

HOME SERVICE MANAGEMENT SYSYTEM

Mini Project Report

Submitted by

SABHAMOL P HANEEFA

Reg. No.: AJC23MCA-2053

In Partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS (MCA)



**AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY**

[Approved by AICTE, Accredited by NAAC, Accredited by NBA.
Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2024-2025

DEPARTMENT OF COMPUTER APPLICATIONS
AMAL JYOTHI COLLEGE OF ENGINEERING AUTONOMOUS
KANJIRAPPALLY



CERTIFICATE

This is to certify that the Project report, “**HOME SERVICES MANAGEMENT SYSTEM**” is the bonafide work of **SABHAMOL P HANEEFA (Regno: AJC23MCA2053)** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under **Amal Jyothi College of Engineering Autonomous, Kanjirappally** during the year 2024-25.

Sr. Dr. Elsin Chakkalackal .SH
Internal Guide

Mr.Binumon Joseph
Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose
Head of the Department

DECLARATION

I hereby declare that the project report “**HOME SERVICES MANAGEMENT SYSTEM**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from Amal Jyothi College of Engineering Autonomous during the academic year 2024-2025.

Date: 07/11/2024
KANJIRAPPALLY

SABHAMOL P HANEEFA
Reg: AJC23MCA-2053

ACKNOWLEDGEMENT

First and foremost, I thank God almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completing this project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our Director (Administration) **Rev.Fr. Dr. Roy Abraham Pazhayaparampil** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a great depth of gratitude towards our Head of the Department **Rev. Fr. Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Mr. Binumon Joseph** for his valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Sr. Dr. Elsin Chakkalackal. SH** for her inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

SABHAMOL P HANEEFA

ABSTRACT

The Home Services Management System is a comprehensive platform designed to streamline and enhance the overall experience of booking and managing various home services such as cleaning, plumbing, electrical work, and more. The system consists of three primary modules: User, Admin, and Worker. Each module is equipped with specific functionalities to ensure efficient management and operation of home services. The User module focuses on customer needs, the Admin module handles the management of operations, and the Worker module facilitates worker-related tasks. Additionally, a future enhancement includes the implementation of machine learning technology for advanced analytics and resource optimization. The Home Service Management System is an all-encompassing web-based platform designed to streamline the organization and execution of various home services. This system caters to users, service providers, and administrators, creating a seamless experience for booking, managing, and coordinating services such as plumbing, electrical repairs, cleaning, and interior design.

The platform empowers users to easily search, book, and manage home service appointments. Users can browse service categories, read provider reviews, view ratings, and book services at their convenience. Once a service is booked, users can track the progress of their service requests, make payments, and provide feedback on the service received. Additionally, a messaging system enables direct communication with service providers, fostering transparency and allowing users to clarify service details or reschedule appointments if needed.

Service providers have their own dedicated interface where they can manage their service offerings, respond to service requests, and update their availability. Providers can showcase their skills, respond to user inquiries, and update the status of ongoing service appointments, ensuring efficient communication and better client satisfaction. The system also enables providers to build credibility through user reviews and ratings, helping them attract more clients.

The Learning Platform offers modules for service-specific skill training and certification, providing service providers with an opportunity for professional development. Providers can also attend webinars and watch instructional videos to improve their skills and expand their service offerings.

Administrators have full access to manage users, providers, and service categories. They can approve provider registrations, monitor service requests, and handle disputes if any arise. The platform provides analytics and reporting tools for administrators to track platform usage, user satisfaction, and provider performance, enabling continuous improvement.

Overall, the Home Service Management System brings convenience, reliability, and efficiency to the home service industry by facilitating communication, providing quality assurance, and promoting trust among users and service providers. This platform is built to adapt and grow with changing user needs, aiming to become an essential tool in the modern home service landscape.

CONTENT

SL. NO	TOPIC	PAGE NO
1	INTRODUCTION	1
1.1	PROJECT OVERVIEW	2
1.2	PROJECT SPECIFICATION	3
2	SYSTEM STUDY	4
2.1	INTRODUCTION	5
2.2	EXISTING SYSTEM	6
2.3	DRAWBACKS OF EXISTING SYSTEM	8
2.4	PROPOSED SYSTEM	9
2.5	ADVANTAGES OF PROPOSED SYSTEM	11
3	REQUIREMENT ANALYSIS	13
3.1	FEASIBILITY STUDY	14
3.1.1	ECONOMICAL FEASIBILITY	14
3.1.2	TECHNICAL FEASIBILITY	15
3.1.3	BEHAVIORAL FEASIBILITY	15
3.1.4	FEASIBILITY STUDY QUESTIONNAIRE	16
3.2	SYSTEM SPECIFICATION	17
3.2.1	HARDWARE SPECIFICATION	17
3.2.2	SOFTWARE SPECIFICATION	17
3.3	SOFTWARE DESCRIPTION	17
3.3.1	PHP	18
3.3.2	MYSQL	18
4	SYSTEM DESIGN	19
4.1	INTRODUCTION	20
4.2	UML DIAGRAM	21
4.2.1	USE CASE DIAGRAM	21
4.2.2	SEQUENCE DIAGRAM	22
4.2.3	STATE CHART DIAGRAM	23
4.2.4	ACTIVITY DIAGRAM	24

4.2.5	CLASS DIAGRAM	25
4.2.6	OBJECT DIAGRAM	26
4.2.7	COMPONENT DIAGRAM	27
4.2.8	DEPLOYMENT DIAGRAM	28
4.2.9	COLLABORATION DIAGRAM	28
4.3	USER INTERFACE DESIGN USING FIGMA	29
4.4	DATABASE DESIGN	33
5	SYSTEM TESTING	38
5.1	INTRODUCTION	39
5.2	TEST PLAN	39
5.2.1	UNIT TESTING	39
5.2.2	INTEGRATION TESTING	40
5.2.3	VALIDATION TESTING	40
5.2.4	USER ACCEPTANCE TESTING	41
5.2.5	AUTOMATION TESTING	41
5.2.6	SELENIUM TESTING	42
6	IMPLEMENTATION	51
6.1	INTRODUCTION	52
6.2	IMPLEMENTATION PROCEDURE	52
6.2.1	USER TRAINING	52
6.2.2	TRAINING ON APPLICATION SOFTWARE	53
6.2.3	SYSTEM MAINTENANCE	54
7	CONCLUSION & FUTURE SCOPE	58
7.1	CONCLUSION	59
7.2	FUTURE SCOPE	60
8	BIBLIOGRAPHY	60
9	APPENDIX	64
9.1	SAMPLE CODE	65
9.2	SCREEN SHOTS	83

List of Abbreviations

- UML - Unified Modelling Language
- ORM - Object-Relational Mapping
- MVT - Model-View-Template
- MVC - Model-View-Controller
- RDBMS - Relational Database Management System
- 1NF - First Normal Form
- 2NF - Second Normal Form
- 3NF - Third Normal Form
- IDE - Integrated Development Environment
- HTML - HyperText Markup Language
- JS - JavaScript
- CSS - Cascading Style Sheets
- API - Application Programming Interface
- UI - User Interface
- HTTP - Hypertext Transfer Protocol
- URL - Uniform Resource Locator
- PK - Primary Key
- FK - Foreign Key
- SQL - Structured Query Language
- CRUD - Create, Read, Update, Delete

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The Home Services Management System is an integrated platform designed to streamline the management, booking, and delivery of various home services like cleaning, plumbing, electrical work, and more. It aims to enhance both customer experience and operational efficiency for home service providers through three core modules: User, Admin, and Worker. Each module serves a specific purpose to optimize the interaction and management of the services offered.

Built using the Django framework for the back-end and HTML/CSS for the front-end, the system ensures scalability, security, and ease of use. The admin dashboard allows system operators to monitor performance, manage users, and generate reports. The platform's architecture supports key features like service request management, scheduling, payment processing, and feedback collection, ultimately delivering a seamless and efficient experience for both customers and service providers.

Key User Groups:

1. **Users:** Individuals who require home services like cleaning, repairs, or installations.
2. **Service Providers (Workers):** Professionals who offer home services and use the platform to manage service requests, update task status, and receive payments.
3. **Administrators:** System operators who oversee platform functionality, manage users and service providers, and generate performance analytics.

The Home Services Management System is designed to improve operational efficiency, enhance customer experience, and provide a comprehensive solution for managing workers and services. With planned enhancements like machine learning for analytics and resource optimization, the platform will continue to evolve, offering a modern, scalable solution to home service management challenges.

1.2 PROJECT SPECIFICATION

The Home Services Management System is a web-based platform designed to streamline the booking and management of various home services, such as cleaning, plumbing, and electrical repairs. The system allows users to easily browse services, book appointments, track service progress in real-time, and make secure digital payments. It aims to improve communication between customers and service providers, thereby enhancing service quality and operational efficiency.

Platform & Technology Stack:

- Back-end: Django (Python)
- Front-end: HTML, CSS, JavaScript
- Database: MySQL (to manage user, worker, and service data)
- User Registration and Authentication:
Secure account creation and login for customers and service providers.
- Service Request Management:
Users can browse available services, book appointments, and track the progress of their requests in real-time.
- Service Provider Management:
- Workers can manage their availability, update task statuses, and respond to new service requests.
- Real-time Service Tracking:
- Users can monitor the status of their booked services and track worker locations during service delivery.
- Admin Dashboard:
- Admins have full control over user and worker profiles, services, financial transactions, and can generate system performance reports.
- Target Audience: Homeowners seeking reliable services for tasks like cleaning, plumbing, and repairs.
- Service providers offering specialized home services.
- Administrators managing platform operations and performance analytics.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System study is a critical phase in the development of any project, involving a thorough analysis of the existing processes, identifying inefficiencies, and designing a new system to resolve these issues. In the context of the Home Services Management System, the system study focuses on evaluating current methods of managing home services like cleaning, plumbing, and electrical work, identifying pain points, and proposing a modern, technology-driven platform to improve the overall experience for users, workers, and administrators.

The current home service industry faces several challenges. Traditionally, service bookings and task management are conducted manually, often through phone calls or basic online forms, which can lead to delays, miscommunication, and inefficient scheduling. There is little or no integration of real-time service tracking, no automation in dispatching workers, and limited visibility into customer satisfaction and service quality. Payments are often managed manually, with delays in processing, and customer feedback mechanisms are underdeveloped, making it difficult to ensure consistent service quality.

The system study aims to analyze these gaps in the existing process and design a unified platform that automates and streamlines the management of home services. By integrating modern features such as real-time service tracking, automated worker dispatch, secure digital payments, and data-driven analytics, the proposed system enhances efficiency, reduces errors, and improves overall customer satisfaction.

Additionally, the study emphasizes the importance of a well-structured User, Admin, and Worker module to address the needs of all stakeholders involved. The User module allows customers to browse services, schedule appointments, track service progress, and provide feedback, while the Worker module focuses on task assignment, attendance tracking, and payroll management. The Admin module oversees operations, including service management, financial oversight, reporting, and complaint resolution.

Through this system study, the Home Services Management System is designed to replace manual, disorganized processes with a technology-driven solution that ensures a seamless, efficient, and scalable platform for managing home services, while also laying the foundation for future growth and advanced functionalities like machine learning-based analytics.

2.2 EXISTING SYSTEM

Many home service businesses currently rely on fragmented systems or manual processes, leading to inefficiencies in service booking, management, and communication. Users face difficulties in finding reliable service providers, making appointments, and tracking service status. Service providers struggle with managing their schedules, tracking payments, and communicating effectively with clients.

Administrators lack the tools needed to oversee operations efficiently, resulting in delays, errors, and decreased service quality

2.2.1 NATURAL SYSTEM STUDIED

The Home Services Management System aims to revolutionize the way home services, such as cleaning, plumbing, and electrical work, are managed and delivered. Currently, the process for booking these services is largely informal and fragmented. Customers often rely on word-of-mouth referrals, local advertisements, or social media to find service providers, leading to uncertainty about reliability and quality. This traditional method not only creates inefficiencies but also leaves room for miscommunication between customers and service providers. One of the primary challenges in the existing system is the lack of a streamlined booking process. Customers frequently face difficulties in scheduling appointments, often resulting in overbookings or missed appointments. Additionally, payment handling is mostly done through cash or checks, which can lead to delays and potential disputes over transactions. There is also minimal tracking of service quality, making it difficult for customers to provide feedback or for service providers to improve their offerings.

Moreover, service providers often struggle with managing their tasks and availability. Without a centralized system, they may find it challenging to keep track of incoming requests, leading to inefficiencies in resource allocation and time management. This lack of organization not only affects the service providers' productivity but also impacts the overall customer experience. The proposed Home Services Management System seeks to address these challenges by creating a digital platform that streamlines the entire process. By integrating functionalities such as realtime service tracking, secure online payments, and automated scheduling, the system aims to enhance efficiency for both customers and service providers. Additionally, implementing features that allow for customer feedback and ratings will promote accountability and encourage service improvement.

2.2.2 DESIGNED SYSTEM STUDIED

The Home Services Management System is an integrated, technology-based platform developed to optimize and enhance the experience of booking and managing various home services, such as cleaning, plumbing, and electrical work. This system is designed to address the inefficiencies of traditional home service management through the incorporation of modern technologies, including user-friendly interfaces, real-time tracking, secure payments, and automation.

Key Components of the Designed System:

Automated Service Request: Users can easily request home services through a web-based platform or mobile app, eliminating the need for phone calls and reducing the likelihood of miscommunication. Customers can select the type of service required and provide detailed descriptions of their needs through an intuitive interface.

Service Browsing: The system offers an extensive catalog of services available for users to explore, complete with detailed descriptions and pricing information. This allows customers to make informed decisions when selecting the services they need.

Booking and Scheduling: Once a service is selected, users can book and schedule appointments directly through the platform, choosing times that best suit their availability. This streamlined process enhances convenience and efficiency.

Payment Gateway: The system integrates secure digital payment options, allowing users to complete transactions online without the need for cash. This ensures a smooth payment process and increases overall user satisfaction.

Service Tracking: Users can track the status of their service requests in real-time, providing transparency and reducing uncertainty. They receive notifications about the progress of their requests, including estimated arrival times.

Feedback and Ratings: After each service is completed, users can rate the service provider and provide feedback on their experience. This feature helps maintain high service quality and encourages providers to improve their performance.

Admin Dashboard: Administrators have access to a centralized dashboard where they can manage all aspects of the system. This includes monitoring user activity, overseeing service provider performance, and generating reports on key metrics such as response times and customer satisfaction.

User and Worker Management: The system allows admins to view and manage profiles for both users and service providers, ensuring that all parties are accurately represented and properly managed within the system.

Complaint and Dispute Resolution: The platform includes a mechanism for logging and tracking user complaints and disputes, ensuring that issues are managed and resolved effectively. **Analytics and Reporting:** Admins can access detailed analytics and reporting tools to evaluate system performance, service quality, and user satisfaction, facilitating informed decision-making for future improvements.

Automated Refunding: In the event of canceled services, the system processes refunds automatically through integration with payment gateways, providing a seamless experience for users.

Future Enhancements: Plans for incorporating machine learning technology will further enhance the system's capabilities. This includes advanced analytics for predicting service demand, optimizing resource allocation, and improving overall operational efficiency.

Conclusion

The Home Services Management System is designed to significantly improve operational efficiency, enhance customer experience, and streamline worker management. By integrating these key functionalities into a cohesive platform, the system effectively addresses the challenges faced by home service management companies and positions itself for future advancements through technology integration.

2.3 DRAWBACKS OF EXISTING SYSTEM

- **Inefficient Communication:** Reliance on phone calls can lead to miscommunication, causing delays in service.
- **Limited Tracking:** Users lack real-time updates on service status, leading to uncertainty about arrival times.
- **Manual Booking:** Booking and scheduling services manually increase the risk of conflicts and missed appointments.
- **Inconsistent Quality:** Variability in service quality and pricing across providers can confuse users.
- **Cumbersome Payments:** Outdated payment methods slow down transactions and increase errors.
- **Weak Feedback Mechanisms:** Limited options for user feedback hinder service improvement and satisfaction.

- **Lack of Analytics:** Insufficient data analysis restricts performance tracking and informed decision-making.
- **Poor User Experience:** Outdated interfaces frustrate users and reduce overall engagement.
- **Limited Service Options:** A narrow range of services restricts user choices and convenience.
- **High Operational Costs:** Manual processes raise costs for service providers, potentially increasing prices for users.
- **Inefficient Resource Use:** Poor allocation leads to underutilized providers or overbooking.
- **Slow Issue Resolution:** Ineffective complaint handling results in unresolved problems and customer dissatisfaction.

2.4 PROPOSED SYSTEM

The proposed Home Services Management System aims to enhance the current functionalities and address the limitations of existing systems. By integrating advanced technology and user-centric design, the system will provide a seamless experience for users, administrators, and service workers. The proposed system will include enhanced modules, improved user interfaces, and future-ready technology features.

Proposed Modules and Functionalities

User Module

- **Registration and Login** ◦ **Enhanced Security:** Implement two-factor authentication for secure account creation and login.
- **Service Browsing**
- **Personalized Recommendations:** Use algorithms to suggest services based on user history and preferences.
- **Booking and Scheduling** ◦ **Smart Scheduling:** Utilize AI to recommend optimal booking times based on user and worker availability.
- **Payment Gateway** ◦ **Multiple Payment Options:** Offer various payment methods, including credit/debit cards, digital wallets, and in-app purchases.
- **Service Tracking** ◦ **Real-Time GPS Tracking:** Enable users to view the live location of service providers and estimated arrival times on a map.
- **Feedback and Rating** ◦ **Instant Feedback:** Allow users to provide feedback immediately after service completion through the app.
- **Notifications** ◦ **Enhanced Communication:** Use push notifications, emails, and SMS to keep users informed about all aspects of their bookings and promotions.

Admin Module

- User and Worker Management
 - Comprehensive Profiles: Maintain detailed profiles with performance metrics, user history, and feedback.
- Service Management
 - Dynamic Service Catalog: Add, update, or remove services based on market demand and user feedback.
- Scheduling and Dispatch
 - Automated Dispatch System: Use AI algorithms to match service requests with the best available workers.
- Financial Management
 - Integrated Accounting Tools: Provide tools for tracking expenses, revenues, and profitability.
- Automated Refunding
 - Instant Refund Processing: Improve the refund process to ensure immediate resolution of cancellations or service disruptions.
- Analytics and Reporting
 - Real-Time Analytics Dashboard: Access real-time data visualizations on system performance, service quality, and user engagement.
- Complaint and Dispute Resolution
 - Efficient Tracking System: Implement a ticketing system for logging and resolving user complaints efficiently.
- Service History
 - Detailed Records: Maintain comprehensive records of service requests, outcomes, and customer preferences for analytics.
- Workers Module
- Profile Management
 - Skill Development Tracking: Track and display workers' certifications, trainings, and areas of expertise.
- Task Assignment : Smart Assignment: Use algorithms to assign tasks based on proximity, skills, and past performance.
- Task Management : Integrated Reporting Tools: Workers can easily report task status, issues, and updates through the system.
- Attendance and Payroll: Automated Payroll Calculations: Integrate attendance tracking with payroll to automate payment processing.
- Communication : Integrated Messaging: Allow direct communication between workers, admin, and users for effective coordination.
- Incident Reporting : Immediate Alerts: Implement a system for real-time incident reporting with automatic notifications to admins.
- Leave Requests : Self-Service Leave Management: Enable workers to submit and track leave requests easily within the system.
- Payment Tracking
 - Detailed Payment History: Provide workers access to their earnings, including detailed transaction records.
- Feedback and Ratings : Performance Tracking: Use user feedback to create performance profiles for continuous improvement.
- Training and Certification: Automated Reminders: Notify workers about upcoming training sessions and certification renewals.

Future Enhancements

- Machine Learning for Advanced Analytics: Utilize machine learning algorithms to analyze data patterns, predict service demand, and optimize resource allocation.
- Promotions and Discounts Module: Manage targeted promotional offers based on user behavior and preferences to drive engagement.
- Loyalty Program Module:
Introduce a loyalty program to reward repeat customers, encouraging longterm relationships.
- AI Chatbot for Customer Support: Implement an AI-driven chatbot to assist users with common queries and support requests 24/7.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Improved User Experience
- Streamlined booking and tracking features enhance customer satisfaction by making it easier for users to access services.
- Increased Efficiency
- Automation reduces administrative tasks, allowing the team to focus on strategic operations and improve overall workflow.
- Enhanced Worker Productivity
- Smart task assignment optimizes worker schedules, improving time management and enabling workers to complete more tasks efficiently.
- Comprehensive Analytics
- Data-driven insights on performance and service quality aid decision-making, helping to identify areas for improvement and growth.
- Scalability
- The modular design accommodates growth and the addition of new services easily, allowing the system to evolve with changing business needs.
- Real-Time Notifications
- Timely updates on service status enhance communication and user engagement, keeping customers informed throughout the service process.
- Enhanced Security

- Secure processes protect user data and financial transactions from threats, ensuring trust and safety for all users.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility study evaluates the technical, economic, and behavioral viability of the proposed Home Services Management System, ensuring that it can be developed and implemented effectively while meeting organizational goals. The feasibility study for the proposed Home Services Management System assesses its technical, economic, and behavioral viability to ensure successful development and implementation aligned with organizational goals. It evaluates the required technology stack, including web and mobile platforms, databases, and cloud services, ensuring compatibility for key system features. Additionally, the study examines integration with existing software like payment processors and CRM systems for smooth operations. The estimated costs of development, implementation, and maintenance are compared against anticipated benefits, such as improved efficiency and customer satisfaction, to project ROI and assess funding adequacy. Stakeholder acceptance and usability are gauged through surveys or focus groups, with attention to training needs for effective adoption. The study also considers potential impacts on current workflows and devises strategies for a seamless transition to the new system.

3.1.1 Economical Feasibility

Economic feasibility examines the costs associated with developing, implementing, and maintaining the Home Services Management System, while assessing its potential to generate revenue and return on investment (ROI).

- Initial Costs:

Development costs include expenses for software development using Django for the back-end and HTML/CSS for the front-end. Additional costs encompass hardware (servers, GPS integration), database management (MySQL), and digital payment integration. Ongoing maintenance and support will incur further costs.

- Revenue Generation:

The platform can generate revenue through service fees charged to users for home services, as well as potential subscription models for frequent users. Partnerships with service providers may create additional revenue streams, including fees for service providers wishing to be listed on the platform.

- Return on Investment (ROI):

By enhancing service efficiency and reducing response times, the system is expected to attract more users, resulting in increased service fees. The improved user experience is likely to lead to repeat customers and positive word-of-mouth, ensuring long-term financial

viability. The predicted ROI is positive due to the system's ability to optimize service delivery and enhance user satisfaction.

3.1.2 Technical Feasibility

Technical feasibility assesses whether the proposed system can be built using the available technology and expertise.

- **Technology Availability:**
The system will be developed using Django, a scalable and secure framework, with HTML/CSS for the front end and MySQL for the database. These technologies are widely used and well-supported, capable of fulfilling the project's requirements.
- **GPS Integration:**
Real-time GPS tracking, a core feature of the system, can be achieved through existing APIs like Google Maps. This ensures accurate tracking and dispatching of service providers, effectively reducing response times.
- **Real-Time Communication:**
Technologies such as WebSockets or third-party messaging APIs are readily available to implement real-time updates and can be seamlessly integrated into the system.
- **Scalability:**
The system is designed to scale efficiently as more users and service providers join the platform, ensuring smooth operation even with increasing demand.
- **Security:**
Secure payment gateways and encrypted data transmission will protect users' financial and personal information.

3.1.3 Behavioral Feasibility

Behavioral feasibility assesses whether the target users—home service clients, service providers, and administrators—are likely to accept and adopt the system.

- **User Acceptance:**
Home service clients are expected to readily adopt the system, as it addresses significant pain points like delays, lack of tracking, and manual payment processes. The convenience of real-time updates, digital payments, and automated service requests aligns with modern user expectations.
- **Service Provider Readiness:**

Service providers are likely to embrace the system due to its capability to simplify service request management, facilitate service history tracking, and streamline payment processes. The platform offers visibility and potential business growth through positive ratings and feedback.

- **Training and Support:**

Minimal training will be required for users and service providers, as the system features a user-friendly interface. However, support resources will be available to ensure smooth adoption and address any issues that may arise.

- **Cultural and Behavioral Factors:**

With increasing reliance on technology for convenience and service optimization, users are likely to adapt well to the digital platform. This trend mirrors shifts in other service industries, such as ride-sharing and food delivery, where similar platforms are widely accepted.

3.1.4 Feasibility Study Questionnaire

- What are the most common types of home services requested by users?
Services like cleaning, plumbing, electrical repairs, gardening, and appliance installation.
- How quickly do users expect service providers to respond after a service request is made?
Users typically expect a response within 15 to 30 minutes, depending on the type of service and provider availability.
- What information should be collected from users when they request home services?
User's name, contact information, service type, location (address), preferred scheduling time, and any specific requirements.
- How do users currently request home services, and what challenges do they face?
Users typically request services through phone calls or existing apps. Challenges include long wait times, lack of transparency in service tracking, and inconsistent service quality.
- What features would enhance user experience in a home services management system?
Real-time tracking of service providers, an intuitive interface, quick booking processes, clear communication channels, and notifications about service status and arrival times.
- What key factors influence the choice of a service provider for home services?
Factors such as service provider reputation, response time, availability, service cost, and user reviews.
- How important is real-time tracking of service requests to users?
Very important, as it provides transparency and reduces anxiety, allowing users to better plan their time while awaiting service.
- What are the expectations for service provider response times?
Users expect service providers to confirm the request within 5 to 10 minutes and to arrive on-site within 30 to 60 minutes.

- How can the system ensure the accuracy and reliability of service provider information?
By implementing a verification process during registration, soliciting user feedback and reviews, and regularly updating provider profiles based on performance metrics.
- What types of reporting or analytics would be beneficial for administrators to monitor system performance?
Useful metrics include service request response times, user satisfaction ratings, service provider performance metrics, system uptime, and overall request volume and completion rates

3.1 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - intel Core i3

RAM -8 GB

Hard disk - 256 GB SSD or above

3.2.2 Software Specification

Front End - HTML5, CSS3, JavaScript, AJAX, J Query.

Back End - Django (Python Framework).

Database - MySQL

Client on PC - Windows 7 and above.

Technologies used - JavaScript ,HTML5 ,CSS3 ,AJAX . J Query PHP

3.3 SOFTWARE DESCRIPTION

3.3.1 Django (Python Framework)

Django is a high-level Python web framework used for building the back-end of the 247 Roadside Assistance* system. It provides built-in features for database management, user authentication, and secure handling of data. Django's scalability ensures that the platform can handle growing numbers of users and requests without performance issues.

Key Features:

- Provides robust ORM (Object-Relational Mapping) for database operations.
- Built-in user authentication system.

- Secure handling of data and integration with third-party APIs (such as GPS tracking services).

3.3.2 MySQL (Database)

MySQL is the relational database management system used to store and manage data for the 247 Roadside Assistance* project. It stores information such as user profiles, service provider details, service requests, transaction history, and feedback. MySQL is widely used for web applications due to its performance and scalability.

Key Features:

- Supports large-scale data storage and efficient querying.
- Provides indexing for faster data retrieval.
- Allows integration with Django using its built-in ORM.

3.3.3 JavaScript and jQuery (Front-End Interaction)

JavaScript is used for making the web pages interactive by handling client-side operations such as form validation, event handling, and dynamic content updates. jQuery simplifies JavaScript operations by offering easy-to-use functions for handling HTML elements, animations, and asynchronous updates (AJAX).

Key Features:

- Enhances user experience with interactive and dynamic web pages.
- Supports asynchronous operations, reducing page reload times.
- Cross-browser compatibility for consistent performance across different browsers.

3.3.4 HTML5 & CSS3 (Web Page Structuring and Styling)

HTML5 is used for structuring web pages, ensuring proper layout and readability, while CSS3 is used for styling and positioning elements on the web pages. Together, they form the foundation of the front-end design.

Key Features:

- Responsive design to ensure optimal viewing on various devices (desktops, tablets, mobiles).
- CSS3 transitions and animations for better user experience.
- HTML5 semantic tags improve accessibility and SEO.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

System design is a crucial phase in developing the Home Services Management System, as it transforms theoretical concepts and requirements into a practical framework. The goal of this phase is to create a detailed blueprint outlining the system's components and functionalities to efficiently meet project requirements. The design focuses on ensuring scalability, security, and user-friendliness while catering to the needs of users, service providers, and administrators. And system, The system design is divided into two main phases: The High-Level Design (HLD) for the Home Services Management System outlines the overall architecture, focusing on major components and their interactions. It includes the User Interface Design, which aims to create intuitive, visually appealing layouts for web pages tailored to users, service providers, and administrators, emphasizing easy navigation, responsive design, and accessibility. The System Architecture defines the relationship between the front-end (user interface), back-end (server-side logic), and database (data storage), ensuring seamless communication and efficient request processing. The Module Design covers essential system modules such as User Registration, which allows users to create accounts and manage profiles; Service Request Management, enabling users to book services, schedule appointments, and track requests; Real-Time Tracking, providing live updates on service provider location and estimated arrival times; and Payment Processing, facilitating secure online transactions and refunds.

The Low-Level Design (LLD) delves into the specific implementation of each module, focusing on detailed component design. This includes the Login System, which ensures secure user authentication and session management; the Service Provider Dashboard, allowing service providers to manage tasks, view schedules, and track payments; and the Admin Panel, which enables administrators to oversee system operations, manage users and service providers, and generate reports. The Database Schema Design specifies how the system will store and manage data, covering user profiles, service requests, and transaction histories. Additionally, the Algorithm Design defines key algorithms, such as matching users with the nearest available service providers based on location and service type, calculating estimated arrival times using real-time traffic data, and generating reports.

4.2 UML DIAGRAM

Unified Modeling Language (UML) diagrams are invaluable tools in system design, as they provide a visual representation of the structure, components, and behavior of a software system. For the Home Services Management System, several UML diagrams can effectively model various aspects of the system. Below are the key UML diagrams commonly used in the project:

4.2.1 USE CASE DIAGRAM

A use case diagram provides an overview of the interactions between users (actors) and the system. It highlights the functionalities that each user role can perform.

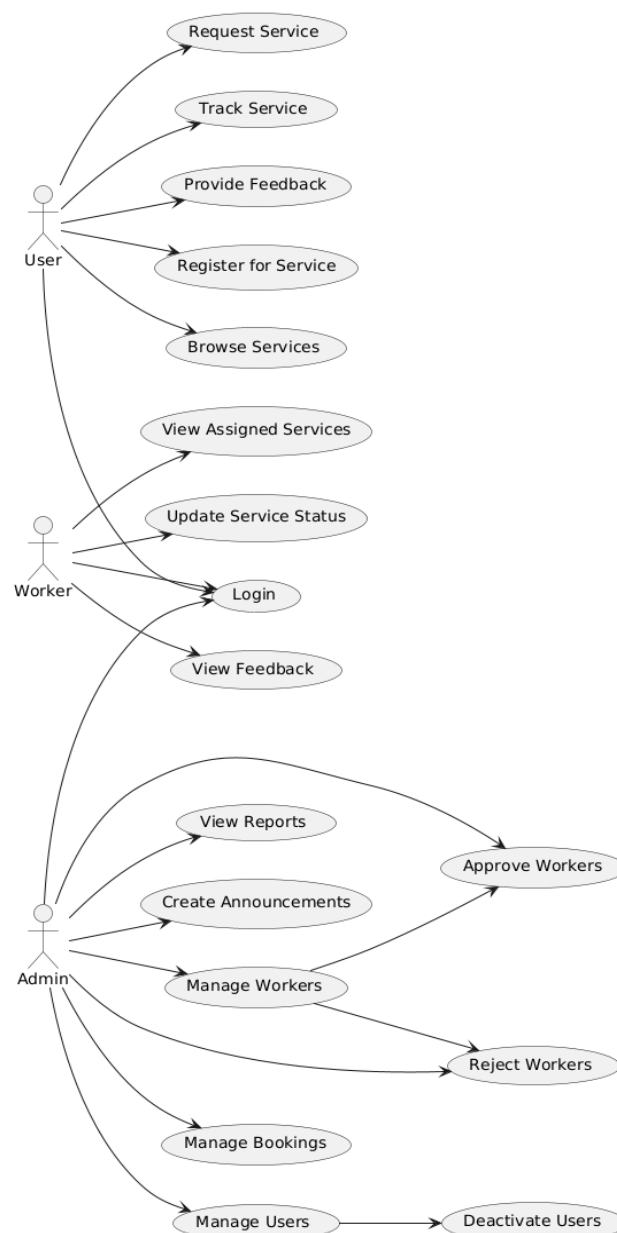


Figure 4.2.1: Use Case Diagram

4.2.1 SEQUENCE DIAGRAM

The sequence diagram shows how objects interact in a particular scenario of a use case.

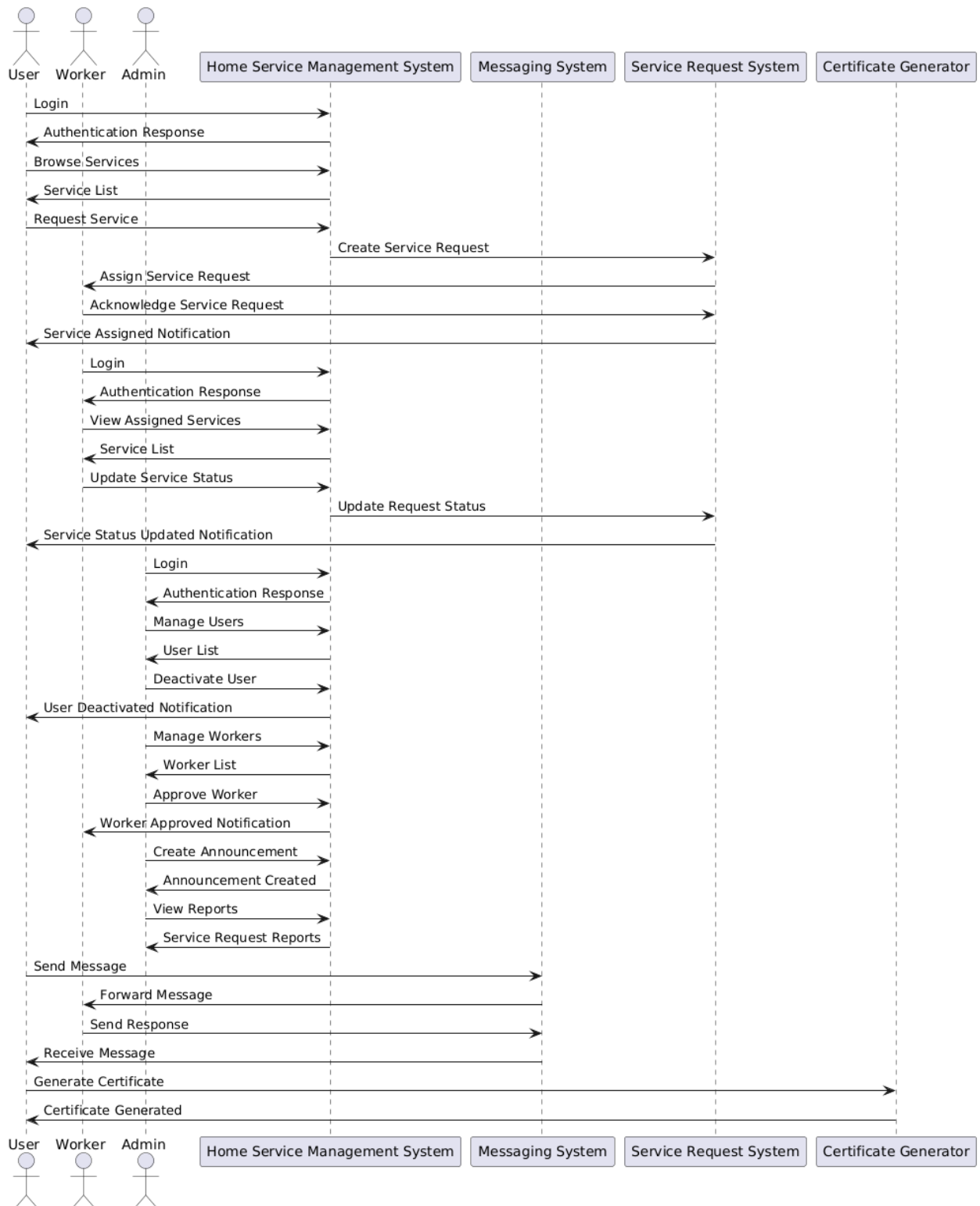


Figure 4.2.2: Sequence Diagram

4.2.2 State Chart Diagram

The state chart diagram shows the state transitions of a particular object.

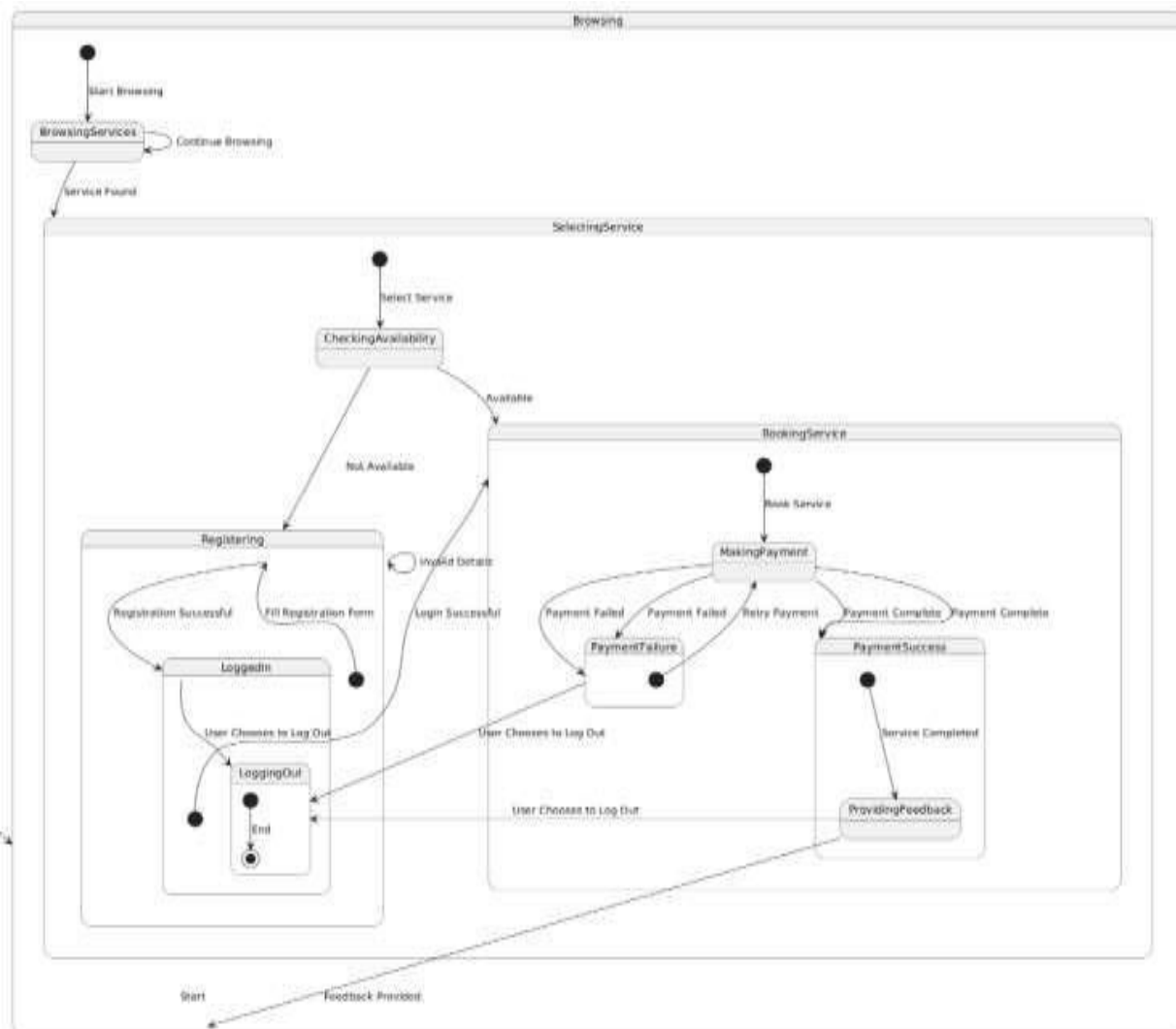


Figure 4.2.2 state chart diagram

4.2.2 Activity Diagram

The activity diagram shows the flow of activities in the system.

User Activity Diagram

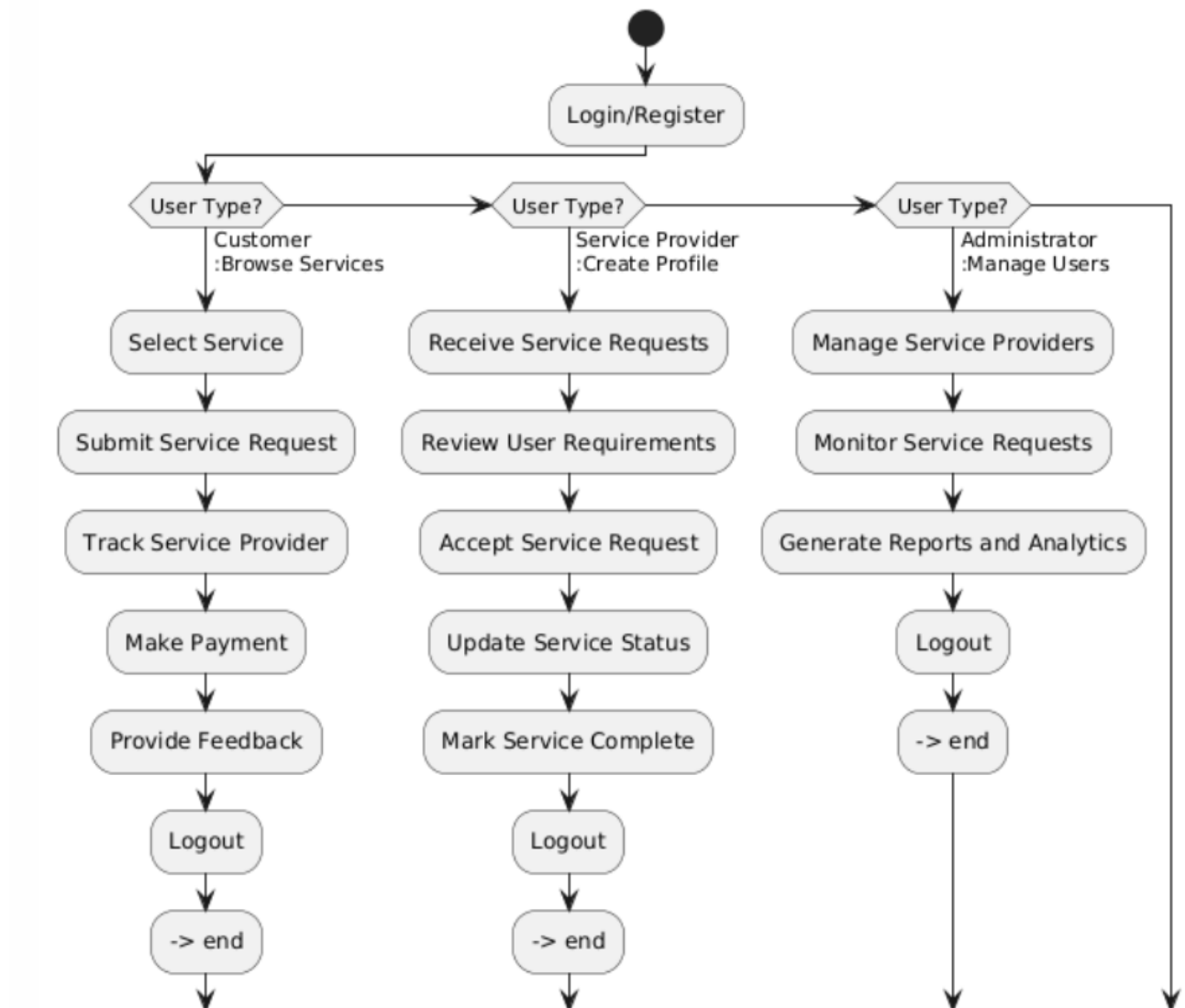


Figure 4.2.4: Activity Diagram

4.2.3 Class Diagram

The class diagram shows the static structure of the DRMS, including its classes, attributes, operations, and relationships

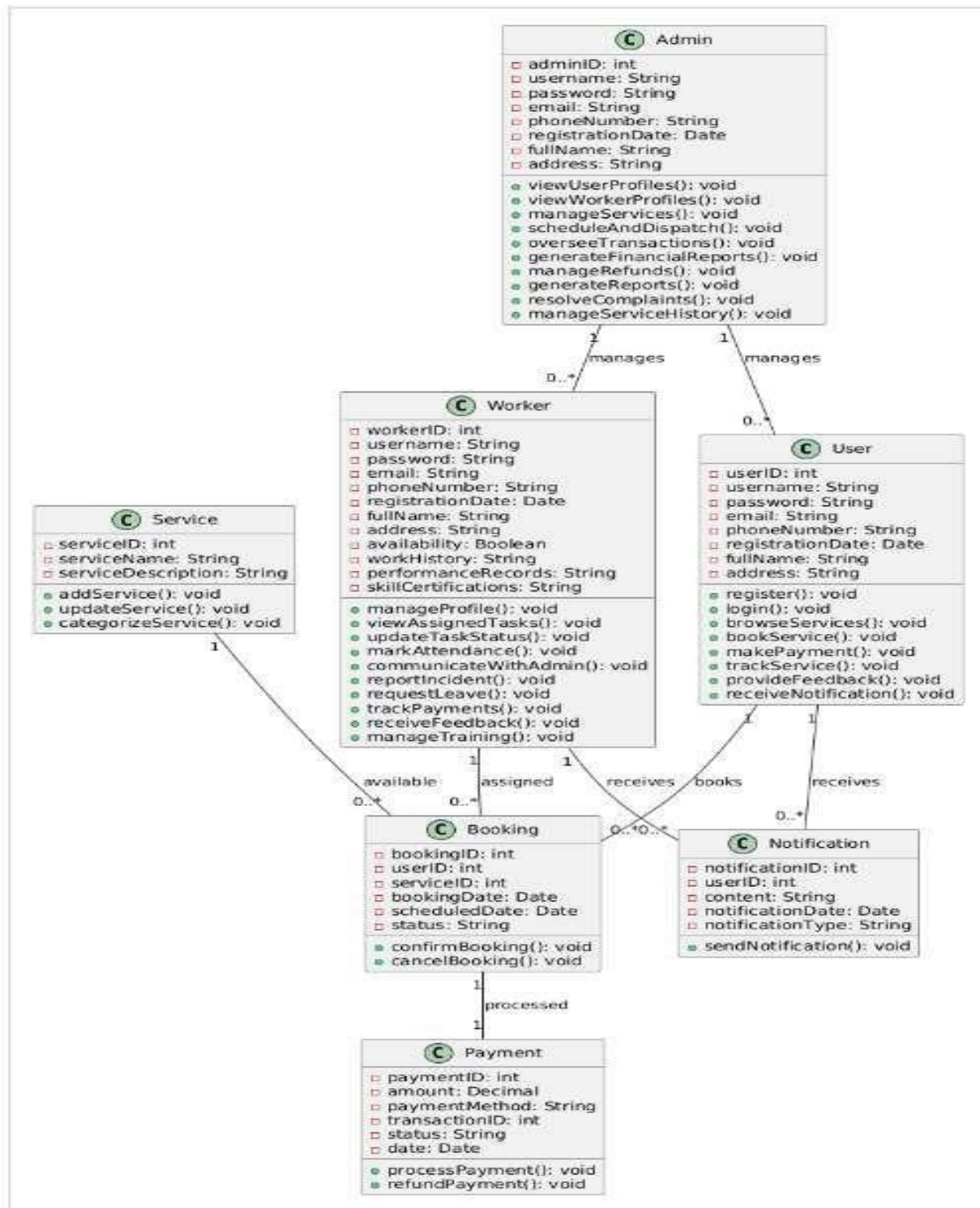


Figure 4.2.5: Class Diagram

4.2.4 Object Diagram

The object diagram shows instances of classes and their relationships at a specific point in time.

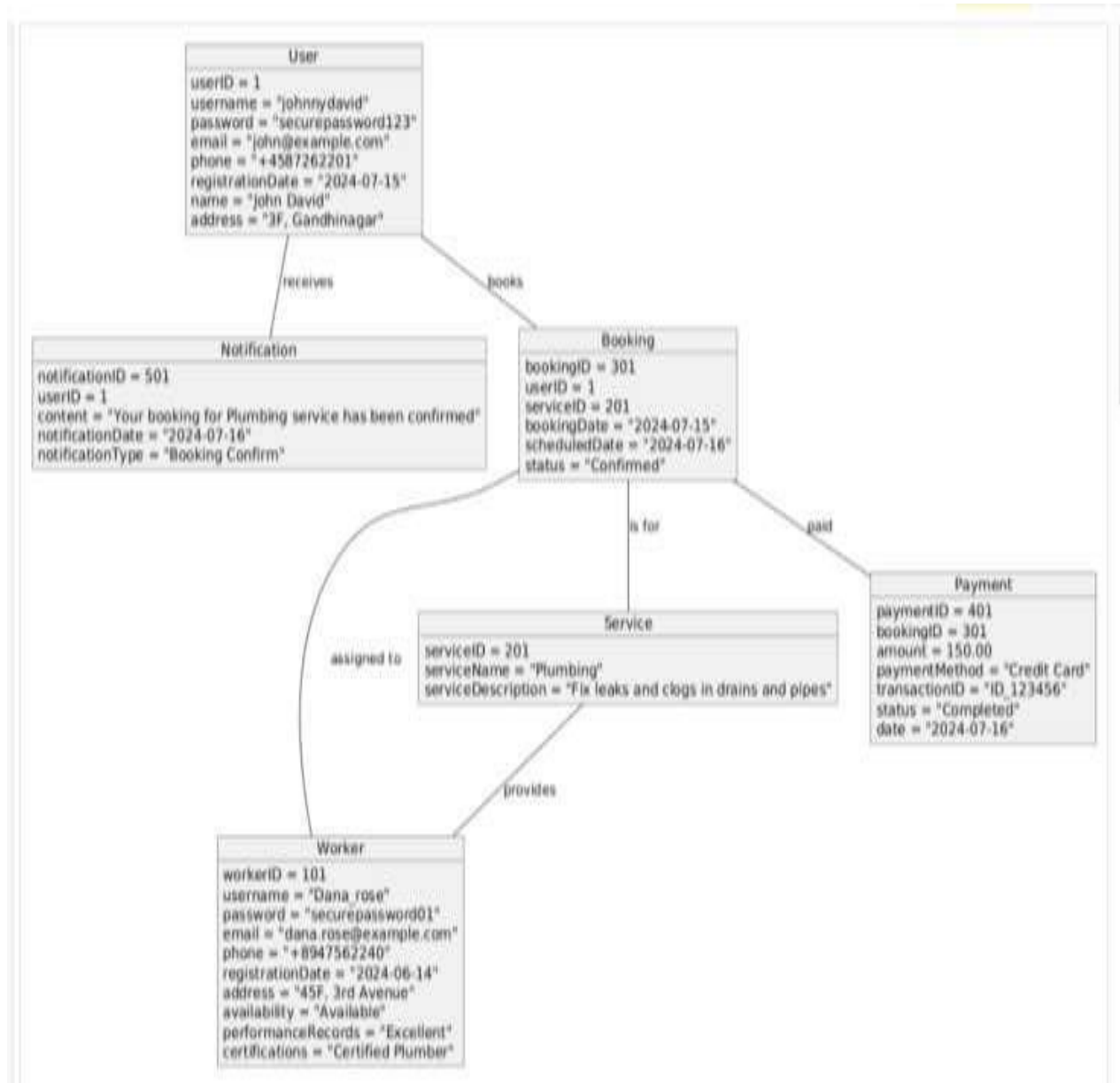


Figure 4.2.6: Object Diagram

4.2.5 Component Diagram

The component diagram illustrates the organization and dependencies among software components.

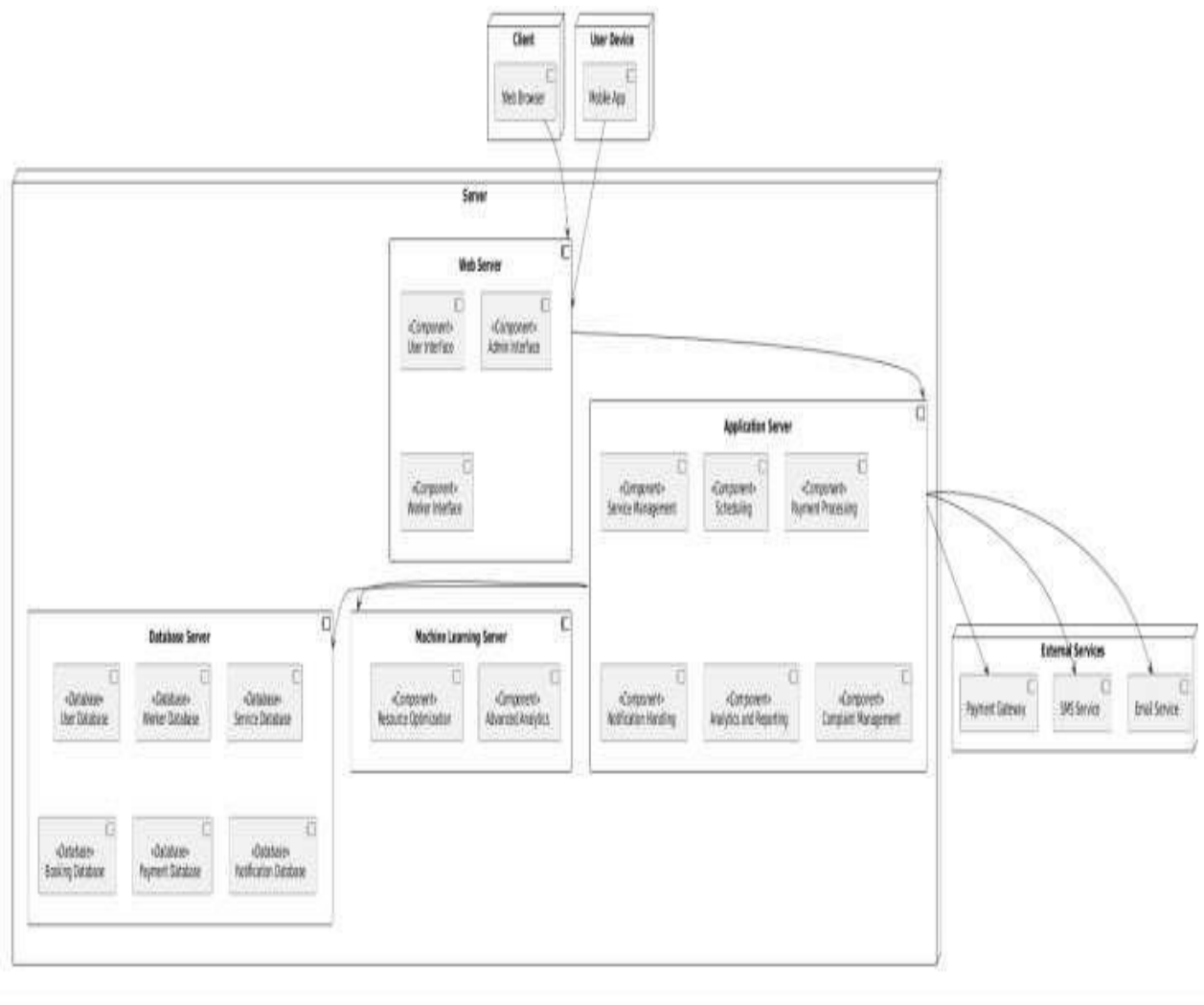


Figure 4.2.7: Component Diagram

4.2.8 Deployment Diagram

The deployment diagram shows the physical architecture of the system, including hardware nodes and software components.

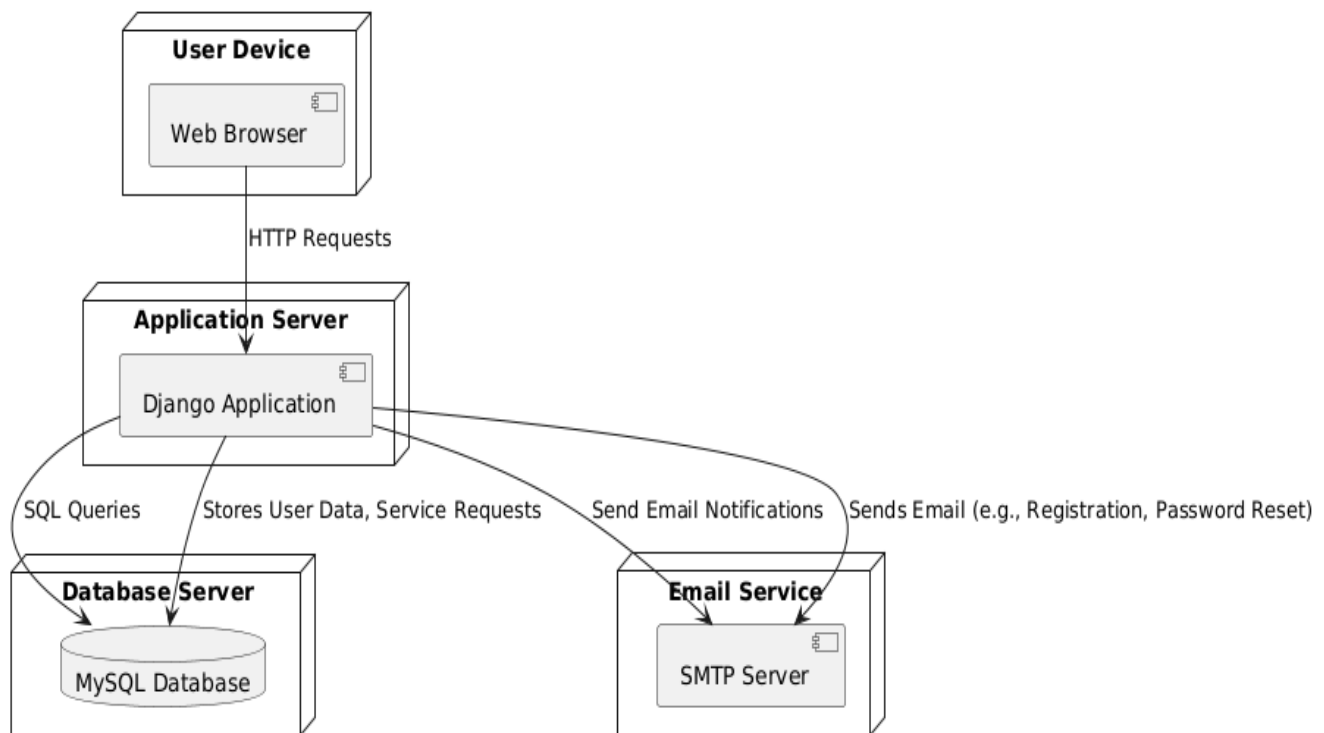


Figure 4.2.8: Deployment Diagram

4.2.9 Collaboration Diagram

The collaboration diagram demonstrates interactions among various objects within the Home Service Management System, emphasizing how users, workers, and administrators collaborate to achieve shared objectives, such as service requests, task assignments, and service fulfillment.

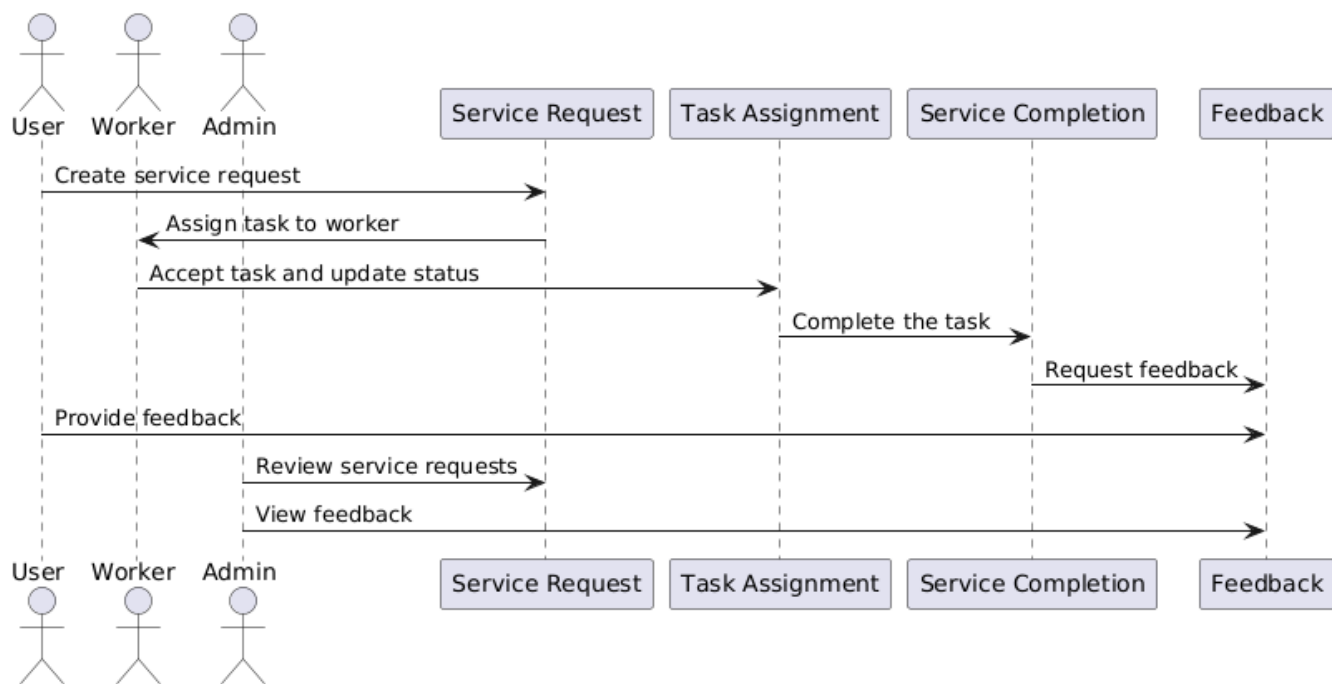


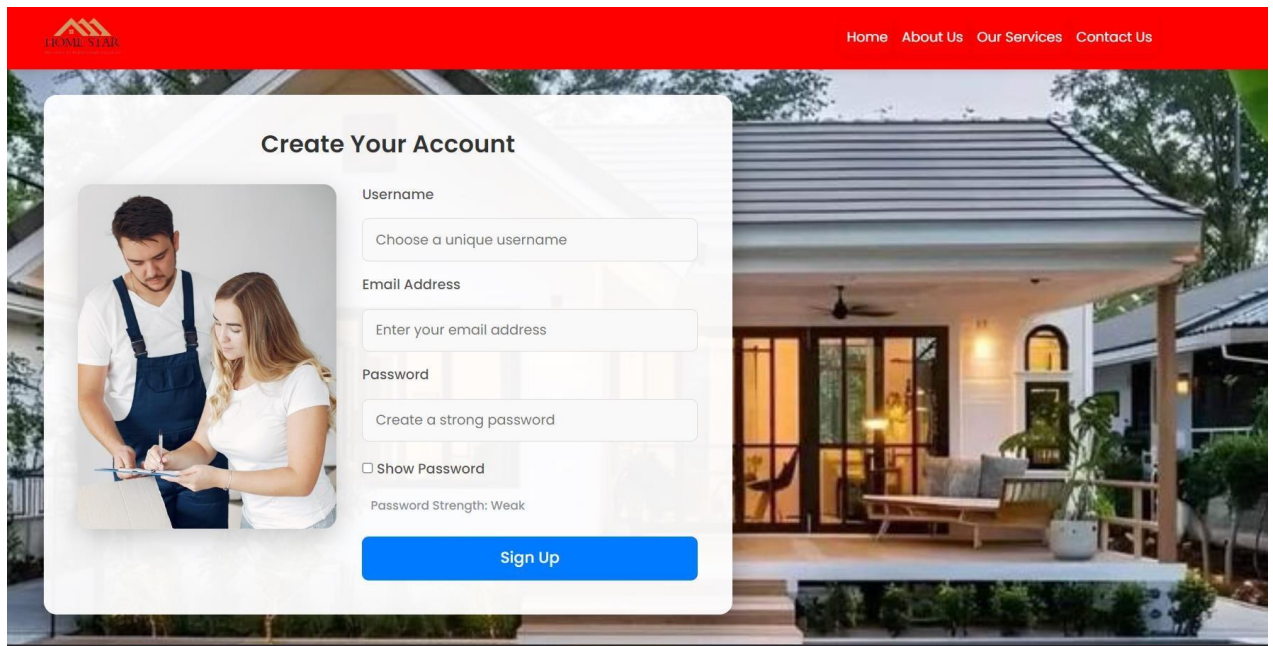
Figure 4.2.9: Collaboration Diagram

4.3 USER INTERFACE DESIGN USING FIGMA

Form Name: Index.html



Figure 4.3.1: Index page

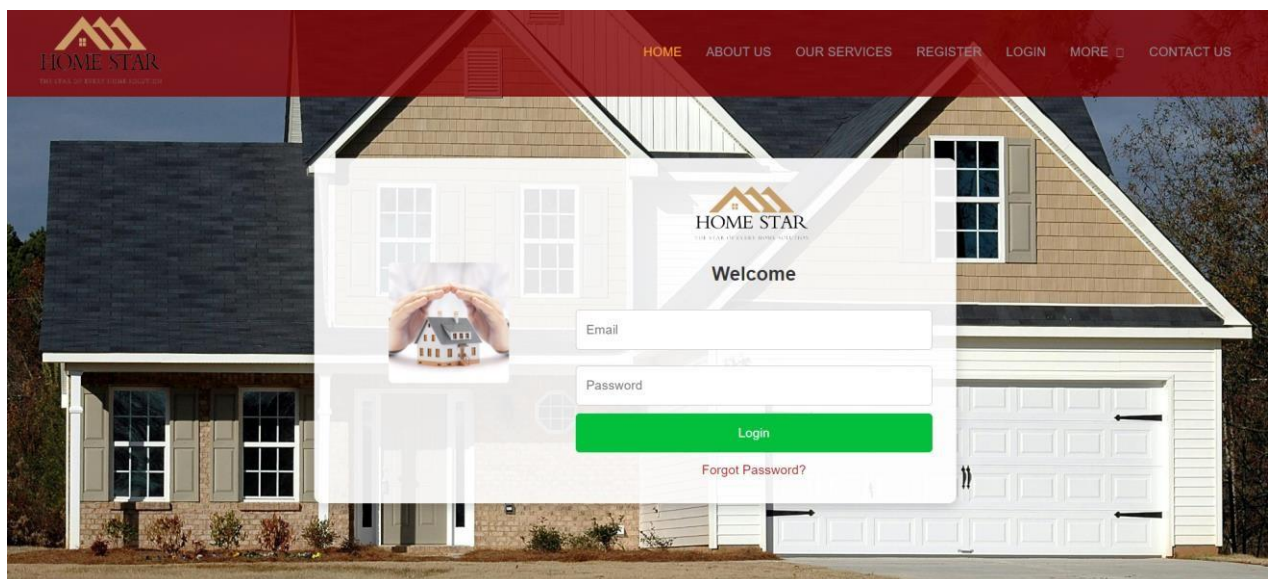
Form Name:User Registration.html

The image shows a user registration form titled "Create Your Account" overlaid on a background image of a modern house. The form includes the following fields and elements:

- Username:** A text input field with the placeholder "Choose a unique username".
- Email Address:** A text input field with the placeholder "Enter your email address".
- Password:** A text input field with the placeholder "Create a strong password".
- Show Password:** A checkbox labeled "Show Password".
- Password Strength:** A label indicating "Password Strength: Weak".
- Sign Up:** A blue button to submit the registration form.

The background image shows a man and a woman looking at a document, and a modern house with a covered porch.

Figure 4.3.2 :User Registration page

4.3.3 Form Design: Login page

The image shows a login form titled "Welcome" overlaid on a background image of a modern house. The form includes the following fields and elements:

- Email:** A text input field.
- Password:** A text input field.
- Login:** A green button to submit the login form.
- Forgot Password?:** A link to reset the password.

The background image shows a modern house with a white garage door. The Home Star logo is visible in the top left corner of the page.

Figure 4.3.3: Login page

4.3.4 Form Design: Booking Form

Book a Service

Booking Date: dd-mm-yyyy

Booking Time: --:--

State: Select State

District: Select District

City:

Street Number:

Phone Number: +91

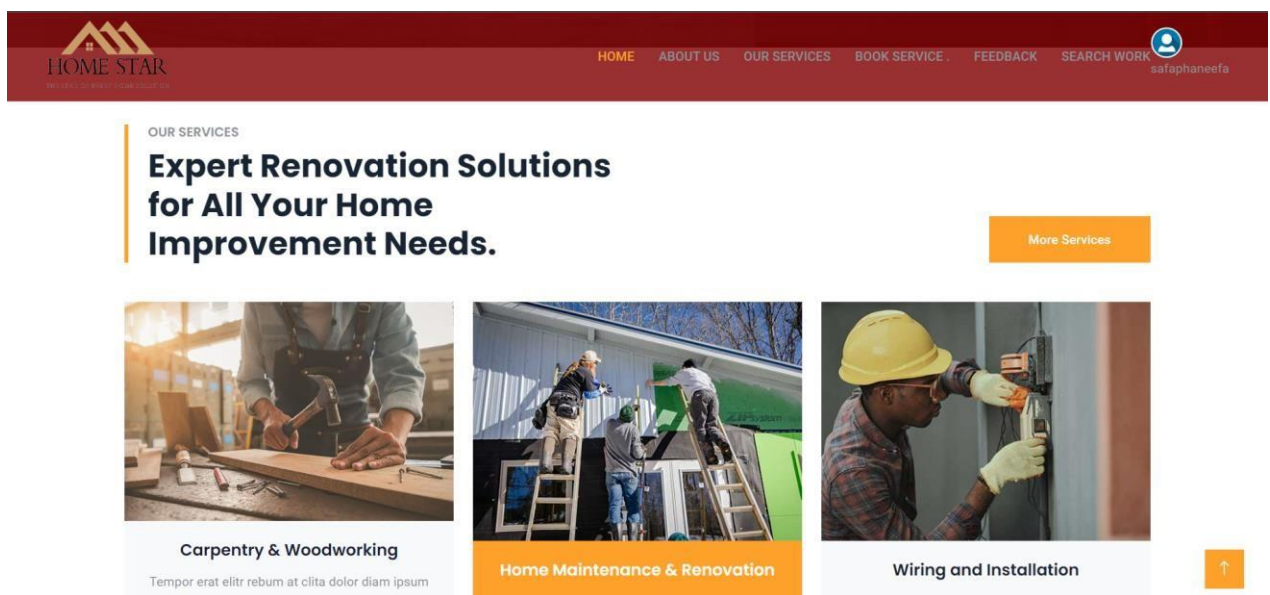
Select Worker Type: Select Worker Type

Address: House Name, Pincode, etc.

Work Description: Describe the work you want done

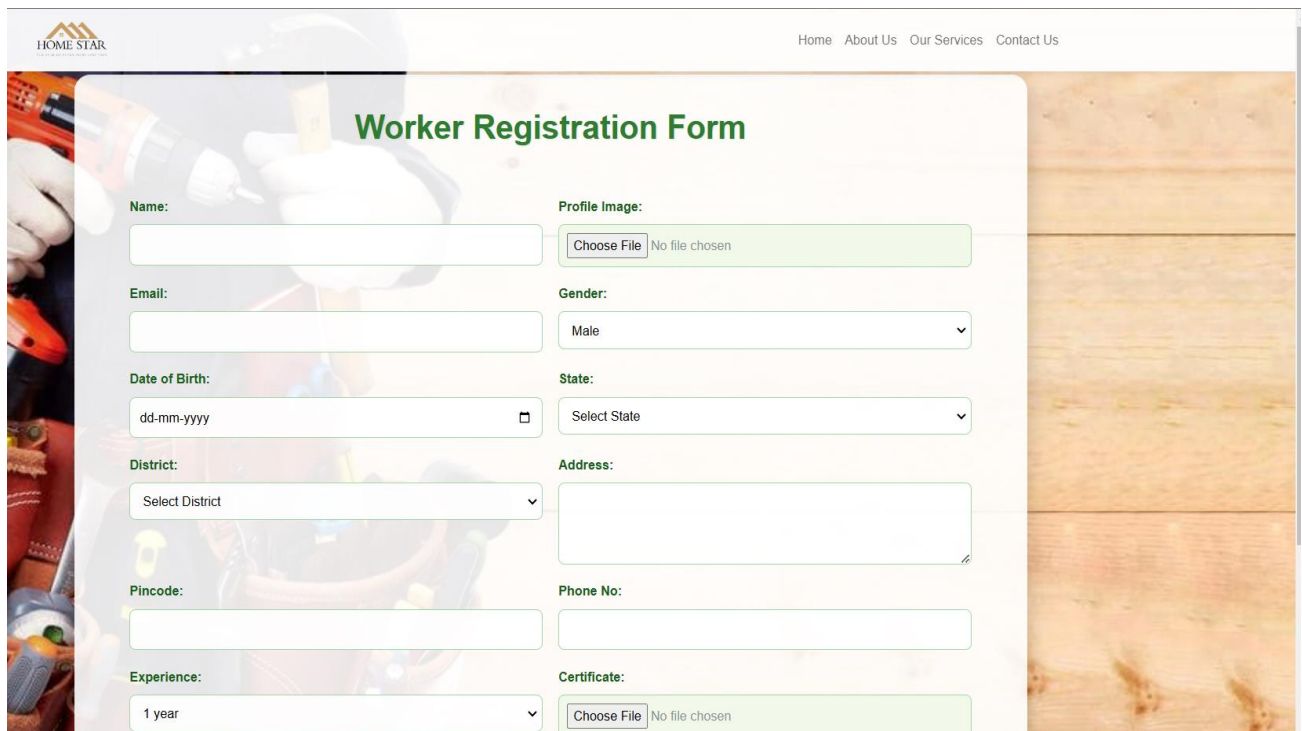
4.3.4 Form Design: Booking Form

4.3.5 Form Design: Service page



4.3.5 Form Design: Service page

4.3.6 Form Design: Worker Registration page



The screenshot displays the 'Worker Registration Form' on the 'HOME STAR' website. The form is titled 'Worker Registration Form' in green text. It includes the following fields and options:

- Name:** A text input field.
- Profile Image:** A file upload button labeled 'Choose File' with the text 'No file chosen'.
- Email:** A text input field.
- Gender:** A dropdown menu with 'Male' selected.
- Date of Birth:** A text input field with the placeholder 'dd-mm-yyyy' and a calendar icon.
- State:** A dropdown menu with 'Select State'.
- District:** A dropdown menu with 'Select District'.
- Address:** A large text input field.
- Pincode:** A text input field.
- Phone No:** A text input field.
- Experience:** A dropdown menu with '1 year' selected.
- Certificate:** A file upload button labeled 'Choose File' with the text 'No file chosen'.

4.3.6 Form Design: Worker Registration page

4.4 DATABASE DESIGN

In the Home Services Management System, the database is designed using a NoSQL approach with MongoDB, providing flexibility to manage various types of service-related data. This structure supports dynamic content such as user profiles, service requests, provider profiles, service history, and interactions like reviews, ratings, and payments. The database is scalable, allowing it to efficiently handle increased user and service provider data as the platform grows.

4.4.1 Relational Database Management System (RDBMS)

In the Home Services Management System, a Relational Database Management System (RDBMS) such as MySQL or PostgreSQL could be used to manage structured data with predefined relationships. An RDBMS provides a structured way to organize data across multiple tables with well-defined relationships, using primary and foreign keys to maintain data integrity and consistency.

This approach is particularly suitable for handling structured data such as user profiles, service categories, service requests, service provider profiles, transaction records, and feedback. The relational model ensures efficient querying, strong data integrity, and ACID compliance, making it ideal for applications requiring complex data relationships and transactions. Using an RDBMS would also simplify data retrieval for reporting and analytics, providing insights into service usage patterns, provider performance, and user satisfaction.

4.4.2 Normalization

While Relational Database Management System (RDBMS) is traditionally used for relational data, the concept of normalization is applied in the Home Services Management System to ensure efficient storage and retrieval of data. In this system, data is organized into multiple tables, with relationships managed through primary keys and foreign keys. For instance, users, service providers, and services are stored in separate tables, but relationships between these entities are maintained using unique IDs as references.

4.4.3 Sanitization

Data sanitization is essential for ensuring the security and integrity of the Home Services Management System. The platform implements strict sanitization processes to validate and clean all user inputs before storing them in the database. This applies to every interaction, such as when users submit service requests, send messages to service providers, or provide feedback.

Sanitization protects the system from common security vulnerabilities, such as SQL injection (a significant threat in RDBMS), XSS (cross-site scripting), and the insertion of malicious data. By ensuring that only valid, clean data is stored, the platform maintains high standards of security and reliability, protecting both user data and system functionality from potential attacks or data corruption. This contributes to a secure and trustworthy environment for users and service providers alike. Sanitization protects the system from common security vulnerabilities, such as SQL injection (a significant threat in RDBMS), XSS (cross-site scripting), and the insertion of malicious data. By ensuring that only valid, clean data is stored, the platform maintains high standards of security and reliability, protecting both user data and system functionality from potential attacks or data corruption. This contributes to a secure and trustworthy environment for users and service providers alike.

4.4.4 Indexing

In the Home Services Management System, MongoDB's indexing capabilities are utilized to optimize query performance, especially for frequently accessed collections such as users, service requests, and service provider profiles. Indexes are created on fields like `userId`, `serviceType`, and `location` to enable faster data retrieval and enhance response times for users.

For example, when users search for specific services or when loading a service provider's profile and related services, the presence of indexes allows MongoDB to efficiently locate the relevant documents without scanning the entire collection. This optimization is particularly valuable as the platform grows, ensuring that performance remains consistent and user experience remains smooth, even as data volume increases.

4.5 TABLE DESIGN**4.5.1. Tbl_users**Eg.Primary key: **loginid**

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique identifier for each user
2	username	CharField (150)		Stores the username of the user
3	email	CharField (150)		email address of the user
4	password	CharField (150)		user's password
5	role	CharField (150)		Stores the role of the user
6	worker_id	CharField (150)	Default='none'	worker if the user is a worker
7	status	CharField (150)	Default='approved'	approval status of the user
8	is_active	BooleanField	Deafult=False	user is active or inactive

4.5.2.Tbl_worker_details

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique if for worker
2	worker_name	CharField (150)		Stores the name
3	email	CharField (150)		email address of the user
4	profile_image	ImageField		worker's profile image
5	gender	CharField (150)		gender of the worker
6	dob	DateField		Date of birth of the worker
7	address	TextField(150)		address of the worker
8	pincode	CharField (150)		Pincode of the worker's
9	experience	CharField (150)		years of experience of the worker
10	certificate	FileField	Null=True, Blank=True	worker's certification documents

Home service management system

11	phone_no	CharField (15)		Phone number
12	services	TextField		Services the worker offers
13	resume	FileField	Null=True, Blank=True	Path to the worker's resume
14	is_active	BooleanField	Default=False	worker is currently active

3.Tbl_bookings

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique identifier for Booking
2	user	ForeignKey	ForeignKey	Links the booking to a user
3	booking_date	DateField		date of the booking user
4	state	CharField (150)		State for the booking address
5	district	CharField (150)		District for the booking address
6	city	CharField (150)	Default='none'	City for the booking address
7	street_number	CharField (150)		Street number for the booking
8	address_line	BooleanField	Null=True, Blank=True	Additional address information
9	booking_time	TimeField		Time of the booking
10	work	CharField (100)		Type of work
11	phone_number	CharField (15)	Null=True, Blank=True	Phone number
12	work_description	TextField		Detailed description of the work
13	created_at	DateTimeField	Auto_add=True	Automatically stores the date and time
14	updated_at	DateTimeField	Auto_add=True	Automatically stores the date and time

Home service management system

4.Tbl_service_details

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique identifier for each user
2	service_image	ImageField	Null=True, Blank=True	image representing the service
3	service_name	CharField (150)		Name of the service
4	description	CharField (150)		description of the service
5	price	DecimalField	Max_digits=10, Decimal_places=2	Price of the service

5.Tbl_feedback

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique identifier for each user
2	username	CharField (150)		Stores the username of the user
3	rating	CharField (150)		Stores rating
4	description	CharField (150)	Null=True, Blank=True	Feedback description
5	created_at	DateField	Auto_add=True	Automatically stores the date and time
6	updated_at	DateField	Auto_add=True	Automatically stores the date and time

7. Tbl_announcements

No:	Field name	Datatype (Size)	Key Constraints	Description of the field
1	id	integer	Primary key	unique identifier announcements
2	title	CharField (150)	Not Null	Title of the announcement
4	password	CharField (150)	Not Null	Content of announcement
5	created_at	DateTimeField	auto_now_add=True	Timestamp for announcement
6	expires_at	DateTimeField	Default='none'	Timestamp announcement will expire

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

Testing is a critical phase in the software development life cycle (SDLC) for the Home Services Management System, aimed at identifying and resolving any bugs or issues before deployment. The testing process ensures that the platform operates as expected, meets the specified requirements, and provides a smooth experience for all user groups, including homeowners, service providers, and administrators. Testing is essential to verify the system's reliability, performance, and security, ensuring that defects are detected and addressed early to reduce the risk of issues post-deployment.

Testing encompasses multiple levels, including unit testing, integration testing, system testing, and user acceptance testing, all of which are essential for maintaining the quality of the platform. Each type of testing has a distinct purpose, ensuring that the platform performs correctly at both the individual component level and as an integrated solution. Automation testing is also employed to speed up the testing process, enabling the project team to uphold high-quality software development practices in an efficient and thorough manner.

5.2 TEST PLAN

A test plan outlines the strategy and scope of the testing process, detailing the types of tests to be executed, the testing environment, testing objectives, and roles and responsibilities. The primary objective of the test plan for Home Star is to identify any defects in the system's functionality, usability, performance, and security. The test plan defines which modules will be tested, what testing methods will be used, and the schedule of testing activities. By adhering to this structured plan, the testing team can systematically assess each component of the platform and resolve any issues before final deployment. The test plan also sets the criteria for test success, identifying the conditions under which the software will be considered ready for release. This ensures that Home Star will meet both functional and non-functional requirements, providing confidence that it will deliver a seamless user experience.

5.2.1 Unit Testing

Unit testing involves testing individual components or modules of the Home Star platform in isolation to ensure that they function correctly. This type of testing focuses on small, manageable parts of the system, such as functions, classes, or database queries, verifying that each performs as expected. In the context of Home Star unit tests will cover critical areas such as user authentication, course enrollment, and service requests to make sure these foundational operations are free of errors. Home StaAmal Jyothi College of Engineering Autonomous,

Home service management system

Kanjirappally Department of Computer Applications The primary objective of unit testing is to catch and fix errors early in the development process, before they cascade into larger issues during integration. This testing is often automated using tools like Jest or Mocha for JavaScript-based platforms like Artistry Hub, ensuring quick identification of issues in isolated code

5.2.2 Integration Testing Explanation

Integration testing in the Home Services Management System focuses on verifying that different components of the platform work together as intended. While unit testing examines each component individually, integration testing ensures that, once combined, these components interact correctly and function seamlessly. For the Home Services Management System, integration testing includes verifying interactions between the front-end interface, back-end services, and the database. For instance, it ensures that users can successfully book a service after logging in, that service requests are properly stored and updated in the database, and that service providers can access their task lists and update the status of completed jobs.

The objective of integration testing is to identify any defects related to the interaction between various components or subsystems, ensuring that all parts of the system work cohesively. This testing phase also includes testing APIs, data flows, and user interactions to guarantee smooth operation across the platform, ultimately providing a seamless experience for homeowners, service providers, and administrators.

5.2.3 Validation Testing or System Testing

Validation testing, also known as system testing, for the Home Service Management System involves testing the entire platform as a complete system. This phase ensures that the system meets the specified requirements and behaves as expected in a real-world environment. System testing for the Home Service Management System will assess the overall user experience, including booking services, managing appointments, interacting with service providers, and tracking service status. This phase of testing covers both functional and non-functional requirements, such as performance, scalability, and security. For example, system testing might evaluate how the platform handles high volumes of service bookings during peak hours, ensures the reliability of service provider availability, and verifies the security of user data during transactions and communication. The goal of validation testing is to ensure that the entire system operates seamlessly and efficiently as an integrated whole, offering a

Home service management system

smooth experience for both users and service providers. This includes evaluating the usability of the user interface, ensuring accurate service scheduling, confirming payment processing functionality, and assessing the performance of customer support channels.

5.2.4 Output Testing or User Acceptance Testing

User Acceptance Testing (UAT), also known as output testing, is the final phase of testing before the Home Service Management System is released to the public. It involves real users—such as customers, service providers, and administrators—testing the platform to verify that it meets their needs and expectations. UAT ensures that the platform's functionality aligns with the defined requirements and that it is intuitive and easy to use for its target audience. In UAT, end users will test key features such as booking services, managing appointments, reviewing service provider profiles, processing payments, and tracking service progress. This testing is critical because it provides feedback from actual users, helping the development team identify any usability issues, functional gaps, or overlooked bugs before full deployment.

The goal of UAT is to ensure the platform is user-friendly, meets the specific needs of both service seekers and providers, and provides a seamless experience that reflects real-world usage. The feedback gathered during this phase will allow the team to make necessary adjustments or improvements before the system goes live.

5.2.5 Automation Testing explanation.

Automation testing involves the use of software tools to automatically execute tests and compare actual outcomes with expected results. For Home Star, automation testing will focus on repetitive tasks such as user registration, login, and profile updates to ensure that these features consistently function as expected. Automation testing helps reduce the time required for testing by running tests in a continuous integration/continuous deployment (CI/CD) pipeline, ensuring faster feedback for developers. Tools like Selenium or Cypress can be used to simulate user interactions, such as navigating the platform or submitting forms, allowing the testing team to catch any errors without needing to manually repeat the tests. Automation testing is especially valuable in large projects like Artistry Hub, where frequent updates require regular testing to ensure stability.

5.2.6 Selenium Testing

Selenium testing is a popular form of automation testing specifically designed for web applications. For the Home Service Management System, Selenium testing will be used to automate browser interactions such as navigating the platform, submitting service requests, managing user profiles, and verifying page rendering. This ensures that the system behaves as expected across different browsers and devices.

Cross-browser compatibility is critical for a platform like the Home Service Management System, as users may access it from various browsers (e.g., Chrome, Firefox, Safari) and devices (e.g., smartphones, tablets, desktops). Selenium enables the creation of robust test scripts that can perform complex interactions, such as booking a service, interacting with service providers, or completing payment transactions.

The key benefit of using Selenium is its ability to run tests across multiple browsers in parallel, ensuring that the Home Service Management System provides a consistent and smooth user experience regardless of the user's browser or device environment. Additionally, automation through Selenium improves the efficiency and effectiveness of testing, reducing manual effort while increasing test coverage.

Example:

Test Case 1

Login

Code

package definitions

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
import io.cucumber.java.en.And;
```

```
import io.cucumber.java.en.Given;
```

```
import io.cucumber.java.en.Then;
```

Home service management system

```
import io.cucumber.java.en.When;

public class stepdefinitions {

    WebDriver driver=null;

    @Given("browser is open")

    public void browser_is_open() {

        System.out.println("Inside step-Browser is open");

        System.setProperty("webdriver.gecko.marionette","C:\\Users\\shuha\\eclipse-
workspace\\sample\\src\\test\\resources\\drivers\\geckodriver.exe");

        driver=new FirefoxDriver();

        driver.manage().window().maximize();

    }

    @And("user is on login page")

    public void user_is_on_login_page() throws Exception {

        driver.navigate().to("http://127.0.0.1:8000/login/");

        Thread.sleep(2000);

    }

}
```

```

@When("user enters email and password")
public void user_enters_email_and_password() throws Throwable{

    driver.findElement(By.id("username")).sendKeys("fathi");

    driver.findElement(By.id("password")).sendKeys("fathi@123");

}

@And("user clicks on login")

public void user_clicks_on_login() {

    driver.findElement(By.id("login")).click();

}

```

} Eg.Screenshot

```

Scenario: Check login is successful with valid credentials # src/test/java/Features/Login.feature:5
Inside step-Browser is open
    Given browser is open                                # definitions.stepdefinitions.browser_is_open()
    And user is on login page                            # definitions.stepdefinitions.user_is_on_login_page()
    When user enters email and password                  # definitions.stepdefinitions.user_enters_email_and_password()
    And user clicks on login                            # definitions.stepdefinitions.user_clicks_on_login()

1 Scenarios (1 passed)
4 Steps (4 passed)
0m33.160s

```

Eg.Test Report

Test Case 1

Project Name: Home Service Management					
Login Test Case					
Test Case ID: Test_1			Test Designed By: Sabhamol P Haneefa		
Test Priority(Low/Medium/High):Medium			Test Designed Date: 02/11/2024		
Module Name: Login Module			Test Executed By : Sr.Dr.Elsin Chakkalackal. SH		
Test Title : User_Login			Test Execution Date: 02/11/2024		
Description: User has a valid email/password					
Pre-Condition :User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to the login page	URL: http://127.0.0.1:8000/login/	Login page should display	Login page is displayed	Pass
2	Enter valid email in email field	Username: admin	User should able to login	User logged in	Pass
3	Enter valid password in password field	Password: Gayathry@123			
4	Click the login button	N/A			
5	Click on login button		Admin dashboard or main page is displayed	Dashboard displayed	Pass
6	Click on "Manage Work		Navigates to the worker management page	Worker management page displayed	Pass
7	Click on "View Certificate " link		Certificate is displayed	Certificate displayed	Pass
Post-Condition: The user is authenticated and redirected to their account dashboard.					

Test Case 2: Add Announcement**Code**

```

package definitions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;

public class book {

    WebDriver driver = null;

    @Given("browser is opens")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\Users\\shuha\\eclipse-
workspace\\sample\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }

    @And("user is on login pages")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/admin_login");
        Thread.sleep(2000);
    }

    @When("user enters email and passwords")
    public void user_enters_email_and_password() throws Throwable {
        driver.findElement(By.id("admin-username")).sendKeys("admin");
        driver.findElement(By.id("admin-password")).sendKeys("admin@123");
    }

    @And("user clicks on logins")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }

    @Then("user clicks on add announcement")
    public void user_clicks_on_add_announcement() {
        driver.findElement(By.id("adminadd")).click(); //
    }

}

```

Screenshot

Home service management system

```

Scenario: logs in and adds an announcement # src/test/java/Features/book.feature:3
Inside step-Browser is open
  Given browser is opens          # definitions.book.browser_is_open()
  And user is on login pages      # definitions.book.user_is_on_login_page()
  When user enters email and passwords # definitions.book.user_enters_email_and_password()
  And user clicks on logins       # definitions.book.user_clicks_on_login()
  Then user clicks on add announcement # definitions.book.user_clicks_on_add_announcement()

1 Scenarios (1 passed)
5 Steps (5 passed)
0m9.784s

```

Test report

Test Case 2

Project Name: Home Service Management

Accept Test Case

Test Case ID: Test_2	Test Designed By: Sabhamol P Haneefa
Test Priority(Low/Medium/High):Medium	Test Designed Date: 2/11/2024
Module Name: Worker Module	Test Executed By : Sr.Dr.Elsin Chakkalackal. S H
Test Title : Admin Login and Add Announcement	Test Execution Date: 2/11/2024
Description: Tests login functionality with valid credentials and adding an announcement.	

Pre-Condition :User has valid username and password

Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to the login page	URL: http://127.0.0.1:8000/admin_login	Login page should display	Login page is displayed	Pass
2	Enter valid email in email field	Username: admin	User should able to login	User logged in	Pass
3	Enter valid password in password field	Password: admin@123			
4	Click the login button	N/A			
5	Click on login button		Admin dashboard or	Dashboard displayed	Pass

Home service management system

			main page is displayed		
6	Click on "Manage Worker"		Navigates to the worker management page	Worker management page displayed	Pass
7	Click on "View Certificate" link		Certificate is displayed	Certificate is displayed	Pass
Post-Condition: User is logged into the system and can view the worker's certificate.					

Test Case 3: Add Announcement**Code**

package definitions;

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import io.cucumber.java.en.And;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.When;
import io.cucumber.java.en.Then;
```

```
public class book {
```

```
    WebDriver driver = null;
```

```
    @Given("browser is opens")
    public void browser_is_open() {
        System.out.println("Inside step-Browser is open");
        System.setProperty("webdriver.gecko.marionette", "C:\\Users\\shuha\\eclipse-workspace\\sample\\src\\test\\resources\\drivers\\geckodriver.exe");
        driver = new FirefoxDriver();
        driver.manage().window().maximize();
    }
```

```
    @And("user is on login pages")
    public void user_is_on_login_page() throws Exception {
        driver.navigate().to("http://127.0.0.1:8000/admin_login");
        Thread.sleep(2000);
    }
```

```
    @When("user enters email and passwords")
    public void user_enters_email_and_password() throws Throwable {
```

Home service management system

```

        driver.findElement(By.id("admin-username")).sendKeys("admin");
        driver.findElement(By.id("admin-password")).sendKeys("admin@123");
    }

    @And("user clicks on logins")
    public void user_clicks_on_login() {
        driver.findElement(By.id("login")).click();
    }

    @Then("user clicks on add announcement")
    public void user_clicks_on_add_announcement() {
        driver.findElement(By.id("adminadd")).click(); // Updated method name
    }
}

```

Screenshot

```

Given browser is openss          # definitions.worker.browser_is_open()
And user is on login pagess      # definitions.worker.user_is_on_login_page()
When user enters email and passwordss # definitions.worker.user_enters_email_and_password()
And user clicks on loginss       # definitions.worker.user_clicks_on_login()
Then user clicks on manage worker  # definitions.worker.user_clicks_on_manage_worker()
And user clicks on view certificate # definitions.worker.user_clicks_on_view_certificate()

1 Scenarios (1 passed)
6 Steps (6 passed)
0m38.523s

```

Test report

Test Case 3	
Project Name: Home Service Management System	
View Certificate Test Case	
Test Case ID: Test_2	Test Designed By: Sabhamol P Haneefa
Test Priority(Low/Medium/High):Medium	Test Designed Date: 2/11/2024
Module Name: Worker Module	Test Executed By : Sr.Dr.Elsin Chakkalackal. SH
Test Title : Admin Login and Worker Management	Test Execution Date: 2/11/2024
Description: Tests login functionality for valid credentials and views worker certificates.	

Home service management system

Pre-Condition : User has valid username and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigate to the login page	URL: http://127.0.0.1:8000/admin_login	Login page should display	Login page is displayed	Pass
2	Enter valid email in email field	Username: admin	User should be able to login	User logged in	Pass
3	Enter valid password in password field	Password: admin@123			
4	Click the login button	N/A			
5	Click on login button		Admin dashboard or main page is displayed	Dashboard displayed	Pass
6	Click on "Manage Worker"		Navigates to the worker management page	Worker management page displayed	Pass
7	Click on "View Certificate" link		Certificate is displayed	Certificate is displayed	Pass
Post-Condition: User is logged into the system and can view the worker's certificate.					

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

This chapter outlines the implementation phase of the Home Service Management System, detailing the procedures, training, and maintenance strategies necessary for the effective deployment and operation of the system. The successful implementation of the system is critical for providing users—customers, service providers, and administrators—with a platform that facilitates seamless service bookings, enhances communication, and improves overall service delivery. The implementation process will focus on ensuring that all stakeholders are equipped to utilize the platform effectively. This chapter will cover the following areas:

6.2 IMPLEMENTATION PROCEDURES

The implementation of the Home Star is structured into several key procedures aimed at facilitating a seamless transition from development to deployment. Each procedure will focus on preparing users, ensuring the platform's reliability, and maintaining optimal performance

6.2.1 User Training

User training is essential for ensuring that all users of the Home Service Management System can effectively navigate and utilize the platform. The training will be comprehensive and tailored to different user roles, including customers, service providers, and administrators.

Training Objectives:

- Familiarize users with the platform's features and functionalities.
- Empower users to create accounts, manage profiles, and interact with others on the platform.
- Teach users how to book services, manage appointments, track service progress, and communicate with service providers.
- Ensure that service providers can manage their service offerings, accept bookings, and update their availability.

Training Methods:

- Workshops: Interactive workshops will be organized, where users can learn in a group setting. These workshops will be designed for different user groups, including customers, service providers, and administrators, and will cover the core functionalities of the platform.

Home service management system

- **Online Tutorials:** Video tutorials and written guides will be made available on the platform, allowing users to learn at their own pace. These resources will cover basic to advanced features, such as how to book services, manage payments, and optimize service offerings.
- **User Manuals:** Comprehensive user manuals will be distributed, providing step-by-step instructions for using the platform's features. These manuals will be tailored to different roles, helping users understand how to navigate the system effectively and perform key tasks.
- **Feedback Mechanism:** After training sessions, feedback will be collected from users to assess their understanding of the system. Surveys and questionnaires will help identify areas for improvement in the training process, ensuring that future sessions address any gaps in user comprehension.

6.2.2 Training on the Application Software

In addition to general user training, specific training will be provided on the application software utilized in the Home Service Management System. This training will ensure that administrators, service providers, and support teams are well-versed in the technical aspects of the platform to manage and troubleshoot the system effectively.

Training Components:

Overview of the Application: A detailed overview of the application architecture and core components of the Home Service Management System will be provided. This session will help users understand how the system operates, including its key modules such as service booking, provider management, payment processing, and customer communication. Understanding the structure of the platform will equip users with the knowledge to efficiently navigate and manage the system.

Hands-On Training: Participants will engage in hands-on sessions where they can directly interact with the application. This practical training will cover tasks such as creating and managing service provider profiles, processing customer bookings, updating service availability, and handling customer feedback. These sessions will give users real-world experience with the platform, ensuring they are comfortable with its features.

- **Support Resources:** Access to a dedicated support team will be available for users who require additional assistance after the training sessions. This team will help resolve any

Home service management system

issues or answer questions regarding the use of the platform. In addition, users will have access to online resources such as FAQs, troubleshooting guides, and help documentation to support their continued learning.

- **Evaluation of Training Effectiveness:** Post-training assessments will be conducted to evaluate user proficiency in the application's core functionalities.

6.2.3 System Maintenance

To ensure the Home Service Management System operates smoothly and remains responsive to user needs, ongoing system maintenance will be implemented. Regular maintenance will address performance, security, and feature improvements to keep the platform functioning at optimal levels.

Maintenance Objectives:

- **Ensure system reliability and performance:** The platform will be kept in top working condition to ensure it performs well even during peak usage periods.
- **Address bugs and issues in a timely manner:** Any technical issues, glitches, or user-reported problems will be promptly fixed to minimize downtime and user dissatisfaction.
- **Implement feature updates based on user feedback:** Regular updates will be introduced to enhance the platform's functionality and meet the evolving needs of users, including service providers and customers.

Maintenance Procedures:

- **Regular Backups:** Scheduled backups of the database, including user data, service records, and transaction histories, will be performed to prevent data loss in case of unexpected events or system failures. Backup procedures will ensure that data can be restored quickly with minimal impact on the platform's operation.
- **Performance Monitoring:** The system will be continuously monitored for performance metrics, such as response times, uptime, error rates, and load balancing. This monitoring will help identify potential bottlenecks or issues early, ensuring that the platform delivers fast and reliable service to all users.
- **User Feedback:** An open channel for user feedback (such as surveys, reviews, or support tickets) will be established to identify areas requiring maintenance, improvement, or new

Home service management system

feature requests. Feedback will be regularly reviewed to ensure that the platform meets user expectations and remains user-friendly.

- **Scheduled Maintenance:** Regularly scheduled maintenance windows will be planned in advance and communicated to users to ensure minimal disruption. During these windows, system updates, security patches, and other essential tasks will be carried out. Users will be notified in advance of any expected downtime or service interruptions.

6.2.4 Hosting

The Home Service Management System will be hosted on a reliable and scalable cloud hosting provider that meets the platform's current and future requirements. This will ensure high availability, security, and performance as the platform grows and serves a diverse user base.

Reasons for Choosing the Hosting Provider:

- **Ease of Deployment:** The hosting provider offers a streamlined deployment process with direct integration with GitHub repositories, simplifying the setup and updates of applications. This will enable seamless version control, quick deployments, and efficient updates, ensuring that the Home Service Management System can easily evolve over time.
- **Scalability:** The hosting solution provides scalable infrastructure that can grow with the platform. As the Home Service Management System experiences increased user traffic and service demand, the provider will be able to adjust resources (such as CPU, memory, and storage) to maintain optimal performance. This ensures that the system remains responsive even during peak usage periods.
- **Full-Stack Support:** The hosting provider supports both frontend and backend deployments, which is essential for the system's architecture. Whether it's managing customer requests, handling service provider interactions, or processing payments, the full-stack support ensures that both the frontend (user interface) and backend (database and server) work seamlessly together, offering a smooth experience for all users.

Render Hosting Setup Procedure:**Step 1: Connect the Repository**

Visit Render's website and log in to your account. • Create a new web service on Render by linking it to the GitHub repository where your Home Star project is stored. • Choose the main branch for deployment and select any necessary build and start commands specific to your application. Visit Render's website and log in to your account.

- Create a new web service on Render by linking it to the GitHub repository where your Artistry Hub project is stored.
- Choose the main branch for deployment and select any necessary build and start commands specific to your application.

Step 2: Configure Environment Variables

In the service settings, add any required environment variables, such as `VITE_BACKEND_URL` for frontend settings or database connection strings for the backend. • Render automatically injects these variables during runtime, ensuring secure and reliable application configuration.

Step 4: Deploy the Application

Render will automatically detect the correct build environment (Node.js, in this case) and initiate the deployment. • Monitor the deployment logs for any errors and verify that all packages and dependencies are properly installed.

Step 5: Testing and Monitoring

Once deployed, visit the application URL to verify that all pages, features, and API Artistry-Hub endpoints are functional. • Use Render's monitoring tools to track application performance and set up automatic redeployment for future changes to the GitHub repository.

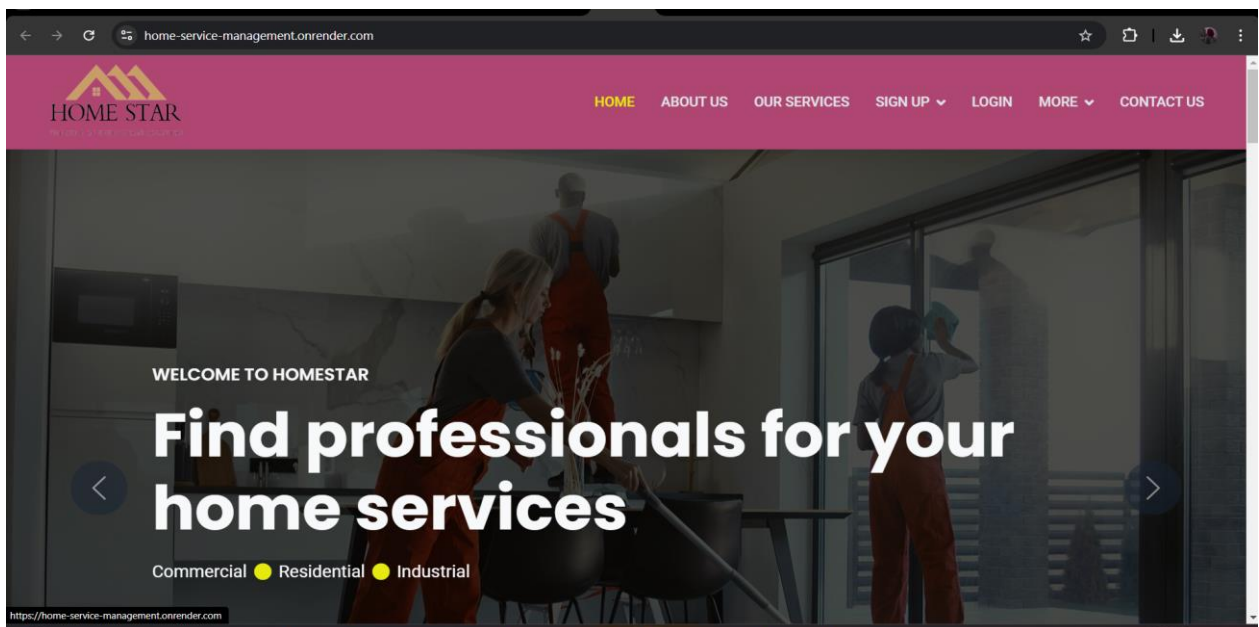
Home service management system

Hosted Link : <https://home-service-management.onrender.com/>

Hosted QR Code :



Screenshot



CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In conclusion, the Home Service Management System is a comprehensive and user-centric platform designed to revolutionize the way home services are managed and delivered. By providing a seamless interface for both customers and service providers, the platform streamlines the process of booking, managing, and tracking services, ensuring an efficient and reliable experience for all users. The system caters to the growing needs of users by offering easy navigation, real-time communication, and a flexible, scalable architecture that ensures high performance even during peak usage.

Through its diverse range of features, the Home Service Management System fosters greater connectivity between customers and service providers, encouraging transparency, trust, and satisfaction. The ability to book services, view service histories, and receive prompt feedback enhances the overall user experience, while the platform's built-in messaging system facilitates clear and direct communication, making interactions smoother and more efficient.

For service providers, the platform offers an opportunity to manage their services, update availability, and interact with customers, thus expanding their reach and boosting engagement. Furthermore, the system ensures that both customers and providers have access to critical support and resources, empowering them to resolve issues promptly and manage their tasks more effectively.

The system is built to be highly scalable, ensuring that it can grow alongside the increasing demand for home services, adapting to new market trends and user needs. Its robust testing, continuous maintenance, and user feedback mechanisms ensure that the platform remains secure, reliable, and responsive, providing an excellent user experience at all times.

Ultimately, the Home Service Management System is more than just a tool for service booking; it is an integrated solution that enhances the entire service experience. By focusing on user satisfaction, operational efficiency, and scalable growth, the platform is positioned to become an essential tool for managing home services in the digital era, offering value to customers, service providers, and administrators alike.

7.2 FUTURE SCOPE

The future scope of the Home Service Management System is highly promising, with numerous opportunities for growth and enhancement that can further benefit both users and service providers. The following potential areas for development will help improve the system's offerings and ensure it remains competitive in the ever-evolving market:

- **Enhanced User Engagement:** The system can leverage advanced analytics and artificial intelligence to enhance user interactions. By analyzing customer preferences, service history, and feedback, the platform can offer personalized service recommendations, suggest relevant providers, and even send reminders for recurring services. This customization would not only improve customer satisfaction but also increase engagement with the platform, encouraging users to make more frequent bookings.
- **Expanded Service Offerings:** Future versions of the platform could introduce new service categories and features such as subscription-based models, emergency service requests, or on-demand services. Implementing features like live tracking of service providers, real-time updates on service status, and live chat for instant support could significantly enhance the customer experience.
- **Improved Service Provider Integration:** Strengthening the connection between customers and service providers is crucial for platform growth. Integrating real-time availability, geo-location tracking, and service-specific ratings and reviews would help users find the most suitable providers more efficiently. Additionally, offering service provider tools for inventory management, service scheduling, and performance tracking could further enhance the operational efficiency of service providers.
- **Mobile Accessibility:** With increasing mobile device usage, developing a mobile application for the Home Service Management System is a natural next step. A mobile app would allow users to book services, track providers, manage appointments, and make payments on the go, ensuring a seamless experience regardless of location. The app would also improve engagement and user retention by offering notifications and alerts for service updates, promotions, or reminders.
- **Subscription Models and Loyalty Programs:** Introducing subscription-based plans for regular customers or loyalty programs can drive customer retention and incentivize repeat usage. By offering discounted rates or exclusive benefits to loyal users, the system can encourage long-term engagement and increase its user base.
- **Enhanced Customer Support:** Expanding customer support features through chatbots, 24/7 customer service options, and AI-driven problem resolution could improve user

Home service management system

satisfaction. Instant support can help resolve issues quickly, thereby reducing frustration and fostering trust in the platform.

- **Collaborative Projects and Service Bundling:** The platform could introduce collaborative offerings where service providers can bundle services together. For example, home maintenance providers could partner with cleaners or decorators to offer discounted bundled packages for users, fostering collaboration within the service provider network and offering added value to customers.
- **Global Expansion and Multi-Language Support:** As the platform grows, expanding its reach to global markets could open new avenues for user acquisition. Incorporating multilingual support and offering region-specific services would help cater to diverse demographics, creating a more inclusive and accessible service management platform.

Overall, the Home Service Management System has significant potential for growth and innovation. By continuously improving its features and expanding its reach, the platform can offer greater value to customers, service providers, and administrators, positioning itself as a leading solution in the home services market.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- Pressman, Roger S. Software Engineering: A Practitioner's Approach. McGraw-Hill, 2014.
- Sommerville, Ian. Software Engineering. Addison-Wesley, 2016.
- Martin, Robert C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.
- Fowler, Martin. Refactoring: Improving the Design of Existing Code. Addison-Wesley, 2018.
- Albahari, Ben, and Joseph Albahari. C# 7.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2017.
- Gamma, Erich, et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
- Johnson, Robert L., and Patrick J. O'Neil. Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design. Addison-Wesley, 2013.
- McWherter, Jeff, and Scott Gowell. Professional Mobile Application Development. Wrox, 2012.
- Reilly, Paul, and Andy Oram. Managing Systems and Data in the Cloud: Practical Guide to Cloud Computing. O'Reilly Media, 2020.

WEBSITES:

- W3Schools. W3Schools Online Web Tutorials. <https://www.w3schools.com>
- Tailwind CSS Documentation. Tailwind CSS - A Utility-First CSS Framework. <https://tailwindcss.com/docs>
- MDN Web Docs. Mozilla Developer Network - Web Documentation. <https://developer.mozilla.org>
- Stack Overflow. Programming Community and Q&A. <https://stackoverflow.com>
- AWS Documentation. Amazon Web Services Documentation for Cloud Solutions. <https://aws.amazon.com/documentation/>
- DigitalOcean Tutorials. Cloud Infrastructure and Developer Documentation. <https://www.digitalocean.com/community/tutorials>

CHAPTER 9

APPENDIX

9.1 Sample Code

```

<!DOCTYPE html>
{% load static %}
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Booking Form</title>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500&family=Poppins:wght@600;700&display=swap" rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css"
rel="stylesheet">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css"
rel="stylesheet">
    <link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">
    <link href="{% static 'css/style.css' %}" rel="stylesheet">
    <style>
        /* Global Reset */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding-top: 0.6px; /* Adjust based on your navbar height */
            min-height: 100vh;
            position: relative;
        }

        .background-slideshow {
            position: fixed;

```

Home service management system

```
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    z-index: -1;
}

.slide {
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-size: cover;
    background-position: center;
    opacity: 0;
    transition: opacity 2s ease-in-out;
}

.slide.active {
    opacity: 1;
}

/* Navbar Styling */
.navbar-custom {
    background-color: #b33067;
    padding: 0;
}

.navbar-nav .nav-item {
    padding: 0 10px;
}

.navbar-nav .nav-link {
    color: #000000;
    font-weight: 500;
}

.navbar-nav .nav-link:hover,
```

Home service management system

```
.navbar-nav .nav-link.active {
    color: #007bff;
}

.profile-icon {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    margin-right: 10px;
}

.username {
    color: #000000;
    font-weight: 500;
}

/* Form Container */
.form-container {
    width: 50%;
    max-width: 1200px;
    margin: 20px auto 50px auto;
    background-image: url('{% static "img/booking.jpg" %}');
    background-size: cover;
    background-position: center;
    padding: 20px;
    box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
    border-radius: 8px;
    transition: transform 0.3s, box-shadow 0.3s;
}

.form-container::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: rgba(255, 255, 255, 0.8); /* Semi-transparent white overlay */
}
```

Home service management system

```
border-radius: 8px;
z-index: -1;
}

.form-container:hover {
  transform: scale(1.01);
  box-shadow: 0px 0px 20px rgba(0, 0, 0, 0.1);
}

.form-heading {
  text-align: center;
  font-size: 26px;
  color: #333;
  margin-bottom: 20px;
}

form {
  display: flex;
  flex-wrap: wrap;
}

.form-group-custom {
  flex: 1 1 45%;
  padding: 5px;
  display: flex;
  flex-direction: column;
  margin-bottom: 20px;
}

label {
  font-weight: bold;
  margin-bottom: 8px;
}

input, select, textarea {
  padding: 10px;
```

Home service management system

```
border-radius: 5px;
border: 1px solid #ccc;
font-size: 16px;
transition: border-color 0.3s;
background-color: #ede0a9; /* Darker yellow background */
}
```

```
input:focus, select:focus, textarea:focus {
border-color: #28a745;
outline: none;
}
```

```
input[type="submit"] {
background-color: #28a745;
color: rgb(220, 217, 138);
border: none;
cursor: pointer;
padding: 10px 20px;
font-size: 16px;
border-radius: 25px; /* Makes it oval */
width: auto; /* Allows the button to shrink */
transition: background-color 0.3s;
display: block; /* Centers the button */
margin: 20px auto 0; /* Adds some top margin and centers */
}
```

```
input[type="submit"]:hover {
background-color: #216830;
}
```

```
textarea {
resize: vertical;
}
```

```
.full-width {
flex: 1 1 100%;
}
```

Home service management system

```
}
```

```
.form-group-full {  
  flex: 1 1 100%;  
}
```

```
.form-row {  
  display: flex;  
  justify-content: space-between;  
  width: 100%;  
}
```

```
.form-group-half {  
  flex: 0 0 48%;  
}
```

```
.phone-input-container {  
  display: flex;  
  align-items: center;  
}
```

```
.worker-profile-img {  
  width: 100px;  
  height: 100px;  
  border-radius: 50%;  
  object-fit: cover;  
}
```

```
.phone-prefix {  
  padding: 10px;  
  background-color: #f0f0f0;  
  border: 1px solid #ccc;  
  border-right: none;  
  border-radius: 5px 0 0 5px;  
}
```

Home service management system

```
#phone_number {  
    border-left: none;  
    border-radius: 0 5px 5px 0;  
}  
  
.badge {  
    padding: 5px 10px;  
    border-radius: 4px;  
    font-size: 14px;  
}  
  
/* Book Now Button Styling */  
#submit-booking {  
    background: linear-gradient(to right, #28a745, #218838);  
    color: white;  
    padding: 12px 30px;  
    border: none;  
    border-radius: 25px;  
    font-size: 16px;  
    font-weight: 600;  
    cursor: pointer;  
    transition: all 0.3s ease;  
    box-shadow: 0 4px 15px rgba(40, 167, 69, 0.2);  
    margin: 20px auto;  
    display: none;  
    width: 200px;  
    text-transform: uppercase;  
    letter-spacing: 1px;  
    position: relative;  
    overflow: hidden;  
}  
  
#submit-booking:hover {  
    background: linear-gradient(to right, #218838, #1e7e34);  
    transform: translateY(-2px);  
    box-shadow: 0 6px 20px rgba(40, 167, 69, 0.3);
```


Home service management system

```

    }

    #submit-booking:active {
        transform: translateY(1px);
        box-shadow: 0 2px 10px rgba(40, 167, 69, 0.2);
    }

    /* Add icon to button */
    #submit-booking::before {
        content: "\f274"; /* Calendar icon from Font Awesome */
        font-family: "Font Awesome 5 Free";
        margin-right: 10px;
    }

    /* Button container for centering */
    .button-container {
        width: 100%;
        display: flex;
        justify-content: center;
        margin-top: 20px;
    }
</style>

<script>
    document.addEventListener('DOMContentLoaded', function () {
        const bookingDateInput = document.getElementById('booking_date');
        bookingDateInput.addEventListener('change', function () {
            const selectedDate = new Date(this.value);
            const today = new Date();
            today.setHours(0, 0, 0, 0);

            if (selectedDate.getDay() === 0) {
                alert("Bookings cannot be made on Sundays. Please select another date.");
                this.value = "";
            } else if (selectedDate <= today) {

```

Home service management system

```

        alert("Bookings cannot be made for today or past dates. Please select a future
date.");

```

```

        this.value = "";
    }
});

```

```

const phoneInput = document.getElementById('phone_number');
phoneInput.addEventListener('input', function () {
    this.value = this.value.replace(/\D/g, "").substring(0, 10);
    if (this.value.length === 10) {
        this.setCustomValidity("");
    } else {
        this.setCustomValidity('Phone number must be 10 digits');
    }
});

```

```

const stateSelect = document.getElementById('state');
const districtSelect = document.getElementById('district');

```

```

const stateDistricts = {
    'Kerala': ['Thiruvananthapuram', 'Ernakulam',
'Kozhikode', 'Alappuzha', 'Kollam', 'Kottayam', 'Pathanamthitta', 'Kasargod', 'Kannur', 'Malappuram',
'Palakkad', 'Thrissur', 'Wayanad'],
    'Karnataka': ['Bangalore', 'Mysore', 'Mangalore', 'Hubli'],
    'Tamil Nadu': ['Chennai', 'Coimbatore', 'Madurai', 'Trichy'],
    'Maharashtra': ['Mumbai', 'Pune', 'Nagpur', 'Nashik'],
};

```

```

stateSelect.addEventListener('change', function () {
    const selectedState = this.value;
    const districts = stateDistricts[selectedState] || [];

    districtSelect.innerHTML = '<option value="">Select District</option>';

```

```

    districts.forEach(function (district) {
        const option = document.createElement('option');
        option.value = district;

```

Home service management system

```

        option.text = district;
        districtSelect.appendChild(option);
    });
});

const bookingTimeInput = document.getElementById('booking_time');
bookingTimeInput.addEventListener('change', function () {
    const selectedTime = this.value;
    const [hours, minutes] = selectedTime.split(':').map(Number);

    if (hours >= 0 && hours < 6) {
        alert("Bookings cannot be made between 12 AM and 5 AM. Please select a
different time.");
        this.value = "";
    }
});
});
</script>
</head>

<body>
    <div class="background-slideshow">
        <div class="slide" style="background-image: url('{% static 'img/work22.jpg'
{% }}');"></div>
        <div class="slide" style="background-image: url('{% static 'img/wood1.jpg'
{% }}');"></div>
        <div class="slide" style="background-image: url('{% static 'img/wood23.jpg'
{% }}');"></div>
        <div class="slide" style="background-image: url('{% static 'img/washing.jpg'
{% }}');"></div>
        <div class="slide" style="background-image: url('{% static 'img/paint.jpeg' {% }}');"></div>
    </div>

    <!-- Navbar Start -->
    <nav class="navbar navbar-expand-lg bg-white navbar-light sticky-top px-4 px-lg-5 py-lg-
0">
        <a href="{% url 'home' %}" class="navbar-brand d-flex align-items-center">

```

Home service management system

```

    <h1 class="m-0">
        
    </h1>
</a>
<button type="button" class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#navbarCollapse">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarCollapse">
    <div class="navbar-nav ms-auto py-3 py-lg-0">
        <a href="{% url 'home' %}" class="nav-item nav-link">Home</a>
        <a href="{% url 'about' %}" class="nav-item nav-link">About Us</a>
        <a href="{% url 'service' %}" class="nav-item nav-link">Our Services</a>
        <a href="{% url 'booking' %}" class="nav-item nav-link active">Book Service</a>
        <a href="{% url 'feedback_form' %}" class="nav-item nav-link">Feedback</a>
        <a href="{% url 'search_work' %}" class="nav-item nav-link">Search Work</a>
        <div class="nav-item dropdown">
            <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">
                
                <span class="username">{{ username }}</span>
            </a>
            <div class="dropdown-menu bg-light m-0">
                <a href="{% url 'logout_view' %}" class="dropdown-item">Logout</a>
                <a href="{% url 'update' %}" class="dropdown-item">Update</a>
                <a href="{% url 'view_profile' %}" class="dropdown-item">View Profile</a>
            </div>
        </div>
    </div>
</div>
</div>
</nav>
<!-- Navbar End -->

<div class="form-container">
    <h2 class="form-heading">Book a Service</h2>
    {% if messages %}

```

Home service management system

```

<div class="alert alert-danger">
    {% for message in messages %}
    <p>{{ message }}</p>
    {% endfor %}
</div>

{% endif %}

<div class="row">
    <div class="col-md-4">
        <h3>Available Workers</h3>
        <div id="worker-list">
            {% for worker in available_workers %}
            <div class="worker-item">
                {% if worker.profile_image %}
                
                {% else %}
                
                {% endif %}
                <h4>{{ worker.worker_name }}</h4>
                <p>Rating: {{ worker.rating }}</p>
                <p>Service: {{ worker.services }}</p>
                <p>Status:
                    {% if worker.status == 'Available' %}
                    <span class="badge bg-success">Available</span>
                    {% else %}
                    <span class="badge bg-danger">Booked</span>
                    {% endif %}
                </p>
                {% if worker.status == 'Available' %}
                <button class="btn btn-primary select-worker" data-worker-id="{{
worker.id }}">Select</button>
                {% endif %}
            </div>
            {% empty %}
            <p>No workers available for the selected criteria.</p>
            {% endfor %}
        </div>
    </div>

```

Home service management system

```

    </div>
</div>
<div class="col-md-8">
    <!-- Your existing booking form here -->
    <form method="post" id="booking-form">
        {% csrf_token %}
        <div class="form-row">
            <div class="form-group-custom">
                <label for="booking_date">Booking Date:</label>
                <input type="date" id="booking_date" name="booking_date" required>
            </div>

            <div class="form-group-custom">
                <label for="booking_time">Booking Time:</label>
                <input type="time" id="booking_time" name="booking_time" min="06:00"
max="23:59" required>
            </div>
        </div>

        <div class="form-row">
            <div class="form-group-custom">
                <label for="state">State:</label>
                <select id="state" name="state" required>
                    <option value="">Select State</option>
                    <option value="Kerala">Kerala</option>
                    <option value="Karnataka">Karnataka</option>
                    <option value="Tamil Nadu">Tamil Nadu</option>
                    <option value="Maharashtra">Maharashtra</option>
                </select>
            </div>

            <div class="form-group-custom">
                <label for="district">District:</label>
                <select id="district" name="district" required>
                    <option value="">Select District</option>
                </select>
            </div>
        </div>
    </form>
</div>

```

Home service management system

```
</div>
```

```
</div>
```

```
<div class="form-row">
```

```
<div class="form-group-custom">
```

```
<label for="work">Services:</label>
```

```
<select id="work" name="work" required>
```

```
<option value="">Select Service</option>
```

```
{% for work in worker_types %}
```

```
<option value="{{ work }}">{{ work }}</option>
```

```
{% endfor %}
```

```
</select>
```

```
</div>
```

```
<div class="form-group-custom">
```

```
<label for="street_number">Street Number:</label>
```

```
<input type="text" id="street_number" name="street_number" required>
```

```
</div>
```

```
</div>
```

```
<div class="form-row">
```

```
<div class="form-group-custom">
```

```
<label for="phone_number">Phone Number:</label>
```

```
<div class="phone-input-container">
```

```
<span class="phone-prefix">+91</span>
```

```
<input type="tel" id="phone_number" name="phone_number" required  
pattern="[0-9]{10}" title="Phone number must be 10 digits">
```

```
</div>
```

```
</div>
```

```
<div class="form-group-custom">
```

```
<label for="city">City:</label>
```

```
<input type="text" id="city" name="city" required>
```

```
</div>
```

```
</div>
```

```
<div class="form-row">
```

Home service management system

```

        <div class="form-group-custom">
            <label for="address_line">Address:</label>
            <textarea id="address_line" name="address_line" rows="3"
placeholder="House Name, Pincode, etc." required></textarea>
        </div>

        <div class="form-group-custom">
            <label for="work_description">Work Description:</label>
            <textarea id="work_description" name="work_description" rows="3"
placeholder="Describe the work you want done" required></textarea>
        </div>
    </div>

    <input type="hidden" name="selected_worker" id="selected-worker-input">
    <div class="button-container">
        <button type="submit" id="submit-booking" style="display: none;">Book
Now</button>
    </div>
</form>
</div>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
    // Background slideshow
    document.addEventListener('DOMContentLoaded', function() {
        const slides = document.querySelectorAll('.slide');
        let currentSlide = 0;

        function showSlide(index) {
            slides.forEach((slide, i) => {
                if (i === index) {
                    slide.classList.add('active');
                } else {

```


Home service management system

```

        slide.classList.remove('active');
    }
});
}

function nextSlide() {
    currentSlide = (currentSlide + 1) % slides.length;
    showSlide(currentSlide);
}

showSlide(currentSlide);
setInterval(nextSlide, 5000); // Change slide every 5 seconds
});

document.addEventListener('DOMContentLoaded', function() {
    const form = document.getElementById('booking-form');
    const workerList = document.getElementById('worker-list');
    const selectedWorkerInput = document.getElementById('selected-worker-input');
    const submitButton = document.getElementById('submit-booking');

    // Handle form submission (Book Now button)
    form.addEventListener('submit', function(e) {
        e.preventDefault();
        if (!selectedWorkerInput.value) {
            alert('Please select a worker before booking.');
            return;
        }
    });

    // Get the form data
    const formData = new FormData(form);
    formData.append('submit_booking', 'true'); // Add flag to identify booking submission
    fetch(`${% url "bookings" %}`, {
        method: 'POST',
        body: formData,
        headers: {
            'X-CSRFToken': '{{ csrf_token }}'
        }
    });

```

Home service management system

```

    }
  })
  .then(response => response.json())
  .then(data => {
    if (data.success) {
      alert('Booking successful!');
      window.location.href = '{% url "booking" %}';
    } else {
      alert(data.message || 'Booking failed. Please try again.');
```

```

    }
  })
  .catch(error => {
    console.error('Error:', error);
    alert('An error occurred. Please try again.');
```

```

  });
});

function updateWorkerList() {
  const formData = new FormData(form);
  formData.append('update_workers', 'true');
```

```

  // Submit to booking URL for worker list updates
```

```

  fetch('{% url "booking" %}', {
    method: 'POST',
    body: formData,
    headers: {
      'X-CSRFToken': '{{ csrf_token }}'
    }
  })
```

```

  .then(response => response.text())
  .then(html => {
    const tempDiv = document.createElement('div');
    tempDiv.innerHTML = html;
    const newWorkerList = tempDiv.querySelector('#worker-list');
    if (newWorkerList) {
      workerList.innerHTML = newWorkerList.innerHTML;
    }
  })
}
```

Home service management system

```
    });  
  }  
  workerList.addEventListener('click', function(e) {  
    if (e.target.classList.contains('select-worker')) {  
      const workerId = e.target.getAttribute('data-worker-id');  
      selectedWorkerInput.value = workerId;  
      submitButton.style.display = 'block';  
    }  
  });  
  ['booking_date', 'state', 'district', 'work'].forEach(function(fieldId) {  
    document.getElementById(fieldId).addEventListener('change', updateWorkerList);  
  });  
});  
</script>  
</body>  
</html>
```

9.2 Screen Shots

9.2.1 Home Page

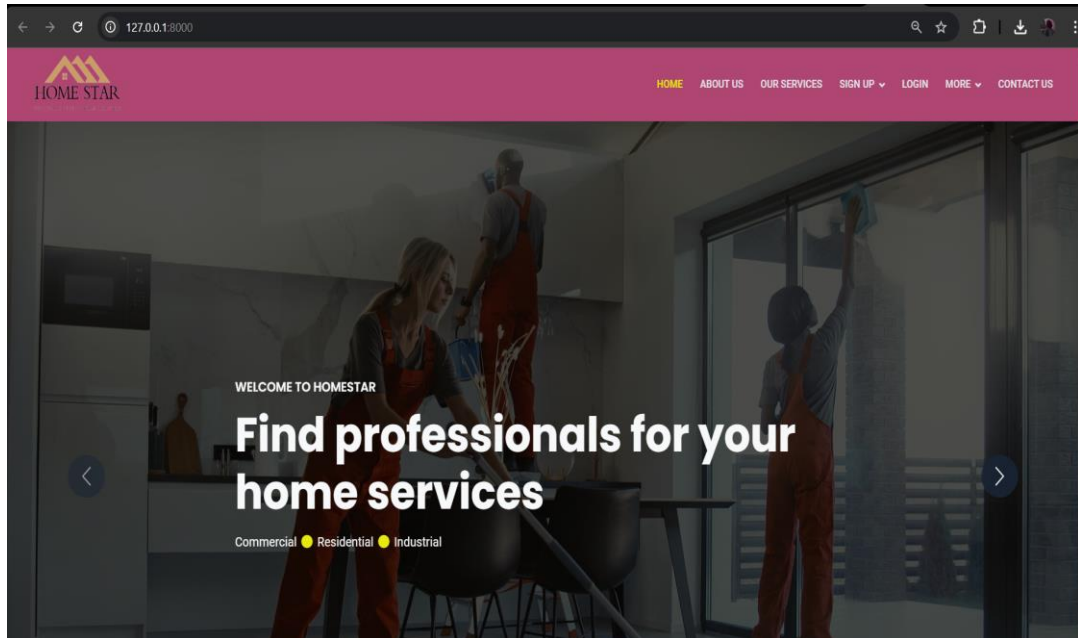


Figure 9.2.1: Home Page

9.2.2 Registration page

A screenshot of the Worker Registration Form on the Home Star website. The form is titled "Worker Registration Form" in green. It contains several input fields and dropdown menus. On the left side of the form, there is a vertical image of a toolbox. The form fields are: Name (text input), Profile Image (Choose File button, No file chosen), Email (text input), Gender (dropdown menu with "Male" selected), Date of Birth (text input with placeholder "dd-mm-yyyy" and a calendar icon), State (dropdown menu with "Select State" selected), District (dropdown menu with "Select District" selected), Address (text input), Pincode (text input), Phone No. (text input), Experience (dropdown menu with "1 year" selected), and Certificate (Choose File button, No file chosen). The top of the page shows the Home Star logo and navigation links: Home, About Us, Our Services, and Contact Us.

9.2.2 Registration page

Home service management system


9.2.3 User Registration

127.0.0.1:8000/register/

HOME STAR

Home About Us Our Services Contact Us

Create Your Account



Username
Choose a unique username

Email Address
Enter your email address

Password
Create a strong password
☐ Show Password
Password Strength: Weak

Sign Up

Figure 9.2.3 User Registration

9.2.4 Add Worker

HOME STAR

Admin Dashboard

Dashboard Logout

Manage Users
View and manage user accounts

Add Service
Create new services for customers

Add Announcement
Post new announcements

Manage Workers
Manage worker profiles and assignments

Manage Bookings
View and manage customer bookings

Manage Workers

[Add Worker](#)






ID	Worker Name	Email	Gender	Date of Birth	Address	Pincode	Phone No	Experience	Profile Image	Certificate	Resume	Status	Action
2		safa	sahamodhaneefa2025@mc.aqa.in	Female	Feb. 12, 2002	pulhinneetti	3323	0934567890	4	View Certificate	View Resume	Active	Approved Delete
5		adhi	adithyanagpu18209@gmail.com	Male	Nov. 14, 2002	dedeendiv	68612	9883283774	1	View Certificate	View Resume	Active	Approved Delete
7		safakutty	sahamodhaneefa@gmail.com	Female	Jan. 28, 2003	parayil house kanjirappally	68698	1234554321	4	View Certificate	View Resume	Active	Approved Delete
8		SAPA	voyagemaster.in@gmail.com	Female	Feb. 12, 2002	Kollam house Kollam p o	68689	1234567899	2	View Certificate	View Resume	Active	Approved Delete
9		shuhab	shuhabaneefa2022@gmail.com	Male	June 16, 1996	malikayil house kollam p o	21212	1234554321	4	View Certificate	View Resume	Active	Approved Delete
10		sanusi	sanusaj2025@mc.aqa.in	Male	July 11, 2002	Kollam house pathanamthitta	68689	5432112345	3	View Certificate	View Resume	Active	Approved Delete
11		Safa2002	sahamodhaneefa2002@gmail.com	Female	May 24, 2001	pulikat house palampra kanjirappally	2121	9876543221	4	View Certificate	View Resume	Active	Approved Delete
13		Sanusi	sanusaj2025@mc.aqa.in	Male	March 19, 2001	Kollam house pulikat	9999	687543217	6	View Certificate	View Resume	Inactive	Approved Delete

Figure 9.2.4 Add Worker

Home service management system

9.2.5 Add Service

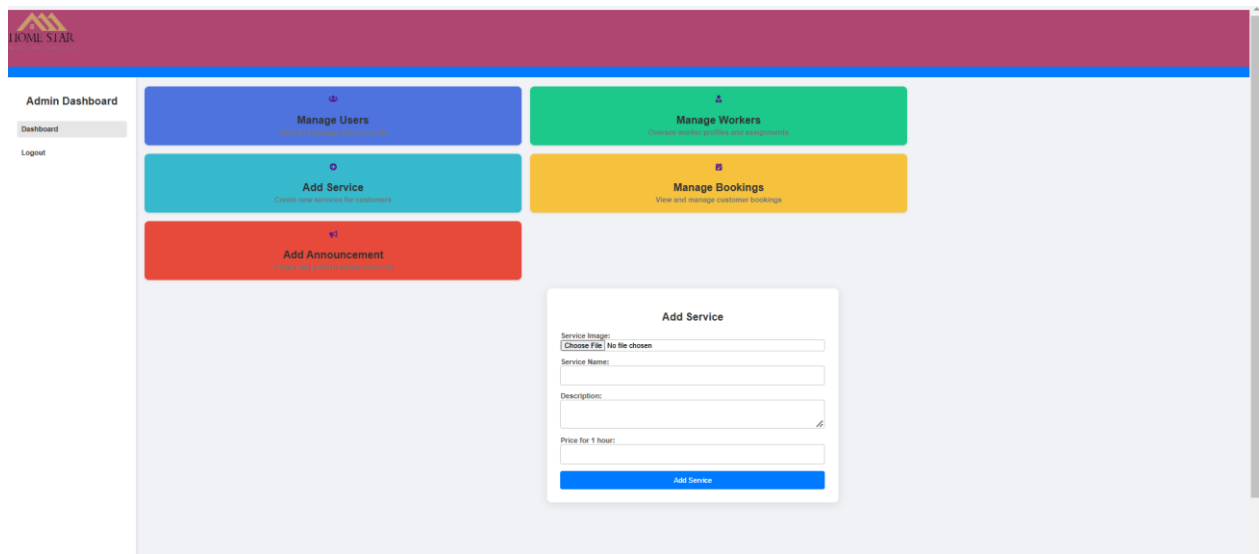


Figure 9.2.5: Add Service

9.2.6 Booking Page

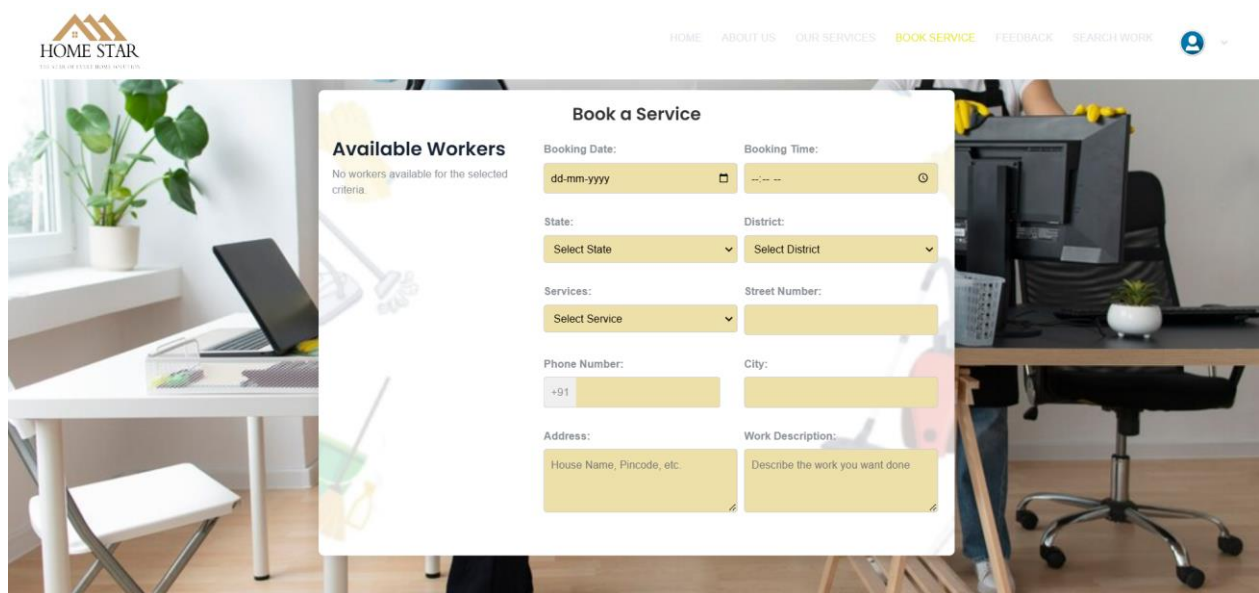
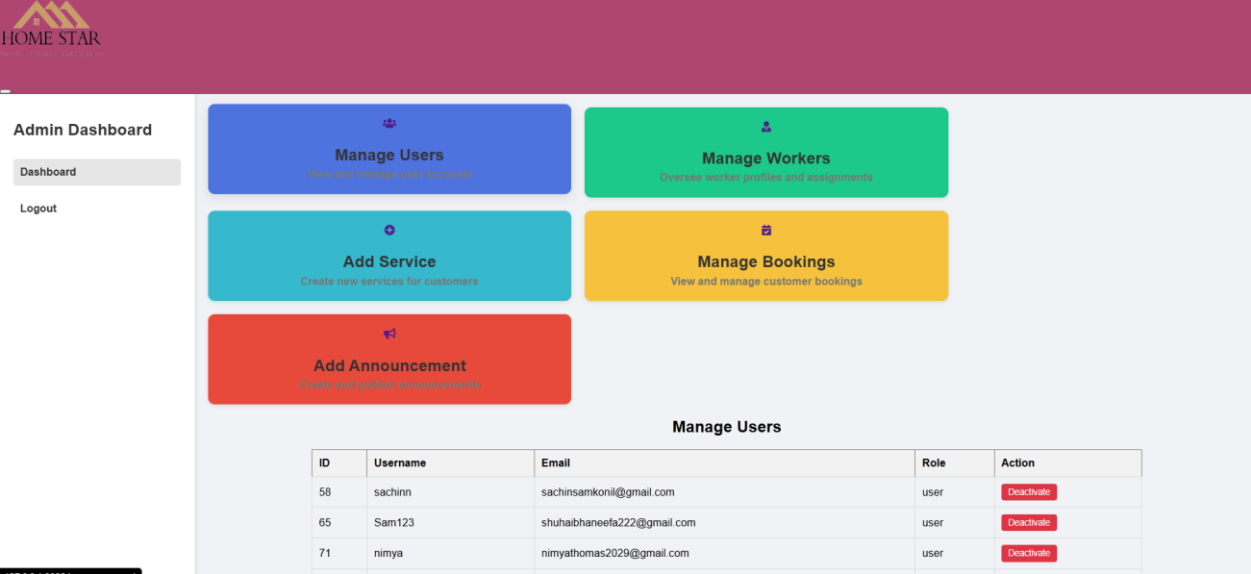


Figure 9.2.6: Service Booking

Home service management system

9.2.7 Manage Users

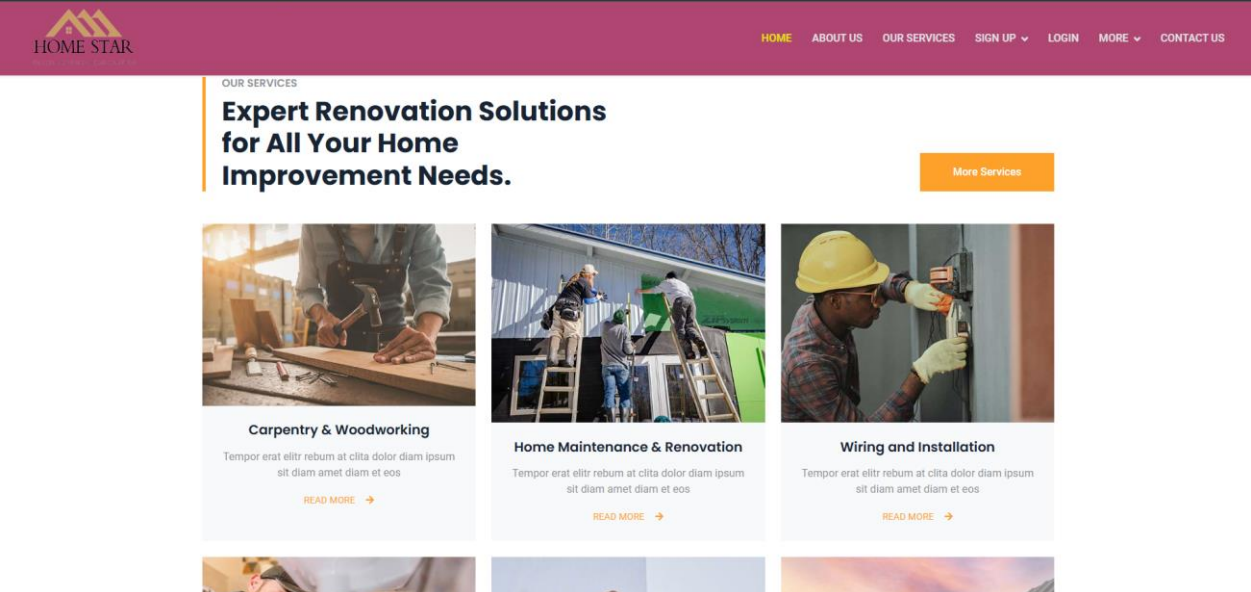


The screenshot shows the Home Star Admin Dashboard. On the left is a sidebar with 'Admin Dashboard', 'Dashboard', and 'Logout'. The main area contains five colored cards: 'Manage Users' (blue), 'Manage Workers' (green), 'Add Service' (cyan), 'Manage Bookings' (yellow), and 'Add Announcement' (red). Below these is a 'Manage Users' table with columns for ID, Username, Email, Role, and Action.

ID	Username	Email	Role	Action
58	sachinn	sachinsamkonil@gmail.com	user	Deactivate
65	Sam123	shuhaibhaneefa222@gmail.com	user	Deactivate
71	nimya	nimythomas2029@gmail.com	user	Deactivate

Figure 9.2.7: Manage Users

9.2.8 Service Page

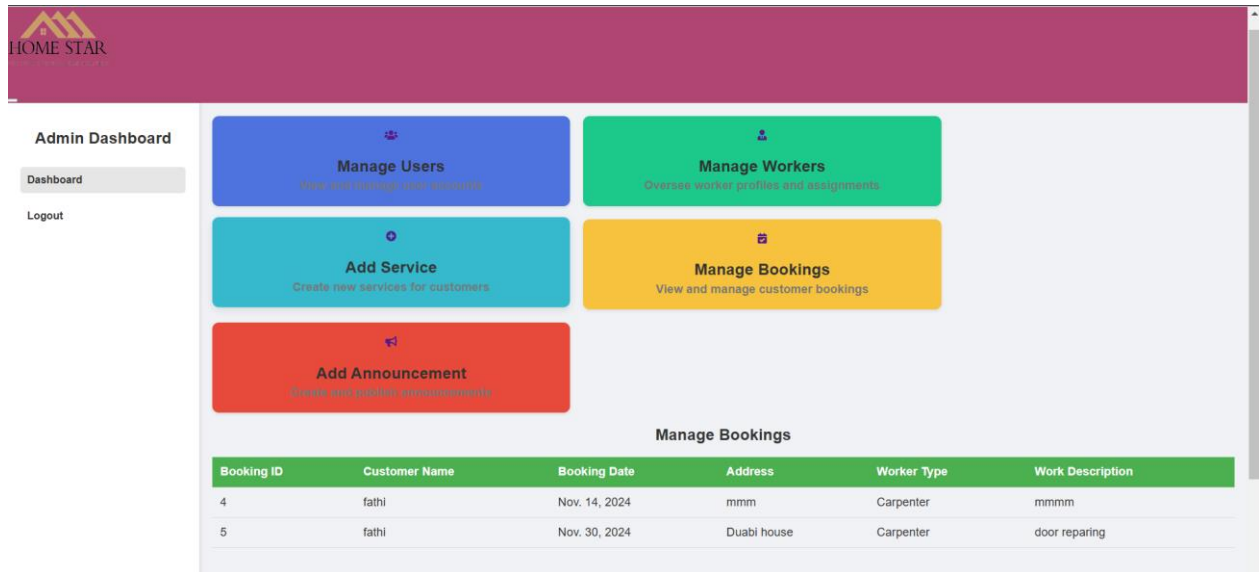


The screenshot shows the Home Star Service Page. The header includes the Home Star logo and navigation links: HOME, ABOUT US, OUR SERVICES, SIGN UP, LOGIN, MORE, and CONTACT US. The main content area features a heading 'Expert Renovation Solutions for All Your Home Improvement Needs.' and a 'More Services' button. Below this are three service cards: 'Carpentry & Woodworking', 'Home Maintenance & Renovation', and 'Wiring and Installation'. Each card includes a placeholder image, a title, a short description, and a 'READ MORE' link.

Figure 9.2.8: Service page

Home service management system

9.2.9 Manage Bookings

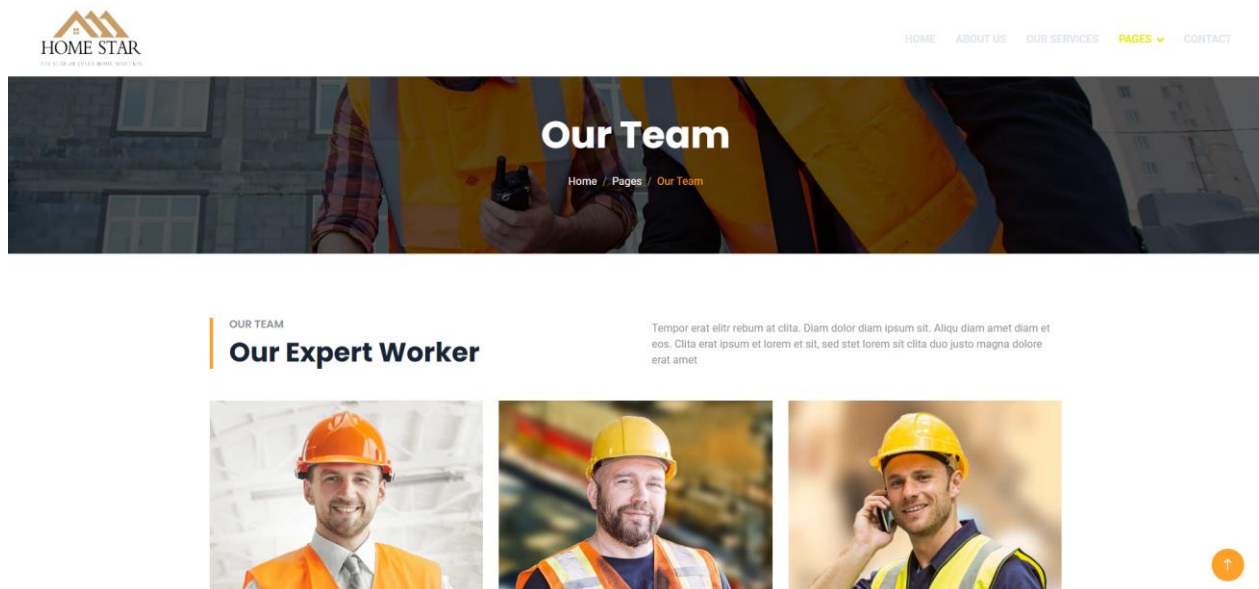


The screenshot shows the Home Star Admin Dashboard. On the left is a sidebar with 'Admin Dashboard', 'Dashboard', and 'Logout'. The main area contains five colored cards: 'Manage Users' (blue), 'Manage Workers' (green), 'Add Service' (cyan), 'Manage Bookings' (yellow), and 'Add Announcement' (red). Below these cards is a table titled 'Manage Bookings'.

Booking ID	Customer Name	Booking Date	Address	Worker Type	Work Description
4	fathi	Nov. 14, 2024	mmm	Carpenter	mmm
5	fathi	Nov. 30, 2024	Duabi house	Carpenter	door repairing

Figure 9.2.9 Manage Bookings

9.2.10 Worker team



The screenshot shows the Home Star website. The header includes the Home Star logo and navigation links: HOME, ABOUT US, OUR SERVICES, PAGES (active), and CONTACT. The main section is titled 'Our Team' with a background image of workers. Below this is a sub-section titled 'OUR TEAM' and 'Our Expert Worker'. It features three images of workers in hard hats and safety vests. To the right of the images is a block of placeholder text.

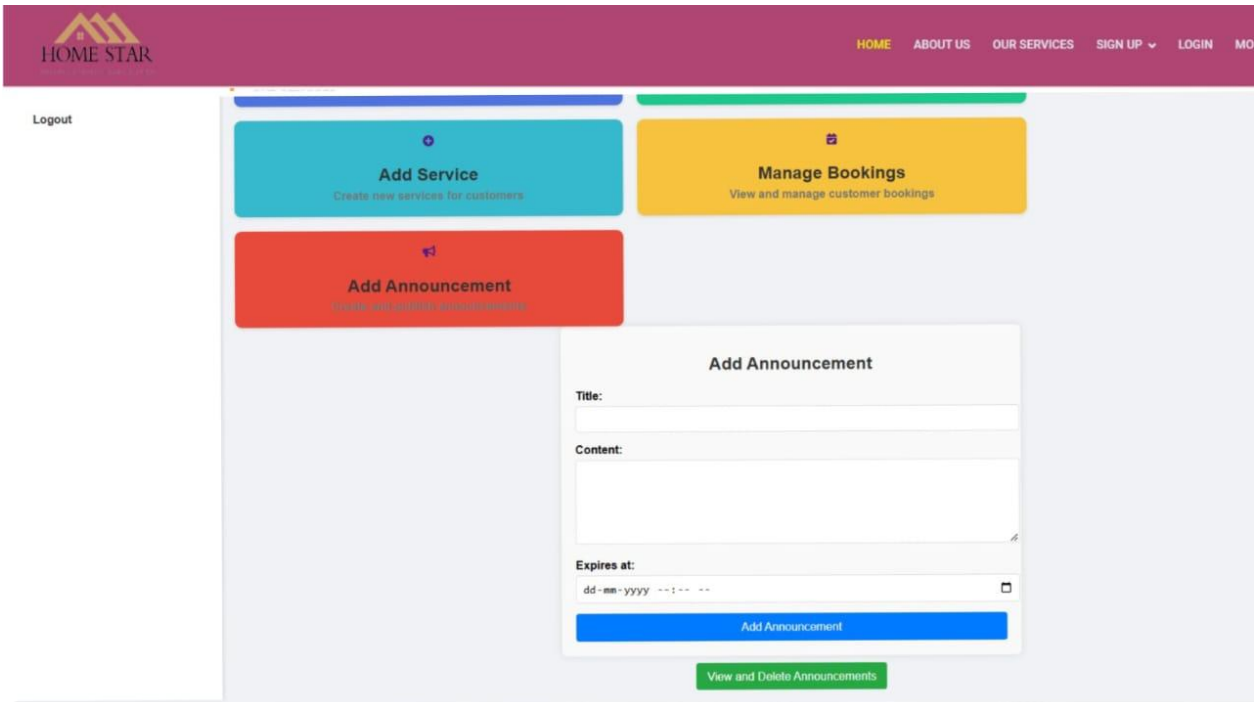
OUR TEAM

Our Expert Worker

Tempor erat elitr rebum at clita. Diam dolor diam ipsum sit. Aliqu diam amet diam et eos. Clita erat ipsum et lorem et sit, sed stet lorem sit clita duo justo magna dolore erat amet

Figure 9.2.10 Workers team

9.2.11 Add Announcement



The screenshot shows the 'Add Announcement' form within the Home Star system. The form is titled 'Add Announcement' and includes the following fields:

- Title:** A text input field.
- Content:** A large text area for the announcement content.
- Expires at:** A date and time picker with a calendar icon.

Below the form is a blue button labeled 'Add Announcement' and a green button labeled 'View and Delete Announcements'.

Figure 9.2.11 Add Announcements

9.2.12 User Home Page

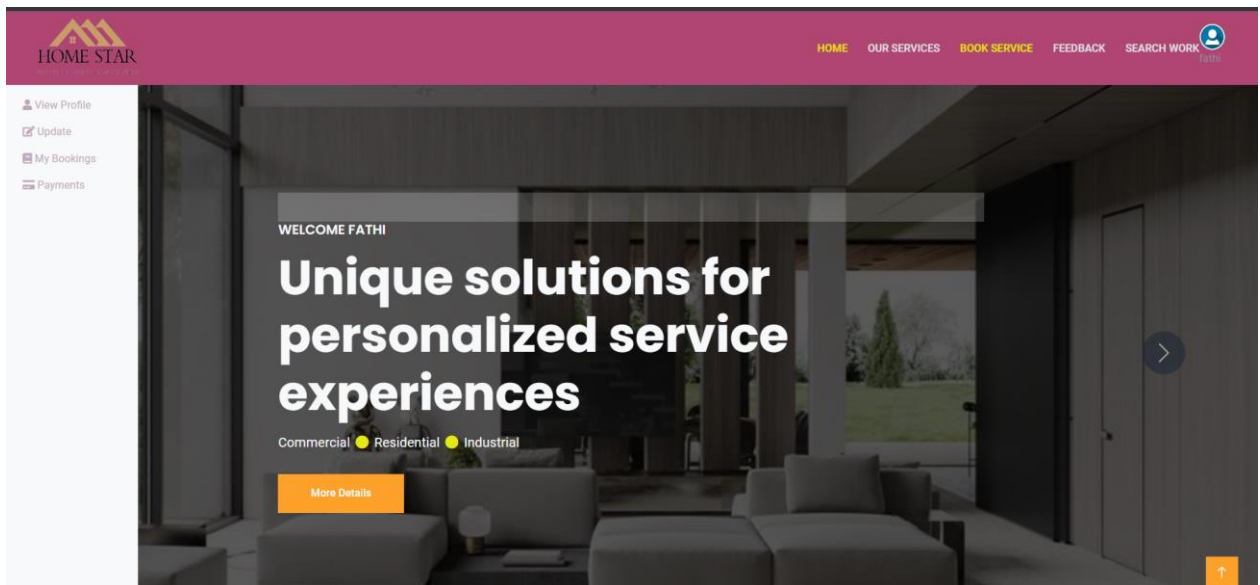


Figure 9.2.12: User Home Page

