POLITECNICO
MILANO 1863

COMPUTER SCIENCE AND ENGINEERING
SOFTWARE ENGINEERING II
2025 - 2026

DD

Design Document

Best Bike Paths

Authors:

Leonardo Guglielmi, Francesco Lo Conte

Version:

1.0
(January 2, 2026)

Contents

1 Introduction

1.1 Scope

1.1.1 Product domain

The scope of the project covers the users interacting with the Best Bike Paths (BBP) system, the data collection processes regarding cycling routes, and the navigation services provided to the community.

For the project, the following users interacting with the system have been identified:

- **Registered Users.**
- **Generic Users.**

Registered Users will be able to record their trips using the mobile application, tracking their performance statistics and path data. During the recording, they contribute to the system by collecting data either automatically (via device sensors) or manually (by reporting obstacles or status). Upon completion of a trip, they can review and confirm the detected anomalies, making them available to the community.

Generic Users (along with Registered Users) can access the platform to search for the best cycling routes between two locations. The system provides them with routes ranked by a "Path Score", calculated based on the aggregated data provided by the community, allowing them to visualize safety information and obstacles on the map.

The system acts as a mediator between the raw data collected from the real world (road conditions) and the end-users, processing this information to ensure safety and reliability.

1.1.2 Main architectural choices

The system is to be implemented using a **microservices-oriented architecture**. This choice is driven by the need for a scalable and maintainable system, capable of handling different loads on different components (e.g., the data ingestion service may face high traffic during weekends, while the reporting service might be less stressed).

This architecture allows for:

- **Independent Scaling:** Individual components can be scaled based on their specific resource requirements.
-
- **Resilience:** If a specific microservice fails (e.g., the weather enrichment service), the core functionality remains available.

- **Technology Agnosticism:** Different teams can develop different services using the most appropriate technologies for each task.

Furthermore, the system adopts a Client-Server model where the mobile application performs significant local processing (Edge Computing) to analyze sensor data before sending it to the backend, optimizing bandwidth and responsiveness.

1.2 Definitions, acronyms, abbreviations

This section contains the definitions for terms that may be technical or specific to the architecture, as well as acronyms and abbreviations used throughout the document.

1.2.1 Definitions

- **Microservice:** A software development technique that arranges an application as a collection of loosely coupled services.
- **API Gateway:** A server that acts as an API front-end, receiving API requests, enforcing throttling and security policies, passing requests to the back-end service and then passing the response back to the requester.
- **Edge Computing:** A distributed computing paradigm that brings computation and data storage closer to the sources of data (in this case, the user's smartphone).

1.2.2 Acronyms

- **BBP:** Best Bike Paths
- **RASD:** Requirement Analysis and Specification Document
- **DD:** Design Document
- **API:** Application Programming Interface
- **REST:** Representational State Transfer
- **DBMS:** DataBase Management System

1.2.3 Abbreviations

- **R*:** Requirement

1.3 Reference documents

This document is based on the following materials:

- The specification of the RASD and DD assignment of the Software Engineering II course a.y. 2025/26.

- Course slides shared on WeBeep.
- The Requirement Analysis and Specification Document (RASD) v1.0 of Best Bike Paths.
- Past Design Documents.

1.4 Overview

1. **Introduction:** this section introduces the project. It contains a high level description of the system, including its architectural style and architectural choices.
2. **Architectural design:** this section is very broad and contains the description of the various interfaces of the system, its deployment and an in-depth description of the components and their interactions (Runtime view).
3. **User interface design:** this section focuses on the user interface design, expanding on what was shown in the RASD.
4. **Requirements traceability:** this section contains a mapping between the requirements defined in the RASD and the design elements defined in the DD.
5. **Implementation, integration and test plan:** this section describes the order in which the various components and subsystems must be developed and tested.
6. **Effort spent:** this section shows the time spent on each section of the document, for each member of the group.
7. **References:** this section contains all the various references used to write this document.

2 Overall description

2.1 Overview

2.2 Component View

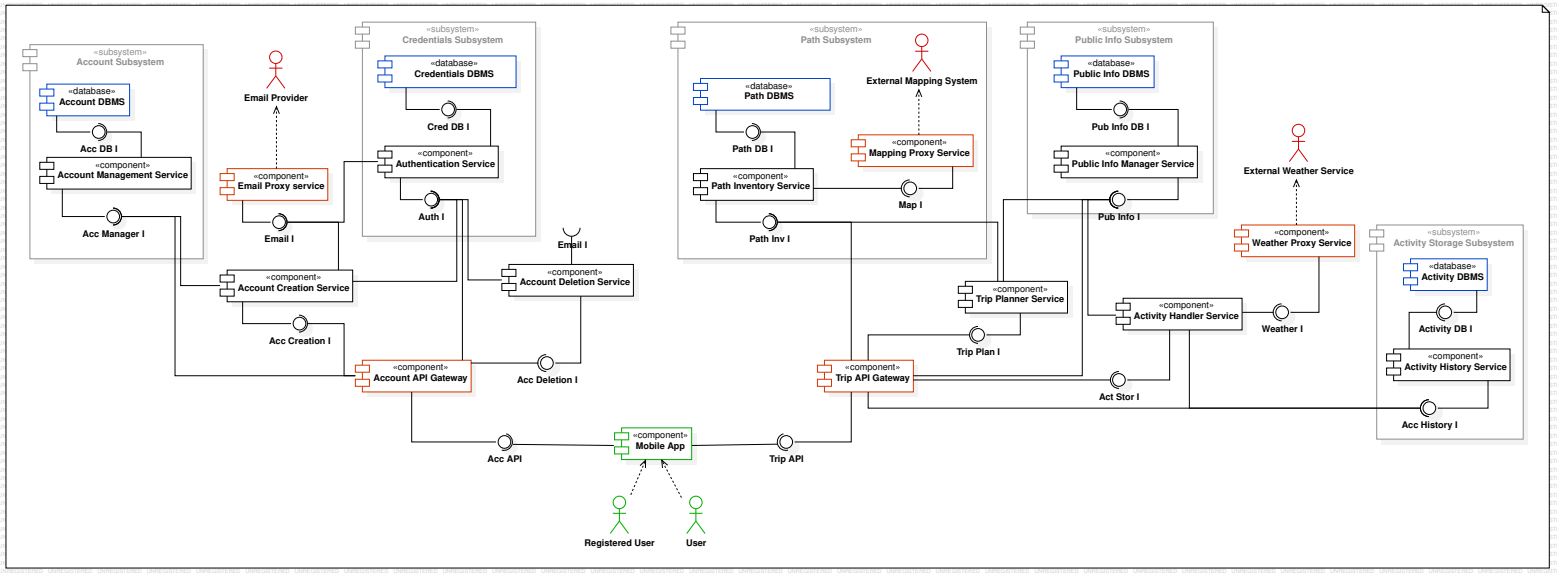
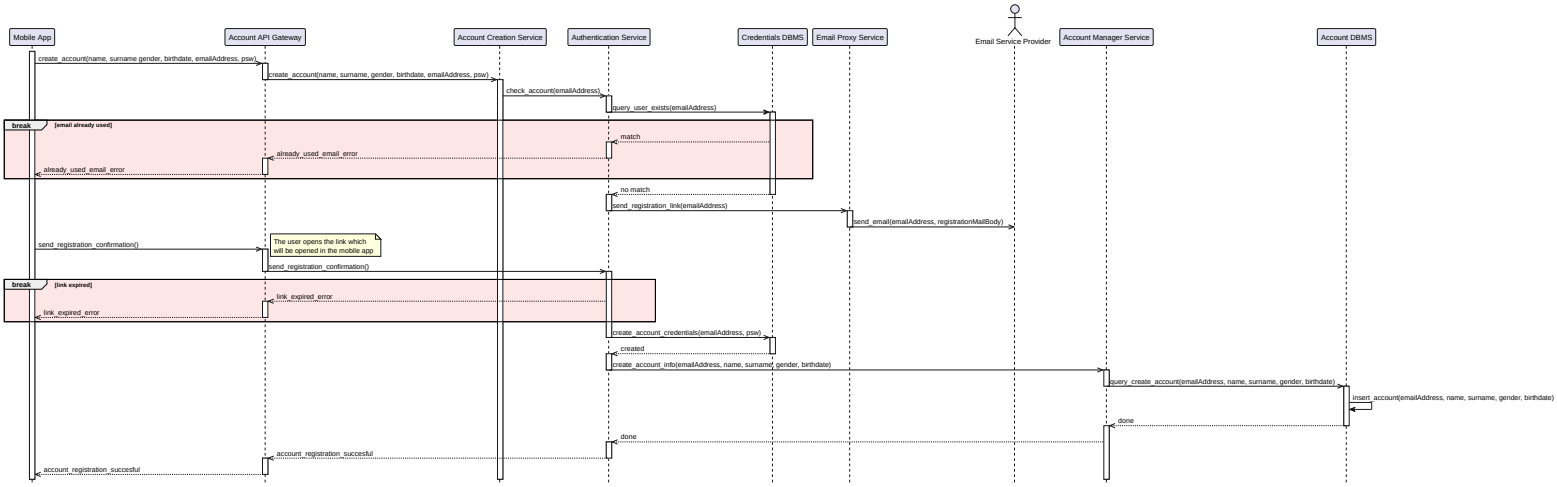


Figure 1: Component Diagram. In red are drown those components external our system; in blue those regarding the Data tier, and in Green the mobile app residing on the user's device.

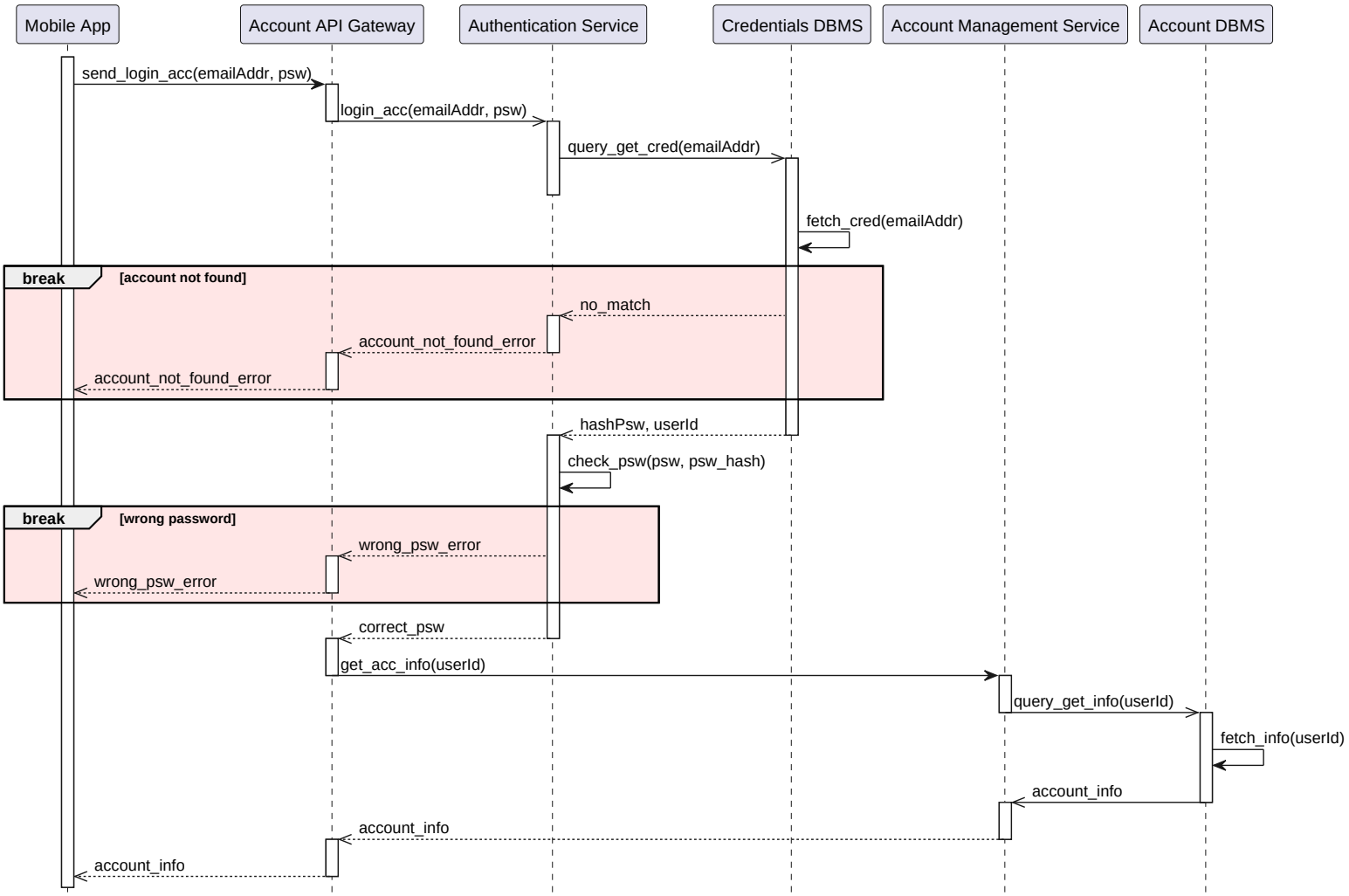
2.3 Deployment View

2.4 Runtime View

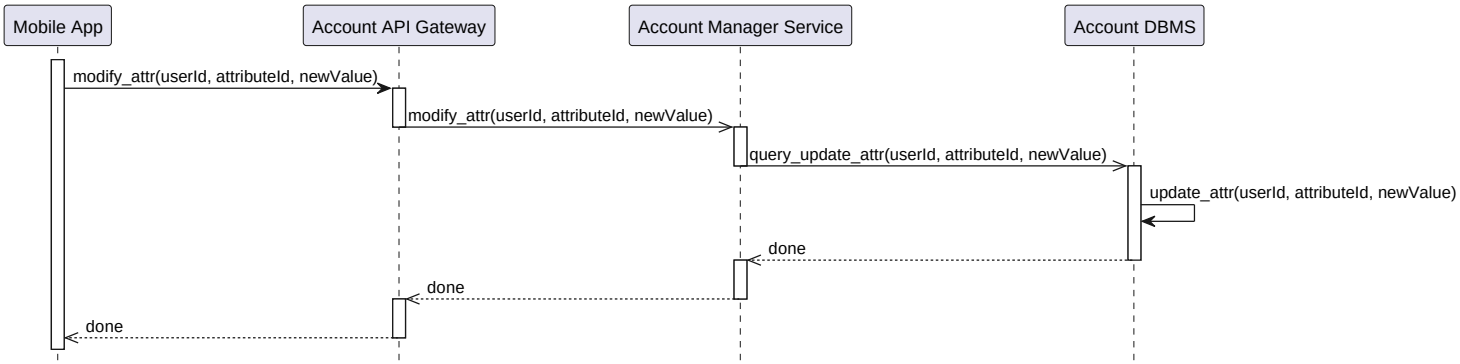
[SD1] Account creation



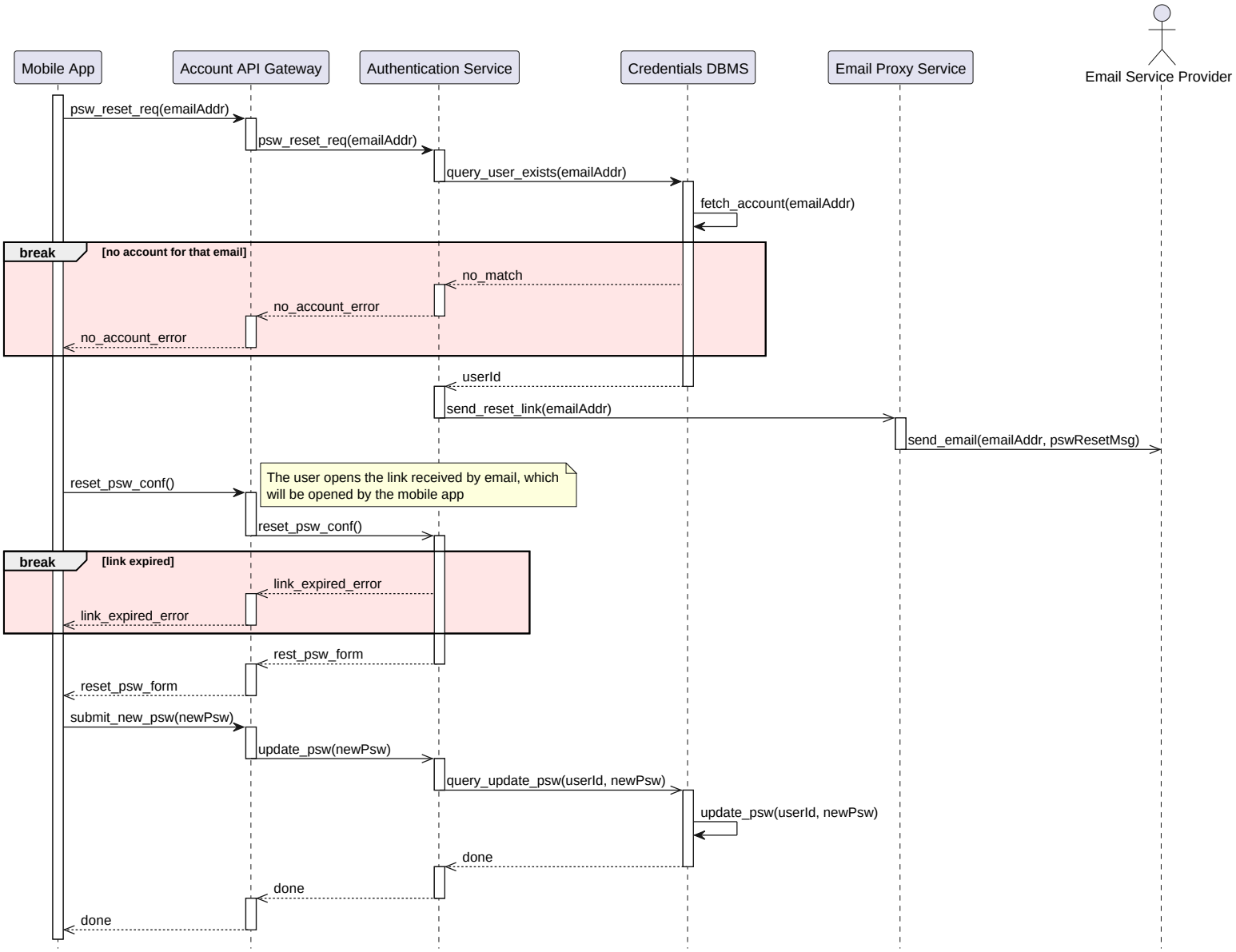
[SD2] Account login



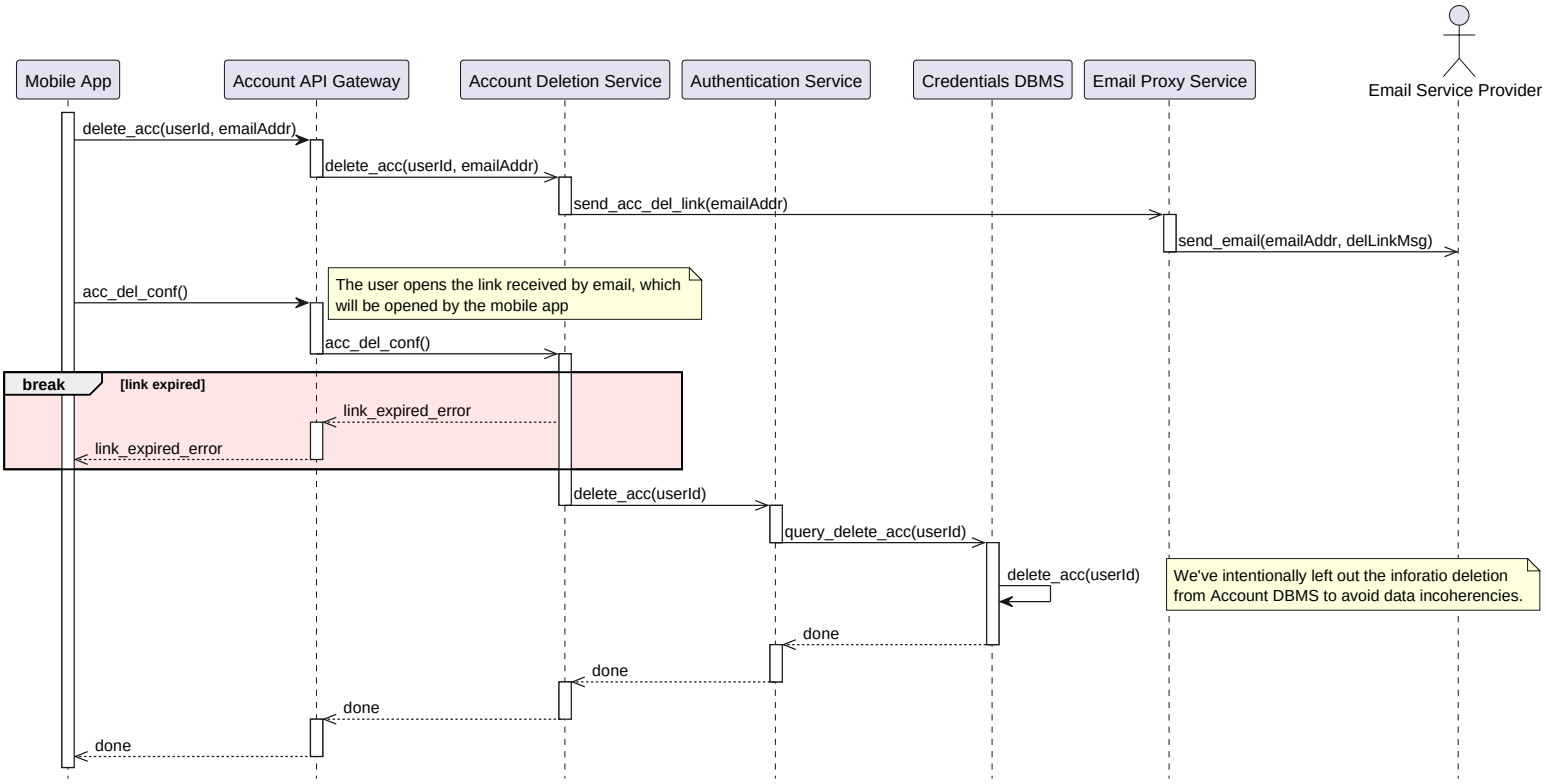
[SD3] Account update



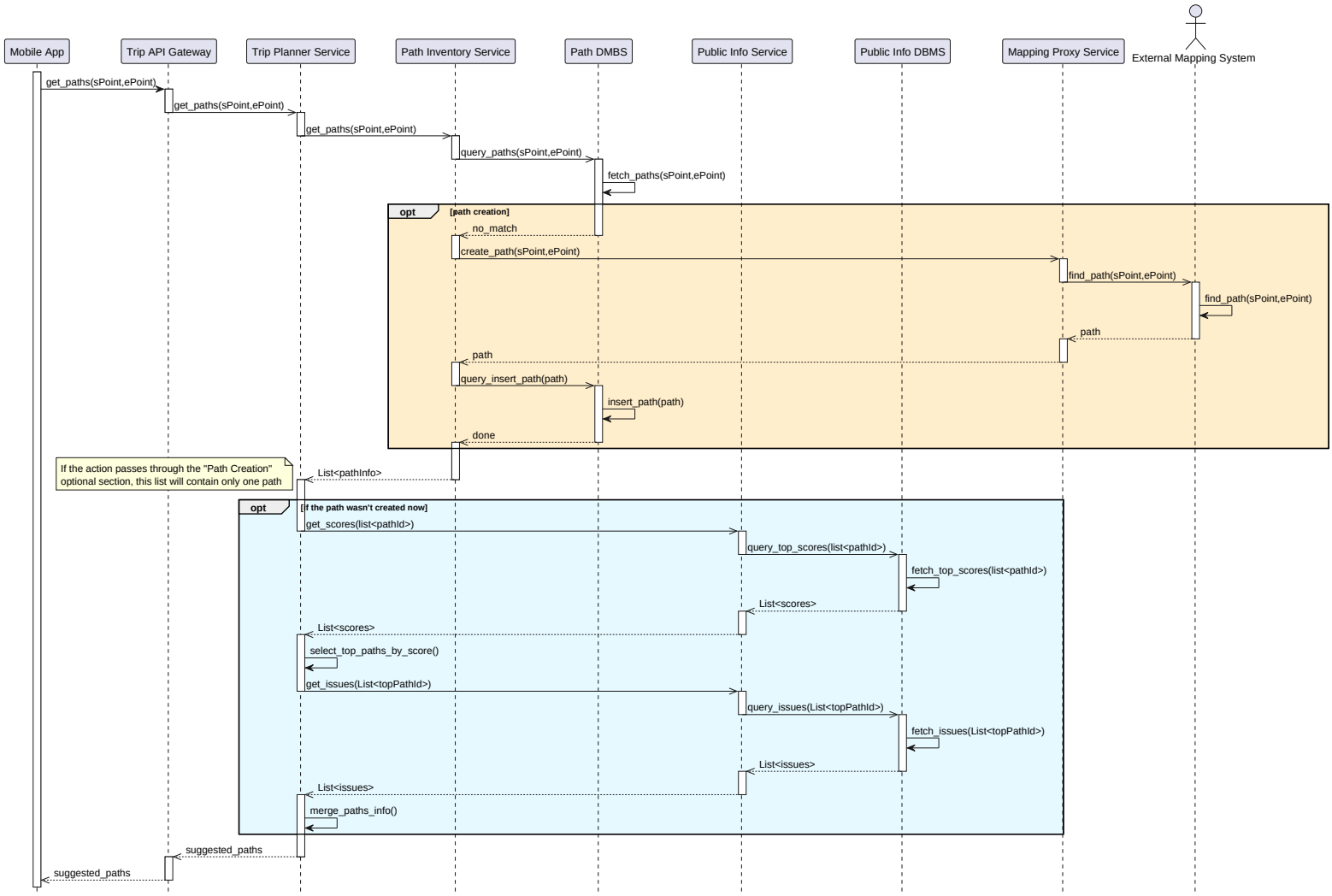
[SD4] Account Password Reset



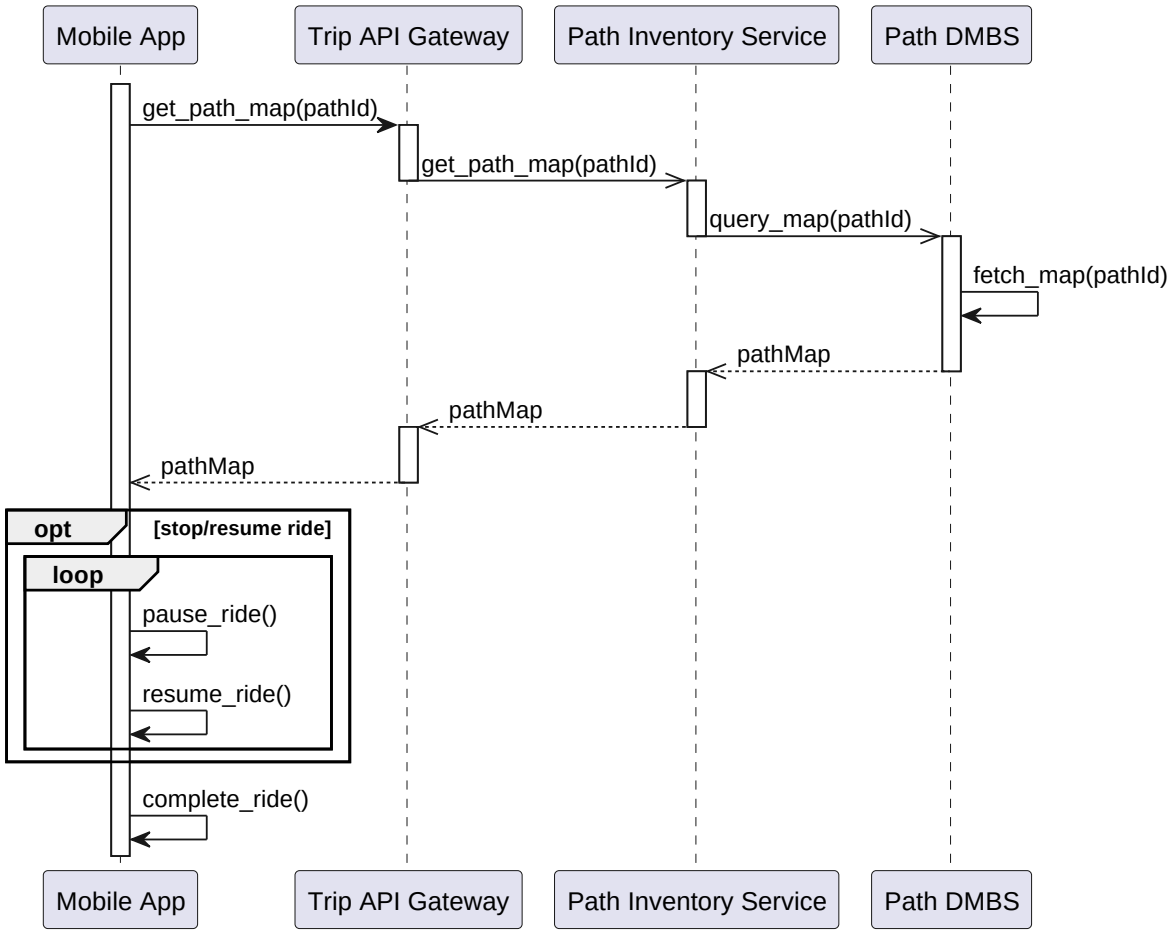
[SD5] Account Deletion



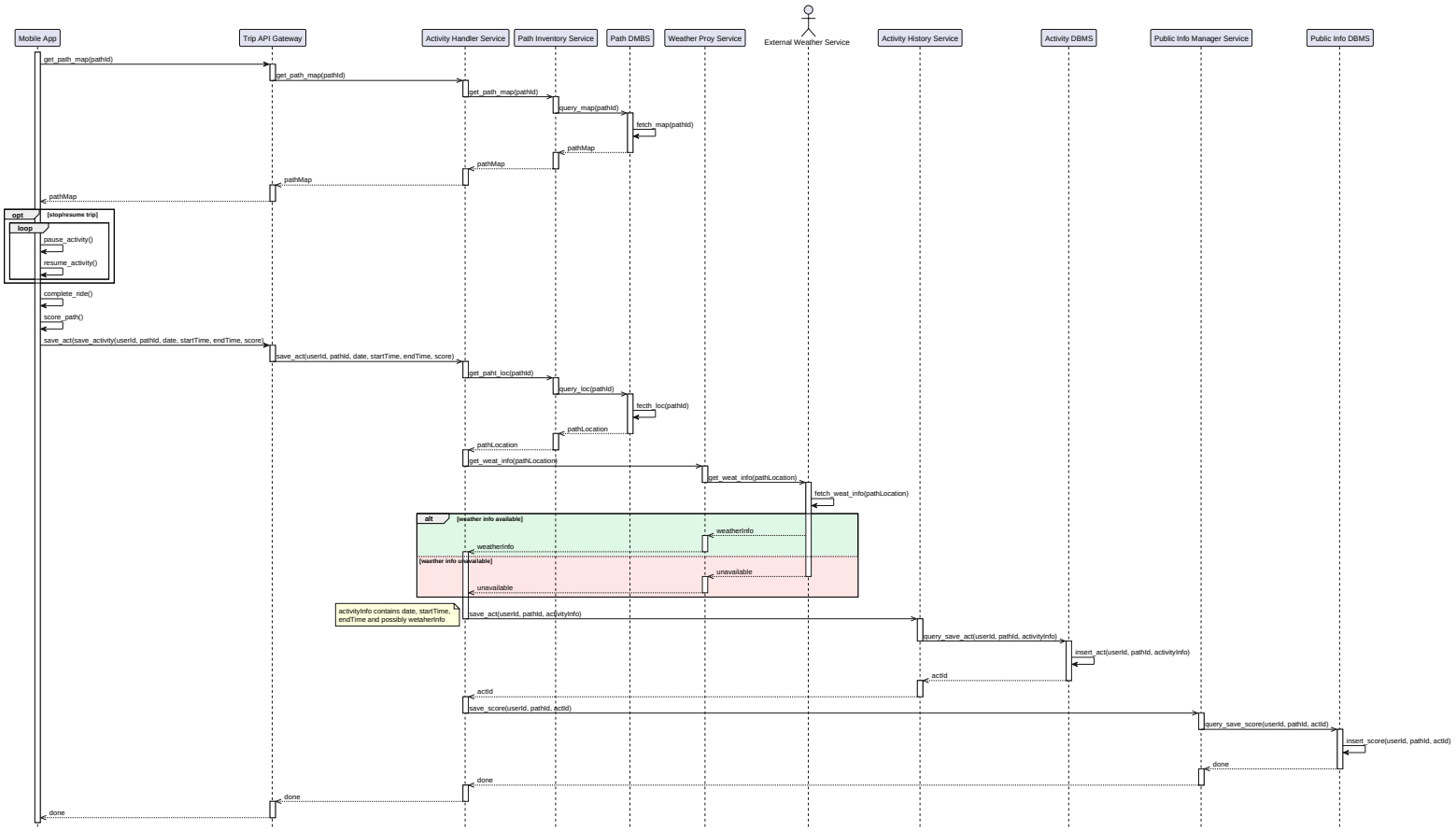
[SD6] Route Planning



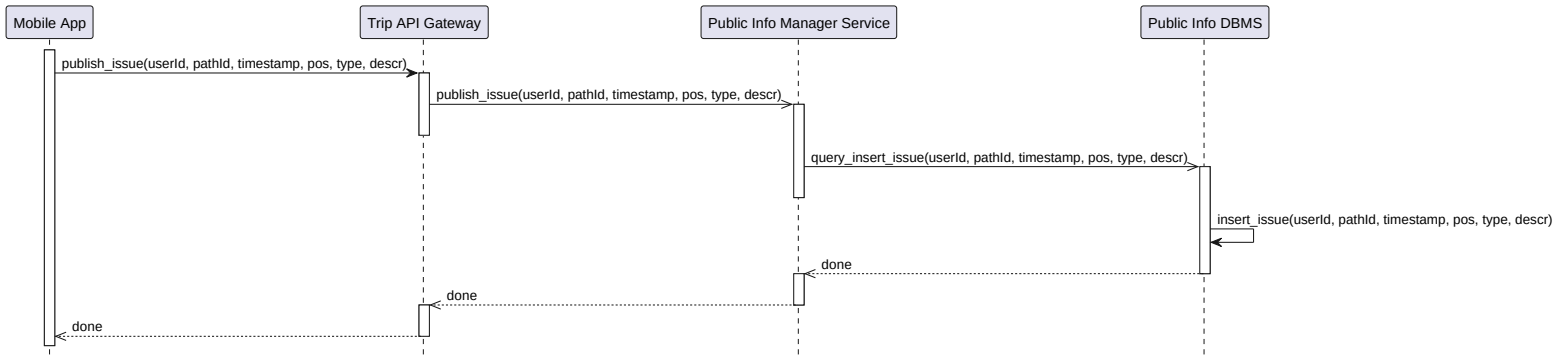
[SD7] Unregistered User Ride



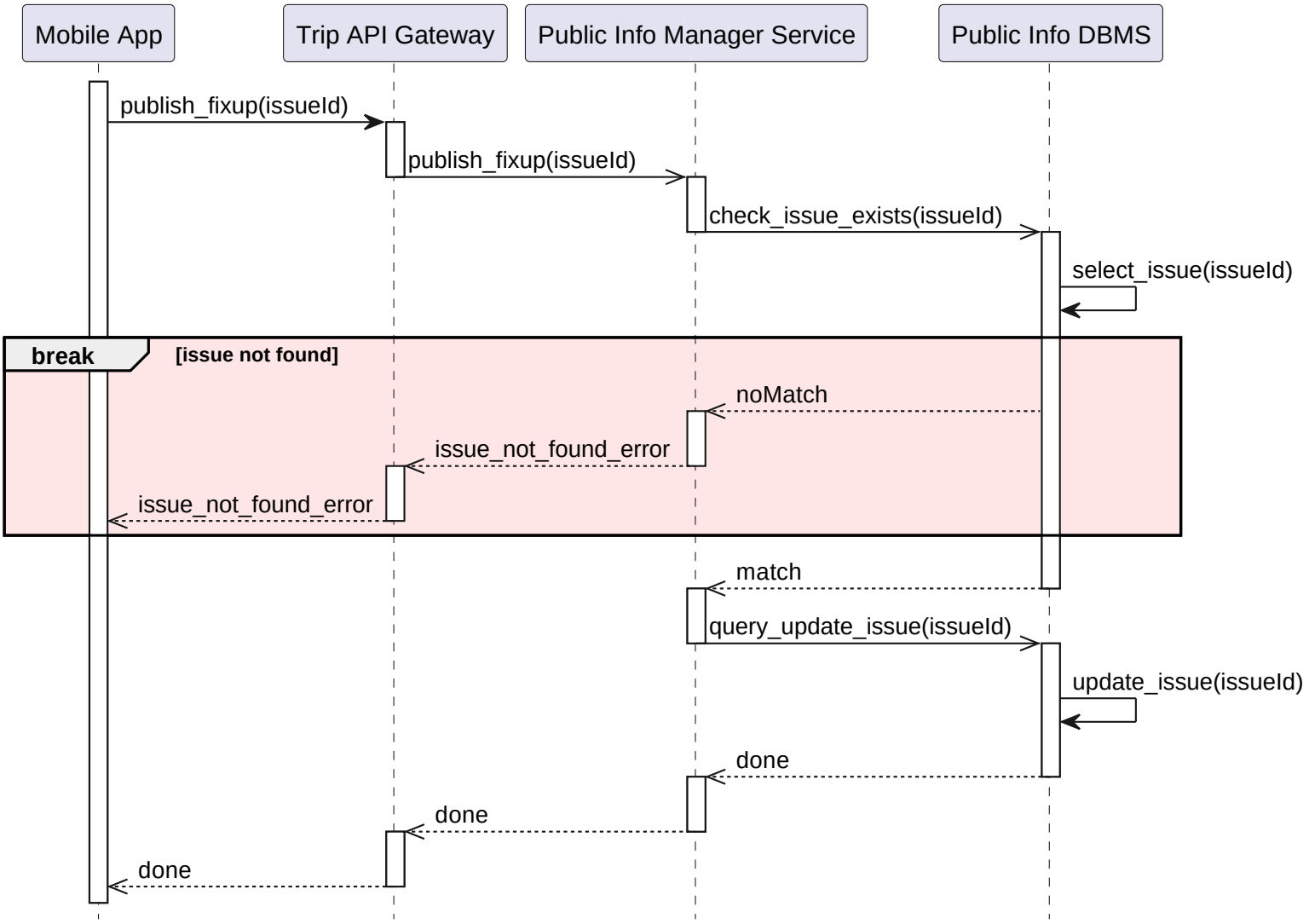
[SD8] Registered User Activity



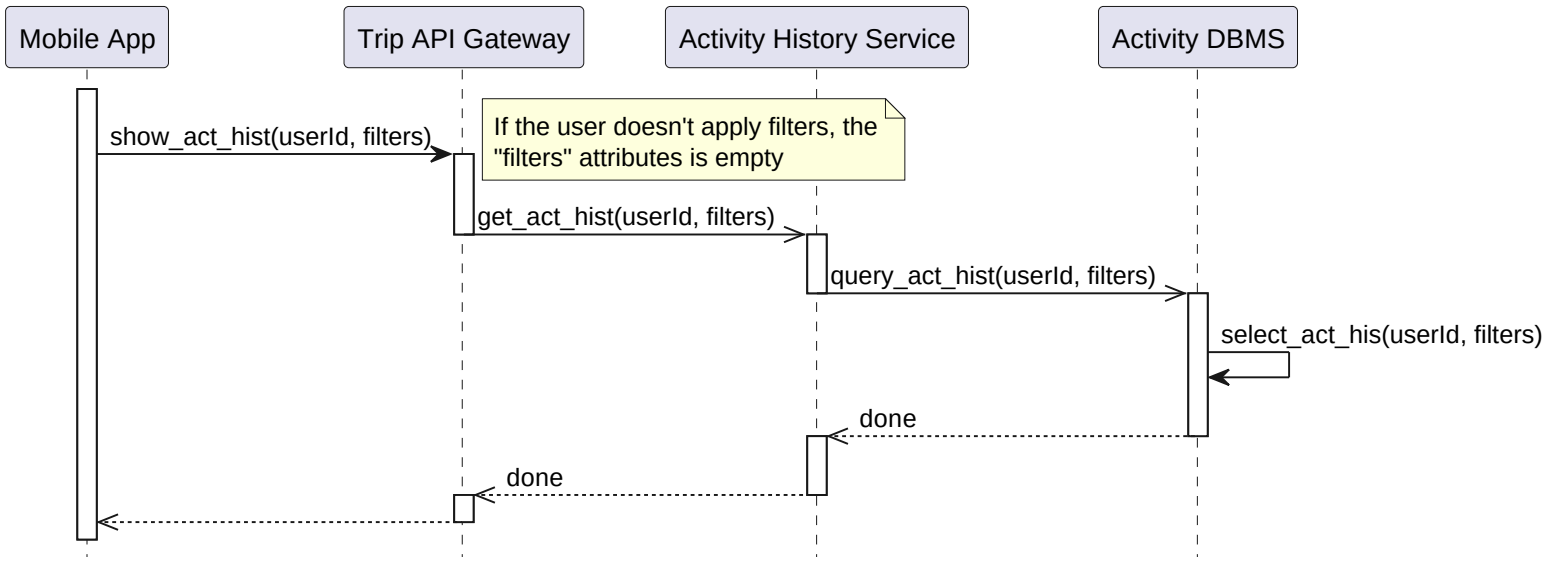
[SD10] Manual Issue Report



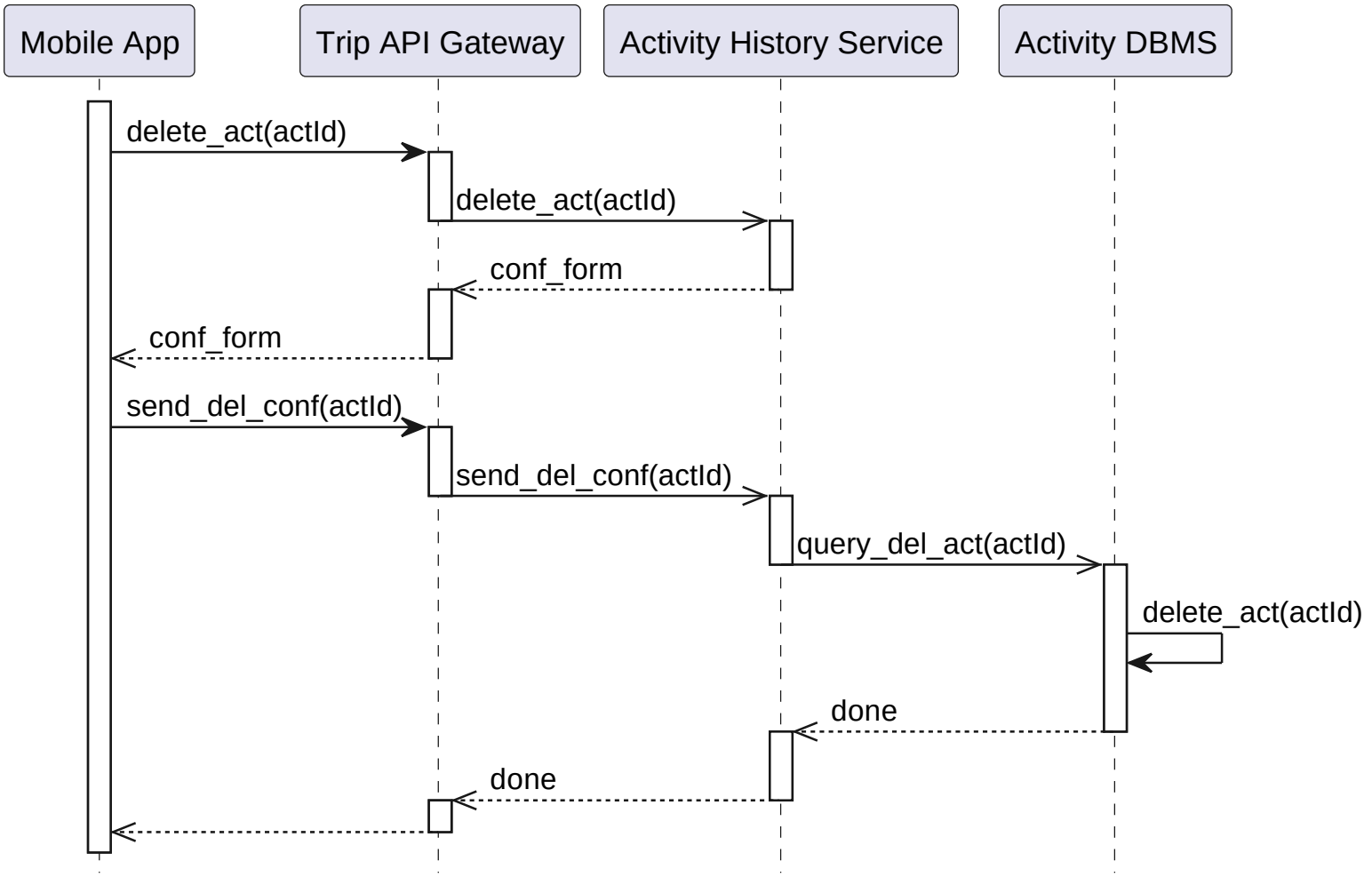
[SD11] Fixup Report



[SD12] Activity History Consultation



[SD13] Delete Activity Record



3 User Interface Design

3.1 Overview

The BBP user interface is designed according to "Minimal Attention UX" principles to ensure safety during mobile use. While the RASD defined the visual appearance through mockups, this section details the navigation logic and the structure of the transitions between views, organized by functional areas.

3.2 Navigation Logic

3.2.1 Entry Point and Authentication Flow

Access to the system is managed through the *Login* screen, which acts as the main hub to direct the user towards the authenticated or anonymous flow.

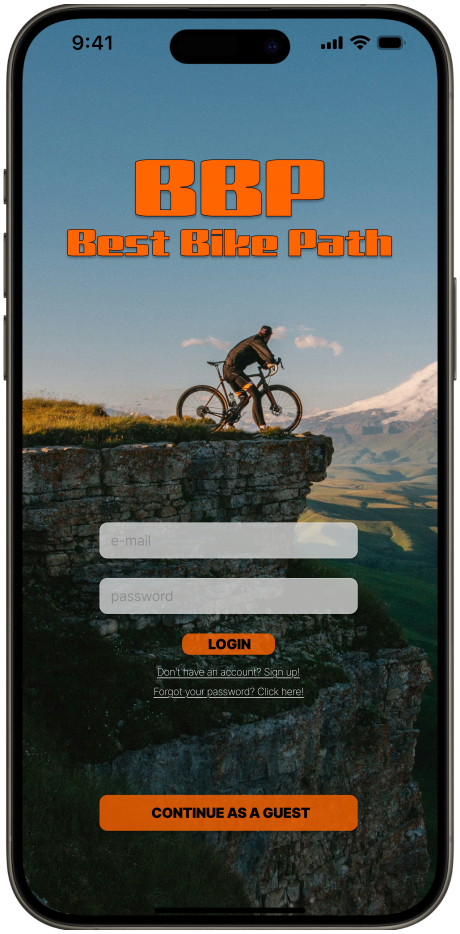


Figure 2: Start Screen and Access Logic

Transitions and Logic:

- **Guest Path:** The ***Continue As A Guest*** button instantiates an anonymous session and redirects the user directly to the *Map View* (Fig. 3) with limited functionality.
- **Registered Path:** Entering credentials and tapping ***Login*** validates the token; if successful, the user is redirected to the *Map View* with the

Bottom Bar enabled.

- **Registration:** The ***Sign up*** link opens the registration form, which, once completed, returns to this screen for the first login.
- **Password Recovery:** The ***Forgot your password?*** link starts the external credential recovery flow via secure email. Once the process is complete, the user remains on this screen to log in with the new password.

3.2.2 Core Experience: Search and Tracking

The map is the central view of the application. From here, the user plans and takes action.

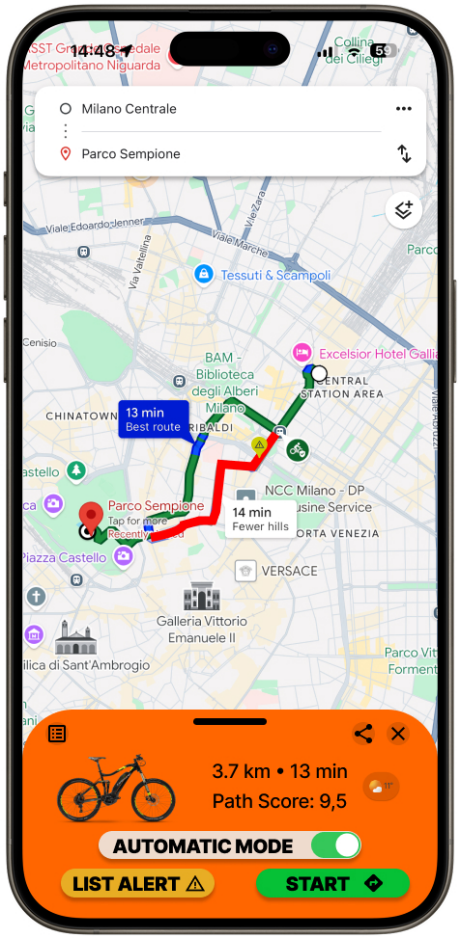


Figure 3: Main Map View and Route Selection

Transitions and Logic:

- **Route Selection:** Interacting with search results or tracks on the map updates the Bottom Sheet shown in the figure.
- **Start Navigation:** The ***Start*** button initializes the Tracking state, placing the interface in "Driving Mode".
- **Alerts:** The ***List Alert*** button opens a modal overlay listing known obstacles on the route.

- **Automatic Mode:** The ***Automatic mode*** toggle switch allows the user to explicitly enable or disable sensor data acquisition for the upcoming trip.
- **Sharing:** The ***Share*** icon (top right) activates the operating system's native share sheet for sharing the route with other users.
- **Text View:** The ***List*** icon (top left of the panel) switches the view from the graphical map to a text list of navigation directions.

3.2.3 Data Governance Flow

At the end of a tracking session, the system prevents an immediate return to the Home page, forcing a critical data validation step.

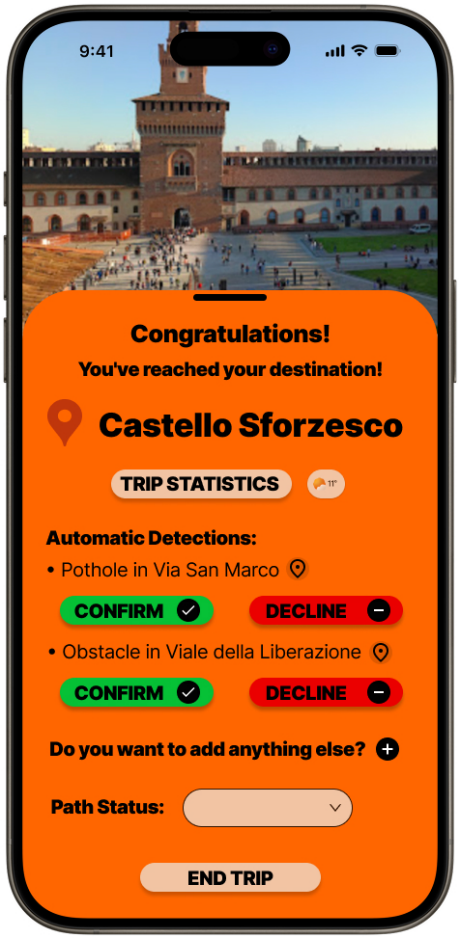


Figure 4: Confirmation and Data Validation Screen

Transitions and Logic:

- **Loop Closure:** This view is accessible only after the `Stop Recording` event and represents the mandatory Data Governance step.
- **Immediate Feedback:** The ***Trip Statistics*** section immediately displays a summary of the performance metrics calculated for the trip just completed, such as distance, duration, average speed, etc.
- **Automatic Validation:** The ***Confirm*** / ***Decline*** toggles allow the

user to validate or reject anomalies detected by the sensors, preventing false positives.

- **Manual Enrichment:** The ***Do you want to add anything else?*** button opens a context menu that allows the user to manually report any issues not detected by the sensors or add specific details.
- **Qualitative Evaluation:** The ***Path Status*** selector allows the user to assign an overall rating on the condition of the path just completed (e.g., Optimal, Average, Poor).
- **Exit and Persistence:** The ***End Trip*** button is the only way out: it finalizes the session, saves the confirmed data to the remote database, and redirects the user to the *History/Profile* view.

3.2.4 Persistence and History

The personal area allows asynchronous consultation of saved data.

Figure 5: View Trip History

Transitions and Logic:

- **Access:** Accessible via the ***Profile*** tab in the Bottom Navigation Bar.
- **Detail:** Tapping on a card in the list opens the detail view of the individual trip.

4 Requirement Traceability

5 Implementation, Integration and Test plan

6 Effort Spent

Student	Section 1	Section 2	Section 3	Section 4
Guglielmi Leonardo		15h		
Lo Conte Francesco	5h		(todo)	

Table 1: Effort spent by each team member per section

Lo Conte Francesco

- 02/12/2025 5h (Section 1,2)

7 References