POLITECNICO
MILANO 1863

Software Engineering 2 project FLAVIO
RICCOBONO

BEST BIKE PATHS

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Flavio Riccobono
Version:	1.0
Date:	23-December-2025
Download page:	https://github.com/FlavioPC02/Riccobono
Copyright:	Copyright © 2025, FLAVIO RICCOBONO – All rights reserved

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

In recent years, cyclists have increasingly felt the need to travel safely within urban and suburban road networks. However, updated and reliable information on the actual conditions of cycle paths, their quality, potential obstacles, and the direct experiences of other cyclists is often lacking.
To address this need, a cyclists' association decided to develop Best Bike Paths (BBP), a system designed to increase safety and awareness within the cycling community.

1.1.1 Goals

- G1 — Cyclists have an accurate awareness of their cycling activities;
- G2 — Relevant information about bike paths is shared by cyclists with the community;
- G3 — Cyclists can choose routes suited to their safety and efficiency needs;
- G4 — Cyclists have access to reliable and up-to-date information on the current state of bike paths;
- G5 — Information available to cyclists is consistent, consolidated, and trustworthy.

The purpose of the Best Bike Paths (BBP) system is to support cyclists in making safe and informed decisions by providing reliable, shared, and verifiable information on bike routes. The system enables users to choose the best route based not only on their personal recordings but also on information contributed by other cyclists.
The fundamental rationale for BBP's existence is therefore to improve cyclists' safety, quality of experience, and autonomy by offering a collaborative, informed, and continuously updated tool.

1.2 Scope

The purpose of this section of the document is to clearly define the real-world context in which Best Bike Paths (BBP) operates and establish the boundary between the real world and the machine. As cyclists interact with physical paths, experience road conditions, and share knowledge with the community, the system records and processes them. This analysis identifies relevant elements of the cycling domain, such as the state of bike paths, user activities, and shared community knowledge, and distinguishes them from the software's internal processes.

1.2.1 World phenomena

- WP1: Cyclists ride along bike paths;
- WP2: The physical state of bike paths changes over time (wear, potholes, obstacles, maintenance);
- WP3: Environmental and meteorological conditions evolve independently of the system;
- WP4: Cyclists form subjective experiences and evaluations of the routes they travel;
- WP5: The cyclist's physical movement generates measurable real-world parameters, which are detected by the sensors of the user's device (GPS, accelerometer, gyroscope);
- WP6: Multiple cyclists independently observe different aspects and conditions of the same path.

1.2.2 Shared phenomena

- World controlled:
  - SPW1: The user enters registration data;
  - SPW2: The user enters information about the status of a bike path;
  - SPW3: The user selects a start and a finish point for route discovery;
  - SPW4: The device's sensors (GPS, accelerometer, gyroscope) send measurements resulting from the cyclist's movement to the system;
  - SPW5: The user publishes recorded information to make it available to the community.
- Machine controlled:
  - SPM1: The system displays any errors or confirmation of registration/login;
  - SPM2: The system displays information about the status of cycle paths;
  - SPM3: The system displays the information it has collected about the trip (average speed, distance traveled, etc.);
  - SPM4: The system displays recommended routes between the selected origin and destination;

1.3 Definitions, Acronyms, and Abbreviations

BBP Best Bike Paths, the software system described in this document.

User Any person who interacts with BBP to browse or use information about bike paths. Users can be registered or not registered.

Registered User A user who has created an account in BBP and can record trips and insert information about bike paths.

Trip A single cycling activity performed by a registered user and recorded by the system.

Bike Path A route intended for bicycle traffic, possibly including roads with low car traffic and speed limits compatible with average cycling speed.

Path Status A qualitative evaluation of a bike path (e.g., optimal, medium, sufficient, requires maintenance).

World The real environment in which cyclists, roads, bike paths and all physical and social phenomena exist independently of BBP.

Machine The BBP software system, including its internal data, processes and interfaces.

Shared Phenomena Phenomena that involve both the World and the Machine, and that cross the boundary between users/devices and BBP.

GPS Global Positioning System, used to obtain the geographical position of the user.

UML Unified Modeling Language, used to describe and model the system.

RASD Requirements Analysis and Specification Document.

Version	Date	Author(s)	Description
0.1	*23/12/2025*	*Flavio Riccobono*	First Releas.

Table 1: Revision history of this document

1.4 Revision History

1.5 Reference Documents

The following documents and resources have been used as references for this RASD:

- **Assignment RDD A.Y. 2025–2026**: "Requirement Engineering and Design Project: Best Bike Paths (BBP)".
- Lecture slides on *Requirements Engineering (RE)* for Software Engineering 2.
- Lecture slides on *Creating a RASD* and the RASD template provided for the course.
- Lecture slides on *UML for Requirements Engineering*.

1.6 Document Structure

This document is structured as follows:

- **Section 1 – Introduction**
  Introduces the BBP project, its purpose and goals, the scope of the system, the main definitions and abbreviations used throughout the document, the revision history, reference documents and the overall structure of the RASD.
- **Section 2 – Overall Description**
  Describes the application domain and the context in which BBP operates. It includes the product perspective (scenarios, domain concepts and world/machine/shared phenomena), the main product functions, the characteristics of the intended users, and the assumptions, dependencies and constraints that affect the system.
- **Section 3 – Specific Requirements**
  Provides a detailed specification of the functional and non-functional requirements of BBP. It includes external interface requirements, functional requirements organised around use cases and related UML diagrams, as well as performance requirements, design constraints and software system attributes.
- **Section 4 – Formal Analysis Using Alloy**
  Presents the formal model (if provided) used to capture selected aspects of the requirements. It describes the objectives of the modelling activity, the Alloy model and the properties that are checked through it.
- **Section 5 – Effort Spent**
  Reports the amount of effort devoted by each group member to the preparation of this document.
- **Section 6 – References**
  Lists all bibliographic references, standards, and additional documentation cited in the RASD.

2 Overall Description

2.1 Product perspective

2.1.1 Scenarios

Scenario 1: Access as a non-registered user A cyclist who wants to access the app without creating an account. From the homepage, users can access the general features offered by BBP, such as finding a route between a starting point and a destination, and can explore the map and inspect available bike paths, viewing information such as their approximate location and current status. However, there are limitations: users cannot record trips, save personal data, or contribute new information.

Scenario 2: User registration A cyclist decides to create an account on BBP in order to record their trips and contribute information about bike paths. After accessing BBP, the user chooses the option to register and is asked to provide the required personal data and credentials. The user enters the requested information, such as an e-mail address, a password and a nickname, and confirms the registration. BBP validates the provided data and creates a new registered user associated with that information. From this moment on, the cyclist can log in using their credentials and access additional functionalities, such as trip recording and the insertion of information about bike paths.

Scenario 3: Record a trip and view statistics A registered user wants to track a new cycling activity. After logging into BBP, the user decides to create a new trip and then begins recording the trip. Once the trip is complete, the user stops recording. BBP stores the trip in the user's personal history and calculates statistics such as total distance and average speed. The user can instantly view a summary of the recorded trip and access a list of previous trips to review these statistics and track their cycling activity over time.

Scenario 4: Insert and publish information about a bike path The registered user logs in to BBP and searches for the corresponding bike path in the system. BBP shows the basic information currently associated with that path. The user then chooses to add new information, including their status (e.g., optimal, medium, sufficient, requires maintenance, . . .) and the presence of relevant obstacles, for instance, potholes. Before storing the contribution, the BBP allows the user to decide whether this information should be made publishable to other users. If the user confirms publication, the new information becomes visible to all users when they inspect that bike path. If the user decides not to publish it, the information is stored, but is not shown as part of the public data available to the community.

Scenario 5: Search for a route between an origin and a destination A user, either registered or not, wants to find a suitable route to travel by bike between two locations. The user specifies an origin and a destination, and BBP analyses the available bike paths and other relevant information to compute one or more bike-friendly routes connecting the two points. The system then presents these routes on a map, together with details such as their approximate length and the status of the bike paths involved. The user compares the suggested routes and selects the one they consider most appropriate for their needs.

2.1.2 Class Diagram

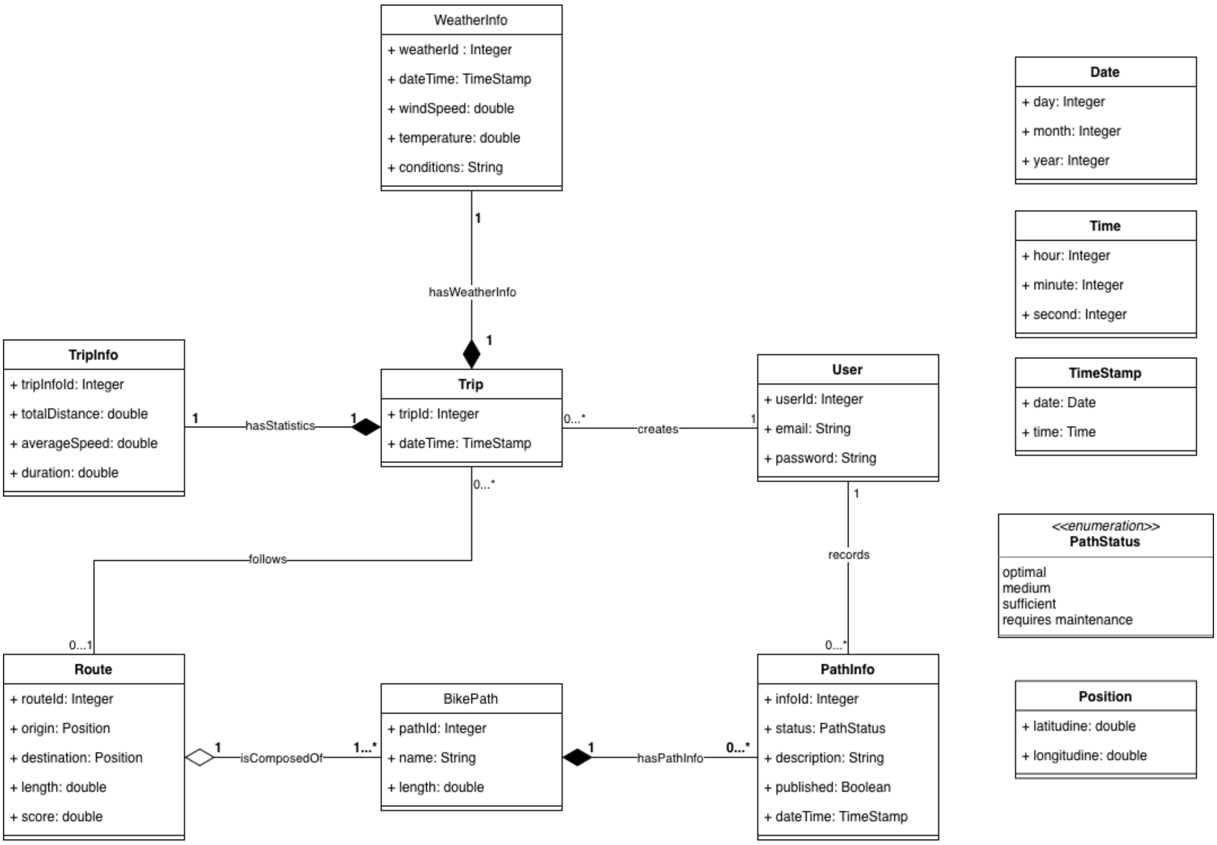

Figure 1: BBP Class Diagram

The domain of BBP is centered around registered users, their cycling activities, the bike paths, and the information shared within the community. A `User` may record several `Trip` instances, each representing a cycling activity carried out at a specific date and time. Every trip is associated with a corresponding `TripInfo`, which summarises the performance metrics of the ride (such as total distance, duration, and average speed), and may optionally include a `WeatherInfo` record providing contextual meteorological data retrieved from an external service. The urban environment is modelled through `BikePath` entities, each describing a bicycle-friendly segment with a name, an identifier, and a length. Users can contribute information about the current conditions of bike paths by creating `PathInfo` records that include a qualitative status, a textual description, and a publishable flag; the status itself is described through the enumerated type `PathStatus`. The system also models `Route` entities, representing suggested cycling routes between an origin and a destination, each composed of one or more bike paths, reflecting how the system combines existing infrastructure to propose safe and suitable paths. Together, these entities capture the essential structure of BBP's domain, describing users, their activities, the bike network, and the shared information that supports route computation and community awareness.

2.1.3 State Diagram

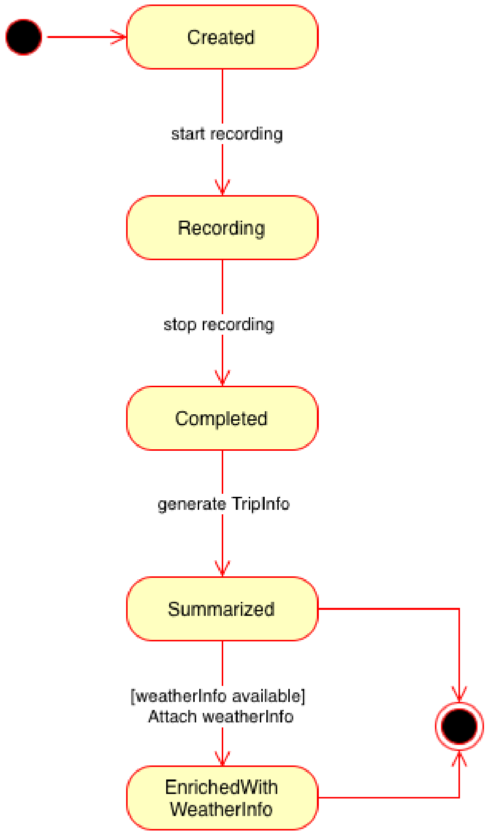

Figure 2: BBP State Diagram

The lifecycle of a `Trip` is characterized by a sequence of states that describe how a cycling activity is recorded and enriched with additional information. When a trip is created, it is initially in the *Created* state. When the user starts recording, the trip moves to the *Recording* state, during which BBP collects the data needed to compute statistics. When the user stops the recording, the trip enters the *Completed* state. The system then computes the corresponding `TripInfo`, moving the trip to the *Summarized* state. If weather information is available for the time and area of the trip, the system attaches a `WeatherInfo` record and the trip reaches the *EnrichedWithWeather* state. Both *Summarized* (when no weather information is found) and *EnrichedWithWeather* act as final states of the trip lifecycle.

2.2 Product functions

User registration and authentication BBP shall allow cyclists to register or authenticate, if already registered, by providing their personal information and required credentials, such as email and password. Registered users will have access to additional features, such as recording their trips and entering information about cycling routes.

Trip recording and history management BBP shall allow registered users to create a new Trip by starting the recording of a cycling activity. During a trip, the system collects the data needed to compute basic statistics. Once the recording is stopped, the system shall store the trip and present a summary including information such as total distance, duration and average speed. Users shall be able to view their past trips and related statistics.

Route search between an origin and a destination BBP shall allow users, either registered or not, to specify an origin and a destination and request bike-friendly routes connecting the two points. The system shall analyse the available bike paths and compute one or more candidate Routes by combining them into complete paths from origin to destination. Each suggested Route shall be visualised on the map together with basic information such as its approximate length and the condition of the involved bike paths.

Map browsing and inspection of bike paths BBP shall allow any user, either registered or not, to browse the map of the metropolitan area and inspect the available bike paths. For each bike path, the system shall display its location, length, the consolidated status, and all publishable PathInfo entries contributed by the community.

Route ranking and selection For each set of candidate Routes between the same origin and destination, BBP shall assign a score to every Route by combining the status of the bike paths composing it with measures of the route's effectiveness in connecting origin and destination (e.g., total distance). The system shall present the suggested Routes ordered according to their score, allowing the user to compare them and select the one they consider most appropriate for their needs.

Insertion and publication of path information BBP shall allow registered users to insert information about the condition of specific bike paths, including a qualitative status and an optional textual description. Before storing the information, the system shall let the user decide whether the new data should be made publishable to the community. Publishable information shall be visible to all users when they inspect the corresponding bike path.

Enrichment of trips with contextual weather information When possible, BBP shall retrieve, from an external service, weather information relevant to a recorded trip (for example, temperature and general conditions) and associate it with the trip summary, so that users can review their cycling activities together with the corresponding environmental conditions.

2.3 User characteristics

Non-registered users These users do not create a BBP account. They can browse the map, view bike paths and their conditions, and request route suggestions between a starting point and a destination. Since they are not authenticated, they cannot log trips or add information about the condition of bike paths. Their needs primarily concern quick access to information collected by BBP.

Registered users These users have created an account and authenticated in the system. In addition to browsing bike paths and requesting routes, they can record their cycling trips and insert information about the condition of specific bike paths. Their needs include accurate recording of activities, clear visualisations of trip statistics, and simple mechanisms to contribute information to the community.

General user characteristics All users interact with BBP through devices equipped with GPS and other sensors (e.g., accelerometer, gyroscope), which the system may use to collect trip-related measurements.

2.4 Assumptions, dependencies and constraints

Domain Assumptions

- **DA1:** BBP assumes that the user's device provides reasonably accurate GPS and sensor measurements (e.g., accelerometer, gyroscope) within normal operational error margins.

- **DA2:** The system assumes that an active internet connection is available whenever needed to retrieve map data, weather information, and to synchronise user contributions.
- **DA3:** BBP assumes that registered users provide truthful and meaningful informationwhen submitting new PathInfo records about bike path conditions.
- **DA4:** As advanced data consolidation is out of scope for single–member groups, the current status of a bike path is assumed to correspond to the most recent publishable PathInfo associated with that path.
- **DA5:** Suggested Routes are constructed by combining bike paths from the BBP inventory; non bike-friendly segments of the urban network are not explicitly modelled.
- **DA6:** BBP depends on external map and weather services, which are assumed to be reachable and functioning when invoked.

Dependencies and constraints

- BBP depends on external APIs for maps and weather retrieval.
- The project scope for single–member groups excludes advanced consolidation of multiple PathInfo into a unified status.
- Only bike-friendly segments of the urban network are represented in the domain model.

3 Specific Requirements

3.1 External Interface

3.1.1 User Interfaces

BBP allows the user to use the following interfaces:

- **Login, Registration and Guest Access**
  The system shall provide an interface where users can:
  - log in by entering their credentials (email and password);
  - create a new account by providing the required registration data;
  - continue as guest, without creating an account. Guest users can use read-only features such as browsing the map, inspecting bike paths and requesting routes, but cannot record Trips or insert PathInfo.
- **Home / Route Search View**
  The home view shall be accessible to both registered users and guests and shall include:
  - two input fields to specify an origin and a destination;
  - a command to request the computation of bike-friendly routes between these two points;
  - a list of suggested Routes ordered by their score, where each Route shows at least its approximate length and its score.
- **Route Map View**
  By selecting a Route from the list, the user is taken to a dedicated map view where the chosen Route is displayed. The Route map view shall:
  - show the selected Route on a map;
  - visually distinguish the different bike paths composing the Route, for instance by colouring them according to their current status;
  - allow the user to inspect the status of an individual bike path by selecting it, showing the most recent consolidated information about that path and all publishable PathInfo entries associated with it.
- **Trip List and Trip Detail View**
  The system shall provide a dedicated view where a registered user can:
  - see a list of all Trips previously recorded by that user;
  - select a Trip to open a detail view showing its full summary, including distance, duration, average speed and, when available, associated weather information.

  From the Trip list view, the user shall also be able to start recording a new Trip. While recording, a dedicated panel shall display live indicators such as elapsed time and an approximation of the distance travelled; when the recording is stopped, the Trip is stored and appears in the list.
- **Path Information View**
  The system shall provide a separate view to manage a registered user's own path information. In this view, a registered user shall be able to:
  - search for a specific bike path managed by BBP;

  - inspect all PathInfo entries previously inserted by that user for the selected bike path, including both publishable and non-publishable entries, with their qualitative status and textual description;
  - add a new PathInfo for the selected bike path by:
    * choosing a qualitative status (e.g. optimal, medium, sufficient, requires maintenance);
    * optionally entering a short textual description;
    * deciding whether the new information should be made publishable to the community.

3.1.2 Hardware Interfaces

BBP is designed to operate on devices equipped with basic sensing and connectivity capabilities. In particular, the system relies on the device's GPS module, which provides geographical positioning data required both for recording Trips and for computing routes based on the user's location. Furthermore, a network interface (such as Wi-Fi or mobile data) is required to enable communication with external services, including map providers, weather APIs and the backend used to synchronise user data.

3.1.3 Software Interfaces

BBP interacts with a set of external software services that support its core functionalities. A map service is used to visualise the metropolitan area, display bike paths and assist in computing bike-friendly routes between an origin and a destination. Additionally, BBP accesses an external weather service to obtain contextual meteorological information associated with recorded Trips, enriching trip summaries with temperature, conditions and other relevant data retrieved at the corresponding time and location.

3.1.4 Communication Interfaces

BBP relies on standard mobile communication technologies to interact with external services and synchronize data. All network exchanges between the application and remote servers occur via HTTPS, ensuring the secure transmission of user credentials, travel data, and route information. The system operates transparently over the device's available connectivity, both Wi-Fi and mobile data (4G/5G), and assumes an active internet connection whenever BBP needs to retrieve map tiles, calculate routes, access weather information, or load user-generated data.

3.2 Functional Requirements

- R1: The system shall allow users to register by providing the required personal data, including an email address and a password;
- R2: The system shall allow registered users to log in using their credentials;
- R3: The system shall allow users to continue as guests without creating an account;
- R4: The system shall prevent guest users from accessing functionalities that require authentication (Trip recording, PathInfo insertion);
- R5: The system shall allow registered users to access their travel history;
- R6: The system shall allow registered users to select a trip and view the recorded information for that trip;
- R7: The system shall allow registered users to create a new trip;
- R8: The system shall allow registered users to start recording data for a new trip;

- R9: The system shall allow registered users to complete a trip;

- R10: While a Trip is being recorded, the system shall acquire the user's geographical position through the device's GPS;

- R11: During recording, the system shall compute information such as average speed and total distance;

- R12: At the end of the recording, the system shall store the Trip and its computed informations;

- R13: The system shall allow all users to enter an origin and a destination;

- R14: The system shall compute bike-friendly Routes connecting the origin and destination using the available bike paths;

- R15: For each Route, the system shall compute a score based on the condition of the bike paths composing it and on the effectiveness of the Route;

- R16: The system shall show the routes sorted by score;

- R17: The system shall allow all users to select a route from those shown to view it on the map;

- R18: The system shall allow all users to select a bike path on the map and view its consolidated status and all publishable PathInfo entries associated with it;

- R19: The system shall allow registered users to search for a bike path in the Path Information View;

- R20: The system shall allow registered users to select a bike path in the Path Information View and inspect all PathInfo entries previously inserted by that user for that bike path, including both publishable and non-publishable ones;

- R21: The system shall allow registered users to enter a new PathInfo for a selected bike path;

- R22: The system shall store each PathInfo and associate it with the contributing user and with the corresponding bike path;

- R23: If the user marks a PathInfo as publishable, it shall become visible to all users when they inspect the corresponding bike path on the map;

- R24: If the user does not mark a PathInfo as publishable, the information shall be stored but not shown to other users; it shall only be visible to the contributing user in the Path Information View.

3.2.1 Use Case Diagram

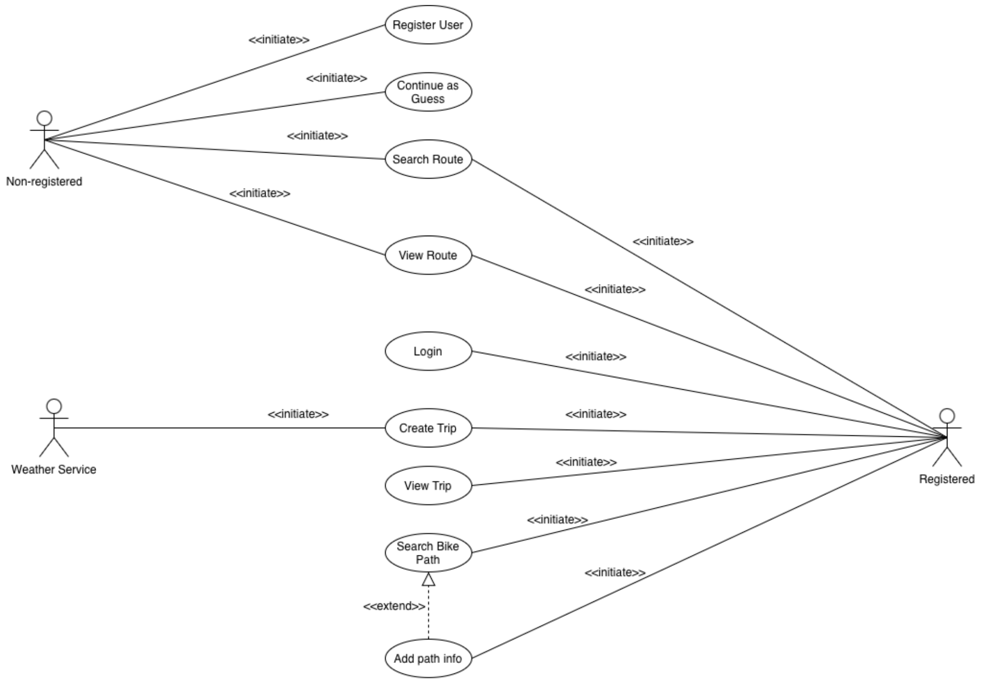

Figure 3: BBP Use Case Diagram

3.2.2 Use cases

Use Case Name	**Register User**
Participating Actors	Non-registered users
Entry Condition	The user is not authenticated and chooses the registration option in BBP.
Flow of Events	1. The user selects the option to register in BBP. 2. The system displays a registration form asking for the required data (e.g. email and password). 3. The user fills in the form with the requested information and submits it. 4. The system validates the data (e.g. checks that the email is well-formed and not already used). 5. If the data are valid, the system creates a new registered user associated with the provided information. 6. The system confirms the successful registration to the user.
Exit Condition	The user has successfully registered to the system.
Exception	• The provided data are invalid or incomplete. • The email address is already associated with an existing account.

Use Case Name	**Log In**
Participating Actors	Registered user
Entry Condition	The user has a valid BBP account and chooses the login option.
Flow of Events	1. The user selects the login option in BBP. 2. The system displays the login form. 3. The user enters email and password and submits the form. 4. The system validates the credentials. 5. If the credentials are valid, the system authenticates the user.
Exit Condition	The user is authenticated and can access functionalities reserved to registered users.
Exception	• The email and/or password are invalid.
Use Case Name	**Continue as Guest**
Participating Actors	Non-registered users
Entry Condition	The user opens BBP and chooses to continue without registering or logging in.
Flow of Events	1. The user selects the "Continue as guest" option. 2. The system grants access to BBP in guest mode. 3. The system enables functionalities available to guests, such as route search and route visualisation.
Exit Condition	The user is recognised as a guest and can use BBP with limited capabilities.
Exception	No specific exception.
Use Case Name	**Search Routes Between Origin and Destination**
Participating Actors	Non-registered user, Registered user
Entry Condition	The user (guest or registered) has access to BBP's home and opens the route search functionality.
Flow of Events	1. The user specifies an origin and a destination. 2. The system validates the input (e.g. checks that both points are valid locations). 3. The system retrieves the set of bike paths relevant for the requested area. 4. The system computes one or more candidate Routes that connect origin and destination using available bike paths. 5. For each Route, the system evaluates a score based on the status of the involved bike paths and on route effectiveness. 6. The system displays the list of Routes, ordered by score.
Exit Condition	A list of candidate Routes between origin and destination is presented to the user.
Exception	• No suitable route can be found between origin and destination. • The origin or destination cannot be resolved or are invalid.

Use Case Name	**View Route on Map and Inspect Bike Paths**
Participating Actors	Non-registered user, Registered user
Entry Condition	At least one Route between origin and destination has been computed.
Flow of Events	1. The user selects one of the suggested Routes. 2. The system displays the selected Route on a map. 3. The user inspects individual bike paths composing the Route (e.g. by selecting or hovering them). 4. For each inspected bike path, the system shows its consolidated status and, when available, a short description, together with all publishable PathInfo entries associated with that bike path.
Exit Condition	The user has visualised a Route on the map and, if desired, inspected the status of its bike paths.
Exception	No specific exception.
Use Case Name	**Create and Record a Trip**
Participating Actors	Registered user
Entry Condition	The user is authenticated in BBP and has access to the Trips functionality.
Flow of Events	1. The user selects the option to create and start recording a new Trip. 2. The system creates a new Trip and begins collecting positional data from the device's GPS. 3. While the Trip is in progress, the system updates distance, duration and average speed. 4. When the cycling activity is completed, the user selects the option to stop recording. 5. The system stops collecting data and computes the final statistics of the Trip. 6. The system stores the Trip and its associated TripInfo in the user's history. 7. When possible, the system retrieves contextual weather information and associates it with the Trip.
Exit Condition	A new Trip, with its summary statistics and optional weather information, is stored and available in the user's history.
Exception	No specific exception.
Use Case Name	**View Trip History and Trip Details**
Participating Actors	Registered user
Entry Condition	The user is authenticated and has recorded at least one Trip.
Flow of Events	1. The user opens the Trip history section in BBP. 2. The system displays a list of Trips previously recorded by the user. 3. The user selects one Trip from the list. 4. The system shows the detailed summary of the selected Trip, including distance, duration, average speed and, when available, weather information.
Exit Condition	The user has consulted the details of at least one recorded Trip.
Exception	• No Trip is available for the user (empty history).

Use Case Name	**Search Bike Path**
Participating Actors	Registered user
Entry Condition	The user is authenticated and opens the Path Info functionality.
Flow of Events	1. The user enters a bike path name. 2. The system retrieves and displays the bike path. 3. The user select the bike path. 4. The system confirms the selected bike path and displays, in the Path Information View, all PathInfo entries previously inserted by the user for that bike path, both publishable and non-publishable.
Exit Condition	The selected bike path is shown in the Path Information View together with the user's existing PathInfo entries and is ready to be updated with new information
Exception	• No bike path matches the search criteria.
Use Case Name	**Add Path Information**
Participating Actors	Registered user
Entry Condition	The user is authenticated and has selected a bike path (UC8).
Flow of Events	1. The user selects the option to add new information for the selected bike path. 2. The system displays a form for PathInfo insertion. 3. The user specifies a qualitative status, optionally adds a description and chooses whether the information is publishable. 4. The user submits the new PathInfo. 5. The system validates the input. 6. If the input is valid, the system stores the PathInfo and associates it with the selected bike path and the user. 7. If the PathInfo is marked as publishable, the system makes it visible to all users when they inspect that bike path.
Exit Condition	A new PathInfo entry is stored; if publishable, it contributes to the information shown for the bike path.
Exception	• The data provided by the user are invalid or incomplete.

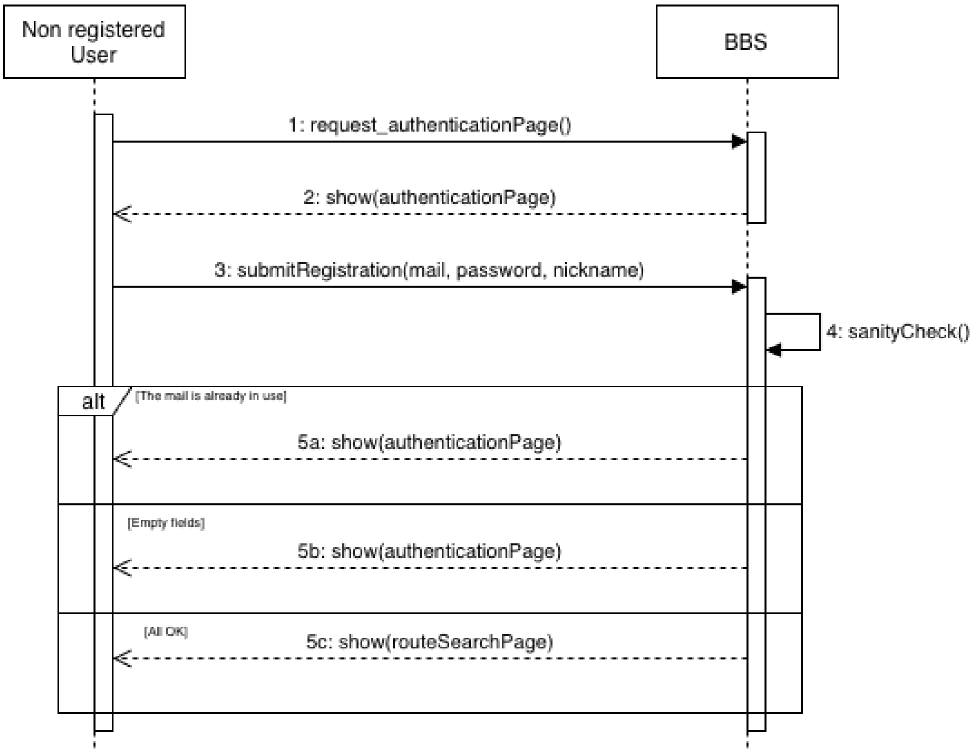
Non registered User
BBS
1: request_authenticationPage()
2: show(authenticationPage)
3: submitRegistration(mail, password, nickname)
4: sanityCheck()
alt
[The mail is already in use]
5a: show(authenticationPage)
[Empty fields]
5b: show(authenticationPage)
[All OK]
5c: show(routeSearchPage)

Figure 4: [UC1] Register User

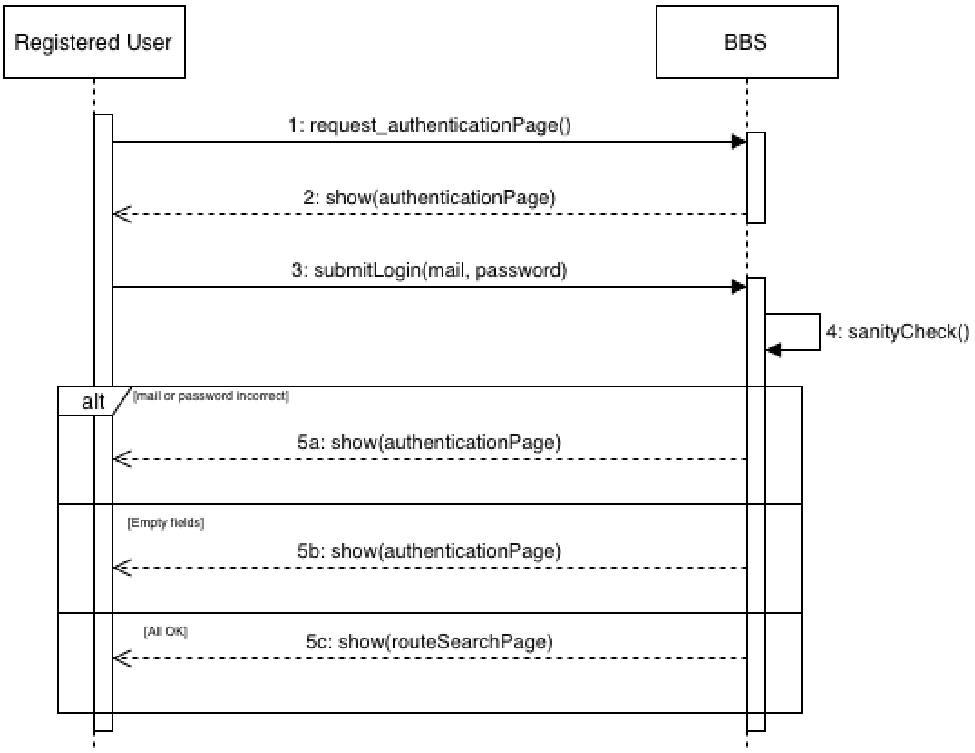
Registered User
BBS
1: request_authenticationPage()
2: show(authenticationPage)
3: submitLogin(mail, password)
4: sanityCheck()
alt
[mail or password incorrect]
5a: show(authenticationPage)
[Empty fields]
5b: show(authenticationPage)
[All OK]
5c: show(routeSearchPage)

Figure 5: [UC2] Login

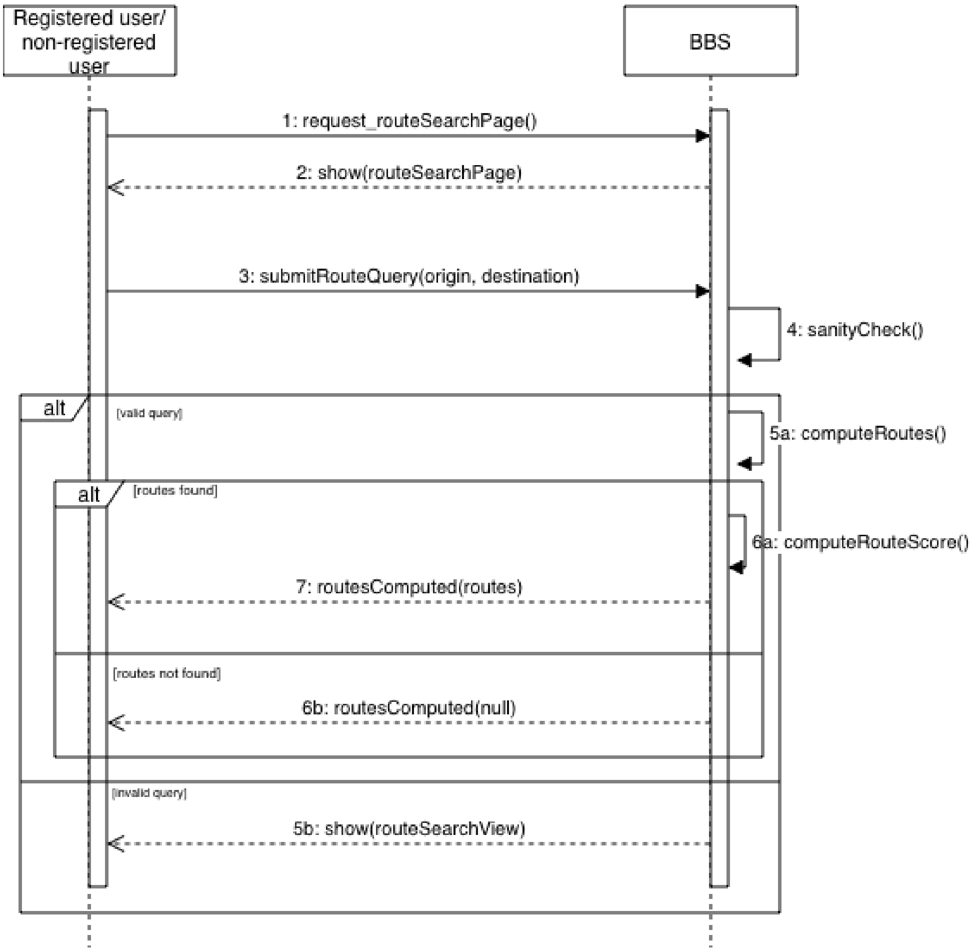

Figure 6: [UC4] Search Routes Between Origin and Destination

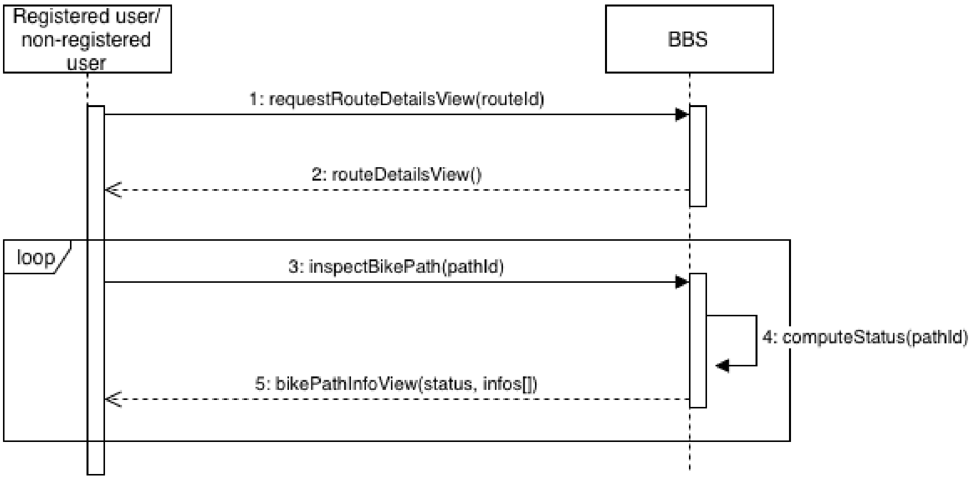
Registered user/ non-registered user
BBS
1: requestRouteDetailsView(routeId)
2: routeDetailsView()
loop
3: inspectBikePath(pathId)
4: computeStatus(pathId)
5: bikePathInfoView(status, infos[])

Figure 7: [UC5] View Route on Map and Inspect Bike Paths

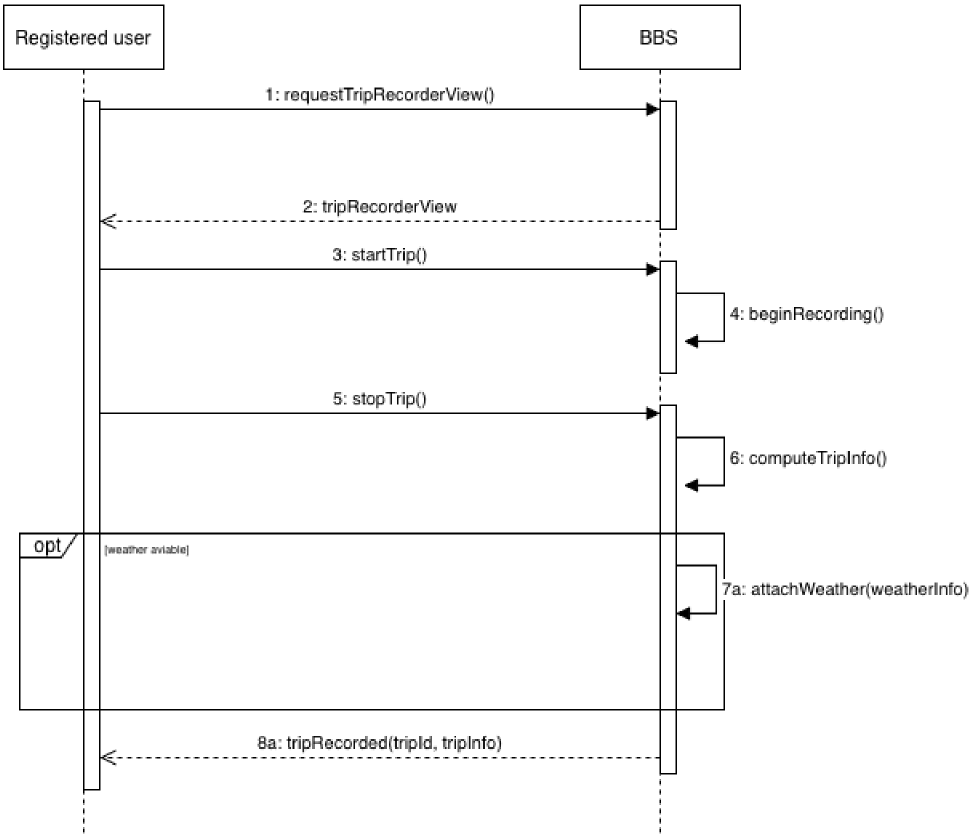

Figure 8: [UC6] Create and Record a Trip

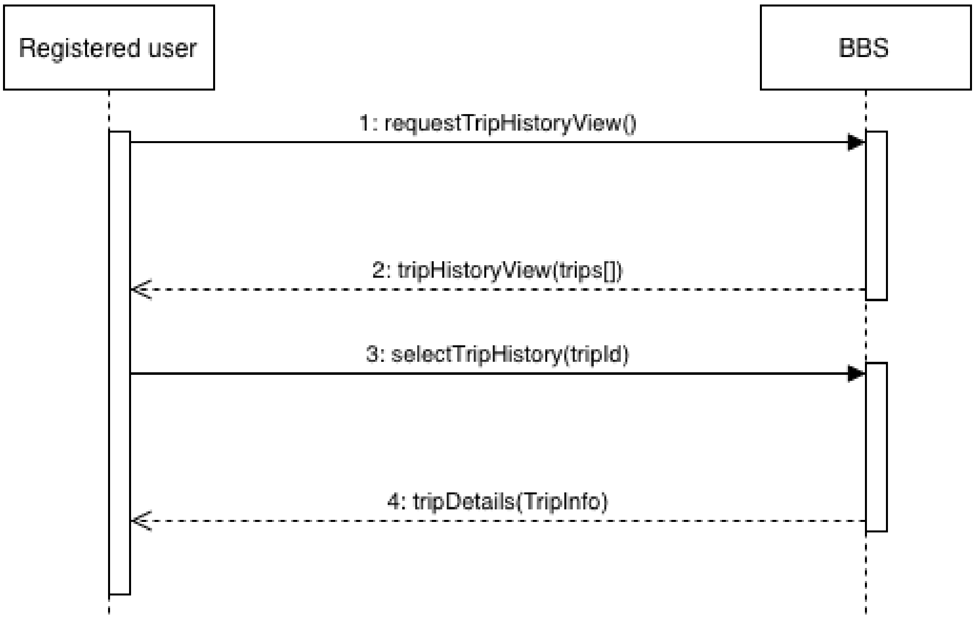
Registered user
BBS
1: requestTripHistoryView()
2: tripHistoryView(trips[])
3: selectTripHistory(tripId)
4: tripDetails(TripInfo)

Figure 9: [UC7] View Trip History and Trip Details

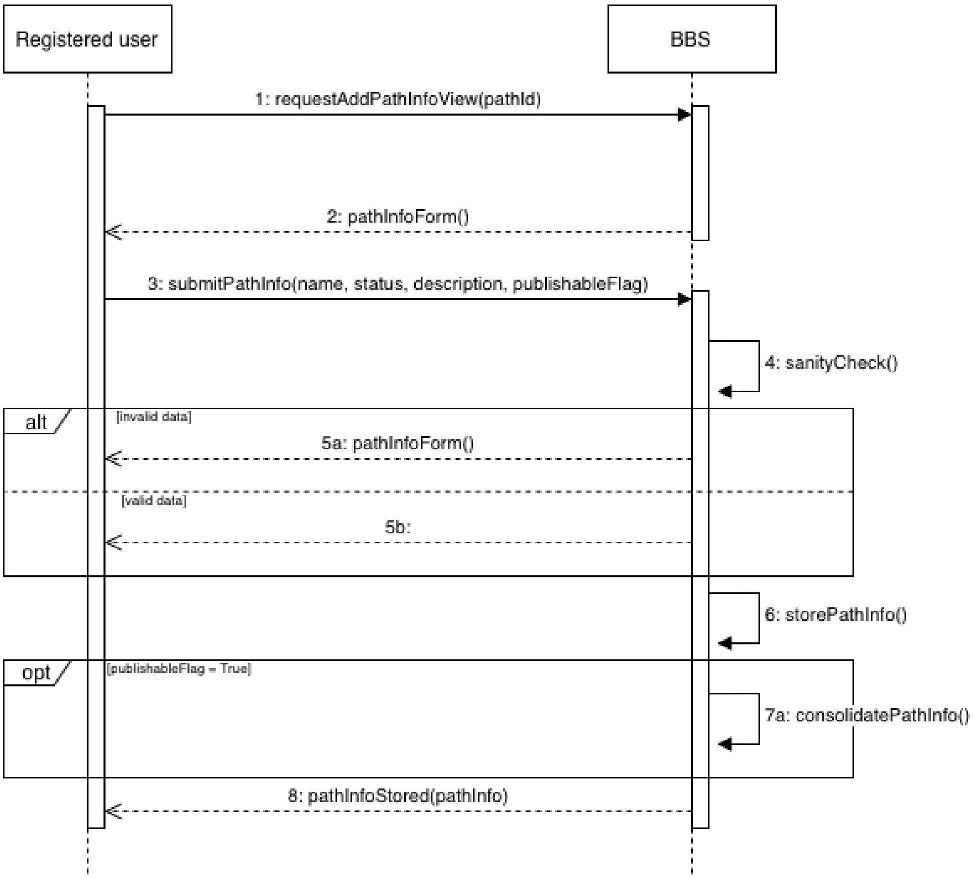
Registered user
BBS
1: requestAddPathInfoView(pathId)
2: pathInfoForm()
3: submitPathInfo(name, status, description, publishableFlag)
4: sanityCheck()
alt
[invalid data]
5a: pathInfoForm()
[valid data]
5b:
6: storePathInfo()
opt
[publishableFlag = True]
7a: consolidatePathInfo()
8: pathInfoStored(pathInfo)

Figure 10: [UC9] Add Path Information

3.2.3 Requirements mapping

G1: Cyclists have an accurate awareness of their cycling activities	
• **R1**: The system shall allow users to register by providing the required personal data, including an email address and a password; • **R2**: The system shall allow registered users to log in using their credentials; • **R5**: The system shall allow registered users to access their travel history; • **R6**: The system shall allow registered users to select a trip and view the recorded information for that trip; • **R7**: The system shall allow registered users to create a new trip; • **R8**: The system shall allow registered users to start recording data for a new trip; • **R9**: The system shall allow registered users to complete a trip; • **R10**: While a Trip is being recorded, the system shall acquire the user's geographical position through the device's GPS; • **R11**: During recording, the system shall compute information such as average speed and total distance; • **R12**: At the end of the recording, the system shall store the Trip and its computed informations;	• **DA1**: The user's device provides reasonably accurate GPS/sensor measurements; • **DA2**: An active internet connection is available when needed to synchronise data; • **DA6**: External map/weather services are reachable when invoked.
G2: Relevant information about bike paths is shared by cyclists with the community	
• **R18**: The system shall allow all users to select a bike path on the map and view its consolidated status and all publishable PathInfo entries associated with it; • **R19**: The system shall allow registered users to search for a bike path in the Path Information View; • **R20**: The system shall allow registered users to select a bike path in the Path Information View and inspect all PathInfo entries previously inserted by that user (publishable and non-publishable); • **R21**: The system shall allow registered users to enter a new PathInfo for a selected bike path; • **R22**: The system shall store each PathInfo and associate it with the contributing user and the bike path; • **R23**: If a PathInfo is marked as publishable, it becomes visible to all users on the map; • **R24**: If a PathInfo is not publishable, it is stored but visible only to its author in the Path Information View;	• **DA2**: Internet connectivity is available to share/retrieve contributions; • **DA3**: Users provide truthful and meaningful PathInfo; • **DA4**: Current BikePath status corresponds to the most recent publishable PathInfo (single–member simplification); • **DA6**: External services (back-end/APIs) are reachable when invoked.

G3: Cyclists can choose routes suited to their safety and efficiency needs	
• **R13**: The system shall allow all users to enter an origin and a destination; • **R14**: The system shall compute bike-friendly Routes connecting origin and destination using the available bike paths; • **R15**: For each Route, the system shall compute a score based on the condition of the bike paths composing it and on the effectiveness of the Route; • **R16**: The system shall show the routes sorted by score; • **R17**: The system shall allow all users to select a route from those shown to view it on the map;	• **DA4**: Route scoring uses the BikePath status derived from the most recent publishable PathInfo; • **DA5**: Suggested Routes are constructed by combining bike paths from the BBP inventory; • **DA2/DA6**: Connectivity and external routing/map services are available.
G4: Cyclists have access to reliable and up-to-date information on the current state of bike paths	
• **R18**: The system shall allow all users to select a bike path on the map and view its consolidated status and all publishable PathInfo entries associated with it; • **R23**: Publishable PathInfo becomes visible to all users when they inspect that bike path on the map;	• **DA4**: “Up-to-date” is realised by using the most recent publishable PathInfo as current status; • **DA2/DA6**: Connectivity and external services are available to fetch/refresh data.
G5: Information available to cyclists is consistent, consolidated, and trustworthy	
• **R18**: Map inspection shows the consolidated status of each BikePath; • **R22**: Each PathInfo is stored and associated with its author and bike path; • **R23**: Publishable PathInfo is visible to all users; • **R24**: Non-publishable PathInfo remains visible only to its author in the Path Information View;	• **DA4**: Consolidation rule = latest publishable PathInfo (project scope simplification); • **DA3**: Users provide information in good faith (lightweight moderation assumed).

3.3 Performance Requirements

- Response time: common operations (route search, map open, bike path inspect) $\leq$ **2 s** under normal conditions.
- PathInfo insertion: save and confirmation $\leq$ **2 s**.
- Trip recording: start/stop feedback $\leq$ **200 ms**; statistics computed at end $\leq$ **1 s**.

3.4 Design Constraints

3.4.1 Standards compliance

- REST APIs over HTTP/HTTPS, payload `JSON`, encoding `UTF-8`.
- Timestamps in **ISO 8601** (UTC); geographic coordinates in **WGS84** (lat/lng).
- Transport encryption **TLS 1.2+**.
- Compliance with **GDPR** for personal data.

3.4.2 Hardware limitations

- Smartphone with GPS and Internet connectivity.

- Sensor accuracy and battery are variable; Trip recording uses lightweight sampling.

3.4.3 Any other constraint

- Consolidated BikePath status = **latest publishable PathInfo**.
- Dependence on external services (maps/routing, weather): if unavailable, core read-only features remain usable.

3.5 Software System Attributes

3.5.1 Reliability

- No Trip data loss: local buffering and deferred sync when network is available.

3.5.2 Availability

- Availability goals as in the table above; graceful degradation when third-party services are down.

3.5.3 Security

- Authentication with password; encrypted traffic (TLS).
- Content visibility: *publishable* PathInfo visible to everyone; *non-publishable* PathInfo visible only to its author (in the personal view).

3.5.4 Maintainability

- Modular architecture (app, APIs, integrations); API versioning; basic error logging.

3.5.5 Portability

- App usable on Android and iOS; standard data formats (GeoJSON/WGS84) and provider abstraction for external services.

4 Formal Analysis Using Alloy

4.1 Signatures

```
sig User {}
sig BikePath {}

// Qualitative condition of a bike path
enum Status { optimal, medium, sufficient, requiresMaintenance }

// Visibility of a PathInfo (publishable or personal-only)
abstract sig Visibility {}
one sig Public, Private extends Visibility {}

// User-contributed information on a bike path (mutable over time)
var sig PathInfo {
  var author : one User,
  var path : one BikePath,
  var status : one Status,
  var vis : one Visibility,
  var t : one Int // logical timestamp
}
```

4.2 Facts

```
fact UniqueTimePerPath {
  always {
    all b: BikePath, ti: Int |
      lone { pi: PathInfo | pi.path = b and pi.t = ti }
  }
}
```

4.3 Functions and Predicates

```
// Predicate: pi is the latest publishable PathInfo on its path
pred isLatestPublishable[pi: PathInfo] {
  pi.vis = Public
  no pj: PathInfo |
     pj.path = pi.path and pj.vis = Public and pj.t > pi.t
}

// Consolidated status of a bike path (current state)
fun consolidated[b: BikePath]: lone Status {
  { s: Status |
      some pi: PathInfo |
        pi.path = b and isLatestPublishable[pi] and pi.status = s
  }
}
```

4.4 Assertions

```
assert ConsolidatedUsesLatestPublic {
  always {
    all b: BikePath |
      (some pi: PathInfo | pi.path = b and pi.vis = Public)
        implies some lp: PathInfo |
                  lp.path = b and isLatestPublishable[lp]
                  and consolidated[b] = lp.status
  }
}

assert PrivateDoesNotAffectConsolidation {
  always {
    all b: BikePath, pri: PathInfo |
      pri.path = b and pri.vis = Private
        implies consolidated[b] in
                { s: Status | some pi: PathInfo |
                                 pi.path = b and isLatestPublishable[pi]
                                 and s = pi.status }
  }
}

assert NoneIfNoPublic {
  always {
    all b: BikePath |
      no { pi: PathInfo | pi.path = b and pi.vis = Public }
        implies no consolidated[b]
  }
}

assert ConsolidatedDefinedIffPublic {
  always {
    all b: BikePath |
      (some pi: PathInfo | pi.path = b and pi.vis = Public)
        iff (some consolidated[b])
  }
}

assert LatestPublishableIsUnique {
  always {
    all b: BikePath |
      lone { pi: PathInfo | pi.path = b and isLatestPublishable[pi] }
  }
}
```

4.5 Predicates

```
pred addPathInfo[u: User, b: BikePath, s: Status, v: Visibility, time: Int] {
  some pi: (PathInfo' - PathInfo) |
    pi.author' = u and
    pi.path' = b and
    pi.status' = s and
    pi.vis' = v and
    pi.t' = time}
```

4.6 Run Show

Following are some models of the system obtained by running the model on Alloy Analyzer.

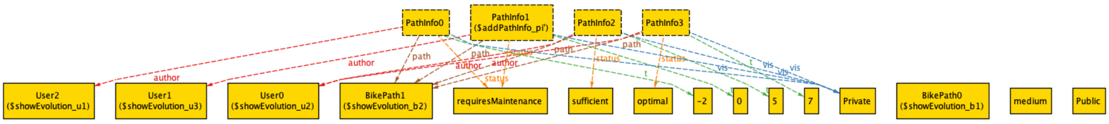

Figure 11: Step 1

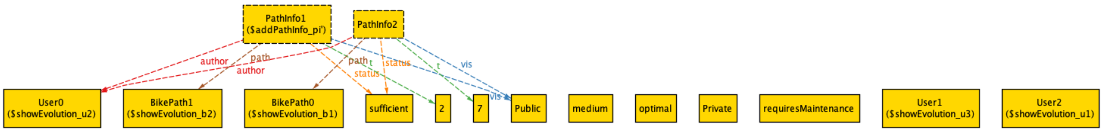

Figure 12: Step 2

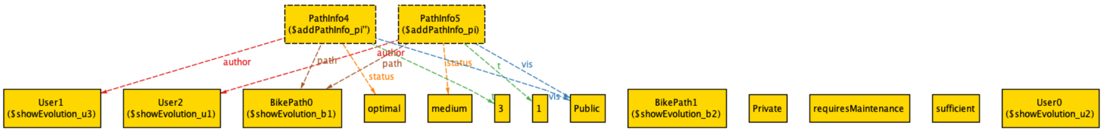

Figure 13: Step 3

5 Effort Spent

RASD Section	**Time spent (hours)**
Introduction	4
Overall Description	5
Specific Requirements	14
Formal Analysis Using Alloy	6
Total	**29**

References