



INSTITUTE FOR ADVANCED  
COMPUTING AND  
SOFTWARE DEVELOPMENT  
AKURDI, PUNE

**System Requirement Specification**

**Document**

**“Slot Booking for Charging an EV at Charging  
Station”**

**PG-DAC SEPT 2021**

*Submitted By:*

**GROUP NO 54**

More Sangram Rajkumar - 210941220108

Joshi Sushant Prakashrao - 210941220080

**Mrs. Shilpa Pawale**  
Project Guide

**Mr. Kashinath Patil**  
Project Guide

## **Purpose of Document**

The purpose of this document is to enlist the software requirements and specifications for the '**Slot Booking for Charging an EV at Charging Station**' which will be a new added feature to current charging stations India.

## **Intended Audience**

This document is intended for developers, users, testers and project managers for the purpose of understanding the design of systems in terms of different perspectives. Further, this document contains functionalities and characteristics of the system along with the working environment. It also includes other information related to systems such as external interface requirements, features and other non - functional requirements.

## **Project Overview**

The Slot Booking for an EV is a facility is used by Indian citizens who owns Electrical Vehicle, to find out the nearest charging station and book charging slots in advance. Currently if people want to charge their vehicles at charging stations, they have to wait till charging slots to get empty and the time required may vary from type of e-vehicles. It is estimated that the total potential demand for the full range of electric vehicles in India (mild hybrids to full electric vehicles) will be in the range of 5–7 million units in new vehicle sales by 2020.

As number of e-vehicles are increasing day by day. Crowd at charging stations is required to manage, because e-vehicle takes time to charge and if crowd is present at charging stations, then it will take more time to get a slot for charging. So, with the help of our project, we trying to minimize the waiting time required at charging stations to get slot by booking slot in advance and also helping user to find out the nearest charging station and we are also providing facility for premium customer not to wait for charging the battery so customer can directly exchange their uncharged battery with already charged battery at charging stations.

## **Assumptions and Dependencies**

### **Assumptions**

1. There is an active internet connection with the system.
2. The system has an internet browser installed.
3. Users know the English language.

### **Dependencies**

Users should remember patterns for authentication while confirming his or her authenticity in the whole authentication process.

## **Purpose and Scope**

Swift Charge is a platform where we are giving facilities to users like finding nearest charging stations, in advance charging slot booking and for premium customers directly battery exchange to exclude the charging time means premium customer don't need to wait for charging they can simply exchange their battery with charging stations and can continue their journey. The purpose of this project is to minimize the waiting time of customers, manage the crowd at charging stations, increase the productivity of charging stations, adding the new charging stations in the current network of charging stations.

There is a wide scope for Ev's in four-wheeler market, auto rikshaws, goods vehicle, bus as well as two wheelers. All of this market is waiting for a change. We can integrate this web app with restaurants and malls which have charging points in their parking, we can also make the mobile application for this Swift Charge. In future demand for e-vehicles will increase more rapidly so need to manage this crowd of e-vehicles at charging stations this can be achieved by increasing number of charging stations.

### **Product Functions**

- The user will be able to find the nearest charging stations.
- User can able to book charging slot in advance.
- Users can be able to exchange existing battery at their end with the already charged battery at charging stations this will also helps to minimize crowd and customer also don't need to wait till battery get charged.
- We are also providing facility to add the valid charging station in current network of charging stations so this will also minimize crowd at charging stations in future.
- Analysis of charging stations such as for how much time charging station is productive and for how much time charging station is idle.

### **User Classes and Characteristics**

#### **1. Admin Class:**

- o The user class contains attributes pertaining to users such as Role, Name, email, password, Phone number etc.

#### **2. Station Class:**

- o The Station class contains the details about the stations like Name, Station registration number, Station location, Number of slots station contains.

#### **3. Customer Class:**

- o The Customer class contains the details about the customer such as Customer Id, Vehicle type, vehicle number, etc.

MVC design patterns will be followed with DAO and Service layers for the user module.

## Operating Environment

### ●Hardware Platform:

o The hardware infrastructure requirements for Swift Charge will be addressed in the Infrastructure and Deployment Architecture document.

### ●Software Platform:

- o Front-end: - React Bootstrap.
- o Back-end: - Spring boot, MySQL.

### ●Supported Tools:

- o Spring Tool Suite, Vs Code.

## Design and Implementation Constraints

- o User interface is only in English. No other language option is available.
- o Users can log-in only with his assigned user-name and password based on Roll.
- o Limited to HTTP.

## User Documentation

User documentation mainly comprises the *resource menu* on the website which will contain manuals particularly for the this project. It will give all the minute details about the authentication process, if any user has any query about any module or functionality, one can refer to it and see how to proceed for the authentication process. This report is the complete documentation of the authentication process. It gives complete details about the authentication process, its functionality, users, software used, hardware requirement, environment and so on.

## External Interface Requirements

### User Interfaces

The main element is webpages using Reactjs. The pages will use JavaScript for the UI design and validation. The website will be responsive. bootstrap will be used for the CSS and styling of the webpages. The webpages will be rendered on the server-side using Servlets.

### Hardware Interfaces

The user-end system can be either a smartphone or a computer device. The application supports all major web browsers. The web browser should be JavaScript enabled.

### Software Interfaces

In software interfaces, Spring and Hibernate are the back-end technology used along with MySQL Database. The front-end technologies include Reactjs, CSS, Bootstrap, JavaScript and. Data will be communicated between these interfaces accordingly.

### **Communications Interfaces**

The main communication interface for interacting with the System will be the web Browser using HTTP request.

### **System Features**

- User Registration.
- User Login.
- Forgot pattern.
- Role base login.
- Slot booking.
- Register Charging Station.

### **Other Non-functional Requirements**

#### **Performance Requirements**

The system should store all the database records of each user properly and the application should be available for use 24\*7 through the server. The system should authenticate and register users properly in a secured manner. Also, the application should be user friendly with a proper user interface which makes it easy for the user to understand. All the options should be present in properly accessible places for user convenience.

#### **Safety Requirements**

All login Ids and passwords of the users should be protected for privacy using whatever constraints required in the database or the application. User records are to be backed up securely across database servers. In case database is hacked by someone and data is deleted a backup server should be present for such purpose.

#### **Security Requirements**

All passwords of the users should be protected for privacy using whatever constraints required in the database or the application. The database should be protected from attacks and unauthorized access. The interface should be protected from attacks. All passwords should be encrypted and stored.

#### **Software Quality Attributes**

##### **Availability**

- The system should run on a variety of operating systems that support the JavaScript language. The system should run on a variety of hardware.

### Accessibility

- The software will be accessible to admin, Customers and Station owner.

### Compatibility

- The software will be compatible with multiple platforms.

### Durability

- The software will be tested for working with multiple users.

### Effectiveness

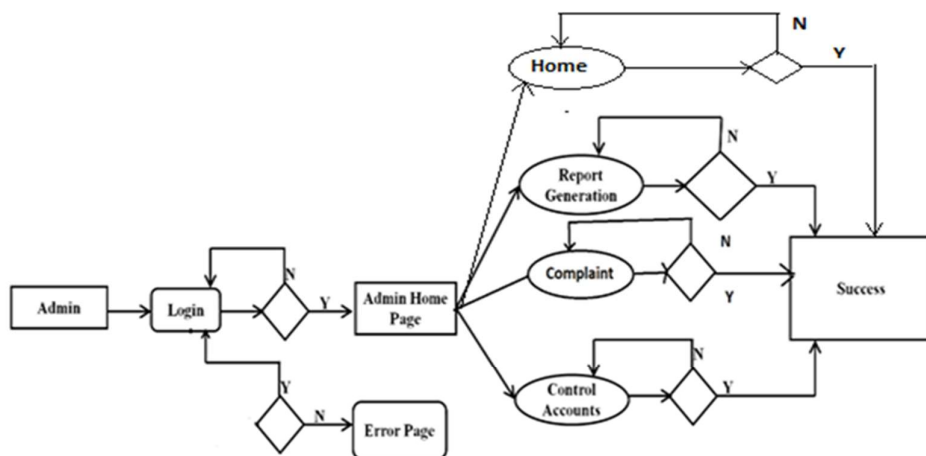
- The software will be made to handle operations effectively.

### Maintainability

The system should be easy to maintain. There should be a clear separation between the interface and the business logic code. There should be a clear separation between the data access objects that map the database and the business logic code.

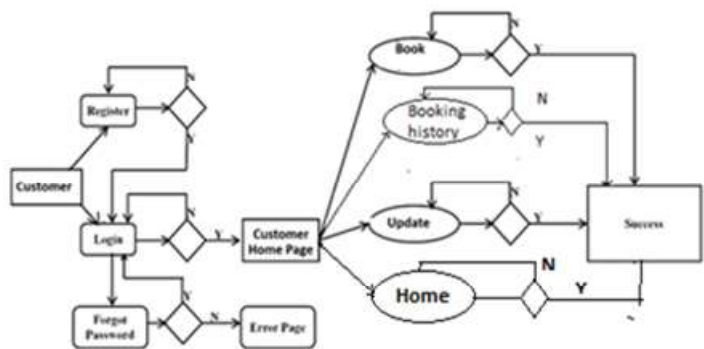
## Use case Diagram:

**Admin :**



*Fig. Page Navigation For Admin*

Customer :



Distributor :

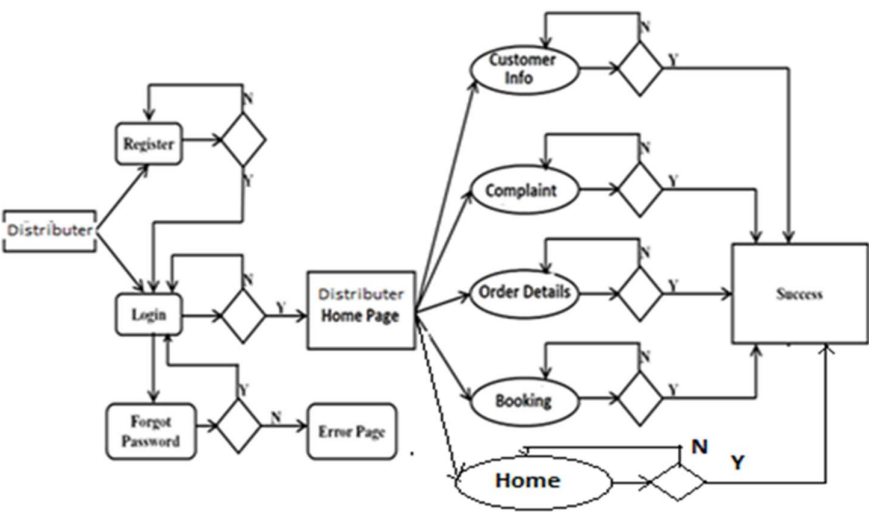


Fig. Page Navigation for Distributer