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 3, 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

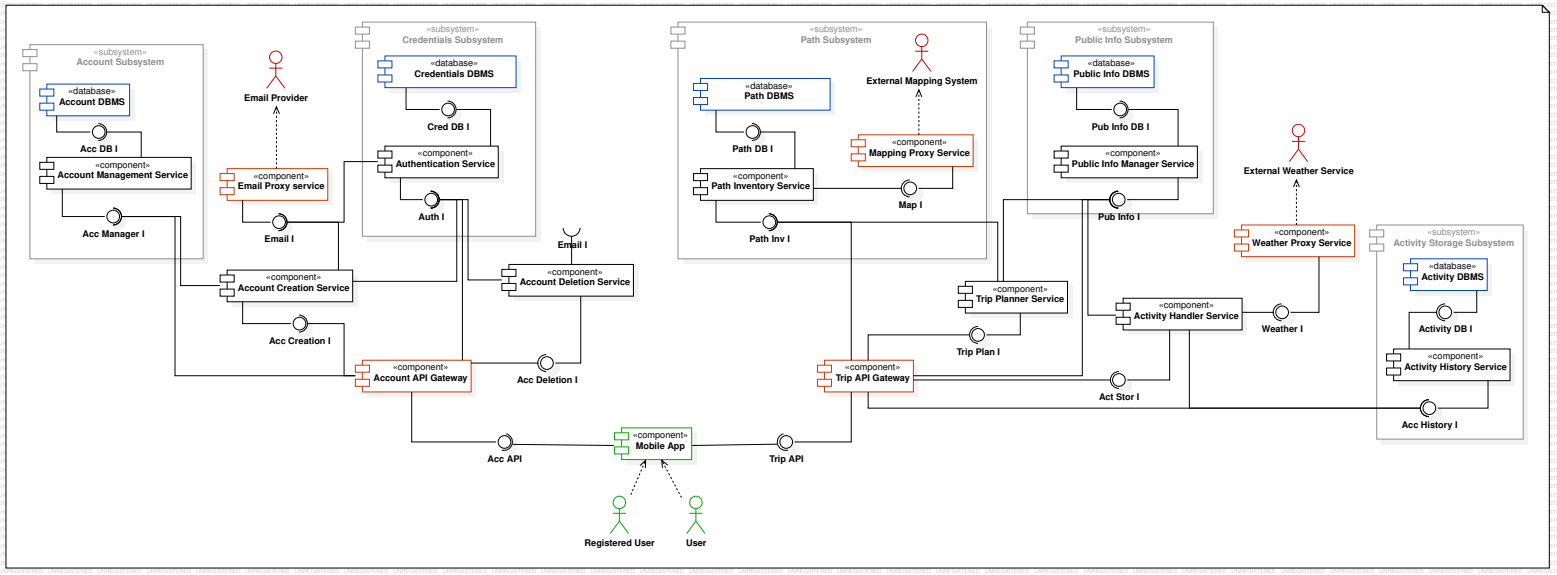


Figure 1: Component Diagram. In red are drown those actors external to our system; in blue the DBMS and those components in Data tier; in green the mobile app residing on the user's device.

Mobile App

The Mobile Application serves as the primary user interface, enabling users to manage their accounts, browse available paths, and initiate trips with real-time monitoring.

Account API Gateway

Implements the Gateway Pattern to provide a unified entry point for all account management services.

Account Creation Service

The Account Creation Service manages the process of creating a new account. It coordinates overhall request by the user, orchestrating the communications

with the user and the data storage by interfacing with the *Authentication Service* and the *Account Manager Service*.

Account Deletion Service

This service handles the account deletion lifecycle. It interfaces with the Authentication Service to execute identity removal and manages outbound communication through the *Email Proxy Service* to verify and confirm the deletion request.

Credentials DBMS

This DBMS stores and manages account credentials.

Authentication Service

This service facilitates the authentication lifecycle. It processes login requests, performs identity verification by querying the *Credentials DBMS*, and acts as the primary interface for any credential-related data retrieval.

Acconnt DBMS

This DBMS stores and manages general account information, like name, surname or birthdate.

Account Management Service

Manages user account information and profile updates. It hanlde the access to the data stored in the *Account DBMS*, providing acting as an interface for all data retrival operation involving account information.

Email Proxy Server

This component works as a middleware between the user *Email Provider* and those services which need to send an email, avoid direct contact with something outside the system and allowing a more decoupled approach.

Email Provider

This actor represents the user email provider.

Trip API Gateway

Implements the Gateway Pattern to provide a unified entry point for all trip services (both rides and activities) and related information, like issues and scores.

Trip Planner Service

This service is responsible for retrieving and merging all the information necessary for the user during trip planning. To achieve this, it interfaces with the Path Inventory Service to retrieve the path and the Public Info Manager Service to retrieve scores and issues.

Activity Handler Service

This service scope is to handle the storage of completed activities by acting as a central coordinator. It executes this task by interfacing with the *Activity History Service* to record finished sessions. It also merges weather information into the record by interfacing with the *Weather Proxy Service*. Furthermore, it handles the storage of activity-related information such as path scores by interfacing with the *Public Info Manager Service* to enrich the final data set.

Path DBMS

This DBMS stores and manages all the informations about the paths.

Path Inventory Service

This service handles the path-related information stored in the *Path DBMS*, acting as management layer for all geographic route data. Upon request, it retrieves the specific information, and in case there isn't a match between the request and the stored data, the service handles the creation of a new path by asking it to the *Mapping Proxy Service*, ensuring that the internal database is updated with new coordinates and route details.

Mapping Proxy Service

This component acts as a security layer for communicating with the External Mapping System, serving as a protected gateway for all geographic requests that cannot be satisfied by the *Path Inventory Service*. Its also ensures that the internal architecture remains independent from interfaces external to our system.

External Mapping System

This actor represent a mapping systsem external to BBP boundaries.

Public Info DBMS

This DBMS stores and manages all the informations about *Publishable Infromation* submitted by registered users.

Public Info Manager Service

This service handles the Publishable Information published by the user, managing Issues and Path Scores. It fulfills this role by handling all requests about data stored in the *Public Info DBMS*, and by handling the issue-status updates within the system..

Weather Proxy Service

This component acts as an intermediary between the system and the *External Weather System* to facilitate secure and reliable activity data enrichment with the meteorological conditions present during the time of the trip.

External Weather System

This actor represents a weather system indipendet from the BBP system.

Activity DBMS

This DBMS stores and manages all the informations about activities.

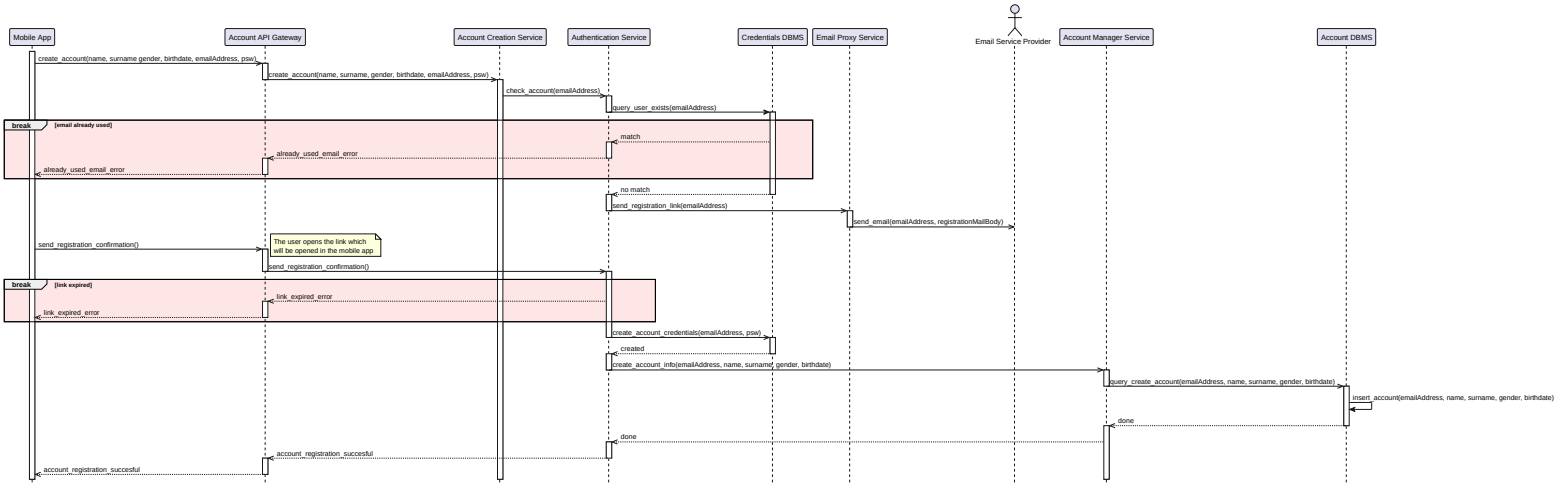
Activity History Service

This service handles the activity-related requests within the system by interfacing with the *Activity DBMS*, serving as the primary management layer for recorded user data. It is directly involved in those operations concerning the activity history of a user.

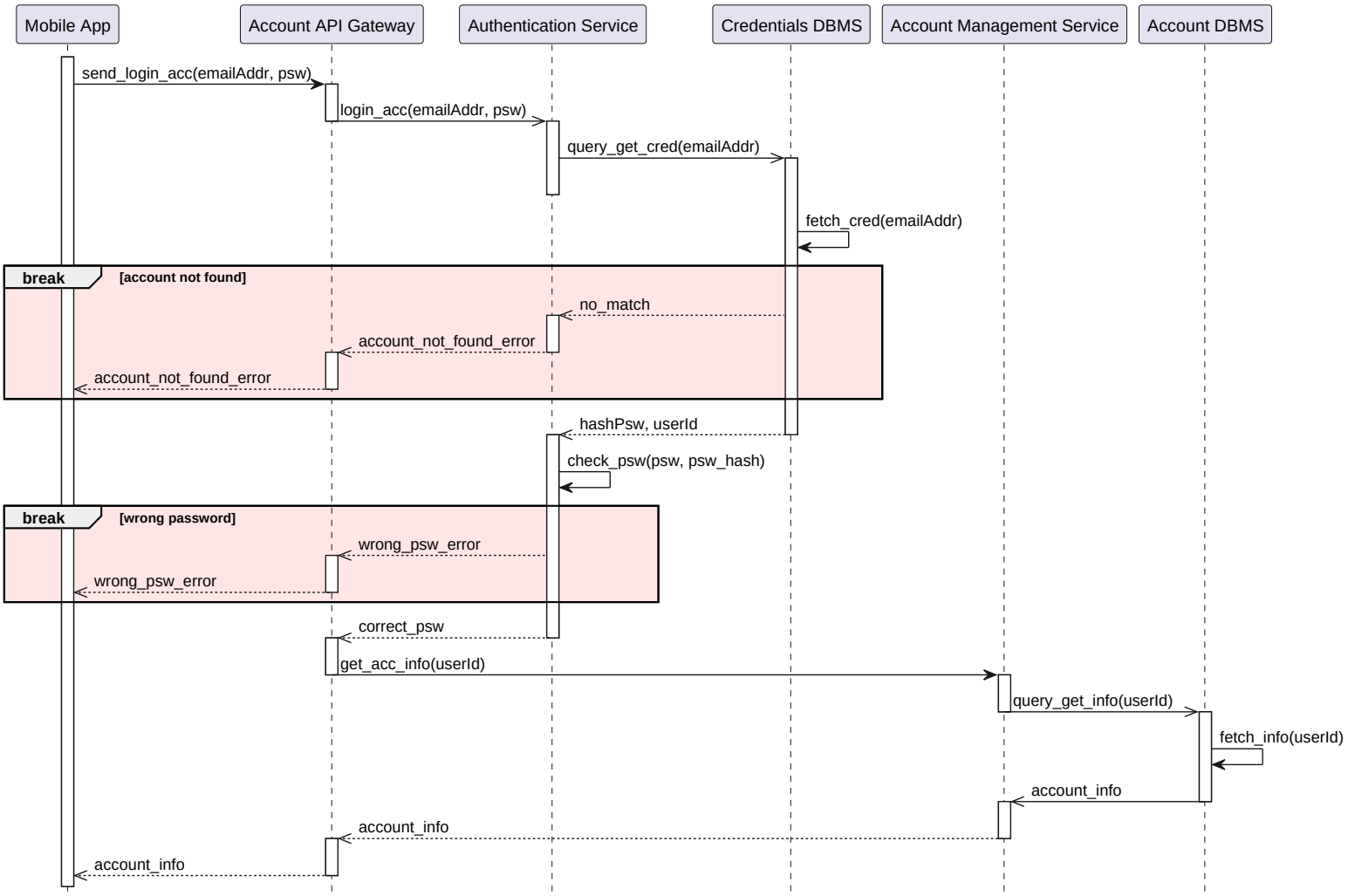
2.3 Deployment View

2.4 Runtime View

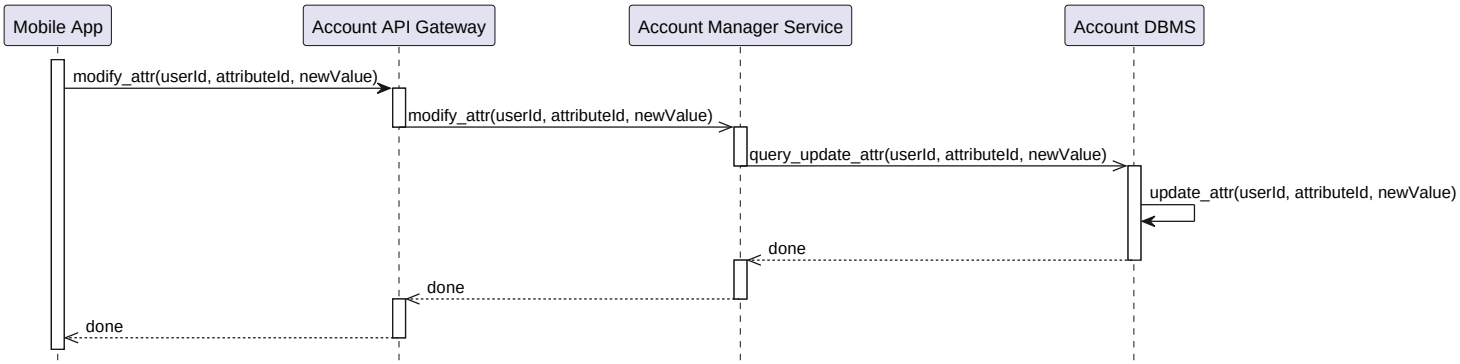
[SD1] Account creation



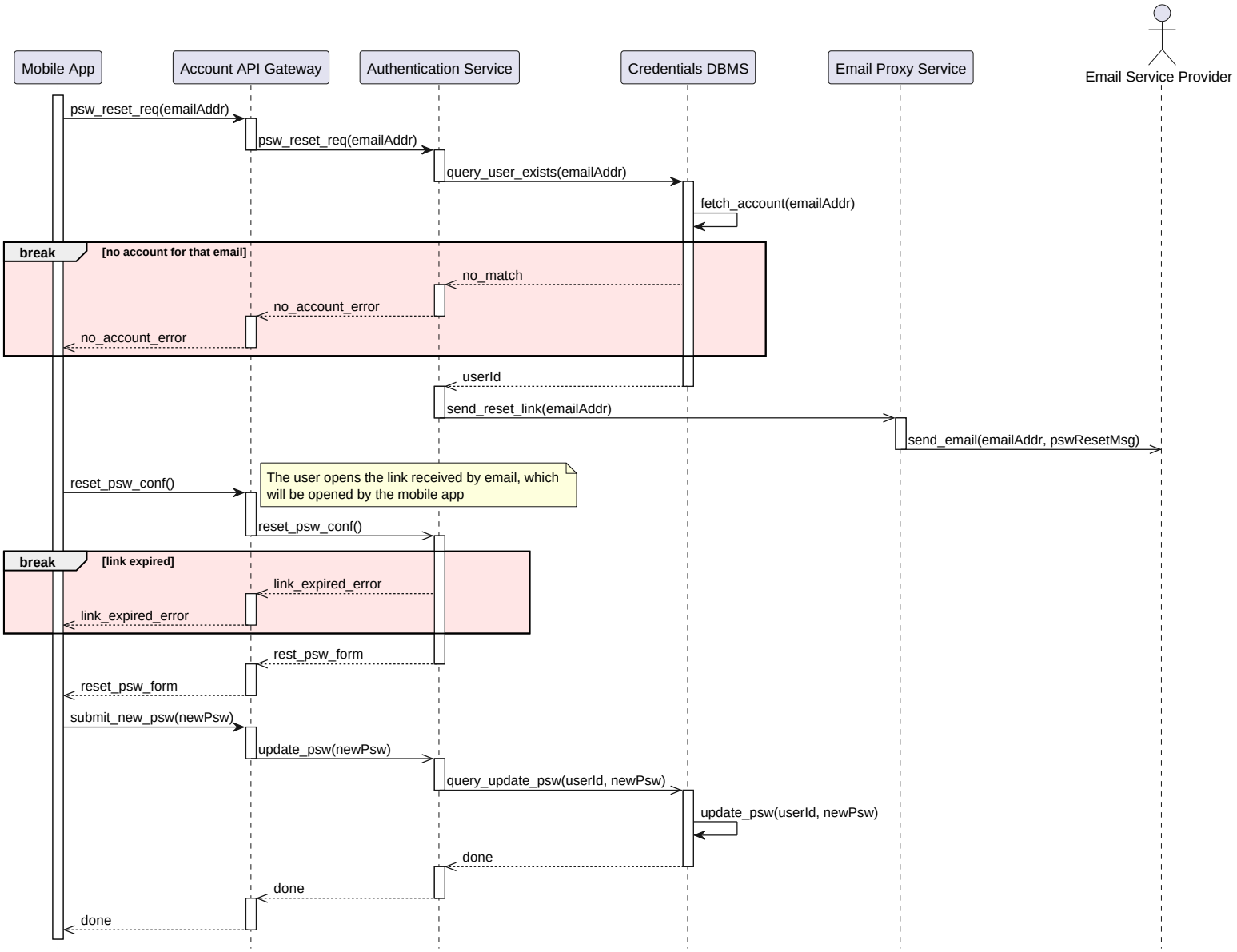
[SD2] Account login



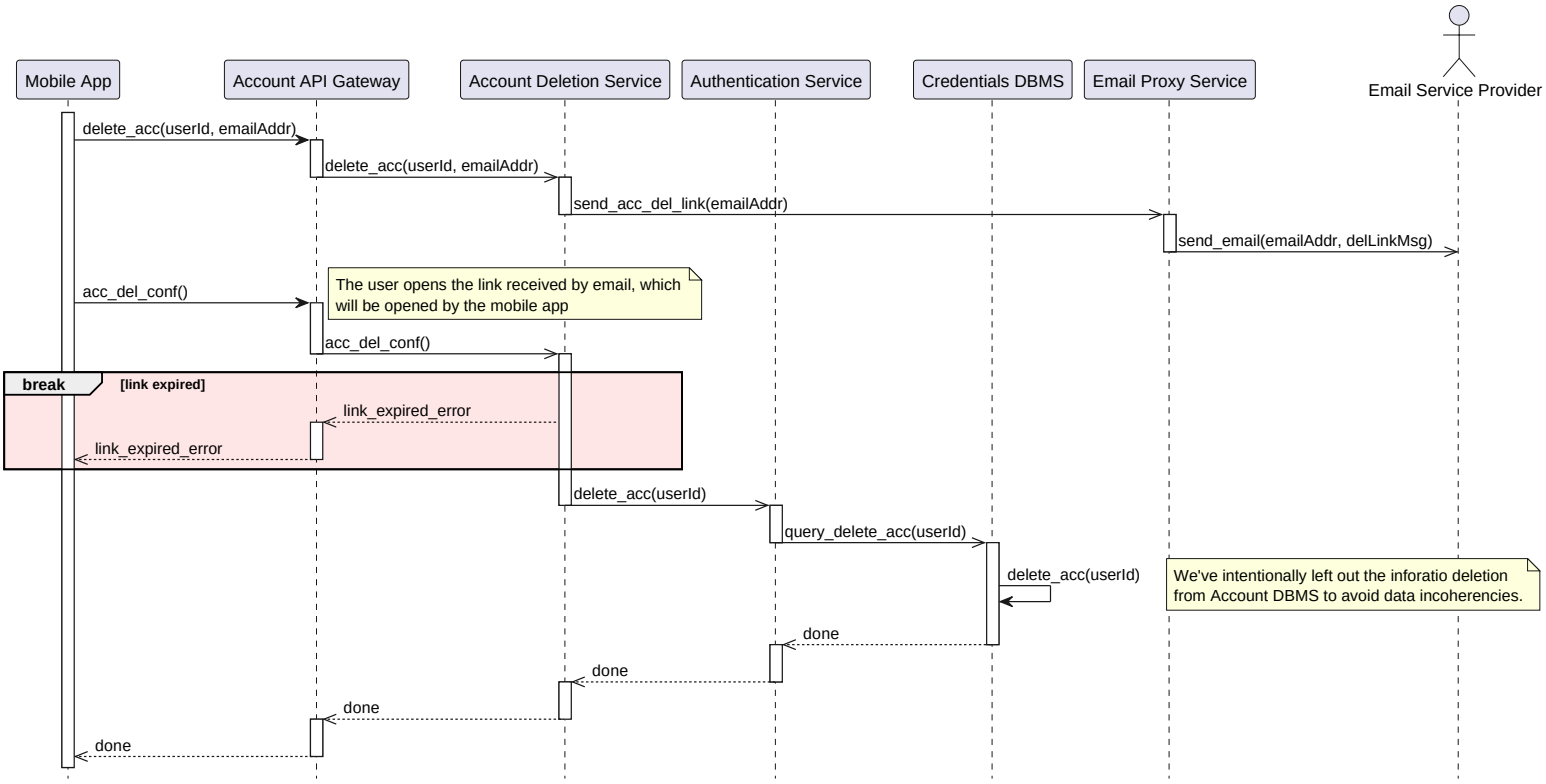
[SD3] Account update



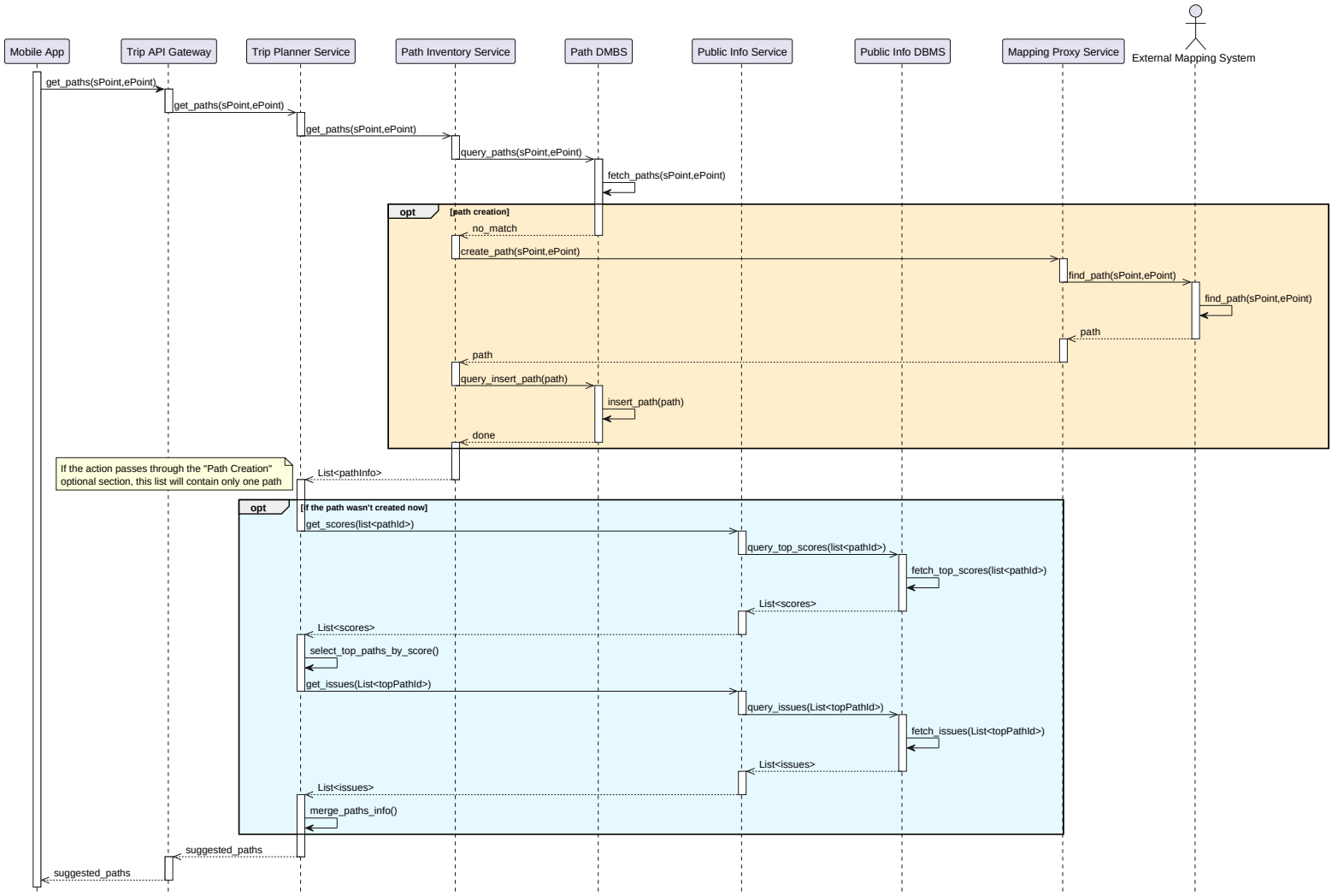
[SD4] Account Password Reset



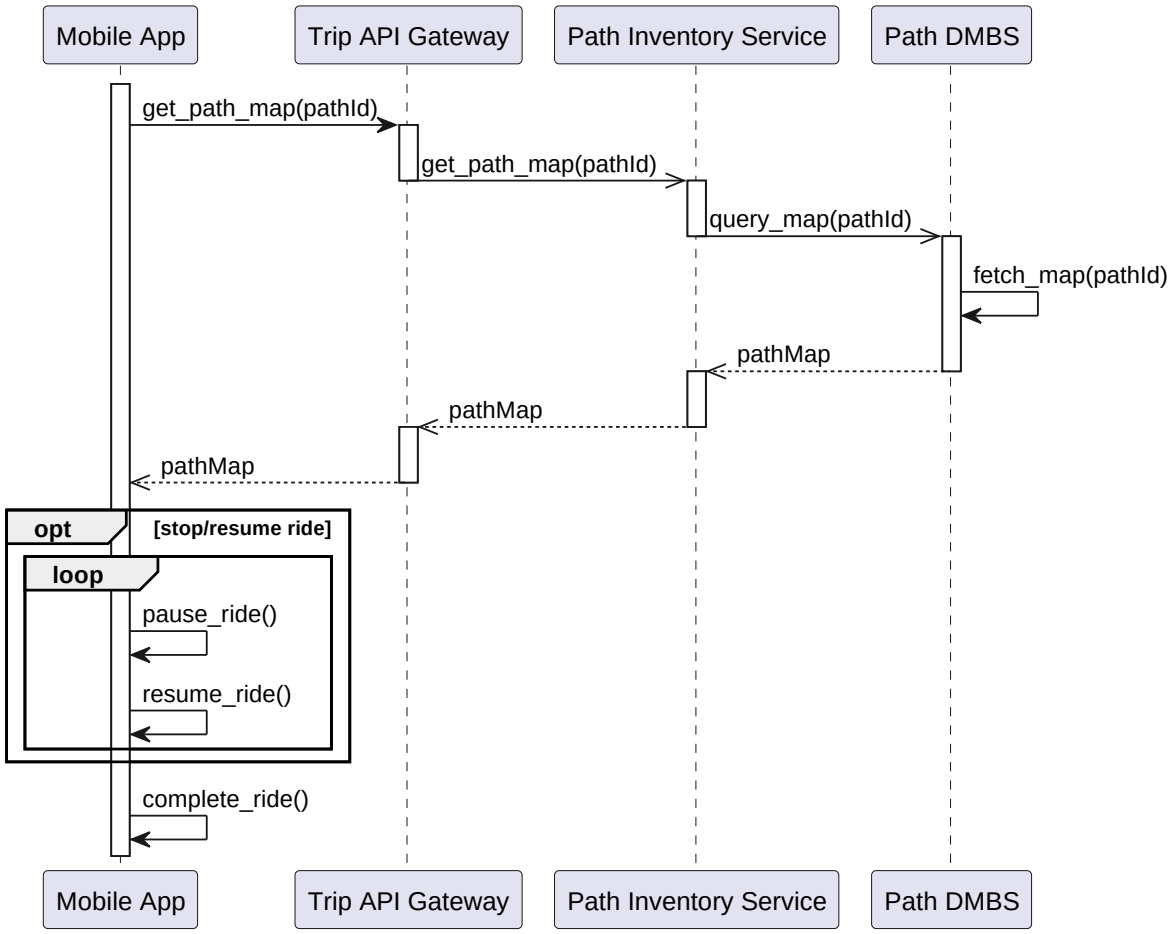
[SD5] Account Deletion



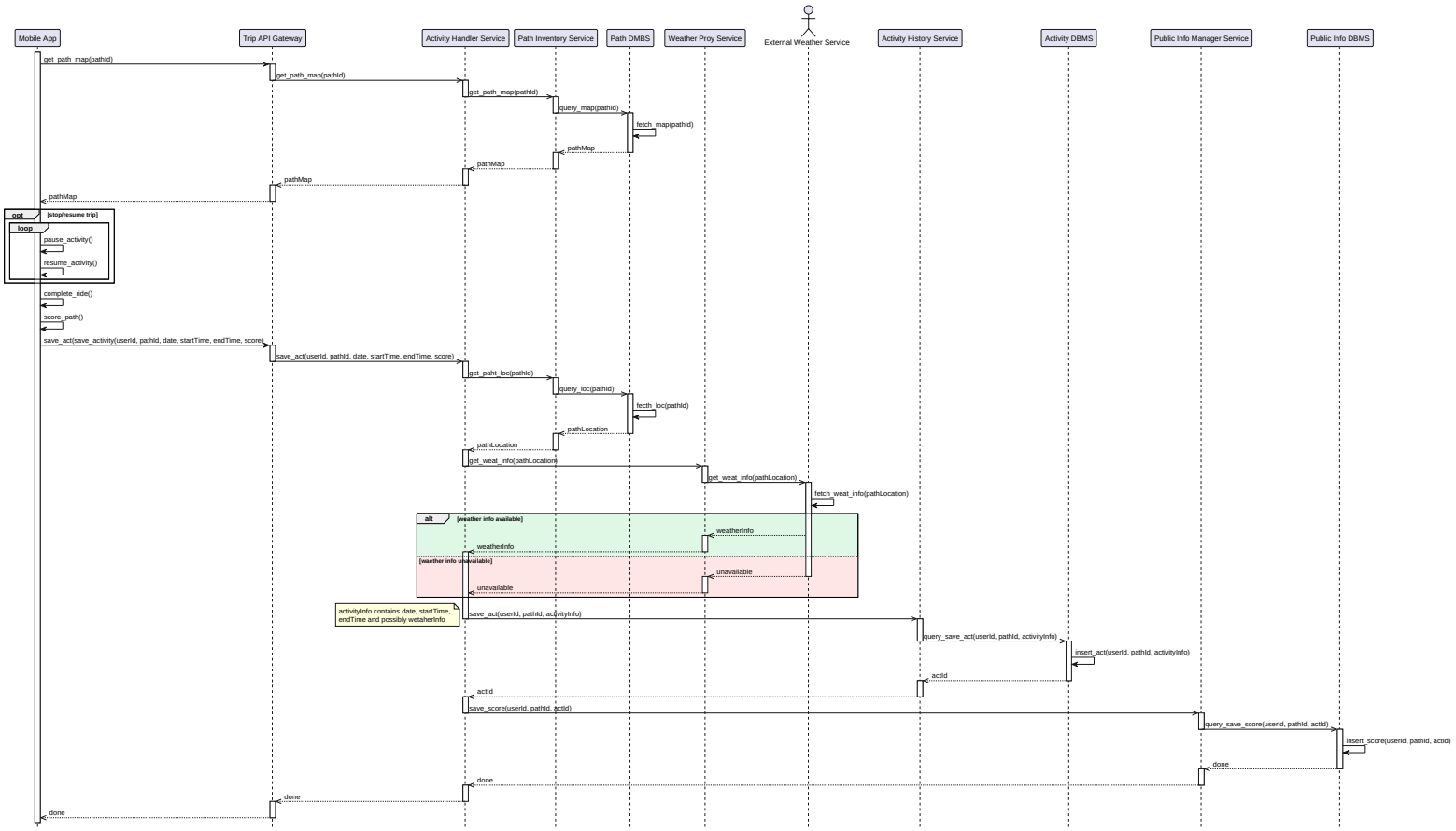
[SD6] Route Planning



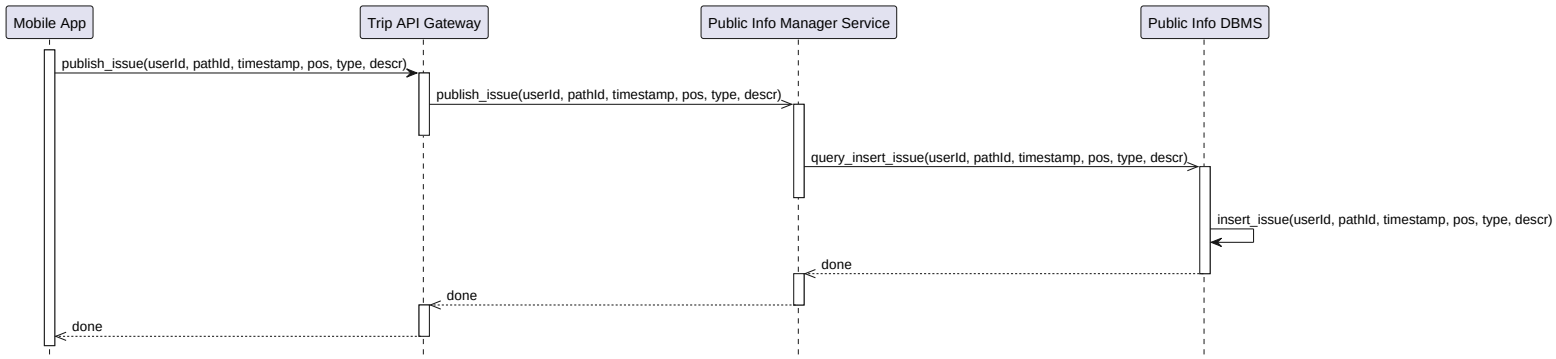
[SD7] Unregistered User Ride



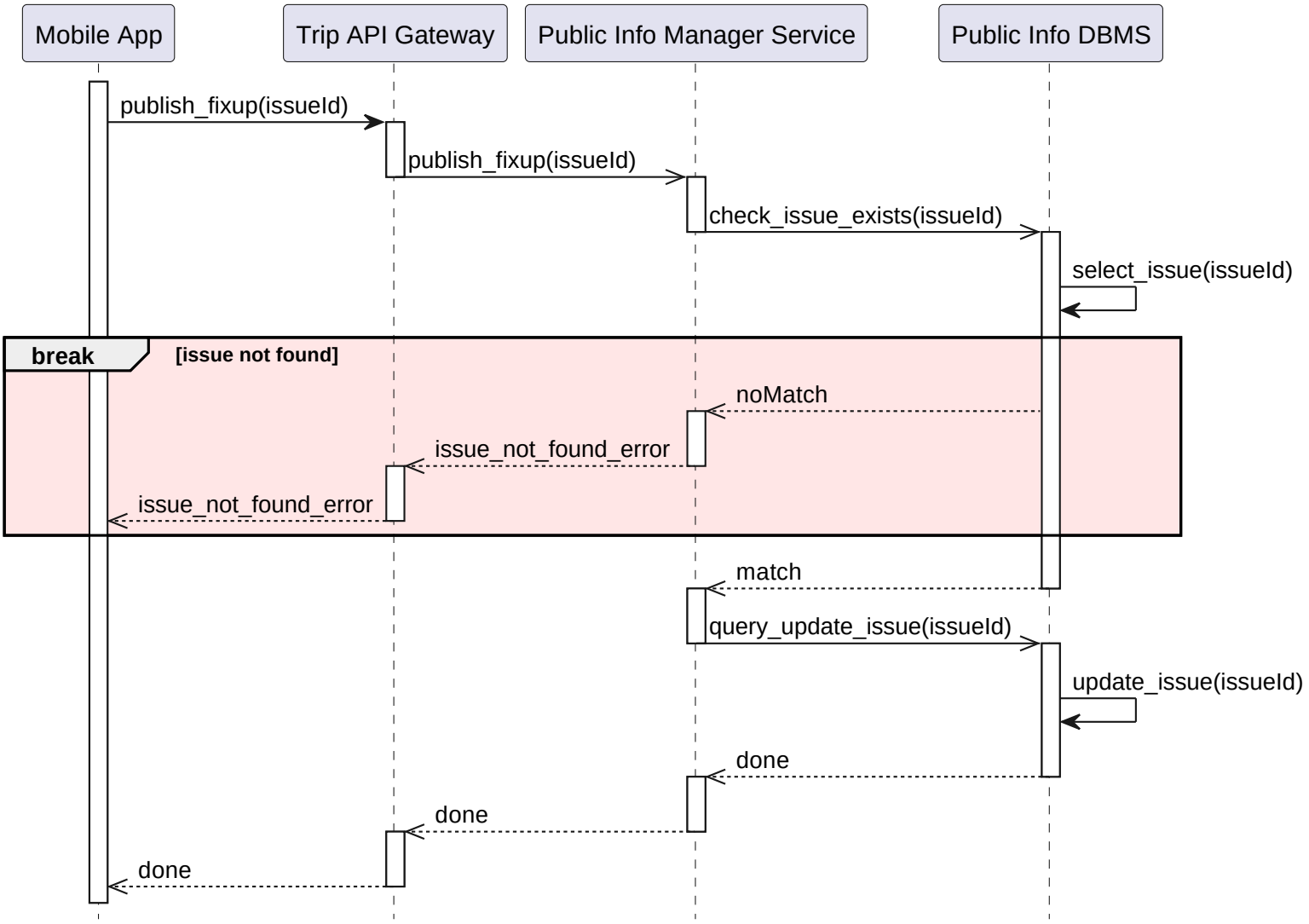
[SD8] Registered User Activity



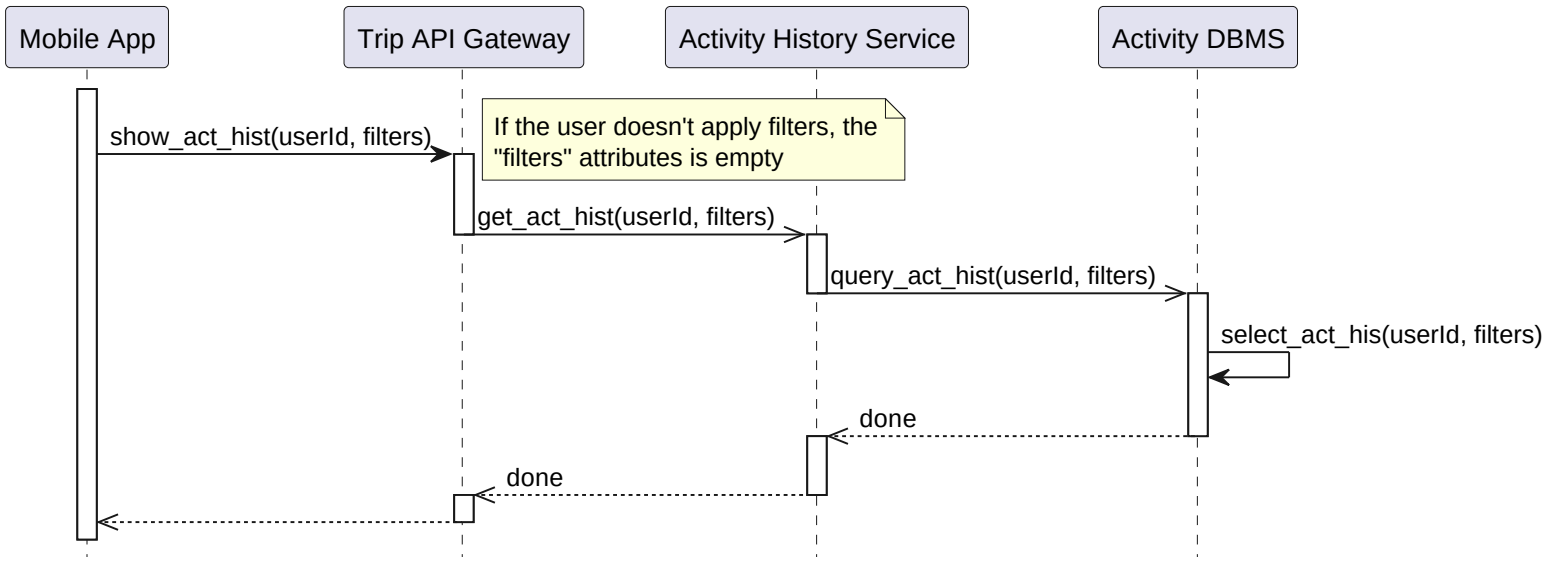
[SD10] Manual Issue Report



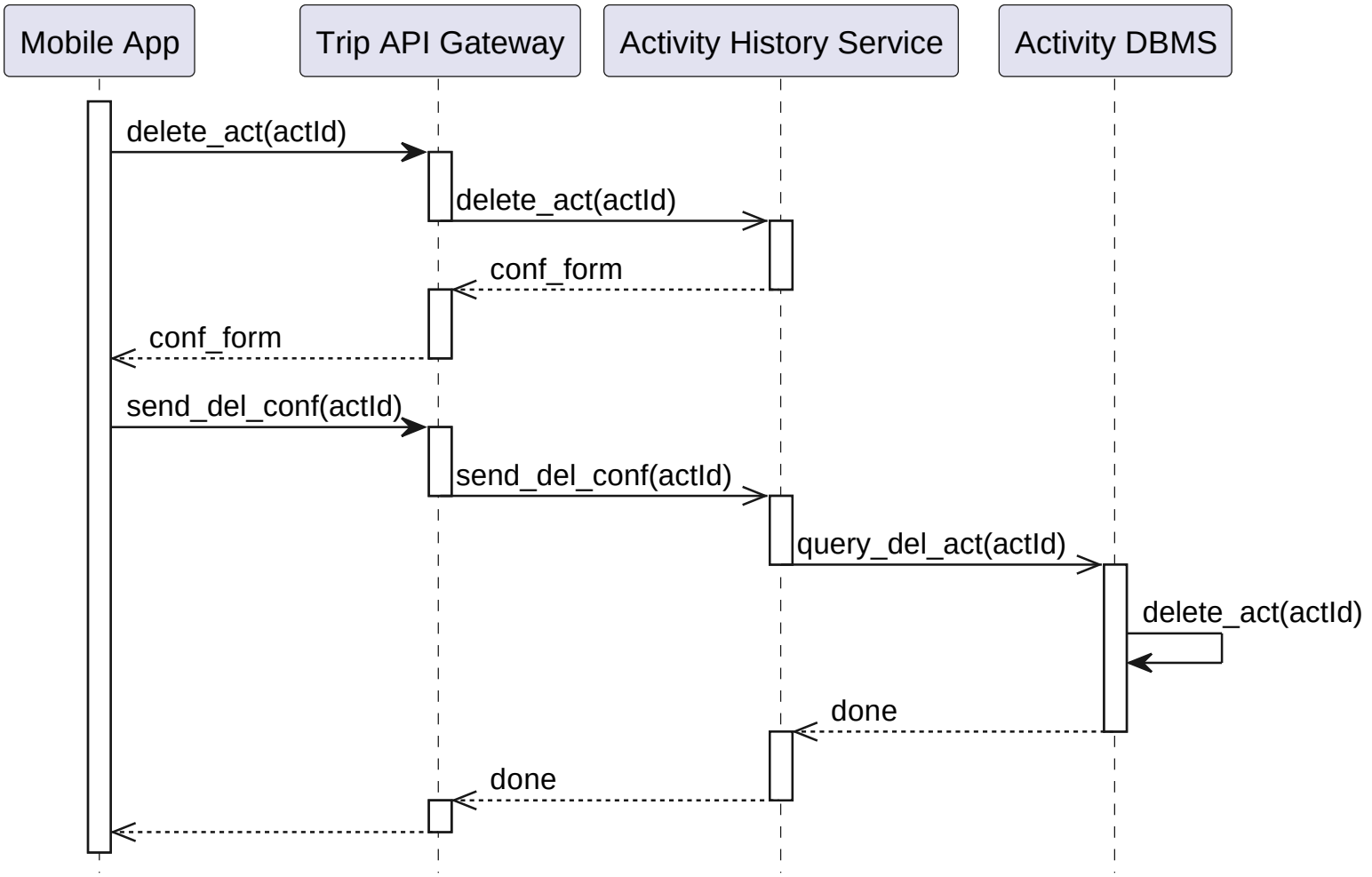
[SD11] Fixup Report



[SD12] Activity History Consultation



[SD13] Delete Activity Record



2.5 Component Interfaces

2.6 Selected architectural styles and patterns

2.7 Other design decisions

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.

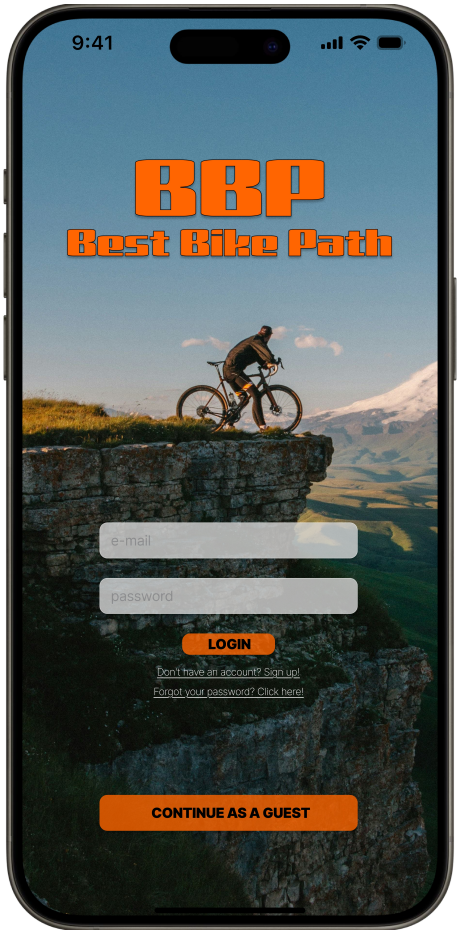


Figure 2: Start Screen and Access Logic

Transitions and Logic:

- **Guest Path:** The ***Continue As A Guest*** button instantiates an anonymous session and redirects the user directly to the *Map View* (see 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.

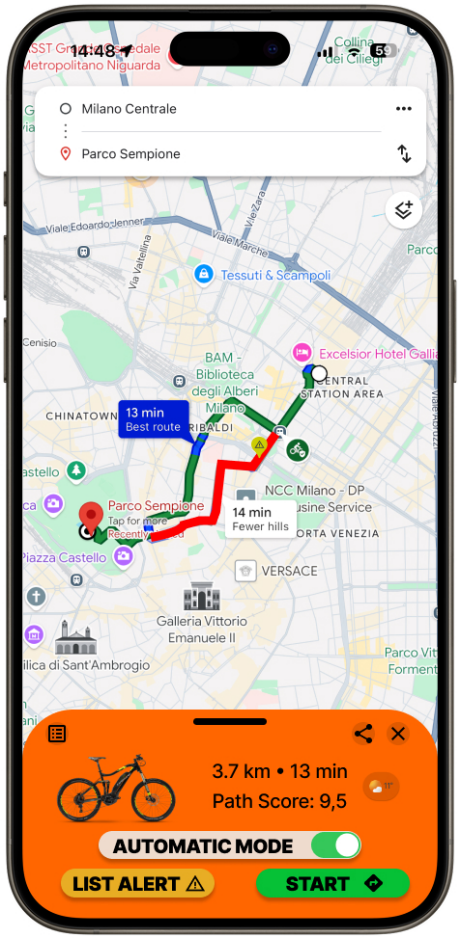


Figure 3: Main Map View and Route Selection

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.

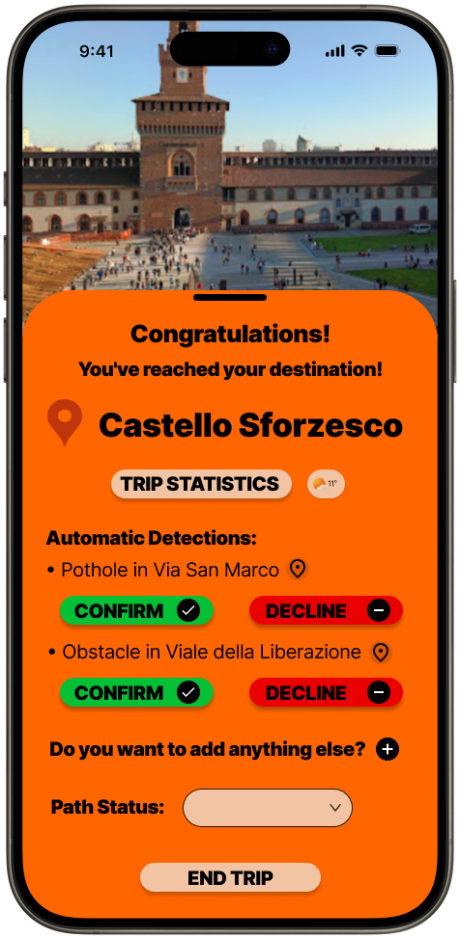


Figure 4: Confirmation and Data Validation Screen

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 (Fig. 5).

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 Requirements Traceability

The following matrix maps each functional requirement to the system components responsible for its realization. An 'X' indicates that the component participates in satisfying the requirement.

ID	Requirement	Mobile App	Acc. API GW	Trip API GW	Acc. Creation Svc	Acc. Deletion Svc	Cred. Subsys.	Acc. Subsys.	Act. Handler Svc	Act. Storage Subsys.	Trip Planner Svc	Path Subsys.	Pub. Info Subsys.	Weather Proxx	Email Proxy
R1	Create an account	X	X		X		X	X							X
R2	Log in	X	X				X	X							
R3	Reset password	X	X				X								X
R4	Update profile info	X	X					X							
R5	Delete account	X	X			X	X								X
R6	Start recording	X		X								X			
R7	Pause/Resume	X													
R8	Track pos. & stats	X							X						
R9	Retrieve weather			X					X					X	
R10	Report path status	X		X									X		
R11	Submit feedback	X		X					X				X		
R12	Report problems	X		X									X		
R13	Enable auto-detect	X													
R14	Analyze anomalies	X													
R15	Review anomalies	X													
R16	Confirm anomaly	X		X					X				X		
R17	Compute routes	X		X							X	X	X		
R18	Visualize routes	X									X	X			
R19	Path Score										X		X		
R20	Display obstacles	X		X									X		
R21	Filter search	X		X							X	X			
R22	View list	X		X						X					
R23	View trip details	X		X						X		X			
R24	Delete activity	X		X						X					
R25	Search activity	X		X						X					
R26	Filter history	X		X						X					

5 Implementation, Integration and Test plan

5.1 Implementation Plan

The system architecture allows for parallel development. However, adhering to the **Critical Component Strategy**, prioritizing high-risk and foundational modules ensures that the core value proposition is secured early in the lifecycle.

5.2 Component Integration Analysis

The following table classifies system components based on the **Impact of Failure**. The scale ranges from 1 to 5.

Component	Rationale for Classification	Impact	Dev. Order
Trip API Gateway	Single point of entry for all core features. If down, the App cannot communicate with the backend.	**5**	1
Path Subsystem	Provides map data and road network topology. Without it, no visualization or routing is possible.	**5**	1
Trip Planner Service	Core business logic. The primary goal of "Best Bike Paths" is finding routes. Without it, the app loses its purpose.	**5**	2
Credentials Subsystem	Manages Authentication (Login) and Security. Critical for user access, though Guest Mode bypasses it.	**4**	2
Account Subsystem	Aggregates *Creation*, *Deletion*, and *Management* services. Essential for user persistence, but Guest Mode mitigates total failure.	**4**	2
Activity Handler Service	Responsible for tracking rides. High value, but users could theoretically still use the map for visual navigation without active recording.	**4**	3
Account API Gateway	Entry point for Auth/Profile operations. If down, login and registration fail, limiting the app to Guest features.	**4**	3
Activity Storage Subsystem	Manages history. Failure prevents viewing past trips, but does not block new rides (which can be buffered locally).	**3**	4
Public Info Subsystem	Crowdsourcing logic (Issue Reporting, Scoring). Important for data enrichment, but the system functions without community reports.	**3**	4
Weather Proxy Service	Auxiliary feature. Lack of weather data reduces UX quality but does not stop any core function.	**2**	5
Email Proxy Service	Used only for password reset or confirmation emails. Failure impacts a tiny fraction of daily user interactions.	**1**	5

Table 2: Component Criticality and Implementation Priority

Impact Severity Legend:

- **5** **- Critical:** Complete system outage or total loss of the application's core purpose.
- **4** **- High:** Major functionality unavailable, but the system may offer degraded service.
- **3** **- Moderate:** Secondary features unavailable; core flow remains intact.
- **1-2** **- Low:** Auxiliary information missing or minor workflows blocked.

6 Effort Spent

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

Table 3: Effort spent by each team member per section

Lo Conte Francesco

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

7 References