

01

CHAPTER



INTRODUCTION



1.1 INTRODUCTION

QuickFix is a mobile-based repairman finder application designed to simplify the process of finding reliable repair services such as electricians, plumbers, carpenters, and appliance technicians. The system connects customers with nearby skilled repair professionals using GPS-based location tracking, ensuring quick and efficient service delivery. Users can easily register, search for required services, view available repairmen, book appointments, and track the repairman's arrival in real time through the application.

The QuickFix app allows repairmen to create profiles, list their services, set availability, and receive booking requests from customers. Once a service is booked, the system facilitates communication between the customer and the repairman, providing details such as service location, estimated arrival time, and job status updates. The platform supports real-time notifications and status tracking, ensuring transparency and convenience for both users and service providers.

This application is particularly useful in urban and semi-urban areas where finding trustworthy repair services quickly can be challenging. By digitizing the service booking process, QuickFix eliminates the need for manual searching or contacting multiple service providers. The system improves efficiency, reduces waiting time, and ensures that customers receive timely assistance during emergencies or urgent repair needs.

One of the key features of QuickFix is its GPS-based tracking system, which helps customers locate nearby repairmen and monitor their movement after booking. The app also includes service history and feedback features, allowing users to rate repairmen based on service quality, thereby improving reliability and trust within the platform.

QuickFix offers several advantages, including ease of use, time savings, and improved accessibility to skilled repair services. It provides a cost-effective solution for both customers and repairmen, promoting local employment opportunities while delivering convenient on-demand services. However, the system also has certain limitations.



1.2 PROBLEM

- Finding reliable technicians for AC service, plumbing, and electrical work is time-consuming and uncertain.
- Existing platforms focus mainly on premium services, have limited emergency support, and do not fully support small or urgent local repair needs.
- This causes inconvenience, delays, and trust issues for users.

1.3 PROBLEM SOLUTIONS

- The QuickFix Repairman Finder App provides a centralized digital platform that helps users quickly find verified and reliable technicians for AC service, plumbing, electrical, and other repair needs, reducing the time and uncertainty involved in manual searching.
- The system uses GPS-based location tracking to connect users with nearby local repairmen, ensuring faster response times, especially during urgent and emergency situations.
- Unlike existing platforms that focus mainly on premium services, QuickFix supports affordable local repair services, making it suitable for small, routine, and emergency repairs.
- The application includes real-time booking, notifications, and service status updates, minimizing delays and improving communication between users and repairmen.



02

CHAPTER



LITERATURE SURVEY



2.1 LITERATURE SURVEY

Literature Survey Study 1: Mobile Application for Domestic Services

Author contributions:

Mrs. G. Aishwaryalakshmi (Lecturer), J. Arisudan (Student), Aryan Kisan Tati (Student), B. Meyanban (Student)

Department of Information Technology
PSG Polytechnic College, Coimbatore, India

● Introduction

In the modern digital era, mobile applications have become an essential part of daily life. People increasingly rely on smartphones to access services quickly and efficiently. However, finding reliable domestic service providers such as plumbers, electricians, cleaners, and carpenters remains a challenge in many areas. Traditional methods of hiring technicians are often based on personal references or local advertisements, which may not guarantee reliability, availability, or quality. To address this issue, a centralized digital platform is required. This project proposes an Android-based Mobile Application for Domestic Services, which acts as a marketplace connecting customers with skilled technicians. The application simplifies the process of searching, booking, and paying for domestic services.

● Problem Statement

In many regions, customers face difficulties in finding reliable domestic service providers such as plumbers, electricians, carpenters, cleaners, and appliance repair technicians. The proposed mobile application solves these problems by offering a structured digital system that ensures convenience, reliability, and efficient service management.



Literature Survey Study 2: FixMate – A Smart Digital Platform for Booking Skilled Household Technicians

Author contributions:

Shahbaz Aazmi, Sufiyan Khan, Sufiyan Ahmed, Saqueef Ahmed, Dr. S.S. Agrawal Dept. of Computer Science & Engineering, SGBAU, Maharashtra, India.

● Introduction

FixMate is a smart digital platform (available on web and mobile) developed to bridge the gap between customers and skilled household technicians. In today's fast-paced lifestyle, people often struggle to find reliable service professionals for household repairs and maintenance. FixMate aims to provide a centralized, transparent, and secure solution for booking verified technicians easily and efficiently.

● Problem Statement

Finding trustworthy and skilled household technicians such as plumbers, electricians, carpenters, AC repair technicians, and cleaners is often challenging. FixMate addresses these issues by offering a structured online system that ensures reliability, transparency, and convenience. It also supports technicians by giving them digital recognition, fair work opportunities, and regular income sources.



03

CHAPTER

● ● PROJECT REQUIREMENTS



PROJECT REQUIREMENTS

3.1 Flutter



Flutter is an open-source UI development framework created by Google. It is used to build mobile, web, and desktop applications using a single codebase. Flutter is popular because it allows developers to create attractive and responsive applications easily. Flutter uses the Dart programming language and provides many pre-designed widgets to build modern user interfaces. It compiles directly into native code, which ensures high performance and smooth animations. It is widely used for Android and iOS app development.

Features of Flutter

- **Cross-Platform Development**
Developers can create apps for Android and iOS using one single codebase.
- **Hot Reload**
Allows instant viewing of code changes without restarting the app.
- **Rich Widgets**
Provides customizable widgets for designing attractive UIs.
- **High Performance**
Compiles to native ARM code for smooth and fast performance.
- **Open-Source Framework**
Free to use with strong community support and documentation.



3.2 Node js



Node.js is an open-source runtime environment that allows JavaScript to run on the server side. It is built on Chrome's V8 JavaScript engine and is mainly used for backend development of web and mobile applications. Node.js is known for its speed and efficiency. It uses an event-driven, non-blocking model, which helps handle multiple requests at the same time. It is widely used to create APIs, servers, and real-time applications.

Features of Node.js

- **Event-Driven Architecture**
Handles multiple user requests asynchronously without blocking operations.
- **Fast Execution**
Uses V8 engine for quick processing of JavaScript code.
- **Scalability**
Suitable for developing large and high-traffic applications.
- **NPM (Node Package Manager)**
Provides thousands of open-source libraries for faster development.
- **Cross-Platform Support**
Runs on Windows, Linux, and macOS systems.



3.3 Firebase



Firebase

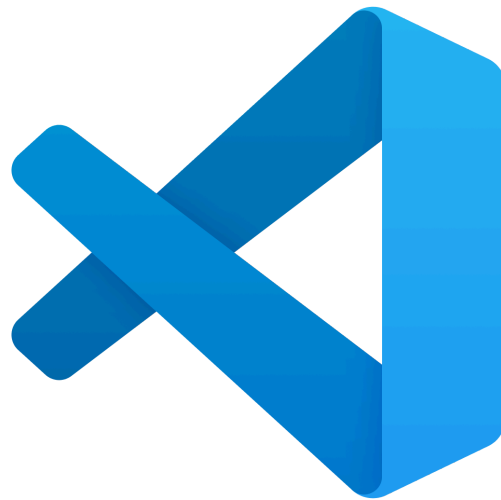
Firebase is a cloud-based Backend-as-a-Service (BaaS) platform developed by Google. It provides backend services such as database, authentication, hosting, and storage for applications. Firebase helps developers build applications quickly without managing physical servers. It supports real-time data synchronization and secure cloud storage, making it suitable for modern applications.

Features of Firebase

- Authentication
Provides secure login options such as email and phone authentication.
- Realtime Database
Synchronizes data instantly across all connected users.
- Cloud Firestore
Flexible NoSQL cloud database for structured data storage.
- Cloud Storage
Securely stores images, documents, and other files.
- Cloud Functions
Allows backend logic to run automatically without managing servers.



3.4 Visual Studio Code



Visual Studio Code (VS Code) is a free source code editor developed by Microsoft. It is lightweight but powerful and supports many programming languages. VS Code is widely used by developers because of its speed, flexibility, and extension support. It provides tools for debugging, running code, and version control, making software development easier and more organized.

Features of Visual Studio Code

- **Extension Support**
Allows installation of extensions for different programming languages.
- **Integrated Terminal**
Enables running commands directly inside the editor.
- **Debugging Tools**
Helps detect and fix errors efficiently.
- **IntelliSense**
Provides smart code suggestions and auto-completion.
- **Version Control Integration**
Supports Git for managing project versions.



04

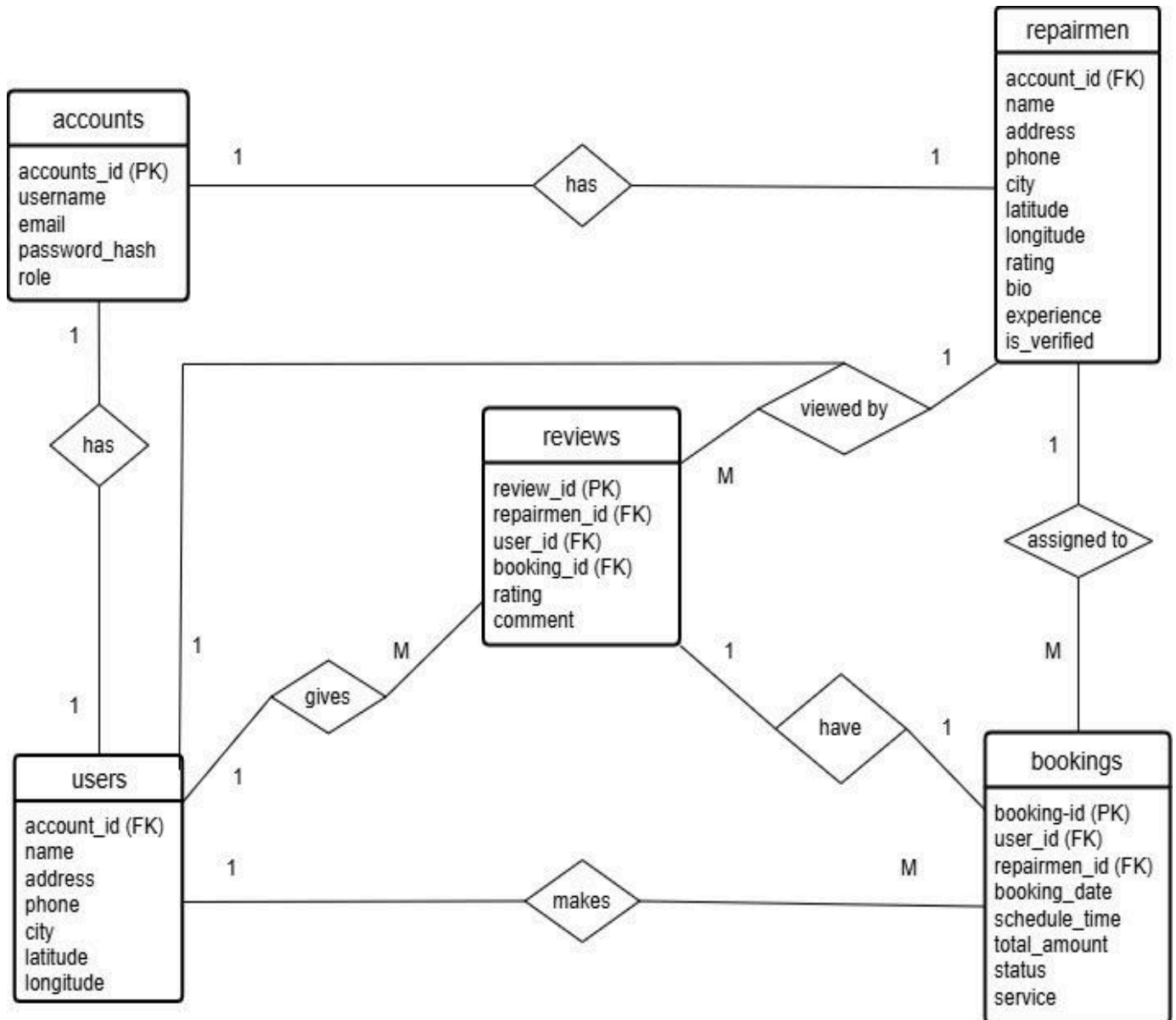
CHAPTER



SYSTEM DESIGN

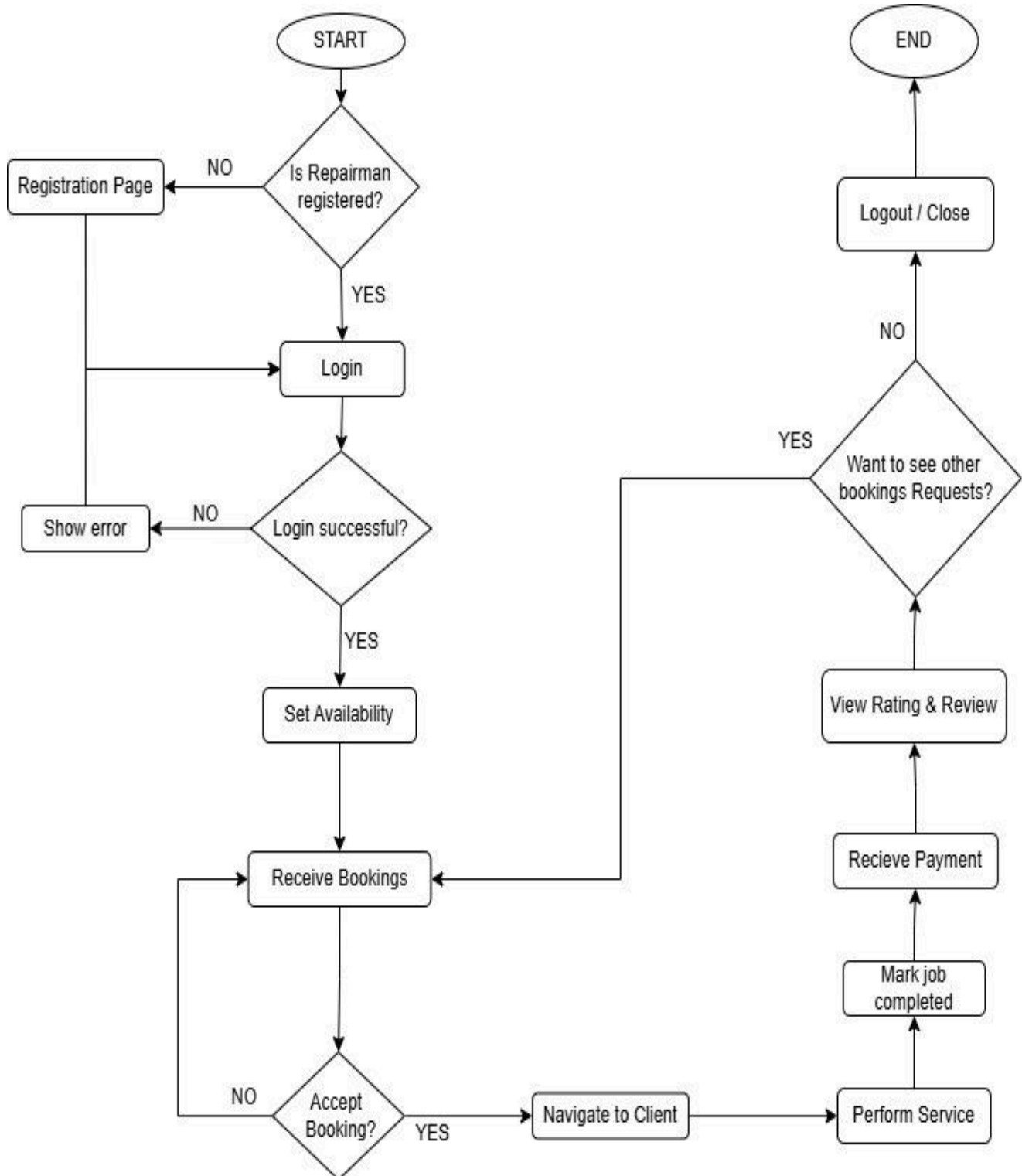


4.1 ER Diagram

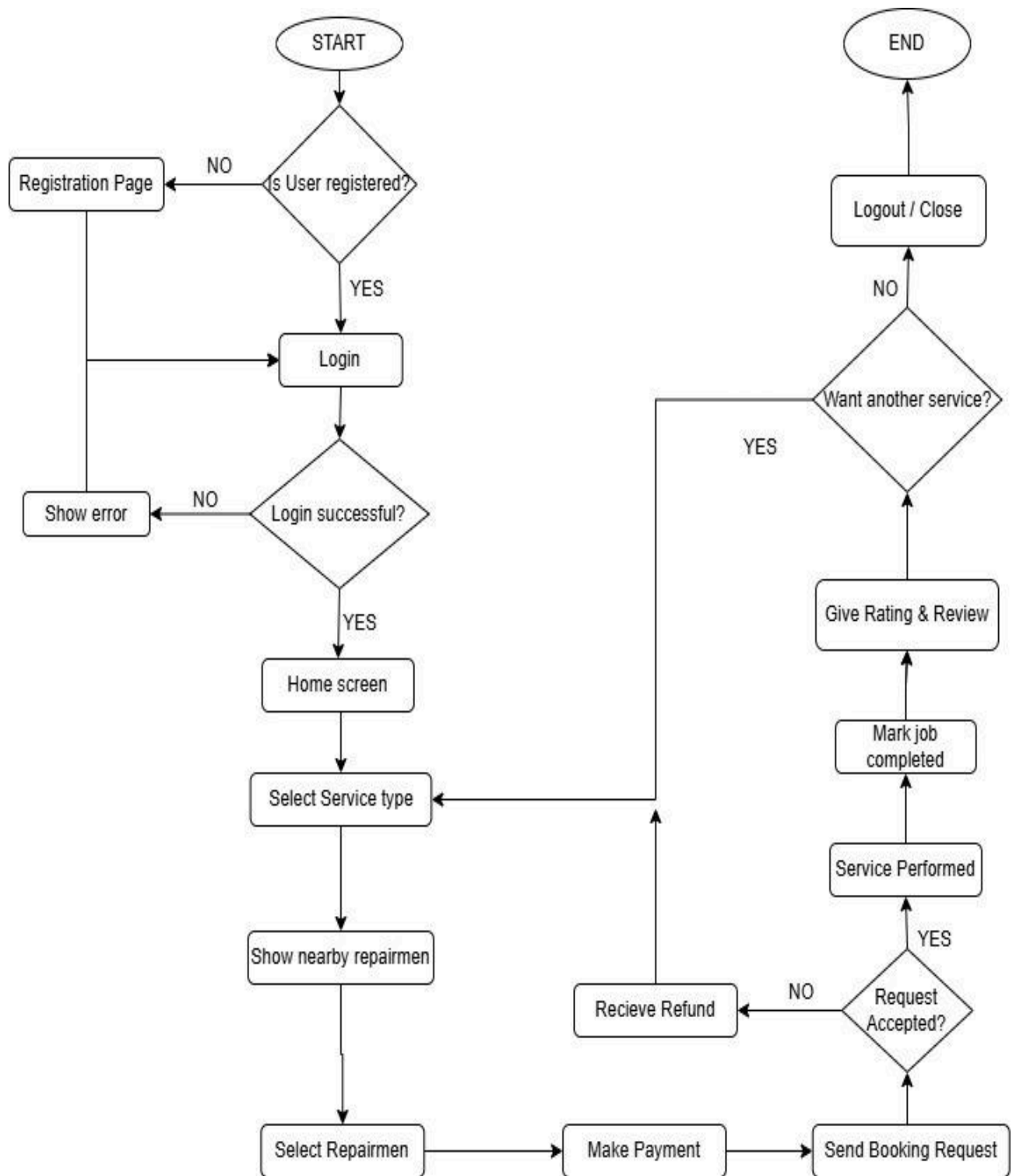


4.2 Flowchart Diagram:

4.2.1 Repairman



4.2.2 Client



4.3 UI/UX Designs



Login

Username or Email

Password

Forgot password?

Log in

Use your QuickFix username or email and password to sign in. Apple and Google sign-

Fig 4.3.1- Log in page

Signup as Repairman

Username

Full name

Email

Phone No

Address

Bicholim

Select Location

Password

Confirm Password

Select Skill

Fig 4.3.2 - Repairman Sign up page

Signup as User

Username

Full name

Email

Address

Bicholim

Select Location

Phone No

Password

Confirm Password

Create account

Fig 4.3.3 - User Sign up page

Repairman Dashboard

Welcome back, **Ankit Repairman**

Today at a glance

Stay on top of incoming work and keep emergency support ready.

Pending 1	Earnings Rs 269.98
---------------------	------------------------------

1
Pending Jobs
Waiting for response

6
Completed
Finished bookings

Emergency Services

You are visible for urgent repair requests. 🔴

Quick Actions

🏠 Home
📅 Booking
📍 Map
👤 Profile

Fig 4.3.4 - Repairmen home page

Welcome back **Riyan user**

Location: **Curchorem**

Need urgent help?

Request repairmen who have emergency service turned on. Open

🔍 Search repair, cleaning, ...

Popular Services

Pick a category and book in minutes

Mechanic

Professional car repair services

Electrician

Professional electrical services

Plum

Professional plumbing

Nearby Repairmans

Trusted professionals available around your area

Suresh AC Technician

🏠 Home
📅 Booking
📍 Map
👤 Profile

Fig 4.3.5 - User Home page



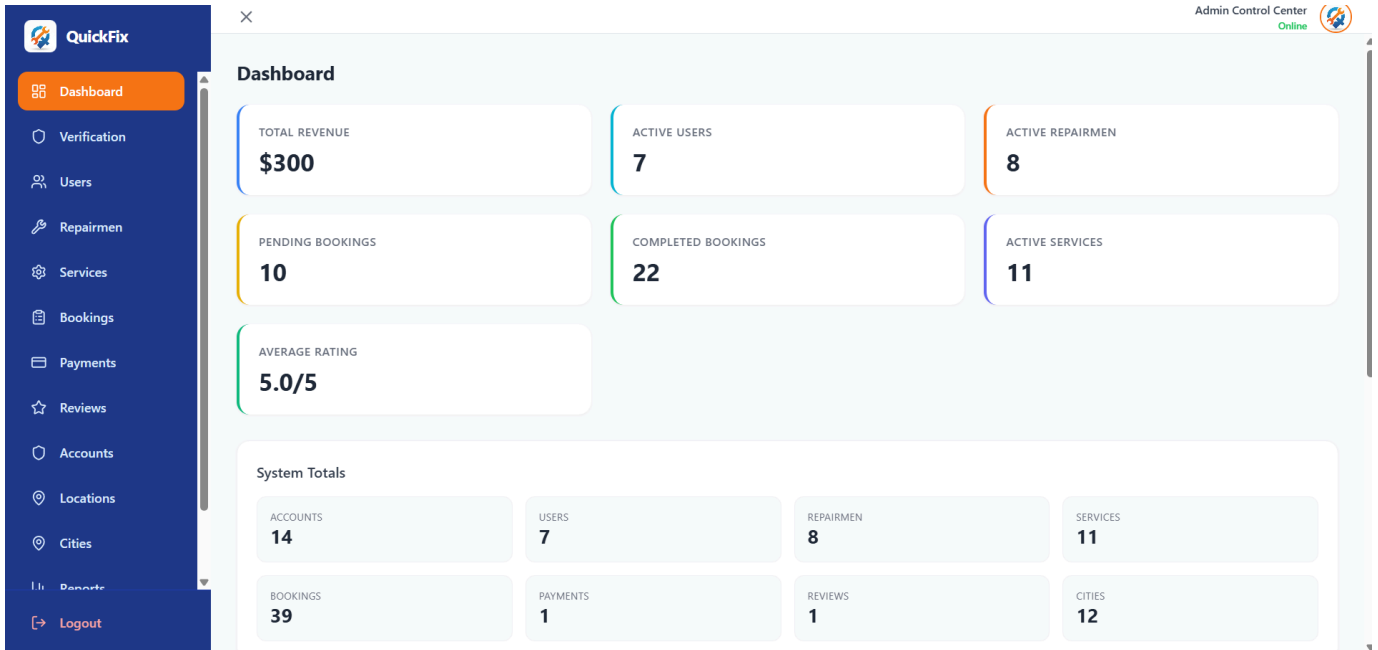


Fig 4.3.6 - Admin panel

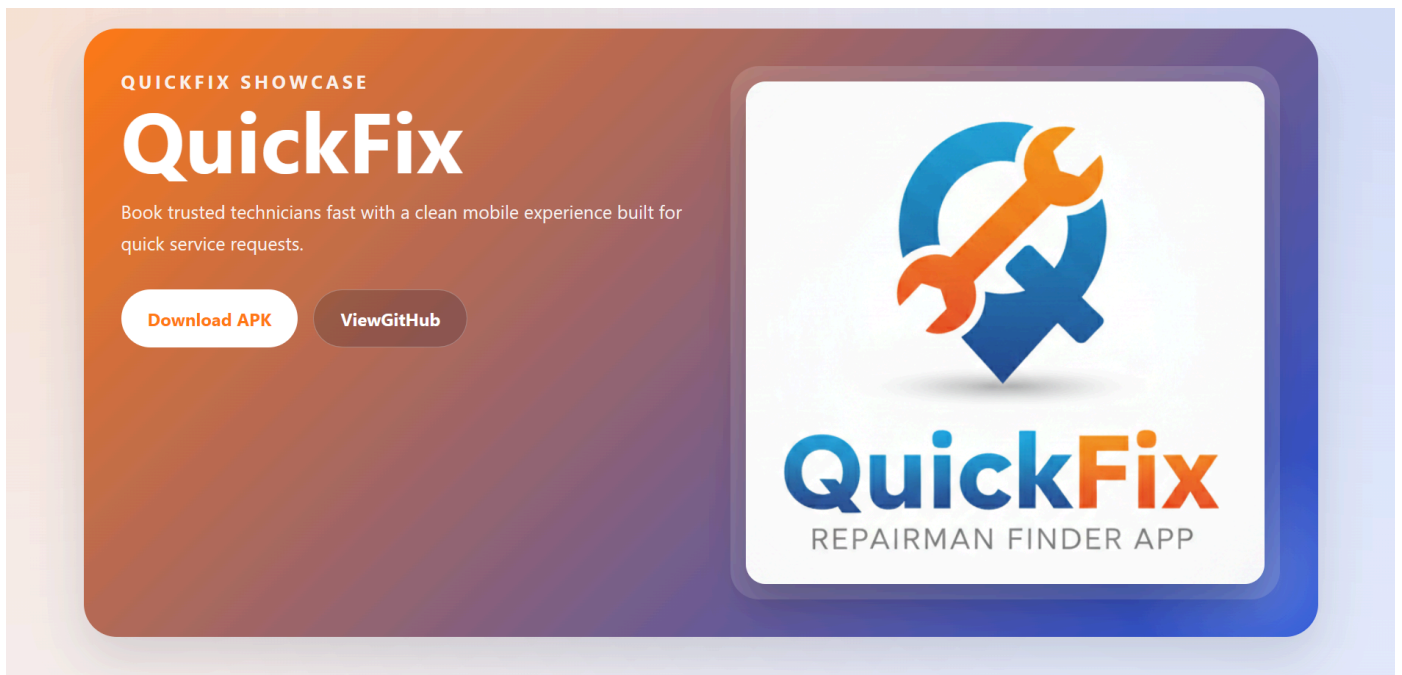


Fig 4.3.7 - Website



05

CHAPTER



IMPLEMENTATION



5.1 UI Design

The User Interface (UI) design of the QuickFix – Repairmen Finder application focuses on providing a simple, intuitive, and user-friendly experience for both customers and repairmen. The UI is designed to ensure easy navigation, quick access to services, and smooth interaction between users and repair professionals.

The interface was designed using Figma, which allowed the development team to create wireframes, layouts, and interactive prototypes before implementing the design in Flutter. The UI follows a clean and consistent design theme, ensuring that all screens maintain uniform colors, typography, and layout structure.

The main UI screens included in the QuickFix application are:

Splash / Welcome Screen: Displays the QuickFix logo and loads the application when it starts.

User Role Selection Screen: Allows users to select whether they are a Customer or a Repairman.

Login and Registration Screen: Enables users to create an account and securely log in to the application.

Customer Dashboard: Displays available repair services and nearby repairmen.

Service Selection Screen: Allows customers to select services such as electrician, plumber, carpenter, or appliance repair.

Repairman Profile Screen: Displays details about repairmen including experience, skills, hourly rate, and ratings.

Booking Request Screen: Allows customers to request repair services and schedule appointments.

Repairman Dashboard: Allows repairmen to view service requests, manage bookings, and update their availability.

The UI design ensures that the application remains simple, visually appealing, and easy to use, helping users quickly find and book reliable repair services.



5.2 Frontend Implementation

The frontend of the QuickFix – Repairmen Finder application is developed using Flutter, an open-source UI toolkit created by Google. Flutter enables developers to build high-performance and visually attractive mobile applications using a single codebase. It uses the Dart programming language and follows a widget-based architecture, which allows the creation of responsive and interactive user interfaces.

Flutter provides a wide range of customizable widgets that help design components such as forms, service cards, navigation menus, dashboards, and booking interfaces. These widgets are used to build important screens in the QuickFix application, including the login and registration page, role selection screen, customer dashboard, service listing page, booking interface, and repairman dashboard.

The frontend is designed to provide a clean and easy-to-use interface for both customers and repairmen. Customers can easily search for repair services, view repairman profiles, send service requests, and track bookings. Repairmen can log in to manage their profiles, accept or reject service requests, and update their availability.

Flutter's Hot Reload feature allows developers to instantly view changes in the application without restarting it, which speeds up the development process. The frontend communicates with the backend services through secure APIs to retrieve and update user and booking data stored in the database.

Overall, the Flutter-based frontend ensures smooth performance, responsive design, and efficient interaction between users and the QuickFix system.



5.3 Backend Implementation

The backend of the QuickFix – Repairmen Finder application is implemented using Firebase, a cloud-based Backend-as-a-Service (BaaS) platform provided by Google. Firebase manages server-side operations such as user authentication, database management, and secure communication between the application and the cloud database.

Firebase Authentication is used to handle user registration and login securely. It verifies user identities using email and password authentication and ensures that only authorized users can access the system. The system defines two roles: Customer and Repairman. Customers can search for services and request repairs, while repairmen can manage their service profiles and respond to booking requests.

The application uses Firebase Cloud Firestore as its primary database to store user profiles, service details, booking requests, and repairman information. Firestore is a cloud-based NoSQL database that supports real-time data synchronization, which ensures that updates such as new service requests or booking confirmations are instantly reflected in the application.

Firebase also ensures secure communication between the frontend and backend using encrypted connections and security rules. Input validation and authentication checks are implemented to prevent unauthorized access and incorrect data entries.

By using Firebase as the backend platform, the QuickFix system achieves high scalability, secure data management, improved performance, and reduced server maintenance, making the application reliable and efficient for everyday use.



5.4 Database Implementation

The database design of the QuickFix – Repairmen Finder application is implemented using Firebase Cloud Firestore, a cloud-based NoSQL database used to store and manage application data such as user profiles, repair services, booking requests, and repairman details.

In Firestore, data is organized into collections and documents. Each collection stores a specific type of data, and each document contains related information in a key-value format. This structure allows efficient storage, retrieval, and updating of data.

The main collections used in the QuickFix database are:

Users: Stores user details such as name, email, phone number, role (customer or repairman), and login information.

Repairmen: Stores information about repair professionals including skills, experience, hourly rate, location, and service description.

Services: Stores different types of repair services such as electrician, plumber, carpenter, appliance repair, and technician services.

Bookings: Stores service request details including customer ID, repairman ID, service type, booking date, and service status.

Reviews and Ratings: Stores customer feedback and ratings for repairmen after service completion.

Each user in the system is assigned a unique user ID, which links their booking requests, profiles, and service records.

Using Firebase Cloud Firestore allows the QuickFix system to provide real-time updates, secure data storage, and reliable data management, ensuring smooth operation of the Repairmen Finder application.



06

CHAPTER

● ● | TESTING



6.1 Test Cases and Scenarios

6.1.1 User Authentication Test Cases

User Login with Valid Credentials

Test Scenario: Verify that a registered user can log in successfully into the QuickFix application.

Preconditions:

- The user must be registered in the system.
- Valid email/username and password must be available.

Test Steps:

1. Open the QuickFix application.
2. Enter a valid email/username.
3. Enter the correct password.
4. Click on the Login button.

Expected Result:

The user should successfully log in and be redirected to the User Dashboard/Home Page.

Status: Pass

Login Page, User Dashboard

User Login with Invalid Credentials

Test Scenario: Verify that login fails when incorrect credentials are entered.

Preconditions:

- The user must have an account in the system.

Test Steps:

1. Open the QuickFix application.
2. Enter an incorrect email or password.
3. Click on the Login button.

Expected Result:

The system should display an error message such as “Invalid username or password” and prevent login.

Status: Pass

Invalid Login Attempt



6.1.2 Repairman Search Test Cases

Search for Repairman by Skill

Test Scenario: Verify that users can search for repairmen based on required service (electrician, plumber, carpenter, etc.).

Preconditions:

- Repairmen must be registered in the system with their skills listed.

Test Steps:

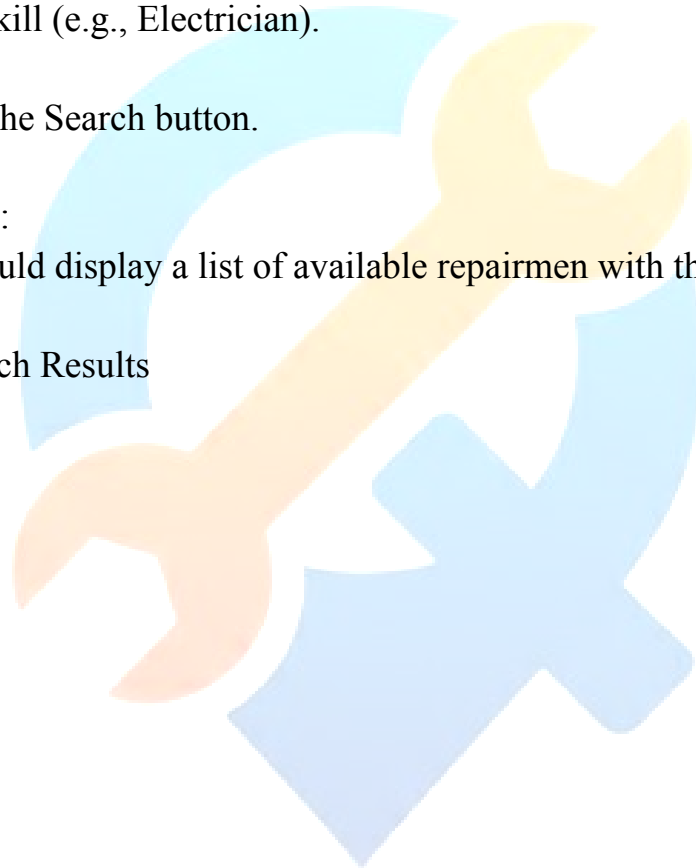
1. Open the QuickFix application.
2. Navigate to the Search Service section.
3. Select a skill (e.g., Electrician).
4. Click on the Search button.

Expected Result:

The system should display a list of available repairmen with the selected skill.

Status: Pass

Repairman Search Results



6.1.3 Booking Service Test Cases

Book a Repairman Service

Test Scenario: Verify that a user can successfully book a repairman service.

Preconditions:

- The user must be logged into the application.
- At least one repairman must be available in the system.

Test Steps:

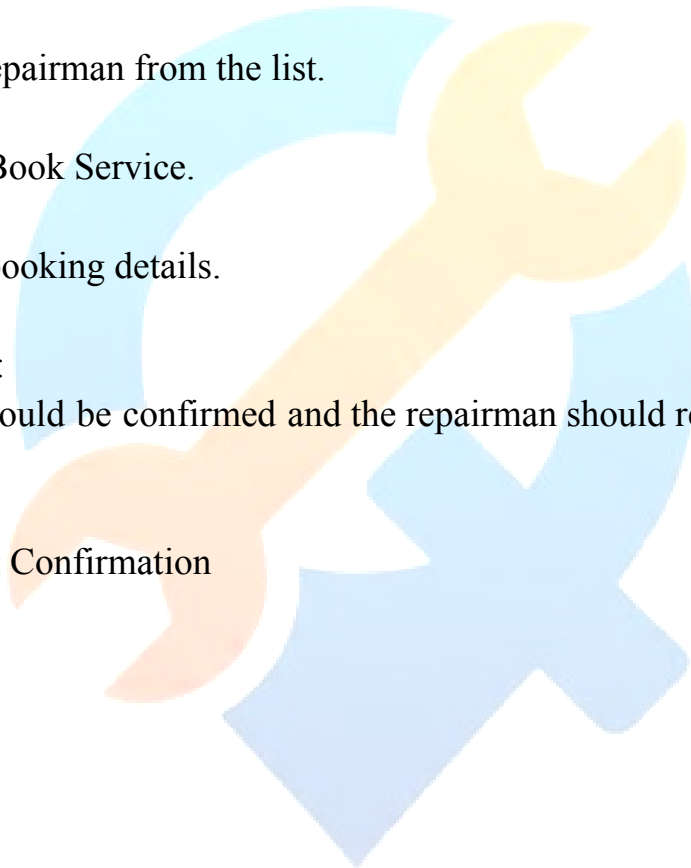
1. Login to the QuickFix application.
2. Search for a repairman.
3. Select a repairman from the list.
4. Click on Book Service.
5. Confirm booking details.

Expected Result:

The booking should be confirmed and the repairman should receive the service request.

Status: Pass

Service Booking Confirmation



6.1.4 Notification System Test Cases

Service Request Notification

Test Scenario: Verify that a notification is sent to the repairman when a user books a service.

Preconditions:

- The repairman must be registered and logged in.

Test Steps:

1. The user books a repair service.
2. The system sends a notification to the repairman.

Expected Result:

The repairman receives a service request notification in the application.

Status: Pass

Service Request Notification

Booking Confirmation Notification

Test Scenario: Verify that the user receives confirmation after booking a repair service.

Preconditions:

- The booking request must be successfully submitted.

Test Steps:

1. The user books a repair service.
2. The system processes the request.

Expected Result:

The user receives a booking confirmation notification with repairman details.

Status: Pass

Booking Confirmation Message



6.1.5 System Performance Test Cases

System Response Time

Test Scenario: Verify that the application loads pages and processes requests within an acceptable time.

Preconditions:

- The application server must be online.
- Internet connection must be available.

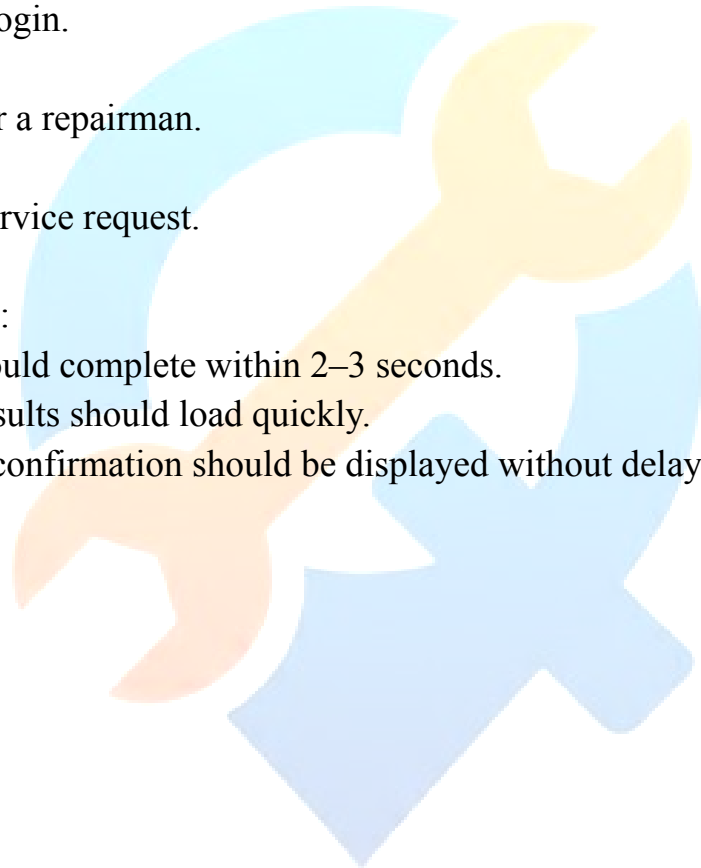
Test Steps:

1. Launch the QuickFix application.
2. Perform login.
3. Search for a repairman.
4. Book a service request.

Expected Result:

- Login should complete within 2–3 seconds.
- Search results should load quickly.
- Booking confirmation should be displayed without delay.

Status: Pass



07

CHAPTER



CONCLUSION AND
RECOMMENDATION

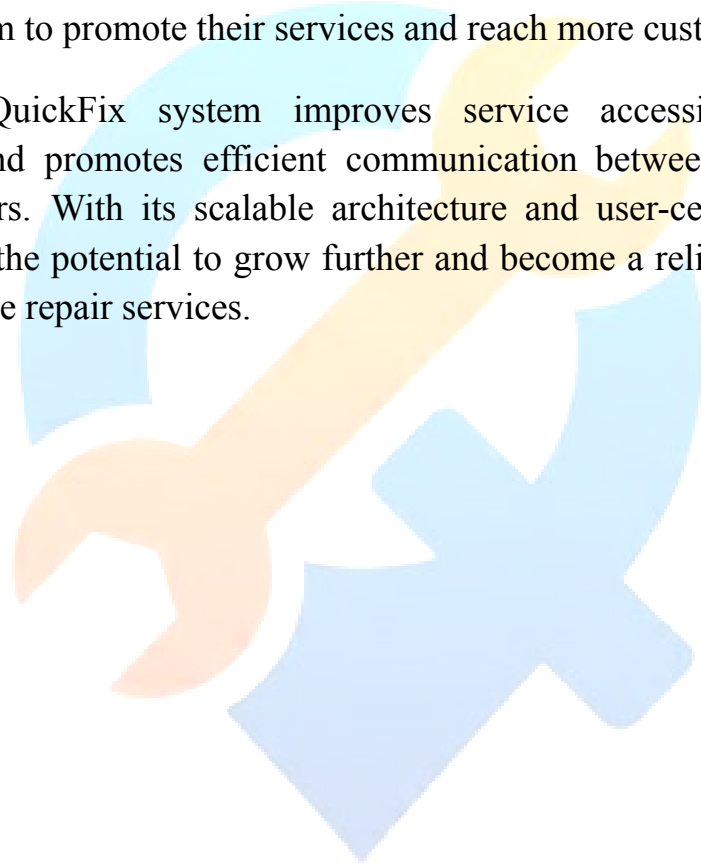


7.1 Conclusion

QuickFix Application provides an effective and modern solution for connecting users with skilled repair professionals quickly and conveniently. The application simplifies the process of finding reliable repair services such as electricians, plumbers, carpenters, and technicians through a user-friendly mobile interface. By integrating technologies like **Flutter, Node.js, and Firebase**, the system ensures smooth performance, real-time data management, and secure communication between users and repairmen.

QuickFix successfully reduces the time and effort required to search for trustworthy repair services in local areas. The platform allows users to view repairmen profiles, experience, hourly rates, and contact details, helping them make informed decisions. At the same time, it provides repair professionals with a digital platform to promote their services and reach more customers.

Overall, the QuickFix system improves service accessibility, enhances convenience, and promotes efficient communication between customers and service providers. With its scalable architecture and user-centric design, the application has the potential to grow further and become a reliable platform for on-demand home repair services.

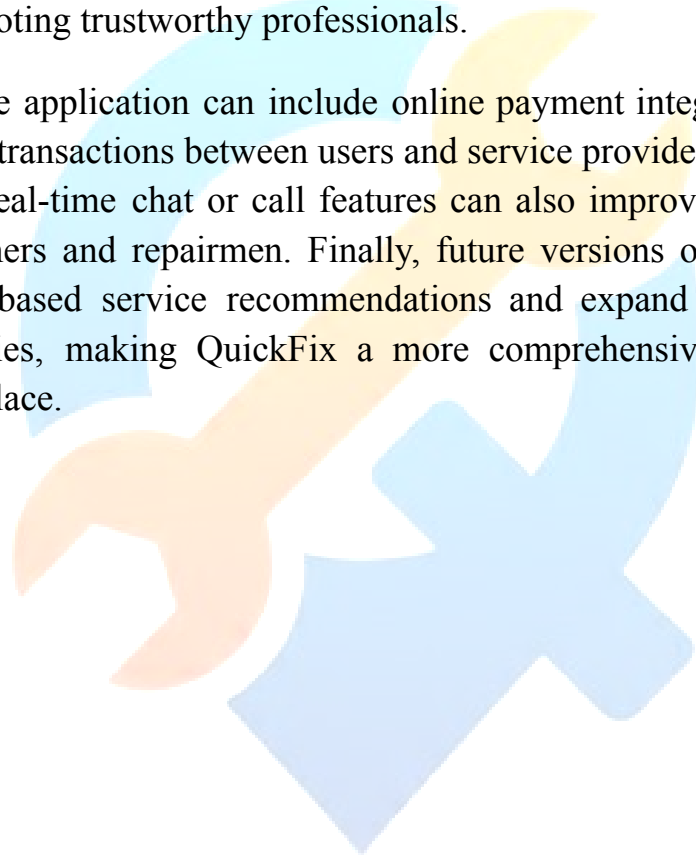


7.2 Recommendation

The QuickFix Application successfully addresses the common problem of finding reliable repair services by providing a centralized digital platform for both users and repair professionals. Through the use of modern technologies such as Flutter for mobile development, Node.js for backend services, and Firebase for real-time database management, the system ensures fast, secure, and efficient service connectivity.

To further enhance the functionality of the QuickFix platform, several improvements can be considered. First, integrating a GPS-based location tracking system can help users find the nearest available repairmen quickly and improve service response time. Secondly, implementing a rating and review system would allow users to share their experiences and help maintain service quality by promoting trustworthy professionals.

Additionally, the application can include online payment integration for secure and convenient transactions between users and service providers. Expanding the platform with real-time chat or call features can also improve communication between customers and repairmen. Finally, future versions of the system can incorporate AI-based service recommendations and expand to support more service categories, making QuickFix a more comprehensive and intelligent service marketplace.



08

CHAPTER

● ● | FUTURE SCOPE,
REFERENCE AND APPENDIX



8.1 FUTURE SCOPE

AI-Based Service Recommendation – Use artificial intelligence to recommend the best repairmen based on user location, ratings, and previous service history.

Online Payment Integration – Add secure digital payment options such as UPI, credit/debit cards, and wallets for easy service payments.

Multi-Language Support – Provide support for multiple languages to make the application accessible to users from different regions.

Smart Notification System – Implement push notifications for booking confirmations, service updates, and promotional offers.

Web Version of Platform – Extend the QuickFix platform to a web application so users can book repair services from both mobile and desktop devices.



8.2 REFERENCE

Flutter Official Documentation – <https://flutter.dev>

Node.js Official Documentation – <https://nodejs.org>

Firebase Documentation – <https://firebase.google.com/docs>

Google Material Design Guidelines – <https://material.io>

Various research papers and online resources related to service-based mobile applications and on-demand platforms.



8.3 APPENDIX

The QuickFix Repairmen Finder Application was developed using modern mobile and web development technologies to provide a reliable platform for connecting users with skilled repair professionals. The project includes modules such as user registration, repairmen registration, service selection, profile management, and booking requests. The application interface was designed with a focus on simplicity and usability so that users can easily find and contact repair professionals based on their needs.

The system architecture consists of a Flutter-based mobile frontend, a Node.js backend for handling application logic, and Firebase for database and authentication services. The appendix includes additional materials such as UI design screenshots, workflow diagrams, and system architecture used during the development of the QuickFix application. These supporting materials help in understanding the overall implementation and functioning of the system.

