American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Trimmer

A Software Quality and Testing Project Submitted
By

Semester: Summer_22_23			**Section:**	**Group No:**
SN	**Student Name**	**Student ID**	Individual Contribution (in %)	Total Marks: 50 Earned Marks:
1.	MEHEDI HASAN RABBI	20-44059-2	100%	

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	**Total Marks (50)**	
Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions	[10 Marks]	
Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements)	[10 Marks]	
Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases	[10 Marks]	
Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk	[10 Marks]	
Approval, Format, Submission, and Defense	[10 Marks]	

Software Test Plan

for

<Trimmer>

Version 1.4 approved

Prepared by <

MEHEDI HASAN RABBI

>

<American International University Bangladesh>

<14 January 2023>

Table of Contents

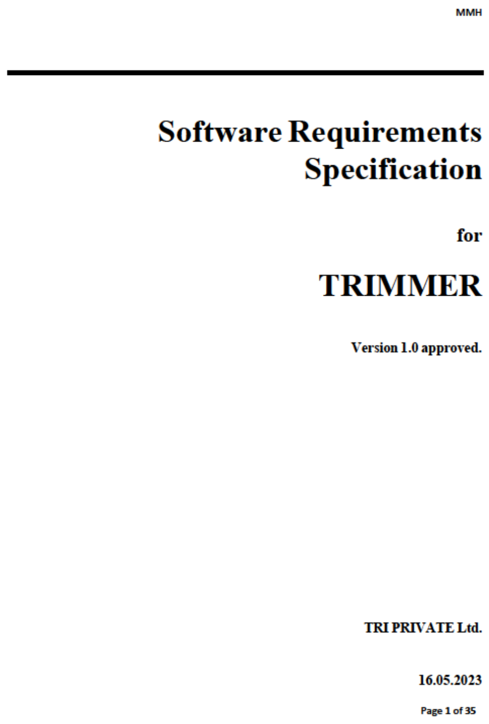
Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.03.06	MEHEDI HASAN RABBI	First Draft
0.2	2023.04.07	MEHEDI HASAN RABBI	Test Case Expansion
0.3	2023.06.25	MEHEDI HASAN RABBI	Improved Risk Analysis
0.4	2023.07.22	MEHEDI HASAN RABBI	Updated Requirement Specification

1. TEST PLAN IDENTIFIER: TP-TRM01.4

2. REFERENCE MATERIALS

Here the Software Requirement Specification of the project is included.



MMH

Software Requirements Specification

for

TRIMMER

Version 1.0 approved.

TRI PRIVATE Ltd.

16.05.2023

Page 1 of 35

3. INTRODUCTION

3.1 Background to the Problem

In today's fast-paced world, personal grooming and appearance play a significant role in people's lives. Barbershops have traditionally been the go-to places for men's grooming needs, offering services like haircuts, shaving, and styling. However, the process of finding a suitable barber and booking appointments can often be time-consuming and inconvenient. People may struggle to identify barbers who align with their preferences in terms of location, pricing, and quality of service. Additionally, barbers face challenges in efficiently managing their schedules and attracting new clients.

The lack of a streamlined and accessible platform connecting clients with barbers has created a disconnect between the demand for grooming services and the means to fulfill that demand efficiently. This disconnect can lead to frustration for both clients and barbers, resulting in missed business opportunities and a suboptimal grooming experience.

The problem of disconnected and cumbersome barber-client interactions is significant for several reasons:

Inefficiency: The absence of a streamlined platform leads to wasted time and effort for both clients and barbers. Clients may spend excessive time searching for suitable barbers, and barbers may have gaps in their schedules due to inefficient booking processes.

Limited Visibility: Talented barbers often find it challenging to reach potential clients beyond their immediate vicinity. This limits their business growth potential and prevents clients from discovering high-quality services.

Customer Experience: Clients' experiences are marred by uncertainty, lack of information, and difficulty in securing appointments. This can deter potential clients from seeking barber services altogether.

Quality Assurance: Clients rely on word-of-mouth or unreliable online reviews to choose barbers. The lack of a trustworthy rating and feedback system may lead to subpar experiences.

Barber Empowerment: Barbers, especially independent or smaller businesses, need a platform that enables them to manage appointments, showcase their skills, and expand their reach without hefty marketing expenses.

Industry Evolution: With the service industry becoming more digitized, clients expect convenient online booking and information access. The traditional methods employed by many barbershops are falling behind these expectations.

Data Security: Handling personal information securely is crucial in today's digital landscape, and an efficient software platform can provide the necessary safeguards.

By addressing these challenges through a comprehensive and user-friendly platform, the proposed software solution aims to revolutionize the barber-client interaction process. It seeks to empower barbers, enhance the client experience, and bring the industry up to par with modern digital standards.

3.2 Solution to the Problem

The proposed solution is a comprehensive and user-friendly software platform that connects barbers and clients, simplifying the process of finding, booking, and managing barber services. This solution involves creating a centralized platform that addresses the challenges faced by both clients and barbers, offering a range of features to enhance the experience for both parties.

Key Features and Benefits:

Search and Discovery: Clients can easily search for barbers based on various criteria such as ratings, location, and pricing. This empowers clients to make informed decisions based on their preferences.

Appointment Booking: The platform provides clients with real-time access to barbers' schedules, enabling them to book appointments conveniently with just a few clicks.

Rating System: Authentic customer feedback and ratings allow clients to select barbers with confidence, ensuring a level of quality assurance.

Comprehensive Services: The software covers a wide range of barber services, including haircuts, shaving, and grooming. This all-inclusive approach caters to diverse client needs.

Barber Profiles: Barbers can create detailed profiles showcasing their skills, expertise, and portfolio. This helps them stand out and attract a broader clientele.

Efficient Management: Barbers can manage their appointments, availability, and services in one centralized location, streamlining their business operations.

Data Security: Robust security measures ensure the safety of personal information, fostering trust between clients and barbers.

Feasibility:

The proposed solution is highly feasible and aligned with the business objective of simplifying barber-client interactions. The availability of modern technology, cloud infrastructure, and digital payment systems makes it possible to create and deploy such a platform. Additionally, the widespread use of smartphones and internet connectivity ensures that both clients and barbers can easily access and use the platform.

Software Description and Purpose:

The software platform is designed to serve as a bridge between clients seeking barber services and barbers looking to expand their clientele. Its purpose is to revolutionize the way barber services are discovered, booked, and managed, providing convenience, transparency, and efficiency. The software's objectives include enhancing the customer experience, empowering barbers, increasing visibility for both parties, and ultimately becoming the go-to platform for anyone seeking top-notch barber services.

Existing Studies and Solutions:

While there might be some individual websites or apps that offer limited appointment booking for barbers, a comprehensive and centralized solution that caters to a wide range of barber services, includes detailed barber profiles, incorporates a robust rating system, and focuses on both customer and barber needs is relatively scarce.

Some existing software solutions include:

Traditional Booking Methods: Many barbers still rely on traditional walk-ins and phone calls for appointments. This approach lacks the convenience and transparency that modern clients expect.

Basic Booking Apps: Some booking apps provide basic scheduling features, but they often lack comprehensive profiles, in-depth service offerings, and a holistic approach to barber services.

Niche Platforms: There might be platforms dedicated to specific barber services, but they may not cover the entire spectrum of grooming services, limiting their appeal and usability.

The proposed software solution aims to fill these gaps by offering a comprehensive, user-friendly, and secure platform that benefits both clients and barbers across a wide range of services.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. **User Profile**

Functional Requirements

1.1 After login into the user id can update his /her name, age, gender and location.
1.2 User can set Phone number or email in case some kind of network issue.
1.3 System will verify if the Phone number or Email is correct or not and send a message if invalid Phone number or Email.

Priority level: Medium
Precondition: User must register with Phone Number or Email.
Cross-reference: N/A

2. **Search Barber:**

Functional Requirements

2.1 This System will automatically provide user nearby barber shop.
2.2 This system will allow customer to choose different kind of barber.

Priority level: High.
Precondition: User need to be a registered member first.
Cross-reference: 1

3. **Booking Barber**

Functional Requirements

3.1 User can select their preferable time and barber shop.
3.2 When user not select specific barber system will automatically choose any barber on selected shop.
3.3 System will send a massage to both customer and barber for confirmed the service.

Priority level: High

Precondition: Customer need to be registered with the service location and valid phone number

Cross-reference: 2

4. **Barber Schedule:**

Functional Requirements

4.1 Barber can set their time schedule.
4.2 System automatically update barber time schedule by selected time.
4.3 Barber can set offers
4.4 System will send a massage to customer for offers.

Priority level: High
Precondition: Barber need to be registered with the service location and valid phone number
Cross-reference: N/A

5. **Cancel Request:**

Functional Requirements

5.1 User can cancel his booking at any time then system will automatically update barber time schedule.
5.2 If customer didn't go to barber shop on time and didn't cancel his booking, then system will automatically cancel the booking and make a red mark on that user.

Priority level: High
Precondition: User need to be booked a barber
Cross-reference: 3,4

4.2 System Quality Attributes

Here we give some important primary quality attributes from user perspective.

QA 1. Availability: The system shall be at least 98.5 percent available on every seven days a week between 8.00 am to 8.00 pm at local time.

Priority level: High
Precondition: Must have maintainability attribute
Cross-reference: N/A

QA 2. Efficiency: There are at least (2) percent of the processor capacity, disk space 1.7 MB/S, memory 88 MB and communication bandwidth 1024kbps shall be available to properly run this system.

Priority level: High
Precondition: N/A
Cross-reference: N/A

QA 3. Flexibility: A maintenance programmer who will be able to add new feature and function including code, modifications and testing into the system with no more than two hours.

Priority level: Medium
Precondition: N/A
Cross-reference: N/A

QA 4. Integrity: When user try to login into the system, then they have to two step verification. One step is when user try to login into the system, the system will send a verification code to the user via mail and user get a verification code to login and the second step is user need to use own password when they create the password to sign up this system.

Priority level: High
Precondition: N/A
Cross-reference: QA-2

QA 5. Reliability: The system shall no more than three experimental runs out of 700 can be lost.

Priority level: High
Precondition: N/A
Cross-reference: QA-1, QA-2, QA-3

QA 6. Maintainability: Suppose there is a problem arise in the system that user can't booked a barber. A maintenance programmer who has experience can solve this problem within 2 hours without any extra helping hand.

Priority level: High
Precondition: N/A
Cross-reference: QA-3

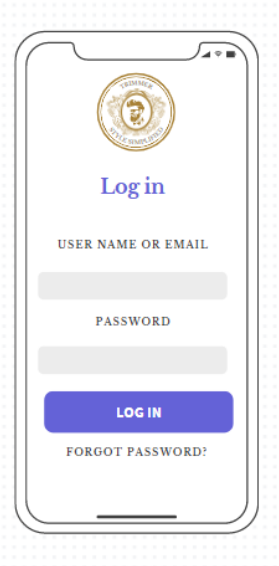
QA 7. Portability: The system was able to run any platform or any operating system. Like Windows, Android, Apple.

Priority level: High
Precondition: N/A
Cross-reference: QA-2.

QA 8. Reusability: The system functions app is free for everyone.

Priority level: Low
Precondition: N/A
Cross-reference: QA-1, QA-2, QA-3.

4.3 System Interface



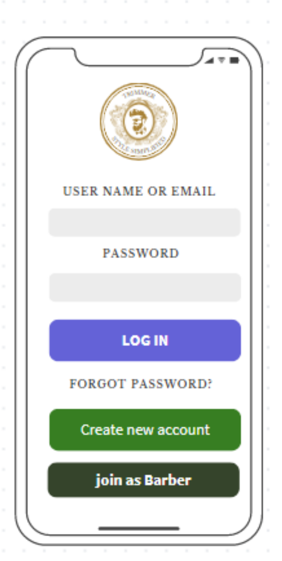
Log in

USER NAME OR EMAIL

PASSWORD

LOG IN

FORGOT PASSWORD?



USER NAME OR EMAIL

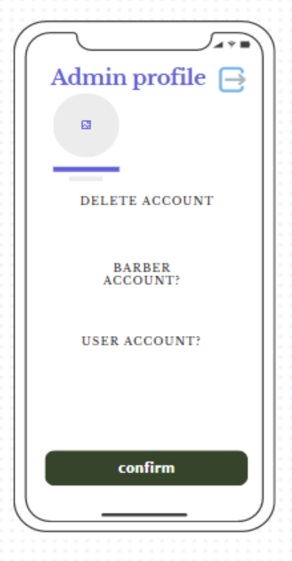
PASSWORD

LOG IN

FORGOT PASSWORD?

Create new account

join as Barber



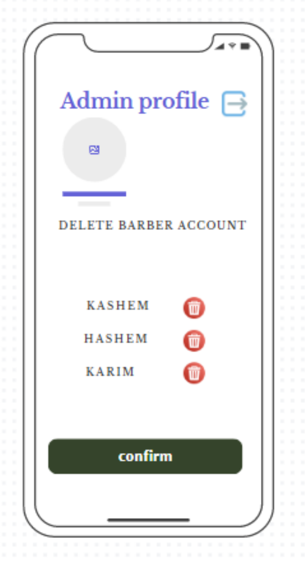
Admin profile

DELETE ACCOUNT

BARBER ACCOUNT?

USER ACCOUNT?

confirm



Admin profile

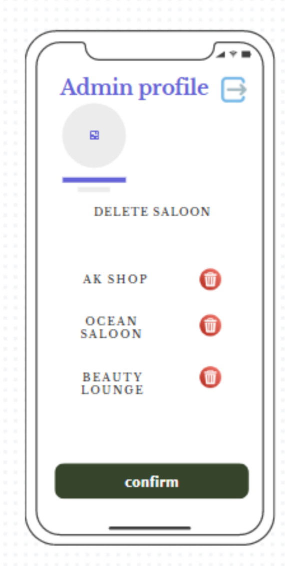
DELETE BARBER ACCOUNT

KASHEM

HASHEM

KARIM

confirm



Admin profile

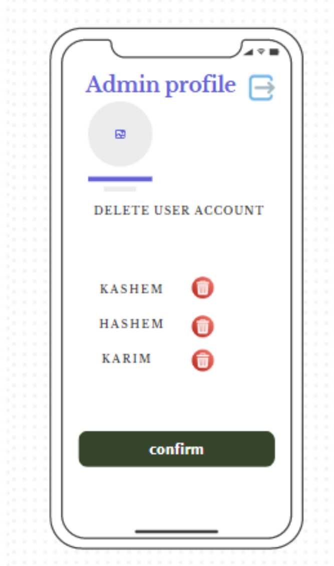
DELETE SALOON

AK SHOP

OCEAN SALOON

BEAUTY LOUNGE

confirm



Admin profile

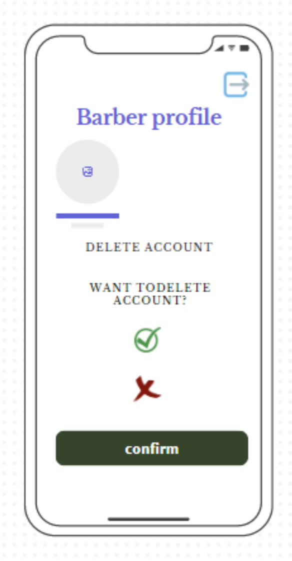
DELETE USER ACCOUNT

KASHEM

HASHEM

KARIM

confirm



Barber profile

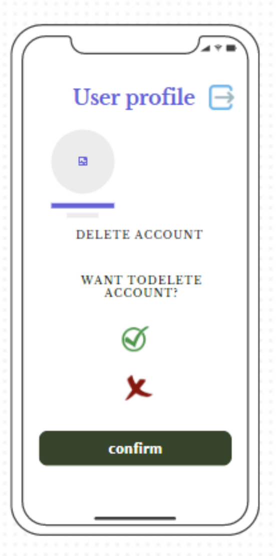
DELETE ACCOUNT

WANT TODELETE ACCOUNT?

✓

✗

confirm



User profile

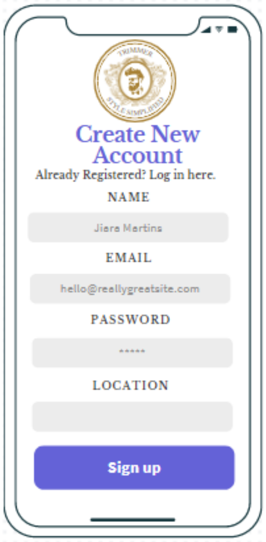
DELETE ACCOUNT

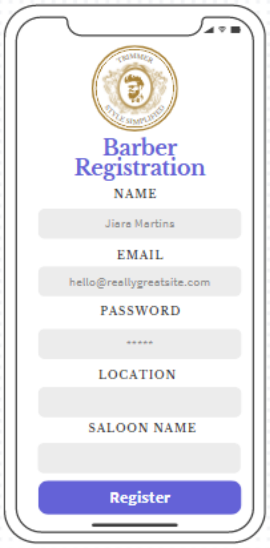
WANT TODELETE ACCOUNT?

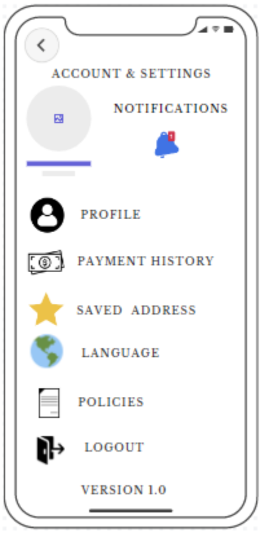
✓

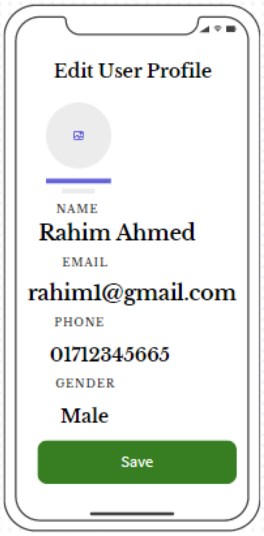
✗

confirm

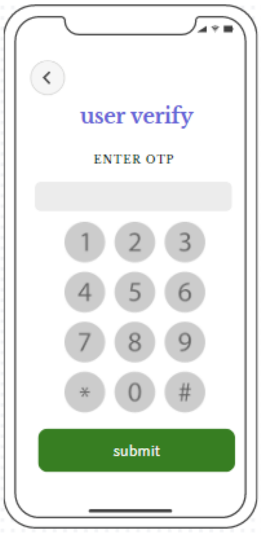


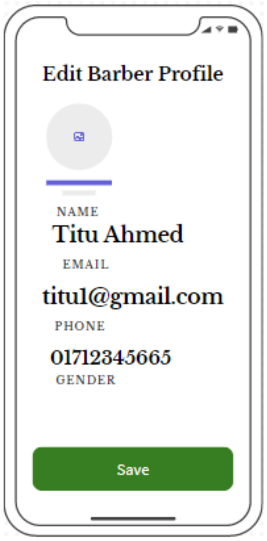




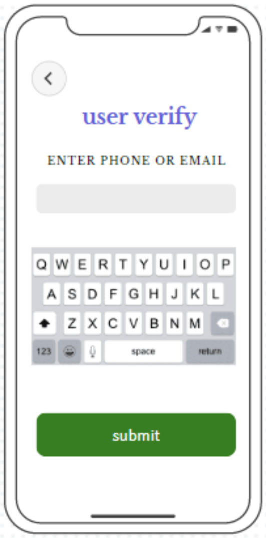


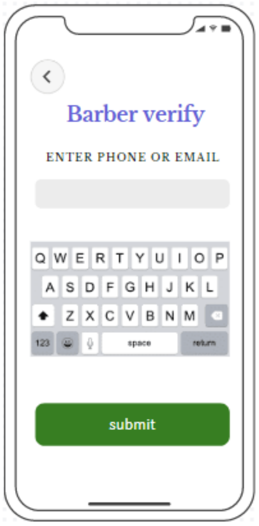
NAME
Rahim Ahmed
EMAIL
rahiml@gmail.com
PHONE
01712345665
GENDER
Male



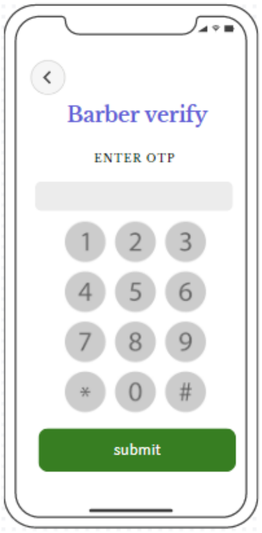


NAME
Titu Ahmed
EMAIL
titul@gmail.com
PHONE
01712345665
GENDER



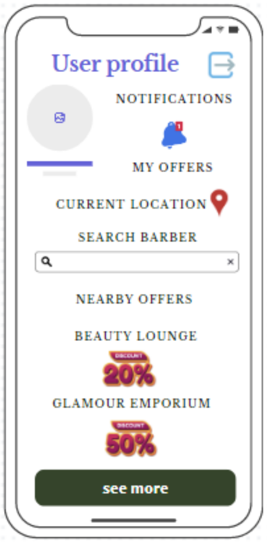


\



Barber verify

ENTER OTP



User profile

NOTIFICATIONS

MY OFFERS

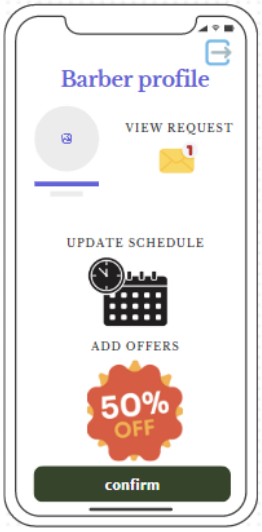
CURRENT LOCATION

SEARCH BARBER

NEARBY OFFERS

BEAUTY LOUNGE

GLAMOUR EMPORIUM

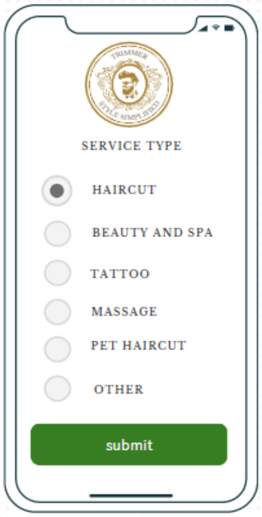


Barber profile

VIEW REQUEST

UPDATE SCHEDULE

ADD OFFERS



SERVICE TYPE

● HAIRCUT

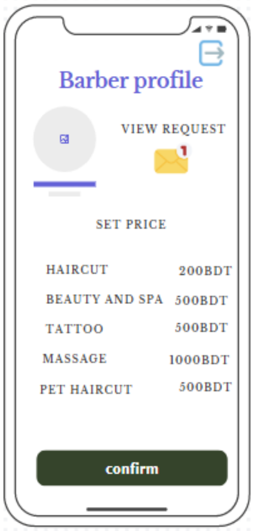
○ BEAUTY AND SPA

○ TATTOO

○ MASSAGE

○ PET HAIRCUT

○ OTHER

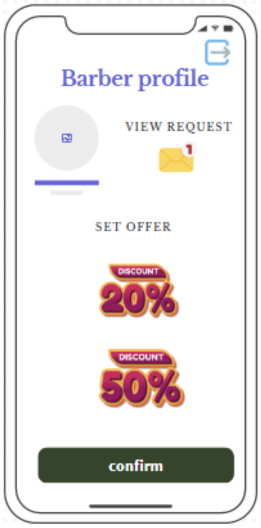


Barber profile

VIEW REQUEST

SET PRICE

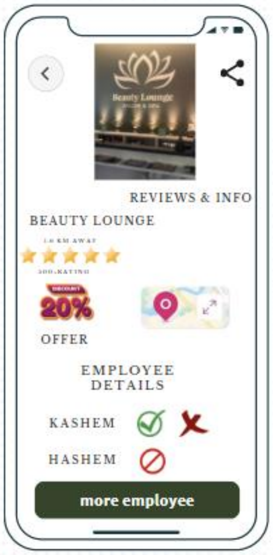
HAIRCUT	200BDT
BEAUTY AND SPA	500BDT
TATTOO	500BDT
MASSAGE	1000BDT
PET HAIRCUT	500BDT



Barber profile

VIEW REQUEST

SET OFFER



REVIEWS & INFO

BEAUTY LOUNGE

1.6 KM AWAY

★★★★★

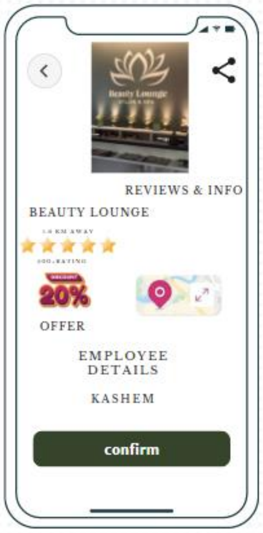
300+RATING

OFFER

EMPLOYEE DETAILS

KASHEM ✓ ✗

HASHEM ⊘



REVIEWS & INFO

BEAUTY LOUNGE

1.6 KM AWAY

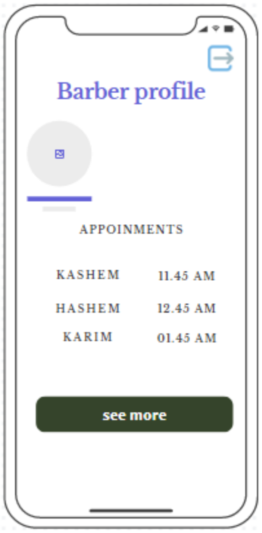
★★★★★

300+RATING

OFFER

EMPLOYEE DETAILS

KASHEM



Barber profile

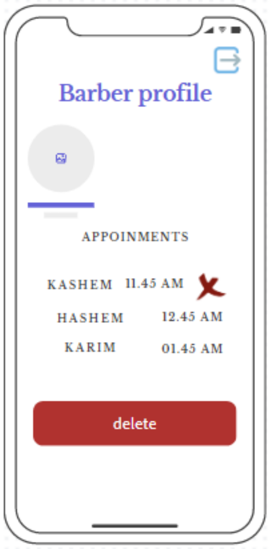
APPOINMENTS

KASHEM 11.45 AM

HASHEM 12.45 AM

KARIM 01.45 AM

see more



Barber profile

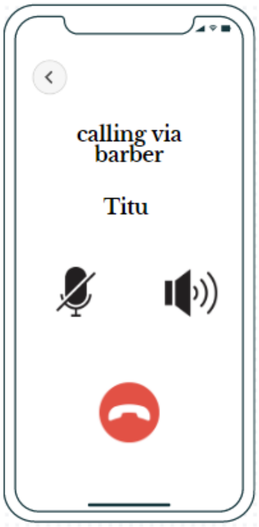
APPOINMENTS

KASHEM 11.45 AM

HASHEM 12.45 AM

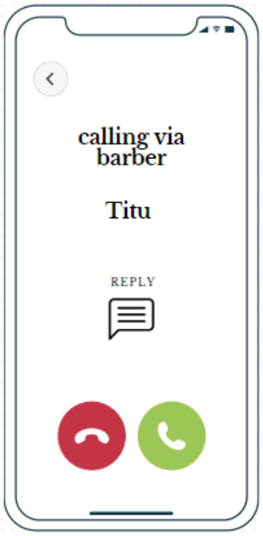
KARIM 01.45 AM

delete



calling via barber

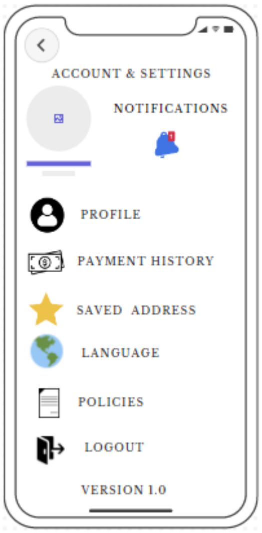
Titu



calling via barber

Titu

REPLY



ACCOUNT & SETTINGS

NOTIFICATIONS

PROFILE

PAYMENT HISTORY

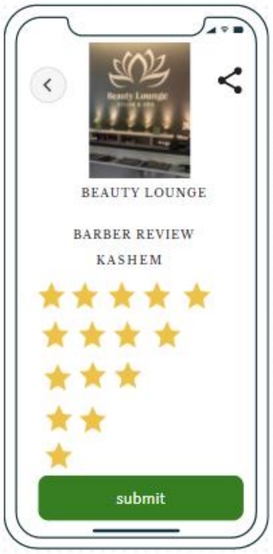
SAVED ADDRESS

LANGUAGE

POLICIES

LOGOUT

VERSION 1.0



BEAUTY LOUNGE

BARBER REVIEW

KASHEM

★★★★★

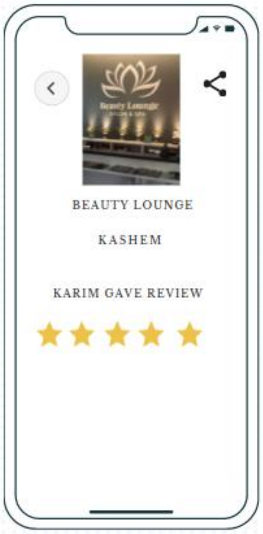
★★★★

★★★

★★

★

submit

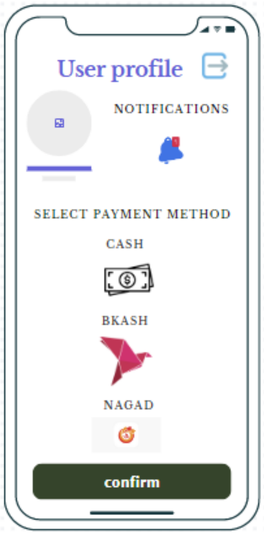


BEAUTY LOUNGE

KASHEM

KARIM GAVE REVIEW

★★★★★



User profile

NOTIFICATIONS

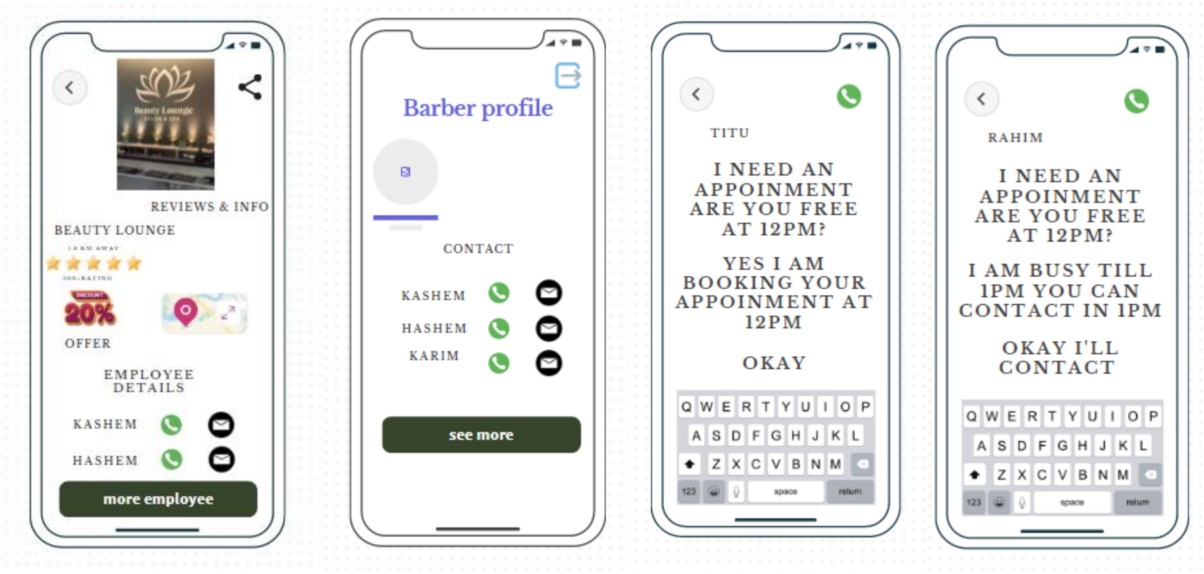
SELECT PAYMENT METHOD

CASH

BKASH

NAGAD

confirm



TITU

I NEED AN APPOINMENT ARE YOU FREE AT 12PM?

YES I AM BOOKING YOUR APPOINMENT AT 12PM

OKAY

RAHIM

I NEED AN APPOINMENT ARE YOU FREE AT 12PM?

I AM BUSY TILL 1PM YOU CAN CONTACT IN 1PM

OKAY I'LL CONTACT

4.4 Project Requirements

1) **Time**: We need 10 months (40 weeks) to build this software.
2) **Environment**: We don’t have any budget so we are working on own work place
3) **Resources**: We are total 4 human resources to build this software.
4) **Equipment**: To build this software we need equipment. Like, 4 devices , necessary IDE and frameworks.
5) **Bandwidth**: We need high bandwidth support. Which is around 50 to 80 Mbps.
6) **Tools**: The system developer needs selenium tools in perform testing activities in week.

5. FEATURES NOT TO BE TESTED

Below are some examples of areas or features that will not be specifically addressed in testing for this project:

- **Third-Party Integration Testing:** Testing the integration of the software with third-party services or systems will not be within the scope of this project. While the software may interact with external services, the testing of these integrations will be the responsibility of the providers of those services.
- **Hardware Compatibility:** The compatibility of the software with various hardware configurations will not be specifically tested. The focus of this project is on the software application itself, and compatibility issues arising from specific hardware setups will be addressed by users on their respective devices.

- **Network Performance:** While the software relies on network connectivity for certain functionalities, extensive network performance testing will not be conducted. Any network-related issues that affect the software's performance will be considered outside the scope of this project.
- **Browser and Platform Compatibility**: Testing the software's compatibility with a wide range of browsers, operating systems, and platforms will not be the primary focus. The software may be optimized for specific browsers or platforms, but comprehensive cross-platform testing is not explicitly included.
- **Security Penetration Testing:** In-depth security penetration testing to identify vulnerabilities and potential breaches will not be part of this project's testing scope. Basic security measures may be implemented, but a comprehensive security audit is beyond the project's objectives.
- **Localization and Internationalization Testing:** Testing the software's adaptation to different languages, cultures, and regions will not be extensively covered. The software may be designed with localization in mind, but specific testing for internationalization aspects may not be included.
- **Performance Under Load**: While basic performance testing might be conducted, in-depth load testing to evaluate the software's behavior under heavy user loads or stress conditions will not be a primary focus of this project.
- **Accessibility Testing**: While efforts may be made to adhere to accessibility guidelines, dedicated accessibility testing for compliance with specific standards like WCAG (Web Content Accessibility Guidelines) might not be explicitly addressed in this project.
- **Usability Testing**: While user experience is important, thorough usability testing involving representative user groups might not be fully covered in this scope. Any usability issues encountered during regular testing will be documented, but extensive usability studies might not be conducted.
- **Regression Testing of All Previous Versions**: While regression testing is important, testing every single feature of all previous versions might not be feasible. Focus will be on critical regression areas and new features.

6. TESTING APPROACH

6.1 Testing Levels

Testing Levels: The testing strategy for the upcoming Trimmer project consists of three essential testing levels: Unit Testing, System/Integration Testing, and Acceptance Testing. Due to the nature of the project and resource considerations, a dedicated full-time independent test person will oversee the System/Integration Testing, while the test manager, along with active participation from development teams, will handle the majority of testing responsibilities.

Unit Testing: During Unit Testing, our developers will play a pivotal role. They will design, execute, and evaluate the unit tests. Approval from the development team leader is essential before the tests are passed to the designated test person. To ensure quality, programmers will present evidence of thorough unit testing, including a test case list, sample output, data printouts, and defect details to the team leader. The designated test person will also receive all unit test information for comprehensive assessment.

System Testing: System/Integration Testing for "Trimmer" will be a collaborative endeavor led by the test manager and development team leader. While no specific test tools have been prescribed, the emphasis will be on meticulously testing the integrated system as a cohesive whole. This phase will commence only after addressing all major defects, ensuring the smooth integration of components. Programs entering this phase can accommodate up to two major defects, provided they don't hinder the overall testing process, and suitable workarounds are available.

Integration Testing: Integration Testing for the Barber Booking Management System (BBMS) project will be a collaborative undertaking involving the test manager, development team leader, and individual developers. In this phase, the aim is to ensure that different components of the software work harmoniously when integrated. Although specific test tools are not specified, the primary objective is to identify and rectify any issues arising from the interaction of these components.

Acceptance Testing: For Acceptance Testing, end-users will take the lead, supported by the test manager and development team leader. This phase will run parallel to the current manual process for a month following the completion of System/Integration Testing. By involving end-users, we aim to ensure the software aligns seamlessly with their needs and expectations.

6.2 Test Tools

Test Tools for Trimmer: For automated testing, we have employed the "Selenium" test tool, alongside the .NET framework for coding purposes. Our development environment of choice is Visual Studio 2019, where we write code using the C# programming language.

Selenium: "Selenium" is a widely used open-source automation testing framework tailored for web applications. This framework empowers developers and testers to simulate user interactions, perform functional tests, and validate web elements within web browsers. Its adaptability to multiple programming languages and integration with diverse testing frameworks makes "Selenium" a potent choice for web testing, encompassing regression and cross-browser testing. Leveraging its capabilities, we can identify and address issues early in the development cycle.

.NET Framework and Visual Studio: The ".NET" framework provides a suite of tools and libraries that facilitate structured and organized test development. Our preferred testing frameworks, such as NUnit, xUnit, and MSTest, assist developers in creating and executing unit tests and integration tests. Visual Studio 2019, our chosen integrated development environment (IDE), seamlessly integrates with renowned testing frameworks. This integration streamlines test creation, execution, and analysis. With its robust debugging features, test explorer, and code coverage tools, Visual Studio ensures efficient testing practices, fostering collaboration between development and testing teams.

C#: C# holds a prominent position as a versatile programming language, extensively used for testing purposes. Its features such as strong typing, object-oriented design, and integration with the .NET framework make it a suitable choice. C# supports test-driven development (TDD) and offers libraries like NUnit, MSTest, and xUnit for creating robust test suites. With C#, our developers can craft automated tests, perform unit testing, and ascertain the software's functionality, reliability, and performance. The expressive syntax, rich tooling, and compatibility with popular testing frameworks make C# an ideal fit for testing across diverse application domains.

6.3 Meetings

- Daily Standup Meetings (15 minutes):

Frequency: Daily, Monday to Friday

Purpose: Quick check-ins to discuss progress, challenges, and plans for the day.

Attendees: QA Manager, QA Engineers, Developers

Agenda: Each team member shares their accomplishments from the previous day, current tasks, and any roadblocks they are facing.

- Weekly Test Planning Meeting (1 hour):

Frequency: Every Monday

Purpose: Plan and prioritize testing activities for the upcoming week.

Attendees: QA Manager, QA Engineers, Project Manager, Product Manager

Agenda: Review the project status, upcoming features to be tested, prioritize testing tasks, discuss any scope changes, and assign responsibilities.

- Bi-weekly Test Review Meeting (1.5 hours):

Frequency: Every other Wednesday

Purpose: Review the test progress, share insights, and address challenges.

Attendees: QA Manager, QA Engineers, Product Manager, Developers

Agenda: Review the completed testing tasks, share test results and findings, discuss any blockers, and align on next steps.

- Monthly Test Metrics Review Meeting (1 hour):

Frequency: First Friday of every month

Purpose: Review overall testing metrics and performance.

Attendees: QA Manager, QA Engineers, Project Manager

Agenda: Present testing metrics such as defect density, test coverage, and test execution progress. Discuss areas of improvement and strategies to enhance testing efficiency.

- Ad-hoc Bug Triage Meetings (as needed):

Frequency: Based on the number of identified defects

Purpose: Review and prioritize reported defects.

Attendees: QA Manager, QA Engineers, Developers

Agenda: Review reported defects, assign severity and priority, and determine the timeline for fixes.

- Release Readiness Meeting (2 hours):

Frequency: Before each major release

Purpose: Ensure the application is ready for release from a testing perspective.

Attendees: QA Manager, QA Engineers, Project Manager, Product Manager, Developers

Agenda: Review the test coverage, test results, critical defects, and decide whether the application is ready for release.

7. TEST CASES/TEST ITEMS

Table: Test Case for **User Profile**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_1			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Create account			Test Execution date: 26-7-2023	
Test Title: Registration method				
Description: Test user valid registration profile				
Precondition (If any): Valid Phone number or Gmail				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Click “signup or join as barber” button 3. Fill the details 4. Click “Signup” button	Email: nnirob180@gmail.com Phone Number :01779776839 Database: Save data successfully.	User should be able to register into Trimmer app	As expected	Pass
Post Condition: The phone number and location will be updated anytime in the database when user select his current location or changes his contact number.				

Table: Test Case for **Booking Barber**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_2			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Booking Barber			Test Execution date: 26-7-2023	
Test Title: Getting barber and offers list.				
Description: Test to see the barber list and booked the barber.				
Precondition (If any): Customer need to login into app with valid ID.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Go to the User profile 2. Click “Search barber” or “Nearby offers”. 3. Select Saloon and barber from list. 4.Click “Sent Request” button	Offers: 20% discount for today Saloon: MegaHair-Khilkhet-2.00pm	User should be able see the slot and booked the barber.	As expected	Pass
Post Condition: The schedule will be updated continuously when a customer booked a barber.				

Table: Test Case for **Confirm Request**

Project Name: Trimmer			Test Designed by	
Test Case ID: TMR_3			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Conformation Request			Test Execution date :26-7-2023	
Test Title: To confirm user schedule request.				
Description: Test Conformation or rejection the customer.				
Precondition (If any): User need to be a sent request first.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Barber Profile 2. Click “View request” button 3.Select the user from list 4. Click “Confirm request or cancel request” button	Nafis89 request for 2.00pm slot Request: Confirmed or Rejected.	Barber should be able to see the request and can confirm or reject.	As expected	Pass
Post Condition: After conforming the request user get a notification from barber and update barber schedule automatically.				

Table: Test Case for **Barber Update**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_4			Test Designed date: 25-7-2023	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Barber Update			Test Execution date: 27-7-2023	
Test Title: Check schedule				
Description: Test barber profile updated schedule and offers.				
Precondition (If any): Have to login as Barber first.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Barber Profile 2. Click "Add offers" button 3. Add new offers to selected schedule. 4. Click "Sent Offer" button	Offer: 20% discount for Today Schedule: 11.00 Am 2.00 Pm	User should be see the offer and schedule.	As expected	Pass
Post Condition: After set the offers, the offer will automatically update to the customer.				

Table: Test Case for **Password Reset**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_5			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: Jafrin Sultana Juthi	
Module Name: Password Reset			Test Execution date: 27-7-2023	
Test Title: Resetting User Password				
Description: Test the functionality of resetting a user's password.				
Precondition (If any): User should have a registered email in the system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app	Reset password link sent to the email address.	User should	As expected	Pass

2. Click "Forgot Password?" option 3. Enter the registered email address 4. Click"Reset Password"button	Email: tokyshahriar555@gmail.com Database: Reset password link sent to the email address.	receive an email with a link to reset the password..		
Post Condition: The user should be able to reset the password using the link received in the email.				

Table: Test Case for **Service Availability**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_6			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: Jafrin Sultana Juthi	
Module Name: Barber Availability			Test Execution date: 27-7-2023	
Test Title: Checking Barber Availability				
Description: Test if barbers' availability is displayed accurately.				
Precondition (If any): Barbers should have set their availability in the system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Click "Search barber" option 3. Choose a specific barber 4. View their availability schedule	Barber: MegaHair-Khilkhet.	User should see the barber's accurate availability schedule.	As expected	Pass
Post Condition: The displayed schedule should match the barber's actual availability.				

Table: Test Case for **Payment Processing**

Project Name: Trimmer			Test Designed by:	
Test Case ID: TMR_7			Test Designed date: 24-7-2023	
Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Payment			Test Execution date: 27-7-2023	
Test Title: Processing Payment				
Description: Test if payment processing works seamlessly.				
Precondition (If any): User should have a valid payment method linked to their account.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the app 2. Choose a barber and select a service 3. Proceed to payment 4. Complete the payment process	Service Cost: $25 Payment Method: Credit Card	Payment should be successfully processed, and a payment confirmation should be displayed.	As expected	Pass
Post Condition: The user should receive a payment receipt and the transaction should be reflected in the app.				

8. ITEM PASS/FAIL CRITERIA

The "passed/failed" criteria in software testing are straightforward measures used to determine the success or failure of individual test cases. These criteria serve as a clear indicator of whether the software meets the expected standards and is ready for release.

1. Passed Criteria: When a test case is marked as "passed," it means that the specific functionality or scenario being tested has performed as expected. The software has met the predefined requirements, and no critical defects were detected during the test. Passing criteria often involve ensuring correct results, proper functionality, and adherence to specifications.

2. Failed Criteria: On the other hand, when a test case is marked as "failed," it indicates that the software did not meet the expected outcome. A test case fails when there are discrepancies between the observed behavior and the desired behavior. This can occur due to functional issues, unexpected errors, performance bottlenecks, or other reasons. Failed criteria typically require further investigation, analysis, and debugging to understand the root cause of the failure.

The decision to pass or fail a test case is based on predefined acceptance criteria, which are established before testing begins. These criteria are designed to align with the software's requirements and user expectations. The goal is to ensure that the software is thoroughly evaluated, and any issues are identified and addressed before the software is released to users. Using passed/failed criteria provides a clear and objective assessment of the software's quality, allowing the testing team and stakeholders to make informed decisions about the readiness of the software for the next stages of development or deployment.

9. TEST DELIVERABLES

1. **Acceptance Test Plan:** The acceptance test plan is a document outlining the strategy and criteria for validating that the Barber Booking System software meets user requirements and is ready for deployment. It defines the scope of testing, identifies the test scenarios, and specifies the entry/exit criteria for acceptance testing. This plan ensures that the software aligns with user expectations and business objectives, serving as a guideline for stakeholders to assess whether the product is suitable for acceptance and ready to move into production.
2. **System/Integration Test Plan:** The system/integration test plan outlines how different components of the Barber Booking System will be tested together (integration testing) and how the entire system will be tested as a whole. It includes the scope, test scenarios, test environment, and criteria for validating that the integrated system functions properly, ensuring that all parts work harmoniously and any potential issues are identified before deployment.
3. **Unit Test Plans/Turnover Documentation:** Unit test plans and turnover documentation provide detailed test cases and documentation for individual features and components of the Barber Booking System. These plans ensure that each part of the software functions correctly on its own, and turnover documentation aids in seamless communication between development and testing teams.
4. **Screen Prototypes:** Screen prototypes for the Barber Booking System are visual representations of software user interfaces, illustrating the layout, design, and interactions. These mockups serve as a tangible preview for stakeholders, aiding in clarifying requirements, gathering feedback, and ensuring that the final product aligns with user expectations before actual development begins.
5. **Appointment Booking Test Cases and Results:** Appointment booking test cases and results provide scenarios for booking appointments in the Barber Booking System. These test cases ensure that users can successfully schedule appointments, manage bookings, and receive confirmation. The results document the outcome of each test case, indicating whether the system functions as expected or if any issues are encountered.
6. **Defect/Incident Reports and Summaries:** Defect/Incident reports and summaries for the Barber Booking System software provide concise records of identified issues or anomalies in the system's functionality or performance. These reports highlight the defect's nature, severity, steps to reproduce, and relevant system information. Summaries offer an overview of all reported defects, enabling prioritization and efficient resolution by the development team, ensuring the software meets quality standards and user expectations.
7. **Test Logs and Turnover Reports:**Test logs and turnover reports are essential documentation in software testing for the Barber Booking System project. Test logs record

detailed information about the execution of test cases, including timestamps, test case IDs, outcomes (pass/fail), and any deviations from expected results. Turnover reports provide a comprehensive summary of the testing phase, highlighting key findings, outstanding issues, and the overall quality of the software. These documents ensure transparency, facilitate communication between testing and development teams, and guide decision-making for further development or deployment, contributing to the delivery of a reliable and user-friendly Barber Booking System.

10. STAFFING AND TRAINING NEEDS

Horizontal Approach: In building a proficient software quality testing team for the Barber Booking Management System project, it's crucial to assemble a group with a diverse array of skills to ensure thorough testing coverage. In this approach, team members have shared responsibilities across different areas of testing.

1. **Testers:** We need testers who possess the ability to meticulously design test cases, execute them rigorously, and effectively report any issues they encounter during testing.
2. **Test Automation Engineers:** This role specializes in creating and maintaining automated test scripts, an essential component for increasing testing efficiency in a horizontal approach.
3. **Quality Assurance Analysts:** Quality assurance analysts play a vital role in this approach, focusing on establishing quality standards and monitoring overall quality to ensure the final product meets user expectations.

Vertical Approach: Alternatively, we can consider a vertical approach, wherein each team member specializes in a specific area, contributing to a more focused and in-depth testing effort.

1. **Domain Specialists:** Considering the application's varied domains, such as appointment scheduling, customer management, and payment processing, having domain specialists who are experts in these specific areas is essential for a comprehensive and thorough testing process.
2. **Agile or DevOps Specialists:** Specialized knowledge of different testing methodologies, such as Agile or DevOps, will enable the team to seamlessly integrate testing into the development process.
3. **Performance and Security Testers:** Vertical specialization includes testers skilled in performance testing (using tools like J Meter) and security testing, who can thoroughly identify vulnerabilities specific to the Barber Booking Management System.

Regardless of the chosen approach, both technical and soft skills are equally important for the team's success.

Technical Skills: In both the horizontal and vertical approaches, training is essential. Team members need to be well-versed in various testing methodologies, including Agile or DevOps, to ensure alignment with the development process. Furthermore, providing training in test automation tools and best practices is vital. Specialized skills in performance testing and security testing must also be imparted to identify vulnerabilities unique to the system.

Soft Skills: Communication, collaboration, and problem-solving abilities are crucial for testers, especially when working closely with developers and other stakeholders. Effective communication of issues enhances the development process and leads to a more cohesive team effort.

Continuous Learning: Lastly, given the rapid evolution of software quality testing, continuous learning is key. Encourage team members to attend workshops, webinars, and conferences. Staying updated on the latest testing trends and technologies ensures the team's skills remain current, contributing to the delivery of a high-quality Barber Booking Management System.

By selecting either a horizontal or vertical approach, or a combination of both, and considering the skills, expertise, and training needs outlined above, we can assemble a capable testing team poised for success in delivering a top-notch Barber Booking Management System.

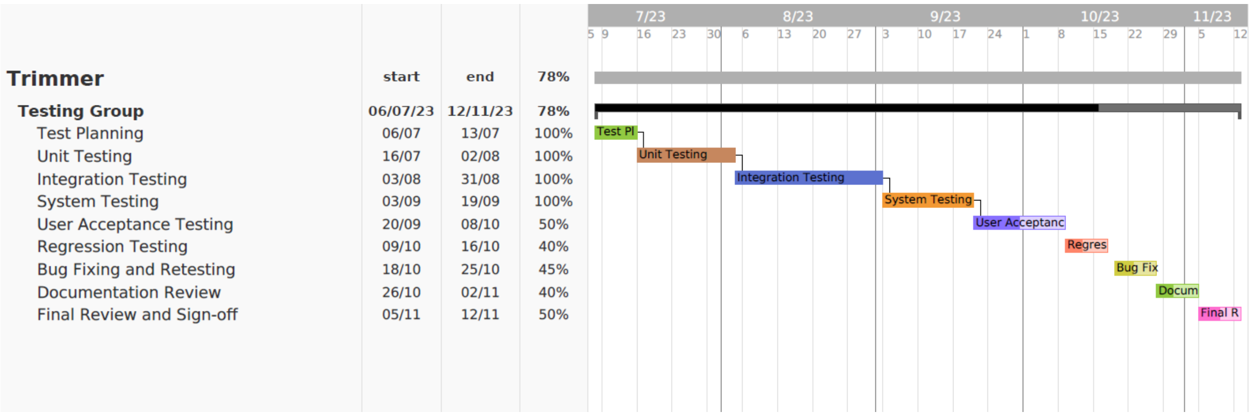
11. RESPONSIBILITIES

In any software development project, a collaborative and well-structured testing process is crucial to ensure the quality, functionality, and reliability of the final product. To achieve this, a diverse group of stakeholders plays vital roles in various aspects of testing. Each stakeholder brings unique expertise and responsibilities to the testing process, contributing to the overall success of the project. From defining testing goals and strategies to executing test cases and providing valuable feedback, the involvement of these stakeholders creates a comprehensive testing ecosystem that enhances the development and delivery of a robust software solution.

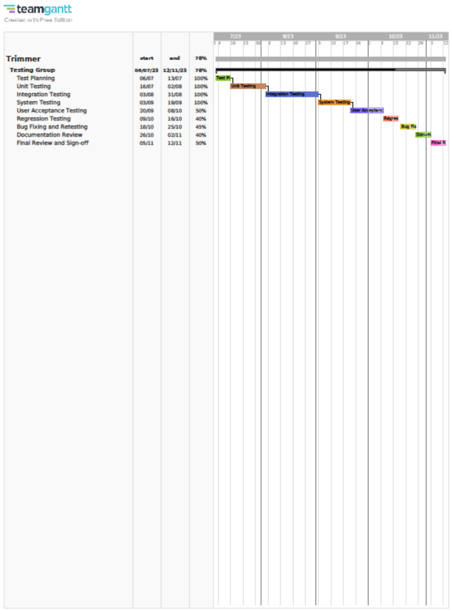
To have a better understanding of the responsibilities of the stakeholders, a tabular representation is provided :

Stakeholder	Role and Responsibilities
Product Manager	Define testing goals and objectives
	Prioritize testing activities based on business goals
	Provide testing resources and budget
	Review and approve test plans and strategies
Project Manager	Coordinate testing efforts with the development team
	Ensure testing milestones are met
	Communicate testing progress and issues to relevant stakeholders
	Monitor testing schedule and resource allocation
Development Team	Develop unit tests and ensure code quality
	Participate in writing and reviewing test cases
	Assist in creating testing environments and data
	Fix and verify defects identified during testing
QA Manager	Develop overall testing strategy and approach
	Define testing methodologies and best practices
	Assign tasks to QA team members and oversee their work
	Review and approve test cases and test scripts
QA Engineers	Design and execute test cases based on requirements
	Perform manual and automated testing
	Document and report defects accurately and comprehensively
	Provide feedback on the quality of the application
Business Analysts	Participate in requirement gathering and validation
	Review test cases to ensure alignment with business requirements
	Assist in defining test data and scenarios
End Users	Provide feedback on usability and functionality
	Report defects and issues during testing
	Validate that the application meets business needs

12. TESTING SCHEDULE



Trimmer	start	end	78%
Testing Group	**06/07/23**	**12/11/23**	**78%**
Test Planning	06/07	13/07	100%
Unit Testing	16/07	02/08	100%
Integration Testing	03/08	31/08	100%
System Testing	03/09	19/09	100%
User Acceptance Testing	20/09	08/10	50%
Regression Testing	09/10	16/10	40%
Bug Fixing and Retesting	18/10	25/10	45%
Documentation Review	26/10	02/11	40%
Final Review and Sign-off	05/11	12/11	50%



13. PLANNING RISKS AND CONTINGENCIES

This section addresses the risks that testing activities might encounter. By identifying these risks, evaluating their likelihood and impact, and formulating mitigation plans, the project can navigate uncertainties effectively, ensuring a smoother testing process and project outcome.

S/N	*Risk Description*	*Probability*	*Impact*	*Mitigation Plan*
1	Insufficient Test Coverage	50%	Lower Quality	Implement comprehensive test case review process to ensure maximum coverage. Prioritize critical scenarios
2	Resource Constraints	30%	Delayed Testing	Cross-train team members to handle multiple testing tasks. Have backup resources identified for critical roles.
3	Changing Requirements	40%	Scope Creep, Repeated Work	Establish a well-defined change management process. Regularly communicate with stakeholders to freeze requirements.
4	Unstable Test Environments	20%	Inaccurate Results	Set up dedicated testing environments that mimic production as closely as possible. Continuously monitor stability.
5	Data Corruption or Loss	15%	Data Inconsistencies	Implement automated data backup and restoration mechanisms. Regularly validate data integrity during testing.
6	Miscommunication Among Teams	25%	Misaligned Efforts	Foster clear communication channels. Regularly conduct cross-team meetings to ensure alignment and shared understanding.
7	Technology Compatibility Issues	20%	Functional Limitations	Thoroughly test software on various platforms and browsers. Keep up-to-date with technology changes and updates.
8	Lack of User Participation	25%	Poor User Experience	Engage users early in the process. Provide clear instructions and support for user acceptance testing.

14. APROVALS

Project Manager

Quality Assurance Manager

Lead Developer

Product Manager

End User Representative