SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Project Title: Accessible Places for Differently Abled

## Submitted by

- Soham Datar – Roll No. 08  
- Kaustubh Gund – Roll No. 17

- Yash Manore – Roll No. 31

- Rex Merciline – Roll No. 32  
- Department: Information Technology  
- Institution: Don Bosco Institute of Technology (DBIT)

## Guided by

- Prof. Prasad Padalkar  
- Designation: Asst. Prof & IQAC Coordinator  
- Department: Information Technology

# TABLE OF CONTENTS

1. Introduction  
2. Overall Description  
3. Specific Requirements  
4. System Models (if any)  
5. Appendices

# 1. INTRODUCTION

## 1.1 Purpose

This SRS document defines the functional and non-functional requirements for our mini project titled **“Accessible Places for Differently Abled.”** The project aims to provide a map-based user interface that helps users locate and review public places in Thane based on their accessibility for differently abled individuals.

## 1.2 Scope

The system is designed to assist users in locating and exploring public places in Thane that are accessible to differently abled individuals. It includes features such as an interactive map with category-wise filtering, accessibility enhancement tools (dark mode, high contrast, font resizing), and UI screens for reviewing and listing place-specific accessible features.

## 1.3 Definitions, Acronyms, and Abbreviations

- UI – User Interface

- UX –User Experience  
- DB – Database  
- API – Application Programming Interface

- SRS – Software Requirements Specifications

- WCAG – Web Content Accessibility Guidelines

- HTML – Hyper Text Markup Language

- CSS – Cascading Style Sheet

- JS – JavaScript

- HTTP – Hyper Text Transfer Protocol

## 1.4 References

- Leaflet.js Documentation – <https://leafletjs.com>

- Bootstrap 5 Documentation – <https://getbootstrap.com>

- WCAG Guidelines – <https://www.w3.org/WAI/standards-guidelines/wcag/>

- Mozilla Developer Network (MDN) Web Docs – <https://developer.mozilla.org>

- Font Awesome Icons – <https://fontawesome.com>

# 2. OVERALL DESCRIPTION

## 2.1 Product Perspective

This is a standalone system initially developed as a web-based frontend application using HTML, CSS, JavaScript, and Leaflet.js. In its current stage, the system focuses on the user interface and accessibility features. In future iterations, the project will be extended to include a Java-based backend, which will handle functionalities such as user authentication, review storage, and database interactions.

## 2.2 Product Features

**- Login and Signup:** A user authentication system for accessing personalized features. Currently implemented as UI; backend functionality will be integrated using Java.

- **Interactive Map View**: An embedded Leaflet.js map shows accessible places in Thane with markers, popups, and category-wise filtering (e.g., hospitals, malls, schools).

**- Accessibility Controls:** Users can toggle dark mode, enable high contrast, and adjust font size for better usability aligned with accessibility guidelines.

**- Data Entry and Retrieval:** Users can leave reviews for places (currently UI-only). Future versions will store and retrieve data from a backend database via Java.

**- Report Generation:** Planned backend feature to generate reports summarizing accessible locations and user-submitted reviews.

**- Admin Panel:** A future dashboard for admins to manage place data, approve reviews, and update accessibility info.

## 2.3 User Characteristics

- Admin: Has full access to all system functionalities, including managing place data, editing accessibility features, reviewing submitted user feedback, and generating reports. Admins are expected to be technically proficient and familiar with the accessibility domain.  
- User: General users can view the interactive map, apply category filters, adjust accessibility settings, and (in future versions) log in to submit reviews. Users may include residents, tourists, or caregivers looking for accessible locations. No technical background is required to use the system.

## 2.4 Constraints

- The system must run on modern web browsers (Chrome, Firefox, Edge) on Windows or Linux operating systems.  
- The current version is built using HTML, CSS, JavaScript, and Leaflet.js for frontend only.

- The backend will be developed using **Java (JDK 17)** and **MySQL** for database management in future phases.

- Internet connection is required to load map data from OpenStreetMap and access external libraries (e.g., Bootstrap, Leaflet).

- User authentication and data persistence are not functional in the current UI-only version.

## 2.5 Assumptions and Dependencies

- Java (JDK 17 or higher) must be pre-installed on the system for backend execution.  
- A MySQL database server must be running and properly configured for data storage and retrieval.

- The system assumes the user has a stable internet connection to load map tiles and external resources (e.g., Bootstrap, Leaflet, Font Awesome)

- Web browsers used should support modern JavaScript and CSS features.

- For future integration, a web server (e.g., Apache Tomcat) will be required to host the Java-based backend.

# 3. SPECIFIC REQUIREMENTS

## 3.1 Functional Requirements

|  |  |  |
| --- | --- | --- |
| ID | Requirement | Description |
| FR1 | Login | User can log in using email and password credentials. |
| FR2 | Add Data | Admin can add new records about accessible places. |
| FR3 | Generate Report | System can generate a summary report in PDF format. |
| FR4 | Search | User can search for a place by ID or name. |
| FR5 | Filter by Category | User can filter map markers by category (e.g., hospital, mall, park) |
| FR6 | Toggle Accessibility | User can toggle dark mode, high contrast, and font size adjustments. |
| FR7 | Submit Review | User can submit a star rating and review (UI only; backend planned). |

## 3.2 Non-functional Requirements

- Performance: The system must respond to user interactions (e.g., map filter, page loads) within **2 seconds** under normal network conditions.  
- Security: User passwords must be securely stored using **hashing algorithms** in the backend (e.g., SHA-256, bcrypt) once implemented.  
- Usability: The UI must be **simple, intuitive, and accessible** for users of all skill levels. Accessibility features like high contrast mode, font size adjustment, and keyboard navigation must be supported.

- Reliability: The system should remain stable under normal usage conditions without crashing or freezing, and must load essential components even in slow networks.

- Portability: The application should work consistently across **Windows, Linux, and modern browsers** (Chrome, Firefox, Edge).

- Maintainability: Code should follow a **modular structure** using clean and well-documented HTML, CSS, and JavaScript. Java backend code should be written following standard design patterns to ease future updates and debugging

## 3.3 External Interface Requirements

### 3.3.1 User Interface

- Home Page: Displays an interactive map with location markers, category filters (e.g., hospitals, parks), accessibility controls (dark mode, contrast, font size), and links to login and features list.  
- Login Page: Contains form fields for email and password input, along with options to login or sign up. UI includes accessibility tools and is visually styled using Bootstrap and Font Awesome.  
- Admin Dashboard: (Planned feature) A secure admin panel for managing place data, updating accessibility features, and generating reports. Will be implemented during backend integration using Java.

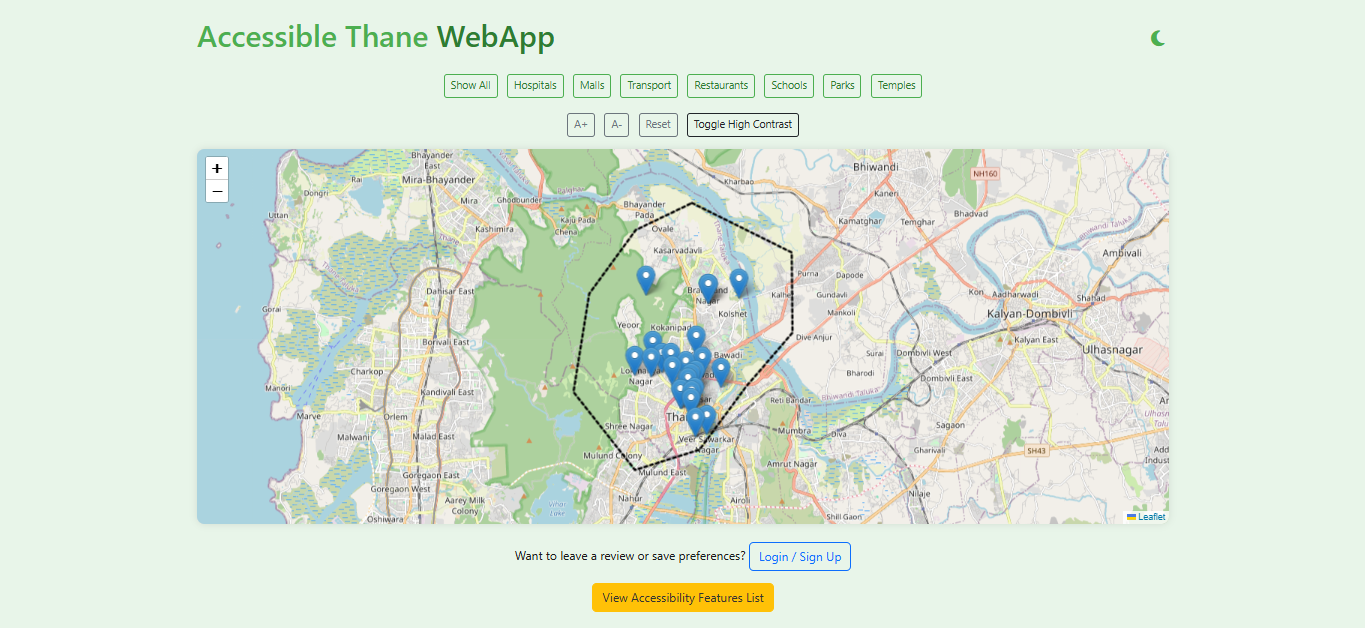
### 3.3.2 Software Interfaces

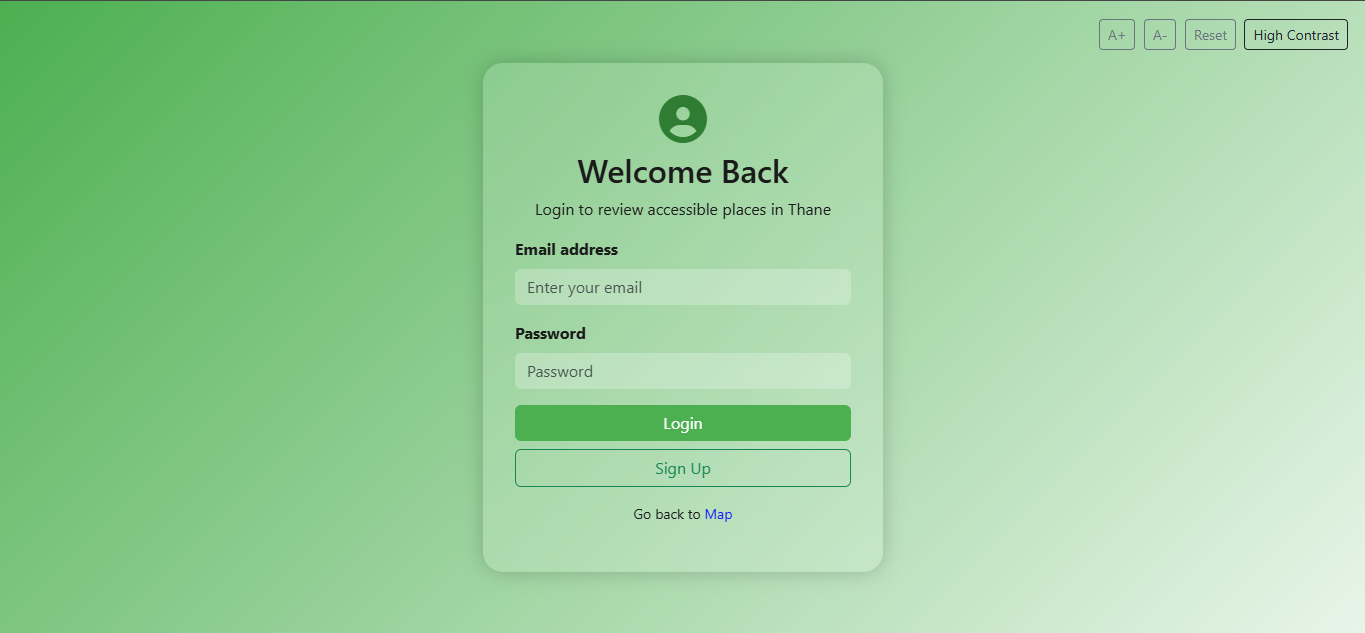
- MySQL database: The backend will interact with a **MySQL** database to store and retrieve user data, reviews, place details, and admin inputs. JDBC will be used to establish the connection from Java.  
- JavaFX/Swing frontend (if GUI): If a desktop-based admin tool is developed in future, **JavaFX or Swing** may be used to create the GUI for managing place records, generating reports, and viewing statistics.

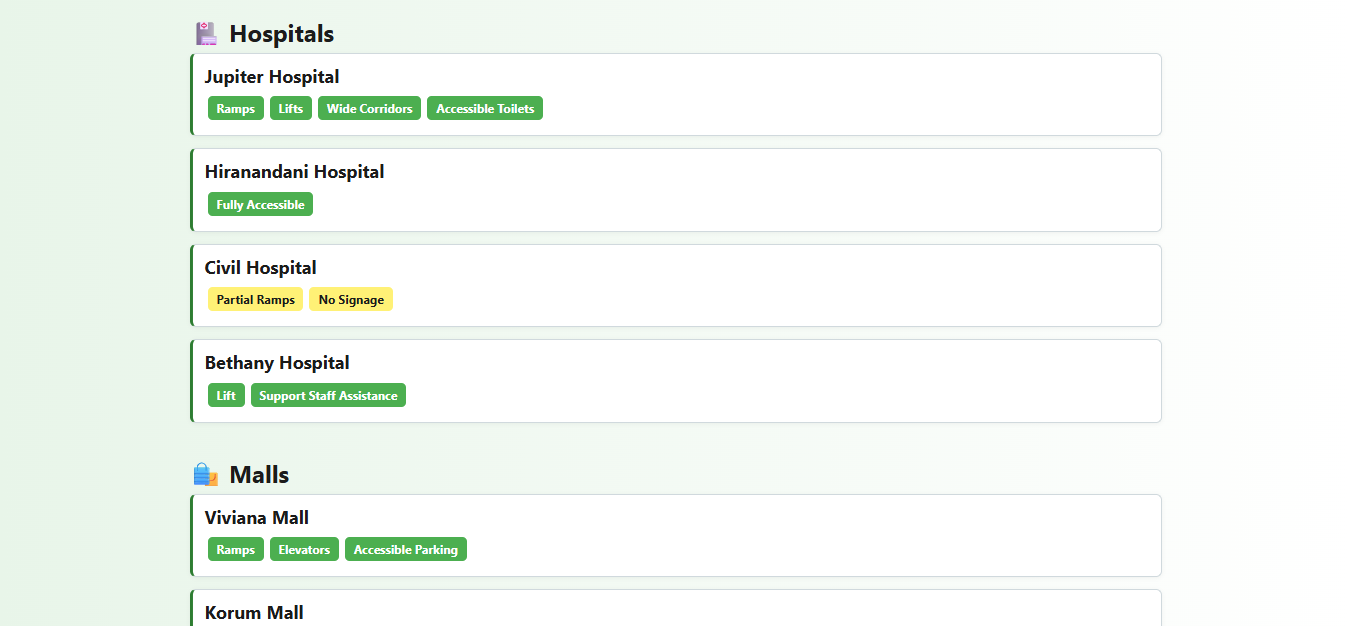
- Leaflet.js API: The web frontend currently uses **Leaflet.js** to render the interactive map and plot markers based on static JavaScript data.

- LocalStorage (WEB): Font size, dark mode, and contrast settings are saved temporarily using **Web LocalStorage** to enhance user experience without backend dependency.

# 4. APPENDICES

- Screenshots of UI  






DECLARATION:  
We hereby declare that this document is a true representation of the software project titled **“Accessible Places for Differently Abled”** developed as part of our academic curriculum. All the work presented here is original and created by us for educational purposes.