Product Requirements

Project: Service Provider

Team: Group 4

Revision History

Date	Version	Description	Author
02-01-2025	1.0	Initial Software Requirement Specification	Dhyanesh Siva Sai Purna Mamilla
02-01-2025	1.1	Updated Stakeholders Information	Monica Sai Meghana
02-01-2025	1.2	Added Non-Functional Requirements	Devendra Harsha Mamilla
02-02-2025	1.3	Added Functional Requirements (1-3)	Dhyanesh Siva Sai Purna Mamilla
02-02-2025	1.4	Added Functional Requirements (4-6)	Monica Sai Meghana
02-03-2025	1.5	Added Functional Requirements (7-9)	Devendra Harsha Mamilla
02-03-2025	1.6	Added Functional Requirements (10-12)	Raghu Nandan Lal Garikipati
02-04-2025	1.7	Use Case Diagram	Raghu Nandan Lal Garikipati
02-05-2025	1.8	Added Use Case Descriptions (1-3)	Dhyanesh Siva Sai Purna Mamilla
02-05-2025	1.9	Added Use Case Descriptions (4-6)	Monica Sai Meghana
02-06-2025	1.10	Added Use Case Descriptions (7-9)	Devendra Harsha Mamilla
02-06-2025	1.11	Added Use Case Descriptions (10-12)	Raghu Nandan Lal Garikipati
02-07-2025	1.12	Finalized Document and Formatting	Raghu Nandan Lal Garikipati

Brief problem statement

The **Service Provider Project** is mainly designed to connect clients with the service providers while maintaining efficient management by administrators. The system consists of multiple entities, including **users, clients, workers, admins, service requests, and bookings**, all connected through a structured relational database. A role(client, worker, or admin) is being assigned to each **user** which helps us to ensue appropriate functions and permissions. Clients can submit **service requests**, specifying the type of service they need and providing descriptions. These requests are linked to the **clients** table and managed through an approval or completion **status**.

The **workers** table stores details of service providers, including the services they offer and reviews they receive. Each worker is associated with a **user account**, ensuring authentication and role-based access control. The **bookings** table connects **clients and workers**, enabling the scheduling of services. It includes details such as **booking date, scheduled service date, and status**, ensuring proper tracking of appointments. The **admins** table contains information about administrators who oversee the system, granting them permissions to manage users, service requests, and bookings.

This system enhances **service efficiency** by providing a structured process for booking and fulfilling service requests. It ensures **data integrity** by maintaining clear relationships between users, clients, workers, and admins. The design also supports **user authentication and role management**, enabling secure access control. By integrating structured workflows for **booking services, managing service providers, and handling administrative tasks**, the Service Provider Project improves customer experience and streamlines operations.

Key Objectives

1. **Efficient Service Request Management** – Allow clients to create and track service requests while ensuring timely updates on request statuses.
2. **Seamless Booking System** – Facilitate scheduling between clients and workers, including date selection and appointment status tracking.
3. **User Role-Based Access** – Implement distinct roles for **clients, workers, and admins**, ensuring appropriate permissions and secure access.
4. **Worker Service Tracking & Reviews** – Enable workers to list their **services offered** and receive reviews from clients for quality assurance.
5. **Admin Oversight & Management** – Grant administrators permissions to **manage users, services, and bookings** to maintain system integrity.
6. **Data Integrity & Security** – Maintain structured relationships between **users, clients, workers, and bookings** for accurate data management.
7. **User Authentication & Authorization** – Implement secure authentication to prevent unauthorized access and ensure user data confidentiality.
8. **Scalability & Performance** – Design the system to **handle a growing number of clients, workers, and service requests** efficiently.

Stakeholders

"Service Providers" Team - Investor and Board of Directors

The **Service Providers** team is the most important team on this project since they are funding it and providing strategic direction. They oversee making sure that the solution fits market demand, providing resources and tracking progress to get your desired results.

"University of North Texas" Staff - Product owner

At the University of North Texas, Dr. Wajdi Aljedaani and his team are the product owners for Service Providers. They work very closely with the development team to manage and guide the project, make important decisions, and ensure that it fits within the project management and data security standards.

Responsibilities of the Product Owner

- **Defining Product Vision and Strategy** – Define clear objectives that explain how efficient vehicle management, a seamless user experience, and business growth contribute to the Service Provider System vision.
- **Stakeholder Communication** – I will serve as the link between the development team and other important parties like investors, service providers, and end-users to guarantee that everyone is in sync and there is minimal conflict.
- **Development Team Guidance** – Leadership, requirement clarification, and product objective understanding are provided to support the development team for efficient execution.
- **Features and Backlog Prioritization** – We determine important features and establish their priority which enables the team to work on high-impact functionalities of the development backlog to increase both user experience and business value.

Clients

Clients are users who require services and can browse available workers, submit service requests, and book appointments. They can track the status of their service requests, leave reviews for workers, and communicate with service providers to ensure their needs are met.

Workers

Workers are service providers who offer specific services to clients. They receive service requests, manage bookings, update their availability, and complete assigned tasks. They can also receive ratings and reviews, which help build their credibility and improve their visibility in the system.

Admins

Admins oversee the entire platform, ensuring smooth operations and managing users, bookings, and service requests. They have permissions to approve or reject service requests, monitor worker performance, resolve disputes, and maintain the integrity of the platform by verifying user accounts and handling security measures.

Software Development Team

The software development team is responsible for the designing, development, executing, testing and implementing of Service Provider system.

- ensuring reliability and system security and ease to use.
- need to ensure and use Agile methodology in development.
- As per the stakeholders' requirements we need to implement the features.
- improve the product depending on how the users feel about it.

System Users

Users: Clients, Workers, Admins

1. **System Navigation:** The various features that are necessary for their purpose and how to use them.
2. **Data Management:** How to search for, modify, and remove data within or from the system.
3. **Communication:** How users can communicate with one another to exchange information.
4. **Support:** How to use channels for help and support if they need standard assistance or encounter any issues.

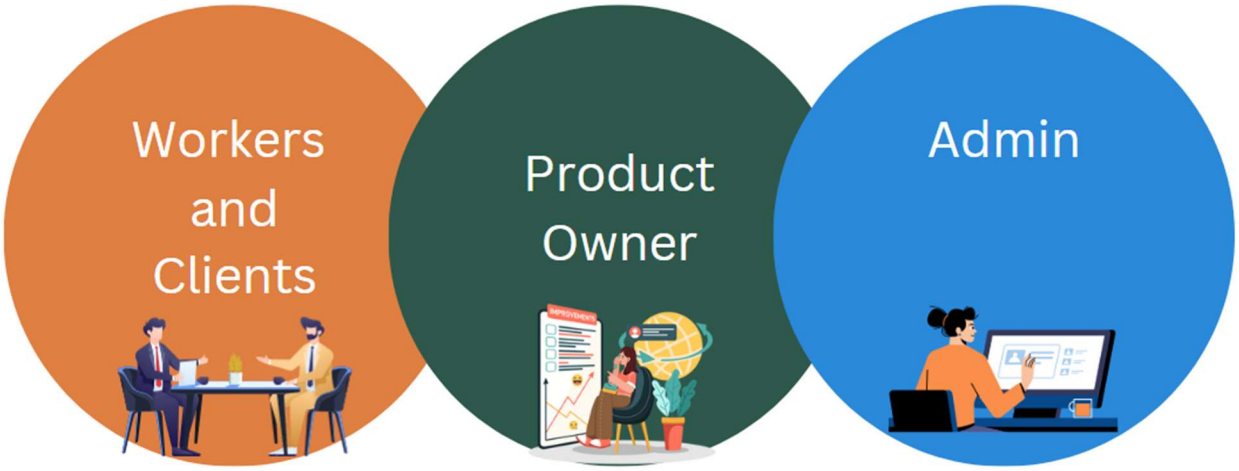

Fig.1 Stakeholders relation

System requirement

Frontend Development:

- **Framework:** React
  - **Version:** Latest stable release (current - React 19.0.0)
  - **Libraries/Tools:** Adding navigation and redirect using React Router, UI components based on React Bootstrap for making API requests using Fetch API/Axios, State managed using useState.
- **Requirements:**
  - **Browser Compatibility:** Modern browsers especially Chrome, Firefox, Safari and Edge.
  - **Responsive Design:** Ensure that the screen fits into all possible resolutions for desktops and is versatile
  - **Performance:** Addresses the need for better load time to selected users.

Database:

- **Database:** Firestore (Firebase)
  - **Version:** Latest stable release
  - **Features:** Continuous work with a high-speed response including to the offline users and highly detailed and secure and compact ways of data storing.
  - **Data Structure:** The system has collection and document structure for accommodating clients, workers, and admin data.

Feature requirements

1. Functional

No.	User Story Name	Description	Release
1	User Registration and Authentication	Secure signup and login system with role-based access control for clients, workers, and admins.	R1
2	Profile Management	Clients and workers can create and update their profiles, including personal details, contact information, and preferences.	R1
3	Service Request Submission	Clients can submit service requests specifying the type of service, description, and urgency	R1
4	Booking System	Allows clients to book appointments with workers, selecting dates and receiving confirmation notifications.	R1
5	Service Provider Listings	A searchable and filterable directory of workers based on services offered, availability, and ratings.	R1
6	Review & Rating System	Clients can provide feedback and rate workers based on the quality of service received.	R1
7	Admin Dashboard	Allows Admins to view statistics, requests, bookings and edit user data.	R2
8	Status Tracking	Allows Workers to update the progress of the work and client to view the progress.	R2
9	Notifications	Booking confirmations, and reminders sent to notifications tab.	R2

10	Search & Filtering Options	Users can filter service providers based on categories, availability, and reviews to find the best match.	R2
11	Analytics & Reports	Generate reports on service usage, worker performance, and customer satisfaction for better decision-making.	R2
12	Live Chat with Service Providers	Real-time messaging between clients and workers, allowing seamless communication for service coordination and updates.	R2

2. Non-Functional

2.1. Performance

- **Response Time:** The app should respond majority of the user requests within 10 seconds in the case of operational requests.
- **Scalability:** The suggested system is capable of supporting concurrent users.

2.2. Reliability

- **Uptime:** The system should be available to the user at 99.9 % access availability, with no or minimal unavailability of the system.
- **Backup:** Creation of backup on a regular basis has to be done for data to maintain integrity or a safe recovery in case of a disaster.

2.3. Security

- **Access Control:** A role based access control must be implemented to ensure information and/ features by the users are only applicable to their position.
- **Authentication:** User authentication must be done securely using strong password acceptance policies and adds multi factor authentication where necessary.

Use case Diagram

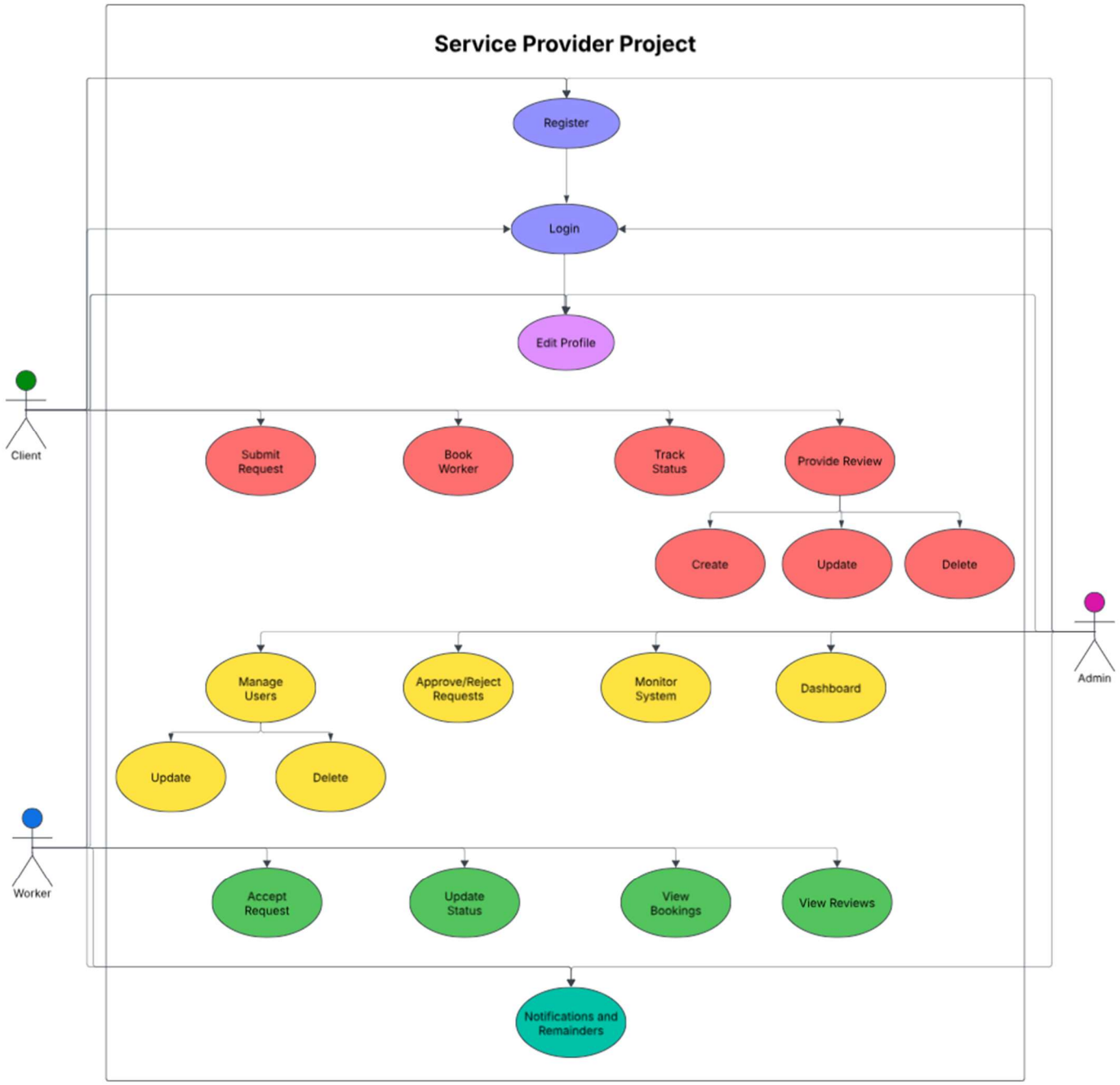

Fig.2 Use Case Diagram.

Use case description

Use Case Number	UC-01
Use Case Name	**User Registration and Authentication**
Overview	Secure signup or sign in system with role-based access control for clients, workers, and admins.
Actor(s)	Clients and Workers
Pre-condition(s)	The portal must be functional and available for access through a URL
Scenario Flow	1. User navigates to the registration or login page. 2. User enters details (username, email, password, role selection). 3. System validates inputs and checks for existing accounts. 4. System registers the user and stores data in the database. 5. User receives a confirmation message.
Alternate Flows	If such email already exists in the system, then the system will ask the user to select another email.
Post Condition	A user account is created and will only be visible to the user upon logging in depending on the user role. The user is successfully logged in and directed to their appropriate home page.

Use Case Number	UC-02
Use Case Name	**Profile Management**
Overview	Clients and workers can create and update their profiles, including personal details, contact information, and preferences.
Actor(s)	Clients and Workers
Pre-condition(s)	The user must be logged in to the system.
Scenario Flow	1. The User logs in and navigates to the profile section. 2. User updates personal details, contact information, or preferences. 3. Worker can add/edit services offered and availability. 4. System validates and saves changes in the database. 5. User receives a success notification.
Alternate Flows	If any of the fields are left blank or empty, the system will display an error message and prompts the user to fill in the missing information.
Post Condition	The user's profile is updated with the new details if the input is valid.

Use Case Number	UC-03
Use Case Name	**Service Request Submission**
Overview	Clients can submit service requests specifying the type of service, description, and urgency
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client logs in and selects "Request Service." 2. Client fills in details (service type, description, urgency). 3. System validates and records the request in the database. 4. Worker gets notified of a new service request. 5. Worker can accept or reject the request. 6. System updates request status and informs the client. .
Alternate Flows	If any of the fields are left blank or empty, the system will display an error message and prompts the user to fill in the missing information.
Post Condition	New Request is saved and available to all other users.

Use Case Number	UC-04
Use Case Name:	**Booking System**
Overview	Allows clients to book appointments with workers, selecting dates and receiving confirmation notifications.
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client browses workers and selects a service provider. 2. Client chooses an available date and time. 3. System checks worker availability and schedules the booking. 4. Worker gets notified of the new booking. 5. Worker confirms or reschedules the booking. 6. System updates booking status and informs the client.
Alternate Flows	Not Applicable
Post Condition	The system saves the booking schedule and notifies the worker. The system saves the Worker confirmation and updates booking status

Use Case Number	UC-05
Use Case Name	**Service Provider Listings**
Overview	A searchable and filterable directory of workers based on services offered, availability, and ratings.
Actor(s)	Clients, Workers
Pre-condition(s)	Not Applicable
Scenario Flow	1. User navigates to the service provider directory. 2. User searches/filter workers based on service type, ratings, or availability. 3. System retrieves and displays relevant worker profiles. 4. User views detailed worker profiles.
Alternate Flows	Not Applicable
Post Condition	The system shows the list of profiles of workers..

Use Case Number	UC-06
Use Case Name	**Review and Rating System**
Overview	Clients can provide feedback and rate workers based on the quality of service received.
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client completes a service and navigates to the rating section. 2. Client provides a star rating and writes a review. 3. System validates and stores the review. 4. Worker gets notified about the feedback. 5. Other clients can view worker ratings
Alternate Flows	Not Applicable
Post Condition	The system stores the rating and review in database.

Use Case Number	UC-07
Use Case Name	**Admin Dashboard**
Overview	Allows Admins to view statistics, requests, bookings and edit user data.
Actor(s)	Admin
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Admin logs in and accesses the dashboard. 2. Admin views system-wide statistics (users, bookings, requests). 3. Admin manages users (approve/reject registrations, deactivate accounts). 4. Admin reviews system performance and logs out.
Alternate Flows	If any of the fields are left blank or empty, the system will display an error message and prompts the user to fill in the missing information.
Post Condition	The system retrieves data related to users, bookings, requests. The system saves the changes made by Admin.

Use Case Number	UC-08
Use Case Name	**Status Tracking**
Overview	Allows Workers to update the progress of the work and client to view the progress.
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client submits a service request or booking. 2. Worker accepts/rejects the request. 3. Worker updates the progress if accepted. 4. System updates the request/booking status and progress.
Alternate Flows	Not Applicable
Post Condition	The system saves the progress and retrieve the data.

Use Case Number	UC-09
Use Case Name	**Notifications**
Overview	Booking confirmations, and reminders sent to notifications tab.
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client and worker receive real-time notifications via notification tab in Service Provider Portal. 2. System logs all status changes for future reference.
Alternate Flows	If any of the fields are left blank or empty, the system will display an error message and prompts the user to fill in the missing information.
Post Condition	The system retrieves the list of notifications for a user..

Use Case Number	UC-10
Use Case Name	**Search & Filtering Options**
Overview	Users can filter service providers based on categories, availability, and reviews to find the best match.
Actor(s)	Clients, Workers
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. User enters keywords in the search bar. 2. System retrieves matching services/providers. 3. User applies filters (ratings, price, location, availability). 4. System displays refined results. 5. User selects a worker for service.
Alternate Flows	If any fields are left blank, the system will display an error message and prompt the user to fill in the missing information.
Post Condition	The system retrieves matching services/providers..

Use Case Number	UC-11
Use Case Name	**Analytics & Reports**
Overview	Generate reports on service usage, worker performance, and customer satisfaction for better decision-making.
Actor(s)	Admin
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Admin accesses the analytics dashboard. 2. System generates reports on service usage, bookings, and worker performance. 3. Admin reviews reports and exports data if needed. 4. Admin makes decisions based on insights.
Alternate Flows	Not Applicable
Post Condition	The system shows generated reports and statistics to Admin.

Use Case Number	UC-12
Use Case Name	**Live Chat with Service Providers**
Overview	Real-time messaging between clients and workers, allowing seamless communication for service coordination and updates.
Actor(s)	Admin
Pre-condition(s)	The user must be logged into the application.
Scenario Flow	1. Client selects an active service request and clicking “Chat with Provider.” 2. The system saves the chat and notifies Worker. 3. Client and Worker exchange communication through messages. 4. The system saves the chat and retrieves chat when opened.
Alternate Flows	Not Applicable
Post Condition	The system saves the chat and retrieves chat when opened.

Individual Contributions:

Student Name	Student-ID	Contributions
Dhyanesh Siva Sai Purna Mamilla	11815670	Software requirements, Worked on Functional requirements(1-3),Use Case description (1-3).
Monica Sai Meghana	11798073	Update stakeholders, Worked on Functional requirements (4-6), Use Case description (4-6).
Devendra Harsha Mamila	11815670	Non-functional requirements, Worked on Functional requirements (7-9), Use Case description (7-9).
Raghu Nandan Lal Garikipati	11754328	Use case diagram, Worked on Functional requirements (10-12), Use case description (10-12).