Cycling Advocacy
Project Plan

Version 1.1

Revision History

Date	Version	Description	Author
2019-10-21	0.01	Initial Draft	Everyone in Team
2019-10-25	1.0	Final version	Everyone in Team
2019-11-10	1.1	Revision 1	Dominik Kotarski

1. Introduction

1.1 Purpose of this document

The purpose of this document is to provide a high-level overview of the Cycling Advocacy project. This document provides general information about the project, such as its background, scope, objectives and impact, as well as information on the project team and its organisation.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction,* describes the contents of this guide, used documentation during developing process etc.
- Section 2, *Background and objectives*, describes the customer, as well as the purpose and objectives of the project
- Section 3, *Project impact*, describes the people interacting with the system, as well as the impact the system will have on their life and work
- Section 4, *Project group*, identifies the project team members, their roles, responsibilities and means of contact
- Section 5, *Development process*, describes the development process with regards to Scrum elements that are used on the project
- Section 6, *Organization and Communication*, describes the project organization such as the meeting schedule, communication tools and working hours of team members
- Section 7, *Initial time plan*, describes the overall timeline of the project, including deadlines and planned Scrum Sprints
- Section 8, *Quality assurance*, describes planned activities and protocols designed to assure product quality
- Section 9, *Project risks*, lists possible risks that may be encountered during development, their descriptions as well as means of avoidance

1.3 Intended Audience

The intended audience is:

- The customer
- The project team
- Project supervisors
- All other project stakeholders

1.4 Scope

The scope of this document includes general project information such as its background, scope, goals, timeline, and impact. Furthermore, this document contains project team information which includes team members, their roles, responsibilities, and means of contact. Project organization details such as schedules, deadlines, communication means, etc. are also included. Finally, this document details possible project risks and the means to avoid them.

This document does not include product implementation details such as architecture or low-level implementation details.

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
Scrum	An agile process framework for managing complex knowledge work
Customer	The project's client
Product Owner	A Scrum role designating the person responsible for communication with the Customer and translating the Customer's requirements into a prioritized list of features to be implemented
Scrum Master	A Scrum role designating the person responsible for ensuring the project team's success and compliance with Scrum principles
Stakeholder	A Scrum role designating any person with a vested interest in the project, that is not part of the project team
Scrum Sprint	A product development cycle which typically lasts from 1 to 4 weeks with a strictly defined duration and deadline
Linux	A family of operating systems
Android	A mobile operating system based on a modified version of the Linux kernel

1.5.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
NTR	Nothing to Report. There is no information on a specific topic available or necessary.
POLIMI	Politecnico di Milano (Polytechnic University of Milan)
FER	Fakultet elektrotehnike i računarstva (Faculty of Electrical Engineering and Computing)
GPS	Global Positioning System
B2W	Bike2Work is an EU co-funded project which promotes a significant energy-efficient modal shift from motorized transport to cycling.

1.6 References

- Scrum: https://www.scrum.org/
- Cyclist's Union: http://sindikatbiciklista.hr/en/
- Fakultet elektrotehnike i računarstva: https://www.fer.unizg.hr/en
- Politecnico di Milano: https://www.polimi.it/en/
- Bike2Work: http://www.bike2work-project.eu/en/

2. Background and objectives

This section contains information about the customer, describes what is the purpose of the project from the customer's perspective and lists project requirements.

2.1 Customer

The customer for this project is the Cyclists' Union. They are a volunteer association founded in 2011 and located in Zagreb. Their main goal is to promote cycling as an efficient, sustainable and healthy way of transport in order to have a clean, safe and pleasant green cities. Using different workshops, education and projects they are constantly improving cyclist conditions and overall quality of life in cities as well.
Currently, Cyclists' Union has more than 2500 members that are actively working on achieving their goals.

2.2 Project purpose

To ensure a comfortable and safe ride, it is important to monitor road conditions. That is why Cyclists' Union wants to monitor and analyze cycling paths to detect rough roads and locate road bumps. Since most people nowadays use smartphones, the idea is to use smartphones, precisely Android phones with different sensors to send detected data of road conditions. Collected data will be used to display problems on particular parts of roads and therefore warn people so they can switch their routes and ride safely, but also to notify the local administration to intervene. This information may encourage more people to use bicycles in their everyday activities which will gradually lead to green cities that they are trying to achieve.

2.3 Requirements

Project requirements are listed below:

- The system should include Android application for collecting sensor data of holes and road bumps
- The system should run services on Linux machine
- The system should visualize data on a map, showing problematic road parts
- The system should integrate with existing systems to report road issues
- The system should store collected sensor data in the relationship database

Optionally, the system could be integrated with the Bike2Work mobilization tool. Bike2Work is an EU co-funded project lead by the European Cyclists' Federation whose main objective is to encourage employees to shift from motorized commuting to cycling when traveling to work. This project is already present in many European countries including Italy and Croatia.

Another possibility is integrating the system with the FixMyStreet tool It is a road issue reporting system that includes various data like GPS location of issue, photos, and issue type. Integration with it would include using existing backend from their official website, where the application itself should handle user privacy, battery usage, etc. Purpose of this integration would be to provide users with a more direct way to help improve their cycling infrastructure. This would include taking photos of road issues and submitting them via our app and FixMyStreet integration.

3. Project impact

People in Zagreb are facing similar issues to those in Italy, Sweden and around the world: mobilizing citizens to cycle more and drive cars less, evaluating the quality of cycling surfaces, communicating infrastructure issues to local government, collecting cyclist behavior data necessary for various types of decision making and others. Nowadays people could use bicycles when going to work, shopping, to visit friends or simply as a form of training. When going in unknown parts of town it is useful to know which routes are more cyclist-friendly. Dangerous road surface conditions with bumpy roads, potholes and rough roads are the major distraction for safe and comfortable cycling. Therefore, application that would detect and visualize problematic roads would be very useful to anyone interested in cycling. It would not only make their ride safer but would also motivate other people to start cycling more.

4. Project group

The project group consists of members from the Polytechnic University of Milan (Politecnico di Milano) as well as the Faculty of Electrical Engineering and Computing (Fakultet elektrotehnike i računarstva) from the University of Zagreb.

The members of the team are as follows:

- **Sandra Kuzmić**
  - FER
  - kuzmic.sandra1@gmail.com
- **Carlo Casiglia**
  - POLIMI
  - carlo.casiglia@gmail.com
- **Saloni Kyal**
  - POLIMI
  - saloni.kyal@mail.polimi.it
- **Dominik Kotarski**
  - FER
  - dominik.kotarski@fer.hr
- **Elena Bakuleva**
  - POLIMI
  - elena.bakuleva@mail.polimi.it
- **Izabella Szydelko**
  - FER
  - izabella.szydelko@gmail.com
- **Federico Ferri**
  - POLIMI
  - federico3.ferri@mail.polimi.it
- **Boris Vezmar**
  - FER
  - boris.vezmar@fer.hr

5. Development process

The project is going to be developed using Scrum methodology. Besides being a useful tool for managing project development in general, it became particularly suitable when team members are in different locations and works at different times, because it reduces the process to small components, making it easier to keep track of the progress and distributing responsibilities. An example of this is the fact that each team member is given tasks with a clearly defined scope, start, end, priority etc.
It is not our intention to dive into the various rules and paradigms of Scrum methodology in this document (for which we refer to the various books and documents online); we are just pointing out a small overview of how we decided to implement some core concepts of Scrum in our process.

5.1 Sprints

The whole development is going to be divided into time units called sprints. In traditional Scrum sprints can be up to a month-long; in our case, though, since we have little time to develop the application (and little experience) we've decided to **begin with a couple of very small one-week sprints**, which are not enough to develop anything, but in the beginning will help us to quickly adjust to our mistakes and fine-tune our organization. In fact, sprints are defined in advance and cannot be changed. Then we will **increase sprint duration up to two weeks** unless we'll experience different necessities.

5.2 Tasks

Tasks are going to be defined in advance to each sprint, during a Scrum event known as Sprint Meeting. This means that we are going to try to decide with the greatest reliability we can what we are going to deliver by the end of the sprint, and what activities we are going through in order to do so. The task should be very small so that every task can be totally clear and straightforward to complete, and so that it can be assigned to a single person, who can autonomously update its status and keep track of changes.

5.3 User stories

User stories are a nice feature of Scrum that aim to express users' true necessities through narrative sentences, in order to go through the whole development process with a constant focus on the requirements. At each sprint begins we are going to select the user stories we are going to cover with that sprint with our tasks, so to be sure we are always working for a precise goal.

6. Organization and Communication

6.1 Roles

- *SCRUM Master*: Carlo Casiglia
  As Scrum Master, Carlo's responsibilities are helping the SCRUM Team be successful and ensuring that they act according to SCRUM practices.
- *Product Owner & FER Team Leader*: Sandra Kuzmic
  As Product Owner, Sandra's responsibilities include communicating with the Customer, identifying product features and translating them into a prioritized list.
- *Polimi Team Leader*: Saloni Kyal. *Team leaders are expected to coordinate and organise their local teams respectively.*
- *Developers*: Boris Vezmar, Dominik Kotarski, Izabella Szydelko, Elena Bakuleva, Federico Ferri

6.2 Customer

- Tomislav Nakić-Alfirević
  Contact info: nakic@gmx.com

6.3 Supervisors

- Ivana Bosnić
  FER Supervisor
  Contact info: ivana.bosnic@fer.hr

- Elisabetta Di Nitto
  Polimi Supervisor
  Contact info: elisabetta.dinitto@polimi.it

6.4 Tools

1. SLACK: It is a platform used for sharing all kinds of information about project planning, daily work, and progress.

2. GOOGLE DRIVE (and other related tools): This is used for building all shared presentations and documentation in general. It contains all the important files about the project accessible to all team-mates easily.

3. SKYPE / MEET: This is used for all video conferences and calls.

4. TRELLO: It helps the team to stay organized.

5. GITHUB: It is a repository that helps us share the code.

6.5 Meetings

Meetings are scheduled according to the mutual availability of team members and supervisors (for meetings with supervisor) which is decided by creating a *Doodle poll* and the members vote in the time slot they are available. We also maintain *Google Calendar* where all the group members share their schedules in order to arrange further meetings. The product owner is in contact touch with the customer via email and updates the team with all the information. One of the team members maintain meeting notes for all the meeting that takes place to avoid misunderstanding and confusions.

There are **three** types of meetings:

1. *Internal group meetings:* this meeting is between all the group members which necessarily happens once a week to update and discuss all the project works through videoconference.

2. *Group meetings with Supervisors:* once a week, the respective local team meet with the supervisor and get together with the team from other location through videoconferencing. This meeting is to report team progress and to receive feedback from the supervisors.

3. *Meeting with the customer:* this meeting depends upon the customers. If the customer would like to meet the product owner or maybe the entire team then this meeting takes place otherwise the product owner is in contact with the customer via email.

6.7 Working Hours

Every time each member works on some tasks, he/she posts a comment under the task assigned to him/her on the Excel Sheet for now (we will shift to Trello later for managing assigned task and no. of hours worked), explaining the work developed, the possible issues encountered and how many hours he/she spent on that task. In this way, it is possible to track the actual working hours taken on every single task.

Moreover, the team is encouraged to constantly update the "Sprint Review Report", especially the "Daily working hours" table. In this way, it is possible to exactly estimate the working hours for each member.

7. Initial time plan

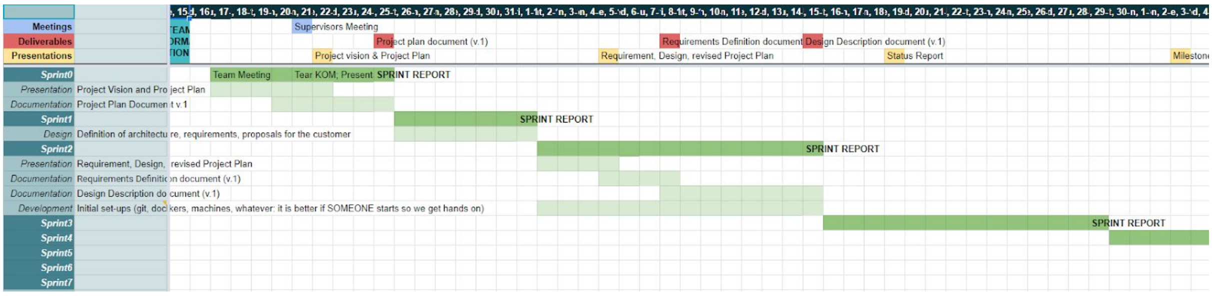
7.1 Deliverables

DOCUMENT / PRESENTATION	DATE
Project Plan and Vision	25-10-2019
Requirements Definition	08-11-2019
Design Description	15-11-2019
Alpha Prototype	03-12-2019
Beta Prototype	17-12-2019
Final Product	14-01-2020

7.2 Planned Sprints

SPRINT	START DATE	END DATE
0	15-10-2019	25-10-2019
1	26-10-2019	01-11-2019

7.3 Gantt Charts



8. Quality assurance

8.1 Branching

The branching method adopted is the master-develop-feature model. It consists of two main branches (master and develop) and various features branches created for the features currently being worked on.

The commits flow from one branch to another via pull requests. To add some local commits to a feature branch a push is enough and this allows different team members to work on the same feature. A pull request is required to add commits from the feature branch to the develop branch. Another pull request is required to add commits from develop to master. Other interactions are not possible using branch rules.

8.2 Code review

Code reviews will be conducted on every pull request on the master branch by at least one team member other than the person submitting the request.

8.2. Testing

Developers will be responsible for testing their own modules. Testing is key to a correct development and we try to keep the development as testing driven as possible. Also, developers will keep the module size as small as possible. This way, the time spent on creating tests is minimized.
Integration and global system testing will be carried out on all project components on every project release. This happens before the new version is made available to the client.
Automatic testing is also planned and the codebase will be equipped with an automatic testing system that will execute all unit tests on every commit to warn developers in case of bugs.

8.2. Regular releases

The client will regularly receive project releases and access to the codebase. This will enable quick feedback on project functionality and quality.

9. Project risks

9.1. Miscommunication with the customer

Poor initial communication with the customer could lead to an incorrect identification and definition of project requirements and features, which can then send the project into a completely incorrect direction. Infrequent communication with the customer could prevent the early discovery of issues with the project. Frequent communication with the customers also allows them to continuously provide valuable input during development helps mold the results of the project into a product that best fits their needs and requirements.

In order to avoid this issue, frequent communication with the customer must be established via organized meetings where the Product Owner discusses the development status and features of the project with the customer.

9.2. Inexperience with the Scrum framework

A lack of experience with the Scrum framework, or failure to abide by its principles could lead to poor organization within the project team and the subsequent failure of the project. Failure to follow Scrum principles could lead to poor communication within the team, poor communication with the customer, failure to meet deadlines, failure to meet customer requirements, poor documentation and a bad resulting product.

In order to avoid this issue, all team members must have a basic knowledge of Scrum as a whole, as well as good knowledge of Scrum elements that are being used for the project.

9.3. Poor communication within the team

Poor communication within the team could result in inefficiency during development, a lack of understanding of features that are to be developed, poor cooperation and even arguments within the team.

In order to avoid this issue a carefully planned meeting schedule should be decided in which the team members meet often and discuss the status and progress of their given tasks, as well as future development plans.

9.4. Failure to meet deadlines

Failure to meet deadlines could lead to organizational problems, frequent schedule changes and the inability to complete the project in time. Additionally, rushed development in order to meet deadlines should also be carefully considered and avoided if possible since the resulting product will most likely be bad.

A concrete risk for this project is the inability to complete the Bike2Work optional module within the deadline. With that in consideration, it is yet to be determined if the module in question will be implemented.

In order to avoid this issue, development should be separated into small tasks with strict deadlines. The progress and status of tasks developed at a certain time should often be checked and evaluated, with resources being allocated in order to best ensure that the timely completion of those tasks.

9.5. Inexperience with technologies and tools

Inexperience with technologies and tools used during development could lead to inefficiency, stagnations, delays and missed deadlines.

In order to avoid this issue development tasks need to carefully be assigned to team members with consideration of whether a member can successfully complete the given task. Team members also need to be introspective and open with the rest of the team, acknowledging and sharing their capability, or lack thereof, of completing certain tasks.

9.6. Lack of quality assurance

A lack of quality assuring activities, routines or protocols could lead to invalid or broken features, and subsequently a bad end product. Additionally, integration of feature implementations that contain bugs, faults or are not in compliance with customer requirements not only adds little to the end product but can also cause issues further down the line if the need for revisions or fixes arises.

In order to avoid this issue, quality assurance routines must be established so that all implemented features are thoroughly reviewed before integration. All new features must pass through verification and validation, in which consideration is given towards whether those features comply with customer requirements and whether they are implemented well.