Bilkent University

CS491 Senior Design Project

E-Way

Project Specification Document

T2511

Furkan Özek, Hüseyin Utku Yüksel, Aziz Üzümcü, Halis Vefa Türkyılmaz

Supervisor: Salih Özgür Öğüz
Innovation Expert: Erkut Alakuş

1. Introduction

1.1 Description

The **E-Way Project** is an intelligent route planning system developed to assist electric vehicle (EV) users by predicting charging station occupancy and optimizing travel routes accordingly. The main objective of the project is to collect, store, and analyze historical occupancy data from EV charging stations across Türkiye, and use this information to estimate current and near-future station availability through artificial intelligence models.

The system continuously retrieves station data from public sources (such as the EPDK EV Charging Network) and stores it in a structured database [3]. Over time, this creates a rich dataset containing patterns such as hourly, daily, and seasonal station usage trends. Using these historical datasets, a machine learning model is trained to predict station occupancy levels based on factors like time of day, day of the week, location, operator, plug type, and station capacity.

When a user initiates a trip, E-Way calculates multiple possible routes and incorporates the AI-based occupancy predictions to determine which charging stations are likely to be available, busy, or congested at the expected arrival time. The system then recommends the most efficient route by avoiding stations with high predicted occupancy and suggesting alternatives with higher availability and shorter waiting times.

In addition to predictive routing, the application aims to enhance user comfort by providing station details, socket types, charging power, and real-time updates when available. The system also highlights **nearby amenities around charging stations**, such as cafés, restaurants, markets, shopping areas, and rest zones. Users can **predefine their preferences**—for example, “prefer stations near a café” or “avoid crowded commercial zones”—and the route generation algorithm adjusts accordingly. Even without predefined preferences, the application displays surrounding amenities during route planning so users can choose the most comfortable stop.

As the system evolves, it will also support more advanced personalization options and integrate additional contextual data such as weather, traffic density, and regional charging demand trends.

Overall, E-Way is designed to make EV travel **smarter, more predictable, and more efficient** by combining historical data analysis, machine learning, user preferences, and intelligent route optimization.

1.2 High-Level System Architecture & Components of Proposed Solution

The overall architecture of the E-Way system is designed as a modular, service-oriented, and scalable multi-layer structure.
It consists of five major layers: **Client Layer**, **Communication Layer**, **Backend Layer**, **Logic Layer**, and **Storage Layer**, all of which interact with external data sources.
Each layer fulfills a specific responsibility in the system and collectively enables intelligent route planning, real-time station monitoring, and occupancy prediction.

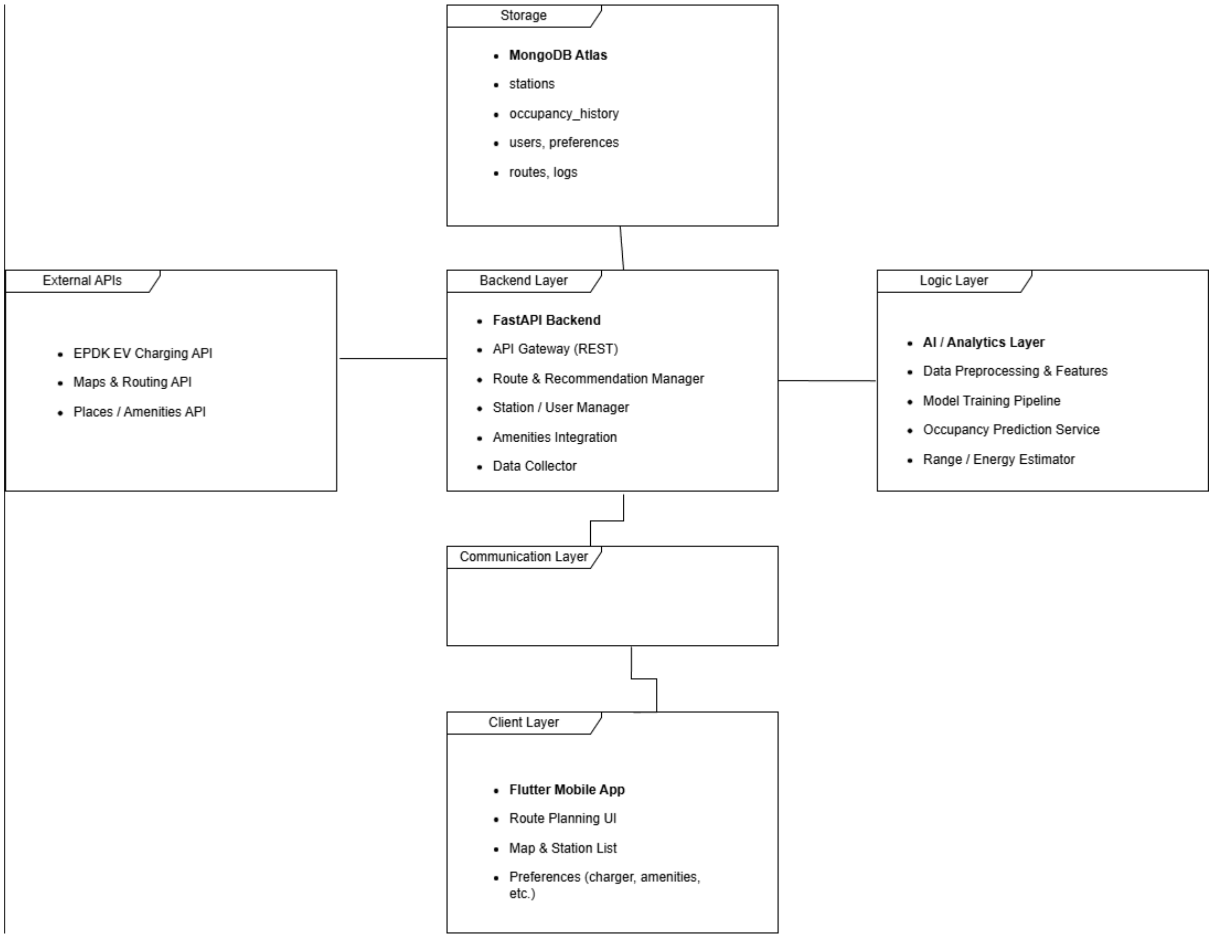

Figure 1. High-level system architecture of the E-Way application, showing the client, communication, backend, logic, storage layers and their interactions with external services.

1.2.1 Client Layer

The **Client Layer** represents the Flutter-based mobile application that runs on Android and iOS devices.
This layer serves as the primary interface between the user and the system.

Responsibilities:

- Provides a user-friendly interface for **route planning**, **map visualizations**, and **station browsing**.
- Allows users to enter **vehicle information**, **battery percentage**, and **travel preferences**.
- Displays AI-driven **occupancy predictions**, recommended charging stops, and nearby amenities.
- Handles user-specific settings such as preferred plug types, amenity preferences (cafe, market, mall, etc.), and route customization.

This layer communicates exclusively with the backend through secure API requests routed via the Communication Layer.

1.2.2 Communication Layer

The **Communication Layer** manages the secure exchange of data between the client application and the backend services.

Responsibilities:

- Ensures HTTP/HTTPS communication over REST APIs.
- Handles request/response formatting (JSON).
- Manages authentication headers and authorization tokens when required.
- Provides a clear separation between the user interface and backend logic.
- Allows the mobile client to remain lightweight while delegating heavy processing to the backend.

This layer ensures consistent, safe, and efficient data transmission throughout the system.

1.2.3 Backend Layer

The **Backend Layer** is implemented using **FastAPI**, serving as the central hub for all system operations.
It coordinates data flow between the client, logic layer, database, and external APIs [7].

Responsibilities:

- Provides a unified **REST API Gateway** for the mobile client.
- Executes **route generation** and integrates with external routing services (e.g., Google Maps API).
- Manages user accounts, vehicle details, and preferred charging behavior.
- Integrates with the **Data Collector**, which periodically retrieves charging station data (locations, socket types, power levels, etc.) from EPDK's EV Charging Network API.
- Communicates with the AI layer to obtain occupancy predictions.
- Returns optimized and personalized route results to users.

This layer acts as the main controller of the entire E-Way system.

1.2.4 Logic Layer (AI & Analytics)

The **Logic Layer** contains the system's intelligence components, including predictive analytics, AI models, and processing utilities.

Responsibilities:

- Performs **data preprocessing**, feature extraction, and dataset cleaning.
- Trains machine learning models using the historical occupancy dataset.
- Predicts **station availability**, **congestion levels**, and **charging duration estimations** based on:
  - time of day
  - day of week
  - location
  - operator type
  - plug type
  - seasonal usage trends
  - historical occupancy curves
- Acts as a decision-support module for route optimization.
- May expand to include weather data, real-time traffic, and regional demand forecasting in future versions.

This makes the system proactive rather than reactive, improving reliability and user comfort.

1.2.5 Storage Layer

The **Storage Layer** uses **MongoDB Atlas** as the cloud database for all system data.

Stored Data Includes:

- **Charging Stations:**
  Station number, name, operator, location, socket types, power levels, city, district.
- **Historical Occupancy Records:**
  Used for AI model training and pattern recognition.
- **User Profiles:**
  Vehicle type, preferred plug, amenity preferences, favorite routes.
- **Routing Logs:**
  Past trips, predictions, and performance metrics for model improvement.

By hosting the data on a scalable cloud platform, the system ensures high availability, durability, and global accessibility.

1.2.6 External APIs

The system integrates several third-party data sources to enhance accuracy and functionality.

Included External Services:

- **EPDK EV Charging API**
  Provides Türkiye’s official station, socket, and operator data.
- **Routing & Maps API**
  Supplies directions, travel times, distances, and map layers.
- **Places/Amenities API**
  Retrieves nearby facilities such as cafes, markets, rest areas, malls, and shops.

These integrations allow E-Way to deliver rich contextual information to users.

1.3 Constraints

1.3.1 Implementation Constraints

Dependence on External APIs

The system relies heavily on third-party services such as:

- EPDK EV Charging Stations API
- Google Maps / Routing API
- Google Places / Amenities API

Any changes in API structure, downtime, or rate limits directly affect the functionality of the system.

• Real-Time Data Limitations

EPDK does not currently provide **real-time occupancy data**.
Therefore, the prediction model must operate using **historical occupancy patterns**, which limits absolute accuracy.

• Mobile Device Constraints

The mobile app cannot perform heavy computation.
AI predictions, route generation, and preprocessing must be executed on the backend.

1.3.2 Economic Constraints

• Limited Cloud Budget

MongoDB Atlas Free Tier provides restricted:

- storage
- bandwidth
- performance

This limits how much historical data can be stored and how frequently updates can be retrieved.

• API Usage Costs

Some third-party services (especially Google Maps, Places, Routes) operate on a pay-per-use model.
To avoid exceeding quotas:

- requests must be minimized
- caching strategies must be used
- route calculations should be optimized

1.3.3 Ethical Constraints

• User Privacy & Data Protection

The system handles sensitive information such as:

- user location
- travel history
- vehicle type
- route preferences

This requires responsible handling, anonymization, and secure storage.
No personal data may be shared with third parties.

• Fair Use of Public Data

EPDK station data is government-owned.
It must be:

- used only for non-commercial academic purposes
- stored and accessed according to public API terms
- not altered in misleading ways

1.3.4 Environmental and Sustainability Constraints

The project is designed to promote environmental sustainability by encouraging the use of Electric Vehicles (EVs). However, the system itself consumes computational resources (cloud servers, database operations). To mitigate the carbon footprint of the digital infrastructure, the backend algorithms are optimized for efficiency to minimize unnecessary processing power and energy consumption.

1.3.5 Social and Political Constraints

- **Social:** The system relies on the digital literacy of EV drivers. The user interface must be accessible to a wide demographic to prevent exclusion based on technical skills.
- **Political:** The project is dependent on government regulations regarding energy data transparency. Changes in EPDK regulations or political decisions concerning energy distribution data could impact the availability of the public charging station API.

1.3.6 Health and Safety Constraints

Since E-Way is a mobile application used potentially while driving, safety is a primary constraint. The application must strictly adhere to driver safety guidelines:

- It must not require complex interactions while the vehicle is in motion.
- Notifications and alerts must be designed to be non-intrusive to avoid distracting the driver.
- Route recommendations must prioritize safe roads and avoid suggesting stops in hazardous locations if such data becomes available.

1.4 Professional and Ethical Issues

User Privacy & Data Protection:

E-Way processes potentially sensitive user information such as preferred routes, travel habits, frequently visited locations, charging behavior, and EV usage patterns. This data represents part of the user's personal mobility profile and must be protected. All user-related data (e.g., login credentials, preferences, historical route logs) will be stored securely, and sensitive fields such as passwords will be hashed. No personal data will be shared with third parties without user consent.

Location Data Sensitivity:

Since E-Way involves real-time geolocation, route tracking, and station proximity analysis, location-based data is highly sensitive. Users must remain fully informed about what location data is collected, how it is stored, and how long it is retained. Users will be provided transparency and explicit control to disable certain data collection features (e.g., occupancy tracking or route history) at any time.

Responsible Use of External Data Sources:

The project relies on external public data sources (e.g., EPDK EV Charging Network Data, map/places APIs). Only publicly accessible or licensed datasets will be used, and the system will not use any scraping or data-collection method that violates API terms of service. The retrieved data will be used strictly for route optimization and prediction, and not repurposed for commercial resale.

Fair and Unbiased AI Predictions:

The occupancy prediction model is trained on historical data collected from charging stations across Türkiye. Since machine learning models may unintentionally embed biases (e.g., geographic inequality, overemphasis on major cities), we will continuously evaluate model outputs to detect and minimize biases. The prediction model will avoid favoring specific brands, operators, or regions unfairly and will instead provide neutral, data-driven recommendations.

Transparency of AI-Driven Suggestions:
Users will be informed when a recommendation (such as avoiding a station or rerouting) is based on AI predictions rather than real-time availability. Clear explanations will be provided within the app to maintain transparency and user trust, ensuring that recommendations do not mislead or pressure users into certain route choices.

Ethical Handling of Charging Station Information:
Although charging station data is public, E-Way will not alter or misrepresent it. The app will avoid generating false impressions about station quality, pricing, or commercial superiority. E-Way's purpose is assistance—not influencing user decisions for the financial benefit of any specific charging network operator.

1.5 Standards

The development and documentation of the E-Way project adhere to widely accepted engineering and software standards to ensure quality, maintainability, and interoperability.

- **Documentation Standards:** The project requirements are documented following the **IEEE 830-1998** standard [1]. This structure ensures that functional and non-functional requirements are clearly defined and verifiable.
- **Modeling Standards:** The system architecture and design diagrams are created using **UML 2.5.1** (Unified Modeling Language) [2]. This includes use case diagrams, sequence diagrams, and class diagrams to visualize system interactions.
- **API Specifications:** The backend REST APIs are designed in accordance with the **OpenAPI Specification 3.0**. This ensures that the communication between the client (Flutter) and backend (FastAPI) is standardized and easily documentable.
- **Data Privacy:** The project complies with **KVKK (Kişisel Verilerin Korunması Kanunu)** and general **GDPR** principles regarding the storage and anonymization of user location and route data [4].
- **Ethical Standards:** The project adheres to the **IEEE Code of Ethics**, specifically focusing on the safety, health, and welfare of the public (drivers) and avoiding conflicts of interest in algorithm recommendations.

2. Design Requirements

2.1 Functional Requirements

FR-1: The system shall allow users to create an account using email and password (or optional Google/Apple login).

FR-2: The system shall allow users to set vehicle information (battery capacity, consumption rate, etc.).

FR-3: The system shall allow users to specify personal preferences such as:

- preferred charging networks (e-şarj, ZES, Voltrun, etc.)
- preferred socket type (AC Type-2, DC CCS, etc.)
- preferred amenities (café, rest area, shopping center, etc.)

FR-4: The system shall store and update user preferences in the database.
FR-5: The system shall allow users to enter a start point and destination to generate a route.

FR-6: The system shall retrieve available charging stations along the route using backend APIs.

FR-7: The system shall calculate multiple route alternatives based on:

- shortest travel time
- lowest cost
- comfort-oriented (amenities-based)
- minimum waiting time (AI-based)

FR-8: The system shall display the recommended route on a map interface.

FR-9: The system shall allow users to view charging stations on the map or as a list.

FR-10: The system shall display station details, including:

- station name & operator
- power (kW) and socket types
- number of total sockets
- address, city, district
- predicted occupancy percentage
- nearby amenities (cafés, restaurants, markets, etc.)

FR-11: The system shall filter stations based on:

- city/district
- network operator
- socket type
- AC/DC
- power rating

FR-12: The system shall retrieve historical occupancy data from MongoDB.

FR-13: The AI model shall predict expected occupancy for the time interval when the user will arrive.

FR-14: The system shall label stations as:

- **Available** (0–40% predicted occupancy)
- **Moderate** (40–70%)
- **Busy/Congested** (70–100%)

FR-15: The system shall integrate occupancy predictions into route optimization by avoiding congested stations and suggesting alternatives.

FR-16: The system shall connect to a places/amenities API (Google Places, OpenStreetMap, etc.).

FR-17: The system shall retrieve nearby amenities for each charging station.

FR-18: The system shall display amenity categories (café, WC, restaurant, etc.) during route selection.

FR-19: The backend shall fetch station data from the official EPDK Charging Station API.

FR-20: The backend shall update the MongoDB database with:

- station metadata
- socket information
- address mapping (city/district parsing)

FR-21: The system shall store user route history and station usage logs.

FR-22: The system shall notify the user when:

- predicted congestion exceeds a threshold
- a better/alternative station becomes available
- battery level becomes insufficient for the current route

FR-23: The system shall allow users to enable/disable notifications.

FR-24: The system shall authenticate users securely

.
FR-25: The system shall validate requests to ensure only authorized users access personal data.
FR-26: The system shall allow administrators to view database statistics:

- number of stations
- occupancy history size
- user accounts
- ML model performance metrics

FR-27: The system shall allow administrators to trigger a manual retraining of the AI model.

FR-28: The system shall provide meaningful error messages when APIs fail (e.g., EPDK API unavailable).

FR-30: The system shall retry essential data collection processes when errors occur.

2.2 Non-Functional Requirements

2.2.1 Usability

- The system shall provide an intuitive and user-friendly interface suitable for drivers who need quick access to route and station information.
- The mobile application shall clearly display routes, charging stations, socket details, and occupancy predictions with minimal interaction.
- Visual elements (colors, icons, occupancy indicators) shall be designed to be understandable at a glance, even while traveling.

2.2.2 Reliability

- The system shall reliably retrieve and present charging station data, even in cases of temporary API delays or failures.
- Critical backend services (routing, occupancy prediction, user data) shall ensure consistent availability with minimal downtime.
- Data synchronization processes (EPDK data fetch, database updates) shall fail safely and retry without corrupting stored information.
- The application shall handle unexpected errors gracefully without crashing.

2.2.3 Performance

- Database queries (e.g., filtering stations, fetching predictions) shall return results efficiently due to indexed collections and optimized data models.
- The system shall support real-time interactions, such as updating station predictions or recalculating routes during a trip.

2.2.4 Supportability

- The system shall be designed in a modular and maintainable structure so that future developers or team members can easily understand, update, and extend the codebase.
- The backend shall follow clean API design principles (RESTful endpoints, clear request/response schemas) to simplify debugging, testing, and integration with new features.

2.2.5 Scalability

- The backend architecture (MongoDB + FastAPI) shall support increasing numbers of users and stored historical occupancy records without noticeable performance degradation.
- The system design shall allow adding new data sources (more stations, new APIs, additional cities/countries) with minimal architectural changes.
- The AI prediction module shall be able to accommodate growth in dataset size and retrain on expanded historical data.

3. Feasibility Discussions

3.1. Market & Competitive Analysis

The rapid adoption of electric vehicles (EVs) in Türkiye and globally has created a growing demand for intelligent charging infrastructure and advanced route planning solutions. As EV usage increases, drivers face challenges such as station congestion, unpredictable availability, varying charging speeds, and limited visibility into station conditions along long-distance routes. E-Way aims to address these challenges by providing data-driven, predictive, and user-friendly navigation tailored specifically for EV drivers.

3.1.1 Market Overview

The EV market in Türkiye is expanding rapidly, supported by government incentives, increased charging networks, and declining EV costs.
Key market observations include:

- **Over 13,000+ registered charging stations** in Türkiye (and rising), many concentrated in metropolitan regions such as Istanbul, Ankara, and İzmir.
- **Widespread expansion of public and private charging networks**, supported by EPDK regulations and operator investments.
- **Increased need for smart charging solutions** as long-distance EV travel becomes more common.
- **Growing interest in predictive and AI-enhanced tools**, as drivers seek more reliable, real-time guidance to avoid delays.

3.1.2 Competitive Landscape

There are several existing platforms that offer basic station listings or simple route planning. However, most lack predictive analytics, personalization, or Türkiye-focused datasets.

1. Google Maps

- Provides EV-specific route planning and charging station locations.
- Does *not* offer detailed socket data, charging power, or historical occupancy analysis for Türkiye.
- Does *not* predict future station availability or recommend stations based on expected arrival time.

2. Eşarj, ZES, and Other Operator Apps

- Each operator provides its own mobile app showing stations and real-time occupancy.
- Coverage is **limited to their own network only** (not a unified national map).
- Lack AI-based prediction or cross-operator route optimization.

3. PlugShare

- Offers crowdsourced station information and user comments.
- Limited real-time accuracy and **no predictive occupancy** capabilities.
- Does not integrate route optimization effectively.

3.1.3 Competitive Advantages of E-Way

E-Way differentiates itself through several unique value propositions:

- **Türkiye-focused national coverage**, integrating data from EPDK and all major charging operators.
- **Historical occupancy dataset** built specifically for the local charging ecosystem.
- **AI-driven prediction engine** estimating station availability at the user's expected arrival time.
- **Personalized routing** based on user preferences (fastest, cheapest, least crowded, amenity-rich).
- **Nearby amenity suggestions** (cafés, shops, rest areas) to enhance user travel comfort.
- **Cross-platform accessibility**, supporting both Android and iOS via a single Flutter application [6].
- **Lightweight, efficient, and real-time backend architecture** optimized for large-scale data handling.

No existing system offers the full combination of **AI predictions**, **multi-operator national integration**, and **personalized routing** specifically for Türkiye.

3.1.4 Overall Market Positioning

E-Way positions itself as a comprehensive intelligent assistant for EV users, focusing on:

- Predictive route planning
- Cross-network charging visibility
- Comfort and convenience
- Locally optimized data-driven navigation

As EV adoption accelerates, platforms like E-Way will play a crucial role in mitigating range anxiety, enhancing charging efficiency, and promoting sustainable travel.

3.2. Academic Analysis

The E-Way project is based on key academic concepts from intelligent transportation systems, electric vehicle routing, machine learning, and geospatial analysis. These research areas form the scientific foundation of the system's predictive and routing capabilities [5].

3.2.1 Intelligent Transportation Systems (ITS)

ITS research focuses on using real-time data, analytics, and automation to improve mobility and travel efficiency.

E-Way aligns with ITS principles by combining station data, prediction models, and route decision-making to support EV drivers.

3.2.2 EV Routing & Charging Optimization

Academic studies highlight the unique challenges of EV routing such as limited range, charging time constraints, and uneven station distribution.

E-Way incorporates these findings by selecting optimal charging stops based on distance, power level, and predicted congestion.

3.2.3 Machine Learning for Demand Prediction

Time-series forecasting and occupancy prediction are well-studied areas.
Models such as regression, tree-based algorithms, or LSTM networks can estimate future demand based on past usage patterns.

E-Way applies these methods to predict station availability using historical occupancy trends.

3.2.4 Geospatial and Mapping Systems

GIS research provides methods for route calculation, spatial filtering, and nearest-station detection.

E-Way uses geospatial techniques to analyze charging locations, calculate routes, and integrate POIs (cafés, shops, rest areas).

Summary

E-Way is grounded in academic work across transportation systems, EV routing optimization, machine learning forecasting, GIS mapping, and user-centered interface design. These fields collectively support the system’s intelligent routing and occupancy prediction features.

5. Glossary

EV: Electric Vehicle
SoC: State of Charge
API: Application Programming Interface
UI: User Interface
ML: Machine Learning
AI: Artificial Intelligence
DB: Database
ETA: Estimated Time of Arrival
POI: Point of Interest
GIS: Geographic Information System
LSTM: Long Short-Term Memory (Neural Network Model)

6. References

[1] IEEE Computer Society, "IEEE Recommended Practice for Software Requirements Specifications," IEEE Std 830-1998, Oct. 1998.

[2] Object Management Group (OMG), "OMG Unified Modeling Language (OMG UML)," Version 2.5.1, Dec. 2017. [Online]. Available: https://www.omg.org/spec/UML/2.5.1

[3] Enerji Piyasası Düzenleme Kurumu (EPDK), "Şarj Hizmeti Yönetmeliği," Resmi Gazete, Sayı: 31797, Apr. 2022.

[4] T.C. Resmi Gazete, "Kişisel Verilerin Korunması Kanunu (Kanun No. 6698)," Apr. 2016.

[5] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle routing problem with time windows and recharging stations," *Transportation Science*, vol. 48, no. 4, pp. 500–520, 2014.

[6] Google Developers, "Flutter Documentation - Build apps for any screen," [Online]. Available: https://flutter.dev/docs

[7] Tiangolo, "FastAPI Framework," [Online]. Available: https://fastapi.tiangolo.com/