

Software Requirements Specification (SRS) Document

1. Introduction

1.1 Purpose

The purpose of this document is to outline the requirements for the development of a web application system aimed at connecting individuals with urgent repair needs to qualified professionals for quick assistance. It also serves as a bridge between users who would like to hire professionals or enter into a project contract. Additionally, it is a system that will facilitate employability, if necessary, for professionals. This document provides a detailed specification to guide the development team.

1.2 Scope

The scope of the system encompasses the creation of a user-friendly platform that seamlessly connects individuals seeking urgent assistance for troubleshooting various issues (e.g., plumbing, HVAC, appliance repairs) with qualified professionals available in their local area. This platform not only facilitates direct communication and hiring but also accommodates individual users, professionals, companies, and administrators, providing a comprehensive solution for their respective needs.

1.3 Objectives

- To provide a user-friendly interface for individuals to submit and describe their urgent repair needs.
- To create a database of local professionals, categorizing them based on their expertise.
- To enable real-time communication between users and professionals.
- To facilitate the scheduling and tracking of service requests.
- To ensure the security and privacy of user information.
- To create a seamless platform for users, professionals, and companies to connect.
- To facilitate efficient communication and hiring processes.
- To implement features for both short-term project-based hiring and long-term employment opportunities.

2. Description

2.1 System Architecture

The system will be built as a web application accessible through standard web browsers. It will consist of a frontend for users and professionals, a backend server, and a database for storing user and project information. The system will integrate with external services for secure authentication and payment processing.

2.2 Product Descriptive

FixIt : Urgent repairs, connected instantly. This title is catchy, emphasizes urgency, and clearly states the purpose of the app. The product is a web-based platform developed using modern web technologies.

2.3 User Characteristics

Users range from individuals seeking short-term services to companies/users looking for long-term professionals.

2.4 User Roles

- **User:** Individuals seeking urgent repair assistance, service and maybe hiring professionals.
- **Professional:** Qualified service providers
- **Company:** Organizations looking to hire professionals for projects.
- **Admin:** Platform administrators managing user accounts and overseeing operations

2.5 Features and functionalities:

User Features:

- **Location services:** Identify user location to connect them with nearby professionals.
- **Urgent request:** Submit a request for immediate repair assistance, specifying the issue and desired timeframe.
- **Professional search:** Search for qualified professionals by category (plumbing, HVAC, etc.), location, and ratings.
- **Booking and scheduling:** Books appointments directly with professionals and manages schedules within the app.
- **Communication tools:** Secure in-app messaging or call functionality to communicate with professionals.
- **Payment processing:** Secure payment system for booking fees and service charges.
- **Review and rating system:** Leave reviews and ratings for professionals after service completion.

Professional Features:

- **Profile management:** Create and manage a professional profile showcasing skills, experience, and certifications.
- **Availability management:** Set availability for appointments and manage schedules.
- **Notification system:** Receive notifications for new repair requests in their service area.
- **Bidding system (optional):** Professionals can offer quotes for complex jobs.
- **Job history and management:** View past jobs, track communication with clients, and manage completed projects.

Additional functionalities:

- **Admin dashboard:** Manage user accounts, professional profiles, and system settings.
- **Content management system (CMS):** Update service categories, FAQs, and other informative content.
- **Reporting and analytics:** Track user activity, professional performance, and service trends.

2.6 Constraints

Professionals should agree to terms of service that discourage direct communication outside the platform for a specific period after project completion.

Professionals and users are encouraged to use the platform's communication tools initially to maintain transparency.

2.7 Assumptions and Dependencies

- The system assumes users have access to a stable internet connection.
- Dependencies include external services for secure authentication and payment processing.

Professionals agree to use the platform for communication initially to maintain transparency.

The platform reserves the right to intervene in disputes and enforce terms of service.

Retaining Clients:

- 1- **Value-Added Services:** Offer additional features such as project management tools, collaboration spaces, or dispute resolution services to incentivize users to stay on the platform.
- 2- **Membership Benefits:** Introduce a membership program for users and companies, providing exclusive benefits, discounts, or access to premium features.

- 3- **Assurance:** Implement a rigorous vetting process for professionals to ensure high-quality services, building trust among users and companies.
- 4- **Incentivize Loyalty:** Reward users, professionals, and companies with loyalty points or discounts for repeated use of the platform.
- 5- **Educational Resources:** Provide resources, webinars, or training opportunities related to the industry for users and professionals, creating a sense of community.
- 6- **Regular Updates:** Keep the platform updated with new features, improvements, and optimizations based on user feedback.

3. Specific requirements

3.1 External Interfaces

3.1.1 User Interface

The user interface should be intuitive and responsive, supporting a variety of devices and screen sizes.

3.2.1 External Services

Integration with secure authentication services for user and professional registration and login.

Authentication: Implement OAuth for third-party login options (Google, Facebook).

Integration with payment processing services for transactions.

3.2 Functional Requirements

3.2.1 User Registration and Authentication

- Users and professionals must be able to register accounts.
- The system must support secure authentication mechanisms.

3.2.2 User Profile Management

- Users and professionals should be able to create and manage their profiles.
- Users can input details about their urgent repair needs.

3.2.3 Professional Profiles

Professionals can create detailed profiles showcasing their skills, experience, and services offered.

Include a rating and review system for both users and companies to provide feedback.

3.2.4 Service Request Submission

- Users must be able to submit service requests, providing detailed information about the issue.
- Users can post projects, describing their requirements and budget.

3.2.5 Professional Search and Selection

- Users should be able to search for local professionals based on expertise, availability, and reviews.
- Users can select a professional for assistance.

3.2.6 Real-time Communication

- Facilitate communication between users, professionals, and companies within the platform.
- Facilitate communication between users, professionals, and companies within the platform.

3.2.7 Scheduling and Tracking

- Users and professionals should be able to schedule service appointments.
- Implement a tracking system to update users on the status of their service request.

3.2.8 Direct Hiring

Allow users and companies to hire professionals directly from the platform for subsequent projects.

3.3 Non-functional Requirements

3.3.1 Performance

- The system must handle concurrent user interactions efficiently.
- Response times should be within acceptable limits.

3.3.2 Security

- Implement secure data transmission (HTTPS).
- Ensure the confidentiality and integrity of user data.

3.3.3 Usability

- The user interface should be intuitive and easy to navigate.

3.3.4 Technology Constraints

The system must be compatible with major web browsers (Chrome, Firefox, Safari, Edge).

Mobile responsiveness must be ensured for a variety of devices.

3.4 Performance Requirements

3.4.1 Response Time

The system should respond to user interactions within 3 seconds.

3.4.2 Scalability

The platform should handle a growing number of users and projects without significant performance degradation.

3.4.3 Technology Performance

Optimize frontend code for fast loading using tools like Webpack.

Implement server-side caching for frequently accessed data.

3.5 Logical Database Requirements

3.5.1 Database Structure

Design a relational database to store user profiles, project details, messages, and transaction records.

3.5.2 Data Integrity

Implement mechanisms to maintain data integrity and consistency.

4.Tools and Technologies:

4.1 Development:

- **Front-end:** HTML5, CSS3, JavaScript, React js for building a user-friendly and responsive interface.
- **Back-end:** Languages like Python (Django/Flask) or JavaScript (Node.js) for server-side logic and database interaction.
- **Database:** MongoDB(or any other suitable relational e.g. MySQL or NoSQL)
- **Maps API:** Integration with Google Maps or Mapbox for location services and displaying professional locations.
- **Authentication:** JSON Web Tokens (JWT)

- **Payment gateway:** Secure payment processing integration like Stripe or PayPal to handle transactions.

4.2 Testing:

- **Unit testing frameworks:** Frameworks like Jest or Mocha to ensure individual components function properly.
- **End-to-end testing tools:** Tools like Cypress or Selenium to simulate user behavior and test overall functionality.
- **Mobile device testing:** Tools like emulators or real device testing to ensure a seamless experience on mobile platforms.

4.3 Deployment:

- **Cloud platforms:** Cloud providers like AWS, Google Cloud Platform, or Azure offer infrastructure for hosting the application(Heroku, etc...).
- **Containerization:** Docker containers can be used to package the application and its dependencies for easy deployment across environments.
- **Continuous integration/continuous delivery (CI/CD):** Automate the build, test, and deployment process for efficient updates.

Supporting information

Table of content

1- INTRODUCTION

- Purpose
- Scope
- objectives

2- DESCRIPTION

- System architecture
- Product descriptive
- User characteristics
- User roles
- Features and Functionalities
- Constraints
- Assumptions and Dependencies

3- SPECIFIC REQUIREMENTS

- External interfaces
- Functional requirements
- Non-functional requirements
- Performance requirements
- Logical Database requirements

4- TOOLS AND TECHNOLOGIES

- Development
- Testing
- Deployment