POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Software Engineering II

Implementation Document

Project: Best Bike Paths

Authors: Ianosel Bianca Roberta, Gholami Vajihe, Errigo Simone
Version: 1.0
Date: 23.12.2025
Project Link: Errigo-Gholami-Ianosel (github.com)

Contents

1 | Introduction

1.0.1. Purpose

This document aims to describe how the implementation and integration testing took place. Implementation is the last step of the Best Bike Path application development cycle.

Testing, instead, means check that the critical parts of the application works in a correct way, as described in the DD document.

The code and the releases can be find on the official CLup repository hosted on GitHub, reachable at this link:
https://github.com/BIA3IA/Errigo-Gholami-Ianosel.

1.0.2. Definitions, Acronyms, Abbreviations

Acronyms

- **API:** Application Programming Interface.
- **APK:** Android Package
- **DBMS:** DataBase Management System.
- **DD:** Design Document.
- **DOM:** Document Object Model.
- **DTO:** Data Transfer Object, represents a link between the user input and a Java Object.
- **HTTP:** HyperText Transfer Protocol.
- **IPA:** iOS App Store Package.
- **JPA:** Java Persistence API.
- **JS:** JavaScript.

- **QR Code:** Quick Response Code.
- **REST:** REpresentational State Transfer (see DD).
- **RASD:** Requirements Analysis and Specification Document.
- **S2B:** Software To Be.
- **UI:** User Interface.
- **URL:** Uniform Resource Locator.
- **UX:** User eXperience.
- **ORM:** Object-Relational Mapping.

Abbreviations

- **something:**

1.0.3. Revision history

- January 28, 2026: version 1.0 (first release)
- February 7, 2026: version 1.1 (fixed some typos)

1.0.4. References

- **Dart** https://dart.dev/
- **Easy Loading** https://pub.dev/packages/easy_loading
- **Flutter Barcode Scanner** https://pub.dev/packages/flutter_barcode_scanner
- **Flutter Local Notification** https://pub.dev/packages/flutter_local_notifications
- **Flutter Printing** https://pub.dev/packages/printing
- **Flutter** https://flutter.dev/
- **Font Awesome** https://fontawesome.com/
- **Google Fonts** https://fonts.google.com/
- **Google Maps Flutter** https://pub.dev/packages/google_maps
- **Hive** https://pub.dev/packages/hive
- **Java:** https://www.java.com/it

- **JavaScript:** https://www.javascript.com
- **jQuery:** https://jquery.com
- **Kotlin** https://kotlinlang.org/
- **Leaflet.js** https://leafletjs.com
- **Pull to refresh** https://pub.dev/packages/pull_to_refresh
- **QR Flutter** https://pub.dev/packages/qr_flutter
- **QRcode.js:** https://jquery.com
- **Semantic UI:** https://semantic-ui.com
- **Spring Framework:** https://spring.io
- **Spring Security:** https://spring.io/projects/spring-security
- **Spring Social:** https://projects.spring.io/spring-social
- **Swift** https://www.apple.com/it/swift/
- **Url Launcher** https://pub.dev/packages/url_launcher
- **Vue.js:** https://vuejs.org

2 | Development

2.0.1. Implemented Functionalities

With respect to the RASD and DD documents, we decided to implement the following functions:

- **something**

For more details regarding the specific functionalities, you are invited to read the RASD document, which contains a very detailed description of them.

We chooses to implement these functionalities in order to simulate (something).

2.0.2. Adopted Development Frameworks

As we said in our DD, the application should follow a 3-tier architecture

In the following pages you will find a list of adopted frameworks and technologies in order to accomplish to this requirements.

Programming languages

Of course, there are some pros and cons about using this type of language:

- **Pros:**
  + something
- **Cons:**
  - something

2.0.3. Java Frameworks

Spring

2.0.4. Other Frameworks

sth

Package it.polimi.se2.somethinh

- **Config:**
- **Controllers:**
- **Entities:**
- **Model:**
- **Repositories:**
- **Security:**
- **Services:**

Test cases

2.0.5. Frontend Structure

Web Application

2.0.6. Mobile App

2.0.7. Other Useful Information

3 | Testing

3.1. Testing

In this section we will briefly describe how we tested the application, following the general guidelines given in the Design Document. We decided to test only the backend and mobile app deployable, because the front-end can be easily tested "by seeing it working".

3.1.1. Backend

Integration Tests

- **UserController:** we tested the Sign Up and login methods, as defined in the DD. In the sign up tests, we tried with both a Customer and a Store Manager profiles. Finally, we tested the correct exchange of the Json Web Token between the parties.

The above test, which were in total (number), were run all together, obtaining a percentage of success of 100%.

Thanks to the obtained result it is therefore possible to state that the backend is sturdy and built on solid source code.

Unit Tests

We wrote unit tests for the main services of the backend, in order to verify that each method was working as expected.

3.1.2. Mobile Application

4 | Build

4.1. Build

Requirements

To build the project, the following requirements must be met:

5 | Installation

5.1. Installation

In this section we will describe how to install and run both the backend and mobile application, using the provided self-contained installer or by compiling the source code.

Requirements

5.1.1. Backend

Windows self contained installer

Cross-platform JAR Package

5.1.2. Mobile Application

Store Builds

Download the application based on your platform following the provided links:

Github Builds (Android only)

Download the application by accessing the Release page on our Github repo or by clicking the link **here**.

6 | Effort Spent

This section provides a breakdown of the number of hours each group member dedicated to completing this document. The work distribution is tracked per section and task.

Section	Ianosel Bianca	Simone Errigo	Vajihe Gholami	Total Hours
Introduction	4 hours	4 hours	5 hours	13 hours
Overall Description	11 hours	7 hours	10 hours	28 hours
Specific Requirements	19 hours	8 hours	12 hours	39 hours
Formal Analysis	7 hours	21 hours	11 hours	39 hours
Final Review & Editing	3 hours	3 hours	3 hours	9 hours
Total Hours	**44 hours**	**43 hours**	**41 hours**	**128 hours**

Table 6.1: Time spent on document preparation

Bibliography

List of Figures

List of Tables