RIDERTRACK

Project Vision and Plan

Version: 2.0.0

16th December 2017

Index

Revision History

Date	***Version***	***Description***	***Responsible***
24/10/2017	0.1	Initial draft	Alessandro Caprarelli
03/11/2017	1.0	First versions	Alessandro Caprarelli
16/12/2017	2.0	- Dropped additional features such as messaging and private data. Plan changed according to it. - Updated tools used and git flow. - Minor fixes.	Alessandro Caprarelli

1. Introduction

1.1 Purpose of the Document

The purpose of the document is to convey to the reader an high level idea of what the project is about, by treating its main peculiarities and purposes. To achieve this goal, the dissertation will begin from the background, the context and the environment in which the project takes place and will conclude with the quality assurance and the project risk, passing through the description of the internal organization of the team, and the information about the time plan arranged for the implementation of the project to be.

1.2 Document Organization

The "Project Vision and Plan" document is essentially structured in six parts:

1. In this ***first section***, it is provided an Introduction that describes the content of the document.
2. In the ***second section***, it is provided a detailed description of the project background and the main problems affecting it, then it is presented the solution and an high level overview of the chosen system.
3. In the ***third section***, it is explained the organization of the project, team roles, meetings and the communication tools used.
4. In the ***fourth section***, it is defined the initial project schedule with the planned sprints and deadlines.
5. In the ***fifth section***, it is explained how the team provides quality assurance to the project.
6. Finally in the ***sixth section***, it is provided a list of all the potential risk that can be involved in the project.

1.3 Intended Audience

The target audience of this document includes:

- **Users:** this category holds three different kinds of users, united by the common potential interest in the project to be and distinguished by habits, interests and, especially, by the way **"Rider Track"** would improve their life experience. Below, the distinction:
    1. *Athletes:* runners, bikers, riders, marathon or trail runners, whoever loves and practises competitive sports, which include an itinerary to follow.
    2. *Spectators and supporters:* whoever is interested in following a competition.

3. *Event organizers:* whoever wants to organize and share an event, supplying the most user-friendly and performing experience to their athletes and spectators.

- **Project customer:** to check if the work planned by the team is coherent with his requirements and to agree upon the major objectives.
- **Developers:** to allow the developers to understand the project and to work individually with a unique shared idea of the goal to be reached.

1.4 Scope

The aim of the ***"Rider Track"*** project is to provide an user-friendly management application for outdoor sport events and, especially, to support multiple forms of tracking sources, even ones not introduced yet. Also, It aims to increase supporter involvement and participants' safety and to improve logistic thanks to real time data.

1.5 Definition and Acronyms

1.5.1 Definitions

Keyword	*Definition*
REST API	API that adheres to the REST architectural constraints
Event Organizer/Administrator	A person who organizes and manages events.
Spectator	A person who follows the progress of an event.
Participant	A person who participates in an event
Sprint	Basic time unit for development used in SCRUM
Project customer	Customer who requested the project
Sport event	An event such as a marathon, cycling race, hiking race ecc. In general a sport event with a route, starting point and an ending point.

1.5.2 Acronyms and Abbreviations

Acronym / Abbreviation	*Definition*
PBI	Product backlog item

MOM	Minutes of meeting
GPS	Global positioning system
FER	Fakultet elektrotehnike i računarstva (hrv.)
POLIMI	Politecnico di Milano
SPOT	Satellite personal tracker
GPS	Global Positioning System

2. Background and Objectives

2.1 Background

Nowadays, there is a specific issue affecting the world of outdoor events. In fact, only a few services are thought to follow the tracking of an event with a fine-grained and real-time data collection.
The biggest problem for event organizers, who want to use such services of tracking, is the cost of the event, which we can quickly estimate an approximation: according to Forbes, more than 50000 runners joined the 46th NYC Marathon in 2016. In order to provide a real-time tracking service to all the participants, the organizer should have accessorized each of them with a GPS device. One of the most common and quite affordable devices is the SPOT GEN 3, which cost varies around $120. Multiplying the unitary cost by the 50000 participant, we can easily realize the value of more than 6 million that have to be invested by the event organizer for such an extended outdoor event.
Another issue, that an event organizers may have to face, is the integration of multiple tracking devices and services, on the same platform. Once the administrator selects a certain company for the race, the event in question will be equipped only with devices and services of that company: this fact, both from the event administrators' and participants' point of view could be restricting.

2.2 Project Vision

Starting from the considerations made at the previous section, the "Rider Track" project aims to overcome the problem explained, through an extensible system which integrates multiple tracking data sources in a unique platform, exploiting the already own users' devices. In particular, the system is based on a web application, addressed to spectators, event organizers and participants. The web application is intended to provide tools for the management of several different kind of competitions, starting from the moment of the event creation and then above all its lifetime, and real-time tracking of all participants.
Moreover, the project includes a native mobile application specifically thought for the participants' needs, which allows them to automatically send detailed data about their progress and position, thanks to the data collection performed by components and sensors embedded in their mobile phone.

2.3 Impact

The "RiderTrack" system is intended for people gathered around an athletic event such as participants (runners, hikers, cyclists...), organizers and spectators.

From the *spectators'* point of view, the system will improve the competition experience by guaranteeing the most accurate tracking possible, which will allow the spectators outside the race to precisely evaluate their progress.
From the *event organizers'* point of view, the system will improve the event management, by simplifying the organization of an event.
From the *participants*' point of view, it will improve and simplify the tracking experience.

2.4 High Level Requirements

In this section the most relevant requirements are described.
Some requirements were dropped. A complete and detailed rationale about the decision can be found in the "Major changes to project plan" chapter.

2.4.1 Details about event

The system must display events details, such as the name, date, category, place, and associated route of the event.

2.4.2 Search for a participant

The system must provide the functionality to select one participant and track only his/her progress during the race.
~~Every user has as option to search for participants of the event. The system must demand to insert an authorization code to the spectators who want to access to the private area of a participant during an event (both private data and communication channel). When it's granted they are able to see personal data of the participant they have searched and also message them. If an authorization code is not granted a guest can only see his/her name, last name and position.~~

2.4.3 Organize an event

An user must be able to create an event. The system must allow the user to create an event after the insertion of proper data, such as name of the event, date of the event, category of the event, associated route, etc.
An user, that has created an event, must be able to modify the information and delete the event.
An user, that has created an event, must be able to open/close the registration of the participants.
An user, that has created an event, must be able to start and stop the tracking of the event.

2.4.4 Track the participant on map

The system must provide a page from which anyone can see the real time progress of an event, and especially the position of all the participants on a map.

2.4.5 Share tracking data

Participant can share his/her tracking data using the mobile app. Firstly, a participant has to login into the app using his/her credentials and then share his/her live position with the system which is then processed.

2.4.6 Participant messaging

~~The Participant can share a private code generated by the system with a certain event spectator. If done so, spectator can access private information about him/her and message him/her during the race. This way, spectator can motivate and inform the participant about his/her and others progress.~~

2.4.7 Registering into the web app

The system must provide the possibility to register in order to create and enroll in events. The registration and login must be offered also using social services such as Facebook and Google.

3. Organization

3.1 Team Members

Alessandro Caprarelli

Alessandro is 24 years old. He's Italian, precisely from a small village in the center of Italy. He currently lives in Milan, where he moved one year ago, with the purpose of attending his Master's Degree program in Computer Science at Politecnico di Milano.

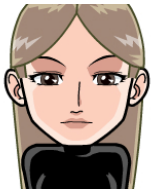
Contact info: caprarelli.alessandro@gmail.com

Mariano Etchart

Mariano studies Electronic and Electrical Engineering at Loughborough University, in the United Kingdom.
His passport is from Argentina, even if he never lived there. In fact, he was born in Brisbane, Australia. From there, he moved to South Africa, where he stayed for some years before leaving for UK. Currently, he is doing an exchange at Polimi.

Contact info: mariano.etchart@gmail.com

Marzia Degiorgi



Marzia is a 24 years old Italian girl, who lives in Caravaggio, a small village near Bergamo. She currently studies Computer Science and Engineering at Politecnico di Milano, where she previously graduated in Telecommunication Engineering.

Contact info: marzia.degiorgi@mail.polimi.it

Giulia Leonardi

Giulia is 23 years old and comes from Senigallia, a small seaside town in the center of Italy. She has lived in Milan for 4 years, where she moved in order to study Telecommunication Engineering at Politecnico di Milano. She achieved her Bachelor Degree in 2016, and she is currently a Master's student of Computer Science and Engineering.

Contact info: giulia.leonardi@mail.polimi.it

Ante Breščić

Ante is a 23 years old student from Croatia. He is currently attending a Master's Degree program in Computer Engineering at FER, in Zagreb, where he obtained his Bachelor Degree in 2016.

Contact info: ante.brescic@fer.hr

Ivan Kvesić

Ivan is 23 years old. He has been studied in Zagreb, Croatia, since the elementary school. He completed there also his high school and he is currently enrolled in the Master's Degree in Computer Engineering at FER, where he got his Bachelor Degree in 2016.

Contact info: ivan.kvesic@fer.hr

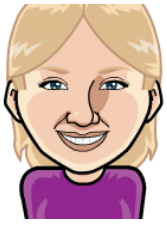
Josip Mališa

Josip comes from Pazin, Croatia, and he is 24 years old. He is studying at FER, in Zagreb, and he has a Bachelor Degree in Computer Engineering.

Contact info: josip.malisa@fer.hr

3.2 Supervisors

Ivana Bosnić



Remote supervisor

Contact info: ivana.bosnic@fer.hr

Elisabetta Di Nitto

Local supervisor

Contact info: elisabetta.dinitto@polimi.it

3.3 Customer

Michal Young

SCORE project proponent

Contact info: michal@cs.uoregon.edu

3.4 Team Roles

Scrum roles:

- **Product Owner**: Alessandro Caprarelli
- **Scrum master**: Mariano Etchart
- **Developers**: Marzia Degiorgi, Giulia Leonardi, Ante Breščić, Ivan Kvesić, Josip Mališa

Technical roles:

- **Frontend developers**: Ante, Giulia, Marzia.
- **Backend developers**: Ivan, Josip, Mariano.
- **Fullstack developer**: Alessandro

3.5 Development Process

The first 4 sprints are one week long, in order to get used to the principles of SCRUM process. From then on, the sprints will be 2 week long.
Once a week, the team holds a meeting with supervisors, where the progress are shown and explained to them.
At the end of each sprint, the team has a “sprint review/retrospective” meeting during which the members discuss what has been done, what is still to do, which issues has raised and what has gone well. Particularly, the team detects what should be changed regarding the work organization and what should be done in a different way.
At the beginning of each sprint, the team performs a “sprint planning” meeting, during which it decides which backlog tasks should be done in the sprint to come and it plans roles and required time for each task.
Every morning, each team member writes down a report (“daily standup”) of what has been done the previous day and what is in his/her plan to do that day. The “daily standup” is managed in Slack. If no report is posted, the rest of the team assume that a certain team member has not planned to work in that day.

3.6 Organization and Communication

3.6.1 Working hours

Every time each member works on some tasks, he/she posts a comment under the task assigned to him/her on Asana, 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 each 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.

3.6.2 Meetings

The team fixed 3 different weekly meetings:

- *Product backlog grooming*: usually on Wednesday, may change according to availability. Product owner and scrum master are required to participate. All the other team members are optional but very welcome.
  Estimated duration: 30 minutes, maximum 1 hour.
- *Sprint review, retrospective and planning*: on Friday from 10 to 12. All the team members should participate.
- *Supervisors' call*: on Friday from 12 to 13. All team members should participate.

The product owner and scrum master will also try to keep constant the communication with the customer.
Other meetings will be scheduled according to the availability of the members and according to needs. Participation and collaboration is always very welcome.
All the meetings will be reported in a MOM document, in which what has been discussed, the conclusion and which actions will follow up are summarized.

3.7 Tools

Planning, sharing and coordinating work in a distributed team would be much harder without the aid of additional managing tools.
Skype and *Talky are used for calls.*
Slack provides to them the possibility of sharing common text messages and of matching daily works and progress.
Asana guarantees the SCRUM process effectiveness and helped the team in coordinating work through different sprints.
Google Drive (and all the related tools) is used for sharing files, editing diagrams, graphs, presentations and the whole documentation in general.
Github is used as a repository in order to share code.

Travis is used in order to automatically build, execute code and deploy new versions.
Visual paradigm is used to make diagrams, such as sequence, state, class diagrams.

4. Project Plan

4.1 Planned Sprints

Sprint	*Start date*	*End date*
0	13.10.2017.	19.10.2017.
1	20.10.2017.	26.10.2017.
2	27.10.2017.	2.11.2017.
3	3.11.2017.	9.11.2017.
4	10.11.2017	23.11.2017.
5	24.11.2017.	7.12.2017.
6	8.12.2017.	21.12.2017.
7	22.12.2017.	4.1.2018.

4.2 Deadlines

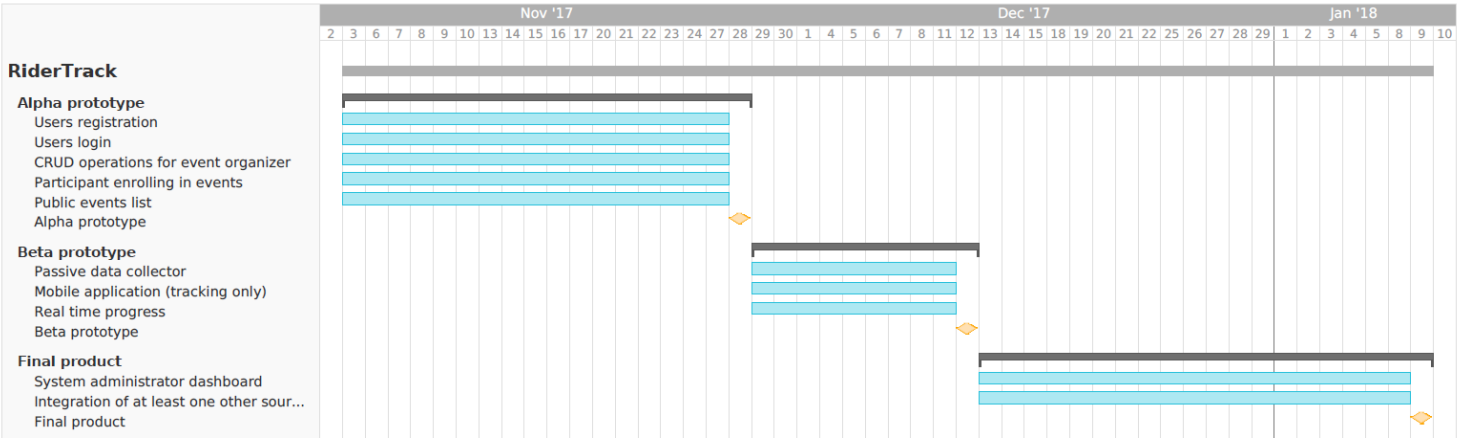
These are the deadlines which the team have to meet for presentations and deliverables both for the exam and the SCORE competition.

Item	*Date*
Alpha Prototype	28-11-2017
Beta Prototype	12-12-2017
Final presentation	9-01-2018

4.3 Initial Time Plan: Gantt Chart

The team will face 3 milestone. In the following Gantt chart you may see the features we are planning to provide for each deadline.
The plan was updated due to the dropping of some requirements. For more detail refer to the “Major changes in project plan” chapter of this document.

1. The first version, the **Alpha prototype**, will allow only the management of the events.
    - the organizer will be able to create, modify and update the status of the events. He/she will be able to publish a ranking at the end of an event.
    - the participant will be able to enroll in events.
    - the guest will be able to see the events, without the tracking.
2. The second version, the **Beta prototype**, will add to the Alpha prototype the real time tracking of an event.
    - the event organizer will be able to select the mobile application as a tracking source.
    - the participant will be able to use the mobile application to be tracked during an event.
    - the guest will be able to see the real time progress of an event.
3. The third version, the **Final Product**, will add to the Beta prototype the possibility to integrate other tracking sources (we plan at least one as an example) ~~and the communication channel between spectators and participants~~.
    - the organizer will be able to select among different tracking sources.
    - the participant will be able to be tracked with others devices and not only our mobile application.
    - ~~the participant will be able to set as private some data and share them only with a restricted set of spectators. He/she will also be able to receive messages during the event.~~
    - ~~the spectators will be able to send messages to participants and to unlock the private data using an authorization code.~~



Note: For a higher resolution version, see the attached Gantt_v2.0.pdf

4.3 Major changes in project plan

In this section the major changes of the project plan are explained in detail.

4.3.1 Dropped messaging and private data features - 8/12/2017

The team decided to not implement the following features, planned at the beginning:

- Possibility to exchange data between spectators and participants.
- Possibility to make private some data about the performance.
- Possibility to send additional data during an event.

The reason behind this is because of time constraints we are unable to deliver everything and we would rather deliver the most important components. The product still delivers all the required components by the customer so the system will not be affected drastically. To begin with, these features were ambitious in nature and over-exceeded the expectation from the customer. The only aspect that will perhaps suffer is the marketability, the system will still be perfectly functional and meet the requirements of the customer.
Refer to MOM013 for more detail about the meeting during which the decision took place.

5. Quality Assurance

In order to keep high the quality of the ***"Rider Track"*** project and to assure a final consistent product, the following actions will be taken:

- Single unit of code such as modules, classes and components will be tested using unit tests.
- Each task has to pass a "Code review and test" stage before being marked as complete. This task will be done by a tester that should not be the same that developed it. Depending on the size of the task and on the components involved more than one person may be needed during the code review and test stage.
- Integration tests will be done in order to understand if new features added are ready to be merged with the stable version without altering its integrity.
- Both unit and integrations tests will be automated. Travis will run all the tests.
- Until all tests are not successfully passed the task cannot be marked as done.
- The Github repository has 2 branches:
    - Master branch in which there is the stable version.
        - Nobody is allowed to directly push into this branch.
        - The code is pushed only through a merge from the development branch.
        - Code is merged into this branch only for the planned releases. There will be 3 merges, one for each milestone.
    - Development branch in which there is the version under development.
        - All the developers push changes into the development branch.
        - At every push, merge issues may arise and the developer tries to immediately resolve them. In this way this kind of issues are not propagated but immediately discovered and solved.
- In the cloud environment will reside two versions of the system: production and development.
    - Changes pushed to the development branch are automatically deployed in the development version, if the tests pass.
    - The development version is manually promoted as the production one when it has the planned features implemented and tested.

6. Project Risks

The risks that the team may encounter during the project:

Description	***Mitigation***	***Probability of occurrence***
Technical		
Difficulties with new technologies	Sharing of knowledge among the team	Medium
Difficulties in integrating some external sources because they don't offer clear or free public APIs.	Focusing on most accessible ones and establish a contact when problems occur	Medium
Quality		
Difficulties in testing the complete environment in a real scenario due to the high amount of people usually participating in a competition.	Simulate high amount of requests	High
Performance		
Difficulties in optimizing the performance of the mobile app and in keeping low the battery consumption.	Testing out different sampling rate of location data.	Medium/High
Management		
Difficulties in following the project schedule and deadlines.	Construct and implement a Gantt Chart to plan ahead of time.	Medium
Difficulties in having resources such as SPOT GEN 3 or Garmin connect.	Plan and arrange for resource or facility use.	High
Difficulties in finding simultaneously free time for all the team members and lack of communication between the team.	Properly use Asana and Slack	Medium

Organizational		
Difficulties with team dynamics and conflict arising from differences in ideas and opinions.	Regular meetings for clear communication. Mindfulness of cultural differences.	Medium
Wrong role and responsibility allocation	Cross functional sub- teams	Low
External		
Difficulties in attaining tracking data from external API's such as Under Armour Connected Fitness.	Early contact with API providers.	Low