SE 118
Requirement Specification for the Project.
Lab Manual 2

Introduction

Home maintenance has become an essential part of modern living, especially in urban areas where busy schedules and limited time make it difficult for individuals to deal with household issues. Plumbing, in particular, is one of the most frequent and critical services required by both households and businesses. Traditional methods of finding and hiring plumbers often involve word-of-mouth references, manual phone calls, and unreliable service timings. These outdated practices frequently lead to delays, miscommunication, and a lack of trust between customers and service providers.

To address these challenges, digital platforms are increasingly being adopted across service industries. A structured and automated system for plumbing services can save time, reduce errors, and create transparency between customers, plumbers, and administrators. The Plumbing Service Management System (PSMS) has been conceptualized to fill this gap by offering a reliable, user-friendly, and efficient solution.

1. Project Scope

The Plumbing Service Management System is designed to overcome the common problems of existing systems, such as delays, miscommunication, and difficulty in tracking jobs. It helps customers easily request plumbing services, track job progress, and view assigned plumber details. At the same time, it allows administrators to manage customers, plumbers, and services while updating job statuses and costs. By automating plumber assignment, job tracking, and cost calculation, the system ensures efficient service management, reduces manual errors, and provides transparency for customers with better control for administrators. Future upgrades may include online payment, invoicing, and customer feedback options.

Step 2: Overall Description

1. Product Perspective

The main purpose of the Plumbing Service Management System is to make plumbing services easier by helping customers book services, track jobs, and letting admins manage plumbers and costs.

2.Product Features

Customer Management – Store and manage customer details like name, phone, and address.

Plumber Management – Maintain plumber information including experience and specialization.

Service Catalog – List available services (Installation / Repair) with descriptions and costs.

Job Assignment – Assign a plumber and service to a customer request.

Status Tracking – Track job progress (Pending, In Progress, Completed).

Cost Calculation – Auto-calculate cost based on service type and requirements.

Job Details Display – View full job details including assigned plumber, customer, and service.

Customer History – Keep a record of all past jobs for each customer.

Plumber Performance – Track plumber work history and specialization-based performance.

Service Records – Maintain details of completed and ongoing services.

Update Mechanism – Update job status and information in real-time.

Easy Reporting – Generate summaries of customers, plumbers, and service usage.

3. User Classes and Characteristics:

(a) Customers / End Users
➔ Frequency of use: Occasionally (only when they need plumbing service)
➔ Functions used: Request service, view assigned plumber, track job progress, check cost
➔ Expertise: Basic smartphone or computer knowledge
➔ Privileges: Limited to their own service requests and information
➔ Importance: Most important user class, as they are the main service users

(b) Plumbers
➔ Frequency of use: Regular (whenever jobs are assigned)
➔ Functions used: View job details, update job status, complete assigned tasks
➔ Expertise: Basic to moderate technical knowledge (using phone or system to update jobs)
➔ Privileges: Access only to their assigned jobs and related details
➔ Importance: Very important, since they are the service providers

(c) Administrators
➔ Frequency of use: Daily (manage the whole system)
➔ Functions used: Manage customers, plumbers, services, assign jobs, update costs, generate reports
➔ Expertise: Moderate computer knowledge
➔ Privileges: Full access (can add, edit, or remove users, plumbers, and jobs)
➔ Importance: Critical for smooth operation of the system

4. Operating Environment

This software, designed for managing plumbing service jobs, will be developed and will operate primarily using the Java programming language.

Hardware Platform

- **Personal Computer (PC) / Laptop:** The software will run on a standard desktop or laptop computer, which will be used for both development and running the application.

Operating System

- **Windows:** The application will be developed and tested on the Windows operating system.
- **macOS:** It should also be compatible with macOS, as Java is a cross-platform language.
- **Linux:** The application can also run on various Linux distributions like Ubuntu.

Software Dependencies

- **Java Development Kit (JDK):** The project requires JDK 8 or a more recent version to compile and run the Java code.
- **Integrated Development Environment (IDE):** An IDE like IntelliJ IDEA, Eclipse, or NetBeans is needed for writing and managing the code.
- **Database:** For simplicity, the project can use a lightweight, file-based database like SQLite or an in-memory database like H2 Database that can be run without a separate server setup. Alternatively, you could just store data in simple text files or serialized objects if the project scope is very small.

Other Requirements

- **Local Machine Operation:** The software will run on a single machine and is not intended for a network or client-server environment.
- **No external APIs or services:** It will not rely on external web services, APIs, or cloud platforms.
- **Command-Line Interface (CLI) or Basic GUI:** The user interface can be a simple text-based command-line interface or a basic graphical user interface (GUI) built with Java's Swing or JavaFX libraries.

5.Design and System Constraints

Hardware Limitations

- The system should run on a standard PC or laptop.
- No high-end hardware is assumed; memory and speed are limited.

Software / Technology

- You will use **Java** for coding.
- Database can be **MySQL, SQLite, or even simple file storage**.

Security Considerations

- Basic input validation is required (no advanced encryption needed).
- Prevent simple errors like empty fields or invalid numbers.

Design / Programming Standards

- Modular and well-documented code.
- Simple naming conventions for classes, variables, and functions.
- Easy to maintain and understand for beginners.

Step 3: System Features

Feature 1: Account Management (Register + Login)

Description and Priority

- **Description:** Allows new customers to register with their details and existing customers to log in securely using phone number and password/OTP.
- **Priority:** High
- **Priority Components (1–9 scale):**
    a. **Benefit:** 9 (essential for onboarding and secure access)
    b. **Penalty:** 9 (without it, booking and other functions cannot be used)
    c. **Cost:** 4 (moderate implementation cost)
    d. **Risk:** 7 (risk of unauthorized access if authentication fails)

Stimulus/Response Sequences

- Customer selects “Register” or “Login”.
- System prompts for name, phone, address (for registration) or phone + password/OTP (for login).
- Customer enters input.
- System validates data against the database.
- If valid → access granted; if invalid → error message with retry option.

Functional Requirements

- **FR1.1** The system shall allow new customers to register with name, phone, and address.
- **FR1.2** The system shall validate uniqueness of the phone number.

- **FR1.3** The system shall allow existing customers to log in with valid credentials.
- **FR1.4** The system shall display error messages for invalid credentials.

Feature 2: Book Service

Description and Priority

- **Description:** Enables customers to book plumbing services (Installation/Repair) by providing type, description, and schedule. A new job is created with status = Pending.
- **Priority:** High
- **Priority Components (1–9 scale):**
    - **Benefit:** 9 (core function of the system)
    - **Penalty:** 9 (system is useless without service booking)
    - **Cost:** 5 (moderate implementation cost for validation and storage)
    - **Risk:** 6 (risk of errors if booking is not recorded properly)

Stimulus/Response Sequences

- Customer selects “Book Service”.
- System prompts for service type (Installation/Repair), description, and schedule.
- Customer enters details and confirms.
- System creates a new job with status = Pending.
- System shows booking confirmation.

Functional Requirements

- **FR2.1** The system shall allow customers to select service type and provide schedule.

- **FR2.2** The system shall create a new job record with status = Pending.
- **FR2.3** The system shall display booking confirmation and job ID to the customer.

Feature 3: Plumber Information (Registration)

Description and Priority

- **Description:** Store plumber details such as name, phone, experience, and specialization when a plumber is added to the system.
- **Priority:** High
- **Priority Components (1–9 scale):**
  - Benefit: 9 (essential for assigning jobs correctly)
  - Penalty: 8 (incorrect or missing data causes assignment errors)
  - Cost: 4 (moderate effort to implement)
  - Risk: 7 (data inconsistency may disrupt workflow)

Stimulus Sequences

1. Admin selects “Add Plumber.”
2. System prompts for name, phone, experience, specialization.
3. Admin enters data → System validates and saves record.
4. Confirmation displayed.

Functional Requirements

- FR1. System shall allow adding a new plumber with all required details.
- FR2. System shall validate input (no empty fields, numeric experience).

- FR3. System shall allow editing plumber info.
- FR4. System shall display plumber info in a list.

Feature 4: Job Assignment & Availability (Usage , Assignment)

Description and Priority

- **Description:** Assign plumbers to customer jobs and track their current status (Available, Busy, On Leave).
- **Priority:** High
- **Priority Components (1–9 scale):**
    - Benefit: 9 (critical for workflow and job allocation)
    - Penalty: 8 (wrong assignment causes delays and dissatisfaction)
    - Cost: 5 (moderate implementation)
    - Risk: 7 (incorrect status or assignment affects operations)

Stimulus Sequences

1. Customer requests service.
2. Admin selects an available plumber.
3. System checks availability and specialization.
4. System assigns plumber → updates status to “Busy.”
5. After job completion, plumber updates status to “Available.”

Functional Requirements

- FR1. System shall display available plumbers for assignment.
- FR2. System shall prevent assigning Busy or On Leave plumbers.
- FR3. System shall update plumber status in real-time.
- FR4. System shall notify plumbers of assigned jobs.

Feature 5: Service Definition & Creation

● Description and Priority

- **Description:** Allows administrators to define and add new plumbing services (e.g., Installation, Repair) with basic details like description and base cost.
- **Priority:** High

Priority Components (1–9 scale):

- **Benefit:** 9 (essential for running the system)
- **Penalty:** 8 (system useless without services)
- **Cost:** 4 (straightforward implementation)
- **Risk:** 5 (minor if misconfigured)

● Stimulus/Response Sequences

1. Admin selects **Add New Service** option.
2. Admin enters service details (Service ID, type, description, base cost).
3. System validates and saves the service.
4. Service becomes available for customers to request.

● Functional Requirements

- **FR1.** The system shall allow admins to create new services with unique IDs.
- **FR2.** The system shall store service details (type, description, cost).
- **FR3.** The system shall make services available for customers to view and request.

Feature 6: Service Cost Calculation

● Description and Priority

- **Description:** Calculates the total cost of a service based on type, complexity, and any additional charges.
- **Priority:** High

Priority Components (1–9 scale):

- **Benefit:** 9 (customers need accurate billing)
- **Penalty:** 8 (wrong pricing leads to confusion)
- **Cost:** 5 (moderate effort)
- **Risk:** 6 (medium risk if formula is wrong)

● Stimulus/Response Sequences

1. Customer requests a service.
2. System checks service type (Installation/Repair).
3. System applies base cost and any additional charges.

4. System displays the final cost to customer and admin.

- **Functional Requirements**
  - **FR1.** The system shall calculate service costs based on service type.
  - **FR2.** The system shall display the calculated cost before confirming a job.
  - **FR3.** The system shall allow future updates in cost formulas.

Step 4: External Interface Requirements

(a) User Interfaces

- **Logical Design:** The system will have a clean and simple layout with login, dashboard, job management, and report sections.
- **GUI Standards:** Easy-to-use, responsive design that works on desktop and mobile.
- **Error Messages:** Simple and clear messages (e.g., “Invalid login, please try again” or “Service not available”).
- **UI Components:**
  - Login Page (for customers, plumbers, admins)
  - Customer Dashboard (request service, check status)
  - Plumber Dashboard (view assigned jobs, update status)
  - Admin Dashboard (manage users, services, and jobs)
  - Reports Page (for admin)

- **Accessibility:** Should support simple navigation, readable fonts, and large text option for better usability.

(b) Hardware Interfaces

- **Devices:** The system will run on computers and smartphones.
- **Data/Control:** Users interact with the system through web or mobile interface.
- **Connections:** Internet connection required to connect to the server and database.
- **Examples:**
  - Customer uses mobile/PC to request a service.
  - Admin uses desktop to manage jobs.
  - Plumber uses phone to update job status.

(c) Software Interfaces

- **Database:** MySQL or any relational database for storing users, plumbers, services, and jobs.
- **Operating System:** Works on Windows, Linux, or Android/iOS (via browser/app).
- **Framework/Tools:**
  - Backend: Java/Python/Node.js (depending on choice).

    - Frontend: HTML, CSS, JavaScript (or any simple framework).
- **APIs:** REST APIs to connect frontend with backend.
- **Data Exchange:** JSON format for data transfer (e.g., `{ "job_id": 101, "status": "Completed" }`).

(d) Communication Interfaces

- **Communication Method:** Internet-based (web app/mobile app).
- **Protocols:** HTTPS for secure communication between user and server.
- **Email/SMS:** Optional feature to notify customers about job updates.
- **Security:** Login authentication, encrypted data storage, and secure transactions.
- **Performance:** System should respond within 3 seconds for user actions like login, job creation, or status updates.

(e) Nonfunctional Requirements

- **Performance Requirements:**
    - System should handle at least 100 users at the same time.
    - Job status updates should appear instantly to customers and admins.
- **Safety Requirements:**
    - System should regularly back up data to prevent loss.
    - Wrong operations (like deleting services/jobs) should ask for confirmation.

- **Security Requirements:**
  - Password-protected login for all users.
  - Accounts locked after 3 wrong login attempts.
  - Data stored in database should be encrypted.

- **Software Quality Attributes:**
  - **Usability:** Easy to use even for non-technical users.
  - **Reliability:** System should be available 24/7 with minimum downtime.
  - **Maintainability:** Code should be modular so updates can be done easily.
  - **Portability:** Can run on different devices (mobile, desktop, tablet).
  - **Flexibility:** Easy to add new features like online payment or feedback in the future.

Team Contribution

This project was completed as a group work with the contributions of three members. The distribution of work among the team members is as follows:

• Tamanna Mogul

Contributed to:

1.User Classes and Characteristics

2. Design and Implementation Constraints

3.System Feature

4.Operating Environment

• Md. Meiad Khan

Contributed to:

1.Project Scope

2.Project Perspective

3.System Feature

4: External Interface Requirements

• Md. Shafinur Rahman

Contributed to:

1.Introduction

2. Product Features

3.Operating Environment

4.System Feature