Software requirements and Documentation

Project Name: EV Mileage Estimator (Travel Mate)
Version: 1.0
Authors: Sienna, Sherab and Karolina

1. Requirements

1.1 Business Requirements

The EV Mileage Estimator app is designed to help electric vehicle drivers plan their trips more effectively by providing real-time travel range estimates, personalized route suggestions, and access to nearby charging stations. The application enables users to select their desired route and return an estimation of the battery needed to cover the route. Additionally, the app tracks battery usage and offers insights into the user's charging behavior, where for example the frequency of charging or the tendency to charge to full battery can be displayed. It also allows users to configure their profile by showing their car model, favorited charging stations and preferred route paths to take.

The main stakeholders include electric vehicle drivers, who are the primary users of the application. Developers and designers are responsible for building and maintaining the system. In some cases, charging station providers may be considered stakeholders if they contribute data or offer partnerships. This is also applicable to EV brands who wish to improve their customer experience. The target users for this platform are people who own electric vehicles and frequently plan long or unfamiliar trips, or who feel currently dissatisfied with the mileage predictions and planning their current resources offer. The app may be attractive to a range of demographics, such as tech enthusiasts or environmentally conscious consumers.

1.2 Objectives and Features

The core objectives of this project is to provide a more usable and accurate alternative for users to prepare their travel by providing a more accurate mileage estimation and a variety of routes based on preferences. Expanding from this, the app will also provide efficient routes in accordance to charging stations and weather conditions/temperature as well as any other factors affecting battery life. Once the app is built, users will be able to plan trips, view their remaining mileage, locate nearby charging stations, save frequently visited stations, and receive data-driven suggestions based on their past charging behavior.

1.3 Data Requirements

The application requires input data such as the user's starting point, final destination, vehicle model, and current battery percentage. For the start of the project, we will be assuming the user is driving a Tesla Model S3. It may also learn from the user's past behavior, including driving style and route choices, which could potentially affect battery life and consider this behavior when planning routes. The system will use MapKit for SwiftUI to access data about available charging stations (Apple Maps Data). As output, the app will generate travel range estimates, charging station recommendations,and trip summaries. It will also provide insights into energy consumption patterns and battery performance over time.

2. System Design

The design will firstly be tailored for the App Store, using Swift for the latest version of iPhone. This is only a starting point to get a product, which we will build off from and use the strong points to create a uniform app designed for all operating systems.

While we initially started using Bubble, we received a lot of technical difficulties and so, due to time constraints, we coded from scratch a product that could be interactable.

For future versions, we aim to create a more robust backend by using the Tesla APIs for battery life, as well as public APIs to gain access to local weather/temperature, up to date charging locations and traffic conditions. With initial use, a database will be constructed and expanded upon based on the user's interaction with the app. This may include the charging times, battery degradation, trip history which the user can retrieve to rerun the same route. This database will be used to train a model to increase the accuracy of prediction of energy consumption and future charging needs.

3. Actors and Use Cases

There are three main actors in the system: the user, the admin, and the system itself. The user is an electric vehicle driver who interacts with the application to plan trips and monitor battery usage. The admin manages the backend, handles user data, and ensures the application is functioning correctly by regularly updating in accordance to the car's model. The system processes data, predicts energy usage, and communicates with APIs.

In a typical use case, the user enters the app, where we assume they are already logged in and synced with their car. After logging in, the user can input their start and end locations to measure how much battery coverage would be needed to complete the trip. The system then calculates a route that includes charging stations if needed. Another use case allows the user to view their past trips including the distance traveled, and their power usage over the past 24 hours, last 3 days or the past week. The user can also save a favorite charging station to their profile. The admin may access the system to manage users and update the database.

Actor	Action	System Response
User	Login	Authenticates
User	Check mileage Inputs Start and Destination	Outputs battery coverage estimation
User	Go to Power Page View Charging history	Fetches past charging data Outputs a trend in a graph
User	Plan Route	Combines mileage + chargers
Admin	Manage Users Improve model accuracy by inputting more data	View reports, flag issues
System	Update predictions	Run ML models periodically, increasing in complexity with more data and runs

4. Functional Workflows

To view battery usage history, the user first opens the app and navigates to their Power & Battery section. Then they select the battery history time option, which displays a graph showing how their battery level changed over time. It also includes information on how often the user charged their vehicle and how much energy was used per trip.

The home page displays the user's current estimated travel range based on its starting and end point. It includes for now a drop down menu of cities to travel to which will be developed further into geolocalisation to increase accuracy of pinpointing the user exactly desired locations. If necessary, the app may recommend charging stations on the way in order for the user to safely complete their trip.

The MapView provides a map of the charging stations within the area of the user. For future development, the user will be able to add new stations, favorite which stations they prefer using, and perhaps leave feedback/reviews for the practicality of this station. The station page will be a way to connect users together to offer their input in their cities' EV usability. The page will also have filters if users wish to find fast charging or regular charging spots. Tapping on a station, the user can view its details such as availability, rating, connector types and more.

The user profile page stores important information about the vehicle, including battery capacity and charging efficiency, range and charging port type. The page also allows customization for the app, if the user wishes to receive a notification that their charging is complete and if there are

nearby stations near them. A payment method is logged in, and its security will be developed for future versions. The payment method embedded in the app will increase efficiency for the user to allow them to be hassle free. Users can edit their vehicle information and update their preferences at any time.

In the route planning feature, the user first enters a starting point and destination. The system takes into account various factors such as terrain, weather, and traffic to suggest an optimal route. It also analyzes the battery status and vehicle efficiency to decide if and where the user should stop to charge. Finally, the app displays a few route options, and the user can choose the fastest, most scenic, or most energy efficient path.

5. Testing Plan

The app will go through several phases of testing. Unit testing will be used to check each individual feature, such as user login or the route planner. Integration testing will ensure that the frontend communicates correctly with the backend and external APIs. System testing will involve testing the entire app from start to finish to make sure everything works as expected. Finally, user testing will involve real users trying out the app and giving feedback to improve its design and functionality. These tests will be conducted periodically to prevent any bugs from appearing, ensuring the user's data is always secure and used appropriately

6. Security Plan

To ensure user safety and privacy, all passwords will be stored in encrypted form and require authentication tokens to strengthen the system. Sensitive information, such as user location and vehicle data, will be stored securely. Users will have control over what data they wish to share with the app. The system will follow privacy regulations such as the General Data Protection Regulation (GDPR) to protect user rights and information. This will be expanded as the app is developed further since this is currently a prototype.

7. Validation and Verification

Validation will involve checking whether the application meets the needs of real users by reviewing the documentation and seeing if the product has been correctly implemented. This will be done through surveys, interviews, and feedback from beta testers. For example, users will be asked whether the app helped them find charging stations easily or whether the estimated mileage was accurate.

Verification will involve technical checks to make sure the app works correctly. This includes comparing expected results with actual outputs, checking for bugs, and reviewing whether each feature performs as intended. Penetration testing may also be practiced to ensure the user's data is difficult to retrieve and to find any clear holes in the system.

8. Deployment Plan

The app will first be released as a beta version in a limited area or among selected users. During this phase, feedback will be collected through in app surveys and system usage logs. Based on this feedback, the development team will fix any issues, improve performance, and enhance usability. Once the necessary changes have been made, a full version of the app will be launched to a wider audience, firstly to Apple products, and as the app develops it will be available to all Operating Systems and car models.

9. Maintenance Plan

After deployment, the app will be updated regularly. Charging station data will be refreshed to reflect real time availability. Support for new electric vehicle models and charging systems will be added as needed. Any bugs reported by users will be addressed promptly. The machine learning model that predicts battery usage will also be improved over time as more data is collected.