Design 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 trips and track personal performance, as well as to share these experiences with others. Having access to updated data about bike paths—such as their conditions, safety, and suitability—can 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 paths. The system promotes collaboration among users, encouraging the community to contribute and maintain reliable data on the status of bike paths.

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 information—including 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 paths between a chosen origin and destination.

1.3 Definitions

- **Automatic Tracking Mode**: A functional mode of the mobile application where the system continuously collects data from the device's sensors (GPS, accelerometer, gyroscope) to reconstruct the cyclist's path and detect road irregularities without requiring manual input during the ride.
- **Data Freshness**: A parameter used by the Merging Engine representing the age

of a submitted report. The system prioritizes more recent data (high freshness) over older data when calculating the consolidated status of a road segment.

- **Draft**: A temporary state of a recorded path (specifically from Automatic Creation) that has been saved locally or on the server but has not yet been validated or finalized by the user. Drafts are not visible in the personal history until confirmed.
- **Merging Engine**: The logical component of the system responsible for consolidating overlapping path segments from different user submissions into a single, consistent global view. It applies the majority consensus algorithm and data freshness logic.
- **Obstacle**: A verified physical impediment or hazard located along a bike path that affects the ride quality or safety.
- **Path Score**: A numerical value computed by the system for each path. Provide a single indicator of the path's "goodness" for ranking search results.
- **Path Segment**: The atomic unit of a path. A path is composed of an ordered sequence of segments.
- **Path Status**: A categorical value assigned to a segment or path indicating its condition. The valid values are: Optimal, Good, Sufficient, Poor, Unsuitable.
- **Path/Trip**: A defined path consisting of a sequence of segments connecting an origin to a destination. A path can be created manually or derived from a recorded trip and can be stored as Private (visible only to the creator) or Public (visible to the community).
- **Post-Ride Validation** : The process where a Registered Cyclist reviews the data collected automatically to confirm real issues and remove false positives before the data is permanently saved.
- **Segment Snapshot**: A specific instance of a road section recorded by a user at a specific point in time. The system aggregates multiple snapshots of the same location to derive the global status.

1.4 Acronyms

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

- **API**: Application Programming Interface
- **GDPR**: General Data Protection Regulation
- **GPS**: Global Positioning System
- **UI**: User Interface

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

- **Version 1.0**: xx/yy/26

1.7 Reference Documents

- Specification Document: Assignment RDD AY 2025-2026
- Course slides

1.8 Document Structure

- **Introduction**: Outlines the purpose, scope, definitions, acronyms, and reference documents for the *Best Bike Paths* project.
- **Architectural Design**: Details the system's high-level architecture.
- **User Interface Design**: Visualizes the application's look and feel via mockups and navigation schemes, linking them to RASD requirements.

- **Requirements Traceability**: Maps functional and non-functional requirements to specific design modules to ensure full coverage.
- **Implementation, Integration and Test Plan**: Defines the development sequence, integration strategy, and testing procedures.
- **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.
- **References**: Software used to develop the document.

2 Architectural Design

2.1 Overview

The *Best Bike Paths* (BBP) system adopts a *Three-Tier Client-Server* architecture. This choice ensures separation of concerns, scalability, and independent maintainability of the user interface, business logic, and data storage.

The architecture consists of the following layers:

1. **Presentation Layer (Client Tier)**: Handles user interaction and data visualization. Consistent with the requirements specified in the RASD, the system supports multiple client types, with limitations for the WebApp Client
2. **Application Logic Layer (Server Tier)**: Hosts the core business logic of the system.
3. **Data Layer (Persistence Tier)**: Manages the persistent storage of data. It ensures data integrity and supports concurrent access from multiple application server instances.

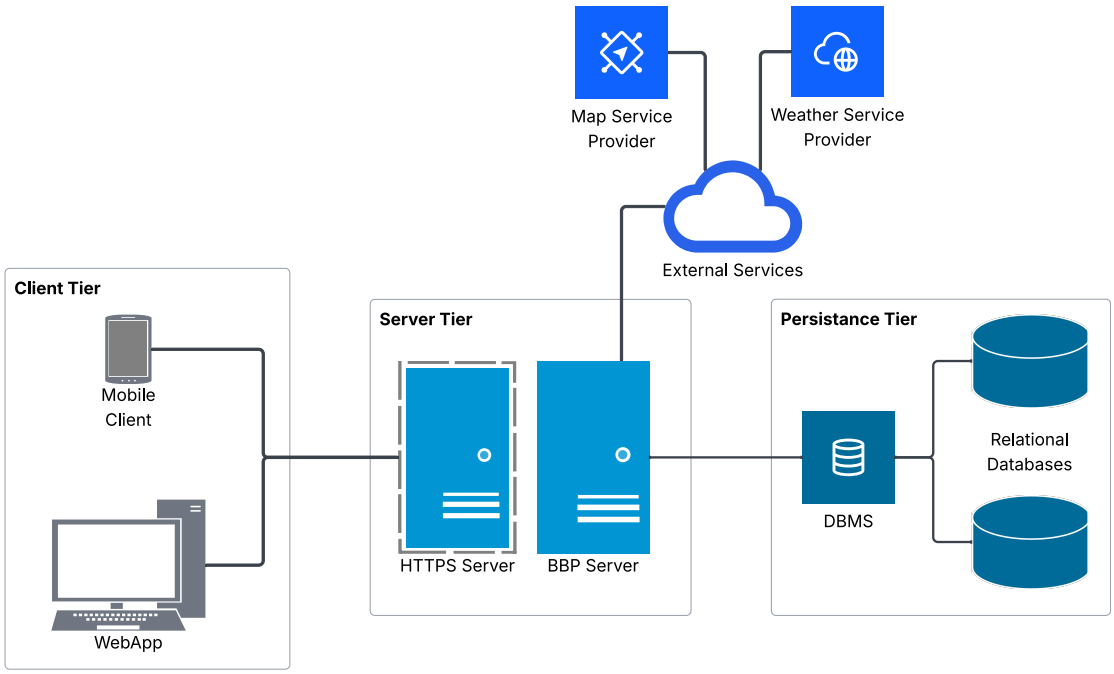

Figure 1: High Level Architecture Diagram

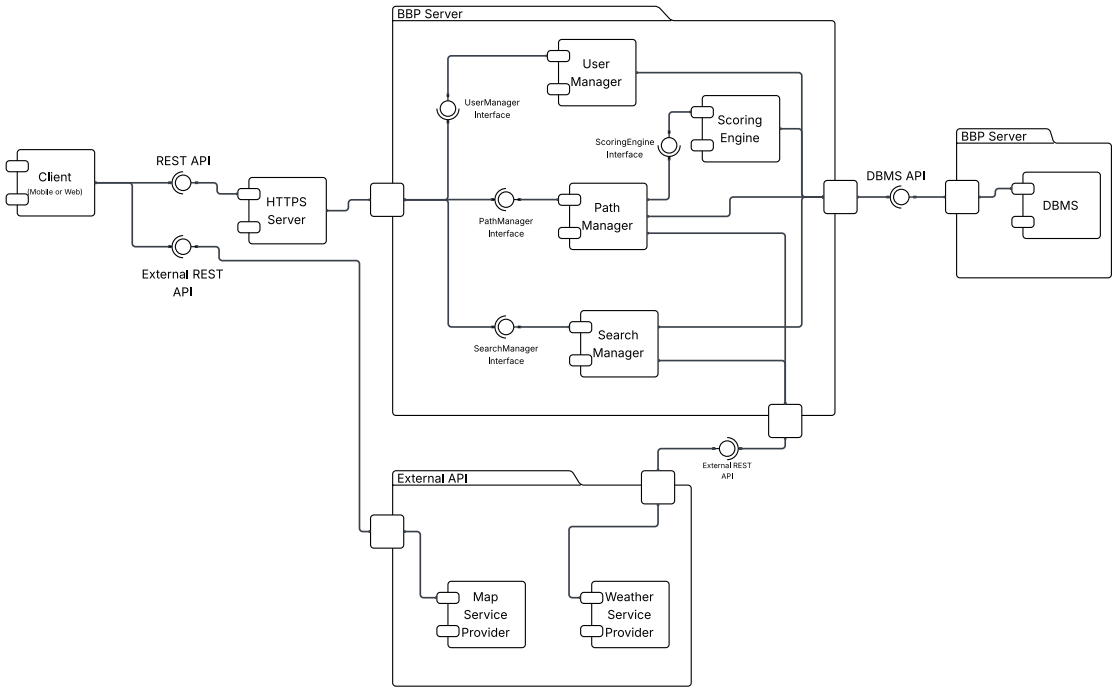

Figure 2: System Components Diagram

2.2 Component View

User Manager: This component handles the lifecycle of user accounts, including registration, authentication, and the modification of profile information.

Score Engine: It is responsible for calculating the "Path Score" by aggregating user feedback to rank routes based on quality.

Path Manager: This component manages the creation and validation of trips, handling both manually inserted paths and those recorded via automatic tracking, publishing, and details retrieving.

Search Manager: It processes user queries to identify and retrieve available bike paths connecting a specific origin and destination, ensuring results are sorted by their score.

HTTPS Server: It acts as the secure entry point for the system, managing encrypted communication between client applications and the backend to ensure the confidentiality and integrity of data in transit.

DBMS: This component is responsible for the persistent storage and retrieval of all system data.

Weather Service: An external interface used to retrieve real-time meteorological data to enrich trip records based on location and time.

Map Service: An external provider responsible for rendering interactive maps, resolving addresses, and supporting the visualization of paths and obstacles.

2.2.1 User Manager Component

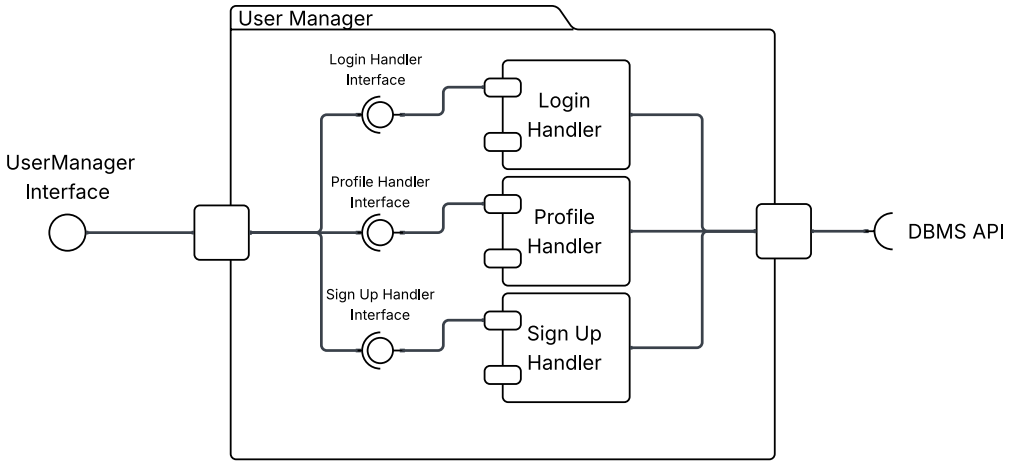

Figure 3: User Manager Component Diagram

Sign Up Handler: This component manages the secure login process for registered cyclists. It is responsible for validating user credentials. It also enforces security constraints by handling password hashing and salting mechanisms to ensure sensitive data is never stored in plain text.

Sign Up Handler: This component handles the sign-up logic for new users. It validates input data (name, surname, email, password) and performs a check to ensure the provided email address is not already associated with an existing account before creating a new profile.

Profile Handler: This component enables registered users to view and modify their personal account information. Processes updates to profile details (such as name and surname) and validates the new input data before persisting changes to the database.

2.2.2 Path Manager Component

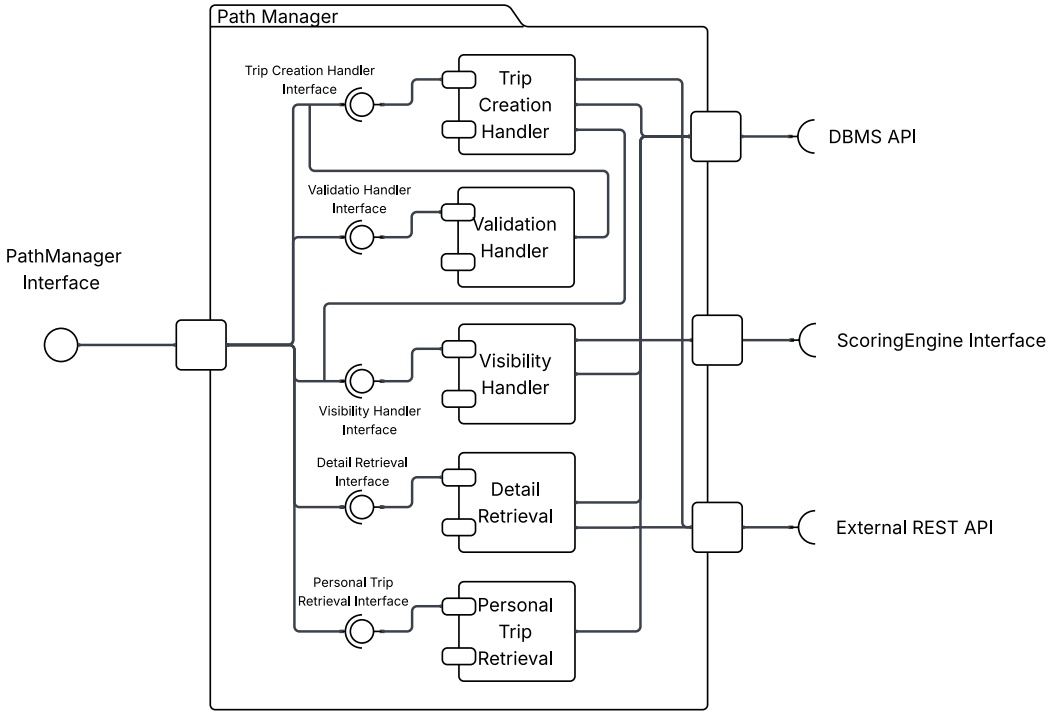

Figure 4: Path Manager Component Diagram

Trip Creation Handler: This component manages the initial recording of a trip. It handles Manual Trip Creation by processing the sequence of streets and status entered by the user. It coordinates with the Weather Service Interface to enrich the trip with meteorological data upon saving.

Validation Handler: This component is responsible for the Post-Ride Validation workflow. It serves the list of automatically detected obstacles and streets to the user for review. Processes the user's confirmation or rejection before the trip is permanently finalized by saving it with the Trip Creation Handler Component

Visibility Handler: This component handles the state changes of a trip. When a trip becomes Public, or changes from public to private, the component triggers the Scoring Engine Component to incorporate the new data into the global dataset.

Detail Retrieval: This component handles the retrieval of comprehensive data for a single specific trip. It serves data for both private trips (for the creator) and public trips (for any user).

Personal Trip Retrieval: This component is responsible for fetching the collection of trips created by a specific registered user. It queries the database to retrieve the personal history list, providing summary details and distinguishing between 'Private' and 'Public' visibility states.

2.2.3 Search Manager

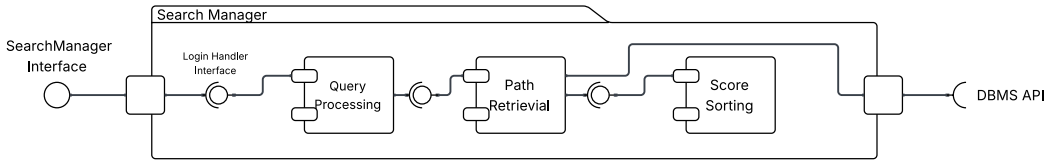

Figure 5: Search Manager Component Diagram

Query Processing: This component validates the user's search inputs, ensuring that the specified origin and destination (provided via address text or map selection) are valid.

Path Retrieval: This component is responsible for querying the database to fetch all available bike paths that physically connect the origin and destination points. It retrieves the list of matching routes along with their associated data.

Score Sorting: This component processes the list of retrieved paths and orders them based on their "Path Score".

3 User Interface Design

4 Requirement Traceability

5 Implementation, Integration and Test Plan

6 Effort Spent