SE 3505 - MOBILE APPLICATION DEVELOPMENT

Report

ROUTE ASSISTANT

HADİSE

PREPARED BY :
Ömer Faruk Kaba - 220717044
Esra Yıldız - 220717015

1.ABSTRACT

Route Assistant is a mobile application developed to support users in discovering, creating, and exploring outdoor routes through an interactive map-based interface. The application is designed to provide a simple and user-friendly experience for individuals interested in activities such as hiking, walking, and route exploration. By focusing on usability and accessibility, Route Assistant aims to make route discovery easier for both casual and active users.

The application is developed using Flutter as the mobile framework and Dart as the programming language. Supabase is used as the backend service to handle authentication, database management, and server-side operations, while PostgreSQL is used as the database system. Google Maps SDK is integrated into the application to provide map visualization and location-based services.

Route Assistant allows users to register and log in securely, explore nearby routes based on their current location, view detailed route information on an interactive map, and share feedback through a commenting system. Throughout the development process, manual testing was performed to ensure the correctness and usability of core features. As a result, Route Assistant successfully meets its primary objectives and demonstrates a practical implementation of modern mobile application development concepts.

2.INTRODUCTION

In recent years, mobile applications have become an integral part of everyday life, providing solutions for navigation, communication, and information access. With the widespread use of smartphones, users increasingly rely on mobile applications to support outdoor activities such as walking, hiking, and route exploration. These applications help users navigate unfamiliar areas and plan their activities more efficiently.

Route-based mobile applications play an important role in enhancing user experience by offering map visualization, location awareness, and interactive features. However, many existing applications are either too complex for casual users or lack community-driven content that allows users to actively contribute and share their experiences. As a result, there is a need for simpler and more accessible applications that focus on usability and interaction.

Route Assistant was developed to address these identified needs by providing a clean and intuitive mobile application for discovering and sharing outdoor routes. The application emphasizes ease of use, interactive map-based navigation, and user participation. By combining modern mobile development technologies with cloud-based backend services, Route Assistant aims to deliver a practical solution for users seeking an efficient and user-friendly route discovery experience.

3.PROBLEM DEFINITION

With the increasing number of mobile navigation and route-based applications, users are presented with many alternatives. However, despite this variety, many existing applications do not fully meet the needs of casual and everyday users. Most route applications are designed for professional or advanced users, which often results in complex interfaces and overwhelming features.

One of the main problems is the difficulty users face when trying to discover routes near their current location. Many applications require multiple steps to access nearby route information or provide limited filtering options. Additionally, users often have limited opportunities to contribute their own routes, reducing community participation and content diversity.

Another important issue is the lack of user interaction and feedback mechanisms. In many applications, users cannot easily share experiences, leave comments, or provide suggestions for routes. This limits the usefulness of the application and reduces user engagement. Furthermore, some applications do not prioritize usability and accessibility, making them difficult to use for individuals with limited technical experience.

These problems highlight the need for a simple, accessible, and community-driven mobile application that focuses on usability, interaction, and location-based route discovery.

4.PROPOSED SOLUTION

To address the problems identified in the previous section, Route Assistant proposes a simple, user-centered mobile application focused on usability and accessibility. The main objective of the application is to allow users to discover, share, and interact with outdoor routes in an intuitive and efficient manner.

Route Assistant provides location-based route discovery by utilizing map services to display nearby routes according to the user's current location. This approach reduces the complexity of route searching and allows users to quickly access relevant route information. The application also enables users to view detailed route data through an interactive map interface, improving overall navigation and understanding.

In addition to route discovery, Route Assistant encourages community participation by allowing users to create and share their own routes. Users can interact with shared content through comments, providing feedback and sharing experiences with others. By combining a clean user interface with community-driven features, Route Assistant delivers a practical solution that directly addresses usability and interaction challenges found in existing applications.

5.TECHNOLOGIES USED

The development of the Route Assistant mobile application required the use of modern and reliable technologies to ensure performance, scalability, and ease of maintenance. The selected technologies were chosen based on their compatibility with mobile development requirements and their ability to support real-time and location-based features.

Flutter was used as the primary mobile development framework due to its cross-platform support and efficient UI rendering capabilities. By using Flutter, a single codebase can be maintained while achieving high performance on mobile devices. The application logic was implemented using the Dart programming language.

Supabase was selected as the backend service to manage authentication, database operations, and server-side logic. It provides a scalable and developer-friendly environment with built-in authentication and real-time capabilities. PostgreSQL was used as the database management system due to its reliability and support for structured relational data.

Google Maps SDK was integrated into the application to provide interactive map visualization and location-based services. This integration allows users to view routes on a map, explore nearby locations, and interact with route data in real time.

Layer	Technology
Mobile Framework	Flutter
Programming Language	Dart
Backend Service	Supabase
Database	PostgreSQL
Authentication	Supabase Authentication
Map Service	Google Maps SDK
API Communication	REST
Target Platform	Android

6.SYSTEM ARCHITECTURE

Route Assistant is designed using a client–server architecture to ensure scalability, maintainability, and clear separation of responsibilities between system components. The architecture consists of a mobile client, a backend service, and a database layer, supported by external services.

The mobile client, developed using Flutter, is responsible for handling user interactions, displaying the user interface, and managing map-based features. It communicates with the backend service to retrieve and store data such as user information, routes, and comments. The client also integrates Google Maps SDK to provide interactive map visualization and location-based functionality.

The backend service is powered by Supabase, which manages user authentication, database operations, and server-side logic. Supabase acts as an intermediary between the mobile client and the database, ensuring secure access to data and handling authorization processes. This structure simplifies backend development while maintaining security and reliability.

The database layer uses PostgreSQL to store structured data, including user profiles, route information, and user-generated content. PostgreSQL was chosen for its robustness and support for relational data. External services such as Google Maps API are integrated to provide real-time map rendering and location services. This architectural design allows Route Assistant to operate efficiently while remaining flexible for future extensions and feature enhancements.

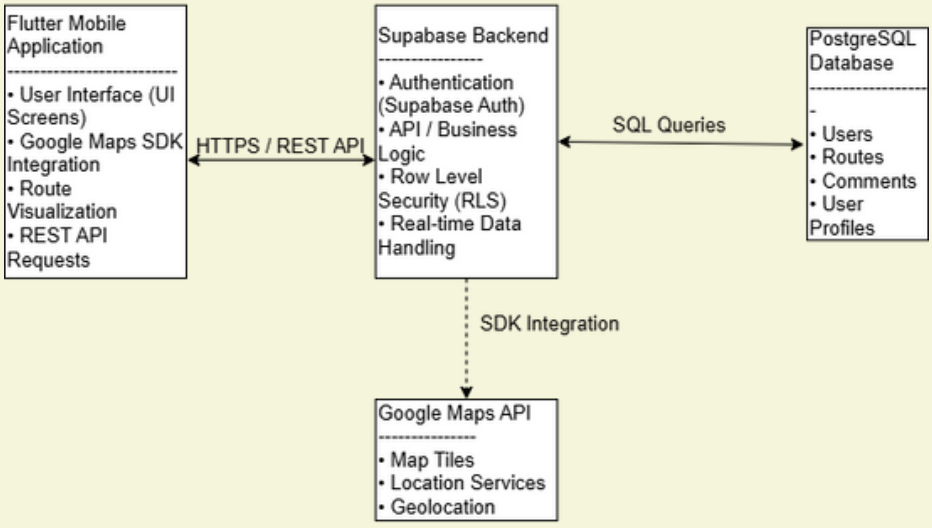


Figure 1. System architecture of the Route Assistant application

7. APPLICATION FEATURES

Route Assistant provides several core features designed to improve route discovery, navigation, and user interaction. These features focus on usability, accessibility, and community participation.

7.1 User Authentication

The application allows users to register and log in securely using email and password authentication. Supabase Authentication is used to manage user sessions and ensure secure access to application features. Only authenticated users are allowed to create routes and interact with content.

7.2 Route Discovery

Users can explore available routes through an interactive map interface. Routes are displayed based on the user's current location, allowing nearby routes to be discovered easily. This feature simplifies navigation and reduces the effort required to find relevant routes.

7.3 Route Details

Each route has a detailed view that presents route information along with map visualization. Users can view the route path, description, and additional details in a clear and structured format.

7.4 Route Creation

Route Assistant enables users to create and share their own routes. This feature encourages user-generated content and supports community-driven growth within the application.

7.5 Comment System

Users can interact with routes by leaving comments and sharing feedback. This feature allows users to exchange experiences and improves engagement within the platform.

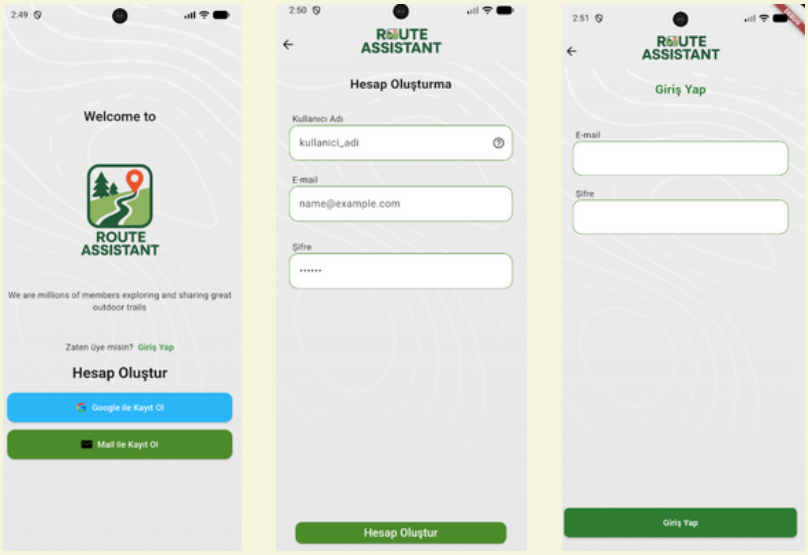
7.6 Profile Management

Users can view and manage their personal profiles. The profile section displays user information and the routes created by the user, providing a personalized experience.

8. APPLICATION SCREENSHOTS

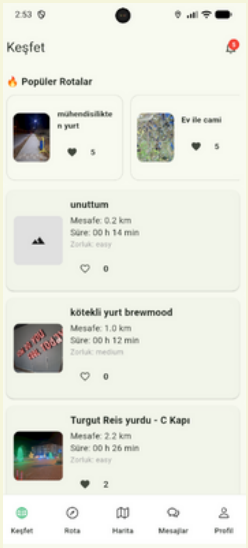
This section presents screenshots of the Route Assistant mobile application to demonstrate its main user interface and core functionalities. The screenshots provide visual evidence of the implemented features and support the explanations given in previous sections.

8.1 Login and Registration Screens



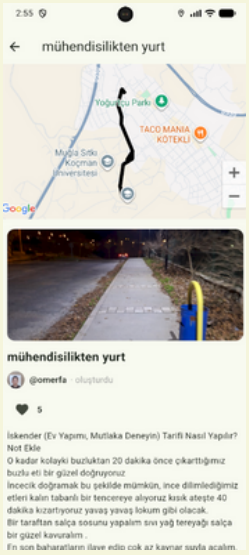
Login and registration screen of the Route Assistant application.
This screen allows users to securely register and log in using email and password authentication.

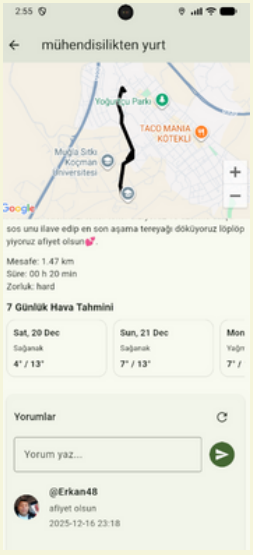
8.2 Home / Explore Screen



Home screen displaying available routes on an interactive map.
Users can explore routes and navigate through the map interface.

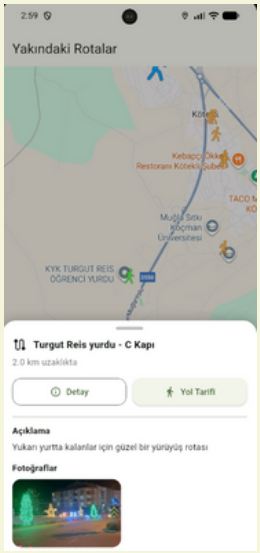
8.3 Route Detail Screen





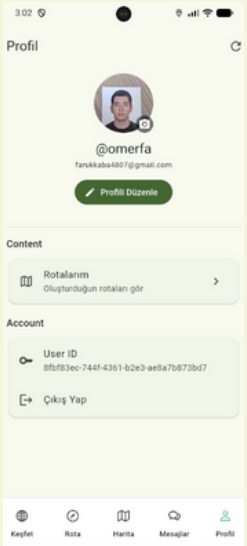
Route detail screen showing route information and map visualization.
This screen provides detailed route data, including the route path and description.

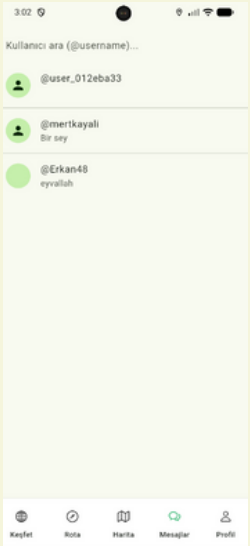
8.4 Login and Registration Screens



Nearby routes screen displaying routes based on the user's current location.
This screen allows users to easily discover routes located near them using location-based services.

8.5 Profile and Message Screens





The profile screen allows users to view and manage their personal information. It displays user details along with the routes created by the user, providing a personalized overview of their activity within the application.

The message screen allows users to exchange messages in real time. This feature supports communication and interaction between users, enhancing the social and collaborative aspects of the application.

9. TESTING

Testing was conducted to evaluate the functional correctness, performance, and stability of the Route Assistant application under different load conditions. Both functional checks and basic load testing scenarios were applied to observe system behavior during concurrent usage.

9.1 Test Scenarios

The following core scenarios were tested:

- User authentication and session handling
- Route retrieval and display
- Location-based route discovery
- Backend API responsiveness

Testing was performed using automated HTTP-based load testing tools and manual verification on Android devices.

9.2 Load Test Results

Three different test configurations were executed with varying numbers of virtual users (VUs) to simulate real-world usage conditions.

-Test Case 1: Low Load Scenario

- Total requests: 9,102
- Successful checks: 100%
- Failed checks: 0.00%
- Average response time: 166 ms
- 95th percentile response time (p95): 183 ms
- Maximum virtual users: 100

Under low load conditions, the application demonstrated stable performance with consistently low response times and no observed request failures.

-Test Case 2: Medium Load Scenario

- Total requests: 89,334
- Successful checks: 99.99%
- Failed checks: 0.00%
- Average response time: 253 ms
- 95th percentile response time (p95): 211 ms
- Maximum virtual users: 750

This scenario reflects a more realistic multi-user environment. The backend maintained high reliability and low latency, successfully meeting defined response time thresholds.

-Test Case 3: High Load Scenario

- Total requests: 224,614
- Successful checks: 92.37%
- Failed checks: 7.62%
- Average response time: 1.76 s
- 95th percentile response time (p95): 7.84 s
- Maximum virtual users: 5000

This test represents an extreme load scenario beyond typical classroom-scale usage. Although response times increased under heavy load, the system remained operational and continued to respond successfully to the majority of requests.

9.3 Evaluation

Based on the test results, Route Assistant performs reliably under low and medium load conditions, which are representative of expected real-world usage. While performance degradation was observed under extremely high load, the system continued functioning without critical failure. These results indicate that the backend architecture is suitable for small to medium-scale user environments and can be further optimized for higher scalability if required.

10. CONCLUSION

In this project, Route Assistant was developed as a mobile application aimed at simplifying route discovery and navigation through an interactive map-based interface. The application successfully integrates modern mobile development technologies to deliver a functional and user-friendly solution.

Throughout the development process, core features such as user authentication, route discovery, route creation, and user interaction were implemented and tested. Quantitative testing results demonstrated that the system performs reliably under low and medium load conditions, which are representative of expected real-world usage. Although performance degradation was observed under extreme load scenarios, the application remained operational and stable.

Overall, Route Assistant meets its primary objectives and provides a solid foundation for future enhancements. The project offered valuable experience in mobile application development, backend integration, and system architecture design. With further improvements such as offline map support, enhanced scalability, and additional social features, Route Assistant has strong potential for continued development.