Institute of Technology University of Moratuwa

SEYONI

A COMPREHENSIVE SMART WORKERS MANAGEMENT SYSTEM

Software Requirements Specification

Version 1.0

Student Details:

M F M Ahzem	21IT0455
A A M Amjad	21IT0457
P A P Dewindi	21IT0462
I M D W Dissanayake	21IT0467
U D S S Kumara	21IT0490

Supervised by:

Ms. M S Madhubhashini

Division of Information Technology

Institute of Technology University of Moratuwa

April 2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the requirements and functionalities of the proposed project. It serves as a blueprint for the development team, guiding them in the design, implementation, and testing phases of the project. Additionally, the SRS document facilitates communication between stakeholders, ensuring a shared understanding of the project scope and objectives.

1.2 Product Scope

1.2.1 Aim and Objectives

Aim

The aim of this project is to develop a robust and user-friendly Workers Management System that connects users with verified service providers, thereby addressing the challenges associated with traditional hiring methods and enhancing transparency and accountability in service delivery.

Objectives

- To provide a platform for users to easily access and request a wide range of home services from verified service providers.
- To ensure reliability and trustworthiness by implementing a thorough verification process for service providers.
- To enhance user experience by incorporating features such as real-time tracking, rating and review system, and secure payment integration.
- To promote efficiency and convenience by streamlining the process of hiring service providers and managing service requests.

1.2.2 Project Boundary

The project boundary is limited to the development and implementation of a mobile application for the Workers Management System. The scope encompasses the following aspects:

- Development of a mobile app compatible with both Android and IOS platforms.
- Inclusion of all functionalities and features outlined in this SRS document, tailored specifically for mobile usage.
- Integration with necessary third-party services such as GPS tracking, payment gateway, and verification services, optimized for mobile application usage.
- Testing and validation of the mobile app on various devices and operating systems to ensure compatibility and usability.
- The targeted geographical area for initial deployment and testing will be specified based on market research and user demographics, with the potential for expansion in the future.

2. Overall Description

2.1 Product Perspective

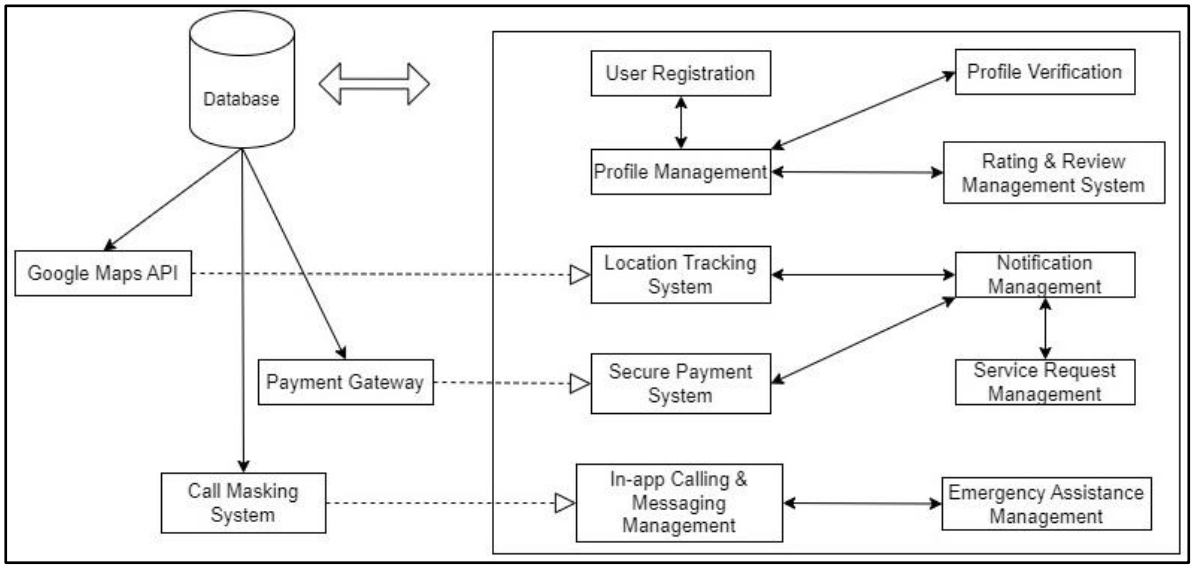

Figure 1: High level architectural diagram

2.2 User Classes and Characteristics

- **Administrators**
  - System administrators responsible for managing user accounts, verifying service providers, and overseeing system operations including customer support.
- **Service Seekers**
  - Users who require services and utilize the application to search for, request, and manage service requests.
- **Service Providers**
  - Individuals or businesses offering services and using the application to receive, accept, and fulfill service requests.

2.3 Operating Environment

The system will operate within a dynamic and diverse environment, accommodating various hardware and software configurations. The following details outline the key components of the operating environment:

Hardware Platform

- The mobile application will be compatible with smartphones and tablets running on iOS (version 11 and above) and Android (version 6.0 Marshmallow and above) operating systems.
- The application should be responsive and optimized for devices with different screen sizes and resolutions to ensure a consistent user experience across devices.
- The system should support standard hardware components commonly found in mobile devices, including GPS sensors, cameras, and touch screens.

Operating System and Versions

- For iOS devices, the application will support iOS 11 and above, ensuring compatibility with a wide range of Apple devices, including iPhones and iPads.
- For Android devices, the application will support Android 6.0 Marshmallow and above, covering a significant portion of the Android user base and enabling access to essential features and functionalities.

Other Software Components or Applications

- The application will interact with external services and APIs, including but not limited to:
    - Google Maps API for location-based services and mapping functionalities.

    - Payment gateway APIs for secure payment processing and transaction management.
    - Firebase or other backend as a service (BaaS) providers for cloud storage, user authentication, and real-time database functionalities.
- The application will integrate with messaging and calling functionalities native to the mobile operating system, allowing users to communicate with service providers seamlessly.
- Compatibility with third-party libraries, frameworks, and development tools used for front-end and backend development, ensuring smooth integration and interoperability with existing software components.

Constraints and Considerations

- The application should be designed to operate efficiently within the constraints of mobile devices, considering factors such as limited processing power, memory, and battery life.
- Compliance with platform-specific guidelines and standards (e.g., Apple App Store Review Guidelines, Google Play Developer Program Policies) to ensure acceptance and distribution through respective app stores.
- Consideration of network connectivity constraints, including support for offline functionality and graceful handling of intermittent or unstable internet connections.
- Adherence to industry best practices and guidelines for mobile application development, including security protocols, data encryption, and user privacy considerations.
- Continuous monitoring and adaptation to evolving hardware and software technologies, ensuring ongoing compatibility and optimal performance across different device configurations and operating system versions.

2.4 Design and Implementation Constraints

Several constraints may influence the design and implementation of the System;

- **Compliance with regulatory policies:** The system must adhere to legal and regulatory requirements governing data privacy, security, and consumer rights.
- **Hardware limitations:** The application's performance may be affected by the capabilities of users' devices, including processing power, memory, and network connectivity.
- **Integration with external services:** The system must seamlessly integrate with third-party services such as GPS tracking, payment gateways, and verification services, which may impose constraints on technology choices and development timelines.
- **Security considerations:** Robust security measures must be implemented to protect user data, transactions, and communications from unauthorized access or breaches.
- **Design conventions and programming standards:** The development team must adhere to established design conventions, programming standards, and best practices to ensure code quality, maintainability, and scalability.

2.5 Assumptions and Dependencies

Assumptions

- **Availability of third-party services**: It is assumed that external services such as GPS tracking, payment gateways, and verification services will be accessible and reliable for integration into the system.
- **Availability of third-party services**: It is assumed that external services such as GPS tracking, payment gateways, and verification services will be accessible and reliable for integration into the system.
- **Stable operating environment**: The application assumes a stable operating environment with consistent internet connectivity and minimal device performance issues.

Dependencies

- **Integration with external APIs**: The project depends on successful integration with external APIs provided by third-party services for functionalities such as GPS tracking, payment processing, and user verification.
- **Availability of development resources**: The project relies on the availability of skilled developers, designers, and testers to execute the design and implementation phases effectively.
- **Timely updates and maintenance**: The project's success depends on ongoing updates, maintenance, and support to address bugs, issues, and user feedback in a timely manner.

3. External Interface Requirements

3.1 User Interfaces

The user interfaces of the mobile application have been carefully designed to provide an intuitive and seamless experience for both service seekers and providers. Below are descriptions of the logical characteristics of each interface, along with sample screen mockups:

Login Page Mockup:

The login page serves as the entry point for users to access the application. It provides fields for users to input their credentials, including username/email and password.

Priority: High

Service Provider's Profile View for Service Seeker Mockup:

This interface allows service seekers to view detailed profiles of service providers. It presents information such as the provider's name, profile picture, service offerings, ratings, reviews, and contact details.

Priority: High

Service Seeker's Home Page Mockup:

The home page serves as the central hub of the application for service seekers. It provides easy access to essential features and functionalities, including searching for service providers, viewing recent activities, accessing user profiles, and initiating service requests.

Priority: High

These mockups showcase the envisioned user interfaces, which will be further refined and implemented in the development phase of the project. The design follows established user experience principles to ensure usability, accessibility, and aesthetic appeal.

3.2 Hardware Interfaces

The system will interact with the hardware components of users' mobile devices, including smartphones and tablets. It will support various device types and screen sizes to ensure compatibility with a wide range of devices. The system will utilize standard data and control interactions between the software and hardware components, with communication protocols such as HTTP and HTTPS for data exchange.

3.3 Software Interfaces

This system will integrate with various software components and external services to enhance its functionality. Key software interfaces include:

- **Google Maps API:** Used for real-time location tracking and mapping functionalities within the application.
- **Payment Gateway API:** Integration with a secure payment gateway for processing financial transactions between users and service providers.
- **Database Management System:** Utilized for storing and retrieving user data, service requests, and transaction records. The system will interact with the database through standard SQL queries and commands.
- **Operating System:** The mobile application will be compatible with Android and iOS operating systems, utilizing native development frameworks and APIs for platform-specific functionalities.

3.4 Communications Interfaces

Our System will utilize various communication functions to facilitate interactions between users and the software. Communication interfaces include:

- **HTTP/HTTPS:** Used for communication between the mobile application and backend servers, ensuring secure data transmission over the internet.
- **SMTP:** Employed for sending email notifications and alerts to users, such as account verification emails and service request updates.
- **FTP:** Utilized for file transfer functionalities, enabling the exchange of documents and media files between users and service providers.
- **WebSocket:** Implemented for real-time communication and notifications within the application, allowing for instant updates on service requests and provider availability.

Reasons for Using Protocols:

- **HTTP/HTTPS:** Ensures secure and reliable communication between the mobile application and backend servers, protecting sensitive user data.
- **SMTP:** Facilitates the delivery of email notifications to users, enhancing communication and providing timely updates on account activities.
- **FTP:** Enables seamless file transfer functionalities, allowing users to share documents and media files with service providers.
- **WebSocket:** Supports real-time communication and notifications within the application, enhancing user engagement and providing instant updates on relevant events.

4. System Designs

4.1 Use case Diagram.

Figure 2: Use case diagram

4.1.1 Use case Description

Use Case Name	Requesting a Service	
Priority	High	
Primary System Actor	Service Seeker	
Other Participating Actors	Service Provider	
Description	This use case describes the process of a service seeker requesting a service through the mobile application.	
Pre-Condition	The service seeker must be logged into the application	
Trigger	The service seeker selects the "Request Service" option from the app menu.	
Typical Course Events	**Actor Action**	**System Response**
	The service seeker fills out the service request form with details such as service type, location, and preferred time.	The request is submitted to the system, and a notification is sent to nearby service providers
Alternate Courses	If the service seeker encounters an issue with the form submission, an error message is displayed, prompting them to correct the errors and resubmit the request	
Conclusion	The service request is successfully submitted to the system, and the service seeker awaits responses from service providers	
Post Condition	The service request is stored in the system, and service providers can view and respond to it	
Implementation Constraints and Specification	The form submission must include mandatory fields such as service type and location. The system must validate the input data to ensure accuracy and completeness.	
Assumption	The service seeker has a stable internet connection and has provided accurate information in the request form.	

Table 1: Use case narrative 1

Use Case Name	Accepting a Service Request	
Priority	High	
Primary System Actor	Service Provider	
Other Participating Actors	Service Seeker	
Description	This use case describes the process of a service provider accepting a service request from a service seeker.	
Pre-Condition	The service provider must be logged into the application and have received a notification for a service request.	
Trigger	The service provider receives a notification for a service request	
Typical Course Events	**Actor Action**	**System Response**
	The service provider reviews the details of the service request and decides to accept it.	The service provider confirms acceptance of the request, and a notification is sent to the service seeker.
Alternate Courses	If the service provider declines the request, the system sends a notification to the service seeker, indicating that the request is no longer available.	
Conclusion	The service request is accepted by the service provider, and further communication regarding service details ensues.	
Post Condition	The service request status is updated in the system, indicating that it has been accepted by a service provider.	
Implementation Constraints and Specification	The system must provide real-time notifications to both service seekers and providers regarding service requests and responses.	
Assumption	The service provider has availability to fulfill the service request and is willing to proceed with the task.	

Table 2: Use case narrative 2

Use Case Name	Completing a Service	
Priority	High	
Primary System Actor	Service Provider	
Other Participating Actors	Service Seeker	
Description	This use case describes the process of a service provider completing a service for a service seeker.	
Pre-Condition	The service provider has accepted a service request and arrived at the service location.	
Trigger	The service provider starts and completes the service task.	
Typical Course Events	**Actor Action**	**System Response**
	The service provider commences the service task and performs the necessary actions to complete it.	The system records the start time of the service and tracks the duration until completion. Upon finishing the task, the service provider marks it as complete in the application.
Alternate Courses	If the service provider encounters any issues during the service task, they can communicate with the service seeker through the application to address them.	
Conclusion	The service task is successfully completed by the service provider, and relevant details are recorded in the system.	
Post Condition	The service status is updated to "Completed" in the system, and the service provider can proceed with payment processing.	
Implementation Constraints and Specification	The system must accurately track the duration of service tasks and record any interruptions or delays encountered during the process.	
Assumption	The service seeker is satisfied with the completed service, and there are no outstanding issues or disputes regarding the task.	

Table 3: Use case narrative 3

Use Case Name	Rating and Reviewing a Service	
Priority	Medium	
Primary System Actor	Service Seeker	
Other Participating Actors	Service Provider	
Description	This use case describes the process of a service seeker providing feedback on a completed service.	
Pre-Condition	The service task has been marked as completed by the service provider.	
Trigger	The service seeker receives a notification prompting them to rate and review the completed service.	
Typical Course Events	**Actor Action**	**System Response**
	The service seeker navigates to the completed service details and provides a rating and written review based on their experience.	The system records the feedback provided by the service seeker and updates the service provider's profile with the received rating and review.
Alternate Courses	If the service seeker chooses not to provide feedback, the system retains the default rating for the service provider.	
Conclusion	The service seeker's feedback is captured and reflected in the service provider's profile, aiding other users in their decision-making process.	
Post Condition	The service provider's profile is updated with the received rating and review, influencing their visibility and reputation within the platform.	
Implementation Constraints and Specification	The system must ensure the authenticity and relevance of the feedback provided by users, preventing misuse or manipulation of ratings and reviews.	
Assumption	The service seeker's feedback accurately reflects their experience with the service provided, and there are no disputes regarding the validity of the feedback.	

Table 4: Use case narrative 4

Use Case Name	Managing Profile Information	
Priority	Medium	
Primary System Actor	User (Service Provider/Seeker)	
Other Participating Actors	N/A	
Description	This use case describes the process of managing profile information by the users within the system.	
Pre-Condition	The user must be logged into the application.	
Trigger	The user accesses the profile management section within the application.	
Typical Course Events	**Actor Action**	**System Response**
	The user edits their personal information, such as contact details, profile picture, bio, and service preferences.	The system updates the user's profile with the edited information and ensures data integrity and security.
Alternate Courses	If the user encounters any issues during profile editing, they can revert to the previous information or contact support for assistance.	
Conclusion	The user's profile information is successfully updated and reflects their current preferences and details.	
Post Condition	The updated profile information is stored securely in the system and is accessible for other users as needed.	
Implementation Constraints and Specification	The system must implement robust data validation mechanisms to ensure that only valid and authorized changes are made to user profiles.	
Assumption	Users have the necessary permissions to edit their profile information, and the system provides intuitive user interfaces for easy profile management.	

Table 5: Use case narrative 5

Use Case Name	Accessing Service History	
Priority	Medium	
Primary System Actor	User (Service Provider/Seeker)	
Other Participating Actors	N/A	
Description	This use case describes the process of accessing the service history for users within the system.	
Pre-Condition	The user must be logged into the application.	
Trigger	The user navigates to the service history section within the application.	
Typical Course Events	**Actor Action**	**System Response**
	The user views past service requests and completed tasks, including details such as service type, date, provider information, and ratings.	The system retrieves and displays the user's service history, allowing for easy reference and review.
Alternate Courses	If the user has not engaged in any previous services, the system displays a message indicating an empty service history.	
Conclusion	The user's profile information is successfully updated and reflects their current preferences and details.	
Post Condition	The user gains insights into their service history, which may inform their preferences and choices within the system.	
Implementation Constraints and Specification	The system must ensure the confidentiality and integrity of user service history data, adhering to privacy regulations and best practices.	
Assumption	Users have the necessary permissions to access their service history, and the system provides efficient data retrieval and presentation functionalities.	

Table 6: Use case narrative 6

Use Case Name	Handling Emergency Situations	
Priority	High	
Primary System Actor	Service Seeker/Provider	
Other Participating Actors	Emergency Services (optional)	
Description	This use case describes the process of handling emergency situations within the system.	
Pre-Condition	The user must be logged into the application, and the emergency feature must be enabled.	
Trigger	The user encounters an emergency situation during a service engagement.	
Typical Course Events	**Actor Action**	**System Response**
	The user activates the emergency feature within the application, indicating the nature and severity of the emergency.	The system immediately notifies the designated emergency services and provides the user's location and relevant details.
Alternate Courses	If the user is unable to activate the emergency feature, they can contact emergency services directly using external means (e.g., phone call).	
Conclusion	The system facilitates swift response and assistance during emergency situations, prioritizing user safety and well-being.	
Post Condition	Emergency services are alerted and dispatched to the user's location, providing timely assistance and support.	
Implementation Constraints and Specification	The system must ensure accurate and reliable communication with emergency services, minimizing response time and potential errors.	
Assumption	Users are aware of the emergency feature and its functionalities, and emergency services are equipped to handle requests from the system.	

Table 7: Use case narrative 7

Use Case Name	Managing Notifications	
Priority	Medium	
Primary System Actor	Service Seeker/Provider	
Other Participating Actors	N/A	
Description	This use case describes the process of managing notifications within the system.	
Pre-Condition	The user must be logged into the application, and notification settings must be accessible.	
Trigger	The user accesses the notification settings section within the application.	
Typical Course Events	**Actor Action**	**System Response**
	The user configures notification preferences, including the types of notifications to receive (e.g., service updates, messages, alerts) and the preferred delivery channels (e.g., in-app notifications, email, SMS).	The system updates the user's notification settings accordingly and ensures proper delivery of notifications based on the configured preferences.
Alternate Courses	If the user encounters issues with notification delivery or preferences, they can adjust settings or contact support for assistance.	
Conclusion	The user has control over the notifications they receive, enhancing their overall experience and engagement within the platform.	
Post Condition	The user receives notifications as per their configured preferences, staying informed and up-to-date on relevant activities and updates.	
Implementation Constraints and Specification	The system must support various notification channels and ensure timely delivery of notifications while respecting user preferences and privacy settings.	
Assumption	Users have the autonomy to customize their notification settings according to their preferences and requirements.	

Table 8: Use case narrative 8

Use Case Name	Accessing Help and Support	
Priority	Medium	
Primary System Actor	User (Service Seeker/Provider)	
Other Participating Actors	Support Team (Admin)	
Description	This use case describes the process of accessing help and support resources within the system.	
Pre-Condition	The user must be logged into the application, and the help/support feature must be accessible.	
Trigger	The user encounters an issue or requires assistance while using the application.	
Typical Course Events	**Actor Action**	**System Response**
	The user accesses the help/support section within the application or initiates a support request through designated channels.	The system provides access to relevant help resources, such as FAQs, tutorials, or contact options for reaching the support team.
Alternate Courses	If the user's issue is not resolved through self-help resources, they can escalate the request to the support team for personalized assistance.	
Conclusion	The help and support feature assists users in resolving issues and obtaining guidance, enhancing their overall experience and satisfaction with the platform.	
Post Condition	The user receives appropriate help or support from the system, resolving their issue or query effectively.	
Implementation Constraints and Specification	The system must ensure timely and efficient handling of support requests, prioritizing user satisfaction and resolution.	
Assumption	Users have access to comprehensive help and support resources within the application, and the support team is equipped to address a wide range of user inquiries and issues.	

Table 9: Use case narrative 9

Use Case Name	Managing Payments	
Priority	High	
Primary System Actor	Service Seeker and Service Provider	
Other Participating Actors	Payment Gateway	
Description	This use case outlines the process of managing payments for completed service engagements within the application.	
Pre-Condition	The service engagement must be completed, and both the seeker and provider must be logged into the application.	
Trigger	The service provider indicates the completion of the service, prompting the initiation of the payment process.	
Typical Course Events	**Actor Action**	**System Response**
	The service seeker confirms the completion of the service and proceeds to make the payment through the designated payment gateway within the application.	The system processes the payment transaction securely through the integrated payment gateway, deducting the agreed-upon service charges from the seeker's account and crediting the provider's account accordingly.
Alternate Courses	In case of payment failures or disputes, the system provides options for resolving issues and retrying the payment process or seeking assistance from support.	
Conclusion	The payment management feature streamlines the financial transactions between service seekers and providers, ensuring prompt and secure payments for rendered services.	
Post Condition	Upon successful payment processing, both the seeker and provider receive confirmation of the transaction, and the service engagement is officially concluded.	
Implementation Constraints and Specification	The system must integrate with a reliable and secure payment gateway to facilitate seamless transactions, adhering to industry standards for data protection and encryption.	
Assumption	Users have access to valid payment methods and sufficient funds to complete the transaction, and the payment gateway operates efficiently without significant downtime or technical issues.	

Table 10: Use case narrative 10

Use Case Name	Registration and Login	
Priority	High	
Primary System Actor	New User (Service Seeker or Provider)	
Other Participating Actors	N/A	
Description	This use case outlines the process of registering and logging into the application for new users.	
Pre-Condition	The user must have access to a compatible device with internet connectivity.	
Trigger	The user initiates the registration process by accessing the application for the first time.	
Typical Course Events	**Actor Action**	**System Response**
	The user provides necessary information such as name, email address, contact number, and creates a password to register for an account.	The system verifies the provided information, creates a unique user account, and prompts the user to log in with the newly created credentials.
Alternate Courses	If there are any errors or invalid inputs during registration, the system notifies the user and prompts them to correct the information before proceeding.	
Conclusion	The registration and login feature enable new users to create accounts and access the application's functionalities securely.	
Post Condition	Upon successful registration and login, users gain access to their personalized profiles and the full range of application features.	
Implementation Constraints and Specification	The system must enforce strong password requirements and implement secure authentication mechanisms to protect user accounts from unauthorized access.	
Assumption	Users have access to valid email addresses and contact numbers for registration, and the registration process is straightforward and user-friendly.	

Table 11: Use case narrative 11

Use Case Name	Profile Verification	
Priority	High	
Primary System Actor	System Administrator	
Other Participating Actors	Service Seeker or Provider	
Description	This use case describes the process of verifying user profiles within the application to ensure authenticity and trustworthiness.	
Pre-Condition	The user must have completed the registration process and submitted their profile details.	
Trigger	The system prompts the user to verify their profile after registration.	
Typical Course Events	**Actor Action**	**System Response**
	The user submits necessary identification documents or credentials for profile verification.	The system reviews the submitted documents and verifies the user's profile based on predetermined criteria.
Alternate Courses	If there are any errors or invalid inputs during registration, the system notifies the user and prompts them to correct the information before proceeding If the submitted documents are incomplete or invalid, the system notifies the user and requests additional information for verification.	
Conclusion	The profile verification feature enhances trust and credibility within the application by ensuring that user profiles are authentic and reliable.	
Post Condition	Upon successful verification, the user's profile is marked as verified, indicating to other users that they have undergone the verification process.	
Implementation Constraints and Specification	The system must handle sensitive user information securely during the verification process, complying with relevant data protection regulations.	
Assumption	Users understand the importance of profile verification for building trust within the application community and are willing to provide necessary documentation for verification.	

Table 12: Use case narrative 12

Use Case Name	In-App Calling and Messaging	
Priority	Medium	
Primary System Actor	Service Seeker or Provider	
Other Participating Actors	N/A	
Description	This use case outlines the functionality for users to communicate with each other through in-app calling and messaging features.	
Pre-Condition	The users must be logged into the application and have access to the messaging interface.	
Trigger	The user initiates a communication session with another user through the application.	
Typical Course Events	**Actor Action**	**System Response**
	The user selects the desired contact from their list of connections and initiates a call or message within the application.	The system establishes a secure communication channel between the users, allowing them to exchange messages or engage in voice calls within the application interface.
Alternate Courses	If there are any technical issues or network interruptions during the communication session, the system notifies the users and provides options for resolving the issues.	
Conclusion	The in-app calling and messaging feature facilitates seamless communication between users, enhancing collaboration and coordination within the application.	
Post Condition	Upon completion of the communication session, users can review their message history or call logs within the application.	
Implementation Constraints and Specification	The system must prioritize user privacy and data security during in-app communications, implementing encryption protocols to protect user messages and calls.	
Assumption	Users have access to stable internet connections and compatible devices for in-app communication, and the application interface is intuitive and user-friendly for initiating calls and messages.	

Table 13: Use case narrative 13

Use Case Name	Search & View Profile (Service Provider's Profile)	
Priority	High	
Primary System Actor	Service Seeker	
Other Participating Actors	N/A	
Description	This use case describes the process of searching for and viewing the profile of service providers within the application.	
Pre-Condition	The user must be logged into the application and have access to the search functionality.	
Trigger	The user initiates a search for service providers based on specific criteria or services.	
Typical Course Events	**Actor Action**	**System Response**
	The user enters search criteria such as service type, location, or provider name in the search bar.	The system retrieves relevant service provider profiles matching the search criteria and displays them to the user.
Alternate Courses	If there are no matching profiles found for the given search criteria, the system notifies the user and suggests refining the search parameters.	
Conclusion	The search and view profile feature enables users to discover and evaluate service providers based on their preferences and requirements.	
Post Condition	Upon viewing a service provider's profile, the user can assess their qualifications, ratings, reviews, and other relevant information to make an informed decision.	
Implementation Constraints and Specification	The system must optimize search functionality for fast and accurate results, considering factors such as location proximity, service availability, and user preferences.	
Assumption	Users understand how to effectively use search filters and criteria to find relevant service providers, and the application provides intuitive navigation for browsing and viewing profiles.	

Table 14: Use case narrative 14

Use Case Name	Accept or Reject Seeker's Request	
Priority	High	
Primary System Actor	Service Provider	
Other Participating Actors	Service Seeker	
Description	This use case outlines the process for service providers to accept or reject service requests from seekers within the application.	
Pre-Condition	The service provider must be logged into the application and have received a service request notification.	
Trigger	The service provider receives a notification for a service request from a seeker.	
Typical Course Events	**Actor Action**	**System Response**
	The service provider reviews the details of the service request, including the required service, location, and requested time.	The system prompts the service provider to either accept or reject the service request.
Alternate Courses	If the service provider is unavailable or unable to fulfill the request, they can reject the service request with an optional explanation.	
Conclusion	The accept or reject seeker's request feature enables service providers to manage incoming service requests effectively, ensuring timely responses and efficient service delivery.	
Post Condition	Upon accepting a service request, the service provider's availability is updated, and the system notifies the seeker of the acceptance. If the request is rejected, the system informs the seeker and may suggest alternative providers.	
Implementation Constraints and Specification	The system must provide clear and intuitive interfaces for service providers to review and respond to service requests promptly. Additionally, it should consider factors such as availability, location, and service preferences when matching providers with seekers.	
Assumption	Service providers are responsive to incoming service requests and understand the importance of timely communication with seekers for efficient service delivery. The application interface provides convenient options for accepting or rejecting requests with minimal effort.	

Table 15: Use case narrative 15

5. System Features

5.1 User Registration and Profile Management

5.1.1 Description and Priority

This feature enables users to register on the platform and manage their profiles, providing necessary information and preferences.

It is of high priority as it forms the foundation for user interaction and personalization within the system.

5.1.2 Stimulus/Response Sequences

Upon accessing the registration section of the app, the system prompts the user to input basic details including name, email, and password. When user completes the registration process and receives a confirmation email, the system creates a user profile with the provided information, enabling the user to access the profile management section for updating personal information, changing preferences, and viewing past activities. Supported user categories are 'Service Seeker' who place the service request, ‘Service Provider’ who is interested in providing services.

5.1.3 Functional Requirements

REQ-1: shall be able to facilitate users to select the user category.

REQ-2: shall be able to facilitate users entering personal details.

REQ-3: shall be able to store personal details.

REQ-4: shall be able to send a confirmation link to the email provided.

REQ-5: shall be able to register the user by making a user profile.

REQ-6: shall be able to provide profile management functionality for users to update personal information.

REQ-7: shall be able to provide password recovery mechanism for users to reset passwords in case of forgotten credentials.

5.2 Profile Verification

5.2.1 Description and Priority

This feature allows administrators to verify profiles of service providers. Provided details such as qualifications, experience, and service offerings are being verified.

Verification of these profiles is of high priority to ensure the reliability and trustworthiness of providers.

5.2.2 Stimulus/Response Sequences

After navigating to the profile creation section, the system prompts the user to input personal details. Upon submission, the information is examined through document verification and background checks. If verification is successful, the system notifies the user and activates the profile, otherwise prompt them to provide additional information or correct discrepancies.

5.2.3 Functional Requirements

REQ-1: shall be able to facilitate the document upload functionality for users.

REQ-2: shall be able to facilitate users entering personal details.

REQ-3: shall be able to facilitate admins the background verification process to validate information provided by the user.

REQ-4: shall be able to send a notification to inform users of profile verification status.

REQ-5: shall be able to facilitate error handling mechanism to prompt users for correction in case of discrepancies during verification.

REQ-6: shall be able to store user data and documents.

5.3 Rating and Review System

5.3.1 Description and Priority

This feature allows users to rate and review service providers based on their experiences.

It is of high priority as it contributes to the transparency and reliability of the platform.

5.3.2 Stimulus/Response Sequences

After the service provider completes a service, the system prompts users to provide a rating and review for the service. Upon inputting their rating and, optionally, writing a review, the system records the feedback in the database and triggers a notification to the service provider who can then view the rating and review on their profile.

5.3.3 Functional Requirements

REQ-1: shall be able to facilitate user interface for rating and reviewing services.

REQ-2: shall be able to facilitate users with a rating scale (e.g. star rating) to provide feedback.

REQ-3: shall be able to provide a text input field for users to write detailed reviews.

REQ-4: shall be able to validate the ratings to ensure they fall within a predefined range (e.g. 1-5 stars)

REQ-5: shall be able to notify service providers to inform of new ratings and reviews.

REQ-6: shall be able to display ratings and reviews on service provider profiles.

REQ-7: shall be able to facilitate users to edit or delete their own reviews if needed.

5.4 Real-time Tracking of Service Providers

5.4.1 Description and Priority

This feature enables users to track the real-time location of service providers while they are ending route to the service location.

It is of high priority to enhance transparency and provide users with visibility into the arrival time of the service provider.

5.4.2 Stimulus/Response Sequences

When the user selects a service provider for a specific service, the system retrieves the real-time location of the service provider and displays it on a map interface for the user; continuously updating as the service provider moves towards the service location. The system allows the user to view the estimated arrival time based on the service provider's current location and speed.

5.4.3 Functional Requirements

REQ-1: shall be able to integrate with GPS tracking technology to obtain real-time location data of service providers.

REQ-2: shall be able to facilitate a user interface for users to view the map with the service provider's location.

REQ-3: shall be able to update the service provider's location continuously.

REQ-4: shall be able to validate the ratings to ensure they fall within a predefined range (e.g. 1-5 stars)

REQ-5: shall be able to calculate and display estimated arrival time.

REQ-6: shall be able to refresh the tracking interface for the latest location updates.

REQ-7: shall be able to notify users in case of technical issues or disruptions in tracking services through an error handling mechanism.

REQ-7: shall be able to compatible with various devices and operating systems for seamless tracking.

5.5 Service Request Management

5.5.1 Description and Priority

This feature enables users to submit service requests, manage their requests, and track the status of ongoing requests.

It is of high priority as it forms the core functionality of the platform, facilitating seamless interaction between users and service providers.

5.5.2 Stimulus/Response Sequences

When the user navigates to the service request section of the app, the system prompts them to select the type of service they require, after which the user inputs details such as service location, preferred time, and any specific requirements. Subsequently, the system generates a service request and assigns it to available service providers, who receive notifications and can accept or reject the requests, while the user receives notifications of service provider responses and can track the status of their request.

5.5.3 Functional Requirements

REQ-1: shall be able to facilitate a user interface for submitting service requests with input fields.

REQ-2: shall be able to integrate with location services to automatically detect and input the user's current location.

REQ-3: shall be able to facilitate the dynamic selection of available service providers based on location, availability, and service type.

REQ-4: shall be able to view detailed information about their service request.

REQ-5: shall be able to provide an option for users to cancel or modify their service requests before they are accepted by a service provider.

5.6 Secure Payment Integration

5.6.1 Description and Priority

This feature allows users to make secure payments for the services they receive through the platform.

It is of high priority as it ensures smooth transactions and financial security for both users and service providers.

5.6.2 Stimulus/Response Sequences

Upon completion of a service provided by a service provider, he can put the charged amount for the service, so that it will be displayed on the seeker's interface. After which the user selects a payment method (such as credit/debit card or digital wallet) and inputs payment details; subsequently, the system securely processes the payment and sends a confirmation to both the user and the service provider.

5.6.3 Functional Requirements

REQ-1: shall be able to integrate with secure payment gateway services to facilitate online transactions.

REQ-2: shall be able to facilitate a user interface for selecting payment methods and entering payment details.

REQ-3: shall be able to encrypt the sensitive payment information to ensure data security during transmission.

REQ-4: shall be able to support multiple payment methods to accommodate user preferences.

REQ-5: shall be able to notify users and service providers upon successful payment processing.

REQ-6: shall be able to provide an option for users to save payment methods for future transactions for convenience.

REQ-7: shall be able to facilitate with error handling system to handle payment failures, incomplete transactions, and other payment-related issues.

5.7 Emergency Assistance

5.7.1 Description and Priority

This feature provides users with access to emergency assistance services directly from the app in case of urgent situations.

It is of high priority to ensure the safety and well-being of users and service providers.

5.7.2 Stimulus/Response Sequences

In the event of an emergency situation, such as a medical emergency or safety threat, both the seeker and provider can access the emergency assistance feature within the app. The system presents options for emergency services like medical, police, or fire assistance. After selecting the appropriate service and confirming the request, the system promptly sends the emergency request to the relevant authorities along with the user's location information, and the user receives confirmation of the emergency request along with any further instructions.

5.7.3 Functional Requirements

REQ-1: shall be able to facilitate user interface for accessing emergency assistance services with options for different types of emergencies.

REQ-2: shall be able to integrate with emergency service providers such as ambulance services, police departments, and fire departments.

REQ-3: shall be able to integrate with GPS tracking functionality to automatically detect and provide the user's location to emergency responders.

REQ-4: shall be able to facilitate communication between users and emergency responders.

REQ-5: shall be able to notify users of the confirmation.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

- The system should respond to user interactions within 2 seconds to ensure a seamless user experience.
- The system should be able to handle concurrent user requests without significant degradation in performance, supporting at least 1000 simultaneous users.
- Data retrieval and loading times should be minimized, with large datasets loading within 5 seconds.

6.2 Safety Requirements

- The system should prioritize the safety and security of users' personal information by implementing encryption protocols for data transmission and storage.
- In case of emergencies, such as the use of the emergency assistance feature, the system should provide immediate access to relevant emergency services, ensuring user safety.
- Compliance with relevant data protection regulations (e.g., GDPR) to safeguard user privacy and prevent unauthorized access to sensitive information.

6.3 Security Requirements

- User authentication mechanisms, such as two-factor authentication, should be implemented to verify user identities and prevent unauthorized access to accounts.
- All communications between the application and external servers should be encrypted using industry-standard protocols (e.g., HTTPS) to protect data integrity and confidentiality.
- Regular security audits and vulnerability assessments should be conducted to identify and address potential security risks proactively.

6.4 Software Quality Attributes

- **Usability**: The system should prioritize ease of use and intuitive navigation to ensure a positive user experience for individuals of varying technical expertise.
- **Reliability**: The system should operate consistently and reliably under normal usage conditions, minimizing system downtime or disruptions.
- **Maintainability**: The system should be designed with modular and well-documented code to facilitate future updates, maintenance, and troubleshooting.
- **Scalability**: The system should be capable of accommodating future growth and increasing user demands by scaling infrastructure and resources as needed.
- **Interoperability**: The system should seamlessly integrate with external services and platforms, allowing for interoperability and data exchange with third-party systems.
- **Robustness**: The system should be resilient to errors and exceptions, gracefully handling unexpected inputs or conditions to prevent system failures or crashes.
- **Testability**: The system should be designed with built-in testing capabilities, allowing for comprehensive testing of all system components to ensure software quality and reliability.