

---

# SOFTWARE REQUIREMENTS SPECIFICATION

for

## E-VEHICLE TRACKER

Prepared by :

1. Aditya Deshmukh (200001019)
2. Pranjali Gavhale (200001024)
3. Hritika (200001029)
4. Mitika Bhadada (200001046)
5. Vansh Kathnawal (200001078)

Submitted to : Dr. Puneet Gupta

February 17, 2022

# Contents

<b>1</b>	<b>Instruction</b>	<b>3</b>
1.1	Purpose . . . . .	3
1.2	Intended Audience and Reading Suggestions . . . . .	3
1.3	Project Scope . . . . .	3
<b>2</b>	<b>Overall Description</b>	<b>4</b>
2.1	Product Perspective . . . . .	4
2.2	Product Features . . . . .	4
2.3	Development Strategy . . . . .	4
2.4	Development Environment . . . . .	5
<b>3</b>	<b>System Features</b>	<b>6</b>
3.1	Driver Module . . . . .	6
3.2	User Module . . . . .	7
3.3	Additional Feature of this Project . . . . .	7
<b>4</b>	<b>External Interface Requirements</b>	<b>8</b>
4.1	User Interfaces . . . . .	8
4.2	Hardware Interfaces . . . . .	8
4.3	Software Interfaces . . . . .	8
<b>5</b>	<b>Non-Functional Requirements</b>	<b>9</b>
5.1	Performance Requirements . . . . .	9
5.2	Safety Requirements . . . . .	9
5.3	Security Requirements . . . . .	9
5.4	Software Quality Attributes . . . . .	10

# **1 Instruction**

## **1.1 Purpose**

The purpose of this document is to build an online system to track and check the availability of the E-vehicles running on campus.

## **1.2 Intended Audience and Reading Suggestions**

This project is a prototype for tracking vehicles and it is restricted within college premises. This has been implemented under the guidance of college professors. This project is useful for the students to track the E-vehicle.

## **1.3 Project Scope**

The purpose of this project is to track the current location of e-vehicles running on campus. The software attempts to cut passenger wait times while also providing a convenient and easy-to-use application. For E-Vehicle drivers, there is a dedicated window. The location of the E-Vehicle will be continuously updated in a real-time database, which will be visualized using Google Maps at the user's end.

## 2 Overall Description

### 2.1 Product Perspective

A smartphone application for both passengers and drivers is included in this system.

The database for the system stores the following information:-

**E-vehicle details:-** It stores the running time and break-time details of each e-vehicle.

**Location:-** This application will show the location of each vehicle running currently on campus.

**Driver's details:-** It contains the details of each driver like his name, his phone number, his photo, etc.

### 2.2 Product Features

The technology allows users to access the real-time position of E-Vehicles on campus. The user can see the current position of all operational e-vehicles as well as their drivers' details. The driver must give his current location as well as his intended destination. It will be used to determine the quantity and location of available E-Vehicles. The user can see the E-current Vehicle's location as well as its destination, reducing their waiting time. In the event of an emergency, the user can immediately urgently request the nearest active E-Vehicle Driver.

### 2.3 Development Strategy

The software model used for this project is "V-Model".

It involves execution in a sequential manner in V-shape. We will make a step by step approach for various processes and will also do regular testing at the end of each process. We will first understand the requirements of this project from the user's perspective and then the Validation process begins with the acceptance test. Each step of the application incorporates some of the functionality that is needed by the user.

As per the V-model, some of the development steps would be :

- Developing the driver module and plotting the location of vehicles for the student.
- Enabling the driver to activate the E-vehicle.

## 2.4 Development Environment

- **App Development Framework:-** Flutter
- **Firebase:-**
  - **Database:-** Cloud Firestore is used to store all of the data. Cloud Firestore is a global-scale NoSQL document database that allows you to effortlessly store, sync, and query data for your mobile and web apps. There are no tables or records in this database, unlike a SQL database. Data is kept as documents within a collection in a cloud fire-store. Furthermore, hierarchies can be built to store related data and expressive queries can be used to quickly retrieve the data required.
  - **Authentication:-** To authenticate users to your project, Firebase Authentication delivers backend services, an easy-to-use SDK, and ready-to-use UI libraries.
- **Geo-Location Services:-** Google map cloud function

## 3 System Features

- **Description and Priority:**

The car tracking system keeps track of active vehicles in real-time by dynamically updating the active vehicle's geolocation on the Firebase database and displaying the updated location to the user in real-time.

This android application includes two modules:

- Driver module
- User module

### 3.1 Driver Module

- **Driver Login:-** Admins can access the program using email and password authentication provided by the admin.
- **Vehicle Tracking:-** System will track the location of the vehicle using the flutter's location plugin. The location plugin subscribes to a listener at the driver's end so that whenever the location of the vehicle changes, it listens to that change and the new location is dynamically updated to the database. Also, this new location of the vehicle is updated on the user's screen in real time.
- **Registration:-** Any user can register as a driver, but only those who have been approved by the Admin/Supervisor are given access to the program. To access his account, the approved driver receives a user email and password.
- **View Active Vehicles:-** The driver can also see the location of other active vehicles, which is constantly updated in the database as soon as the vehicle's location changes, and is likewise updated in real-time on the driver's screen. When a driver taps on any of the active car's markers, he receives information about the driver who is now operating that vehicle, as well as the ability to use the chat service to coordinate his movement with other drivers.

- **Add-On Feature:-** The driver is given an interface to toggle the vehicle's status, whether it is active or inactive. For successfully designating the car as active, the driver is prompted with a dialogue box in which he must select the location to which he is travelling as well as the vehicle number of the current vehicle he is driving. In addition, the driver/admin receives an interface that dynamically updates the number of seats in the vehicle that are unoccupied, and the total number of seats in the vehicle is displayed to users to improve the user experience.

## 3.2 User Module

- **View active vehicles:-** The driver is given an interface to toggle the vehicle's status, whether it is active or inactive. For successfully designating the car as active, the driver is prompted with a dialogue box in which he must select the location to which he is traveling as well as the vehicle number of the current vehicle he is driving. In addition, the driver/admin receives an interface that dynamically updates the number of seats in the vehicle that are unoccupied, and the total number of seats in the vehicle is displayed to users to improve the user experience.
- **SOS/ Emergency:-** In the event of an emergency, the user will be able to see which vehicle is closest to them and phone the driver for immediate assistance.
- **Add-On Features:-** The user gets vehicle and driver information by tapping on any of the active vehicles. By pressing the current location button in the center, the user can relocate the map camera to his current location.

## 3.3 Additional Feature of this Project

- Flutter geolocator plugin for real-time car tracking and visualization through Google maps.
- Information such as vehicle status and location is displayed graphically.
- Implementation of Global Positioning System (GPS)
- Easy access and personal use thanks to a user-friendly UI.

## 4 External Interface Requirements

### 4.1 User Interfaces

- **Software:-** Flutter (We are using flutter for both frontend and backend of this project)

### 4.2 Hardware Interfaces

- Android
- IOS

### 4.3 Software Interfaces

Following are the software used for the E-vehicle tracking application

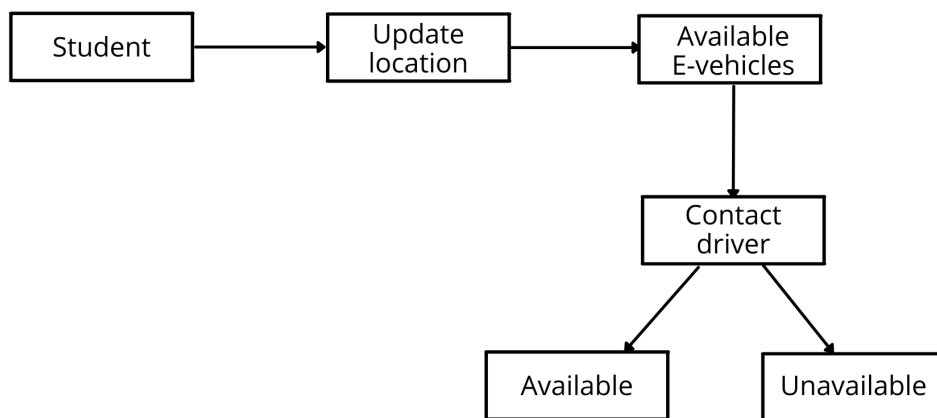
Software Used	Description
Operating System	We have chosen Android and iPhone operating system for its best support and user-friendliness.
Database	To save the student's location and driver's location we have chosen cloud firestore.
Dart	To implement the project we have chosen Dart language.



## 5 Non-Functional Requirements

### 5.1 Performance Requirements

The steps involved in the working of our application are depicted in the diagram shown below :



### 5.2 Safety Requirements

If a catastrophic failure causes significant damage to a large portion of the database, the recovery method restores a previous copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed-up log, up to the time of failure.

### 5.3 Security Requirements

Like many other applications, security systems require database storage. Vendors must, however, carefully select their database partner due to the unique requirements of the security sector.

## 5.4 Software Quality Attributes

- **Portability:-** The app will run on a variety of data processing platforms without translating it to a foreign language and with minimal or no changes.
- **Maintainability:-** User's evolving needs should be able to evolve the app. It refers to the ease with which a system can be changed.
- **Adaptability:-** The app would be able to adapt to changing requirements to a reasonable extent.
- **Secure:-** It won't disclose the details of any user that he/she enters during registration.