# *Software Engineering*
# *Software Requirements Specification (SRS) Document*

*Emilio Flores*
*02/26/2023*

# Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|
| 1st Draft | Emilio Flores | All sections being Filled | 2/26/23 |

# 1. Request Proposal

## 1.1 Purpose

The purpose of this document is to define and describe the requirements of the EV Charging Station web-app and to spell out the system's functionality and its constraints.

## 1.2 Overview

The users for the system are owners and operators of electric vehicles of various Makes and Models. I am the sole developer of this application and will be including sources that prove useful to its development.

The product is a web application that can provide users with real-time data regarding the location of EV Charging stations without constraint to whether or not their vehicle is of a particular brand. The app will offer additional functionality for users who wish to create an account such as availability at the station, booking of charging time, view charging rates, which makes/models are eligible for charging at the charging station, favoriting of their preferred stations and view and offer reviews of the stations they search/visit.

## 1.3 Requirements

### Functional Requirements

- User Registration & Login: User should be able to create an account to access features and login in if they have already created an account.
- Station search: Users should be able to search for Charging Stations on a map or by a list and filter the results by closest distance, lowest price and highest rating.
- List Station Details: The app should provide information as to which brands of electric vehicles can be charged at the station.
- Favorite Location: Give users the ability to have a list of favorite stations if they have an account for the app.
- Availability: Check availability of chargers in real time at each station
- Reserve Times: App should provide users the ability to save their spot at a pre-designated time based on availability.
- Station Ratings: Users should be able to provide a rating a provide feedback of the station the wish.
- Charging Reminders: If they wish to set a reminder to charge their vehicle, users should have the ability to set a reminder that will sent an automated email/message to themselves.

### Non-Functional Requirements

- Performance: The system should be capable of handling a large volume of concurrent users.
- Security: The system must be capable of securely storing users' email and personal information.
- Usability: The system is expected to be user friendly and intuitive to use with navigation being easy to understand.
- Reliability: The system requires a 24/7 up-time and must be accessible to both mobile and stationary users. Consider having a back-up of all databases/source code should the system encounter a fault.

## 1.4 Value

The charging station locator is being developed with the intention of simplifying electric vehicle owners' ability to locate and reserve time-slots for charging stations in densely populated areas and for areas where charging stations for public use are not easily located. The best ways to generate revenue for such a service at this time are to have ad partners. In the future, if the

service is capable of providing premium features that bring proprietary or innovative features to users which require increased funding to implement and maintain, there could be the potential for a subscription-based model. Advertising could be generated via generic and targeted advertising with Google Ads as well as advertising partners, particularly those who feature vehicle accessories.

# 2. Feasibility Analysis

## 2.1 Technical Feasibility

Projected technologies necessary to build the proposed application are a database server, web server and a web application capable of calling (1) or more APIs to retrieve information related to EV charging stations on a national scale.

Compatibility issues may arise in stages where incorporation of multiple charging network providers begins. The existing services that allow users to reserve times on certain networks may not be accessible to users outside of the ecosystem developed by the specific charging provider.

Though there is clear risk involved, the project can prove to be feasible.

- There are familiar users features (search, filter, profile creation/management)
- The technology to be used has examples of working models in nearly every web-based application in production today with similar sized or larger user-base scale (Grub-Hub, DoorDash, Uber, GasBuddy, Gas Guru, etc.)
- Project team is estimated to be of a size of 1-3 people.
- System is being built from the ground up, there is no pre-existing design structure to use as a basis.
- Charging station data is readily available and accessible.

## 2.2 Economic Feasibility

| **Development Costs** | **Operational Costs** |
|---|---|
| • API use fees<br>• Training costs | • AWS Hosting<br>• API calls (fee per call) |
| **Tangible Benefits** | **Intangible Benefits** |
| • Increase workflow/productivity<br>• Save time<br>• Competitive Pricing | • Brand recognition<br>• Co-operative integration with other networks<br>• Encourages station condition responsibility |

## 2.3 Organizational Feasibility

Strategic Alignment: Provide a quality service first, the support will grow when quality is emphasized; this is the strategy under consideration. With this strategy, the application must perform each of its intended requirements without hesitation or fault and do so in a timely manner.

# 3. Risk Assessment

## 3.1 Risks, Likelihood and Mitigation Tactics

- Data quality risk: The accuracy of the availability, pricing, and location information are crucial to the operation and usefulness of the system.
    - Likelihood: Medium
    - Impact*:* If the data is inaccurate or incomplete, users may not be able to locate stations, have up-to date booking/availability information
    - Mitigation: Data quality control

- Performance risk: The system may experience performance issues, such as slow response time, unavailability, or system crashes.
    - Likelihood: Medium
    - Impact: Poor user experience
    - Mitigation: Performance Testing
- Security risk: The system may be vulnerable to security threats (hacking, data breaches, and unauthorized access)
    - Likelihood: High
    - Impact*:* Compromised security and privacy
    - Mitigation: Database encryption, limit on login attempts
- Resource risk: The system will require significant to remain functional as the number of users increases.
    - Likelihood: Medium-High
    - *Impact:* System will not function effectively and efficiently or scale with ease
    - Mitigation: Carefully plan and manage resources
- Delay Risk: There is a risk of delay if problems are encountered during development.
    - Likelihood: High
    - Impact: Increase in the time to complete the project
    - Mitigation: Utilize schedule and track each phase. Factor-in delay time into schedule and drop functional requirements if necessary and possible.

# 4. Project Work Plan & Gantt Chart

The project work plan will provide an estimated timeline of completion for each task of the project. The actual time required to complete each stage will depend on availability of resources and unforeseen complications throughout the development stages of the project. The Gantt chart will be a visual representation of the work plan outlined below.

## 4.1 Project Work Plan

1. Requirements Gathering and Analysis (2 weeks)
    - Define scope
    - Collect and analyze requirements
    - Identify stakeholder requirements
    - Outline use cases

2. System Design (3 weeks)
    - Develop system architecture
    - Develop necessary documentation
    - Plan database layout

3. Back-end Development (5 weeks)
    - Implement the database
    - Develop the recipe recommendation algorithms
    - Implement user feedback and personalization

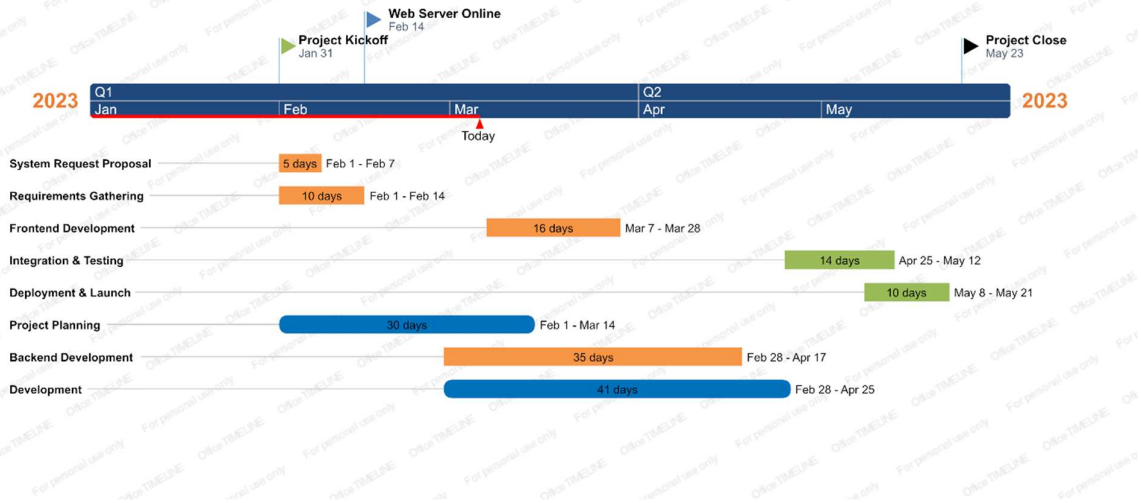
4. Front-end Development (3 weeks)
    - Design and develop design
    - Develop front end UX to ensure maximum usability.

5. Integration and Testing (2 weeks)
    - Integrate the front-end and back-end systems
    - Perform unit and integration testing
    - Identify and correct errors and bugs

6. Deployment and Launch (2 weeks)
    - Deploy the system to cloud server
    - Configure the system for use
    - Perform testing
    - Launch system

7. Maintenance and Support (Ongoing)
    - Provide maintenance and support
    - Implement new features and functionality as needed
    - Monitor system security and performance

## 4.2 Gantt Chart



# 5. Use Cases

.

## 5.1 Textual Use Case #1

| | |
|---|---|
| **Title** | Station Search |
| **Actor** | User |
| **Description** | A search can be done for charging stations using ZIP Code, City and/or State and can be sorted based on a specified radius. |
| **Preconditions** | o User has provided one of 3 available search parameters<br>o The system has generated a list of charging stations in the location specified in the search |
| **Normal Course** | o User visits the URL<br>o User enters a location in the search bar & specifies a proximity range<br>o The system query using an API call and returns results in a list<br>o The user selects a location<br>o The system expands the list item to reveal location details & pricing |
| **Alternative Course** | o User specifies an entire city rather than a specific ZIP code |
| **Exceptions** | o If the system experiences technical issues or is unavailable, user will be unable to access the system<br>o If the user provides inaccurate or limited information about their location, the platform will be unable to generate results |

## 5.2 Textual Use Case #2

| | |
|---|---|
| **Title** | Add to Favorites |
| **Actor** | User |
| **Description** | Allows a registered user to have list of favorite stations and have the ability to add stations to their Favorites List |
| **Preconditions** | o The user has a registered account with the system<br>o They have logged in using some method of authentication<br>o User has entered a valid search parameter |

| | |
|---|---|
| **Normal Course** | ○ The user logs into their account<br>○ User searches locations based on the criteria they must specify in the search<br>○ Locations have a button to click that will add the list to their Favorites list |
| **Alternative Course** | ○ User can add from featured list of locations rather than through a search |
| **Exceptions** | ○ If the system experiences technical issues or is unavailable, user will be unable to access the system<br>○ If the user provides inaccurate or limited information in the search, no options to pick from will be available<br>○ If the user has already added a location to their list, it cannot be added twice |

## 5.3 Use Case Diagram