DE-40 (DC&SE)

NUST COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

EVnav+

(Electric Vehicle Smart Navigation and Charging Station Deployment)

PROJECT REPORT

DE-40 (DC&SE)

Submitted by

NS Ali Hassan Abid

NS Muhammad Moiz Alvi

NS Syed Muhammad Ali

NS Waqar Ali Ahsan

BACHELORS

IN

COMPUTER ENGINEERING

YEAR

2022

PROJECT SUPERVISOR

DR. Wasi Haider Butt

YEAR 2022

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

PESHAWAR ROAD, RAWALPINDI

EVnav+

(Electric Vehicle Smart Navigation and Charging Station Deployment)

PROJECT REPORT

DE-40 (DC&SE)

Submitted by

NS Ali Hassan Abid

NS Muhammad Moiz Alvi

NS Syed Muhammad Ali

NS Waqar Ali Ahsan

BACHELORS

IN

COMPUTER ENGINEERING

YEAR

2022

PROJECT SUPERVISOR

DR. Wasi Haider Butt

COLLEGE OF

ELECTRICAL AND MECHANICAL ENGINEERING

PESHAWAR ROAD, RAWALPINDI

DECLARATION

We thus certify that no part of the work referred to in this Project Thesis has been submitted in support of an application for any other university's degree or certificate. We are totally liable for any disciplinary action taken against us if any act of plagiarism is discovered, depending on the degree of the confirmed violation.

ACKNOWLEDGMENTS

We are truly grateful to Allah the Almighty, the Most Merciful, for bestowing upon us His blessings, courage, and wisdom, because of which we have been able to achieve such a lofty goal. In the face of adversity, we are grateful to our parents and family for their unfailing support and encouragement. We appreciate **Dr. Wasi Haider**, for his continual guidance and time spent providing us with support and encouragement at every stage. Without his efforts, we would not have been able to accomplish so much. We also want to express our gratitude to all our colleagues and friends who assisted us both intellectually and emotionally.

Thank you to everyone who assisted; without everyone's support and admiration, things could have turned out differently. We hope that this initiative will add value to people's lives and that it will be developed so that this technology can be used more widely.

ABSTRACT

With the increase of climate change and depleting fossil fuel reserves, implementation of electric technology in the automotive industry is gaining traction. As a result, countries all over the world are working towards building efficient charging infrastructure and auxiliary equipment to assist this step towards complete removal of internal combustion engines from our roads. With the development of electric vehicles (EVs), optimization of charging station facilities is a growing concern. The major issue/drawback of EVs is the amount of time it takes for charging and there is a huge research and market gap in this domain in Pakistan. Our vision is to help reduce queues and delays at charging stations by optimizing the selection of station for each user. This paper outlines the development of a complex software that works towards improving the charging time and making the navigation towards a charging station more efficient for the consumer. Our app decides which charging station is optimal based on availability of station, charger type of EV, battery level of EV, range of the vehicle in the said battery level, and the distance and time it takes from the user’s current location to the charging station. Also, our project helps predict the optimal location for the deployment of new charging station based on EV's data of a certain area. This is done using three major applications: User App, Admin App, and Super Admin App. We aim to help bring innovation in electric vehicle technology in our country and ease its use for the consumer for a cleaner and greener Pakistan.

Table of contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Transportation is responsible for around 24% of total energy consumption worldwide; with a considerable amount being met by fossil fuel energy production in 2018 [1]. The dependency on hydrocarbon fuels leads to issues in financial burden and stability of energy supply, along with their devastating effects on the environment.

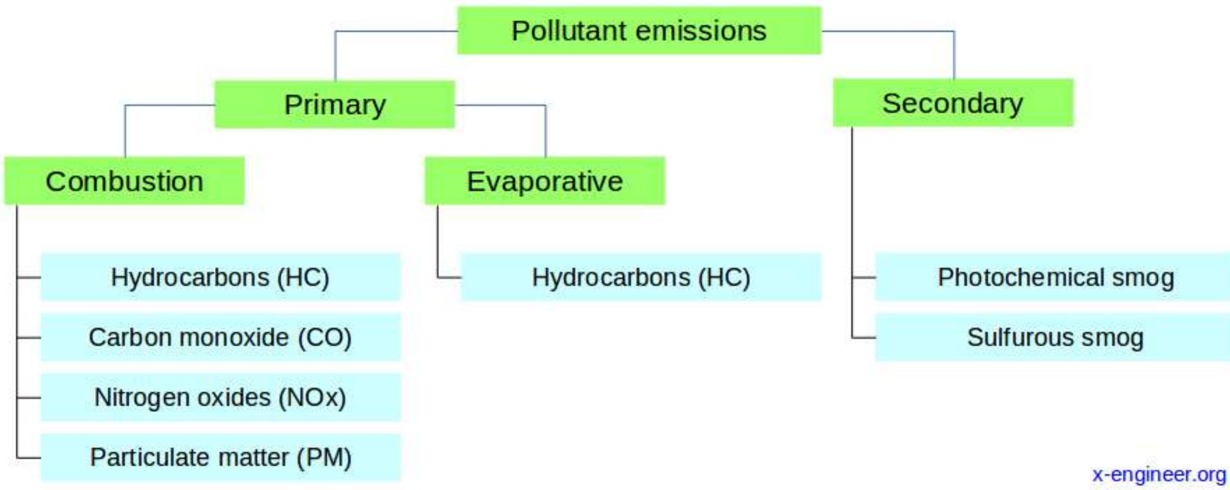

Figure 1: Effects of vehicle pollution on human health

Also, with the depletion in global reserves of fossil fuels, production of underwhelming unconventional resource productions i.e. shale oil, and the ever-increasing environmental impact has forced breakthroughs to change the use of energy, vehicles, tools, and infrastructures that utilize non-renewable resources. This led to the development of electric vehicles (EVs) as the future of transportation due to their energy efficiency and eco-friendliness [2]. The concept of EVs has seen a rapid growth over the last decade. In 2018, there was a 63% increase in electric passenger car stocks compared to the previous year, reaching around 5 million [3].

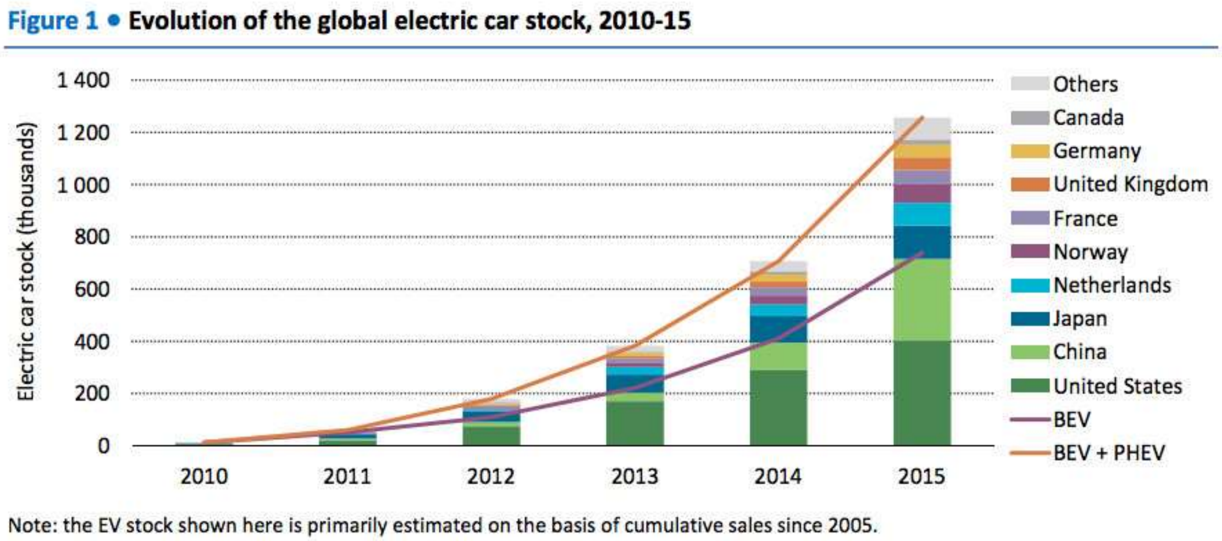

Figure 2: Rapid growth of electric cars worldwide over the last decade.

With the fruitful deployment of EVs over the next few years, introduction of globally agreed standards, associated universal peripherals, universal charging hardware infrastructure and most importantly, user-friendly software on both private and public property is required. This efficient infrastructure is required due to the increase in electric load on the grid with the growing number of electric vehicles. Over the next decade, the quantity of charging stations, along with their sites with become vital to meet customer demand regarding electric supply. Many countries have already begun to establish these infrastructures, for example the British government has announced to prohibit the sale of all conventional petroleum and diesel vehicles by 2040 [4].

Figure 3: A typical electric vehicle charging station.

In contrast, some countries have made limited decisions regarding the implementation of EVs and related rules and regulations. Pakistan, with a limited utilization of electric vehicles and localized infrastructure i.e. charging stations, is included in this list of countries. There are approximately 85 charging station's locations identified across M-1, M-2, M-3, M-4, M-5, M-9, and N-5. Out of these, 15 have been identified as prioritized regions for the development of charging infrastructure [5]. According to the Paris Climate Agreement, Pakistan has pledged to reduce the green-house gas emissions by 20% by 2030.

Figure 4: Current charging stations in main areas of Pakistan.

Due to the currently limited EV footprint in the country, charging infrastructure at these locations would serve the consumer requirements. But with the rapid increase in this technology globally, along with international pressure to reduce green-house gas emissions, EVs will gain a larger share in the transportation sector very soon. Also, with the approval of the National Electric Vehicle Policy (NEVP) in 2019, EV adoption has been fast-tracked and incentivized both users and manufactures to expand the market. Following the reports produced by the National Highway Authority (NHA), National Highways and Motorway Police (NH&MP), and the National Transport Research Centre (NTRC), data has been collected in terms of current EVs and their expected growth over the next few years in the country. The data highlights the need for efficient infrastructures and software to fulfill the needs of the consumer.

1.2 Problem Statement

With the rapid increase in EV technology worldwide, it is crucial to develop user-friendly software to meet the merging of standards and charging technology of EVs. A major drawback in EVs is the amount of time it takes to charge the vehicle. Although there is endless data available for choosing the suitable location for the installation of charging infrastructure regarding economic, social, and technical variables, there is a research and market gap in development of a software to optimize charging of the vehicle and to make it easier for the consumer to locate and choose the appropriate charging station to fulfill their requirements. The time for charging an EV can range from 30 minutes to over 12 hours. Furthermore, the size of the battery and speed of charging point are major factors that affect this time duration. A typical EV with a 60kWh battery takes around 8 hours to full charge with a 7kW charging point. This leads to long queues and delays. Hence, it is difficult for an EV user to locate the appropriate charging station that has the required charging port type, and whether they will be able to charge their vehicle in time as required. Currently, there is no application that solves this drawback and provides the user with sufficient information to ease their use of EVs. The presence of this problem limits the implementation of EV technology in Pakistan.

1.3 Objectives

Our vision is to reduce these long queues and delays at charging stations by optimizing the selection of station for each use. Our software decides which charging station is optimal based on the availability of station, battery level of the EV, duration of charge, and the type of charger to be used for charging the vehicle. Our project also predicts the optimal location for the deployment of new charging stations based on the constant accumulation of data of a certain area. This is done by the development of three major applications, i.e. a user app, admin app, and a super admin app. The user apps help direct the consumer to an optimal charging station based on the above-mentioned factors, while the admin app generates new charging stations based on areas with maximum number of requests, and the super admin panel is responsible for maintaining the overall analytics of users, EVs, and charging stations.

Chapter 2: Electric Vehicle Standards

2.1 Scope in Pakistan

With the launch of the Auto Industry Development and Export Policy (AIDEP) 2021-2026, the implementation of EV technology is accelerated [6]. The government also announced to reduce the sales tax to 1%, incentivizing local production. The main drawback in expanding this technology in Pakistan is due the difficulty to popularize a foreign concept amongst the masses and to inspire local auto manufacturers to produce great amounts. Although these tax breaks seem encouraging enough for foreign and local auto makers to participate in the EV sector, the resistance to change and fear of the unknown makes this transition not only from combustion engines to motors but is also the transition of mindsets. Hence, our project aims to make this transition not only easier, but quicker.

Furthermore, with the unveiling of the China-Pakistan Economic Corridor (CPEC), foreign involvements of auto companies were encouraged to enter the Pakistani market, especially EV manufacturers [7]. These include a wide range of prices that target diverse income groups of the country. These foreign investors include China, South Korea, and Japan, who believe that Pakistan has a huge potential for this market. Karakoram Motors is a local manufacturer that is the authorized manufacturer and assembler of the Dynasty IT electric cars [8].

Figure 5: Dynasty IT EV manufactured by Karakoram Motors.

Moreover, Rahmat Group has acquired 25 acres of land for the establishment of an electrical complex in Nooriabad for bringing electric cars to Pakistan according to Dawn News [9]. This includes electric buses, two-wheelers, vans, and trucks. Nissan, Audi, Hyundai, and Renault are already in communication with the Ministry of Industries and Production Pakistan to produce locally manufactured EVs [10]. Audi launched its e-Tron 50 Quattro in 2020 as well and is commonly seen in the roads of the posh areas. Also, BMW had launched its top-of-the-line EVs, i3 and i8 which gained huge popularity in Pakistan. With the spread of awareness among local automobile buyers and producers about the advantages of EV technology over conventional combustion engines, and the creation of charging infrastructure, we will witness exponential growth of EV technology in Pakistan over the next couple of years.

Figure 6: Introduction of electric vehicles in the public transport in Pakistan.

2.2 Software Development Requirements

The creation and preservation of multifaceted software systems, with their growing requirements and wide range of design patterns is a difficult task. To sketch a software architecture for an EV charging station finder and management platform, a succession of design patterns was studied to generate a compact base assembly for the Application Programming Interface (API). Multiple cloud design patterns were tested during the deployment procedure to develop a suitable multi-tenant cloud application. For our application, we identified three main design patterns categories for the development of the software:

1- Ideal solutions used by expert object-oriented designers to generalize the problems faced by software designers during software design patterns development process.

2- Cloud development patterns that help construct a dependable, scalable, and secure application in the cloud.
3- Multitenancy model that ensures an isolated, stable, customizable, and scalable cloud architecture for the multiple tenants of the application.

Figure 7: Software development cycle

With a comprehensive study of many existing and possible project developments and design patterns, we chose the design pattern that offers the best solution for the interaction between the context i.e. charging station navigation, problem i.e. long waiting time, and solution i.e. optimization for reducing the time for charging. We created a stencil that delivers the necessary data to fulfill the core of a suitable assembly and to identify and fix potential flaws that the proposed solution could entail.

With the growth of electric vehicles in the industry and innovation, and with government's financial incentives, the effervescence is constantly advancing. In this growing e-mobility market growth, it is crucial to manage the charging procedure of EVs. The project architecture design has a key role in handling and delivering the qualitative software systems. Michael Hahsler [11] illustrated the use of open-source application to product the principles summarized above. The development process for the charging platform is done by considering the DP role. With the DP influence, the project architecture is built, and a software solution is created based on a compact structure that can be used to easily edit and update with the flow of the project. It also allows us to implement unit tests, that is important in a thriving complex project.

Using this implementation, we used DP to build the services instance, which consequently assured the services dependencies. Hence, ensuring a well-organized distribution of the resources available and to keep the workings decoupled.

2.2.1 Repository Design Pattern

We used the repository design pattern to make an "abstraction layer between the data access and the business layer of the application" [12].

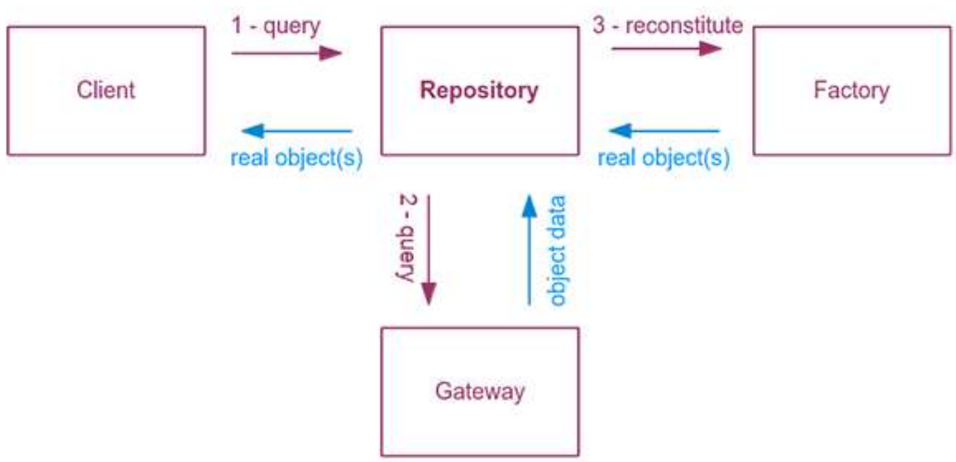

Figure 8: Repository Design Pattern. `

This pattern is suitable for our design because it ensures code reusability and maintainability. Using specific classes called repositories, the data access is maintained the communication is done via a middle ware and established with the database. For our CRUD operation i.e. create, read, update, and delete, this generic repository will allow us to fulfill these functions. These functions can be called upon by any repository class in our project. For the implementation of this pattern, the following scheme shows pattern implementation from services usage to the abstract layer:

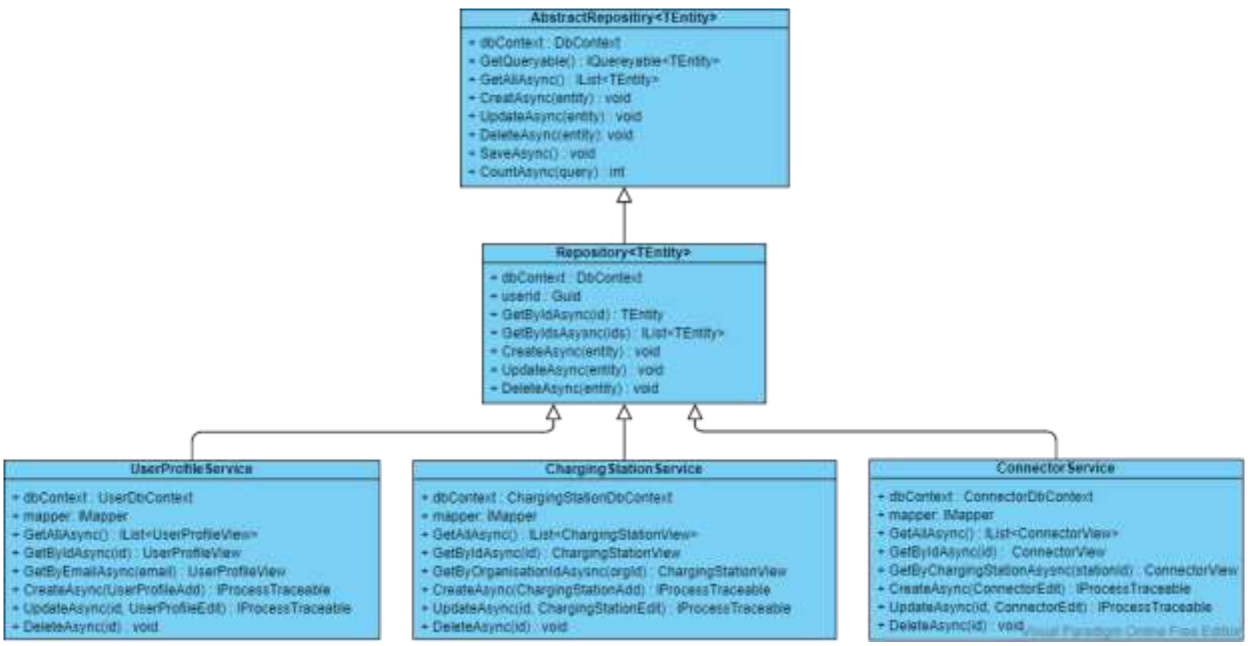

Figure 9: Repository class level diagram

Chapter 3: Software Requirements

Software requirements detail how the needed program with work alongside the hardware and external surfaces. Also, the speed and response time of the system, portable ability of the software across various platforms, and the ability to recover after a crash are also linked with these requirements. Furthermore, every software has its limitations, hence these shortcomings with the security and quality of the software play a key role in requirements as well. As the user or client communicates in plain English, the system analyst works on the documentation of these requirements in the technical language of study. This makes it possible for the software development team to understand the documentation and implement it in the software.

For an effective and details software requirement program, the following features are critical:

1- The requirements of the user are to be in a common language i.e. English
2- A structured language needs to be used to write the technical specifications in the company.
3- Pseudo code needs to be utilized for the description of the design.
4- Mathematical and conditional symbolizations should be used for DFDs and alternative mathematical symbols.

3.1 Functional Requirements

In a software development process, the software requirement specifications include both functional and non-functional requirements. In functional requirements, all features, and functions of a product that the designers need to use to fulfill the duties are called functional requirements. Hence, they play a key role and need to be implemented explicitly for both the stakeholders and team working on the software. It outlines the behavior of a system under certain conditions.

<u>User Registration:</u>

Aim: To add a new user

Input:

- Email
- Name
- Phone Number
- Password

- Password Confirmation (Validation)

Output: User added successfully

Process: Saved to firebase

User authentication:

Aim: Login of the already added user

Input:

- Email
- Password

Output: User logged in successfully/unsuccessfully

Process: Compare email and password from firebase

Add car(s):

Aim: Owner adds a new car into database collection

Input: Add a car, its make, model, range, efficiency

Output: Car successfully added

Process: Car pushed in database

Select car:

Aim: User selects their car from given database collection

Input: Select a listed car

Output: Car selected successfully against user

Process: Car pushed in database against user

Charging Station Allocation:

Aim: To book a charging station for user when requested

Input: Recharge request generated by user

Output: Charging Station allocation toggles against user request

Process: Book charging station for user

Data manipulation:

Aim: to create, delete or update data of users/cars/stations

Input: Change the data of users/cars/stations

Data Types: String, Bool, Double

Process: Data to be persisted in firebase

Data Coupling:

Aim: To link collections and database

Input: Unique ID in each collection

Output: Data exchange in collections

Process: Unique ID (child) will associate with Collection (parent)

Data Association:

Aim: Create relation between user and car

Input: one user

Output: many cars against single user

Process: one to many relation

Table 1: Software Functional Requirements

	Aim	Input	Output	Process
User Registration	To add a new user	Email, Name, Phone Number, Password and Password Confirmation	User added successfully	Saved to Firebase
User Authenticatio n	Login already existing user	Email and Password	User logged in successfully/un successfully	Compare email and Password from firebase
Add Cars	Owner adds a new car into database collection	Add a car, its make, model, range, efficiency	Car successfully added	Car pushed in database
Select Car	User selects their car from given database collection	Select a listed car	Car selected successfully against user	Car pushed in database against user
Charging Station Allocation	To book a charging station for user when requested	Recharge request generated by user	Charging Station allocation toggles against user request	Book charging station for user
Data Manipulation	to create, delete or update data of users/cars/statio ns	Change the data of users/cars/statio ns	String, Bool, Double	Data to be persisted in firebase
Data Coupling	To link collections and database	Unique ID in each collection	Data exchange in collections	Unique ID (child) will associate with Collection (parent)
Data Association	Create relation between user and car	one user	many cars against single user	one to many relation

3.2 Non-Functional Requirements

Alongside the functional requirements, the non-functional requirements also play a vital role. These include the software system’s attributes. On the consensus of usability, mobility, responsiveness, and security, the software system is assessed. An example of a non-functional demand is “How fast does the website load?”. If these requirements are not met, this can lead to user dissatisfaction. For a software development process, there are many agile backlogs. These non-functional requirements allow the design of the system to be limited across these backlogs. For example, if 10,000 visitors are simultaneously using the website, it should take only 3 seconds for the website to load.

Availability of system

If a new module is added, it should not disrupt the availability of the main pages, checkout, or product pages. This deployment process should also not exceed over an hour. If any of the remaining pages face a problem, a popup or notification should be released that includes a message indicating the presence of an issue and should have a countdown to inform the user about the availability of the page.

Performance parameters

For clients opening the website on LTE mobile connection, the load time for the front-page should not exceed 3 seconds.

System compatibility

The service-oriented architecture must be followed by this website application.

Maintainability of system

The automatic email facilities may be unreachable for approximately three hours if they go offline.

Client’s usability

Within 5-tab clicks, keyboard users navigating a website using tab must be able to access the "Add to cart" button from a product page.

Data integrity

The organization must save backups of all the modifications made to the database. Also, to ensure integrity of data, all record transactions should be preserved as well.

Database Reliability

During a failure of a database update, the update process should go back to the relevant updates.

Serviceability of system

The automatic emails sent to applicants can be modified and altered by uploading an XML file, no recompiling is required.

3.3 Domain Requirements

Domain necessities are potentials explicit to software type, with an explicit business vertical and purpose. Both the functional and nonfunctional requirements are included in the domain. All domain requirements have a commonality: they need to satisfy the present standards or acknowledge the feature sets that are common to that type of project.

3.4 Architecture

Various small and single-purpose widgets are put together to make up a larger widget that produces great effects. The quantity of concepts of design are limited to a minimum where suitable, while the vocabulary utilized remains vast.

The number of design concepts is restricted to a bare minimum where feasible, while the entire vocabulary is maintained vast. The software that we used for our project, Flutter, utilizes an identical fundamental concept i.e. widget, that is used to represent the drawing to the screen, scaling, and positioning (layout), the management of the state, interactivity of user, and navigation, animations, and theming. For example, in the animation layer, the pair of animations, tweens, and ideas occupy most of the design space. Moreover, render objects are included in the rendering layer of the specific layout, hit testing, painting, and accessibility. These result in a wide range of vocabulary: various render objects and widgets, tween kinds and animations.

To reach the upmost bar of possible permutations, the hierarchy of the classes are intentionally made wide and short, focusing on composable widgets that are responsible for the performance of one function efficiently. Simple characteristics i.e. alignment and padding are performed as independent components instead of being included in the core maximize the number of possible permutations, the class hierarchy is intentionally short and wide, focused on tiny, composable widgets that each perform one thing well. Even simple characteristics like padding and alignment are done as independent components rather from being incorporated into the core, making key features abstract.

Padding, alignment, rows, columns, and grids are all widgets. These planned widgets don't have their own visual depiction. Instead, their main aim is to alter the layout of another widget. This compositional method is also used by Flutter's utility widgets.

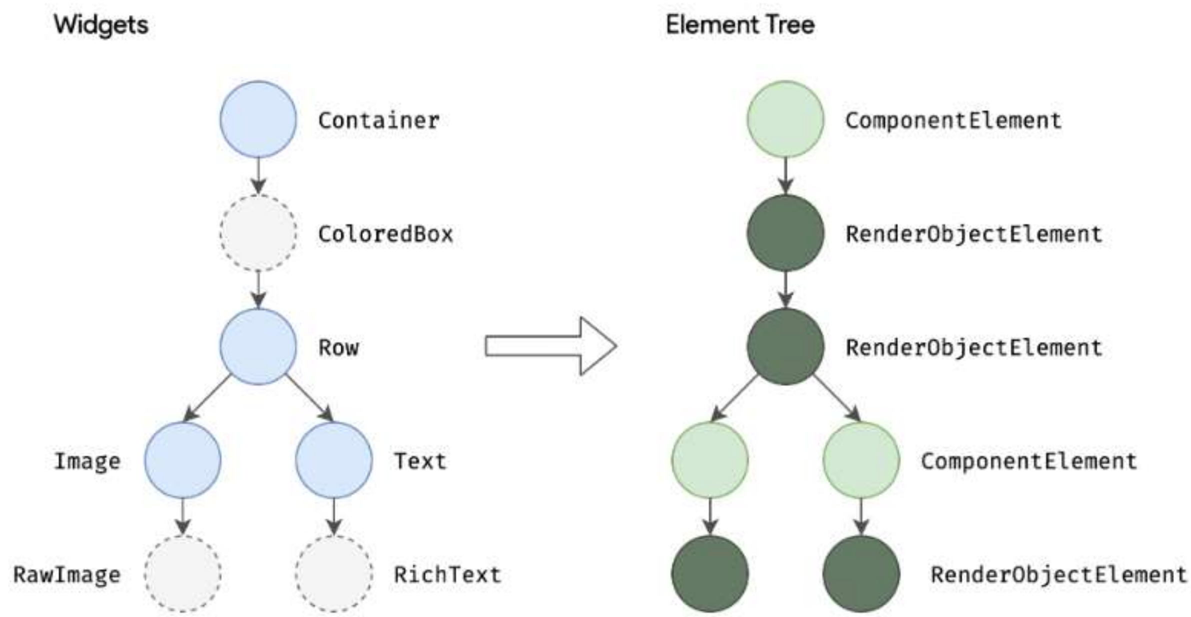

Figure 10: Application widget tree structure

Container, for example, is made up of multiple widgets that oversee layout, painting, placement, and scaling. As you can see from the source code, Container is made up of the Limited Box, Constrained Box, Align, Padding, Decorated Box, and Transform widgets. One of Flutter's distinguishing features is the ability to study the source code for any widget. Rather than sub classing Container to get a customized effect, you may combine it with other widgets in new ways, or just create a new widget based on Container.

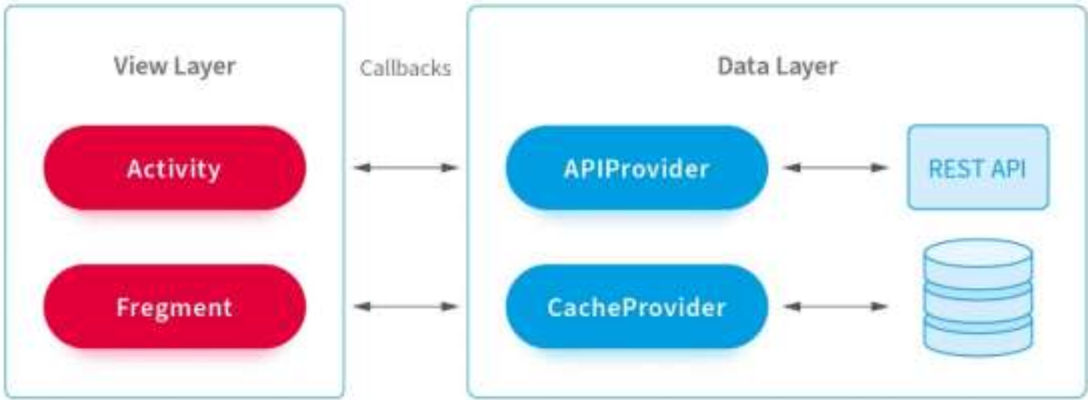

Figure 11: Application Architecture

3.5 Class level Diagram

Class diagrams depict properties, processes, and relationships between classes to explain systems. Structure diagrams are what the Unified Modeling Language (UML) names them. They operate based on object orientation principles. The interaction between things is described by this orientation.

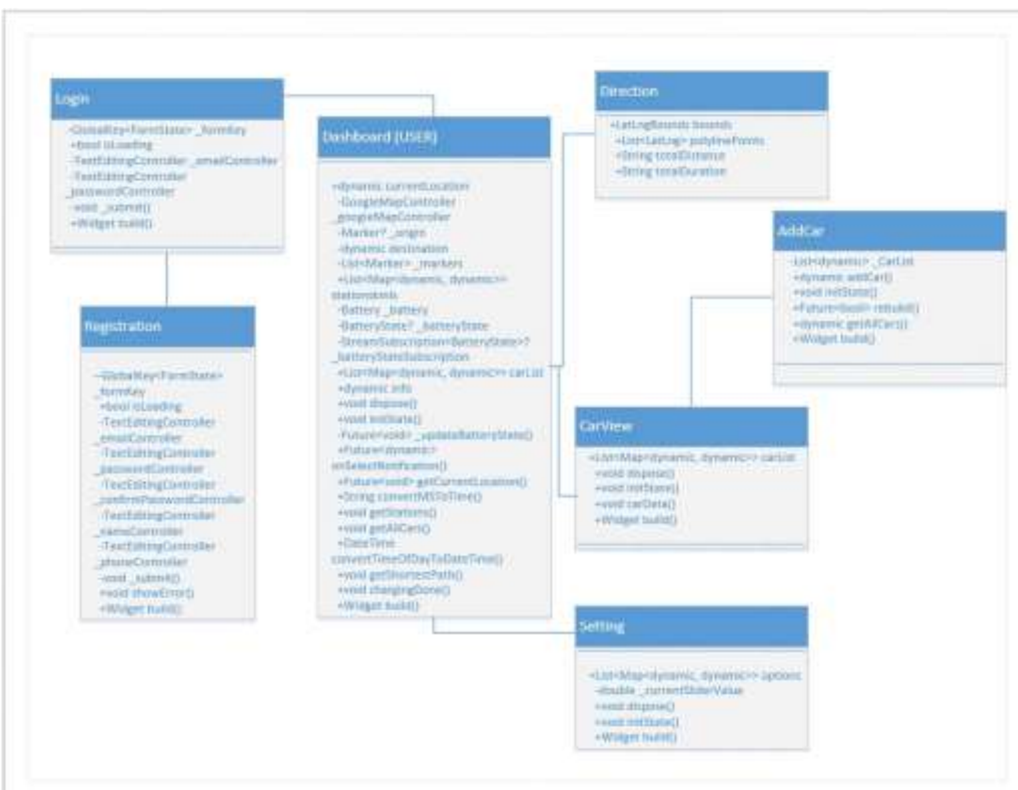

Figure 12: Class Level Diagram (user)

Containing the use of UML and class diagrams, you may design models with attributes, relationships, operations, and intersections. The pathways between classes are visualized in a class diagram as aggregations and associations, as well as the passing on of characteristics and behavior across classes. These are expressed as generalizations.

UML relies heavily on class diagrams. They are based on object orientation concepts and may be used at various stages of a project. They appear as the domain model during the analysis, attempting to produce a representation of reality. The program is used to model software during the design process, and it may also be used to create source code during the implementation phase. Class diagrams are an essential component of every software development project since they serve as the foundation for all software products.

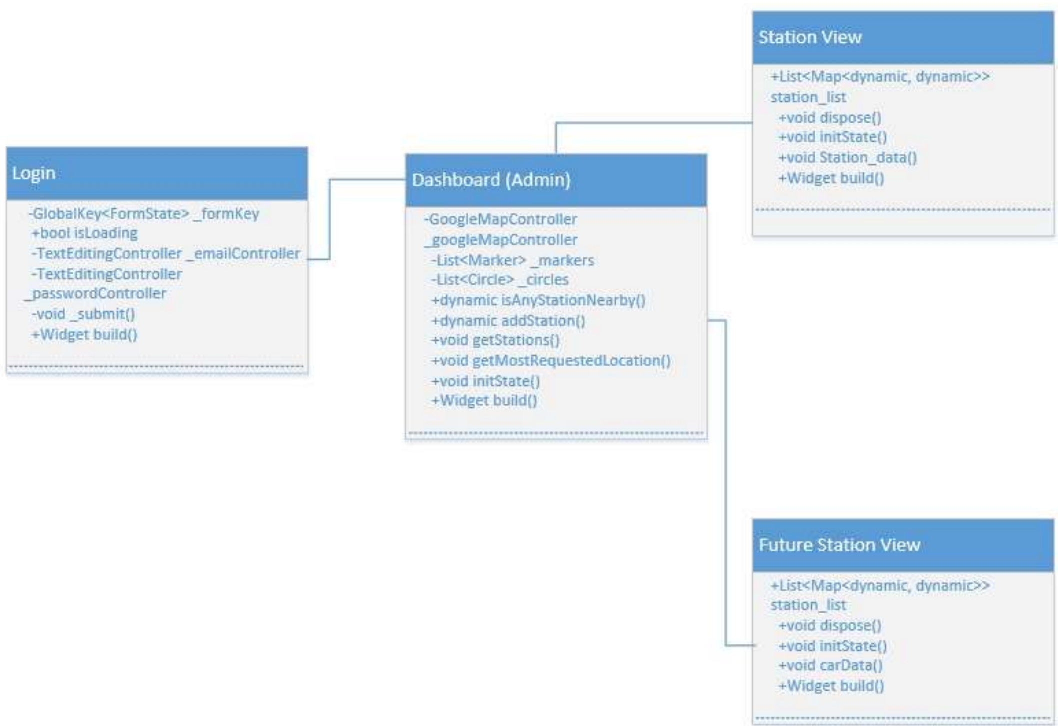

Figure 13: Class Level Diagram (admin)

3.6 Flow Chart

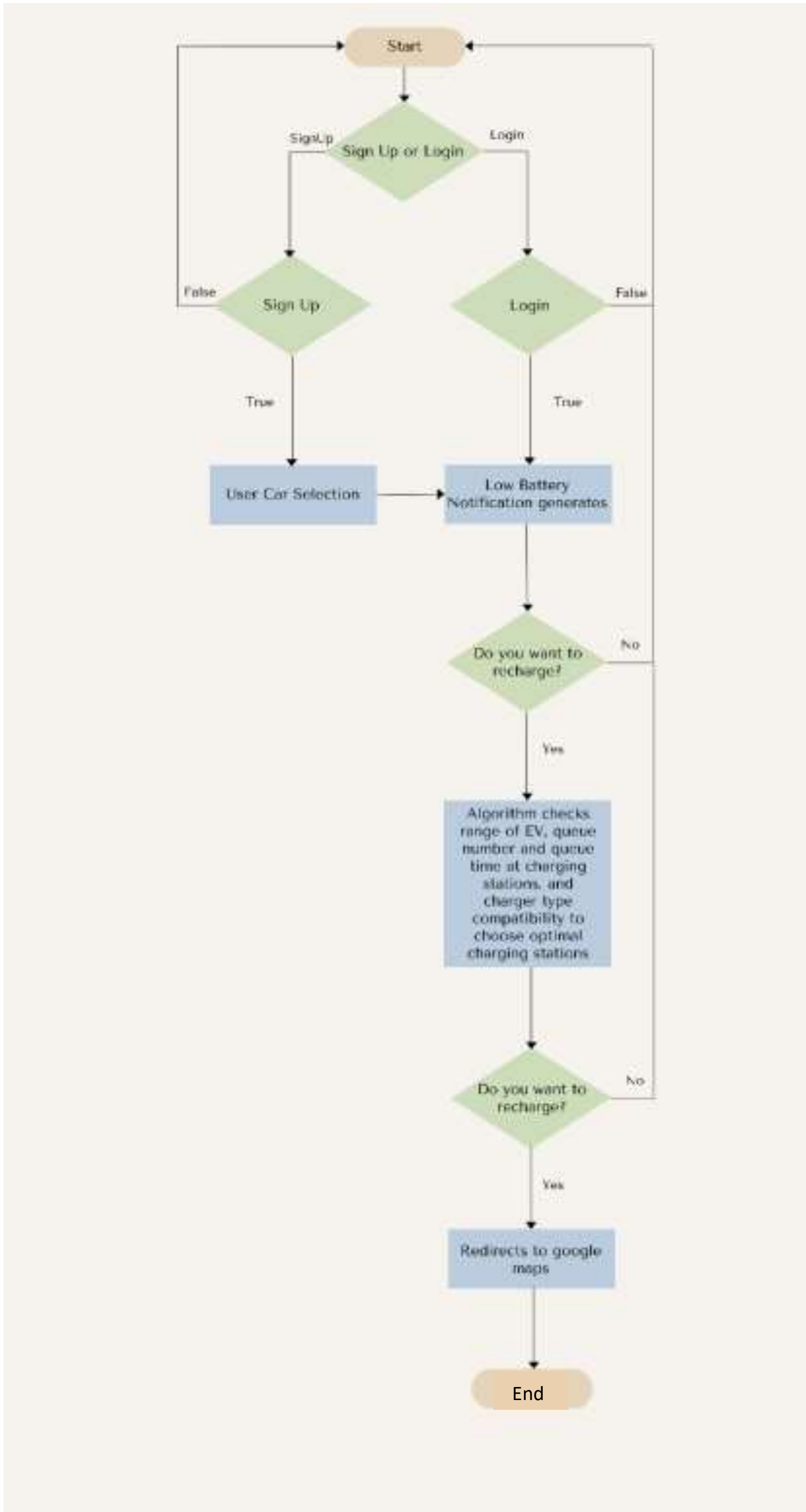

Figure 14: Flow Chart Customer Walkthrough

Chapter 4: Data Collection

4.1 Introduction

Before beginning the data collection for the vehicle models and types, we need to understand the EV requirements in terms of the charging infrastructure. As our goal is also to help setup new and updates charging stations in the future, we need to describe the different factors that play in charging an EV.

4.2 Charging Connector Types and Speeds.

Firstly, we have the different types of charging connector types and speeds. The connector fits into the socket of the vehicle like how a phone is plugged into the charging connectors. As different types of phones have different types of charging i.e. lightening for iPhone and c-type of androids, EVs are compatible with different types of charging plugs as well. The four main types of EV charging are:

1- Slow
2- Fast
3- Rapid
4- Ultra-rapid

These show the power outputs and charging speeds that are offered for the charging of the vehicle. Each type of charger is related to a set of connectors and are designated for low and high use of power with either AC or DC charging [13].

The following table outlines this classification:

Table 2: Classification for EV charging connector types and speeds.

Types	**Voltage / V**	**Current Plugs**	**Range in miles**	**Information**
Level 1	120	AC Plug	5 miles for every hour of charging	Dedicated circuitry is required
Level 2	240	AC Plug	10-20 miles for every hour of charging	Home or public charging equipment installation is required. These stations are the most common and have identical charging rates as the home systems
Level 3-DC fast chargers	480	DC Plug	30-40 miles for every 30 minutes of charging	It is not compatible with al EVs

With the growing demand for charging stations in public, equipment is needed that supports fast charging at higher voltages and currents [14]. As the publicly accessible charging stations are increasing, so are the EV networks globally. Hence an app development for the navigation of charging stations is crucial.

For top-up charging at work and home, AC current chargers are used. For this purpose, there is basically one kind of charge point socket, while some might use a traditional 3-pin plug for emergency backup. The type 2 charging is universal and can be understood by relating it to how both android and iPhone chargers have the same wall socket for charging, but the cable is specific to the phone type.

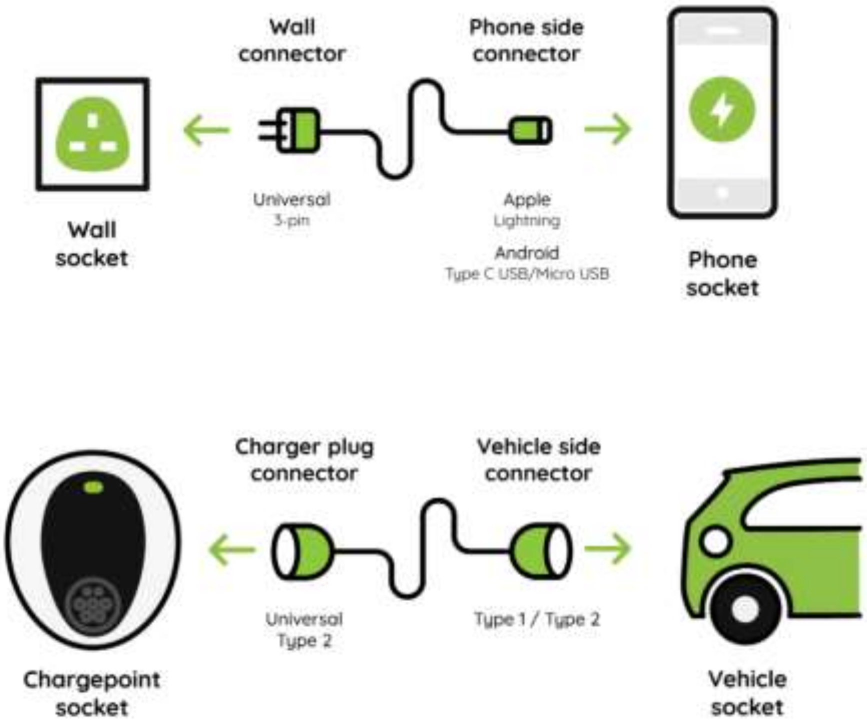

Figure 15: Illustration to show the relation between car charging sockets and phone charging sockets.

Fast chargers are typically rated as 7 and 22 kW, with single or three phase 32A [15]. Although most of these fast chargers have AC charging, but some networks are using 25kW DC chargers that utilize the CHAdeMO or CSS connectors that are describes later in this section.

The slow and fast chargers have the following types [16]:

1- Type 2, with a typical power rating of 7kW for single phase and 22kW for three-phase, giving a range of 25 and 75 miles. These feature a universal connector with a standard socket for charging that all fit too. A driver brings the correct cable to these connectors and is like a wall plug for charging of phones.

Figure 16: Type 2 charger

2- 3-pin plug, that gives a power rating of 2.3kW and a range of 8 miles. This features slow charging and usually not the best practice and should only be used for emergencies.

Figure 17: 3-pin plug

The ranges are given for approximately an hour of charging.

Moreover, for enroute charging, rapid charging is more suitable. There are mainly three main types of DC car-side connectors:

1- CHAdeMO with a typical power rating of 50 and 100kW, giving approximately 75 and 150 miles respectively. these feature an original DC connector.

Figure 18: CHAdeMO charger

2- Combined Charging System (CCS) with provides powers of 50, 150 and 350 kW, providing a range of 75, 225, and 525 miles respectively. These have high power, a neat arrangement with 2 x Type 2 pins, and will probably become the most famous DC standard.

Figure 19: CSS charger

3- Type 2 that has typical power ratings of 150 and 250 Kw, providing a range of 225 and 375 miles respectively. These include only tesla superchargers that provide DC via a type 2 connector.

Figure 20: Type 2 charger

Figure 21: Tesla Type 2

These ranges are for 30 minutes of charging. Models that use the CHAdeMO rapid charging include the Mitsubishi Outlander and Nissan Leaf, while those compatible with CSS charger include the Kia e-Nitro, BMW i3, and Jaguar I-Pace. The Tesla Model S, Model 3, and Model X are exclusively compatible with the Supercharger network, while the Renault Zoe is the only model that makes the maximum use of rapid AC charging [17].

Rapid and ultra-rapid chargers, as the name suggests, are the quickest way to charge the vehicle, often found at locations near main routes or motorway services. These have a high power direct and can be alternating or direct current. With this charger, the EV can be recharged to about 80% in a time duration of 10 to 15 minutes, although a new EV might take about an hour using a standard 50kW rapid charger [18]. This power shows the maximum speed available for charging, but this speed will reduce as the battery reaches a full charge. These times are usually shown as a charge to 80% because the charging becomes slower after this threshold. Furthermore, all these rapid chargers relate to charging cables to the units and can only be utilized on EVs that are compatible with rapid charging. As these connector types can be easily identified, these compatibilities can be easily checked using the vehicle manual or checking the on-board inlet.

4.3 Types of Electric Vehicles

Various types of electric vehicles are continuously being developed and innovated to provide potential user choices and to overcome any drawbacks linked with current vehicle models. The four main types of electric vehicles are [19]:

1- Battery Electric Vehicle (BEV)

2- Hybrid

    a- Hybrid Electric Vehicle (HEV)

    b- Plug-in Hybrid Electric Vehicle (PHEV)

3- Fuel Cell Electric Vehicle (FCEV)

The following figure briefly describes the four types of electric vehicles currently available in the market:

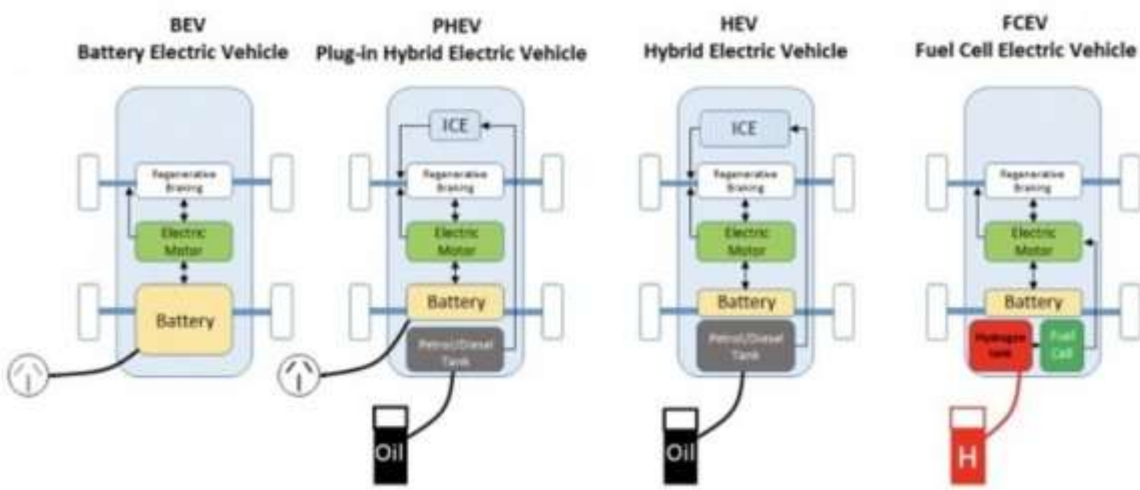

Figure 22: Four main types of electric vehicle architectures.

For our project, we will be focusing on Battery Electric Vehicles only. This type of vehicle is driven solely by an electric drive-train powered by a battery. These do not have an internal combustion engine and the electric energy is stored in a battery pack with varying size according to vehicle power and torque requirements. This battery is plugged into an electricity grid and charge is stored in the battery, consequently utilized by the vehicle. This type of vehicle can include one or more electric motors to provide power.

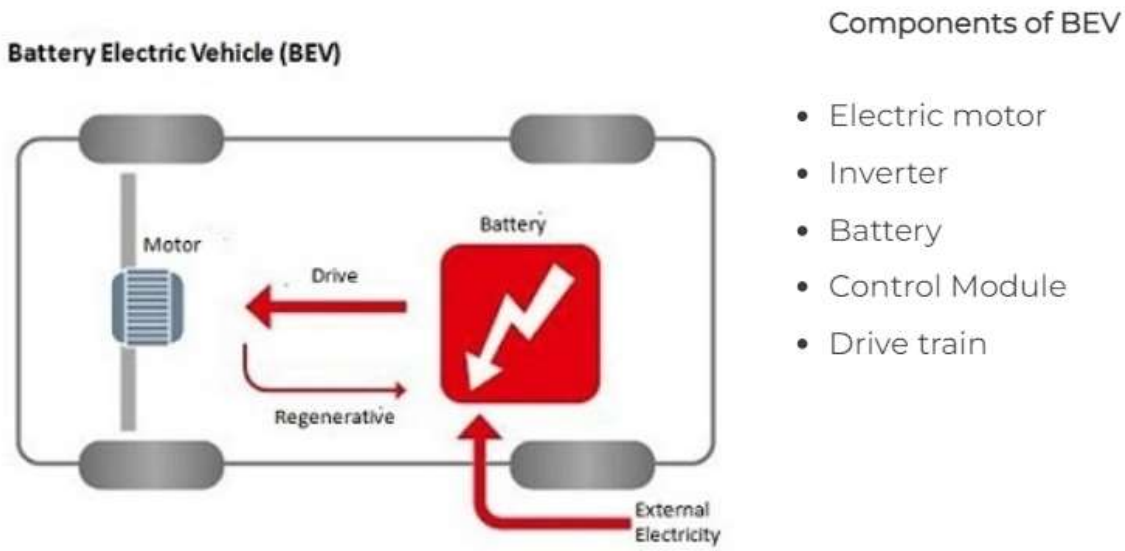

Figure 23: Main components and architecture of a BEV

The battery of an EV undergoes continuous cycles of ‘discharging’ and ‘charging’ when driven and plugged in to a charger respectively. By the repetition of this process repeatedly, the amount of charge that can be held by a battery decrease. This consequently decreases the time and range between each charge. Most manufacturers include a 5 to 8-year warranty with their battery, but currently, an electric battery lasts around 10 to 20 years. To understand the working principle of the electric motor required only a simple explanation. When the cars accelerator is pressed, the motor powers the car by taking the energy stored in the batteries. Some cars include regenerative braking where the kinetic energy in the wheels is used to power the batteries as the brakes are pressed. This makes the electric motors work as generators as the forward motion of the car is converted back into electricity. Most vehicles use lithium-ion (Li-ion) batteries as they can easily be recharged. They have a high energy density compared to typical nickel-cadmium and lead-acid rechargeable batteries. This allows manufacturers to reduce space and size of the battery packs. These batteries have also proven to be safer than other alternatives as safety is crucial when it comes to automobiles and their frequent use. Also, electric vehicles are equipped with charging safeguards to protect the battery for the repeated process of charging sessions.

As our application includes all vehicle models and makes that are compatible with charging stations, we collected a comprehensive and detailed data sheet for the hundreds of models available in the market for EVs. This information was classified according to Company name, body type i.e.

hatchback, SUV, sedan, small van, cabriolet, station/estate etc., model, efficiency in Wh/km, fast charge in km/h, range in km, and type of charger. The table below summarizes our findings in the form of a bar graph:

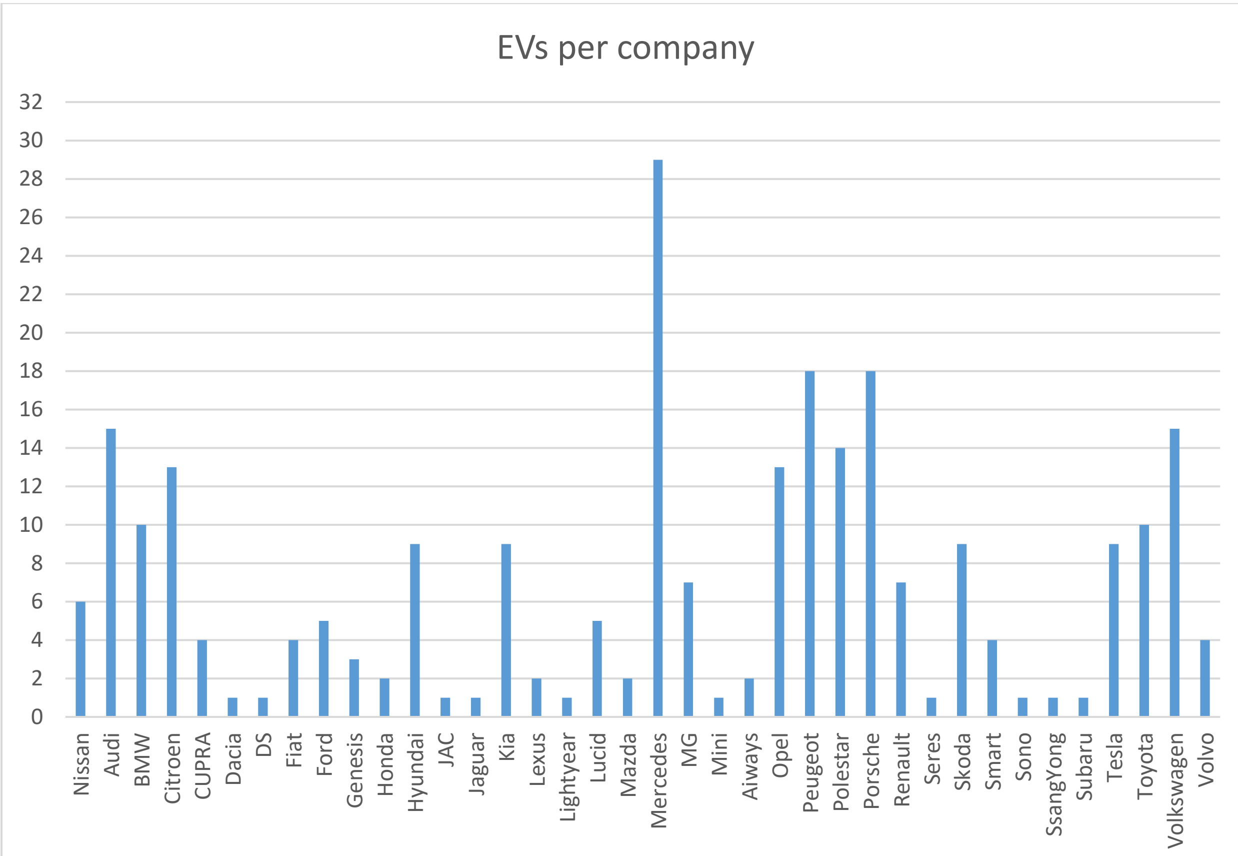

Figure 24: Complete EV data collection chart

Chapter 5: Tools and Technologies

5.1 Introduction

Our project mainly consists of three interfaces i.e.

- User App
- Admin App
- Super-admin panel

Both the user app and admin app are developed using Flutter with the same database i.e. Firebase used at backend for the two-mobile application, a super admin panel is also available that is developed with the help of React JavaScript used for performing all CRUD operations. Google Maps APIs have also been used to help extract information for the algorithms that will benefit the user as well as the station provider.

5.2 Android Studio

Coding that is more efficient. Instant Run, as soon as there is a change in the code, it is picked up. This alteration is evident without restarting or rebuilding of the application. During the development of an application, this increases the speed of the development process using the intelligent code editor. This editor comes up with suggestions to enhance the compilation and analysis of the code. The dropdown selection simplifies checking of this advice.

Figure 25: Android Studio Logo

The benefits of using the android studio include:

- Rich and Connected Apps
- Firebase and Cloud Integration
- Easy to Use
- APK Analysis

5.3 Dart and Flutter SDK

Dart is used by Flutter because it eliminates the requirement for a separate declarative layout language like JSX or XML. Dart's layout is declarative and programmatic, which allows developers to comprehend and see the code quickly and simply. Furthermore, because the layout is in a single language and a standard, it makes it simple for Flutter to give extra tooling.

Figure 26: Dart/Flutter Logo

Another reason for the duo's popularity is that Dart, when necessary, employs Just in Time compilation. This cuts down on development time and allows for speedier responses.

Most Dart's features are like those of static and dynamic languages, making it simple for developers to learn and grasp the language.

Architecture

Flutter is a UI toolkit that works across platforms and makes it easier for application to interact with the platform services directly. It also allows the reusability of the code across various operating systems i.e. android and iOS. The main of this software is to simplify the work of developers in the development process of making high-performance systems that appear natural across the many platforms. But this function lets the programs to maintain their differences in code.

The applications of flutter work in virtual machine (VM) in the development process. This allows the efficient reloading of any alterations without the need for a complete recompile. These applications are compiled directly to the machine code for release, whether they're directing Intel x64 or ARM commands, or to JavaScript if they're aiming the web. The framework is free of cost and open source, with a permissive BSD license, and a lively bionetwork of third-party packages to augment the basic functionality.

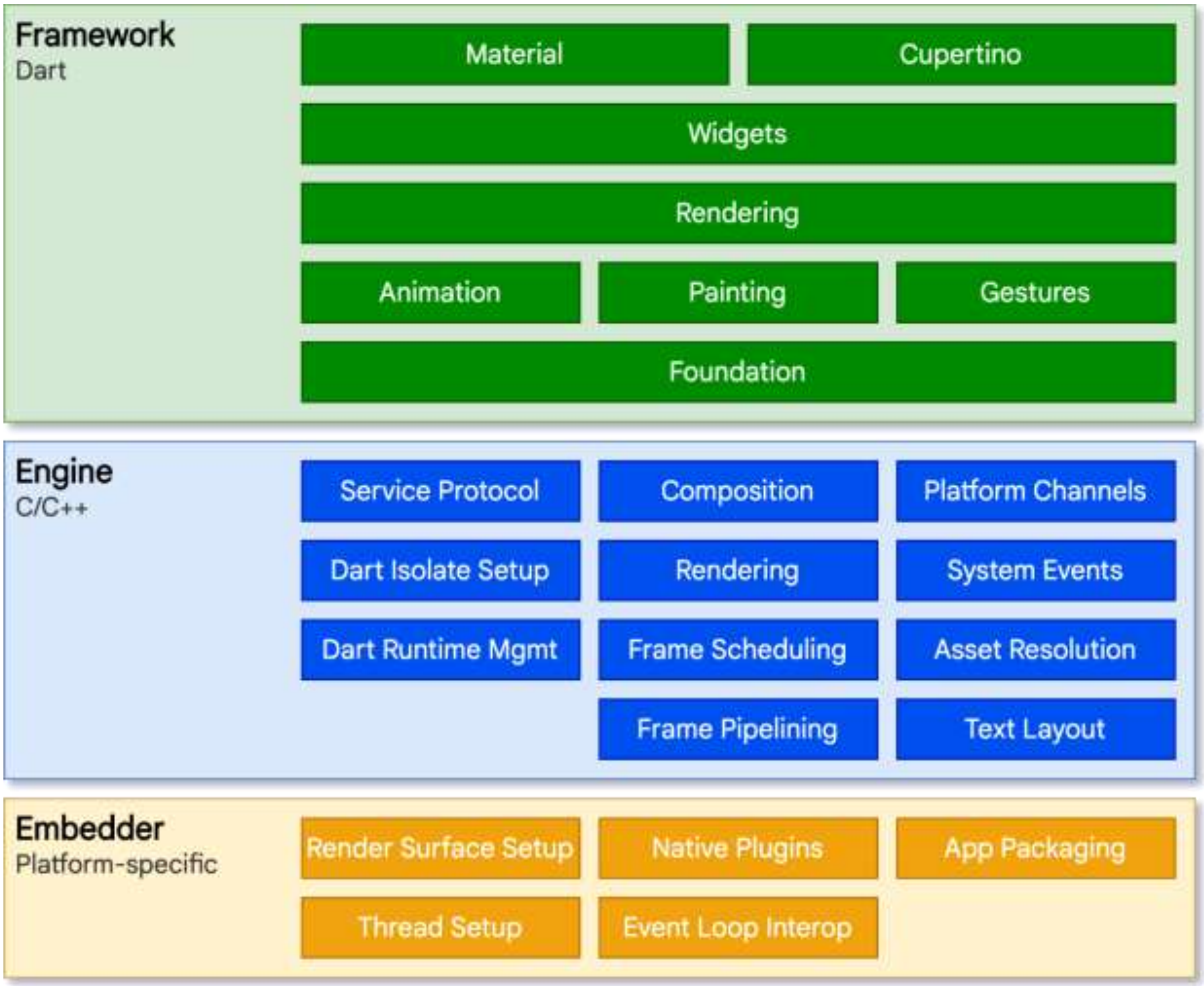

Figure 27: Flutter Architecture

5.4 Firebase

By providing safe admittance to a database directly from the code on giving safe access to the database directly from client-side code. The database of firebase Realtime helps in the creation of collaborative and complex applications. This data is locally stored, while the Realtime events continue to process during user offline mode. This creates a responsive environment. When connectivity is regained, this database updates any discrepancies instantly between the remote and local data changes or updates.

Figure 28: Firebase Logo

The description about the format of data and whether the data is in written or read form provides an expression-based powerful rule language in the Firebase Realtime database. This language is known as Security Rules. Developers of software can decide the users of the data and how they can access it during the use of Firebase authentication,

The Realtime Database is a NoSQL database. This means that it is different from the relational database due to its functionality and optimization. Only fast actions are allowed in the API of Realtime database. Hence, a real-time experience can be created to serve many people with optimal responsiveness. This results in the critical issue of the way people will access the data and organization of this said data.

5.5 Google Maps APIs

The Google Maps API allows you to develop amazing apps that are only limited by your imagination. You may develop new sorts of visualizations that bring your data to life by combining the many components offered. What's more, your data may now teach others about the environment in which they live. People can now notice patterns that were previously unseen, and these patterns can provide your company a huge competitive advantage.

Figure 29: Google Maps Logo

The main drawback for a new developer is that the free Google Maps API license is relatively restrictive. It is the licensing, not the technology, that is restricting. You must discover ways to play with the API that do not violate the restrictions while yet allowing you to get the full taste of it

5.6 ReactJS

The main aim of ReactJS is the creation of a User Interfaces (UI) that aids in the optimization of application speed. Using a virtual DOM (JavaScript object), the speed of an app is increase. This DOM has proved to be speedier than the conventional DOM. Furthermore, ReactJS can be used on both server and client side which increases the app's speed. The virtual DOM in JavaScript is quicker than the conventional DOM. ReactJS may be used on the client and server side, and other frameworks as well. This is done by the employment of data patterns and components to increase ability to read, alongside helping to assist the maintenance of larger applications.

Figure 30: ReactJS Logo

Chapter 6: User App Development

6.1 Introduction

For making the decision to travel, the owner of the EV first needs to figure out if the residual power in their EV is enough to reach the destination. If the power is not sufficient, the user will go to the nearest charging station. Without proper navigation, the user's choice will be independent to any traffic or queuing data and can be considered as 'selfish' as no consideration of taken for the operation of the charging station. Hence, this unguided behavior will lead to inefficient charging situations, where congestion at the station could become serious. This problem can be resolved by providing the users with a platform to navigate charging stations suitable for their situation and requirements.

For the design of an effective navigation system, the above-mentioned problems can be dealt with by the development of an integrated system with multi-network collaborative working mode by considering the traffic information and dynamic queue information for the charging stations. The modules used in this system are EVs, ITS (intelligent transportation system), and IEC (information exchange center). The functions of each member are as follows:

1. The ITS will provide information regarding the traffic for the IEC to designating navigation strategies to the charging station. This information will directly reduce the energy and time consumed by the EV to reach the suitable charging station.
2. Dynamic queue information will be automatically uploaded to the IEC and updated in the channel. The charging facility and its geographical location, along with the queuing situation will directly influence the choice for charging station.
3. With the continual updates to the system, the IEC will enable an efficient use of the charging stations and provide the user with the optimized navigation. After the receival of request by the user, the IEC will combine information from all members of the system to design the navigation strategy customized for the current user. If user accepts the proposed navigation, charging reservation will be made at the designated charging station.
4. Users of the software and their choices will directly affect the congestion rate at the charging stations and indirectly affect the choices of the other users. This makes the framework work a whole in a broader sense.

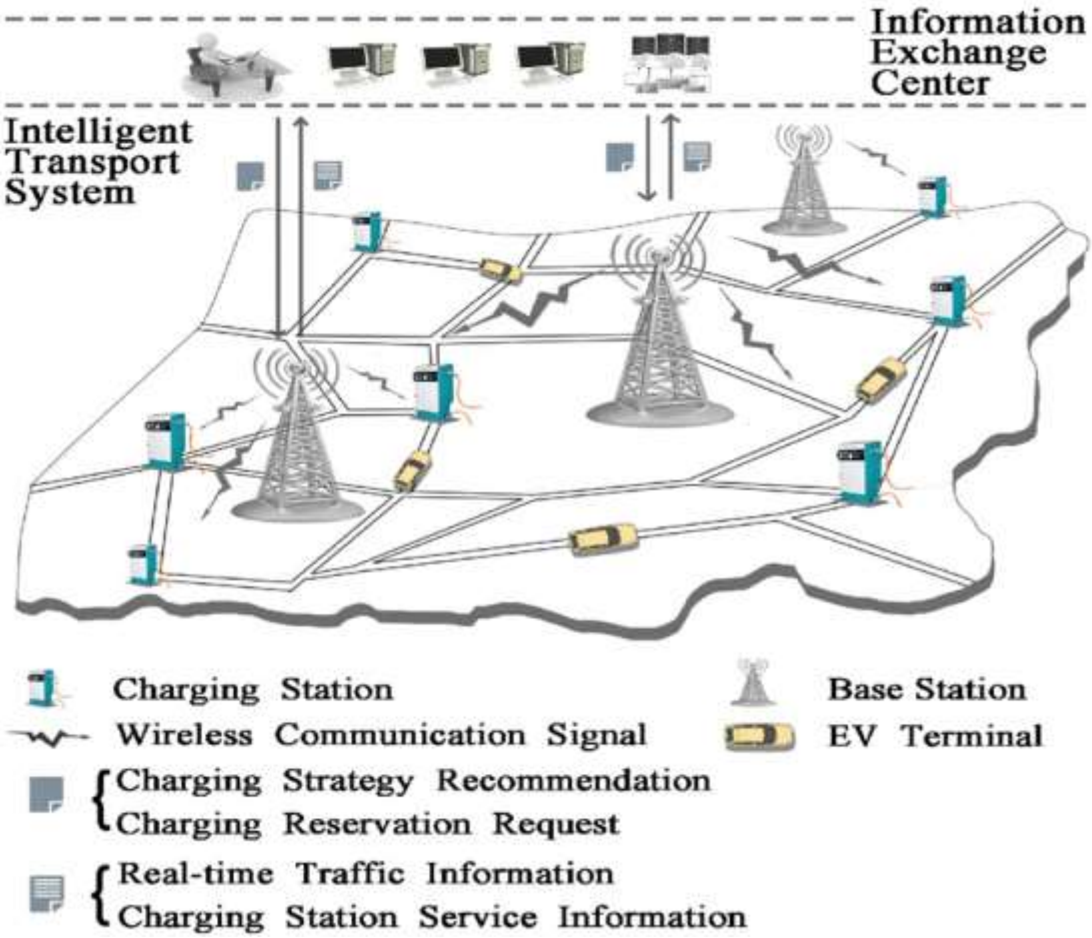

Figure 31: Navigation process for the integrated system.

During this entire operation, the system exchanges information only with the IEC, hence the information link and transmission is reduced as much as possible. Additionally, the IEC only updates the information at the charging station if the EV comes in for charging. The following assumptions are made:

1- The time of designating the charging navigation and communication delay of the navigation strategies is neglected.
2- All charging stations work dependently and are not in competition with each other.
3- The EV completed the entire charging at the charging stations and leaves immediately after the charging is done.

6.2 System Level Diagram

Smart navigation will be started as soon battery level of electric vehicle will drop to a certain level, Data is managed in states in flutter application and pushed into real time database, on the retrieval of data a path will be computed from user source to destination.

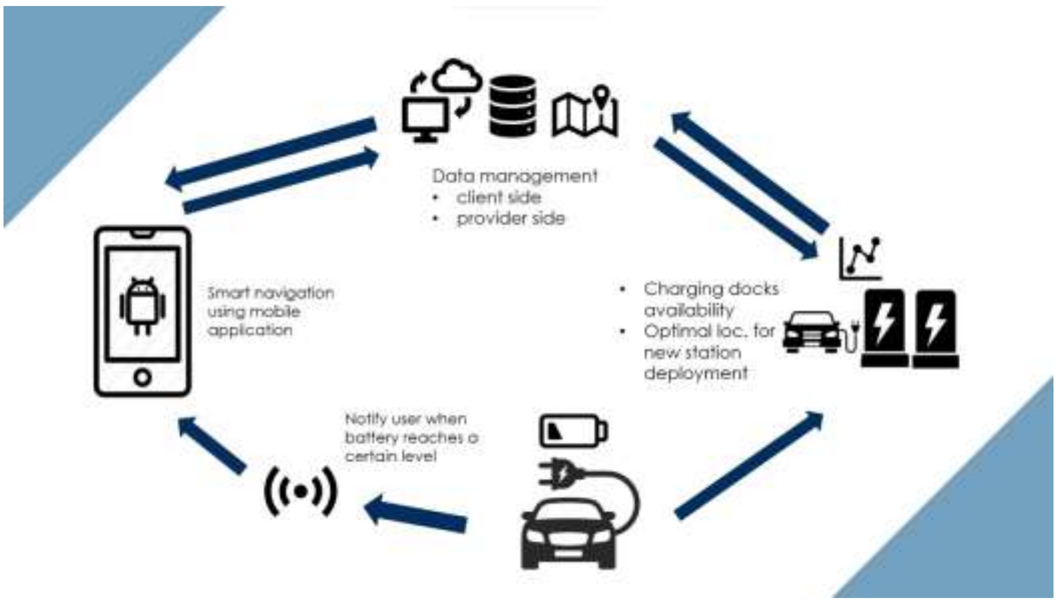

Figure 32: System Level Diagram

The initial state of user is null when car battery reaches its minimum limit it will switch its state to true and user will be given by an optimal path, this state also gathers other nested states for the allocation of charging station queue, queue time, electric vehicle range and its charger type. Decision will be based on the parameters given to each user nested collection structure and after which a new path will be generated on real time.

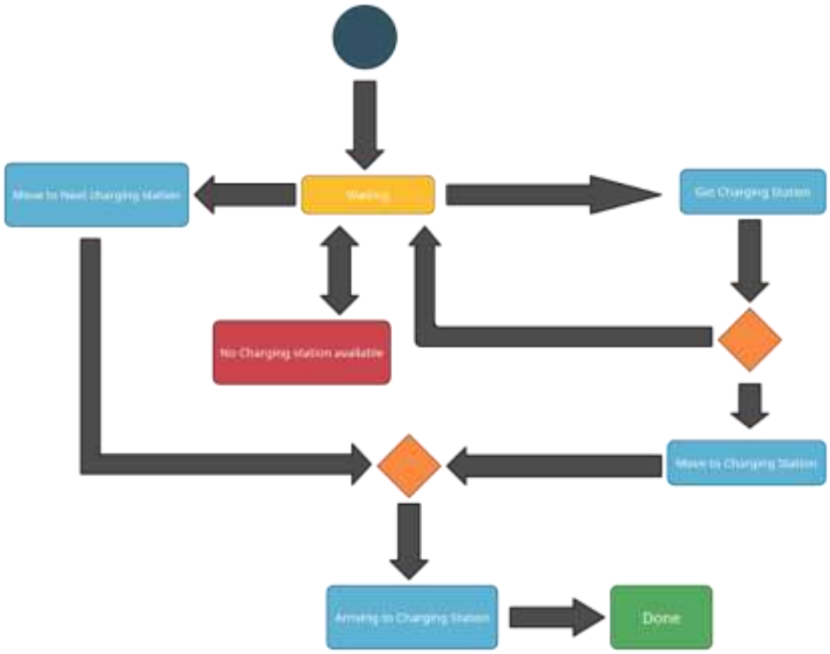

Figure 33: State Level Diagram

When the task queue is not empty, the electric vehicle starts in the state of Waiting and arrives at Get Destination. The electric car will then choose a driving path based on the present location and the destination location using the algorithm described above.

If R, go to CS. The electric car will then enter Replace Battery mode, where it will replace the battery with a fully charged one. The electric vehicle will thereafter enter Move to Next Stop state when the transition T7 is fired. The electric car will then return to its Get Destination state, with the starting point being the location of a charging station to refill the battery. The aforesaid method will then optimize a new path based on the new starting point and destination.

6.3 UI Creation

The following mind map was used to draw an abstract idea for the project during the initial stages:

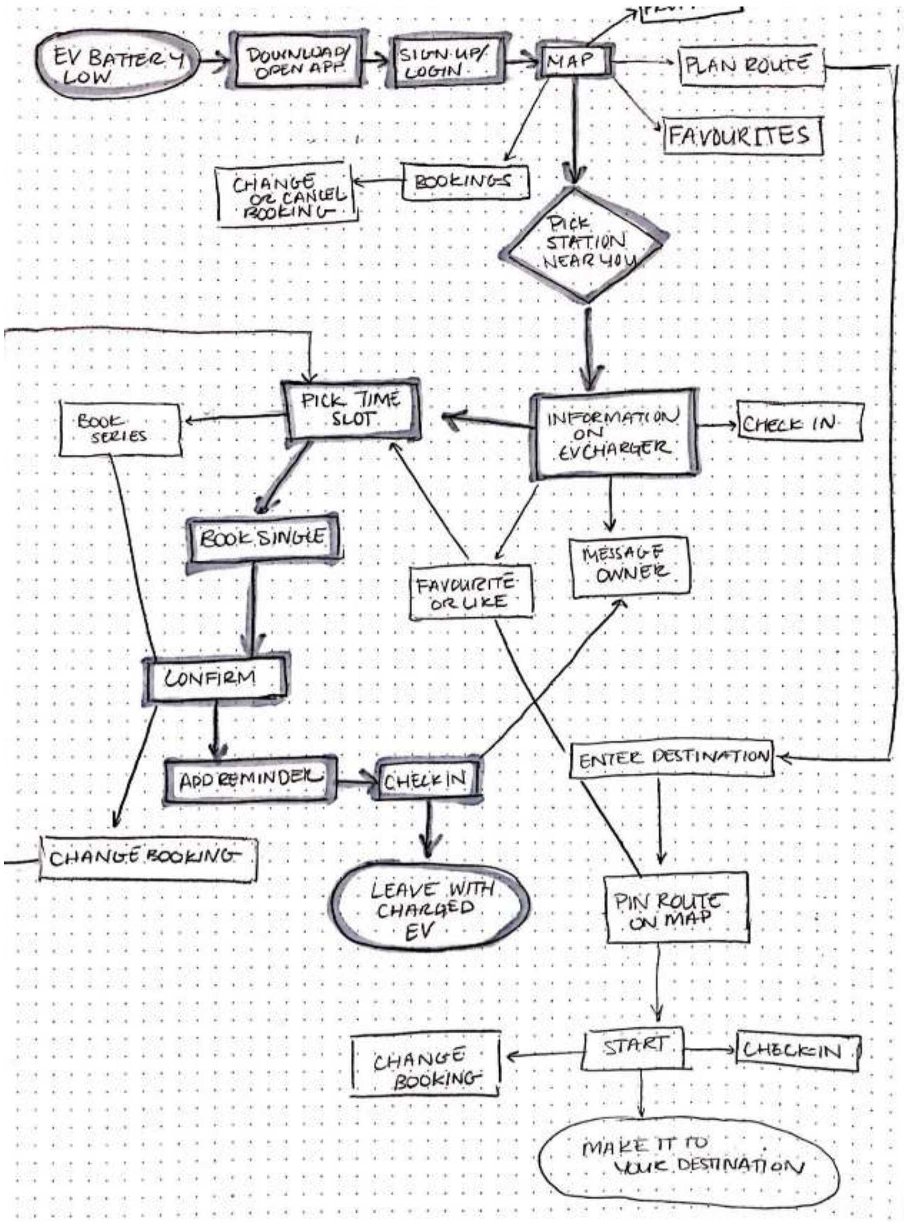

Figure 34: Mind Map

6.3.1 Mood Board

The design of user interface is made by extensive research on optimal UI for all age groups and personalities. As the app resonates with clean energy and eco-friendliness, we used blue as the primary color as it symbolizes trust according to color psychology. Also, blue represents connectivity and futurism.

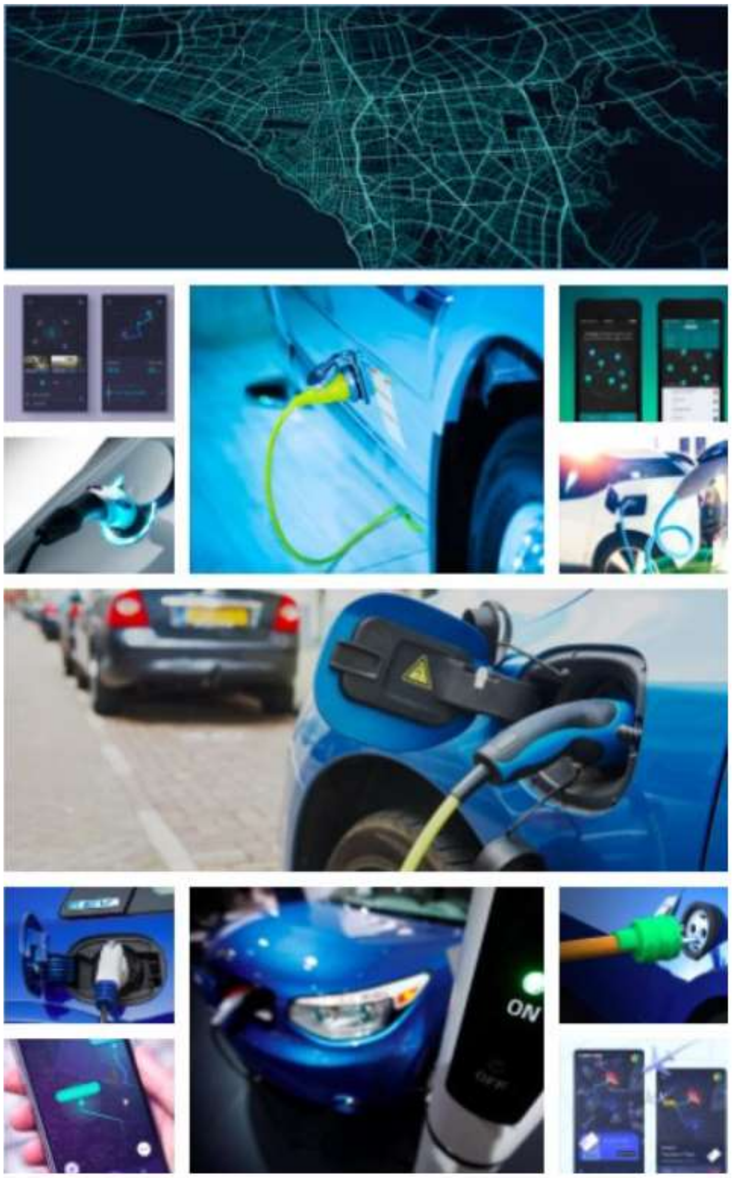

Figure 35: Blue color represents connectively and futurism.

Furthermore, we aimed to portray a sense of movement and speed and hence kept the interface minimalistic and simple to use, while also used creative colors and fonts to attract the user.

The logo was developed to represent the brand. Using the above-mentioned color psychology, blue was selected for the logo as well as to convey trust to the user. The varying degrees of blue represents the aspect of growth and innovation in our application. The design also adds movement and speed to portray rapidness and a futuristic look.

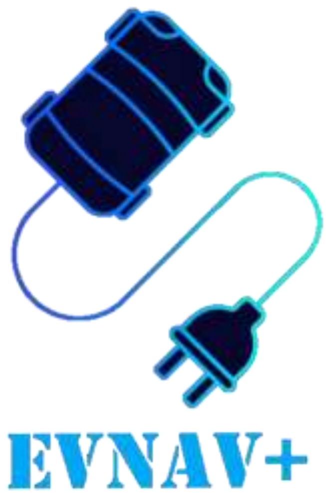

Figure 36: Application logo

6.4 API Key Generation

Google maps services and functionalities were cross integrated with our mobile applications to use the functionalities provided by Google Maps itself and to upgrade and work on those services for the creation of our own algorithms.

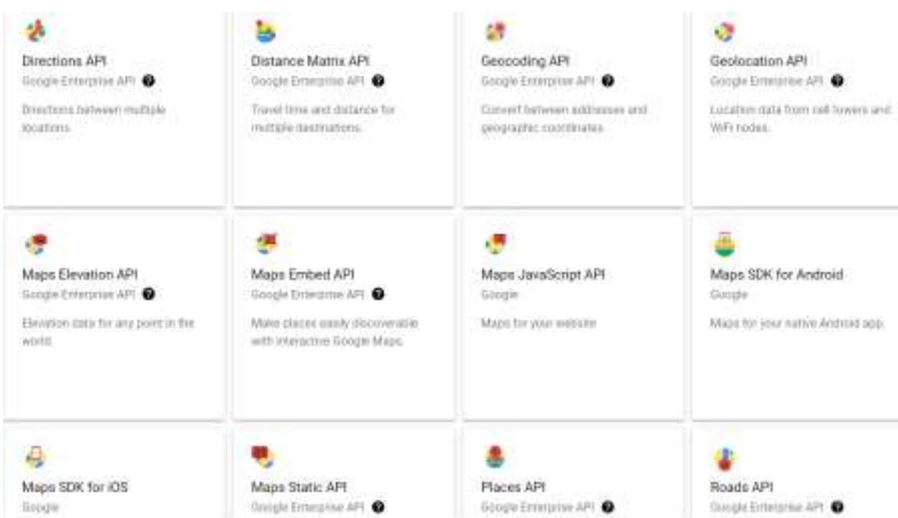

Figure 37: APIs used in our project

6.5 Integration of Google Maps

The primary purpose of our application is the navigation of users to the optimal charging station. The application used by most users around the world for navigation is google maps. Hence, connecting these two pieces of information, we decided to integrate google maps onto our application to not only make the application versatile and efficient, but also to make it easier for the user to use as they are most definitely already comfortable with the google maps interface for navigation.

Google maps can be used to develop customized applications based on maps for any company. This includes any interactive or static map, browser, or mobile based application. As google maps allows customization in their lines, markers, polygons, images, and colors, these features can be tailored to make it suitable for any company. We chose to keep the default setting for google maps as it would be the most easy-to-use for our users. The ‘places’ provide an extensive location data for over a hundred million locations around the globe. This helps the users receive important information relating to decision-making and guidance as addresses, contact details, opening hours and rating and reviews. Also, ‘routes’ feature is used to provide real-time and comprehensive data for the traffic information that is vital for our application. These routes are used to select the shortest and fastest route to the suitable charging station from the user’s current location.

6.6 Algorithm Implementation (For Navigation Towards Optimal Charging Stations)

As our design is based on optimal route selection and the navigation system is based on crowd sensing, we use the location of users with the information of their battery range and charger type also catering knowledge of other users at nearby station to provide the optimal rout. This information is directly collected and sent to administration using the devices. This information includes number of EVs at the arriving and present at the charging station. This information is passed onto the control center and is used to make the optimal EV route selection and navigation. This information is then sent back to the user via wireless communication i.e. 4G and WIFI. As the information of the charge station and traffic is updated dynamically, the decision making of the application should also be updated dynamically.

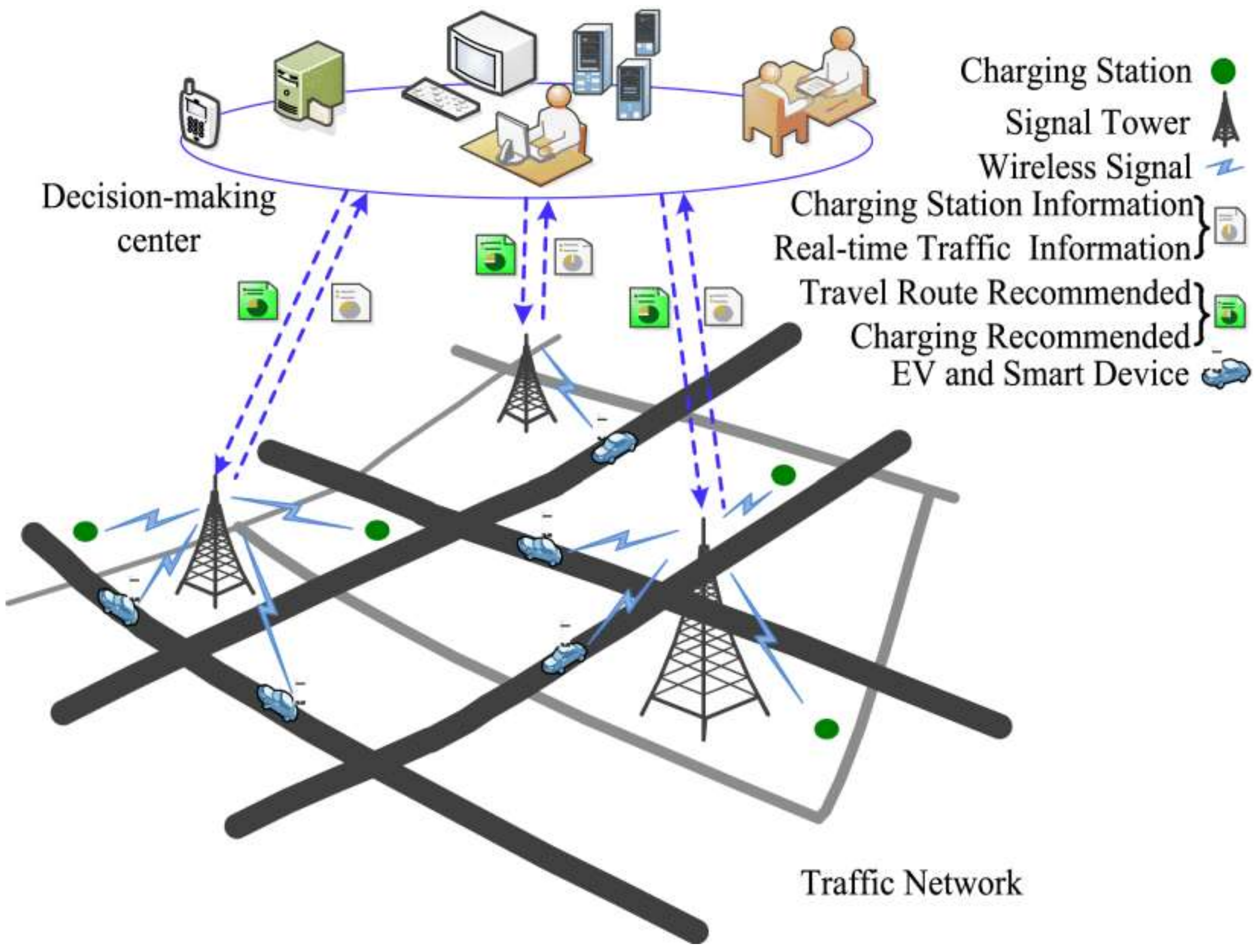

Figure 38: Route selection & charging navigation strategy according to crowd sending for EV

The algorithm basically works in order of selection of the following factors:

1- Range of battery
2- Charger type
3- Distance
4- Waiting time

All four of these factors play a critical role in choosing the optimal charging station for the vehicle. The information for range of battery is updating according to the battery type of vehicle and related data, the charger type is pre-set according to vehicle model and make but can also be changed by the user manually in application, the distance is found using google maps, and the waiting time is decided by current and past data of all users currently utilizing the application. The range of the battery shows us how far the vehicle can travel in the current charging and whether it can reach the charging station in time. The four factors are run simultaneously in the algorithm and the optimal choice of charging station is selected. If no charging station fits any of the four main factors, the user is informed that there is no charging station available in the vicinity that suites them.

When an EV user plans for charging their vehicle, they face on of the following situations:

- They can reach the nearest charging station directly as it in under its range also, the charging station is free, and the port type is compatible.
- The nearest charging station is occupied so it must look for the next nearest charging station.
- All the compatible charging stations in the vicinity of the vehicle's range are already occupied so it moves to the one with least queue time.
- No charging station available within the range of the vehicle or queue time exceeds a certain limit.

For the first problem statement there are further two cases;

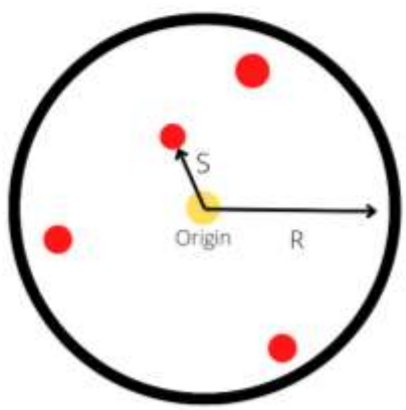

Figure 39: Nearest Charging Station Selected

The nearest charging station is free and compatible with your car hence the car I redirected towards it.

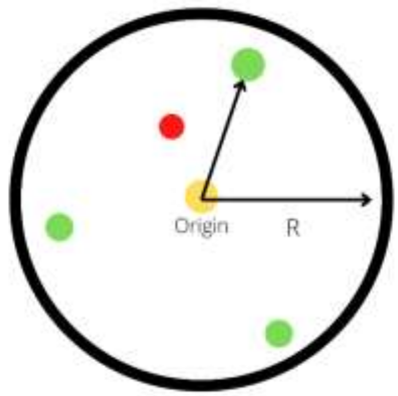

Figure 40: The nearest station is not compatible;
it moves to the nearest compatible one

If the nearest station is not compatible with the car so it moves to the next nearest station which is also compatible with the vehicle.

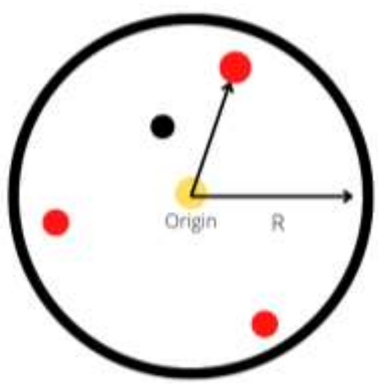

Figure 41: Nearest Station is filled so it moves
to the next available Station

S1 exceeds queue time threshold, user is redirected towards S2

The nearest station exceeds a certain queue time due to which the car isn't directed towards the nearest and chooses the next nearest and available station.

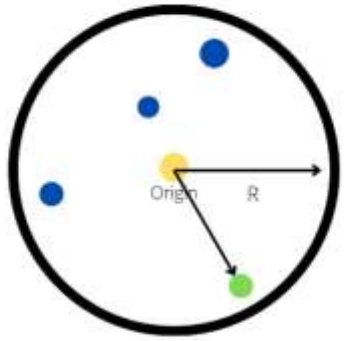

Figure 42: All Stations within the range are filled so it moves to the one with least queue time

S1, S2 and S4 exceed queue threshold while S3 doesn’t and can be used for charge.

If all the stations within the range are already allocated to other users, the algorithm redirects the user to the station with least queue time also catering for the range as well.

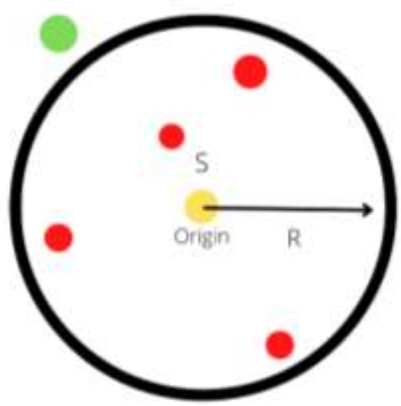

Figure 43: All Charging Stations within range can't be used

S1, S2, S3, S4 are either incompatible or exceeds the queue threshold.

All the charging stations within the range of vehicle can’t be used due to which user cannot be directed towards any destination.

6.7 Algorithm Integration

The platform used for bringing the designed algorithm to life is Flutter SDK (Software development kit) with the use of DART language. With multiple iterations, an optimal user experience was designed.

1- The user is first introduced to the login/registration screen when they click on the application.
2- The dashboard shows up where the user can easily select their electric vehicle company and model out of our wide selection of electric vehicles available.

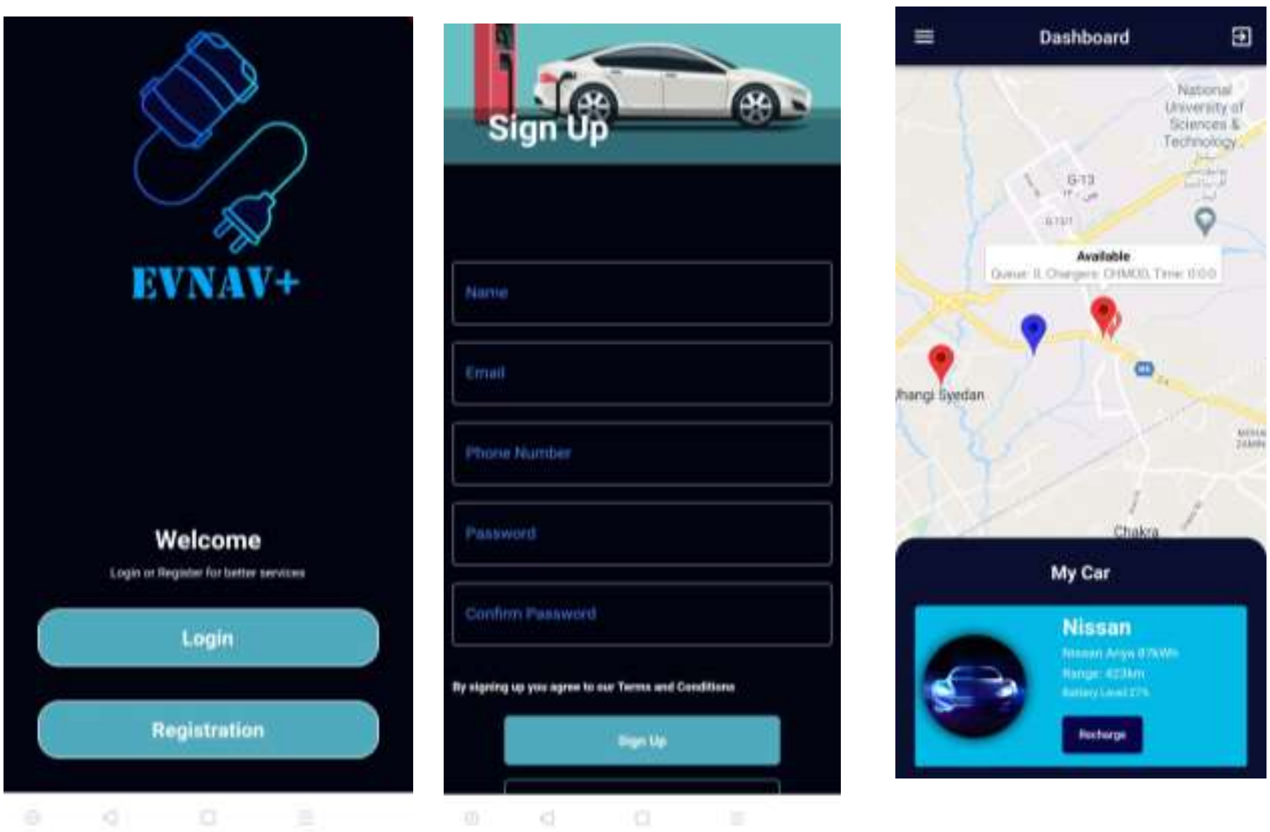

Figure 44: User Interface

3- The dashboard also includes a map that displays the charging stations in the vicinity.
4- If recharge is required, the user can press the recharge button at the bottom of the screen.
5- The app then asks user to select the amount of time required for charging i.e. 1-hour charging.

6- The algorithm sets in play and finds the optimal charging station for the user. The User App displays directions to the suitable charging station on the map. This navigation is done directly by google maps.

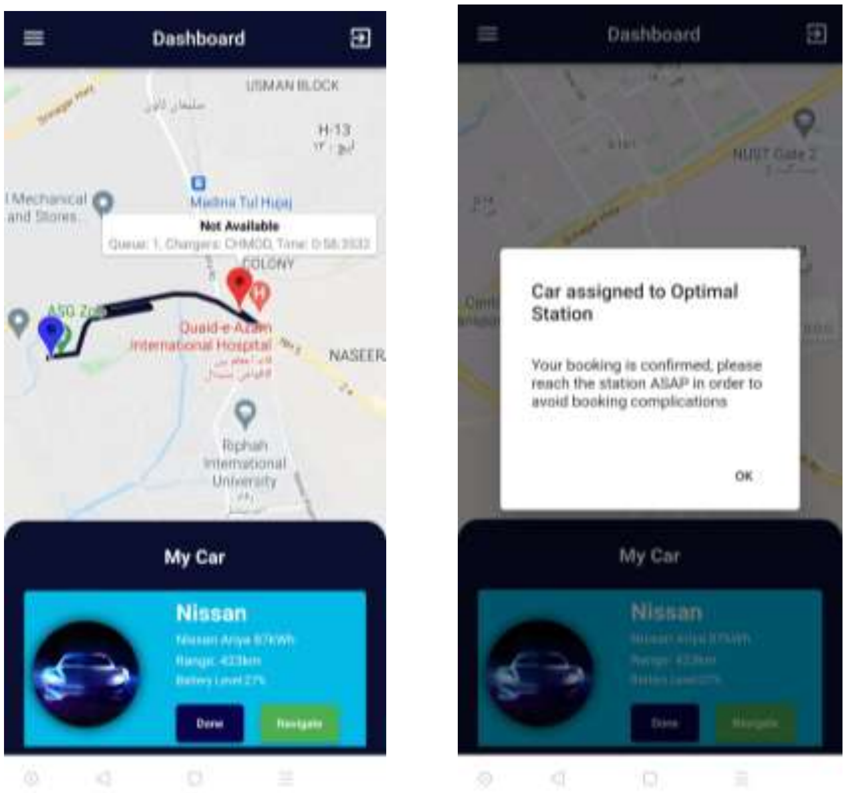

Figure 45: Optimal Charging Station Allocation

7- If the battery level is below the threshold of 30% (assumption), the user will receive a notification as a reminder to charge the vehicle. If the user wants to charge the vehicle at that time, they will press the recharge button.

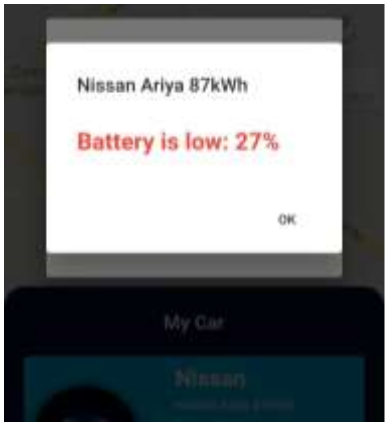

Figure 46: Low Battery Notification

8- The algorithm will again provide directions to the nearest charging station.

9- If all the charging stations are greatly occupied or there is no station in your vicinity the following notification is generated.

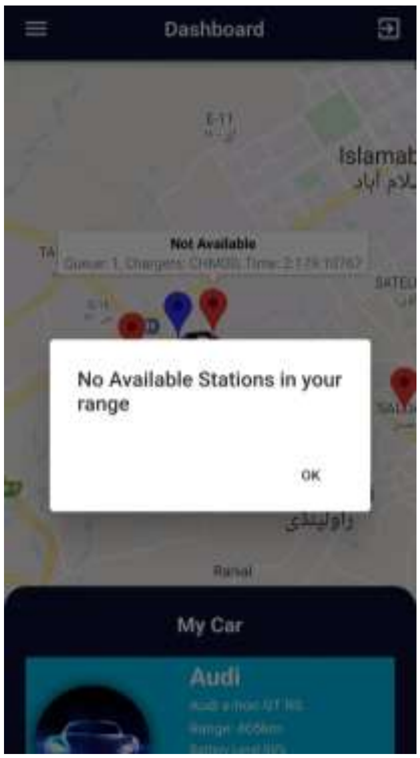

Figure 47: Charging Station out of Range

Chapter 7: Admin App Development

7.1 Introduction

While the user app covers the front-end for the customer and provides them with the optimal charging stations, the admin app runs current and past data for the creation of new charging stations based on requirements. As the number of electric vehicles is rapidly increasing, the need for new charging stations is increasing as well. The selection of location for the construction of a new charging station is very complex and time-consuming as multiple factors need to be considered. Our application already has data, including the requirements of the public and the demand and supply of the entire country regarding charging stations, we implemented an additional application that can be used to make the process of the creation of a new charging station easier.

7.2 Algorithm Implementation (For Future Charging Stations)

Like user app, google maps was integrated in the admin app as well to provide real-time data for the areas where maximum requests are being made. The following simplified steps are used by the back-end of application to find places for future charging stations. An area of 1 km has been used:

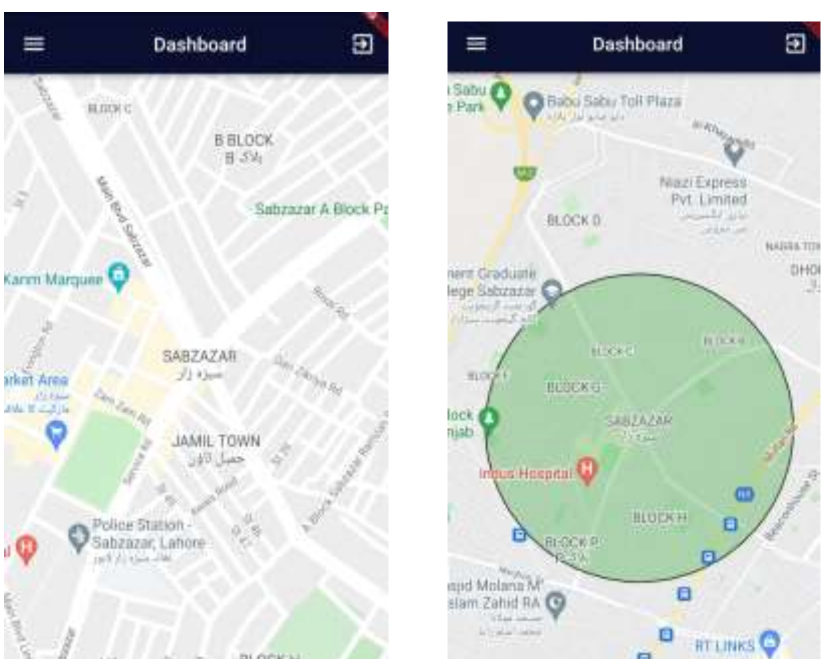

Figure 48: Charging Station Request Generation

1- The initial variable is set to 0 for the specific longitude and latitude. The number of requests is input to this variable and as this variable reaching a certain number ‘a’, the area on map turns green. Variable ‘a’ is set as 5 for testing.

Figure 49: Number of Requests exceed set threshold

2- As more requests are made, the value of ‘a’ increases. A certain threshold is set i.e. 5 during testing, after which the 5th request will turn the area of map as red.

3- The region for the requests is set as a scalable variable as well, i.e. 1 km for testing. Only requests within this radius will be counted towards the variable ‘a’. When the station is added to the indicated area, the app will notify ‘station added successfully’.

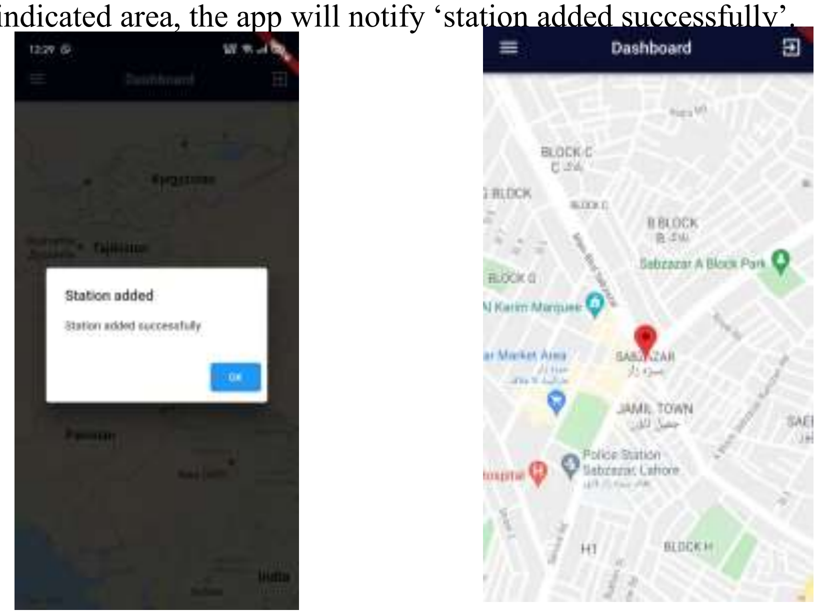

Figure 50: New Charging Station Deployed

Chapter 8: Super Admin Web Portal

The user and admin mobile apps are for our customers to use, but how do we maintain the apps we have created? This is where the Super Admin Web Portal comes in, it was built with the idea of app maintenance in mind. This is where we as a company operating this entire process can create, read, update, and delete our users, stations, cars, and station deployment requests.

CRUD: Create, read, update, and delete (CRUD) are the four basic operations of persistent storage in computer programming. CRUD is also a term for user interface patterns that make it easier to browse, search, and change data using computer-based forms and reports.

Data can be stored in a storage mechanism's location/area.

A storage location's most important attribute is that its material is both readable and updatable.

A storage location must be created before it can be read or altered; that is, it must be allocated and initialized with content. The storage location may need to be destructed (that is, completed and deallocated) later. The core activities of storage management, known as CRUD: Create, Read, Update, and Delete, are made up of these four actions. Below is the startup screen of our web admin panel, the admin enters the password and gains access to the portal. The password is authenticated against our database.

Figure 51: Admin Web Portal Login

Figure 52: Admin Web Portal Dashboard

Above is the dashboard of the web portal. It displays the total number of stations, users, and cars that are present in our database. A donut chart also helps represent the total number of available and allocated charging stations in real time. The station panel presents all the stations from our database, their allocation, longitudes and latitudes, the charger types available, the queue number and queue time at each station. All this data can have CRUD operations performed on it.

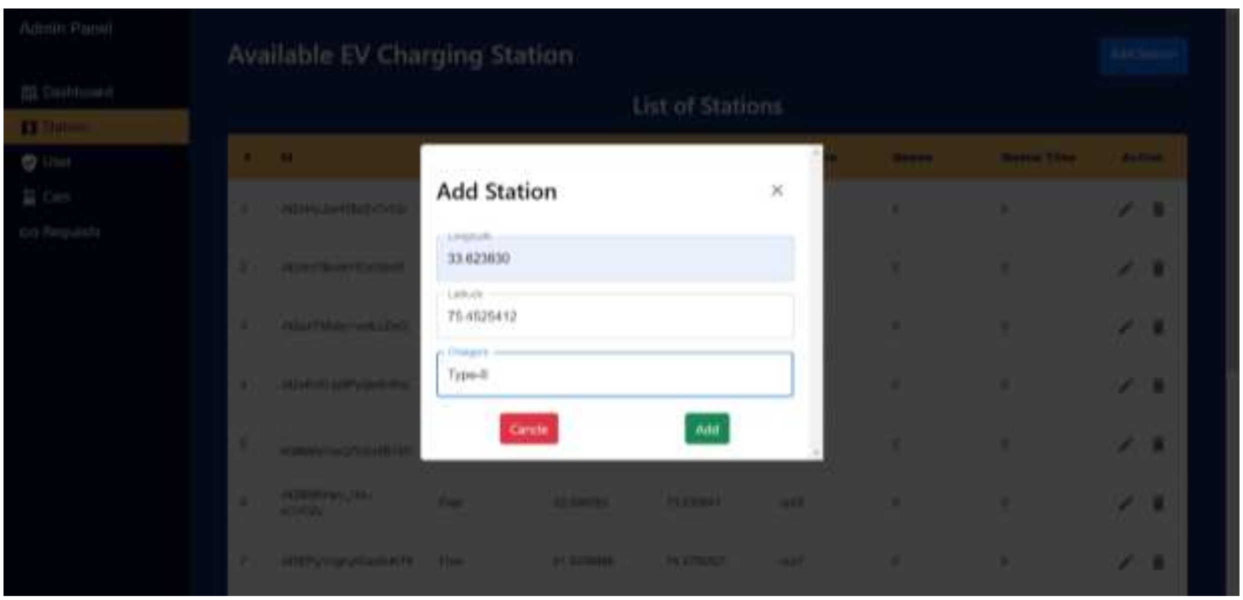

Figure 53: Adding a New Charging Station in our Database

We can add new charging stations as required, their long lats and charger type is to be entered.

The user panel presents all the users from our database, their names and phone numbers are fetched. The admin can delete a user if or when required but cannot add one because user registration is only possible through the user mobile app.

The cars section presents our entire collection of EVs. Each car's company name, model, charger port type, efficiency and max range are available. All this data can be edited, or deleted, whenever required.

We can add new EVs as required, their body type, charger port type, company name, model, efficiency, and max range are to be entered.

Figure 54: Web Portal Charging Station Request Center

The charging station deployment requests panel presents to the admin the total number of requests generated in an area, we can edit/delete this data for the sake of creating dummy data and make it presentable as a prototype. A representation of adding a request for the sake of creating dummy data.

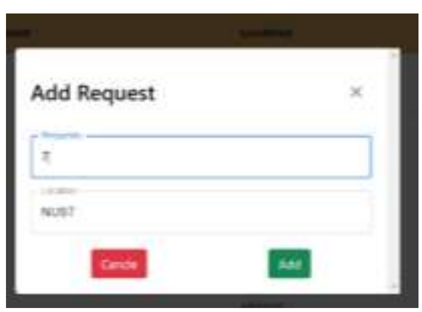

Figure 55: Adding a New Charging Station Request

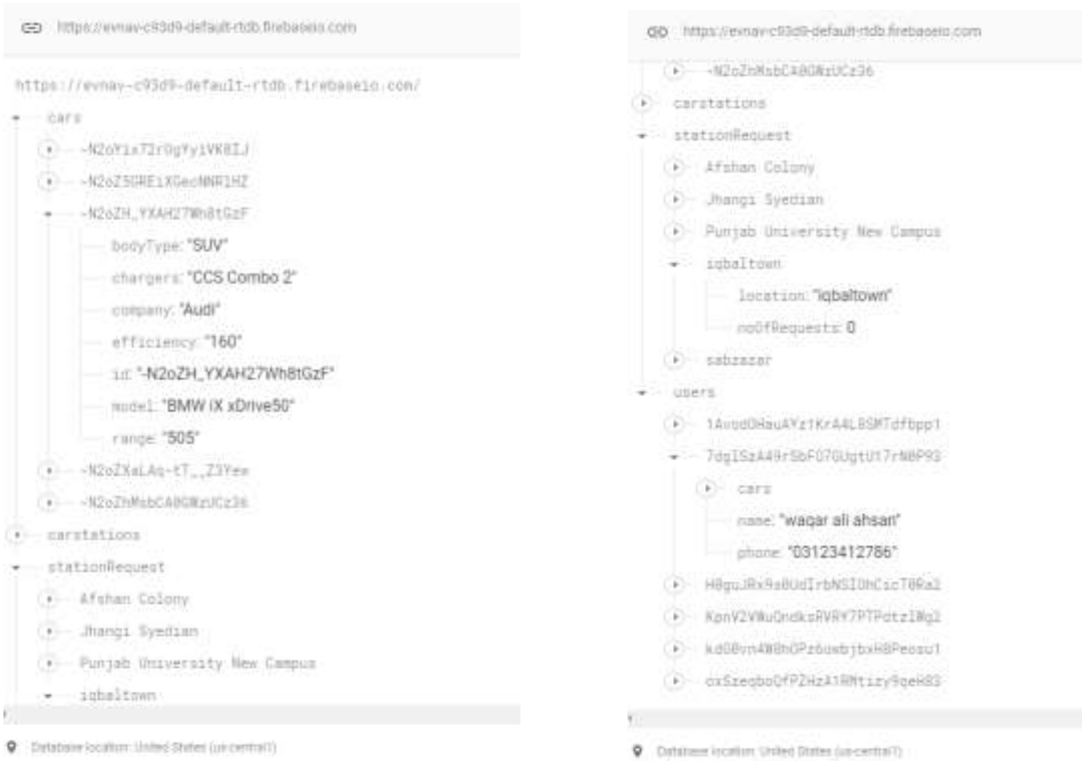

Figure 56: Database Collections


```json
"stationRequest": {
  "Afshan Colony": {
    "location": "Afshan Colony",
    "noOfRequests": 0,
    "requestUsers": [
      "n1p98uDxBya8UzCDxe5evape0Yz2",
      "DsRARL45W8gZA0ECE7uyaDFSvmN2",
      "IOwzcInCHBZbbHrGRiJ5KxUZsKs1"
    ],
    "totalNoRequest": 1
  },
  "Jhangi Syedian": {
    "location": "Jhangi Syedian",
    "noOfRequests": 0,
    "requestUsers": [
      "Q8mAJ1tnSxOJGttbVaBb6qT1DyG3"
    ],
    "totalNoRequest": 0
  },
```


Figure 57: Nested Objects (JSON)

All the relations of data in our collections are being stored in the form of objects, each object is nested and is being accessed through a unique ID.

Chapter 9: Future Scope and Development

With the growth of EV technology, the electric charging stations' management platform required continues development and innovation. In future work, we main to consider other design patterns for enabling a superior level of maintenance and growth of application based on blockchains such as security or utility tokens, wallets, and decentralized network interfaces. The scope of the software can be expanded to include development of a platform for electric vehicle charging stations that includes implementing communications between user and mobile app, including and improving the functionalities of payments, optimizing existing user paths, improving the multi-tenancy software architecture, and implementing a user management process. We can link with electric car charger manufacturers and provide them with a solution for ecosystem manageability for electromobility. Furthermore, we can add the following features to the application:

1- Boarding to help users through the sign-up process during the initial start of app.
2- Include a "plan your trip" feature that allows users to book charging stations in advance to save time by suggestions of on-route charging stations for longer trips.
3- Display facilities in nearby areas for the user to do something during the charging of their vehicle.
4- Include a customer service bot to help users. The benefits by this feature are illustrated below:

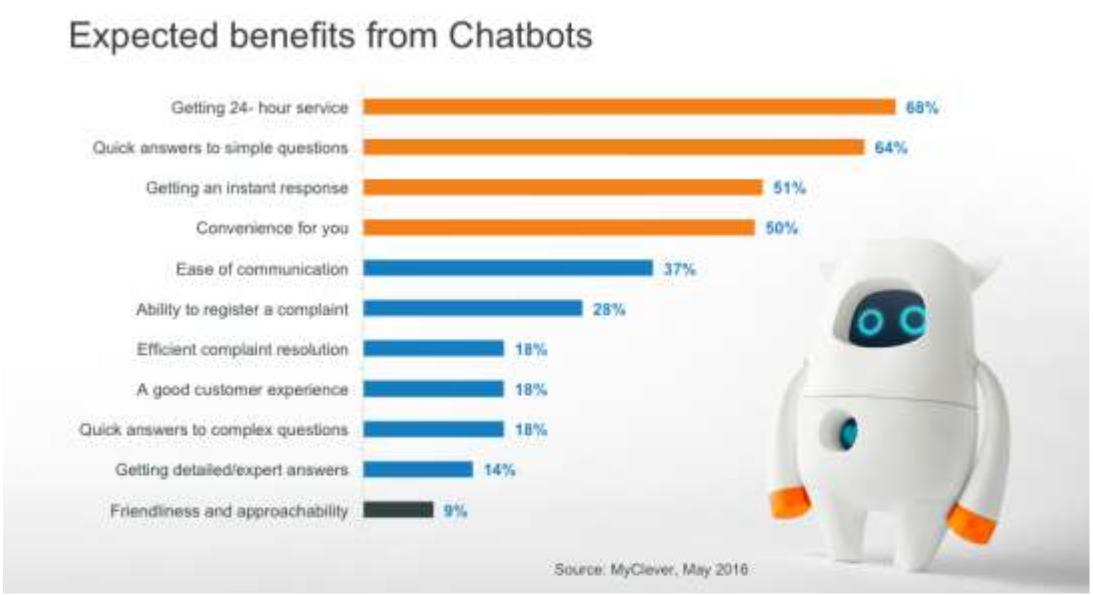

Figure 58: Expected benefits from chatbots

Moreover, we aim to continue our work on electric vehicle charging management with its impact towards protocols of communications and the designing of a prediction model for EV charging in the events of a journey. With the ability to auto-scale, we aim to improve our software for rapid updating without down-time and being resilient and reliable with a heavy workload. We can also include remote access for the users to allow them to access the status of their charger. This will allow users to control their charger as well i.e. switch it on/off remotely.

Furthermore, we will use the current model of the design pattern as the bases for other complex projects. Our design serves as an important tool in the art of software development. Our smart EV charging navigation can become a low-cost SaaS (software as a service) solution that provides a way for a user to enlist as a tenant or a business and use the platform service for their own charging stations; a single database for several clients or tenants, using the same software and containers for the deployment of solution to provide availability, scalability, and reliability.

References

[1] O. Ekren, C. H. Canbaz and C. B. Guvel, "Sizing of a solar-wind hybrid electric vehicle charging station by using HOMER software," *Journal of Cleaner Production,* vol. 279, 2021.

[2] J. Lee, G. Park, Y. Han and S. Yoo, "Big data analysis for an electric vehicle charging infrastructure using open data and software," *Jeju National University,* 2017.

[3] N. Rietmann and T. Lieven , "A comparison of policy measures promoting electric vehicles in 20 countries," *The Governance of Smart Transportation Systems,* pp. 125-145, 2019.

[4] A. Poran and L. Tartakovsky, "Energy efficiency of a direct-injection internal combustion engine with high-pressure methanol steam reforming," *Energy,* pp. 506-514, 2015.

[5] J. Rizvi, "85 locations identified for EV charging stations at motorways," THE NEWS, 2022.

[6] T. Ashraf, "The Future of Electric Vehicles (EVs) in Pakistan: From 2021-2026," OLX, 2021.

[7] T. N. International, "Malaysian automaker Proton plans to introduce electric cars in Pakistan," 2018.

[8] ProPakistani, "China's largest electric vehicle manufacturer is coming to Pakistan," 2019.

[9] "Future of Electric Vehicles in Pakistan," Techjuice, 2018.

[10] "Future of Electric Vehicles in Pakistan," PAGE, 2018.

[11] M. Hahsler, "Creational Pattern," 2020.

[12] M. Spasojevic, "Repository Pattern," ASP.NET, 7 March 2022. [Online]. Available: https://code-maze.com/net-core-web-development-part4/.

[13] L. Luo, W. Gu, S. Zhou, H. Huang and S. Gao, "Optimal planning of electric vehicle charging stations comprising multi-types of charging facilities," *Applied Energy*, pp. 1087-1099, 2018.

[14] A. Nayak, "EV Charging Station Finder App: Types, Features, Benefits and Development Cost," *Bacancy*, 19 May 2022.

[15] M. Nicolas, S. Hall and N. Lutsey, "Quantifying the electric vehicle charging infrastructure gap across US markets," *International Council on Clean Energy*, 2019.

[16] L. Jia, Z. Hu, W. Liang and W. Lang, "A novel approach for urban electric vehicle charging facility planning considering combination of slow and fast charging," *International Conference on Power System Technology*, 2014.

[17] W. Liu, M. Zhang, B. Zeng and L. Wu, "Analyzing the impacts of electric vehicle charging on distribution system reliability," *IEEE PES Innovation Smart Grid Technologies*, 2012.

[18] A. Poullikkas, "Sustainable options for electric vehicle technologies," *Renewable and Sustainable Energy Reviews*, 2015.

[19] "Electric Vehicle Basics," *California Air Resources Board.*