

A green shield with red and white logo

AI-generated content may be incorrect.

**The University of Jordan**

**King Abdullah II School of Information Technology**

Department of Computer Information Systems

Supervised by: DR. Hamad Al-Sawalqah

**The Assemblers**

|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| **Mohammed Loai Abulathoa** | **2221273** |
| **Zaina Abu-Ghazzeh** | **0224066** |
| **Rahaf Elayyan** | **0228557** |
| **Heba Hamed** | **0226454** |
| **Rama Sharkas** | **0222532** |

Contents

[1.0 Project Initiation 5](#_Toc199624623)

[**1.1 Project Overview** 5](#_Toc199624624)

[**1.2 Problem definition** 5](#_Toc199624625)

[1.2.1 Problem Statement 5](#_Toc199624626)

[1.2.2 Issues (of the present situation) 6](#_Toc199624627)

[1.2.3 Objectives for Each Issue 8](#_Toc199624628)

[1.2.4 Requirements 9](#_Toc199624629)

[1.2.5 Constraints 11](#_Toc199624630)

[**1.3 Feasibility study** 11](#_Toc199624631)

[1.3.1 Technical Feasibility 11](#_Toc199624632)

[1.3.2 Operational Feasibility 12](#_Toc199624633)

[1.3.3 Economic Feasibility 12](#_Toc199624634)

[1.3.4 Schedule Feasibility 18](#_Toc199624635)

[1.3.5 Legal feasibility 19](#_Toc199624636)

[1.4 Recommended Solutions and expected project deliverables 20](#_Toc199624637)

[1.5 Local and Global Impact of the Proposed Solution 21](#_Toc199624638)

[**1.6 Naming Conventions and Definitions (Terms, Acronyms, and Abbreviations)** 21](#_Toc199624639)

[2.0 Project Management Plan 22](#_Toc199624640)

[**2.1 Project organization** 22](#_Toc199624641)

[**2.2 Roles and Responsibilities** 23](#_Toc199624642)

[**2.3 Software Process Model** 24](#_Toc199624643)

[**2.4 Tools And Techniques** 25](#_Toc199624644)

[**2.5 Word Breakdown** 26](#_Toc199624645)

[2.5.n Project Tasks 26](#_Toc199624646)

[2.5.n.1 Task Description & Breakdown 26](#_Toc199624647)

[2.5.n.2 Deliverables and Milestones 27](#_Toc199624648)

[2.5.n.3 Resources Needed (Skills, Hardware, and Software) 27](#_Toc199624649)

[2.5.n.4 Dependencies and Constraints 29](#_Toc199624650)

[**2.6 Assigning Team Members To Tasks** 31](#_Toc199624651)

[**2.7 Project Schedule (Gantt chart and PERT diagram)** 33](#_Toc199624652)

[**2.8 Risk Analysis and Plans** 35](#_Toc199624653)

[**2.9 Monitoring, Reporting, and Controlling Mechanisms** 39](#_Toc199624654)

[3.0 Software Requirements Specifications (SRS) 41](#_Toc199624655)

[3.1 System Stakeholders and Requirements Sources 41](#_Toc199624656)

[3.2 User Requirement Definition 42](#_Toc199624657)

[3.3 Use Case Diagrams 44](#_Toc199624658)

[3.4 System Functional Requirements Specifications 49](#_Toc199624659)

[3.5 Textual Description for Each Use Case 54](#_Toc199624660)

[3.6 Non Functional Requirements 63](#_Toc199624661)

[3.6.1 Performance Requirements 63](#_Toc199624662)

[3.6.2 Dependability Requirements 63](#_Toc199624663)

[3.6.3 Security Requirements 63](#_Toc199624664)

[3.6.4 Usability Requirements 63](#_Toc199624665)

[3.6.5 Operational & Environmental Requirements 63](#_Toc199624666)

[3.6.6 Maintainability Requirements 64](#_Toc199624667)

[3.7 Data Requirements 64](#_Toc199624668)

[4.0 Analysis And Design 68](#_Toc199624669)

[4.1 Activity Diagrams 68](#_Toc199624670)

[4.2 Sequence Diagrams 72](#_Toc199624671)

[4.3 ERD Design 96](#_Toc199624672)

[4.4 Architecture Design 97](#_Toc199624673)

[4.6 Graphical User Interface Design 98](#_Toc199624674)

[5.0 References: books and tools 106](#_Toc199624675)

# Project Initiation

## **1.1 Project Overview**

Being able to locate trustworthy home repair and maintenance services has been an obstacle for many. With **The Assemblers** homeowners will no longer struggle with finding what best meets their standards for a maintained and cleaner home. **The Assemblers** simplifies this process through linking homeowners with reliable service providers online. Whether homeowners are dealing with plumbing problems, air conditioning service, or regular upkeep, they can conveniently schedule services on demand, monitor technicians in real-time, and process secure payments, all from a single place. **The Assemblers** also help service providers manage their workload by enabling them to log onto appointments, accept task requests, and adjust their availability. To ensure that homes stay in top shape, the system includes automated reminders for planned maintenance. **The Assemblers** allows homeowners to choose the best experts for their needs with confidence, all thanks to user reviews and ratings. **The Assemblers** aims to improve home maintenance efficiency, dependability, and accessibility, making it easier for homeowners to get the help they need when they need it.

## **1.2 Problem definition**

### 1.2.1 Problem Statement

Finding trustworthy home maintenance and repair services can be a challenging and time-consuming task for homeowners. From searching for reliable professionals to scheduling appointments and ensuring smooth payments, the process often lacks convenience and transparency. Many homeowners also overlook essential maintenance tasks like AC servicing or plumbing inspections, leading to unexpected breakdowns and costly repairs.

On the other hand, service providers struggle with managing bookings, responding to job requests efficiently, and keeping track of their availability. Without a structured system, they risk missed opportunities, scheduling conflicts, and dissatisfied customers.

This project aims to bridge the gap by creating an intuitive web platform that connects homeowners with verified service providers. The platform will simplify home maintenance, ensuring homes stay in top condition while helping service providers manage their workload efficiently.

### 1.2.2 Issues (of the present situation)

* **Lack of Reliable and Verified Service Providers:** Homeowners often struggle to find trustworthy professionals. Many rely on word-of-mouth recommendations or online searches, which may not always be reliable.

Unverified service providers can lead to poor workmanship, fraud, or incomplete jobs.

* **Time-Consuming and Inconvenient Booking Process:** Traditional methods (calling different providers, negotiating prices, and scheduling visits) are inefficient and time-consuming. Many homeowners face delays due to unresponsive providers or last-minute cancellations.
* **Unclear Pricing and Hidden Costs:** There is often no standardized pricing, leading to unexpected costs for homeowners.

Some service providers charge different rates depending on the customer, leading to inconsistencies and lack of trust**.**

* **Lack of Real-Time Tracking and Communication:** Homeowners have no way to track the technician’s arrival, leading to uncertainty and wasted time.

Poor communication between customers and service providers results in missed appointments and misunderstandings about the scope of work.

* **Difficulty in Managing Payments Securely:** Many homeowners prefer cash payments, which can be inconvenient and risky.

Service providers may struggle with payment collection and delayed transactions.

There is no structured invoicing or transaction history for reference.

* **Limited Maintenance Awareness and Scheduling:** Many homeowners neglect routine maintenance tasks such as AC servicing, plumbing inspections, or electrical safety checks, leading to costly repairs later.

Without automated reminders, these essential tasks are often forgotten.

* **Inefficient Job Management for Service Providers:** Service providers struggle with managing multiple bookings, tracking job statuses, and updating their availability. Double bookings, scheduling conflicts, or last-minute cancellations lead to loss business and poor customer experience.
* **Lack of Customer Reviews and Ratings**: Homeowners have limited ways to assess a service provider’s credibility and past work.

Without a review system, service providers lack motivation to maintain high-quality service.

* **No Centralized Platform for Home Maintenance Needs:** Homeowners must rely on multiple sources (different websites, phone calls, or physical visits) to find and hire professionals.

There is no unified system that connects service providers and homeowners efficiently.

### 1.2.3 Objectives for Each Issue

1. Ensuring Reliable and Verified Service Providers through implementing an authentication process for service providers, such as background checks and trusted customer reviews along with providing a rating and feedback system to aid homeowners in choosing trusty professionals.

2. Streamlining the Booking Process by utilizing an easy-to-use online booking system that allows homeowners to schedule services quickly. In addition, enabling automated scheduling and notifications to minimize delays and cancellations.

3. Providing Transparent and Standardized Pricing by displaying clear pricings for different services to eliminate unknown costs and offering price estimates based on service type and location to guarantee fair and consistent pricing for all.

4. Implementing Real-Time progress tracking.

5. Ensuring Secure and Efficient Payment Management by organizing multiple secure payment options, including online transactions and cash.

6. Enhancing Maintenance Awareness and Scheduling through introducing automated reminders for routine maintenance tasks (e.g. AC servicing and plumbing inspections).

7. Optimizing Job Management for Service Providers via developing an interface to handle bookings, update availability, and track job statuses. Additionally, implementing a booking system with automated time-slot management helps lower scheduling conflicts and last-minute cancellations.

8. Integrating Customer Review and Rating System by allowing users to leave reviews and rate service providers based on job quality and competence. Furthermore, boosting high-quality service providers through a rating-based ranking system ensures homeowners’ ability to make informed decisions when choosing professionals.

9. Developing a Centralized Platform for Home Maintenance by creating a unified platform that connects homeowners with multiple service categories in one place.

### 1.2.4 Requirements

For the system to work properly, it needs to fulfill the following requirements:

• **User Accounts & Security:** Homeowners must be able to register and safely log in using

the system.

• **Service Booking:** Users must be able to book services by selecting the type of

maintenance, choosing a provider, and scheduling an appointment.

• **Real-Time Tracking:** The platform needs to offer real-time tracking to show the technician’s progress.

• **Secure Payments:** A payment system must be integrated to support transactions via credit

cards and other options like PayPal.

• **Service History Tracking:** Users should be able to view previously done services, including

details of completed maintenance.

**• Review & Rating System:** To guarantee quality and trust, users must be able to rate and

review service providers.

**• Automated Notifications:** The platform must send reminders and notifications for

upcoming maintenance tasks.

**• Service Provider Dashboard**: In order to handle reservations, get job requests, and update

availability, service providers need to have a dashboard.

**• Admin Panel:** To manage users, confirm service providers, and address problems, the

system needs to have an admin panel.

**• Client Dashboard:** To view past bookings, pending requests and upcoming bookings as well as book new services and track the progress of ongoing appointments.

### 1.2.5 Constraints

Technical Constraints

1. **Progress Tracking** – Requires real time updates from assembler, which the assembler may forget to do.
2. **Latency & Performance** – Ensuring quick response times for booking, payment, and tracking features requires efficient backend architecture.
3. **Data Security & Privacy –** Sensitive user information (addresses, payments) must be protected against breaches and unauthorized access.

## **1.3 Feasibility study**

### 1.3.1 Technical Feasibility

**1. Existing Infrastructure**

* **Hardware**: The team has access to necessary hardware resources, such as personal computers which are sufficient for building and hosting the platform, any device with internet access can use this website.
* **Software**: The team is proficient in Design languages (HTML, CSS, JavaScript for front-end).

**2. Technical Capabilities**

* The team has experience in:
  + UI / UX Design
  + Frontend development
  + SQL language

**3. Technology Availability**

* The required technologies and tools for the project are readily available in the market, including:
  + **Payment Gateways**: Visa, Mastercard and PayPal.

**4. Conclusion**

**The Assemblers** is technically feasible to develop with the team's current resources. Overall, the technical requirements align with the team's capabilities, they should be able to learn if needed, the project can proceed.

### 1.3.2 Operational Feasibility

The system is designed to run efficiently so that it provides smooth experience for all users. It provides a simple procedure for booking, live updates, and safe transactions, making it simple and reliable to use. Service providers can handle their assignments and timetables, maximizing efficiency.

### 1.3.3 Economic Feasibility

**a) Development Costs**

**Personnel**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Number of Members | Cost per Month per employee | Total Cost for 3 Months |
| Project Manager | 1 | 1800 JD | 5400 JOD |
| System Analyst/  Administrator | 1 | 1200 JD | 3600 JOD |
| Front-end Developer | 1 | 1500 JD | 4500 JOD |
| Back-end Developer | 1 | 1800 JD | 5400 JOD |
| UI/UX Designer | 1 | 1000 JD | 3000 JOD |
| QA Specialist | 1 | 1200 JD | 3600 JOD |

**Total Personnel Costs for 3 months = 25500 JOD.**

**Hardware and Software**

|  |  |  |  |
| --- | --- | --- | --- |
| Hardware/Software | Quantity | Cost | Total Cost |
| Acer Predator Helios Neo 16 (2024) NEW 14Gen Intel Core i9 14900HX 24-Cores w/ Nvidia RTX 4060 8GB DDR6 & 2.5K IPS 165Hz Display (including Windows 11) | 6 | 1200 JD | 7200 JOD |
| Cloud Services | 1 | 120 JD | 360 JD/3 months |
| Domain & Hosting | 1 | 7 JD/month | 21 JD/ 3 months |

**Total Hardware and Software Costs = 7581 JOD**

**Total Development Costs = 33081 JOD**

**b) Operating Costs**

|  |  |  |
| --- | --- | --- |
| Operating Cost | Cost per Month | Total Cost for 12 Months |
| Domain Name | 10 JD | 120 JOD |
| System Maintenance & Updates | 120 JD | 1440 JOD |
| Server Hosting | 40 JD | 480 JOD |
| Internet Connection | 28 JD | 336 JOD |

**Total Operating Costs = 2376 JOD**

**c) Tangible Benefits**

|  |  |
| --- | --- |
| Tangible Benefits | Estimated Savings |
| Increased Booking Efficiency | 18000 JD |
| Paperwork, printing, and storage costs reduction | 700 JD |
| Automated Payment & Scheduling Benefits | 7000 JD |

**Total Tangible Benefits = 25700 JD**

**d) Intangible Benefits**

* Increased accessibility for homeowners to find and book trusted service providers.
* Enhanced service provider efficiency through job automation and real-time tracking.
* Improved customer trust and pleasure through secure payments and rating systems.
* Business growth through data-driven insights into customer demand trends.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Year | Costs (JD) | Benefits (JD) | Discount (12%) | Accumulated Costs (JD) | Accumulated Benefits (JD) |
| Year 0 | 33081 | 0 | 1.000 | 33081 | 0 |
| Year 1 | 2376 | 25700 | 0.893 | 35241 | 23361 |
| Year 2 | 2447 | 28270 | 0.797 | 37262 | 46712 |
| Year 3 | 2520 | 31097 | 0.712 | 39155 | 70066 |
| Year 4 | 2596 | 34207 | 0.636 | 40928 | 93429 |
| Year 5 | 2674 | 37628 | 0.567 | 42589 | 116796 |

**e)** **Financial Projections & Return on Investment (ROI) Calculation:**

The system is expected to reach its break-even point in approximately 2 years, as accumulated benefits surpass accumulated costs by the end of Year 2.

**Lifetime ROI is:**

Lifetime ROI = (Estimated Lifetime Benefits - Estimated Lifetime Costs)/ Estimated Lifetime Costs = (116796 – 42589) / 42589 = 174%

**Annual ROI is:**

Annual ROI = Lifetime ROI / Project Duration (Years)

= 338% / 5 = 34.8%

**Net Present Value (NPV)** = Cumulative Benefits – Cumulative Costs

= 116796 – 42589 = 74207 JD

**The Assemblers** is economically feasible, with a solid ROI of 174% and an expected break even within 2 years. If we took the growing demand for digital service platforms in Jordan into account, this project has high likelihood for profitability and scalability, making it a sustainable long-term investment.

### 1.3.4 Schedule Feasibility

Tasks, Durations, and Dependencies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task No. | Description | Duration | Dependencies | Responsible staff | Resources |
| **T1** | Initial stakeholder meeting | 2 | None | All team members | None |
| **T2** | Requirements specifications | 6 | T1 | All team members |  |
| **T3** | System architecture design with the detailed one | 15 | T2 | Rama | LUCID CHART |
| **T4** | Designing Wireframes | 3 | T2,T3 | Rahaf | GoodNotes |
| **T5** | UI/UX Design prototyping | 13 | T2, T3 | Rahaf | Figma |
| **T6** | Frontend development | 18 | T4,T5 | Zaina, Rama, Heba Mohammed | Cursor |
| **T7** | System integration | 10 | T5, T6 | Mohammed, Heba |  |
| **T8** | Initial testing | 5 | T7 | All team members |  |
| **T9** | User acceptance testing | 11 | T7 | Heba, Rama |  |
| **T10** | Deployment planning | 5 | T8, T9 | Mohammed, Rahaf |  |
| **T11** | Deployment and monitoring | 8 | T10 | Heba, Zaina | NONE |

### 1.3.5 Legal feasibility

**1. Business Licensing & Registration:** we will need to register the platform as a business with Jordan’s Ministry of Industry, Trade, and Supply.

when the platform generates revenue (e.g., through service commissions), it must comply with taxation laws under the Income and Sales Tax Department.

**2. Service Provider Verification & Compliance:** Some home repair services, such as plumbing, electrical work, and AC maintenance, require licensed professionals. The platform should verify provider certifications and ensure they comply with Jordanian labor laws.

You might need to partner with Jordanian professional associations to ensure compliance.

**3. Contractual Agreements:** A Terms & Conditions agreement should define service expectations, provider responsibilities, and dispute resolution.

Liability clauses must be in place to protect homeowners, service providers, and the platform from legal issues.

**4. Consumer Protection & Data Privacy:** Jordan has a Cybercrime Law and Personal Data Protection regulations. The platform must secure user data, including payment details and personal information, to avoid legal penalties.

Transparency about how user data is collected and stored is crucial.

**5. Payment & Tax Regulations:** Online payments must comply with Jordanian e-commerce and financial transaction laws.If the platform handles transactions, it may need to register with the Central Bank of Jordan for financial compliance.

Service providers must declare their earnings and pay appropriate income taxes.

**6. Insurance & Liability:** Consider liability insurance for damages or accidents during services.

Service providers may need occupational insurance to cover injuries on the job.

## 1.4 Recommended Solutions and expected project deliverables

To address the challenges homeowners and service providers face in home maintenance and repair services, we propose developing a web platform that connects homeowners with verified service providers in Jordan. The platform will streamline the entire process, from booking services to secure payments and tracking job progress in real-time.

**Key components of the solution include:**

**User-Friendly Interface** – A seamless web experience for both homeowners and service providers.

**On-Demand & Scheduled Booking –** Users can request immediate service or schedule maintenance tasks in advance.

**Real-Time Technician Tracking –** progress tracking.

**Secure Online Payments –** Integration with licensed payment gateways for smooth transactions.

**Automated Maintenance Reminders and Notifications –** Notifications for routine home maintenance (e.g., plumbing checks, AC servicing), offers, and appointment reminders.

**Service Provider Management –** A dashboard for providers to manage job requests, availability, and earnings.

**Customer Reviews & Ratings –** A transparent feedback system to ensure quality service.

**Legal & Compliance Measures** – Verification of service provider licenses, data security, and compliance with Jordanian laws.

## 1.5 Local and Global Impact of the Proposed Solution

Locally, it will make it easier for homeowners to access quality service providers, ensuring that maintenance and repairs are done on time. It will also create job opportunities for qualified technicians by giving them a platform where they can find employment. Additionally, improved maintenance services will help homes stay in good condition, preventing costly future repairs. Globally, if the platform expands to other cities or countries, it can help homeowners worldwide locate reliable home maintenance services. The system can also be modified to other places by adding new service types and features according to regional needs. In general, this project has the potential to make the management of home maintenance services more convenient and efficient all over the world.

## **1.6 Naming Conventions and Definitions (Terms, Acronyms, and Abbreviations)**

**Terms:**

* **The Assemblers**: The proposed platform that connects homeowners with service providers for repairs and maintenance.
* **Service Provider**: A professional or company offering repair and maintenance services.
* **End-User**: The homeowner or customer using the platform to book services.

**Acronyms:**

* **DBMS** - Database Management System

**Abbreviations:**

* **JD** - Jordanian Dinars
* **DB** – Data base
* **UI** – User interface
* **UX** – User experience

# 2.0 Project Management Plan

## **2.1 Project organization**

**The Assemblers** works to ensure smooth execution, cost efficiency, and timely completion through following a collaborative structure within the team. The project mentioned requires the following structure to guarantee a fully operational and efficient system:

**- Project manager:** Oversees the overall progress of the project.

**- System Analyst:** Gathers user requirements from homeowners and service providers and conducts market research.

**- Front-end Developer:** Designs and implements the user interface of the web and responsive design.

**- Back-end Developer:** Develops APIs & database systems to manage services (e.g. payments, user profiles, authentication)

**- UI / UX Designer:** Creates designs and prototypes for the software for seamless user experience.

**- Quality Assurance (QA) Tester:** Conducts tests for all features to identify bugs in the system.

## **2.2 Roles and Responsibilities**

***2.2 Roles and Responsibilities***

**Project Manager**

**System Analyst**

**Front-end Developer**

**Back-end Developer**

**UI/UX Designer**

**Quality Assurance (QA)**

## **2.3 Software Process Model**

Incremental Development will be used in **The Assemblers** by the following process:

1. **Planning and Requirements Gathering:**

* Conduct meetings to gather core functionality needs.
* Document functional requirements and user stories.

1. **Initial Development and Testing:**

* Build core features: user login, booking system, and service tracking.
* Perform unit testing and bug fixing.

1. **Iteration and Refinement:**

* Integrate feedback into the next development cycle.
* Add advanced features like automated reminders and enhanced dashboards.

## **2.4 Tools And Techniques**

* **Communication:** Google meets, WhatsApp, and Google Doc to connect the development team, stakeholder. Communication tools help with communication among the team members and provide easier collaboration.
* **Code Review and Collaboration Platforms:** Cursor, ChatGPT, VS-Code for code development of processes and review of code.
* **Design Tools:** GoodNotes (for wireframes) and Figma (a design tool used by the UI/UX Designer) to create prototypes of the website’s interface.
* **Programming Languages:** HTML, CSS (bootstrap or plain), JavaScript.
* **Testing Tools:** Manual testing through visual studio code.
* **Flowcharts and Diagrams:** Lucidchart and StarUML used for creating diagrams, flowcharts and information representation graphically.

## **2.5 Word Breakdown**

### 2.5.n Project Tasks

Each task is broken down into specific activities necessary for the development of **"The Assemblers"** platform. These tasks will be executed in a structured manner to ensure a smooth development process.

### 2.5.n.1 Task Description & Breakdown

|  |  |  |
| --- | --- | --- |
| **Phase** | **Description** | **Task number** |
| **1.Specification** | 1.Define project goals and objectives and scope.  2.Gather stakeholder requirements.  3.Conduct feasibility analysis.  4.Validate with stakeholder. | T1  T2  T3  T4 |
| **2.Design** | 1.Design database schema (ERD)  2.Design user interface  3.Design system functionalities. | T5  T6  T7 |
| **3.Implementation** | 1.Implement UI  2.Implement functionalities  3.Link implemented sides | T8  T9  T10 |
| **4.Testing** | 1.Testing the functionality  2.Final validation | T11  T12 |

### 2.5.n.2 Deliverables and Milestones

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | **Phase** | | |  |  |  | | --- | --- | --- | | |  | | --- | | **Deliverables** |  |  | | --- | |  | |  |  | | --- | |  | | | **Milestones** | | --- |  |  | | --- | |  | |
| |  | | --- | | Project Planning |  |  | | --- | |  | | 1.Requirement Specification Document, Project Timeline | |  | | --- | | 1.Completion of project plan approval |  |  | | --- | |  | |
| |  | | --- | |  |   Architecture & Design | |  | | --- | | 2.System Architecture Document, UI Mockups |  |  | | --- | |  | | |  | | --- | | 2.Finalization of architecture and design |  |  | | --- | |  | |
| |  | | --- | | Implementation |  |  | | --- | |  | | |  | | --- | | 3. Developed core modules, Integrated front-end and back-end components | | 3. Completion of main feature implementation |
| Testing | |  | | --- | |  |   4.Test Reports, Bug Fixes, Security Audits | |  | | --- | |  |  |  | | --- | |  |   4.Successful completion of functional and security tests |

### 2.5.n.3 Resources Needed (Skills, Hardware, and Software)

**1. Skills:**

**(a) Full-stack Development:**

* Frontend Development: Proficiency in HTML, CSS, JavaScript for responsive and interactive UI development.
* Backend Development: Experience in developing APIs for frontend-backend communication.
* Version Control: Experience using Google Drive, for code collaboration and tracking issues.
* Collaboration among development teams to prevent code conflicts.

**(b) UI/UX Design:**

* Ability to design user-friendly interfaces using Figma.
* Knowledge of design principles and color theory to enhance aesthetics and accessibility.
* Feedback from stakeholders and collaboration with frontend developers.

**(c) Quality Assurance (QA) & Testing:**

* Experience in manual and automated testing.
* Experience in bug tracking and reporting using project management tools.

**(d) Project Management & Communication**:

* Strong communications skills to actively participate in team discussions.
* Ability to document processes using Lucid chart to create flowcharts and system diagrams.

**2. Hardware:**

* High performance laptops/desktops with sufficient CPU and memory.
* Reliable internet connection for seamless collaboration and cloud-based development.

**3. Software:**

1. Development Tools:

• VS Code for coding.

• GitHub for management and collaboration.

1. Design & Prototyping:

• Figma for UI/UX design.

• Lucidchart and StarUML for diagrams and flowcharts.

1. Testing & QA Tools:

• Manual Testing.

### 2.5.n.4 Dependencies and Constraints

**1.UI/UX Designer:**

Dependencies:

* Depends on collaboration with the frontend developer.

Constraints

* Limited availability of skilled UI/UX designers may affect the speed and quality of design deliverables.
* Time constraints may impact refinement process of design prototypes to meet project deadlines

**2.Full-stack Developer:**

Dependencies:

* Relies on the completion of system architecture design and API development tasks
* Depends on collaboration with the UI/UX designer

Constraints:

* Time constraints may limit the implementation of all desired features within project deadlines.

**3.Development IDE:**

Dependencies:

* Relies on the availability of necessary software licenses and subscriptions for development IDEs and collaboration tools.

Constraints:

* Resource constraints may limit the availability of licensed software tools and impact developers' productivity and efficiency.
* Time constraints may restrict the setup and customization of development environments within project timelines.

**4. Design Tools:**

Dependencies:

* Depends on collaboration with the UI/UX designer to ensure consistency and alignment between design tools and design deliverables.

Constraints:

* Availability of licensed design software tools may impact the accessibility and functionality of design tools for designers.
* Time constraints may limit the iteration and refinement process of design prototypes within project deadlines.

**5. Testing Tools:**

Dependencies:

* Relies on the completion of frontend development tasks
* Depends on collaboration with developers and QA teams

Constraints:

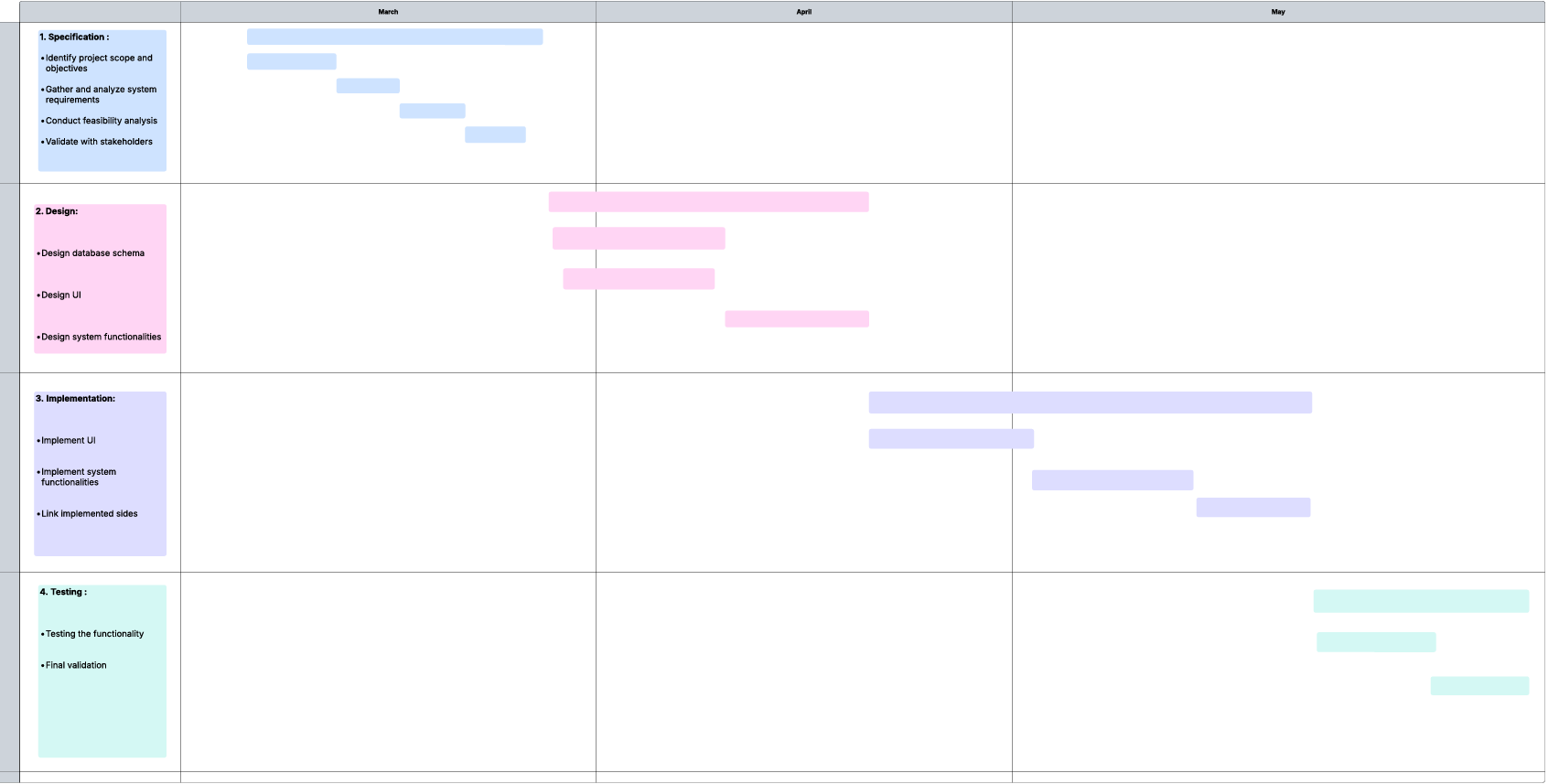
* Availability of licensed testing software tools may impact the effectiveness and coverage of testing activities.
* Time constraints may limit the execution and reporting of test results within project timelines.

## **2.6 Assigning Team Members To Tasks**

|  |  |  |
| --- | --- | --- |
| **Task** | **Assigned To** | **Description** |
| **UI/UX Design** | Rahaf | * Design the website’s layout, pages, components * create sitemaps * create wireframes |
| **Frontend Development** | Zaina, Rama, Heba, Mohammed | * Implement UI using HTML, CSS * develop responsive and interactive components |
| **QA Tester** | All Team Members | * Test case management, execution, and reporting. |
| **Project Manager** | Heba | * Oversees the overall progress of the project |
| **System Analyst** | Mohammed | * Gather user requirements from homeowners and service providers and conducts market research. |

## **2.7 Project Schedule (Gantt chart and PERT diagram)**

**(Gantt Char)**

****

**(PERT diagram)**

**A diagram of a flowchart

AI-generated content may be incorrect.**

## **2.8 Risk Analysis and Plans**

**Risk Identification:**

|  |  |
| --- | --- |
| Risk Type | Possible Risks |
| Technical Risks | 1. System crashes due to high demand; if several users try to simultaneously book services, the system may crash & become unresponsive. 2. Bugs in real-time tracking or automated reminders; tracking may lag or fail causing confusion for users. |
| Security Risks | 1. Payment fraud and unauthorized transactions; using stolen credit cards or fraudulent payment methods to book services, leading to financial loss. 2. Data breaches exposing customer and service provider information; personal details such as addresses, phone numbers, and payment information being stolen. |
| Project Schedule Risks | 1. Poor project management leading to missed deadlines; lack of structured planning, unclear responsibilities, or inefficient communication could slow progress. 2. Unexpected coding or integration challenges may extend development timelines.; if found late in testing, the project may require extra development time. |
| Resource Risks | 1. Overworked team members lead to burnout; if team members are overloaded with tasks due to an aggressive development timeline, productivity and code quality may suffer. 2. Lack of post-launch support for bug fixes; support and attention to fixing any bugs is crucial, offering the required resources and people to take responsibility for these things could be challenging. |

**☹**

**Risk Analysis:**

|  |  |  |
| --- | --- | --- |
| Risk | Probability | Effect |
| 1. System crashes due to high demand | **Medium** | **High** |
| 1. Bugs in real-time tracking or automated reminders | **Medium** | **Medium** |
| 1. Payment fraud and unauthorized transactions | **Medium** | **High** |
| 1. Data breaches exposing customer and service provider information | **Medium** | **High** |
| 1. Poor project management leading to missed deadlines | **Medium** | **Medium** |
| 1. Unexpected coding or integration challenges | **High** | **Medium** |
| 1. Overworked team members leading to burnout | **High** | **High** |
| 1. Lack of post-launch support for bug fixes | **Medium** | **High** |

**Risk Plans:**

**Contingency Plan for High Impact and High Probability Risks**

**1.Technical Risks:**

1.1 System Crashes Due to High Demand

**Contingency Actions:**  
▪ **Scalable Infrastructure:** Use cloud platforms with auto-scaling to manage spikes in traffic.  
**▪ Load Testing:** Conduct stress and performance testing to identify breaking points.  
▪ **Queueing Mechanisms**: Introduce queuing systems for user requests when traffic exceeds system capacity.

1.2 Bugs in Real-Time Tracking or Automated Reminders

**Contingency Actions:**  
▪ **Unit and Integration Testing**: Specifically focus on location-based features and scheduled reminders.  
▪ **Fallback Features**: Include manual options or alternate notification methods.  
▪ **Bug Logging System**: Implement real-time error logging and automated alerts.

**2. Security Risks**

2.1 Payment Fraud and Unauthorized Transactions

**Contingency Actions:**  
**▪ Secure Payment Gateways:** Integrate gateways that support fraud detection.  
**▪ Transaction Monitoring:** Flag suspicious patterns and notify the admin immediately.  
**▪ User Verification:** Add verification steps like email or phone OTPs during critical operations.  
**▪ Refund & Freeze Protocols:** Have policies in place to temporarily freeze suspect transactions.

2.2 Data Breaches Exposing Personal Information

**Contingency Actions:**  
**▪ Enhanced Security Protocols.**

▪ **Incident Response Plan.**

▪ **Employee Training:** Conduct regular training for employees on best practices for data security and recognizing phishing attempts.

**▪ Penetration Testing:** Conduct regular vulnerability assessments by third-party security experts.

**3. Project Schedule Risks**

**3.1 Unexpected Coding or Integration Challenges**

**Contingency Actions:**

**▪ Development Buffers:** Include time cushions for integration tasks in the project timeline.

**▪ Prototyping:** Develop early prototypes for high-risk modules to expose potential integration problems.

**▪ Documentation and Standards:** Use consistent API and integration documentation to reduce friction.

**4. Poor Project Management Leading to Missed Deadlines**

**Contingency Actions:**

**▪ Agile Methodology:** Break work into manageable sprints with review checkpoints.

▪ **Clear Deliverables:** Define detailed tasks and deadlines for each project phase.

**▪ Progress Monitoring:** Use tools like Jira or Trello for visual tracking.

▪ **Scope Management:** Reassess non-essential features if deadlines are at risk

## **2.9 Monitoring, Reporting, and Controlling Mechanisms**

**Monitoring Mechanism:**

* **Progress Tracking**: Schedule reoccurring meetings to check in on completed tasks, track development milestones, and make sure all is going according to schedule.
* **User Engagement Analytics**: Use analytics to see how homeowners and service providers use the platform and where/how we can make things smoother.
* **System Performance Monitoring**: Monitor things like server uptime, loading speeds, and API requests to catch and fix performance issues before they arise.
* **Security Audits**: Run monthly security checks to identify vulnerabilities and implement necessary security updates.

**Reporting Mechanisms:**

* **Weekly Progress Reports**: Each team member will share a quick report every week on what they’ve worked on, any roadblocks they’ve hit, and what’s next on the to-do list.
* **Bug & Issue Reports:** Log every reported bug or problem into a central system so we can prioritize and fix them effectively.
* **User Feedback Reports**: Regularly collect feedback from users through surveys, reviews and ratings.
* **Financial Reports**: Monthly financial reports will help us track costs, revenue, and spending so we stay financially stable.

**Controlling Mechanisms:**

* **Issue Tracking System**: A ticketing system helps track and prioritize bugs, feature requests, and user-reported problems so nothing is missed.
* **Performance Benchmarks:** Performance benchmarks defining how fast pages should load and how quickly bookings should be processed and test them regularly.
* **Risk Review Sessions**: Every two weeks, reassess potential risks, discuss any new challenges, and modify the strategy as needed.
* **Compliance Checks**: Regular checks will be made to ensure we’re following security standards, payment regulations, and service provider verification rules.

# 3.0 Software Requirements Specifications (SRS)

## 3.1 System Stakeholders and Requirements Sources

**System stakeholders**

|  |  |  |
| --- | --- | --- |
| **Stakeholder’s  ID** | **Stakeholder** | Description |
| St1 | Homeowners  (Clients) | Individuals who use the website to request home maintenance and repair services. |
| St2 | Service Providers  (Assemblers) | Verified professionals who offer maintenance services and receive service requests. |
| St3 | Website Admin | Responsible for managing users, services, verifying service providers and overall platform control. |
| St4 | Business Owners | The team who are funding and managing the platform. |
| St5 | Development team | The group responsible for designing, implementing and maintaining the system. |

**Source of requirements**

|  |  |
| --- | --- |
| Source | Details |
| Stakeholders’ interviews | Input from potential users and supervisors to identify needs. |
| Team brainstorming sessions | Collaborative sessions between developers and designers to define system features. |
| Project supervisor guidance | Instructor feedback helped refine and validate technical and functional requirements. |
| Analysis and Research | Competitor analysis, market research, and legal requirements. |

## 3.2 User Requirement Definition

**3.2.1** The system shall allow clients to register, log in, and manage their personal profiles (name, phone, address, password, profile picture).

**3.2.2** The system shall allow Assemblers to register and create a professional profile including services offered, certifications, availability, and work area.

**3.2.3** The system shall allow clients to browse available services and filter them by category.

**3.2.4** The system shall allow clients to send service requests by selecting service type, preferred date/time, and location.

**3.2.5** The system shall notify Assemblers of incoming service requests and allow them to accept or reject requests.

**3.2.6** The system shall update the status of each service request (pending, accepted, in-progress, completed, cancelled) and notify users of any changes.

**3.2.7** The system shall allow clients to rate and review Assemblers after job completion.

**3.2.9** The system shall allow Assemblers to manage their service availability calendar.

**3.2.10** The system shall allow the Admin to approve or suspend Assembler accounts.

**3.2.11** The system shall allow Admins to manage user accounts (edit/delete), handle reports, and monitor platform activities.

**3.2.12** The system shall provide the Admin with access to dashboards showing number of bookings, active users, and top-rated Assemblers.

**3.2.13** The system shall allow Assemblers to view their job history.

**3.2.14** The system shall ensure data security and restrict access based on user roles (Client, Assembler, Admin).

**3.2.15** The system shall provide a secure login and password reset mechanism.

## 3.3 Use Case Diagrams

A diagram of a company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a person

AI-generated content may be incorrect.

## 3.4 System Functional Requirements Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req ID | Requirement Description | Priority | Increment | Stakeholder ID |
| 1.0 | As a homeowner, I want to be able to register for an account and log in securely and be able to retrieve ,y password in case I forgot it in order to access the platform's services . | 1 | Must include email verification and provide homeowner with forgot password services. | St1 |
| 1.1 | As a service provider, I want to be able to register on the platform and submit my credentials for verification. | 1 | Must include email verification and enable user to submit required documents. | St2 |
| 1.2 | As a homeowner, I want to be able to browse available home maintenance services by category , | 2 | Allow homeowner to browse using categories. | St1 |
| 1.3 | As a homeowner, I want to be able to view service provider profiles, including their services offered, pricing information , ratings, and reviews, so that I can choose the best provider for my needs. | 2 | Allow homeowner to view service provider profile details , reviews ,services and service pricing. | St1 |
| 1.4 | As a homeowner, I want to be able to book home maintenance services by selecting the service type, choosing a preferred provider, and scheduling an appointment at a convenient time, so that I can easily arrange for necessary home repairs . | 1 | Allow homeowner to select service type , pick the provider , schedule appointments and book service. | St1 |
| 1.5 | As a homeowner, I want to be able to track the assigned technician's progress, so that I can manage my time effectively and prepare for their arrival. | 2 | Enable tracking of assembler using progress bar. | St1 |
| 1.6 | As a homeowner, I want to be able to make secure payments for services rendered through the platform using various methods (e.g., credit/debit cards) and view the price breakdown so that financial transactions are safe, convenient, and traceable. | 1 | Must include payment verification and secure transactions. | St1 |
| 2.0 | As a homeowner, I want to be able to view my complete service history, including details of services booked, provider information, dates ,so that I can keep track of past services . | 3 | Enable homeowner to view service history , booking details ,dates . | St1 |
| 2.1 | As a homeowner, I want to be able to rate and write reviews for service providers after a service is completed. | 2 | Enable homeowner to rate and write reviews. | St1 |
| 2.3 | As a homeowner, I want to receive automated notifications about booking confirmations, and reminders for scheduled or routine maintenance. | 2 | Provide notification system to send notifications to user regarding their booking details. | St1 |
| 2.4 | As a service provider, I want to have a dedicated dashboard where I can manage my profile, set my availability, view and accept/ reject job requests, manage my schedule so that I can efficiently run my business on the platform. | 1 | Enable service provider to manage profile and add required details. | St2 |
| 2.5 | As an administrator, I want to have a comprehensive admin panel to manage user accounts (homeowners and service providers), oversee service provider verification, manage service categories, monitor platform activity, handle disputes, manage content, and configure system settings, so that I can ensure the platform operates smoothly, securely, and effectively. | 1 | Provide administrators with tools to manage user accounts. | St3 |
| 2.6 | As a homeowner, I want to see clear pricing information or estimates for services before booking, so I can make informed financial decisions and avoid unexpected costs. | 3 | Allow homeowner to view pricing information and estimates. | St1 |
| 2.7 | As an administrator, I want to be able to review and verify service provider applications, including background checks and certifications, so that only trustworthy and qualified professionals are listed. | 1 | Provide administrators with tools to review and verify applications. | St3 |
| 2.8 | As a service provider, I want to receive notifications for new job requests, booking confirmations, cancellations so that all users stay informed and manage schedules effectively. | 1 | Provide service provider with notification system for new job requests , booking confirmations and payment receipts. | St2 |
| 2.9 | As a homeowner, I want to be able to contact headquarters In cases of emergencies and misunderstandings in order to receive the help I need. | 1 | Provide homeowner with contact details for headquarters | ST1 |
| 3.0 | As a homeowner, I want to be able to edit my profile in order to display contact information when the assembler needs to reach me. | 2 | Allow homeowner to edit their profile. | ST1 |

## 3.5 Textual Description for Each Use Case

|  |  |
| --- | --- |
| Use Case Name | Log in |
| Actor | Client (St1) |
| Pre-condition | User has a registered account |
| Stimulus | User selects "Log in" and submits credentials |
| System Behavior | **Basic flow:**  1. User provides credentials (email/phone + password) 2. System verifies credentials 3. Successful: Grants access to dashboard 4. Failed: Displays error message  **Alternative Flow (Forgot password):**  1. User requests reset and chooses where to receive the OTP 2. System sends OTP  3. User sets new password |
| Post-condition | **Success:** User gains access to their account dashboard, and can view booking history, booking a service, upcoming bookings, editing profile, notifications, rating and reviewing.  **Failure:** System displays error message. |

|  |  |
| --- | --- |
| Use Case Name | Book a Service |
| Actor | Client (St1) |
| Pre-condition | User is logged in |
| Stimulus | User selects "Book a Service" from dashboard |
| System Behavior | **Basic flow:**  1. User selects service type (e.g., plumbing, electrical) 2. System displays available providers 3. User selects provider and time slot 4. System processes payment 5. Booking request registered  6. Booking confirmation guaranteed  **Alternative Flow (Cancel booking):**  User cancels before service date. |
| Post-condition | **Success:** - Booking request added to user's history - Notification sent to provider/user - Payment recorded  - User can view request status.  - After booking confirmation, user can track, rate and review the service.  **Failure:** - Error message displayed (e.g., "Slot unavailable") |

|  |  |
| --- | --- |
| Use Case Name | Log in |
| Actor | Service Provider “Assembler” (St2) |
| Pre-condition | Provider has a registered account |
| Stimulus | The provider selects "Log In" and enters credentials |
| System Behavior | **Basic flow:** 1. System verifies credentials 2. Success: Grants access to provider dashboard 3. Failure: Displays error message **Alternative flow (Forgot Password):** 1. Assembler requests reset and chooses where to receive the OTP 2. System sends OTP  3. Assembler sets new password |
| Post-condition | **Success:**  Assemblers gain access to their account dashboard menu which includes profile overview, appointments schedule, upcoming and previous jobs, job requests, and notifications.  **Failure:** System displays error message. |

|  |  |
| --- | --- |
| Use Case Name | Manage Job Requests |
| Actor | Service Provider “Assembler” (St2) |
| Pre-condition | Assembler is logged in and has job requests |
| Stimulus | Assembler views pending job requests |
| System Behavior | 1. System displays request details (location, service type, client info) 2. Assembler selects Accept/Reject 3. System updates booking status |
| Post-condition | **Accepted:** - Job added to schedule - Client notified  **Completed:**  - Mark job as done  **Rejected:**  -Request discarded -Option to contact client |

|  |  |
| --- | --- |
| Use Case Name | Log in |
| Actor | Website Admins (St3) |
| Pre-condition | Admin account exists |
| Stimulus | Admin enters credentials in login portal |
| System Behavior | 1. System verifies credentials 2. Success: Grants access to Admin dashboard 3. Failure: Displays error message |
| Post-condition | Admins gain access to their account dashboard where they can edit their profile, view Analytics and Service details.  Admins can manage users (view or ban them) and review Assembler applications, deciding whether to accept or reject them. |

|  |  |
| --- | --- |
| Use Case Name | Become an Assembler |
| Actor | User (Potential Service Provider) |
| Pre-condition | The user has accessed the platform via browser. |
| Stimulus | User selects "Become an Assembler" option |
| System Behavior | **Basic Flow:** 1. System displays requirements  2. Applicant enters personal information and uploads required documents  3. Applicant answers qualification questions  4. Submits application  5. Accepts Privacy Policy, Terms & Conditions  6. The system sends applications to the admin, who can review them and make a decision, which will then be sent to the applicant.  **Alternative Flow:**  Incomplete application: System prompts for missing items |
| Post-condition | **Success:** - Application submitted to admin queue  - Confirmation sent to applicant - Admin can approve/reject  **Failure:** - Error message specifies missing items |

|  |  |
| --- | --- |
| Use Case Name | Checkout Booking |
| Actor | Client (St1) |
| Pre-condition | 1.User is logged in  2.Has selected a service  3.Valid payment method |
| Stimulus | User clicks “Checkout” after service selection |
| System Behavior | **Basic Flow (Logged In):** 1. Displays booking summary with options to edit details or change the assembler 2. Displays price breakdown 3. User enters payment method  4. After terms agreement, booking is confirmed  **Alternative Flows:**  - (Not logged in) Redirects to Log In page  - (Incomplete payment details) Error message specifies missing items |
| Post-condition | **Success:** - Booking confirmed  **Failure:** 1. Redirected to login  2. Missing items specified |

## 3.6 Non Functional Requirements

### 

### 3.6.1 Performance Requirements

* The system should load any page within **4** seconds.

### 3.6.2 Dependability Requirements

* The system should be available **95%** of the time (excluding updates).

### 3.6.3 Security Requirements

* Passwords must be in a **secure, encrypted** form.
* Each user role (Client, Assembler, Admin) must only access features they’re allowed to.

### 3.6.4 Usability Requirements

* The platform must work on desktop.
* New users should be able to request a service without help.
* The interface must use clear and simple language (English).

### 3.6.5 Operational & Environmental Requirements

* The platform should run online 24/7.
* It should work well even with slow internet (5 Mbps).
* It must run in modern browsers like Chrome or Edge.

### 3.6.6 Maintainability Requirements

* The system should be easy to update without affecting other features.
* Error messages and logs must help developers identify problems quickly.
* The database design must be well organized for future changes.

## 3.7 Data Requirements

**1. Entities and Data Fields**

**User:**

Common table for Clients, Assemblers, and Admins.

* UserID (PK, unique)
* Name (varchar)
* Email (varchar, unique)
* Password (varchar, encrypted)
* Phone (varchar)
* Role (varchar: Client / Assembler / Admin)
* Location (varchar – Work Area or Address)
* RegistrationDate (date)
* Status (varchar: Active, Suspended)

**Application:**

Submitted by users who want to become Assemblers.

* ApplicationID (PK)
* UserID (FK → User)
* Resume (varchar – file path or URL)
* Answers (text – qualification form)
* Status (varchar: Submitted, Under Review, Accepted, Rejected)

**Job:**

Service orders requested by Clients.

* JobID (PK)
* ClientID (FK → User)
* AssemblerID (FK → User, nullable)
* Title (varchar – service type)
* Description (text)
* Date (date – scheduled)
* Time (time)
* Location (varchar)
* Status (varchar: Pending, Accepted, On the Way, On Site, Completed, Cancelled)

**Request:**

Tracks job requests and assembler responses.

* RequestID (PK)
* JobID (FK → Job)
* ClientNote (text – optional)
* AssemblerNote (text – optional)
* Status (varchar: Pending, Approved, Denied)

**Schedule:**

Availability of each Assembler.

* ScheduleID (PK)
* AssemblerID (FK → User)
* Date (date)
* IsAvailable (boolean)

**Rating:**

Client feedback after job completion.

* RatingID (PK)
* JobID (FK → Job)
* ClientID (FK → User)
* AssemblerID (FK → User)
* Score (int: 1–5)
* Review (text)
* DateSubmitted (date)

**Payment**

Payment details per job (future extension ready).

* PaymentID (PK)
* JobID (FK → Job)
* Method (varchar: Cash, Card, Transfer, etc.)
* Amount (decimal)
* Status (varchar: Paid, Pending)
* Details (text – optional)

**Notification:**

Messages sent to users about system events.

* NotificationID (PK)
* UserID (FK → User)
* Message (text)
* Timestamp (datetime)
* IsRead (boolean)

# 4.0 Analysis And Design

## 4.1 Activity Diagrams

4.1.1 Admin Dashboard Activity

Admin Login

Accept or Reject ?

View application

Open assembler application

View Announcements

Access Admin Dashboard

View Summary Overview

Open "User Management"

Open "Service Requests"

View All Requests

Drop

view

Select User Type (Client or Assembler)

Client

Assembler

approve

ban

reject

Update User Status

4.1.2 Client Dashboard Activity

Client login

Cancel

View progress

Open “upcoming requests

Open “pending requests”

Book again

View booking details

Open “past bookings”

Submit booking

Select location & payment

Choose assembler

Book a new service

Return to dashboard

Edit profile

Access client dashboard

4.1.3 Assembler Dashboard Activity

Assembler login

Access assembler dashboard

View job progress

Open “pending jobs”

Return to dashboard

Edit profile

View schedule

Open “upcoming jobs”

View status bar

View job details

Accept or deny?

Deny

Accept

4.1.4 Visitor Activity on Main Page

Apply to become an assembler

Help center

View assemblers

Log in

View services

Main page

## 4.2 Sequence Diagrams

A grid of graph paper with text

AI-generated content may be incorrect.

A diagram of a diagram

AI-generated content may be incorrect.

A diagram of a company

AI-generated content may be incorrect.

A diagram of a website

AI-generated content may be incorrect.

A diagram of a website

AI-generated content may be incorrect.

A diagram of a website

AI-generated content may be incorrect.

A diagram of a profile

AI-generated content may be incorrect.

A screen shot of a screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A diagram of a payment method

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A diagram of a person

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A diagram of a diagram

AI-generated content may be incorrect.

A diagram of a schedule

AI-generated content may be incorrect.A diagram of a system

AI-generated content may be incorrect.A graph with text and a diagram

AI-generated content may be incorrect.

A graph with text and a line

AI-generated content may be incorrect.

A diagram of a job

AI-generated content may be incorrect.A grid with a diagram

AI-generated content may be incorrect.A diagram of a graph

AI-generated content may be incorrect.A screenshot of a graph

AI-generated content may be incorrect.A screenshot of a project management system

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer screen

AI-generated content may be incorrect.A screen shot of a notification

AI-generated content may be incorrect.A diagram of a profile

AI-generated content may be incorrect.

A screen shot of a notification

AI-generated content may be incorrect.

A diagram of a graph

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

A graph with text and a diagram

AI-generated content may be incorrect.A diagram of a application

AI-generated content may be incorrect.A diagram of a project

AI-generated content may be incorrect.A graph with text and arrows

AI-generated content may be incorrect.A diagram of a program

AI-generated content may be incorrect.A screen shot of a computer screen

AI-generated content may be incorrect.

A diagram of a service

AI-generated content may be incorrect.

A diagram of a user flow

AI-generated content may be incorrect.

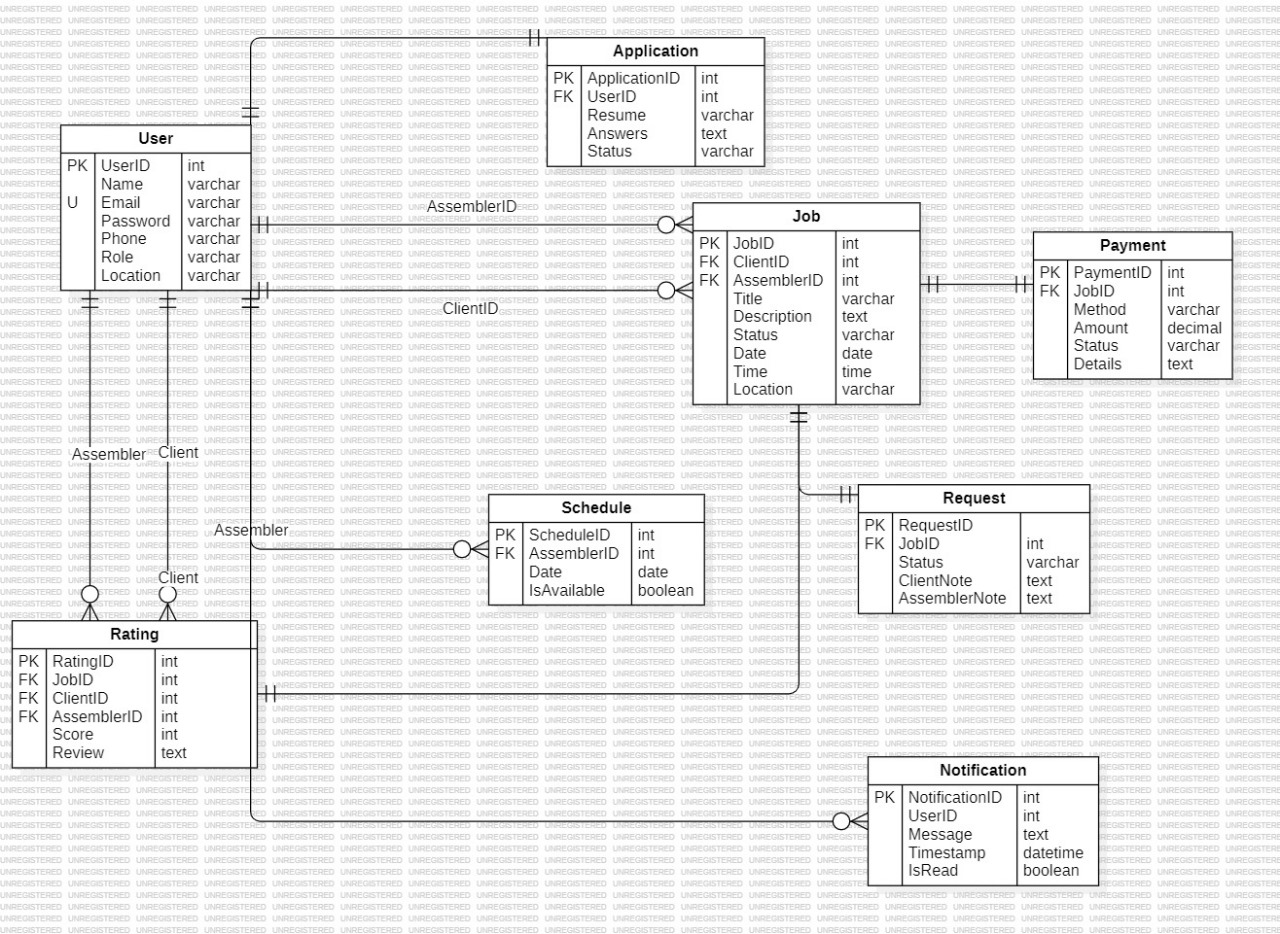
A diagram of a assembly

AI-generated content may be incorrect.

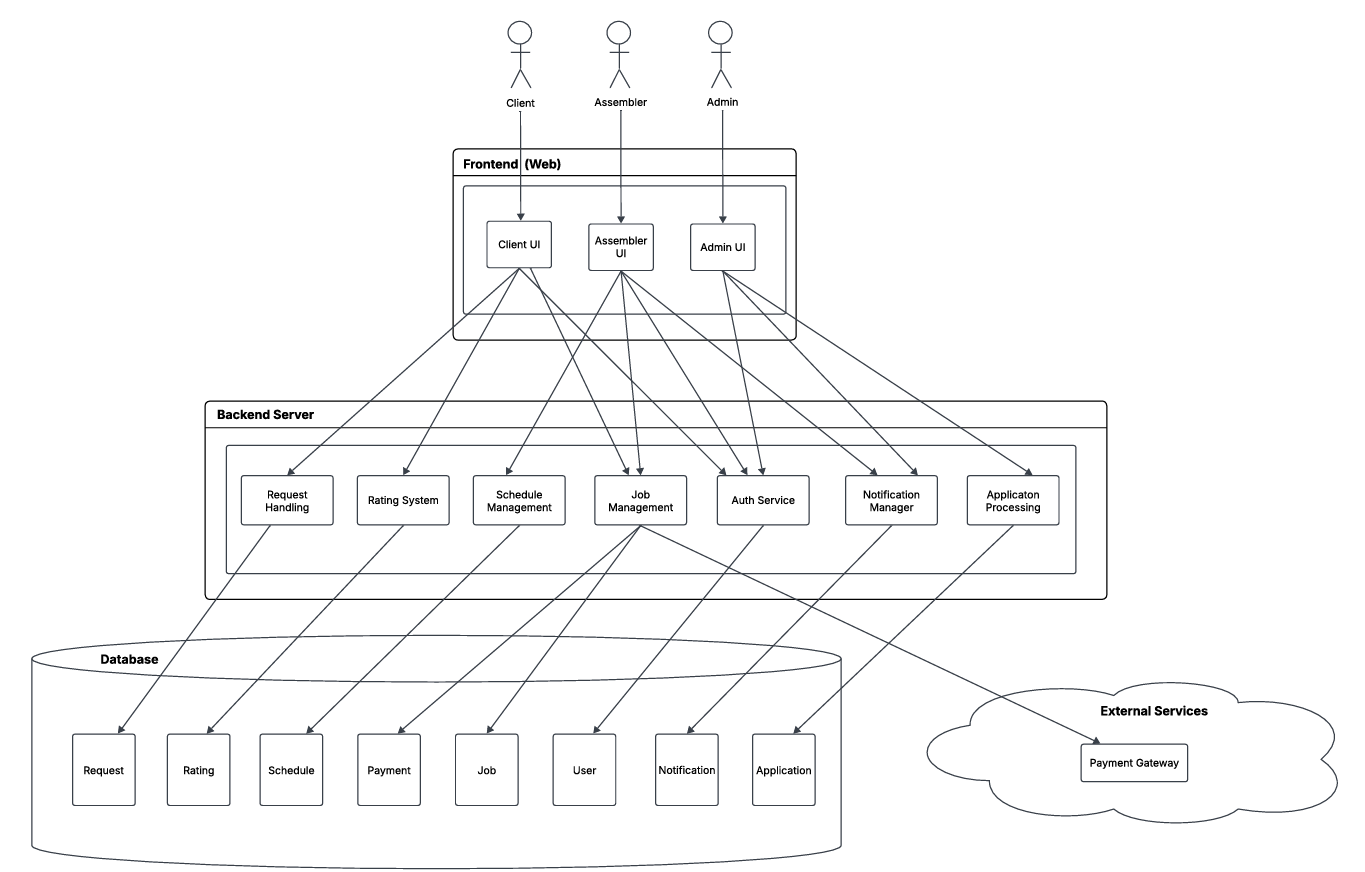
A diagram of contact form

AI-generated content may be incorrect.

## 4.3 ERD Design

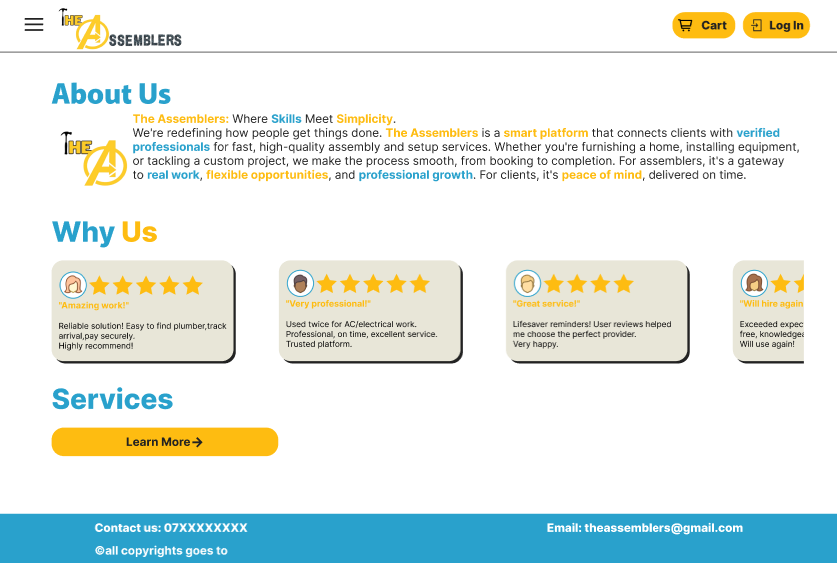


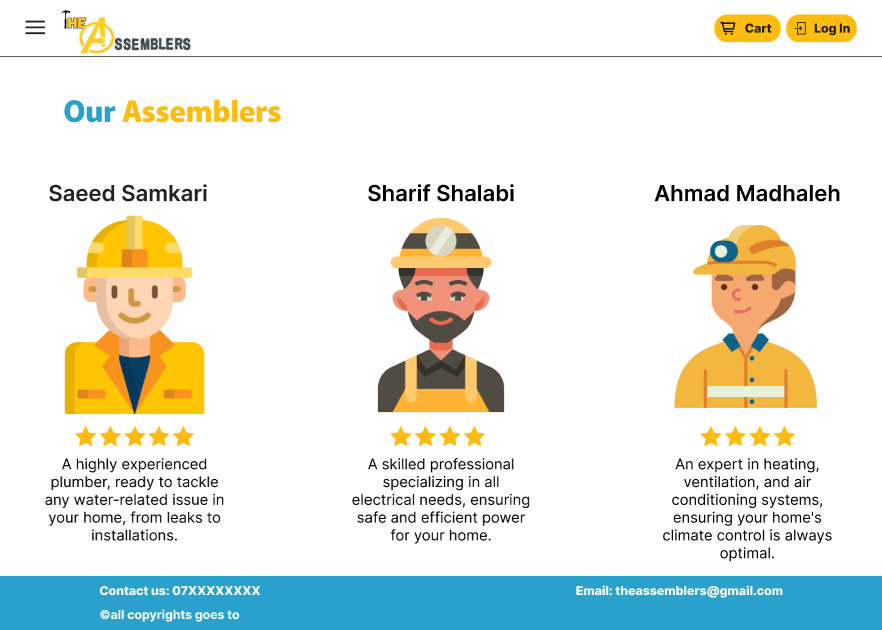
## 4.4 Architecture Design

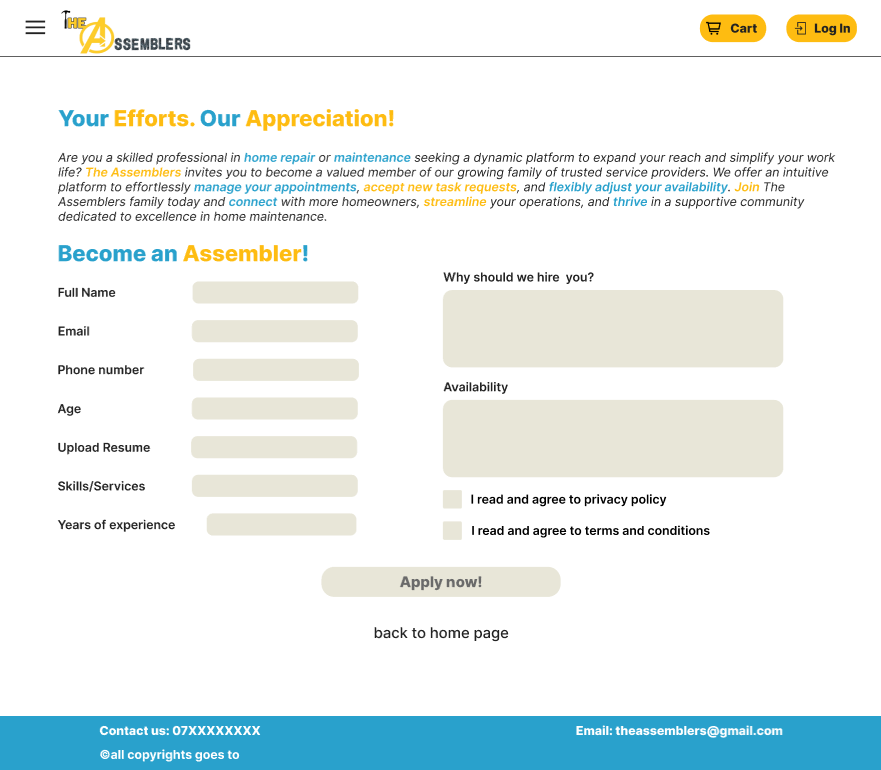


## 4.6 Graphical User Interface Design

Main Page



Our Assemblers

Become an Assembler

A screenshot of a phone

AI-generated content may be incorrect.Menu (Part of the pages)

Log In

A screenshot of a login form

AI-generated content may be incorrect.

A screenshot of a login form

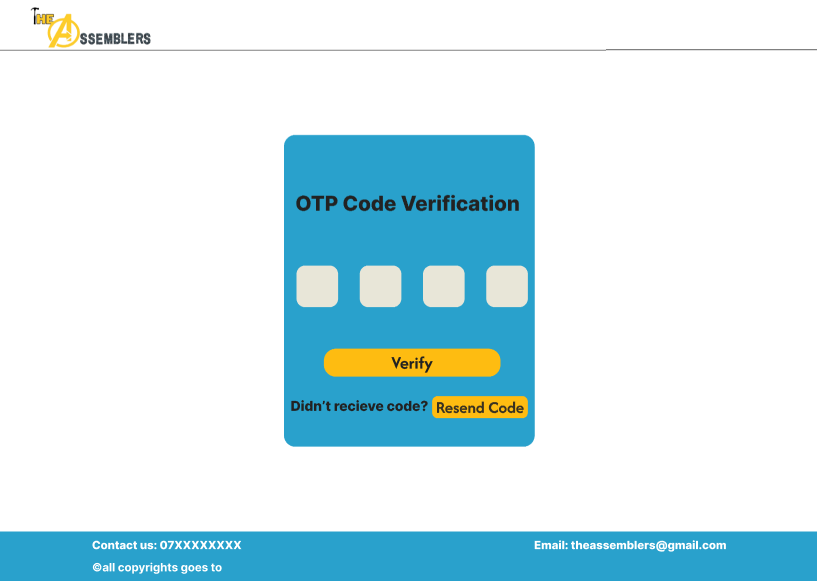
AI-generated content may be incorrect.Sign Up

Forgot Password

A screenshot of a computer screen

AI-generated content may be incorrect.

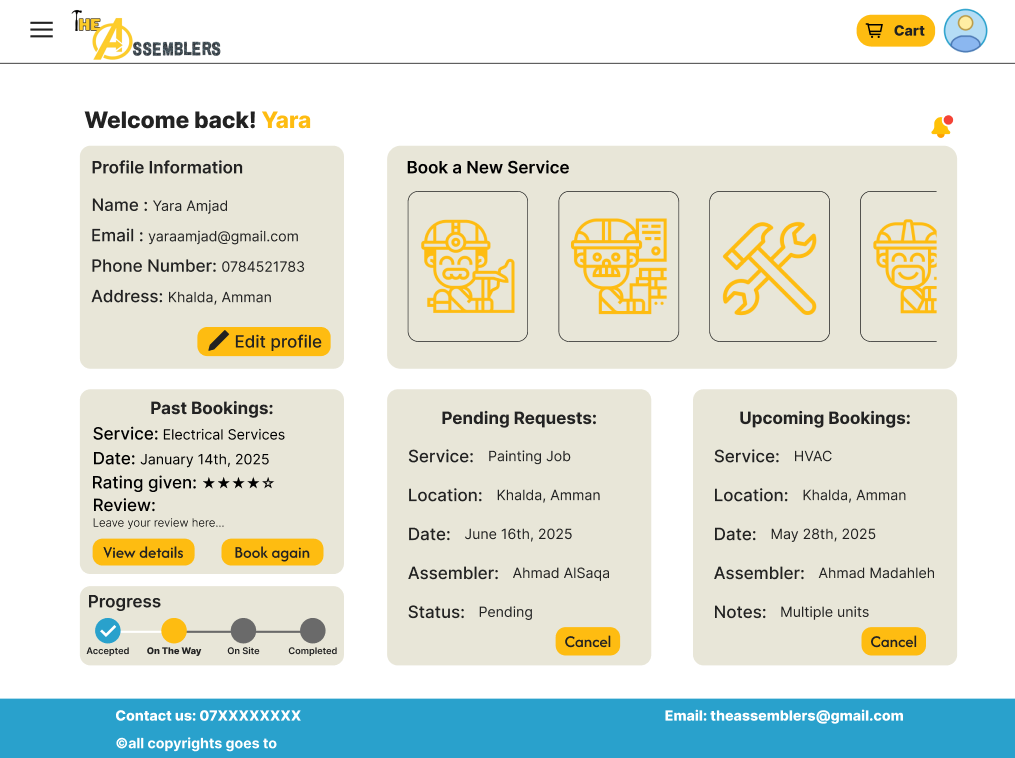
Verification Code



A screenshot of a computer screen

AI-generated content may be incorrect.Create New Password

Client Dashboard



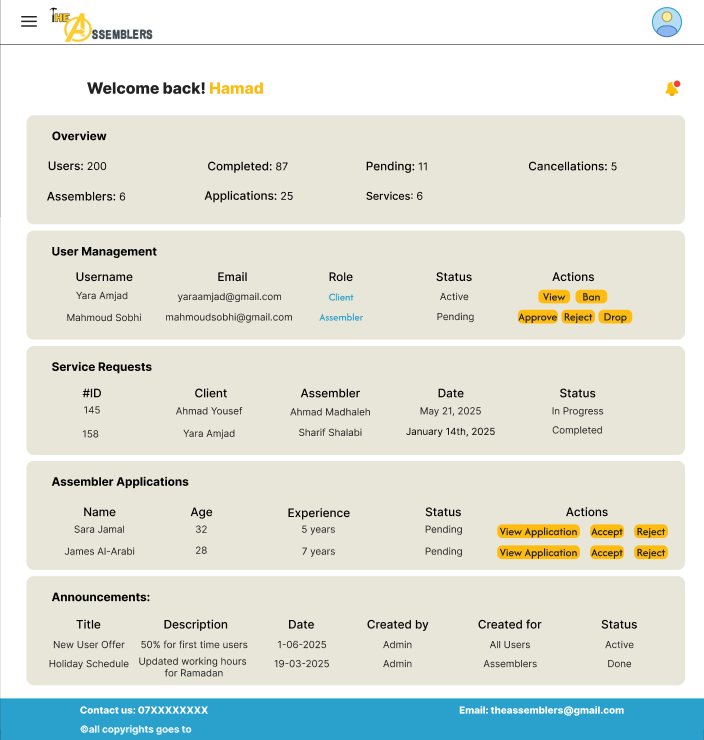
Services (Logged In)

A screenshot of a website

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.Assembler Dashboard

Admin Dashboard

Checkout

Payment Gateway

A screenshot of a computer

AI-generated content may be incorrect.

# 5.0 References: books and tools

Resources

Software Engineering(Tenth Edition) - Ian Sommerville

Chat GPT

How to Design a Website – A UX Wireframe Tutorial(Youtube)

How to Write Software Requirements - 12 Do's and Don'ts -Soft Kraft

UML Tutorial - Use Case, Activity, Class and Sequence Diagrams - Essential Software Modeling(Youtube)

https://www.garoapp.com/#:~:text=تطبيق%20GARO%20هو%20منصَّة%20إلكترونية,فرصة%20البحث%20عن%20فنّيين%20مختص%D9%90ّين%20

Pintrest(for UI)

System Design for Beginners Course(Youtube)

Figma

W3Schools

StarUML

React 19 & Tailwind: Vibe Coding with Cursor (Youtube)