System Requirements Documentation for EVCharge Hub

1. Introduction

1.1 Purpose of the Document

This technical blueprint outlines operational parameters for *EVCharge Hub*, a B2C SaaS solution enabling EV owners to geolocate, reserve, and transact at charging stations. Governs developer-stakeholder agreements.

1.2 Scope of the Product

EVCharge Hub is an online solution that enables EV drivers to identify, book, and pay for charging stations in real time. The platform offers features such as live station availability updates, route optimization with charging stops, and secure transactions. Its primary objective is to alleviate range anxiety by ensuring seamless access to charging infrastructure.

1.3 Document Structure

This SRS is organized into the following sections:

- **Section 2:** General system overview
- **Section 3:** Detailed requirements
- **Section 4:** Supplementary information

1.4 Terminology and Acronyms

- **EV:** Electric Vehicle
- **EVSE:** Electric Vehicle Supply Equipment
- **OCPP:** Open Charge Point Protocol
- **API:** Application Programming Interface
- **kWh:** Kilowatt-hour

2. System Overview

2.1 System Context

EVCharge Hub is a web application that integrates with mapping services, payment gateways, and charging station networks. It operates on cloud-based infrastructure to support location tracking, transaction processing, and user data storage.

2.2 Technical Components

- Mobile-friendly website
- User registration and authentication
- Charging station search with filters (e.g., location, connector type, availability)
- Secure payment processing
- Administrative controls for station management

2.3 Target Users

- **EV Drivers:** Individuals seeking charging stations, ranging from tech enthusiasts to general users.
- **Station Operators:** Business owners managing station details such as pricing and availability.
- **Administrators:** Personnel overseeing system operations, user accounts, and content moderation.

2.4 Limitations

- Ensures smooth functionality across all popular browsers, including Chrome, Firefox, Safari, and Edge
- Designed with a responsive layout to support both smartphones and tablets
- Adheres to PCI-DSS regulations to guarantee secure handling of payment transactions
- Utilizes external APIs to fetch up-to-date information from stations in real time

2.5 Assumptions and Dependencies

- Assumes users are connected to the internet and using current web browsers
- Charging providers consistently supply reliable and timely data
- Payment services like Stripe and PayPal maintain strong uptime and reliability
- Google Maps API continues to be available for location-based functionality

3. Detailed Requirements

3.1 System Interfaces

- **User Interface:** Responsive web design for all devices.
- **Hardware Compatibility:** Supports mobile devices with GPS functionality.
- **Software Integrations:**
    1. Google Maps API for location services.
    2. Stripe/PayPal API for transactions.
    3. OCPP for station communication.
- **API Specifications:** RESTful API for third-party integrations.

3.2 Functional Specifications

- **User Authentication:** Supports account creation using email or third-party logins via OAuth 2.0 (e.g., Google or Apple)
- **Station Lookup:** Allows users to search and filter stations by location, connector type, and real-time availability
- **Smart Routing:** Suggests optimal travel paths with charging points based on the vehicle's range
- **Secure Payments:** Processes credit card transactions through Stripe, ensuring PCI-DSS compliance
- **Station Administration:** Enables operators to modify pricing, update station status, and manage maintenance timelines
- **User Reviews**: Lets users leave star ratings (1–5) and provide feedback on station performance

3.3 Non-Functional Specifications

- **Performance:**
    - Station search results are delivered in under two seconds
    - Payment transactions are completed in five seconds or less
- **Security:**
    - User information is protected using AES-256 level encryption
    - Passwords are securely stored using Bcrypt hashing
- **Availability:** Maintains an uptime of 99.5% or higher

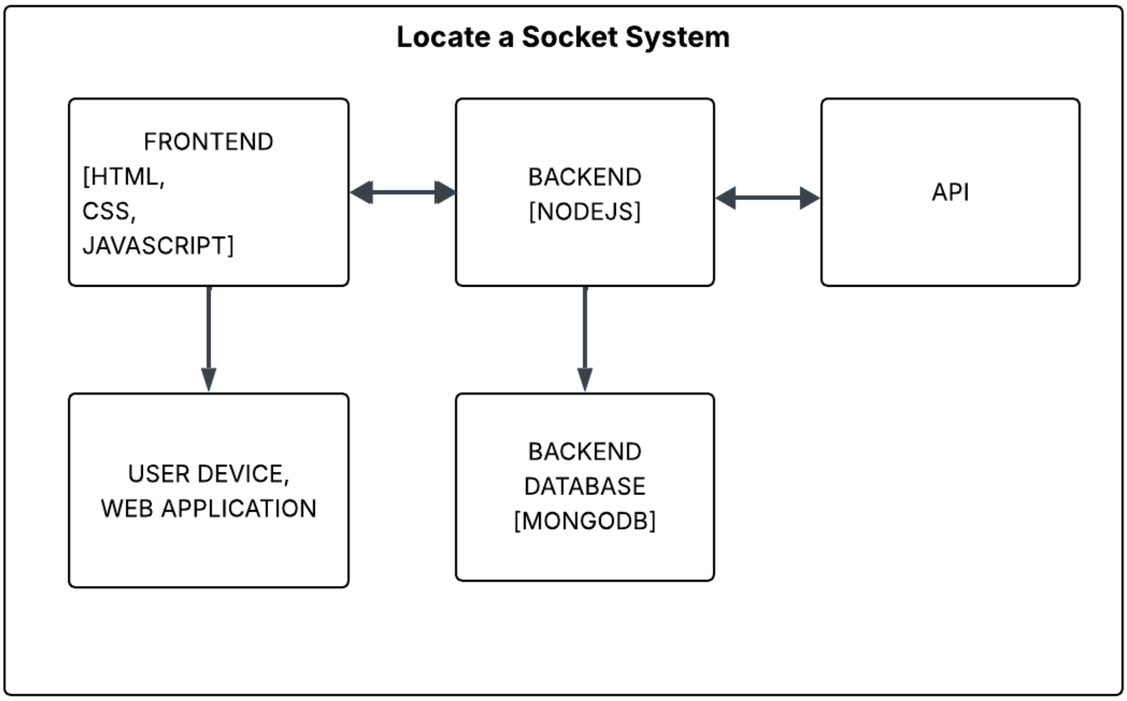
- **Accessibility:**

    - Meets WCAG 2.1 AA standards for web accessibility
    - Core features are reachable within three user interactions

4. Additional Information

4.1 Architecture

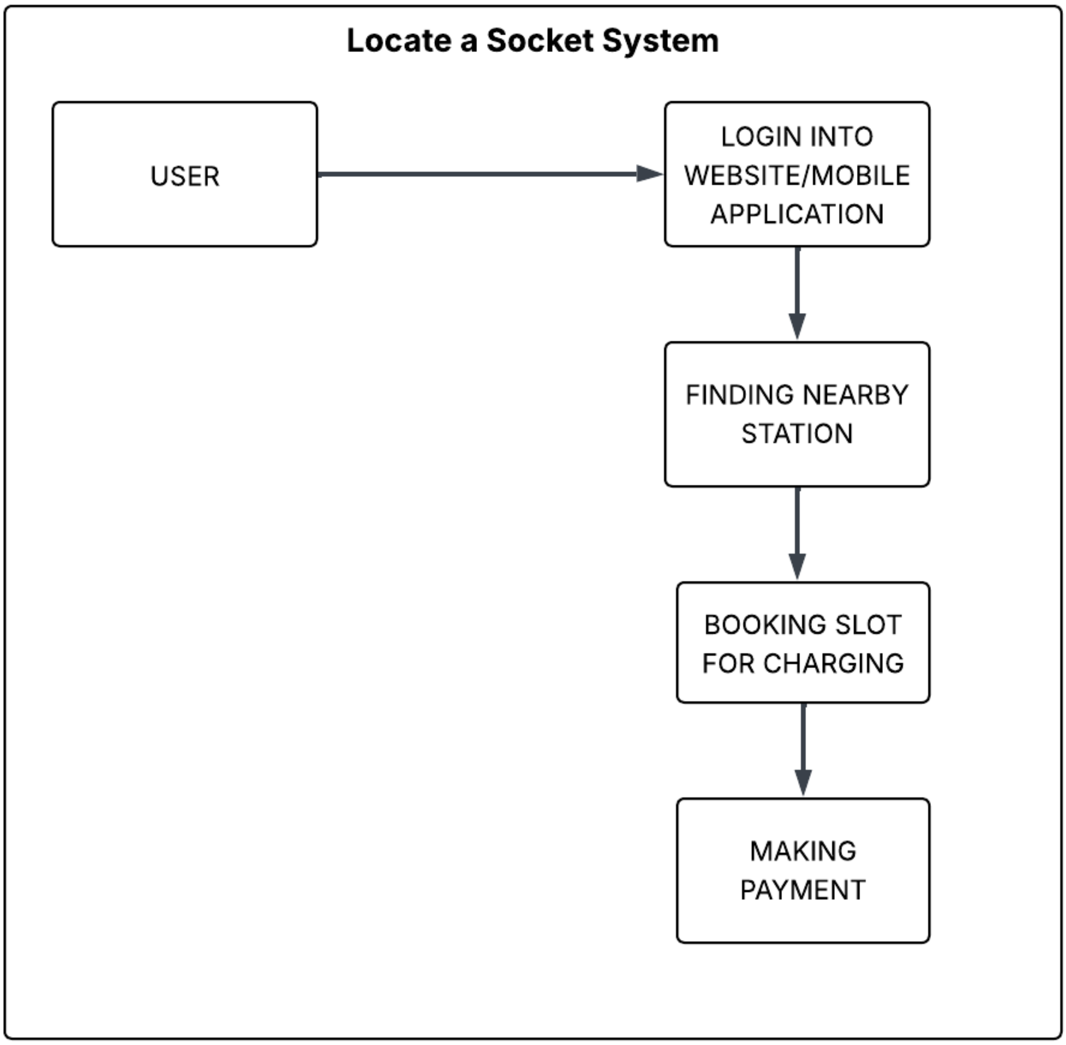
- **High-Level Design:** The system follows a layered structure with a web frontend communicating through an API server, which connects to a database and integrates with third-party APIs.
- **Deployment:** Deployed using Amazon Web Services (AWS), utilizing EC2 for computation, RDS for database management, and S3 for file storage.
- **Workflow Diagrams:** Visual workflows depict how users book charging slots and complete payments.


Locate a Socket System
FRONTEND [HTML, CSS, JAVASCRIPT]
BACKEND [NODEJS]
API
USER DEVICE, WEB APPLICATION
BACKEND DATABASE [MONGODB]

4.2 Use Cases

- **Use Cases**: Users can search for nearby charging stations, book a time slot, and finalize the transaction.

- **Interaction Diagrams**: Visual representations show how users, the system, and third-party services communicate throughout the process.


Locate a Socket System
USER
LOGIN INTO WEBSITE/MOBILE APPLICATION
FINDING NEARBY STATION
BOOKING SLOT FOR CHARGING
MAKING PAYMENT

4.3 API Documentation

- REST API endpoints with sample requests/responses for station search and payments.

4.4 Compliance and Legal

- Secure Payments: Follows PCI-DSS standards to protect all payment transactions
- Data Protection: Complies with GDPR requirements to safeguard the personal data of EU users
- Legal Documentation: Includes clearly defined Terms of Service and a comprehensive Privacy Policy

4.5 References

1. IEEE Computer Society. (1998). *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. IEEE.
2. Google Developers. (2023). Google Maps Platform Documentation. Google LLC.
3. PCI Security Standards Council. (2022). *Payment Card Industry Data Security Standard (PCI DSS) v4.0*.