

Software Requirements Specifications

Localized Professional Service And Product Network

Project Code:

LPSN-0099

Internal Advisor:

Dr. Hassam Ali

External Advisor:

Mr. Irfan Olak

Project Manager:

Prof. Dr. Muhammad Ilyas

Project Team:

Muhammad Rizwan (Team Lead) (BSCS51F22S003) TL
Dua Kiran Faraz (BSCS51F22S002) TM
Waleeja Hafeez (BSCS51F22S049) TM

Submission Date:

13/11/2025

Project Manager's Signature

Document Information

Category	Information
Customer	(UOS)
Project	Localized Professional Service And Product Network
Document	Software Requirement Specifications
Document Version	1.1
Identifier	LPSN-0099
Status	Draft
Author(s)	Team of this project "LPSN" (Muhammad Rizwan ; Dua Kiran Faraz ; Waleeja hafcez)
Approver(s)	Prof Dr.Muhammad Ilyas
Issue Date	13 Nov,2025
Document Location	
Distribution	1. Prof Dr.Muhammad Ilyas 2. Dr.Hassam Ali 3. Project team

Definition of Terms, Acronyms and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Term	Description
LPSN	Localized skilled workers and product network
AWS	Amazon web services

Sept. 15, 2003
14

Page 2 of

1. Introduction

1.1 Purpose of Document

This document provides a detailed description of the Software Requirements Specification (SRS) for our project (localized professional service and product network "LPSN"). It outlines the purpose, features, constraints, interfaces of the system. It explains what the system will do, its features, limitations and how it will interact with users.

1.2 Project Overview

The LPSN is a web and mobile-optimized platform designed to facilitate economic activity and community building within a specific, restricted geographical area (Geo-Fence), such as a village, towns, or society. The software will connect three primary user groups: Clients, Local Professionals/Skilled Workers (Services), and Local Vendors (Products). Its key goal is to keep local wealth within the community by making local resources easily available, advertised, and sold through a unified digital window.

1.3 Scope

Included in Scope (The system WILL do):

1. The system will contain a dedicated 'Service Listing' module for professionals to describe their skills, rates, and availability.
2. Search and filtering functionality for services and products based on categories and proximity.
3. A dedicated 'Product Marketplace' module for vendors to advertise, display, and sell local products.
4. An integrated communication/chat feature for users to connect directly (Client-Professional/Vendor).
5. A user rating and review system for services and products.

Excluded from Scope (The system WILL NOT do):

1. Advanced inventory management or logistics/delivery features.
2. Complex features like bidding for services or integrated scheduling/calendar management.
3. Integration with external social media platforms for login/sharing.

2. Overall System Description

2.1 User characteristics

Most important user:**Client:**

Residents within the Geo-Fence seeking local services or products. May have varying levels of technical proficiency (requires simple, intuitive interface).

Professional / Workers:

Skilled individuals offering services (plumbers, tutors, technicians). Needs easy profile creation and listing management.

Vendors:

Individuals or small businesses selling local products (food, crafts, produce). Needs simple product listing and order management.

Admin:

System administrators responsible for user verification, content moderation, and Geo-Fence management.

2.2 Operating environment

The software will be a web-based application (optimized for mobile web browsing) and potentially a cross-platform mobile app.

1. Hardware Platform (Server):

The system will run on cloud infrastructure services like Amazon web services so that we could assure that system will remain reliable and can be accessible from anywhere i.e (*Cloud-based server (e.g., AWS, Google Cloud etc)*)

2. Operating System (Server):

The backend server will run on a Linux-based operating system (e.g., Ubuntu Server, Amazon Linux) that will be hosted on cloud infrastructure. This choice is based on its stability, performance, and widespread support for Python/Node.js backend frameworks i.e (Linux (Ubuntu or similar)

3. Operating System (Client):

Any modern OS supporting a web browser (Windows, macOS, Android, iOS).

4. Development Environment:

Node.js (Express framework), JavaScript, MongoDB (NoSQL database).

5. Web Browser:

Latest stable versions of Chrome, Firefox, Safari, and Edge.

2.3 System constraints

The system constraints are discussed briefly according to the right sequence . The constraints are discussed as:

1. Software constraints:

Must be developed using the specified MERN stack technologies: Node.js, JavaScript, MongoDB. No use of legacy frameworks.

2. Hardware constraints:

Must operate on standard cloud hosting infrastructure. Mobile optimization is mandatory.

3. Legal constraints:

The system must comply with local data privacy laws relevant to the Geo-Fence region.

4. User constraints:

The software must be easy to understand and use because mostly skilled workers are unaware of technologies and it must be easy to handle for them.

5. Off the shelf components :

This section might be updated in next some versions of this specific document because for now there are no such constraints regarding to this aspect

3. External Interface Requirements

The external interface constraints are discussed as follows

3.1 Hardware Interfaces

The system will interact with standard client-side hardware components:

1. Mobile Devices:

Must utilize the device's GPS/location services for Geo-Fence verification during registration and location-based searching.

2. Input Devices:

Standard mouse, keyboard, and touchscreens.

3. Display:

Responsive design to accommodate various screen sizes (mobile, tablet, desktop)..

3.2 Software Interfaces

1. Database:

MongoDB (Version 5.x or later) for persistent data storage.

2. Server Runtime:

Node.js (Version 18.x or later) with Express.js framework.

3. Mapping API

(Optional but recommended for Geo-Fence): Integration with a third-party mapping API (e.g., Google Maps API) to define and verify user locations against the Geo-Fence boundary.

4. Authentication/Security Libraries:

Relevant Node.js/Express libraries for handling secure login and token management (e.g., JWT).

3.3 Communications Interfaces

1. Network Protocol: HTTPS (for all communication).
2. Web Standard: RESTful API for communication between client (frontend) and server (Node.js backend).
3. Chat/Messaging: Web-socket protocol (e.g., using Socket.io) for real-time messaging between Clients and Professionals/Vendors.
4. Email: SMTP service integration for sending confirmation emails, password resets, and notifications.

4. Functional Requirements

F1: User Management (Authentication & Authorization)

1. **Registration:** The system shall allow new users to register as a Client, Professional, or Vendor.
2. **Geo-Fence Verification:** The system shall require users to confirm their location is within the defined Geo-Fence during registration.
3. **Login:** The system shall allow users to log in securely using an email/username and password.
4. **Profile Management:** The system shall allow all users to update their personal information, password, and location details.
5. **Role-Based Access:** The system shall enforce access controls based on the user's role (Client, Professional, Vendor, Admin).

F2: Professional & Service Listing Module:

1. **Listing Creation:** Professionals shall be able to create one or more service listings, including Title, detailed Description, Category (e.g., Plumbing, IT, Teaching), Rates, and Availability.
2. **Service Search:** Clients shall be able to search for services by keyword, category, and within a specified radius (proximity) within the Geo-Fence.
3. **Rating Display:** Each service listing shall display its average user rating and total number of reviews.
4. **Contact Professional:** Clients shall be able to initiate a private chat with a Professional directly from the listing page.

F3. Vendor & Product Marketplace Module

1. **Product Listing:** Vendors shall be able to list products for sale, including Product Name, Description, Price, Stock Status (In Stock/Out of Stock), and product images.

2. **Product Search:** Clients shall be able to search and filter products by keyword, category (e.g., Produce, Crafts, Baked Goods).
3. **Shopping Cart:** Clients shall be able to add multiple products from one or more vendors to a temporary shopping cart.
4. **Order Request:** Clients shall be able to submit an order request (confirming items and quantity) which is sent to the respective Vendor(s).

F.4. Communications Module

1. **Real-Time Chat:** The system shall provide a secure, real-time private messaging feature between Clients and Professionals/Vendors for inquiry and transaction discussion.
2. **Notifications:** The system shall generate notifications (e.g., "New Order Request," "New Message") for relevant user actions.

F.5. Review & Feedback Module

1. **Service Review:** Clients shall be able to submit a rating (1-5 stars) and a textual review for a Professional after a service has been completed (based on order status).
2. **Product Review:** Clients shall be able to submit a rating and review for a purchased product.

5. Non-functional Requirements

5.1 Performance Requirements

1. **Response Time:** 90% of all page loads and API calls (excluding file uploads) must complete within 2 seconds under peak load conditions (up to 50 concurrent users).
2. **Scalability:** The system must be designed to handle scaling of the MongoDB database and Node.js instances to support up to 1000 to 10,000 active registered users within the Geo-Fence.

5.2 Safety Requirements

1. **Data Integrity:** All data stored in MongoDB must be validated upon entry to prevent injection attacks and ensure consistency.
2. **Input Validation:** The system must employ strict input validation on all user input fields to prevent Cross-Site Scripting (XSS) and other vulnerabilities.
3. **Error Handling:** The system shall provide clear, non-technical error messages to the user and log detailed technical errors on the server side.

5.3 Security Requirements

1. **User Authentication:** Passwords must be stored using strong, one-way hashing algorithms (e.g., bcrypt).
2. **Data Transmission:** All data exchanged between the client and server must be encrypted using SSL/TLS (HTTPS).
3. **Authorization:** The system must implement robust access control to ensure a user can only modify their own profile, listings, and orders.
4. **Admin Access:** Administrative functions must be restricted to a limited set of verified Admin users via a separate, secure portal.

5.4 User Documentation

The following user documentation components will be delivered with the software:

1. Online Help Center/FAQ for Clients.
2. Quick-Start Guide for Professionals on creating service listings.
3. Quick-Start Guide for Vendors on adding products and managing orders.

6. Assumptions and Dependencies

Assumptions:

1. **User Device Access:** It is assumed that the majority of target users within the Geo-Fence have access to modern smartphones and a stable internet connection.
2. **Content Upload:** Users (Professionals/Vendors) are responsible for providing high-quality, relevant images and descriptions for their listings and products.
3. **Geo-Fence Definition:** It is assumed that the specific geographical boundaries of the restricted area will be clearly defined and provided to the development team before the initial deployment.

Dependencies:

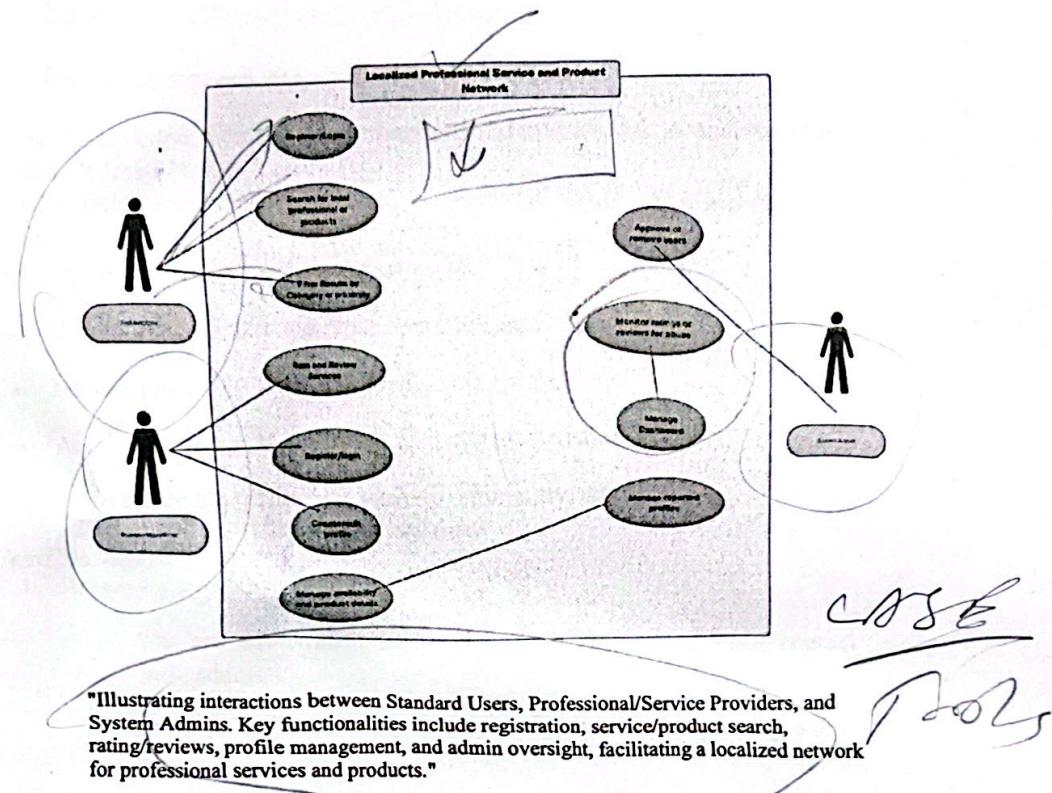
1. **Node.js/MongoDB Stability:** The project is dependent on the continued stability and support of the Node.js runtime environment and the MongoDB database.
2. **Third-Party APIs:** The project is dependent on the uptime and continued availability of any integrated mapping service API (if used for Geo-Fence or proximity searching).

7. References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
1)	Project Proposal	Oct 06, 2025	https://github.com/muhammadrizwani99/FYP

8. Appendices

Figure 1: Localized Professional Service and Product Network Use Case Diagram



Use Case Explanation

Overview of the System:

The system "Localized Professional Service and Product Network" is an online platform that connects local residents/users with professionals or producers offering products or services within a local area.

It also includes a system administrator who manages the platform, ensuring smooth operation and maintaining quality control.

Actors in the System

1. Resident/User

These are customers or people who want to:

- Find local professionals or products.
- Register/login to the platform.
- Search for specific services/products based on category or location.

- Rate and review services after use.

2. Professional/Producer

These are service providers or small business owners who:

- Register/login to the system.
- Create and manage their professional profiles.
- Update availability and product/service details.
- Receive ratings and reviews from residents/users.

Purpose: To advertise their services or products and interact with local customers.

3. System Administrator

This actor manages the entire platform to maintain trust, order, and functionality.

They can:

- Approve or remove user accounts.
- Monitor reviews and ratings for abuse or spam.
- Manage reported profiles and handle complaints.
- Access and manage a dashboard with system statistics.

Purpose: To ensure the system runs securely, efficiently, and fairly.

Use Cases Explained

For Resident/User:

1. Register/Login

- The user creates an account with personal details (e.g., name, contact info, address).
- Logging in allows access to features like searching, reviewing, and rating.

2. Search for Local Professional or Products

- The user searches the database using keywords or categories (e.g., plumber, electrician, home baker).

3. Filter Results by Category or Proximity

- Search results can be refined by distance (nearby providers) or by category (e.g., beauty, repair, food).
- This enhances user experience and relevance of results.

4. Rate and Review Services

- After availing a service or buying a product, users can leave ratings (e.g., stars) and text feedback.
- Helps others make informed decisions and maintains platform credibility.

For Professional/Producer:

1. Register/Login

- o Professionals create accounts to be listed in the system.
- o They can later edit or update their profiles.

2. Create/Edit Profile

- o Includes uploading photos, describing services/products, pricing, and contact info.
- o Keeps their public listing up to date.

3. Manage Availability and Product Details

- o Allows professionals to specify working hours, availability, and product stock details.
- o Keeps users informed about when the service or product is accessible.-

For System Administrator:

1. Approve or Remove Users

- o Admin verifies new registrations to prevent fake or duplicate accounts.
- o Can also remove inactive or fraudulent users.

2. Monitor Ratings or Reviews for Abuse

- o Admin checks for inappropriate, spammy, or biased reviews.
- o Ensures a healthy feedback environment.

3. Manage Dashboard

- o Provides the admin with access to system analytics ----- (user count, activity logs, reports, etc.).

4. Manage Reported Profiles

- o Handles complaints or reports about suspicious profiles, spam, or unprofessional behavior.
- o Admin may take corrective actions such as suspension or warning.

Relationships

Association Lines:

Show which actor interacts with which use case.

Example: The line from Resident/User to Rate and Review Services indicates direct interaction.

Include Relationships (implicit):

- Search for local professional or products → Filter results by category or proximity (filtering is a part of searching).
- Monitor ratings or reviews → Manage Dashboard (admin monitoring is part of dashboard management).