

# Software Requirement Specification

February 29, 2024

## 1 Introduction

### 1.1 Purpose

ChargEase is an Android application designed to streamline the electric vehicle (EV) charging experience. It aims to simplify the process of locating, reserving, and using charging stations, thereby promoting the efficient utilization of charging infrastructure and contributing to the sustainable growth of electric mobility.

### 1.2 Document Convention

This document follows the IEEE Standard for Software Requirements Specifications (IEEE Std 830-1998).

### 1.3 Intended Audience and Reading Suggestions

This document contains software functionality, software and hardware requirements and user documentation.

- Developer: The developer who wants to read, change, modify or add new requirements into the existing program may need to first consult this document and update the requirements in an appropriate manner so as not to change the actual purpose of the system or make the system inconsistent.
- User: The user of this program reviews the diagram and the specification provided in the document and check to determine whether the software has all the suitable requirements and if the software developer has implemented all of them. The user can also consult the user guide in the event of any confusion for clarifications.
- Tester: The tester needs this document to prepare their test cases to validate that the initial requirements of this project is actually implemented in the deliverable.

This document need not be read sequentially; users are encouraged to jump to any section they find relevant.

## **1.4 Project Scope**

The scope of the project includes the development of an Android application that provides users with the ability to locate nearby charging stations, check real-time availability, reserve charging slots, and make payments for charging sessions. The application will also feature a user-friendly interface, secure authentication, and authorization mechanisms, and community engagement features.

## **1.5 Overview of Developer's Responsibilities**

- Maintain appropriate coding standards and design.
- Contribute to technical design documentation.
- Contribution in accordance with the software development lifecycle.

# **2 Overall Descriptions**

## **2.1 Product Perspective**

ChargEase is a standalone application that operates on Android devices. It interacts with external systems, such as charging station databases and payment gateways, to provide users with real-time information and secure booking capabilities.

## **2.2 Product Function**

The primary functions of ChargEase include:

- Locating nearby charging stations
- Checking real-time availability of charging slots
- Reserving charging slots
- Making payments for charging sessions
- Providing a user-friendly interface
- Implementing secure authentication and authorization mechanisms
- Encouraging community engagement

## **2.3 User Classes and Characteristics**

The application is designed for EV users who require access to charging stations. Users may vary in technical proficiency, but the application is intended to be user-friendly and accessible to all.

## **2.4 Operating Environment**

ChargEase operates on Android devices running Android OS version 6.0 (Marshmallow) or higher. It requires an internet connection to access real-time data and communicate with external systems.

## **2.5 Design and Implementation Constraints**

- Hardware Limitations: Devices must have Wi-Fi or mobile data that supports IEEE standards.
- Safety and Security Considerations: The system must prevent unauthorized users from accessing the files.
- Reliability Requirements: Total number of bugs in the system shall not exceed 1% of the total line number of code, except connection reliability, which is out of range.

## **2.6 User Documents**

User documentation will be provided within the application, including a user guide and FAQs to assist users in navigating the application and understanding its features.

## **2.7 General Constraints**

The application is subject to constraints related to Android OS compatibility, network availability, and external system dependencies.

## **2.8 Assumptions and Dependencies**

- The interface of the resulting system will be easy to use and accessible without a time or location constraint.
- The users have access to proper internet connection.
- The users have an Android phone.
- It also depends on external systems, such as charging station databases and payment gateways, to provide real-time information and secure booking capabilities.

# **3 External Interface Requirements**

## **3.1 User Interface**

- Interface to provide navigation to the user.
- Interface to provide payment.

- Interface to view and book slots.

### **3.2 Hardware Interface**

NA

### **3.3 Software Interface**

- Flutter
- Firebase

### **3.4 Communication Interface**

The application communicates with external systems via APIs and internet protocols to retrieve real-time data and process transactions.

## **4 Hardware and Software Interface**

### **4.1 Hardware Requirements**

- An Android device to operate.
- Processor: A minimum of 1 GHz processor.
- RAM: A minimum of 1 GB RAM.
- Storage: Sufficient internal storage space to install the application and store user data.

### **4.2 Software Requirements**

- The application operates on Android OS version 6.0 (Marshmallow) or higher.
- Flutter
- Firebase

## **5 Functional Requirements**

This section gives the details of the features and functions this system provides for various users and additional information on how each module works.

### **5.1 Locating Nearby Charging Stations**

- Utilize GPS technology to determine the user’s current location.
- Display nearby charging stations on a map interface.
- Offer options for filtering and sorting charging stations based on user preferences.

### **5.2 Checking Real-Time Availability of Charging Slots**

- Integrate with charging station databases or APIs to fetch real-time data on slot availability.
- Display the availability status of each charging station or slot within the app.

### **5.3 Reserving Charging Slots**

- Allow users to select a preferred charging station and reserve a slot for a specified time period.
- Provide confirmation of the reservation and display relevant details such as location, time, and cost.

### **5.4 Making Payments for Charging Sessions**

- Integrate a secure payment gateway to facilitate transactions within the app.
- Offer multiple payment options and ensure the security of user payment information.

### **5.5 Providing a User-Friendly Interface**

- Design a visually appealing and intuitive interface with easy navigation.
- Use clear and concise language to guide users through the app’s features and functionalities.

### **5.6 Implementing Secure Authentication and Authorization Mechanisms**

- Require users to sign in with credentials or use biometric authentication to access the app.
- Employ encryption protocols to protect user data and secure communication channels.

## 5.7 Encouraging Community Engagement

- Incorporate features for users to share their charging experiences, ratings, and reviews.
- Facilitate communication and interaction among users through forums, chat rooms, or social media integration.
- Offer incentives or rewards for active participation and contribution to the community.

# 6 Non-Functional Requirements

## 6.1 Performance Requirements

- Response Time: The application must respond to user interactions within a reasonable time frame (e.g., less than 2 seconds).
- Loading Time: The application must load and display data within a reasonable time frame (e.g., less than 5 seconds).
- Transaction Processing Time: The application must process payment transactions within a reasonable time frame (e.g., less than 10 seconds).
- Network Latency: The application must minimize network latency to ensure a smooth user experience.
- Battery Consumption: The application must minimize battery consumption for prolonged usage.
- Memory Usage: The application must optimize memory usage to prevent crashes or slowdowns.

## 6.2 Safety Requirements

- User Safety: The application must not compromise user safety.
- Data Safety: The application must protect user data and payment information.
- Charging Station Safety: The application must ensure that charging stations are safe and compliant with relevant standards.

## 6.3 Security Requirements

- Authentication: The application must authenticate users securely.
- Authorization: The application must authorize users based on their roles and permissions.

- Data Protection: The application must protect user data and payment information using encryption.
- Secure Communication: The application must use secure communication protocols.
- Payment Security: The application must use a secure payment gateway.

## 6.4 Software Quality Attributes

- Reliability: The application must be reliable and available for use.
- Usability: The application must be intuitive and user-friendly.
- Interoperability: The application must be interoperable with external systems.
- Accessibility: The application must comply with accessibility standards.
- Localization: The application must support multiple languages.
- Integration: The application must integrate seamlessly with external systems.
- Feedback: The application must provide timely feedback to users.
- Data Synchronization: The application must synchronize data across devices.
- Testability: The software must be modularized for easy testing.
- Extensibility: The software must support modifications without failure.
- Robustness: The system should handle operations without error.

## 7 Other Requirements

Besides functional and non-functional requirements, there are other requirements like user friendliness and tolerance of errors.

- User friendliness: System should provide an easy-to-understand and user-friendly interface.
- Tolerance: System should maintain a minimum accuracy level.
- Licenses: The application must adhere to licenses and regulations.
- Permissions: Sufficient permissions must be obtained to work with user data.