Requirement Analysis and Specification Document

Best Bike Paths

Giuliano Crescimbeni

Luca De Nicola

POLITECNICO
MILANO 1863

Contents

1 Introduction

1.1 Purpose

In the world of cycling, it is often useful to record information about routes and track personal performance, as well as to share these experiences with others. Having access to updated data about bike pathssuch as their conditions, safety, and suitabilitycan greatly enhance both the enjoyment and safety of cyclists.

Best Bike Paths (BBP) was conceived in this context, with the goal of creating a digital platform where cyclists can explore, record, and share information about cycling routes. The system promotes collaboration among users, encouraging the community to contribute and maintain reliable data on the status of bike paths.

1.1.1 Goals

- **G1 - User Registration and Authentication:** The system shall allow users to register and authenticate themselves to access the full range of functionalities provided by the application.
- **G2 - Trip Recording and Manual Path Registration:** The system shall enable registered users to record their cycling trips, store them for activity tracking purposes, and manually insert bike path information by specifying street names and their current status.
- **G3 - Meteorological Data Integration:** The system shall enrich recorded trip data with meteorological information (including weather conditions, temperature, and wind speed) retrieved from external services, when available.
- **G4 - Automated Mobile Data Acquisition:** The system shall acquire data automatically from users' mobile devices during cycling activities, including GPS coordinates for path reconstruction and accelerometer/gyroscope data for obstacle detection (e.g., potholes).
- **G5 - User Validation of Automated Data:** The system shall require users to confirm or correct automatically acquired information before making it available to the community, ensuring data accuracy and minimizing false positives.
- **G6 - Community-Driven Path Information Publishing:** The system shall allow registered users to insert bike path information and make it publishable to the community, contributing to the shared knowledge base.

G7 - Universal Path Search and Map Visualization: The system shall allow any user (registered or non-registered) to specify an origin and destination, and visualize the available bike path(s) between these two points on an interactive map.

G8 - Path Scoring: The system shall compute a score for each bike path based on its current status and effectiveness in connecting origin to destination, and visualize paths to users ordered by their computed score.

G9 - Multi-Source Data Consolidation: The system shall merge bike path information collected from multiple users by considering data freshness and the number of confirming reports to determine the most accurate status assessment.

1.2 Scope

Best Bike Paths (BBP) is an application designed to support cyclists in discovering, recording, and sharing information about bike paths. Through BBP, registered users can record their rides, visualize the paths on a map, and obtain performance statistics such as distance, speed, and duration. When available, the system automatically integrates meteorological informationincluding temperature, wind speed, and weather conditions.

Users can insert path information either manually, by specifying the streets and their conditions, or automatically, by allowing the application to collect GPS, accelerometer, and gyroscope data during a trip. This data helps identify irregularities such as potholes or rough road segments. Before being published, automatically detected issues must be validated by the user to ensure reliability.

The platform provides map-based search and visualization tools allowing any user, registered or not, to explore available routes between a chosen origin and destination.

1.2.1 World Phenomena

WP1 - A cyclist rides along a bike path

WP2 - A cyclist seeks to discover a new route

WP3 - A person recommends a starting point, destination, or place to another cyclist

WP4 - Changes in weather conditions

WP5 - Physical deterioration of a bike path

WP6 - Appearance of an obstacle along a bike path

WP7 - Irregularities or bumps on the road surface cause vibrations on the cyclists device

WP8 - A user enables localization services

1.2.2 World-controlled Shared Phenomena

SP1 - A user registers or logs into the system by entering their credentials

SP2 - A cyclist records a bike route by manually entering route information (e.g. street names, status, obstacles ecc.)

SP3 - A cyclist publish (or make private) a previously saved route

SP4 - A cyclist starts or stops the automatic tracking mode of a ride

SP5 - A cyclist confirms or corrects the information acquired by the automatic tracking system

SP6 - A cyclist decides to publish a personal recorded route

SP7 - A user searches for available routes between two locations

SP8 - A user views the information of a specific bike route

1.2.3 Machine-controlled Shared Phenomena

SP9 - The system adds supplementary data after a route is submitted (such as total distance covered, average speed, and other performance metrics)

SP10 - The system contacts an external weather service

SP11 - The system enriches a route by adding weather data retrieved from the external weather service

SP12 - The system collects data from the cyclists device sensors and creates a new trip

SP13 - The system detects irregularities along the route and presents them to the user at the end of the trip

SP14 - The system assigns a score to each route based on its condition and its effectiveness in connecting the selected origin and destination

SP15 - The system displays the available routes on a map between two points based on their score

SP16 - The system merges data acquired from multiple users for the same route, based on data freshness and the number of consistent confirmations

1.3 Definitions

1.4 Acronyms

- BBP: Best Bike Paths
- UML: Unified Modeling Language

1.5 Abbreviations

- Gn: Goal number n
- Rn: Requirement number n
- Dn: Domain assumption number n
- WPn: World phenomena number n
- SPn: Shared phenomena number n
- UC: Use case

1.6 Revision History

1.7 Reference Documents

1.8 Document Structure

- **Introduction**: Presents the purpose and goals of the document, the context in which the system operates, and a brief overview of the software to be developed.

- **Overall Description**: Provides a high-level view of the system, describing its main functions, the type of users involved, and the constraints under which it operates.
- **Specified Requirements**: Details the functional and non-functional requirements of the system. This section includes use cases, scenarios, and detailed interactions between users and the system, as well as performance, reliability, and usability requirements.
- **Formal Analysis**: Describes the formal modeling of the systems behavior through conceptual tools such as UML diagrams, Alloy models, or other formal methods. This section ensures the consistency and correctness of the requirements.
- **Effort Spent**:Summarizes the amount of work dedicated to each phase of the project, specifying the contribution of each team member and the approximate time spent on the various activities.
- **Software Used**: Software used to develop the document.

2 Overall Description

2.1 Product Perspective

2.1.1 Scenarios

Cyclist registers on BPP: The cyclist Bob accesses BPP platform for the first time and decides to create an account. He opens the registration form and enters the required information, including his email address, password, and general personal details (such as name, surname, and date of birth). Once the registration is completed successfully, Bob gains access to the main dashboard of the platform, where he can start exploring available routes or contribute by creating his own.

Cyclist manually uploads a route: After completing a cycling trip from street A to street B, Bob logs into BPP and clicks on the Create New Route button on the homepage. Manually enters the details of the route - including street names, and possible obstacles - and add a short descriptions to help other cyclists. The system compute statistics like average speed, total distance and other metrics and enrich them with meteorological information. Before saving, Bob set path status to "Optimal" and decide to make the route public (visible to all users), enabling the "Publish" option. Once submitted, the route is stored in the system and appears in Bobs personal list of recorded paths. A score is given to the path based on the uploaded information.

Cyclist uploads a route using automatic mode: After enabling the location services on her device, Alice goes to her BPP profile settings and activate the automatic tracking mode. Some time later she goes for a bike ride while carrying her mobile device. During the ride, BPP automatically records the route, collecting data such as path segments, total distance, average speed, and detected obstacles. The system enriches with meteorological information. Once the trip is completed, Alice accesses the platform to review the automatically generated route. She confirms the details of the route, confirms the accuracy of the detected obstacles, and decides to keep the route private. Finally, she clicks the Save Route button to store the finalized version in her personal collection. A score is given to the path based on the uploaded information.

Unregistered user searches for routes: Eva, a user who is not registered on the platform, navigates to the Search section of BPP, which consists of an interactive map and a search panel. She enters Street A as the starting point and Street B as the destination. After clicking the Search button, several available routes instantly appear on the map connecting the two locations. Among them, the highest-rated route is the one previously uploaded by Bob. Eva clicks on this route to view its detailed information, including distance, path conditions, and reported obstacles.

Merging of route information: Alice and her friend, both registered users of BPP, manually record the same route from Street A to Street B. Each of them enters the route details and sets the path status to Sufficient. Since these submissions contain more recent data than the previously stored version of the route, the system automatically merges their information with the existing record. The route is updated with a path status of Sufficient, determined by majority consensus, and the newly provided details are incorporated to keep the route information up to date and more reliable.

Publication of a previously saved route: Alice, who had previously recorded a private route, decides to make it public. She navigates to the My Routes section, selects the specific route, and changes its visibility status to Public. Once confirmed, the route becomes available to all users and is automatically merged with existing records of the same path submitted by other cyclists. The system performs the merge based on the **upload date** of the route rather than its **publication date**, ensuring data consistency and chronological accuracy in the shared route database.

2.1.2 High-level class diagram

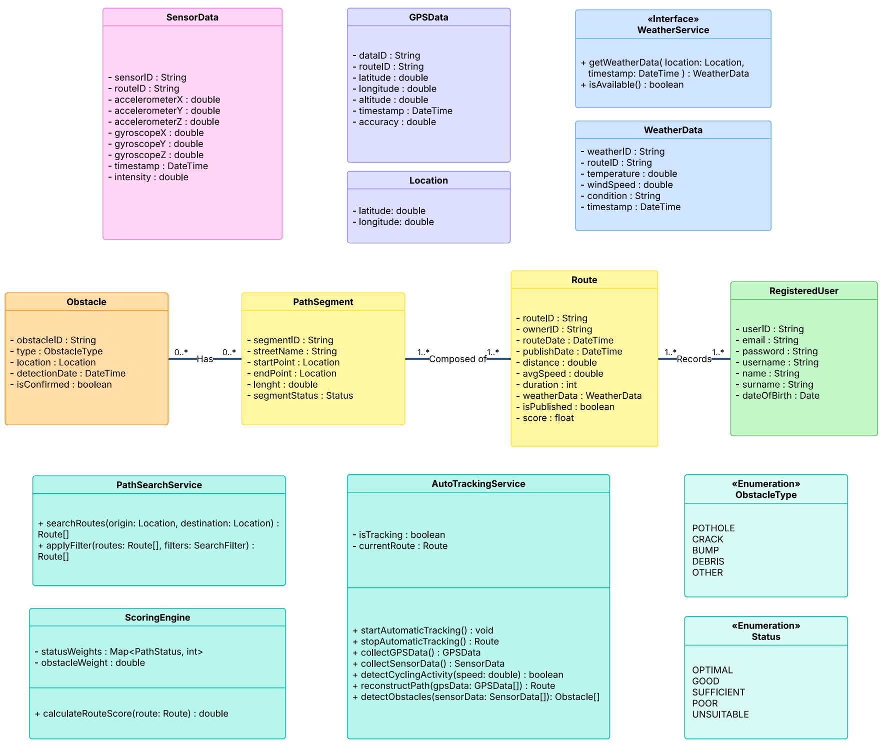

Figure 1: Class diagram of the BBP system

This diagram illustrates how the system manages cycling data. The **Route** entity is composed of multiple **Path Segment** objects. A distinct Status is identified for both entities: the segment status is explicitly defined by the user, while the overall route status is calculated based on the statuses of its composing segments. Additionally, users can identify and insert specific **Obstacle objects** associated with a segment.

Segment states never get overwritten: each PathSegment is the same physical stretch observed at a **different point in time**. When computing the score of a given Route, for every stretch that compose it the system gathers all matching segments for that same stretch coming from other routes, then derives the score from the majority status, weighted by how fresh the observations are.

2.1.3 State Diagrams

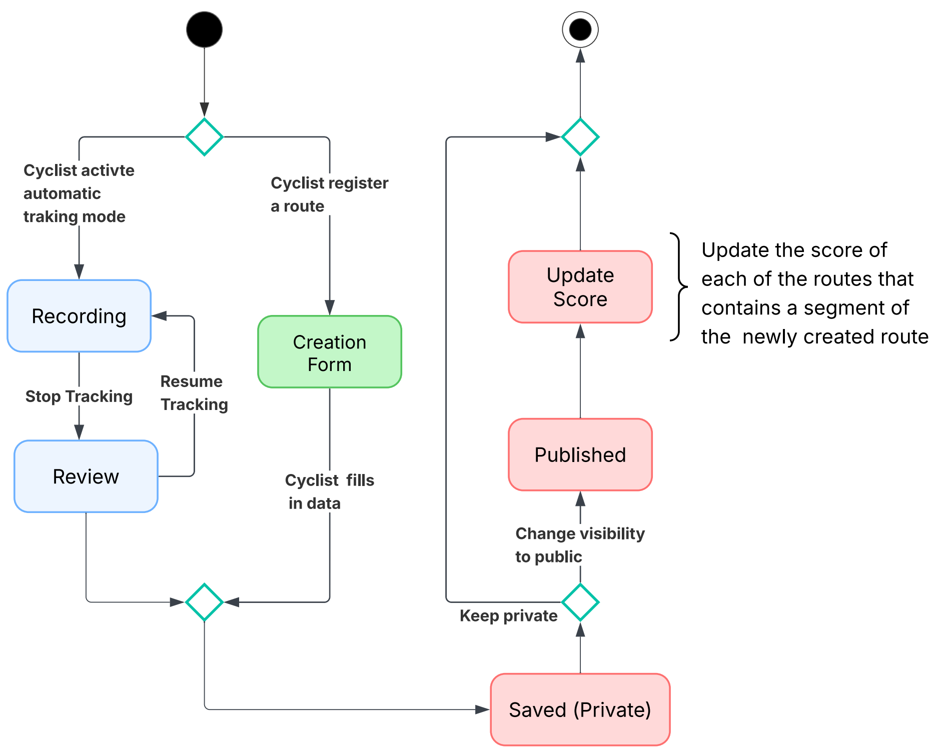

Figure 2: State diagram of the Route class

In the state diagram shown in **Figure 2**, the lifecycle of a single **Route** during its creation is described. It is important to note that once a new Route is created, the score of every existing Route that includes any of the path segments contained in the new one is recomputed and updated.

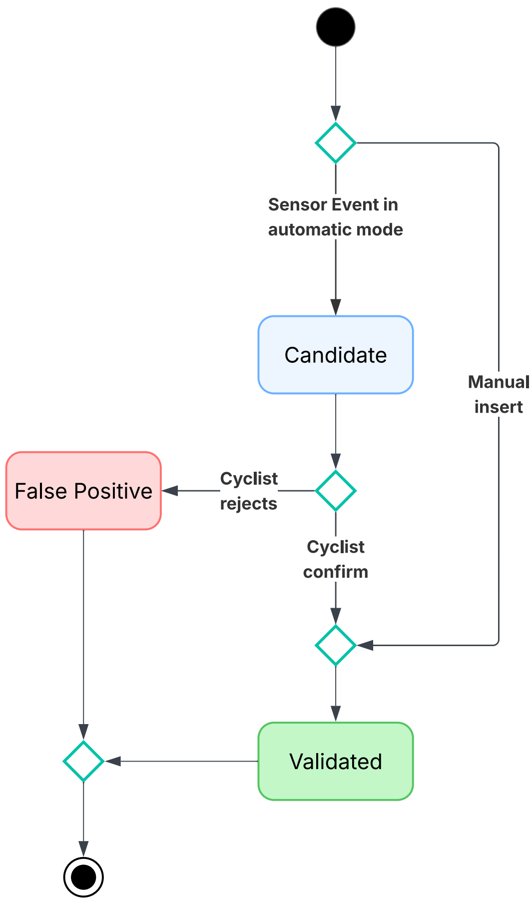

Figure 3: State diagram of the Obstacle class

In the state diagram shown in **Figure 3**, the evolution of a single **Obstacle** is illustrated, covering both manual insertion and automatic validation. An obstacle created through the automatic mode, of course, does not require a second verification step.

2.2 Product Functions

Registration and Login: This functionality allows cyclists to create a personal account on the platform to access advanced features. Registered users can manage their profile, maintain a personal history of recorded tracks, and contribute to the community by publishing their routes.

Route Recording and Data Acquisition: The system offers two distinct modes for recording cycling trips. Users can manually insert path information by specifying street names, current status, and obstacles. Alternatively, an automatic tracking mode utilizes the device's sensors (GPS, accelerometer, and gyroscope) to reconstruct the path and automatically detect irregularities or road issues. To enhance the quality of the information, the system automatically enriches recorded trips with meteorological data retrieved from external services. Furthermore, to minimize false positives from automatic detection, the system requires users to review and validate detected obstacles and path details before the data is stored or published.

Dynamic Scoring and Data Merging: The system automatically merges path information collected from multiple users, prioritizing data freshness and majority consensus to determine the most accurate status of a route segment . Based on this consolidated data, the system computes a specific score for each bike path, reflecting its safety and effectiveness.

Path Search and Visualization: The platform provides an interactive map interface that allows any user (registered or unregistered) to search for bike paths between a specific origin and destination. The search results visualize available routes ranked by their calculated score, allowing cyclists to easily identify the best and safest options for their ride.

2.3 User Characteristics

The BBP platform is designed to serve a diverse community of cyclists, from casual to more athletic. Users are categorized according to their interaction with the system: Registered cyclists and Unregistered cyclists.

2.3.1 Registered Cyclist

The registered cyclist is an active contributor willing to share personal ride data to support the community; therefore, he needs a system that enables him to easily publish routes and report road obstacles. Beyond contribution, he benefits from the community's information, requiring the capability to discover new paths and assess their conditions and safety scores derived from other users' data before starting. Since he interacts with the application primarily while riding,the automated data acquisition method could be require to minimize manual input and ensure safety during the activity. Furthermore, he plays a crucial role in maintaining overall data quality by reviewing and validating automatically detected anomalies before they are shared.

2.3.2 Unregistered Cyclist

The unregistered user is an individual who accesses the platform primarily for information retrieval without the need for performance tracking. His main goal is to find the best available bike path between two locations; therefore, he needs an immediate and accessible search tool that does not require authentication. Furthermore, he relies heavily on the systems scoring mechanism to select routes that are not only fast but also safe and in good condition.

2.3.3 Common Characteristics

Both user categories share a fundamental need for reliability. Whether recording a trip or searching for one, users rely on the accuracy of the Path Score and the map visualization to make decisions about their route. Both groups interact with map-based interfaces and require clear visual indicators of path status to plan their rides effectively.

2.4 Assumpion, dependencies and constraints

2.4.1 Domain Assumption

The following assumptions are properties of the real world that the system takes for granted. They must hold true for the software to function correctly and provide meaningful data.

- **D1 -** Users possess a mobile device equipped with functioning GPS, accelerometer, and gyroscope sensors.
- **D2 -** During the Automatic Tracking Mode, the mobile device is securely mounted to the bicycle, to ensure that sensor data correctly reflects road obstacles.
- **D3 -** The GPS signal is available and has sufficient accuracy in the areas where the cyclist rides to allow for correct path reconstruction.
- **D4 -** Users act in good faith when uploading a path in manual mode and when validating automatically detected obstacles, confirming only real issues and discarding false positives.
- **D5 -** Users have access to an internet connection for map visualization, route searching, and data synchronization.
- **D6 -** The external services integrated into the system provide accurate, truthful, and up-to-date information.

2.4.2 Dependencies

The system relies on external services to provide specific functionalities that are outside the scope of BBP development.

- Map Service Provider: The system depends on third-party mapping services to render the interactive map.
- Weather Service API: The system depends on an external meteorological service provider to retrieve real-time weather data.

2.4.3 Constraints

- The system must comply with local data protection regulations (e.g., GDPR), particularly regarding the storage and sharing of location history, which is considered sensitive personal data.
- Since the application uses power-intensive components (GPS and sensors continuously) during rides, the software must be optimized to minimize battery drain, ensuring it can track long-distance trips without depleting the user's device.
- The route scoring algorithm is constrained by the "freshness" of the data; the system must implement a decay logic where older reports have significantly less weight than recent ones.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The application interface is organized into the following areas:

- **Public Section** (Accessible to Unregistered and Registered Users):
  - **Landing and Search Section**: The entry point of the application, featuring an interactive map and a search interface where users can specify origin and destination to find bike paths. This section displays routes filtered by their calculated score and status.
  - **Route Detail Section**: A view dedicated to displaying specific information about a selected route, including total distance, average time, path status and a list of reported obstacles.
  - **Login and Registration Section**: The interface allowing new users to sign up and existing users to authenticate to access the private area.
- **Private Section** (Accessible only to Registered Users):
  - **User Profile Section**: Displays user details and aggregate statistics, such as total km traveled.
  - **My Routes Section**: A personal space where the user can view their history of recorded trips, distinguishing between private and public routes.
  - **Manual Upload Section**: A form-based interface allowing users to manually insert route details, defining the start/end points, street names, and path status.
  - **Active Tracking Interface**: A dedicated screen for the "Automatic Mode". It displays real-time metrics (speed, duration) and provides controls to Start, Stop, and Pause the recording. It is designed for minimal interaction to ensure safety while riding.
  - **Post-Ride Validation Section**: A crucial interface presented at the end of an automatically tracked trip. It lists the obstacles and anomalies detected by the sensors, requiring the user to confirm or discard them before saving the route

3.1.2 Hardware Interfaces

The system does not require any dedicated hardware. Basic platform functionalities can be accessed from any device with an Internet connection and a compatible web browser or mobile application. Advanced features, such as automated tracking of routes, are available when using a mobile device equipped with Internet connectivity and built-in sensors, including GPS, gyroscope and accelerometer. No additional external devices or proprietary hardware interfaces are needed.

3.1.3 Software Interfaces

The platform relies on external software services to provide core functionality. In particular, it must integrate a map provider to render the map and display bike paths. In addition, it must acquire weather data from an external API to retrieve meteorological information associated with specific locations and routes, so that such data can be shown within the user interface.

3.1.4 Communication Interfaces

Client applications must have access to the Internet, either through a fixed network connection or a mobile data network. Communication between client and server for all data exchange is carried out over HTTPS, ensuring confidentiality and integrity of the transmitted information. Additional application-level communication (such as third-party map and weather services) also relies on secure HTTP-based APIs.

3.2 Functional Requirements

3.2.1 Requirements List

Requirements for **Goal 1**:

[**R1.1**] The system allows unregistered users to sign up to the platform by providing a valid email address, a password, and their personal details (name, surname).

[**R1.2**] When a user registers, the system verifies that the provided email address is not already associated with an existing account.

[**R1.3**] The system allows registered users to log in by providing their registration email and password.

[**R1.4**] The system allows registered users to modify their profile information (name, surname) and change their password.

To satisfy **G1**, these domain assumptions must hold:

[**D5**] Users have access to an internet connection for map visualization, route searching, and data synchronization.

Requirements for **Goal 2**:

[**R2.1**] The system allows registered users to manually create a new route by specifying the sequence of street names, the starting point, and the destination.

[**R2.2**] The system allows registered users to assign a specific status (e.g., Optimal, Good, Sufficient, Unsuitable) to a manually inserted route or segment.

[**R2.3**] The system allows registered users to add obstacles information to a manually created route.

[**R2.4**] The system allows registered users to save a new trip into their personal route history.

[**R2.5**] The system allows registered users to delete a previously saved route from their personal collection.

To satisfy **G2**, these domain assumptions must hold:

[**D4**] Users act in good faith when uploading a path in manual mode and when validating automatically detected obstacles, confirming only real issues and discarding false positives.

[**D5**] Users have access to an internet connection for map visualization, route searching, and data synchronization.

Requirements for **Goal 3**:

[**R3.1**] The system automatically queries the external Weather Service API upon saving of a route.

[**R3.2**] The system retrieves the following meteorological data based on the route's location and timestamp.

[**R3.3**] The system associates the retrieved weather data with the specific Route, making it visible in the route details.

[**R3.4**] If the external weather service is unavailable or unreachable, the system saves the route without meteorological data without interrupting the saving process.

[**R3.5**] The system does not allow users to manually modify the automatically retrieved weather data, ensuring the objective consistency of the environmental conditions recorded.

To satisfy **G3**, these domain assumptions must hold:

[**D6**] The external services integrated into the system provide accurate, truthful, and up-to-date information.

Requirements for **Goal 4**:

[**R4.1**] The system shall sample the user's position at regular intervals during the Automated Tracking Mode to reconstruct the path.

[**R4.2**] The system shall collect continuous data streams from the mobile device's accelerometer and gyroscope sensors while tracking is active.

[**R4.3**] The system shall analyze the sensor data to identify road obstacles.

[**R4.4**] When a potential obstacle is detected via sensor analysis, the system shall automatically save it for a user review, tagging it with the current GPS coordinates and timestamp.

[**R4.5**] The system shall execute data acquisition processes in the background, ensuring tracking continues even if the device screen is locked or the application is minimized.

[**R4.6**] The system shall calculate instantaneous metrics based on the distance between consecutive GPS points and the time elapsed.

To satisfy **G4**, these domain assumptions must hold:

[**D1**] Users possess a mobile device equipped with functioning GPS, accelerometer, and gyroscope sensors.

[**D2**] During the Automatic Tracking Mode, the mobile device is securely mounted to the bicycle (e.g., on the handlebars) to ensure sensor data reflects road conditions rather than body movements.

[**D3**] The GPS signal is available and has sufficient accuracy in the areas where the cyclist rides to allow for correct path reconstruction.

Requirements for **Goal 5**:

[**R5.1**] The system displays a "Post-Ride Review" summary to the registered user immediately after stopping the automatic tracking, listing all the obstacles detected during the trip.

[**R5.2**] The system allows the user to confirm a detected obstacle, permanently associating it with the recorded route.

[**R5.3**] The system allows the user to discard a detected obstacle as a "false positive", removing it from the final route data.

[**R5.4**] The system allows registered users to modify the type of a detected obstacle if the automatic classification is incorrect.

[**R5.5**] The system allows registered users to visualize the location of each detected obstacle on the map during the review phase to aid in the verification process.

To satisfy **G5**, these domain assumptions must hold:

[**D4**] Users act in good faith when uploading a path in manual mode and when validating automatically detected obstacles, confirming only real issues and discarding false positives.

[**D5**] Users have access to an internet connection for map visualization, route searching, and data synchronization.

Requirements for **Goal 6**:

[**R6.1**] The system allows registered users to change the visibility status of a saved route from "Private" to "Public".

[**R6.2**] The system allows registered users to change the visibility status of a previously published route from "Public" back to "Private", removing it from the global map and search results.

[**R6.3**] Upon the publication of a route, the system automatically triggers the Scoring Engine to incorporate path segments into the global dataset.

To satisfy **G6**, these domain assumptions must hold:

[**D5**] Users have access to an internet connection for map visualization, route searching, and data synchronization.

Requirements for **Goal 7**:

[**R7.1**] The system allows any user (registered or unregistered) to search for bike paths by specifying an origin and a destination via text input (addresses) or by selecting points directly on the map.

[**R7.2**] The system displays the available routes connecting the specified origin and destination on an interactive map.

[**R7.3**] The system visualizes the Path Score and Status of the retrieved routes.

[**R7.4**] The system allows users to select a specific route from the search results to view its detailed information.

[**R7.5**] The system orders the search results by default based on their Path Score, presenting the best-rated routes first.

To satisfy **G7**, these domain assumptions must hold:

[**D5**] Users have access to an internet connection for map visualization, route searching, and data synchronization.

[**D6**] The external services integrated into the system provide accurate, truthful, and up-to-date information.

Requirements for **Goal 8**:

[**R8.1**] The system shall automatically compute a numerical Path Score for every published route stored in the database.

[**R8.2**] The scoring algorithm shall calculate the score as a weighted function of the Path Status of the composing segments and the presence of obstacles.

[**R8.3**] The system shall automatically trigger a recalculation of the Path Score for a route whenever the status of its underlying segments is modified or updated, ensuring the score always reflects latest informations.

[**R8.4**] The system shall utilize the computed Path Score to rank the results of a route search, displaying the highest-scoring routes at the top of the list.

To satisfy **G8**, these domain assumptions must hold:

[**D6**] The external services integrated into the system provide accurate, truthful, and up-to-date information.

Requirements for **Goal 9**:

[**R9.1**] The system shall identify geographically overlapping path segments from different published routes by analyzing their GPS coordinates to determine which user submissions refer to the same physical road section.

[**R9.2**] The Scoring Engine shall calculate the consolidated Path Status for a specific road section by applying a majority consensus algorithm to the overlapping segments submitted by different users.

[**R9.3** The merging algorithm shall assign a weight to each user submission based on its Data Freshness. Recent reports must have a significantly higher influence on the than older reports.

[**R9.4**] The system shall retain the history of merged segments to allow for the re-evaluation of the status if new data arrives.

To satisfy **G9**, these domain assumptions must hold:

[**D4**] Users act in good faith when uploading a path in manual mode and when validating automatically detected obstacles, confirming only real issues and discarding false positives

4 Formal Analysis using Alloy

5 Effort Spent