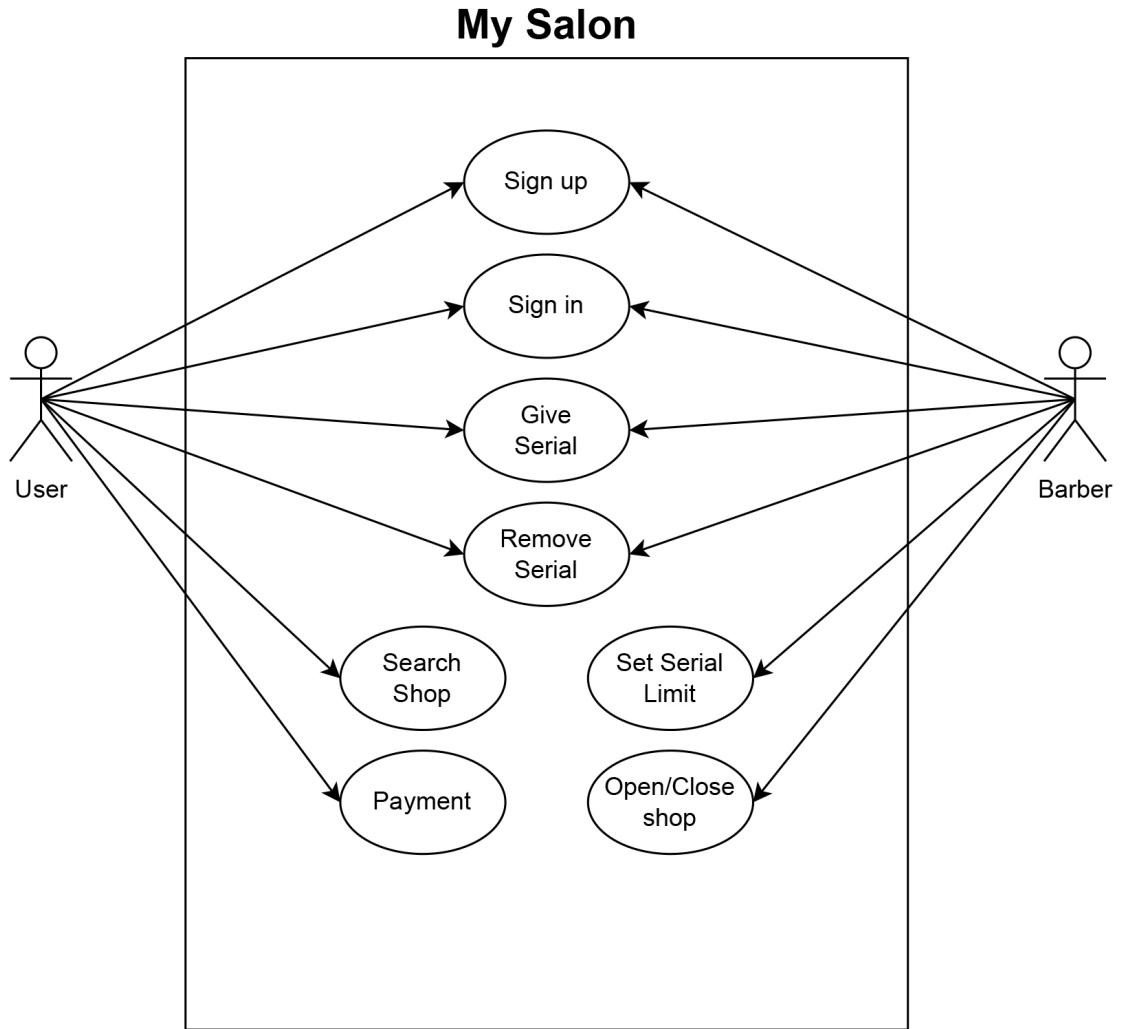


Figure 1: Use Case Diagram

#### Actor:

The actors of the system are the Barber and the User.

#### Use Case:

**Sign-Up:** This use case denotes a set of actions required for the user and Barber to register with the applications.

**Sign-In:** This use case denotes a set of actions required for the user and Barber to log in to the applications.

**Give Serial:** User request to serial to a specific shop.

**Remove Serial:** The user and Barber both can remove the serial.

**Search Shop:** Users search shop in location base or specific shop name base. And he can see details.

**Payment:** There is an advance payment system. Users can pay in advance.

**Set Serial Limit:** Barber can set the limitation of the serial as he wishes. People can't request over the limit.

**Open/Close Shop:** Barber opens his Shop at any time but closes when finishes the serial.

## 5.2 Dataflow Diagram

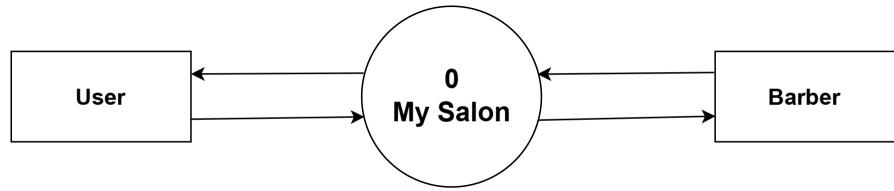


Figure 2: Context Level DFD

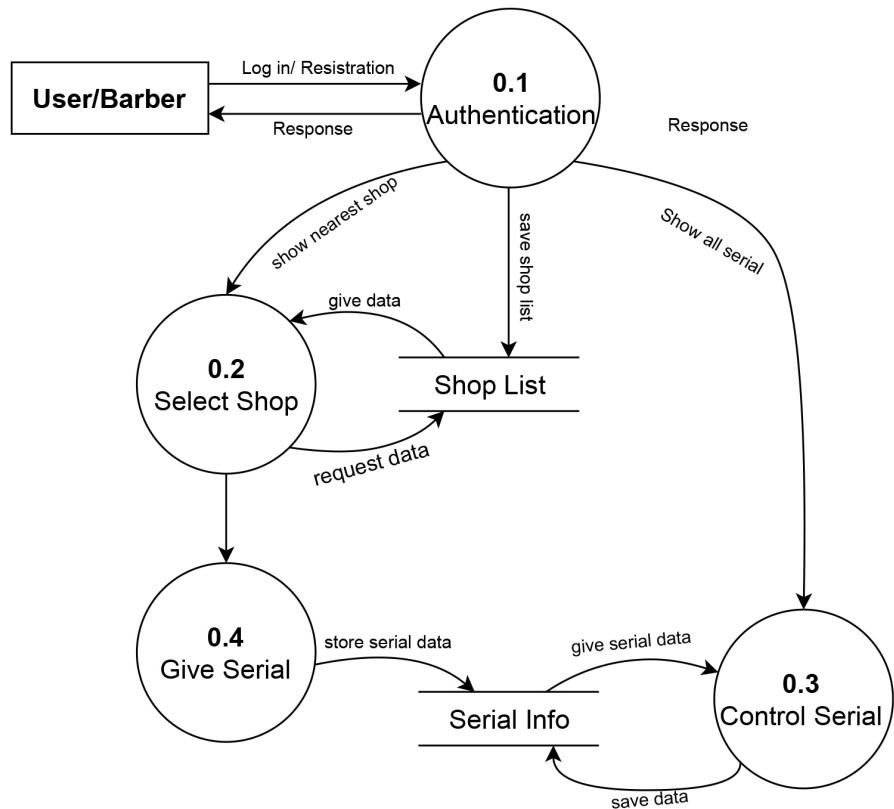


Figure 3: Level 1 DFD

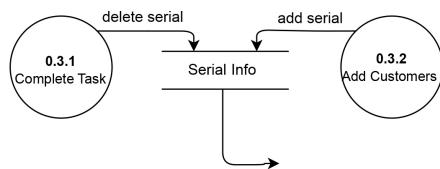
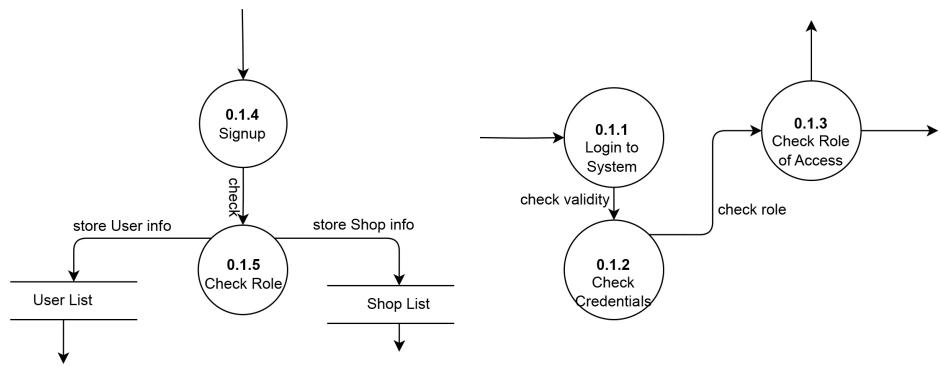


Figure 4: Level 2 DFD

### 5.3 Sequence Diagram

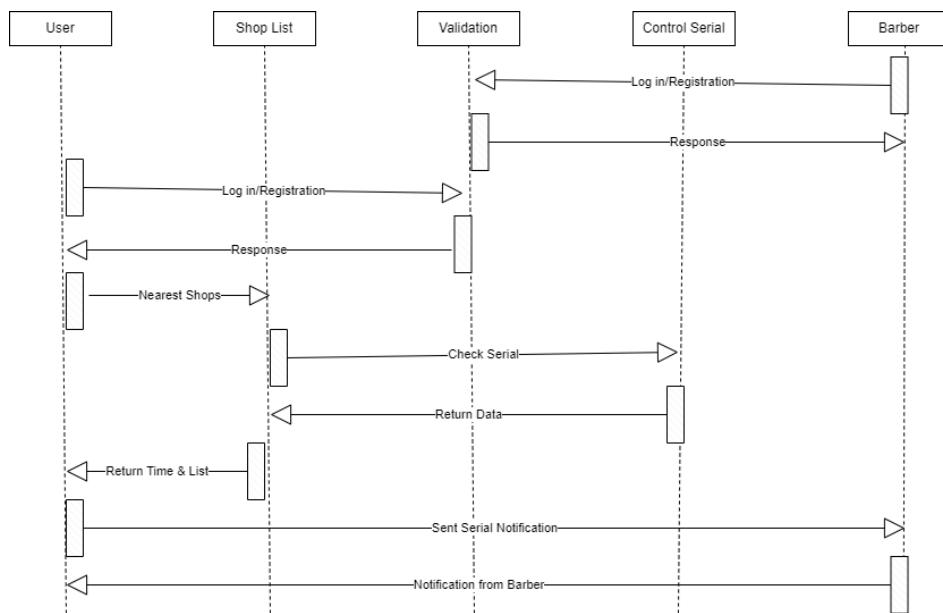


Figure 5: Sequence Diagram

## 6 Implementation

### 6.1 Technologies Uses

**Client-side:** We utilized Flutter for the client-side development of My Barber app. Flutter provides a comprehensive framework for building natively compiled applications for mobile, web, and desktop from a single codebase. With its rich set of widgets and expressive UI, Flutter ensures a visually appealing and consistent experience across various platforms.

**Database:** Here we use the Firebase database. Firebase, by Google, is a streamlined development platform offering real-time databases, authentication, cloud storage, and serverless functions. With simple deployment, analytics, and machine learning integration, it accelerates app development, making it a preferred choice for building scalable and feature-rich applications.

### 6.2 Coding

**Code Development:** After selecting the technologies, we initiated the implementation by translating the software design into executable code using Flutter for the front end. The iterative nature of the waterfall model allowed us to focus on specific modules in each iteration, ensuring a systematic and manageable development process.

**Code Review:** During the implementation, we conducted code reviews through a combination of code walkthroughs and inspections. This collaborative approach ensured adherence to coding standards, identified potential issues and iteratively maintained the codebase's quality and maintainability.

**Version control:** Version control systems, such as Git, played a crucial role in managing code versions and tracking changes. Regular commits, branching, and tagging allowed us to maintain a clear history of the project, facilitating collaboration among team members.

### 6.3 Some Screenshots of My Salon

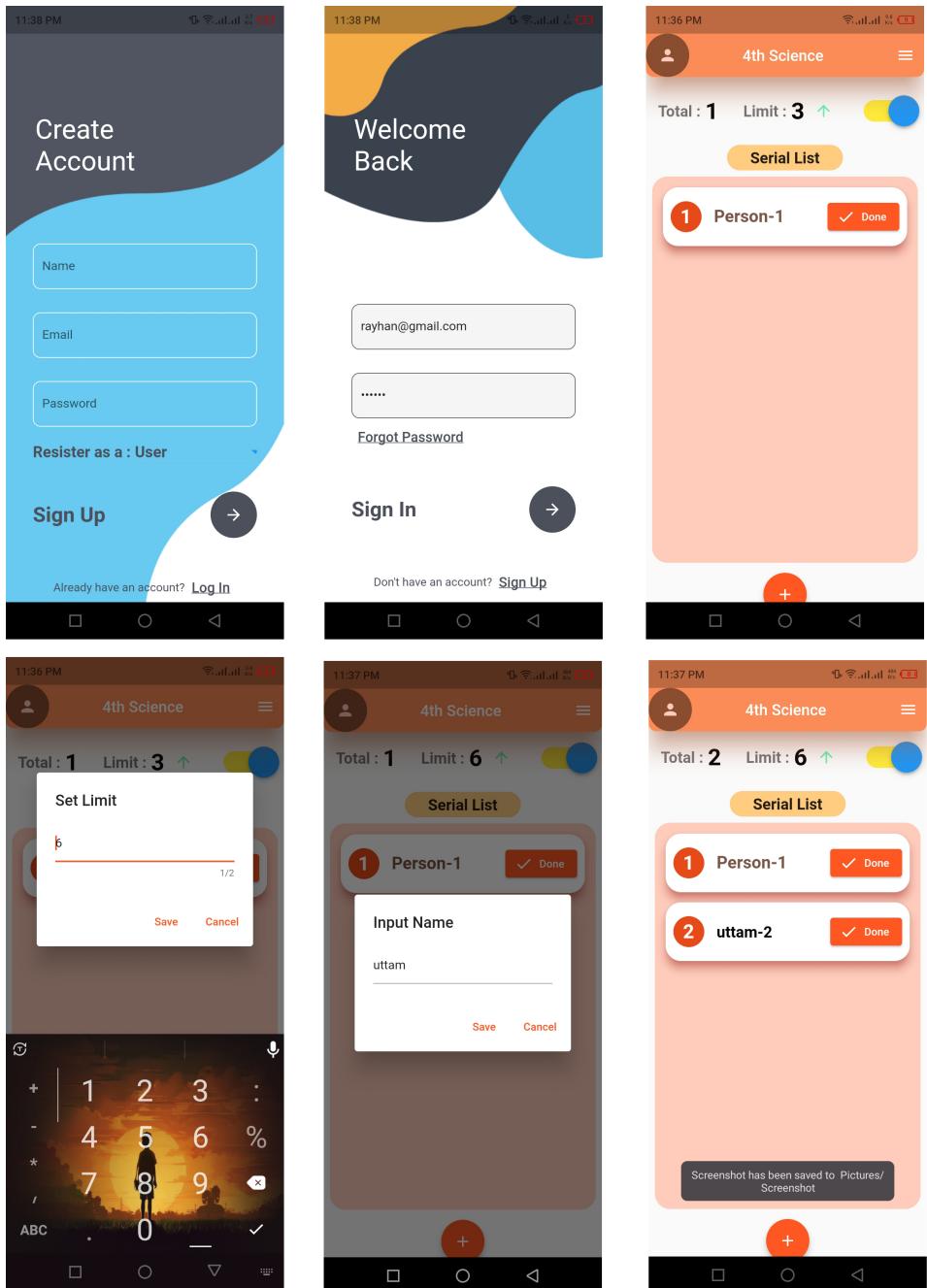


Figure 6: Barber Interface

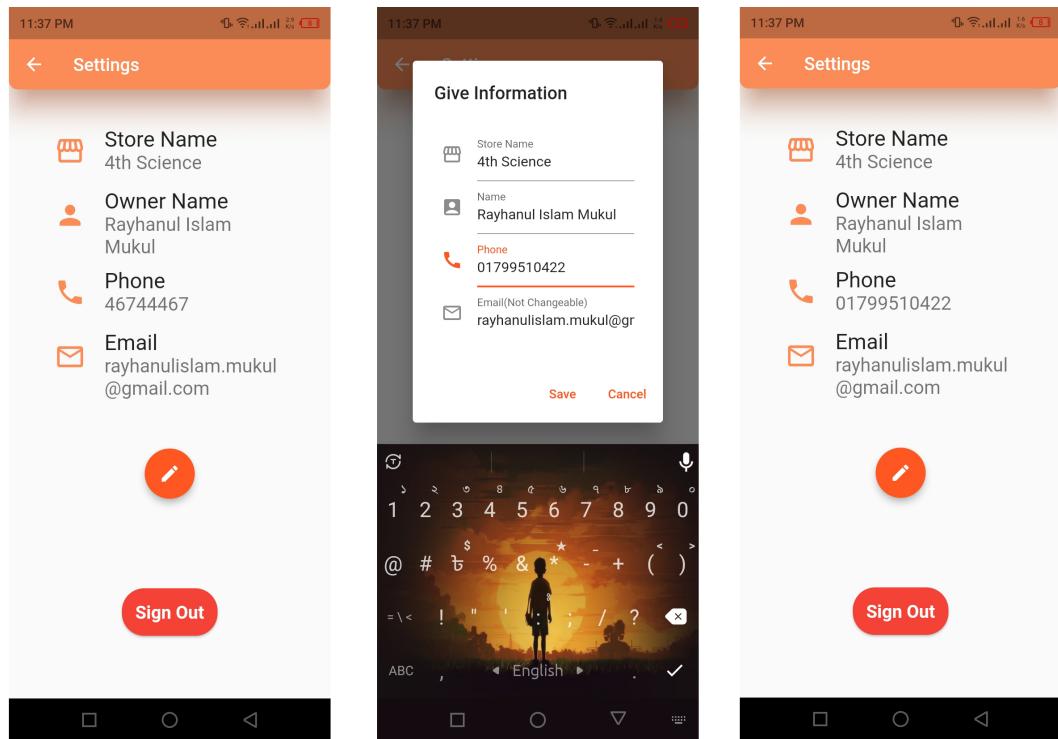


Figure 7: Barber Interface

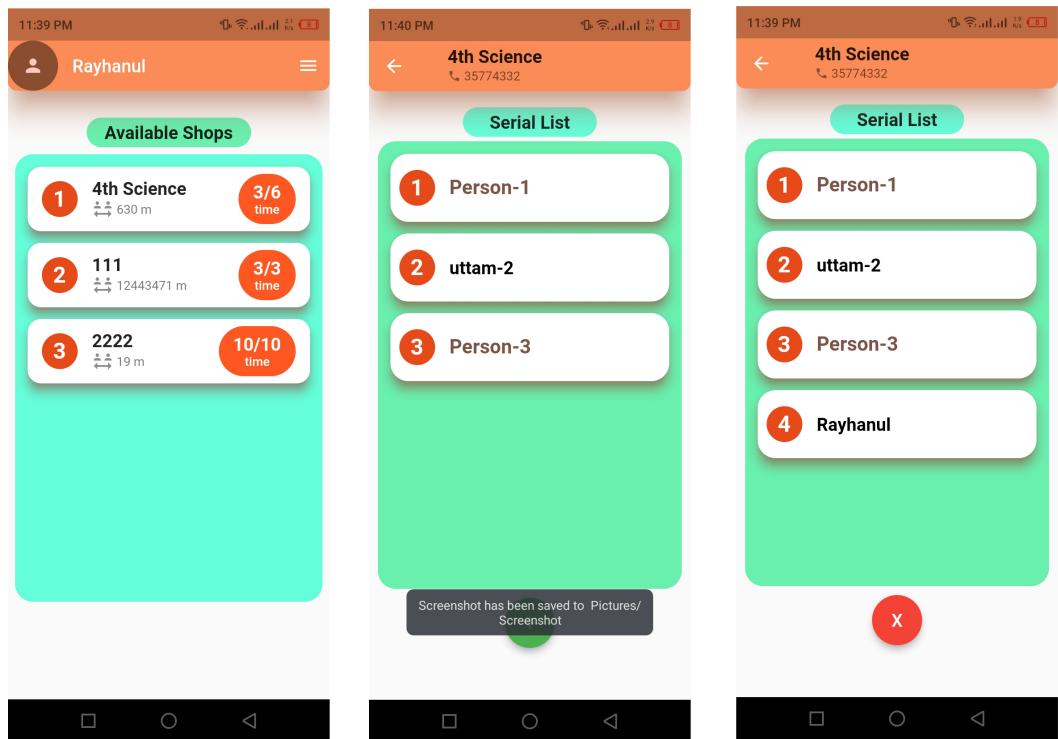


Figure 8: User Interface

## 7 Testing

### 7.1 Unit Testing

We tested individually all functional and it worked perfectly.

### 7.2 White-Box Testing

- Statement Coverage Testing: We performed statement coverage testing to assess the extent to which our code was executed during testing. By systematically examining each line of code, we aimed to identify any unexecuted statements and improve overall code reliability.
- Branch Coverage Testing: Our white-box testing also included branch coverage analysis to ensure that all possible decision outcomes within the code were exercised. This helped us uncover any unexplored branches and verify the correctness of our control flow structures.
- Multiple Condition Coverage Testing: Multiple condition coverage testing was performed to scrutinize complex decision structures involving multiple conditions. By testing all possible combinations, we ensured that the app's logic remained robust and reliable.

### 7.3 Black-Box Testing

During the black-box testing phase, we employed boundary value analysis to examine the app's behavior at the edges of its input ranges. This approach allowed us to identify potential issues related to minimum and maximum values, ensuring the robustness of the app's input handling.

### 7.4 Security Testing

Security testing is a crucial aspect of software testing that focuses on identifying vulnerabilities and weaknesses in a system to ensure it can withstand malicious attacks and unauthorized access. The objective is to protect the system from potential threats, safeguard sensitive data, and ensure compliance with security standards.

### 7.5 Integration Testing

The integration testing phase focused on verifying the proper functioning of different components when integrated. This ensured seamless communication between various modules, preventing any unexpected issues that might arise during interaction.

## **8 Conclusion**

In conclusion, the development of our My Salon app marked a significant achievement, showcasing a robust foundation with essential functionalities. The implemented features including the available shops, real-time suggestions, Barber's profile, user's profile, and the booking system, are the core functionalities that are essential for the app to be successful. The app is designed to be user-friendly and intuitive, with a simple and clean interface.

We can't implement many things in this short time. We will bring the updated version of the software and make updates from time to time