**Gym Locator**

Software Requirements Specification

# **1. Introduction**

## 1.1 Purpose

The purpose of this document is to define the requirements for the development of a Gym Locator application. The Gym Locator will allow users to find local gyms based on their location and provide detailed information about each gym.

## 1.2 Scope

The Gym Locator application will include the following features:

* Address input for users to enter their location.
* Gym search functionality to fetch gyms within a specified radius.
* Display of gym information, including name, address, and brief description
* Gym details page with additional information like facilities, classes, and contact details
* Integration with mapping or geolocation APIs to fetch gym data.
* Integration with other APIs for additional gym-related data (optional)
* User-friendly interface with navigation, responsive design, and appropriate styling

## 1.3 Audience

The target audience for the Gym Locator application includes fitness enthusiasts, individuals looking for nearby gyms, and anyone interested in exploring local fitness options.

# **2. Functional Requirements**

## 2.1 Address Input

Users can enter their address or use geolocation to detect their current location.

## 2.2 Gym Search

The application will interact with mapping or geolocation APIs to fetch gyms within a specified radius (e.g., 10km) based on the user's location.

The application will display a list of gyms that match the search criteria, including gym name, address, and brief description.

## 2.3 Gym Details

Users can click on a gym to view additional details.

The application will display the gym's facilities, classes, pricing, opening hours, contact details, and any additional information available.

The application will provide a link or button that directs users to the gym's website page for more information or to book a visit.

## 2.4 Integration with Additional APIs

The application can integrate with other APIs, such as Google Places API, to retrieve additional gym-related data like amenities, ratings, reviews, or images.

The retrieved data can be displayed on the gym details page or within the gym cards.

## 2.5 User Interface

The application will have a user-friendly interface with intuitive navigation and appropriate use of icons, fonts, and CSS animations.

The application will employ responsive design principles to ensure usability on various devices and screen sizes.

# **3. Non-Functional Requirements**

## 3.1 Performance

The application shall provide fast and responsive performance when fetching and displaying gym data.

The gym search results and gym details page shall load within a reasonable time frame.

## 3.2 Security

User registration and authentication (if implemented) shall follow secure practices to protect user information.

API requests shall be made securely using HTTPS protocols.

## 3.3 Usability

The application shall have an intuitive and user-friendly interface, allowing users to easily search for gyms and access gym details.

The user interface shall follow consistent design and adhere to accessibility guidelines.

## 3.4 Reliability

The application shall handle errors and exceptions gracefully, providing appropriate feedback to users when errors occur.

The application shall have robust error handling mechanisms to prevent crashes or data loss.

## 3.5 Technology Stack

The application shall be developed using suitable technologies, frameworks, and libraries, such as HTML, CSS,JavaScript, and a backend framework (e.g., Node.js, Django) if required.

## 3.6 Scalability

The application shall be designed and developed to accommodate a growing number of gyms and users.

The backend infrastructure shall be scalable to handle increased traffic and data storage requirements.

## 3.7 Documentation

The application's code shall be well-documented, following industry-standard documentation practices.

User documentation or a user guide shall be provided to explain the application's functionality and usage.

## 3.8 Testing

The application shall undergo thorough testing to ensure functionality, performance, and reliability.

Testing shall include unit testing, integration testing, and user acceptance testing.

# **4. Constraints**

The Gym Locator application shall comply with the terms and conditions of the used mapping, geolocation, and any other integrated APIs.

The project should be developed within the given time frame and available resources.

# **5. Assumptions and Dependencies**

The availability and accuracy of mapping or geolocation APIs for fetching gym data within the specified radius.

Access to other APIs (if integrated) and compliance with their terms of use and licensing requirements.

Users have a stable internet connection to access and use the application.

# **6. Development Timeline**

The development of the Gym Locator application will be divided into various phases, including requirements gathering, design, implementation, testing, and deployment.

A detailed development timeline and milestones will be created to ensure a systematic and timely delivery of the application.

This Software Requirements Specification document outlines the functional and non-functional requirements for the Gym Locator application. It provides a clear understanding of the application's purpose, features, and constraints. Use this document as a reference throughout the development process to ensure that the application meets the specified requirements.