

PedalaMi Final Report

Version 1.0

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

Revision History

Date	Version	Description	Author
2022-01-14	1.0	First version	PedalaMi Team

Introduction	4
Purpose of this document	4
Document organization	4
Intended Audience	4
Scope	4
Project background	4
Project goals	5
Project impact	5
References	6
Project results	6
Overview of the results	6
Deliverables	6
Requirements	13
Functional Requirements	13
Non-Functional Requirements	15
Missing requirements and possibilities for improvements and extensions	15
Guidelines for software installation and operation	15
Installing Flutter (only for developers)	16
Adding a SHA certificate fingerprint for Firebase (only for developers)	16
Installing the app via APK (only Android)	17
Installing NodeJS and running the server locally (only for developers)	17
Structure of the source code	18
Project work	19
Work organization and routines	19
Team members' roles and responsibilities	19
Total project effort	20
Sprints analysis	23
Positive experiences	26
Possible improvements	27

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

1. Introduction

1.1 Purpose of this document

The purpose of this document is to provide a reader with the overall information about the achieved results of the project, how the project work was organized and completed, and what was the team members impressions throughout this process.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes the contents of this document and its scope, including the references used during the development process.
- Section 2, *Project background*, provides a contextualized overview of this project.
- Section 3, *Project Results*, gives an overview of the project deliverables and states the fulfilled requirements.
- Section 4, *Project Work*, describes in detail the development process, with the help of many graphs based on appropriate metrics

1.3 Intended Audience

The intended audience is:

- **Team members**, the entire team should have full access to this document, to have a final report of the development process.
- **Supervisors**, the professors from POLIMI and MDH should have full access to this document, to evaluate and grade the entire process.
- **Customer**, Deloitte Digital, to whom deliver the final product, in order to provide them with a final report of the project results.

1.4 Scope

This document summarizes the whole development process and the achieved results.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

2. Project background

The aim of the PedalaMi project is a simple and user-friendly design as well as to offer the best possible user experience while cycling in a city. This project promotes the use of bikes as a green and sustainable alternative to non-zero-emission vehicles, which still are the most preferred choice in the field. This is achieved by providing users with an interactive system that facilitates cycling, thanks to a fun reward and challenge-based system to enrich the standard riding experience, making it more attractive to everyday cyclists.

The customers of the project are Guido Perrucchini and Giuseppe Valerio Gramazio, who are employed by Deloitte Digital as developer consultants.

2.1 Project goals

The goal set by the customers for this project is to create a product that encourages the use of bicycles by ensuring a possibility for a better and healthier lifestyle for the application users, and, at the same time, reduce urban traffic and even pollution.

The main idea of PedalaMi is based on nudge and gamification techniques. For example, a user, riding a bike, can obtain badges and collect points to later redeem rewards. Users can also join existing teams or create their own teams to participate and compete in private or public team events.

2.2 Project impact

Nowadays, almost all cities suffer from traffic congestion, which also leads to high levels of air pollution, often above the maximum acceptable threshold. As this project encourages its users to ride bicycles instead of using cars, PedalaMi has a positive effect on the reduction of traffic jams, positively impacting not only its user's lifestyles but also the entire society. Guiding people towards a more carbon-free attitude, PedalaMi also helps in the reduction of air pollution, allowing its users to have an active part in the effort to contribute to the planet's sustainability.

2.3 References

- For more detailed information about the PedalaMi project planning see the **Project vision and plan document**.
- To read more about the requirements gathering process of the project see the **Requirements Definition document**.
- Detailed information about the design process of the PedalaMi software is described in the **Design Description document**.
- More in-depth analysis of the sprints are provided in the sprint reports.
- A description of the testing process is provided in the dedicated report.
- The GitHub repo is available at the following link: <https://github.com/pedalami/pedalami>
- “Scrum Handbook” by Jeff Sutherland.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

3. Project results

3.1 Overview of the results

The overview of the results of PedalaMi project is a hybrid platform application for both Android and iOS, along with a web dashboard to provide the user an easy and convenient access, as well as Heroku for the backend side deployment.

PedalaMi app provides the following main features:

- Allows the user to create a profile and log in to their account, to edit username and profile picture.
- Allows the user to record a ride and get points from them, and to see previous rides and statistics.
- Displays the air quality at the start of the ride.
- Assigns extra points for riding in adverse weather.
- Allows users to create and join teams.
- Allows users to join individual and team events, to earn more points in case of victory.
- Assigns badges to users based on their statistics
- Allows the user to spend points to redeem rewards.

Furthermore, the web dashboard allows the user to log in, to monitor their statistics and to see previous rides.

3.2 Deliverables

The deliverables were produced throughout the course of this project, along with various revisions:

- **Documentation:**
 - Project Vision & Plan document
 - Requirements Definition document
 - Design Description document
 - Acceptance Test Planning document
 - Acceptance Test Report document
 - Final Project Report document
- **Developed Products:**
 - PedalaMi hybrid application (iOS & Android)
 - Web dashboard
 - PedalaMi server

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

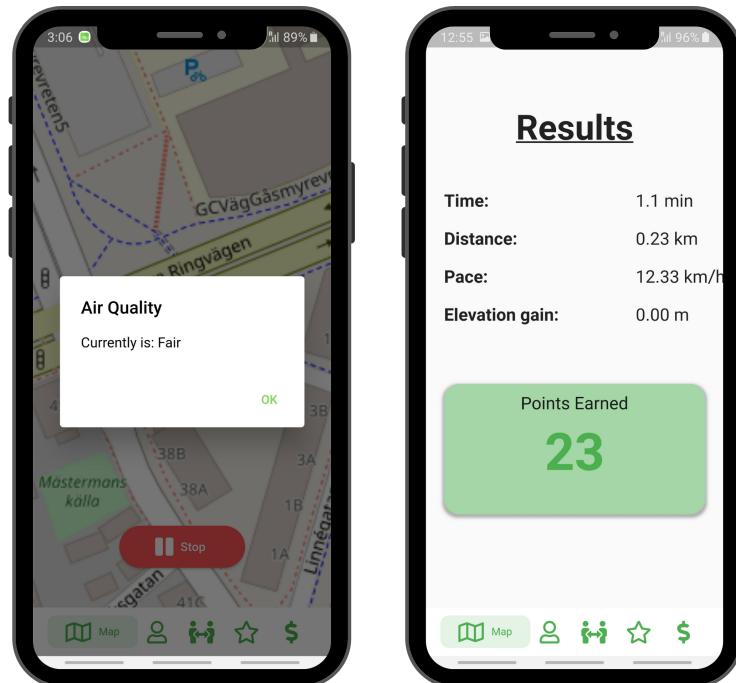
➤ Mobile application screenshots shown below.

After the user has logged in, he is prompted to a map after allowing the use of GPS on the device.

After the user position is retrieved he can start a ride tapping the *Start* button.



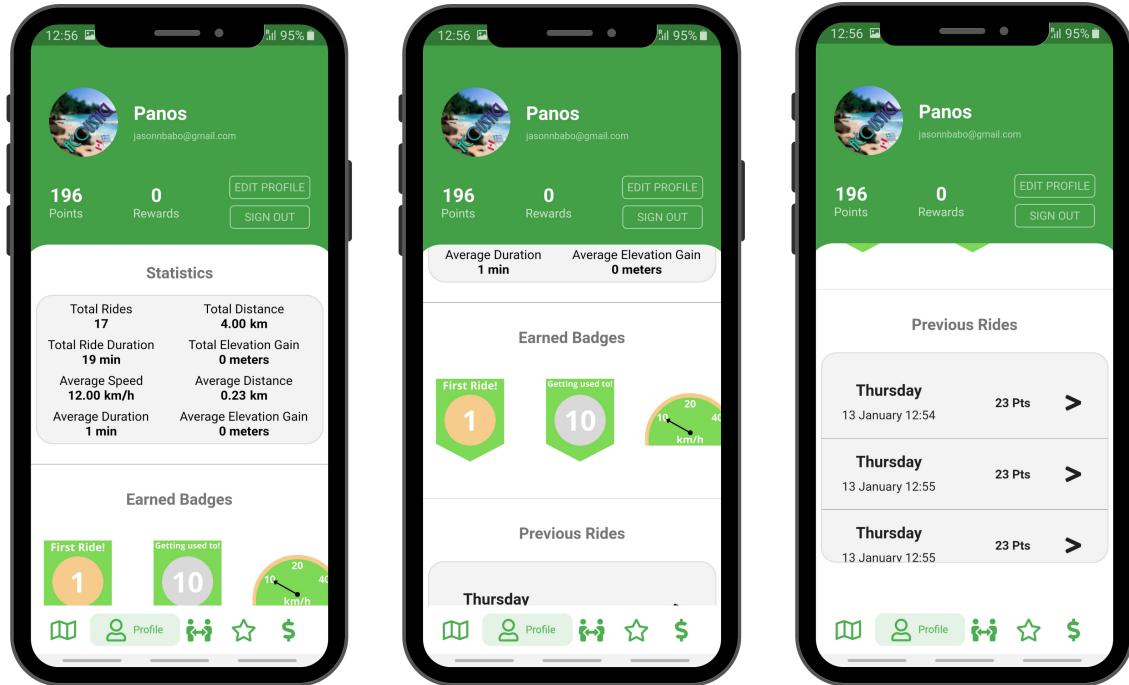
Then, a message pops up, informing the user about current air quality. After riding, the ride stops once the user taps on “Stop”. Then ride statistics appear on screen.



PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

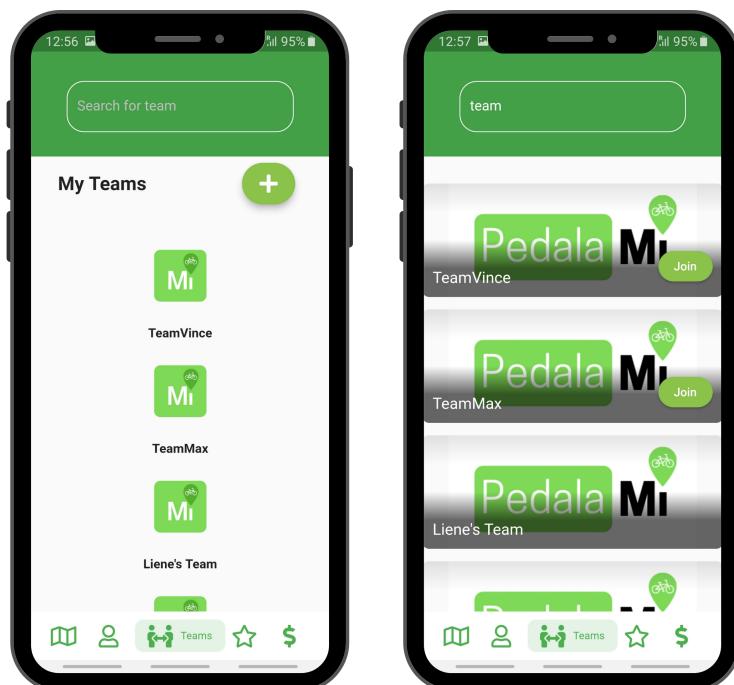
When tapping on the profile page icon, the profile page is displayed, where user statistics are shown.

When scrolling down, the badge section and ride history is displayed.



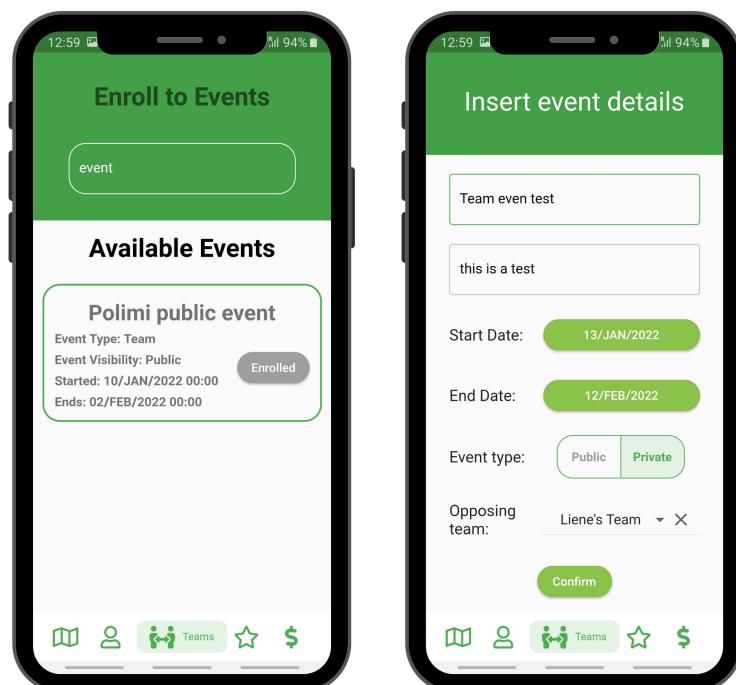
When tapping on the teams icon, a list of joined teams is shown along with the teams search bar.

Thus, the user can search for a team and join it if he wants.



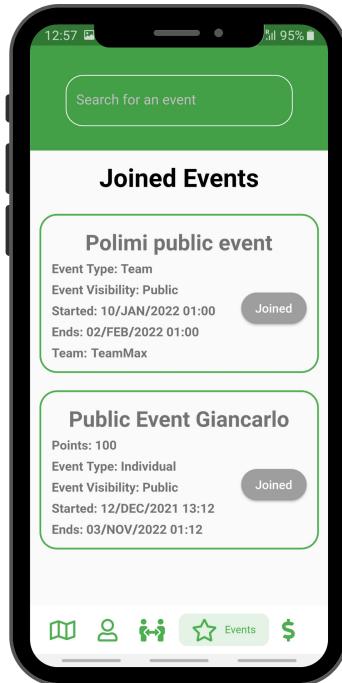
PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

When clicking on one of the teams created by himself, as team admin the user has several options: he can see team members, create events, enroll to team events and accept or deny invitations.

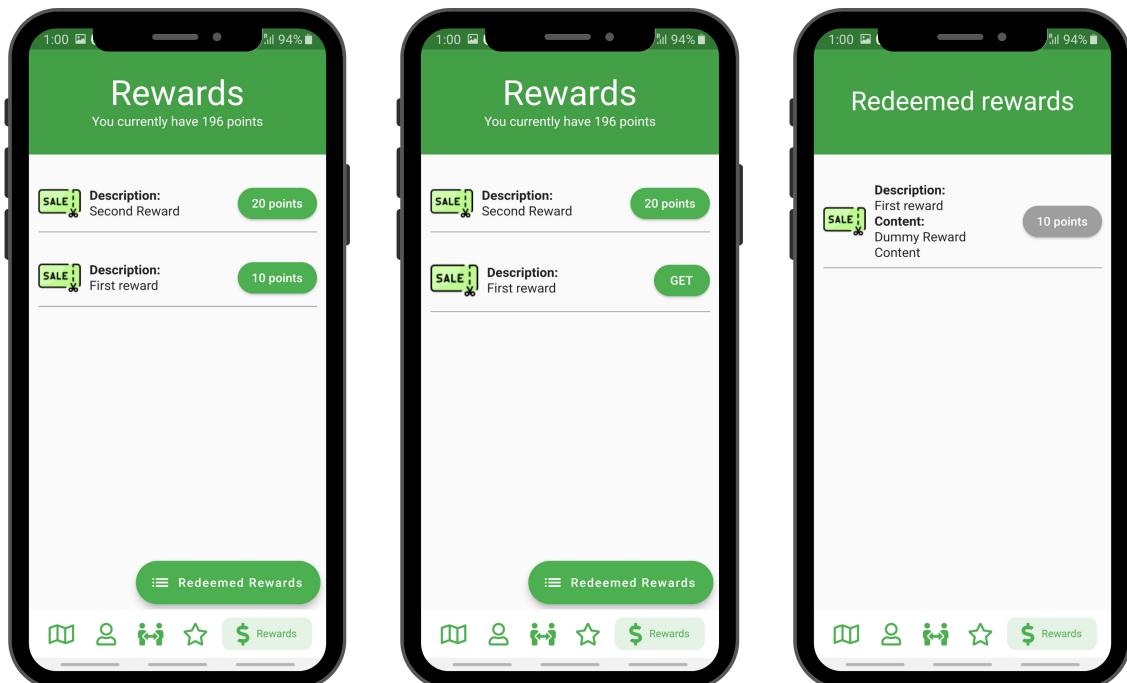


PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

When tapping on the events icon, a search page of public events appears and a list if joined any.

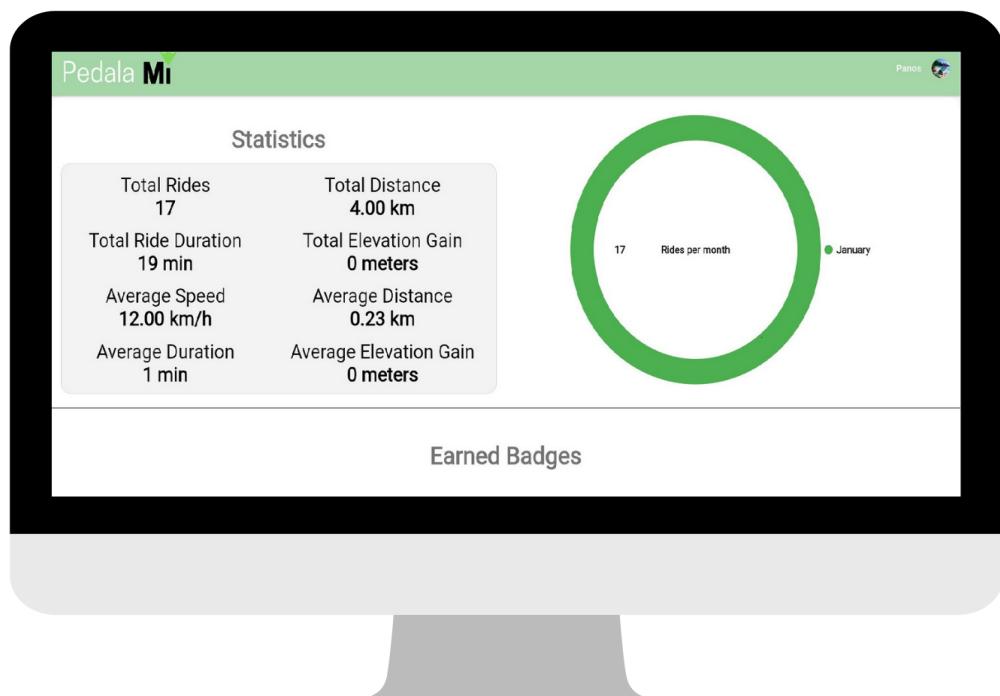
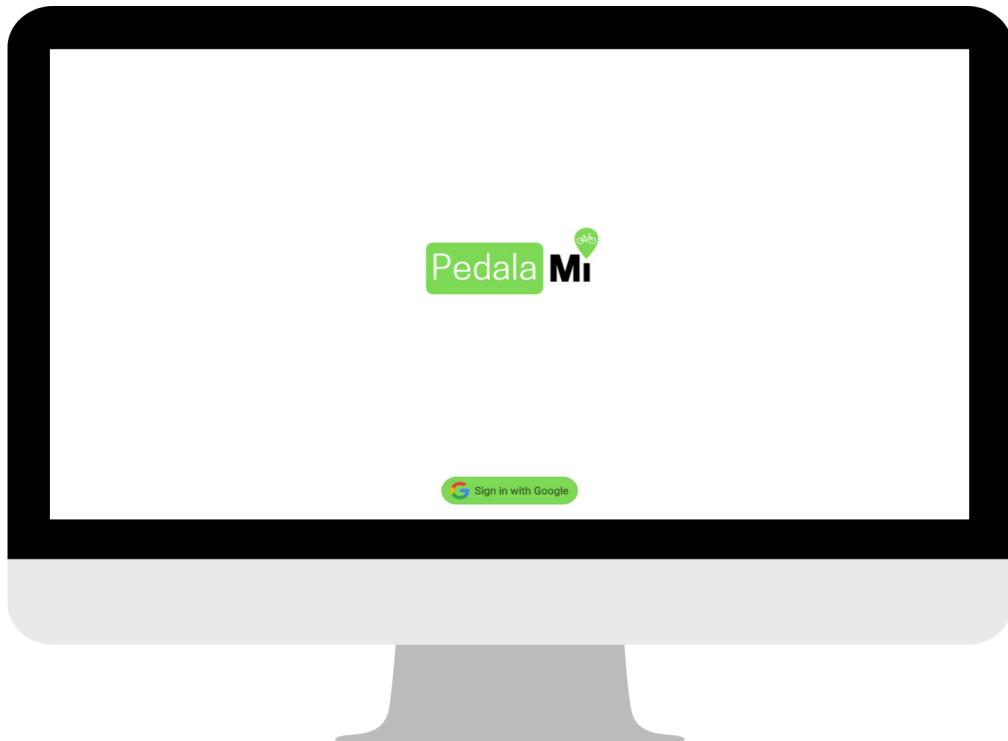


Finally, when the user clicks on the dollar sign for rewards, a list of available rewards is shown to be claimed if the user has enough points to spend.



PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

➤ Web dashboard screenshots shown below:



PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

3.3 Requirements

The requirements gathering process was performed by analyzing the initial project proposal and obtaining some high-level requirements in it. The project team discussed them further with the customer to specify their priorities.

3.3.1 Functional Requirements

The specified functional requirements are listed in the table below.

ID	Description	Source	Motivation	Satisfied?
R1	The application must have authentication, allowing users to login	Project group	Users should be authenticated to track and share their progress	YES
R2	The application must allow users to record bike rides	Customer	It is the main functionality of the application	YES
R2.1	The app should allow users to see the rides they made	Customer	This is to let the user look at previous rides	YES
R3	The application must collect users' rides statistics and display them	Customer	Doing so, users can track their progress	YES
R4	The application must allow users to edit their username and their profile picture	Project group	Users want to keep their personal information up to date	YES
R5	The application must allow users to earn points from their rides	Customer	This in order to encourage the users to use the bicycle	YES
R6	The application must assign badges to the users as a reward, based on their rides' statistics	Customer	This in order to make the riding experience more funny and challenging	YES
R7	The app should allow users to redeem rewards	Customer	This in order to reward the users for their efforts in cycling	YES
R8	The app should display information about the redeemed rewards	Project group	This in order to allow users to keep track of their rewards	YES
R9	The application should allow	Customer	This in order to team up with	YES

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

	users to create teams and become team admins		other users and propose team events	
R10	The application should allow users to search for teams and join them	Project group	This in order to let users compete and earn more points	YES
R11	The application should allow team admins to create private team events	Project group	This in order compete with another team and earn more points	YES
R12	The application should allow system admins to create individual public events	Project group	This to incentivize users to make rides in order to win bonus points	YES
R13	The application should allow system admins to create exclusive public events	Project group	This in order to incentivize users to unlock badges	NO
R14	The application should allow a Team admin to propose a public team event that has to be approved by a System Admin	Project group	This in order compete with other teams and earn more points	YES
R15	The application should display to the user the available events and allow users to enroll for them	Project group	This to allow users to compete and earn more points	YES
R16	The application could allow team admins to invite other teams to join private events	Project group	This to give team admins the possibility to challenge other teams	YES
R17	The app should give extra points to the user if they bike in adverse weather	Customer	This to incentivize the usage of bicycles even in adverse weather condition	YES
R18	The web-app should display all the user's statistics	Customer	Doing so, users can check their progress without the need of the mobile app	YES
R19	The app could alert the user when the air quality is low	Customer	This is to warn consumers that cycling in such conditions may be dangerous to health.	YES

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

3.3.2 Non-Functional Requirements

ID	Description	Source	Motivation	Satisfied?
NR1	Do not ask for or store user's personal information like name, surname, age, phone number and address. Only the email is required, in order to let the user login. All the stored data should not be used for purposes different from the ones of the application and should not be shared with any third party company.	Customer	Users should feel comfortable by not sharing their sensitive data with third party companies.	YES
NR2	Prevent battery drain while using the application, having no more than a 25% battery discharge per hour.	Customer	Having a power-consuming app leads users to uninstall it	YES
NR3	Localization permission must only be requested and used during the ride tracking phase. Any other functionality of the application must be available also without these permissions granted.	Customer	Users should be able to do all the basic operations with the application without the need of granting permission when not needed.	PARTIALLY

3.3.3 Missing requirements and possibilities for improvements and extensions

In this project all the requirements were implemented and satisfied, except requirement R13 (the application should allow system admins to create exclusive public events).

Then, the requirement NR3 was partially satisfied since the application, in the current state, always retrieves the localization after the map is loaded but, if a user does not give permission to retrieve the position or location is disabled, the app can still work.

PedalaMi is a big and complex project, but it could be improved in the future by:

- implementing requirement R13,
- implementing an algorithm to check if the user is really riding a bike,
- adding a live chat for the members of a team,
- improving the point assignment algorithm,
- check statistics of members of the same team,
- adding a feature that could find nearby riders to ride together or challenge each other live,

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

- showing the path on a map of an old bike ride also on the web dashboard (currently not implemented because OpenStreetMap plugin is not supported on web).

3.4 Guidelines for software installation and operation

This section illustrates how to install the needed software. Note that for non-developers, it is only required to install the release APK, since no actions are required to configure the server. For the sake of completeness, also steps to install and configure the development tools are provided. Furthermore, no release package has been provided for iOS, due to the inability to test it.

3.4.1 *Installing Flutter (only for developers)*

First install an IDE that supports development with Flutter. For the development of our project we chose Android Studio as it was the most extensive IDE with Flutter support. A more lightweight alternative to Android Studio would be Visual Studio Code.

After installing Android Studio, install the Flutter plugin for Android Studio when you first open the IDE.

After the plugin setup, depending on the target OS (i.e., Linux, Windows or macOS) install the SDK from the Flutter website:

<https://docs.flutter.dev/get-started/install>

Inside the Android Studio application, navigate to *Settings -> Languages & Frameworks -> Flutter*. Write the path to the downloaded Flutter SDK in the Flutter SDK path.

Navigate to Dart that exists in *Languages & Frameworks* and enter the Dart SDK path. The Dart SDK is included in the Flutter folder downloaded before, under /bin or /bin/cache folder.

Now Android Studio should be ready for Flutter development.

3.4.2 *Adding a SHA certificate fingerprint for Firebase (only for developers)*

During development of the app it is needed to add the computer's SHA fingerprint in the Firebase console for security purposes, otherwise the application will crash when trying to establish a connection to Firebase.

For this step a *Java Development Kit (JDK)* must be installed on the machine. JDK can be install from the following website:

<https://www.oracle.com/java/technologies/downloads/>

In a terminal, navigate to the android folder in the root of the project. Then, write the following

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

command to retrieve the fingerprint.

```
./gradlew signingReport
```

Or for Windows users:

```
.\gradlew signingReport
```

The command will most likely return several different SHA keys. The needed key is the one with variant name debug and config name debug.

Provide the retrieved fingerprint to the PedalaMi admins, who will add the fingerprint to the Firebase Console.

The connection is made with the file found in:

Project root -> android -> app -> google-services.json.

To run the app, select the desired target device in Android Studio and select the `main.dart` file in the run configuration.

3.4.3 Installing the app via APK (only Android)

The following guide refers to the Android app. No steps required to configure and run the server, as it is hosted on Heroku and publicly available.

The target smartphone must have the “Install unknown apps” option enabled. Since the procedure to enable that option is strictly phone-dependent, it is not provided in this section.

Once a user has enabled the “Install unknown apps” option and has downloaded the APK file on his mobile phone, he has to open it with a file manager in order to start the installation procedure. At this point the user has to give the necessary permissions to install the app and, after a while, the application will be ready to be opened.

3.4.4 Installing NodeJS and running the server locally (only for developers)

To run the server locally, it requires NodeJS installed on the target machine. According to the platform, follow the guide at <https://nodejs.org/it/download/package-manager/> to install NodeJS and npm.

Then, be sure to have all dependencies resolved by running

```
npm install
```

in the project root folder (i.e., where the `server.js` file is contained).

Finally run

```
npm start
```

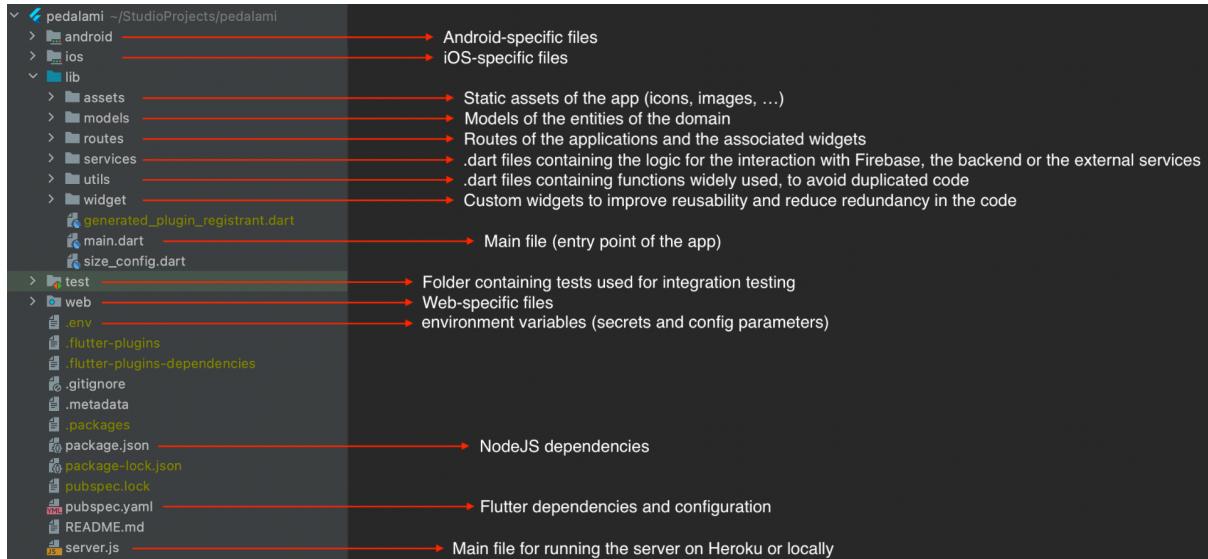
in the project root folder.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

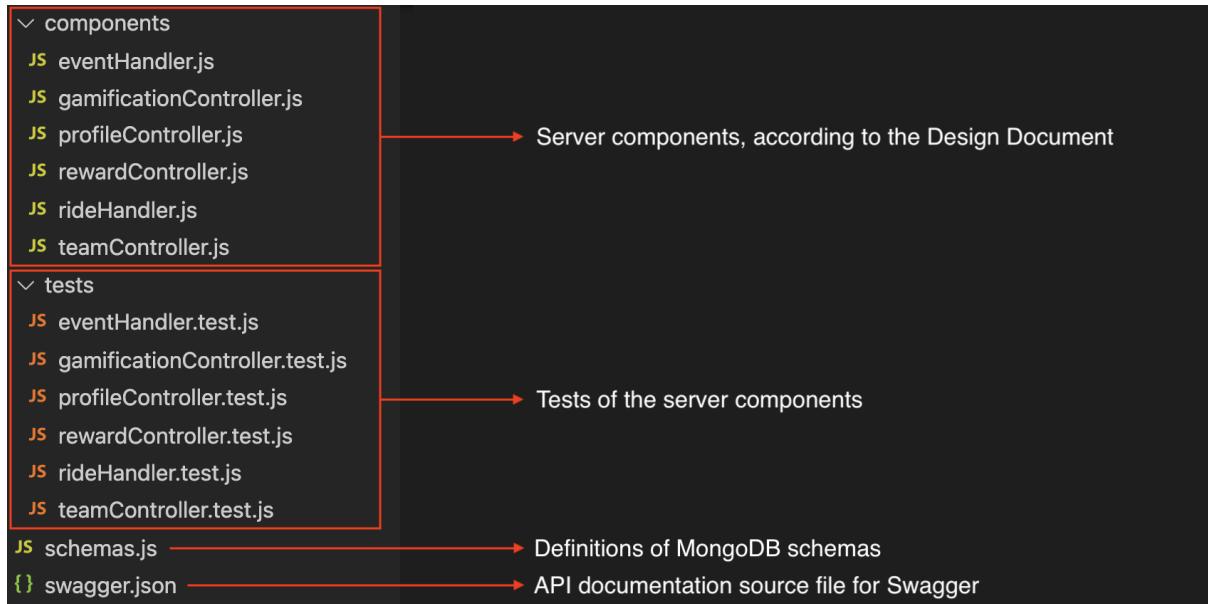
Note that this step requires the `.env` file containing the secrets for establishing the connection with the database. Anyway, for security reasons, that file is not published on GitHub and should be provided by PedalaMi admins on explicit request.

3.5 Structure of the source code

3.5.1 Front-end



3.5.2 Back-end



PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

4. Project work

The project was developed by a team of 9 people, where 6 students were from Politecnico di Milano (Polimi) and 3 - from Mälardalen University (MDH). Initially our team consisted of 10 people, but shortly after the development phase began, one student dropped out of the project.

Project work was planned and evolved around the deadlines of the deliverables listed in section 3.2 [Deliverables](#), as well as based on expected delivery dates for different phases of the product (MVP, Alpha and Beta prototypes, and Final Product).

For more detailed information about the PedalaMi project planning see the Project vision and plan document.

4.1 Work organization and routines

The whole project work was organized according to the agile method called Scrum and by adapting its main components like Sprints, Sprint planning, daily stand-up meetings, etc. to our team's needs and possibilities.

Our project development process was divided in four different sprints, each taking two weeks or ten working days to complete. Velocity for each sprint was calculated based on initial Product Backlog (see Table 1 at the end of this section), where after estimating the effort required by each user story in points, points were summed-up and divided by the number of sprints (i.e., by 4). Besides all the tasks for the product development process, also document writing was included in each sprint.

Telegram was used daily if a quick response was needed by any of the team members: in that way the communication was fast and direct and the entire team was involved, solving any occurring problem.

In the beginning daily stand-up meetings were held every day on Discord, but due to the differences in team member's schedules, daily meetings were then held either individually or in a setting only of sub-team. Later, the entire team was updated with the necessary information through Telegram or through the shared folder of sprint documents on Google Drive. Google Drive was also used for any file sharing between the team members and for shared document writing.

Initially, meetings with the customers were scheduled to be held at least once every two weeks by the entire team but later, only the Product Owner stayed in contact with the customers to provide the rest of the team with any clarifications or feedback needed on the work in progress.

The meetings with supervisors were held every Friday morning on Zoom. Supervisors gave answers to every question, guided the team during the complicated project work, and also gave highly appreciated feedback for the work that has been done. Usually, after the meetings with supervisors, an internal team meeting was held to recap the status of the project and decide further steps needed.

During the development process GitHub was used for the version control to ensure that all the

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

material is backed-up online and always available for each team member, supervisors and also for customers. Also the project feature of GitHub was helpful, to keep track of the tasks status on a kanban board.

The PedalaMi server was implemented using NodeJs and deployed to Heroku. App data was mainly stored in a document-based database, MongoDB, while Firebase was used mainly for authentication-related storage.

In order to create a hybrid mobile application and a web dashboard, the front-end part for the software was developed using Flutter framework.

The overview of the system is shown in Figure 1.

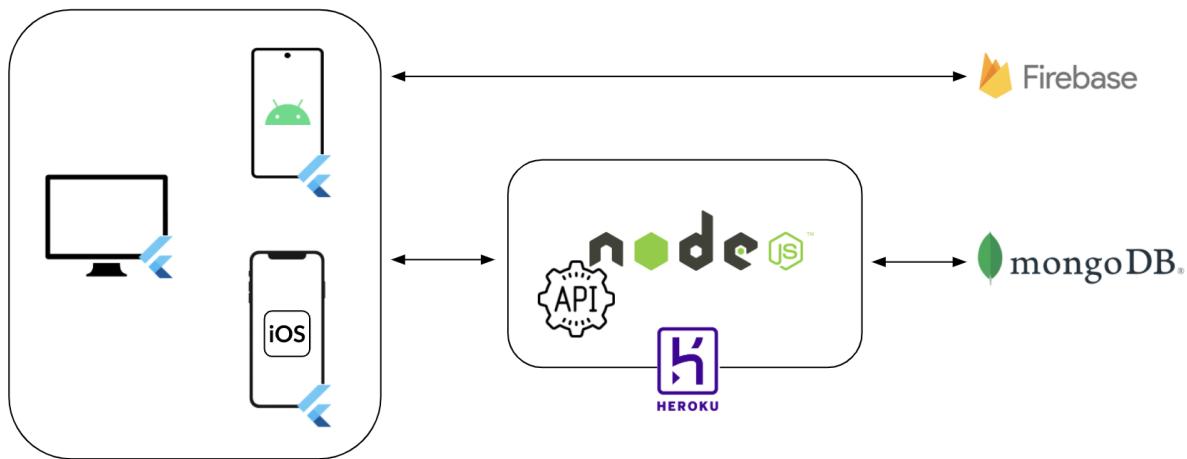


Figure 1: High-level overview of the entire system

Product Backlog

ID	Name	Priority	Estimated effort (points)	Sprint
Non registered users				
US1	Account creation	5	10	1
Registered User				
US2	User authentication	5	8	1
US3	Profile Editing	3	14	1
US4	Record a Ride	5	15	2
US4.1	Provide a Ride History	4	12	3
US5	Points Collection	5	7	2

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

US6	Badge Unlocking	5	8	3
US7	View Reward	3	8	3
US8	Redeem a Reward	5	6	3
US9	Team Creation	5	15	2
US10	Team Search & Participation	3	12	2
US11	Events Overview	3	11	3
US12	Events Enrolling	3	10	3 (5p); 4 (5p)
US13	Bonus for Adverse Weather	1	9	4
US14	Check user statistics	5	14	2 (4p); 3 (10p)
US15	Retrieve user information on the web app	1	15	4
US16	Air quality alert	1	8	4
Team admin				
US17	Private Team Events Creation	3	13	4
US18	Public Team Events Proposal	1	7	4
System admin				
US19	Public Individual Events Creation	3	8	3
US20	Public Team Events Approval	1	4	4
US21	Exclusive Events Proposal	1	4	-
	Document writing	5	40	1; 2; 3; 4

Table 1: Product backlog

4.1.1 Team members' roles and responsibilities

For the overall development process the team was divided into two distinct sub-teams: one for the front-end development and other - for the back-end development. The role for each team member was decided and assigned early in the project planning phase to ensure the clear vision of its responsibilities.

Initial role assignment for the team of 10 members are shown in the table below:

	Member	Role	University
1	Liene Andersone	Front-end Developer	MDH
2	Panagiotis Bampilis	Front-end Developer	MDH

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

3	Gezim Bllicu	Front-end Developer	MDH
4	Marcus Mörstrand	Scrum Master & Front-end Developer	MDH
5	Massimiliano Giovagnola	Front-end Developer	POLIMI
6	David Støvzbæk	Back-end Developer	POLIMI
7	Lorenzo Italiano	Product Owner & Back-end Developer	POLIMI
8	Vincenzo Riccio	Back-end Developer	POLIMI
9	Giancarlo Sorrentino	Back-end Developer	POLIMI
10	Emanuele Triuzzi	Back-end Developer	POLIMI

However, during the first sprints the workload of each sub-team became unbalanced due to uneven possibility to deliver finished tasks on time, and lack of knowledge for some team members from the front-end team to complete the assigned tasks. Later on, the decision was made to move one member from the back-end team to join the front-end sub-team. Also the Scrum Master role changed after the second sprint to ensure the possibility for one more team member to experience this responsibility. The product owner stayed the same person throughout the project hence the need to communicate with customers regularly.

The final setting of the team members, their roles and responsibilities are shown in table below:

	Member	Role	University
1	Liene Andersone	Front-end Developer	MDH
2	Panagiotis Bampilis	Front-end Developer	MDH
3	Marcus Mörstrand	Front-end Developer	MDH
4	Massimiliano Giovagnola	Front-end Developer	POLIMI
5	David Støvzbæk	Front-end Developer	POLIMI
6	Lorenzo Italiano	Product Owner & Back-end Developer	POLIMI
7	Vincenzo Riccio	Scrum Master & Back-end Developer	POLIMI
8	Giancarlo Sorrentino	Back-end Developer	POLIMI
9	Emanuele Triuzzi	Back-end Developer	POLIMI

4.2 Total project effort

PedalaMi was a very demanding project, partially due to the lack of confidence of many team members with the technology stack.

The entire project required a total of 1579 working hours, spread across 4 sprints, which were preceded by a warm-up phase and followed by this current concluding phase.

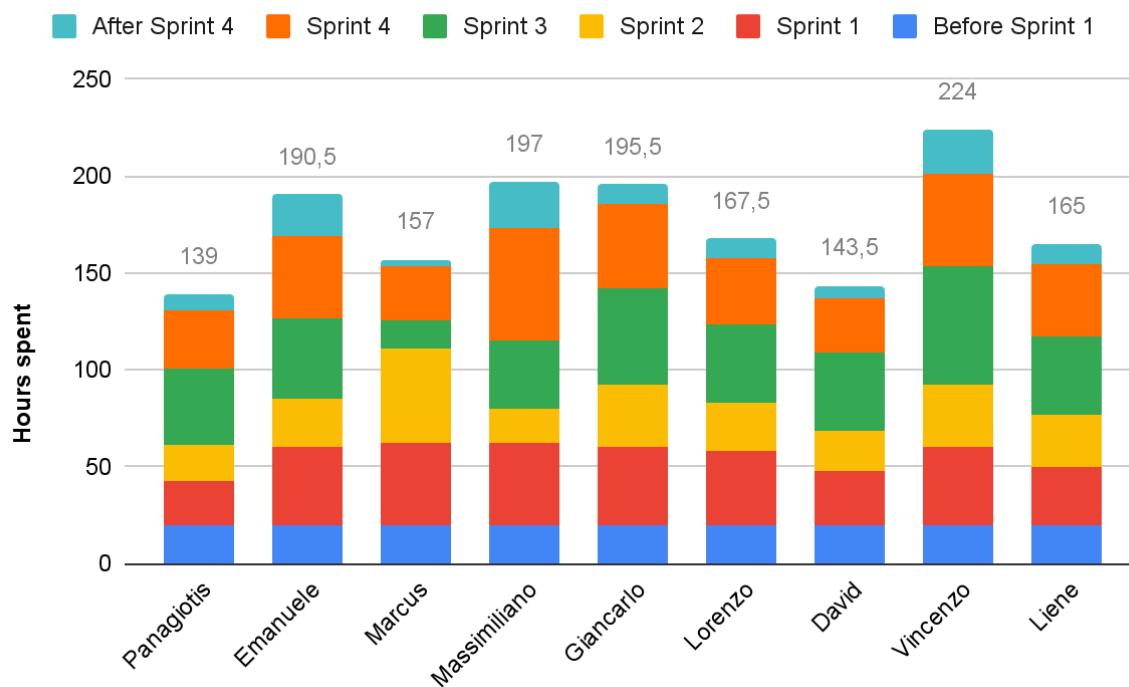
The warm-up phase was mainly about agreeing on many organizational choices, about setting up all the tools that were supposed to be used in the development and about sharing a common vision of the project. It required an average of 20 hours per team member, most of which were spent together in online meetings.

The working hours spent by each team member in each sprint are illustrated in the following table.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

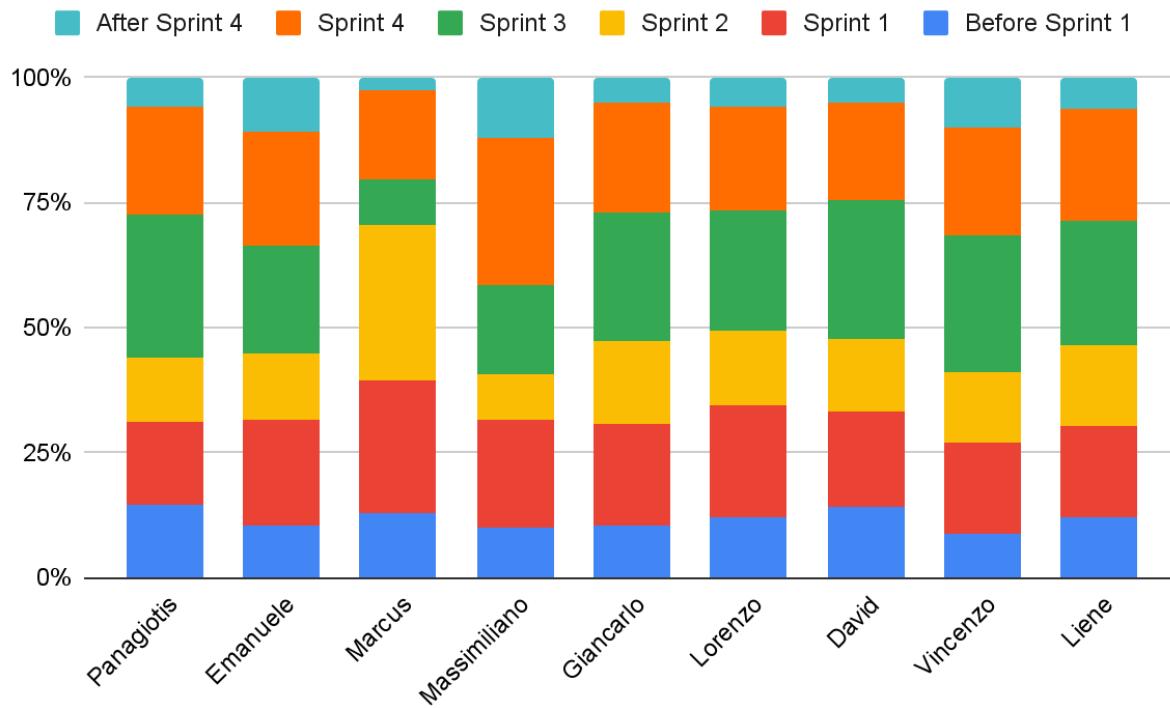
	Before Sprint 1	Sprint 1	Sprint 2	Sprint 3	Sprint 4	After Sprint 4
Panagiotis	~20	23	18	40	30	8
Emanuele	~20	40	25	41,5	43	21
Marcus	~20	42	49	14	28	4
Massimiliano	~20	42	18	35	58	24
Giancarlo	~20	40	32,5	50	43	10
Lorenzo	~20	38	25	40	35	9,5
David	~20	27,5	21	40	28	7
Vincenzo	~20	40	32	61,5	48	22,5
Liene	~20	30	26,5	41	37	10,5

Details on the distribution of working hours between the 6 phases of the project are provided through the following graphs.

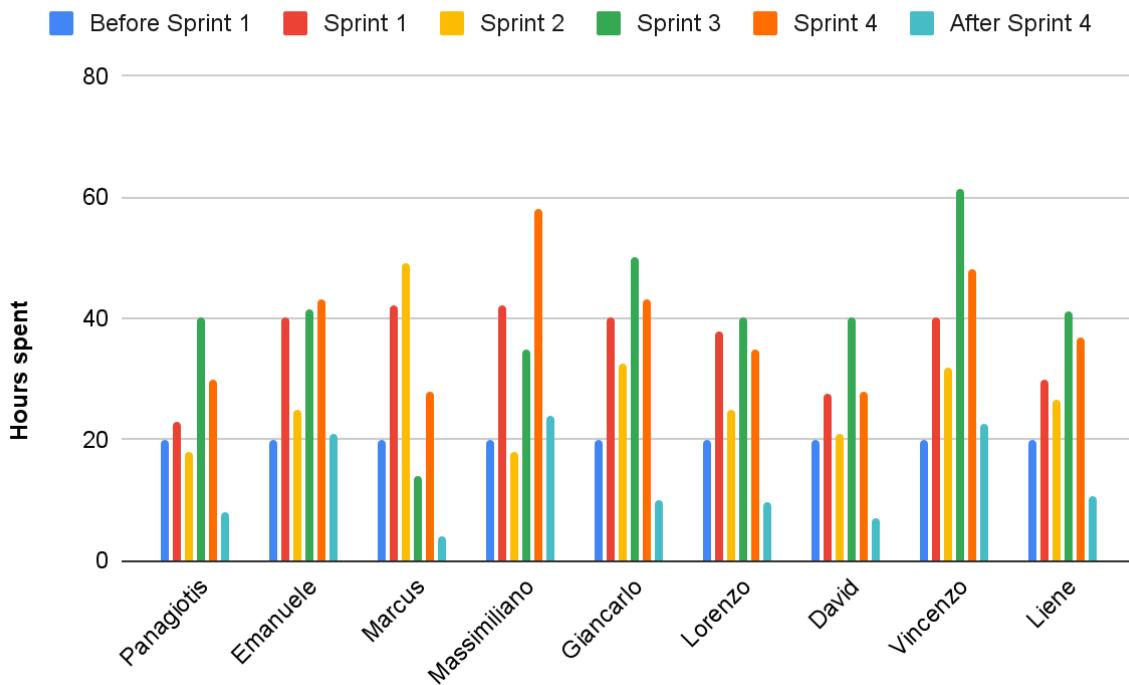


Graph 1: Distribution of each member's working hours across the different phases of the project

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

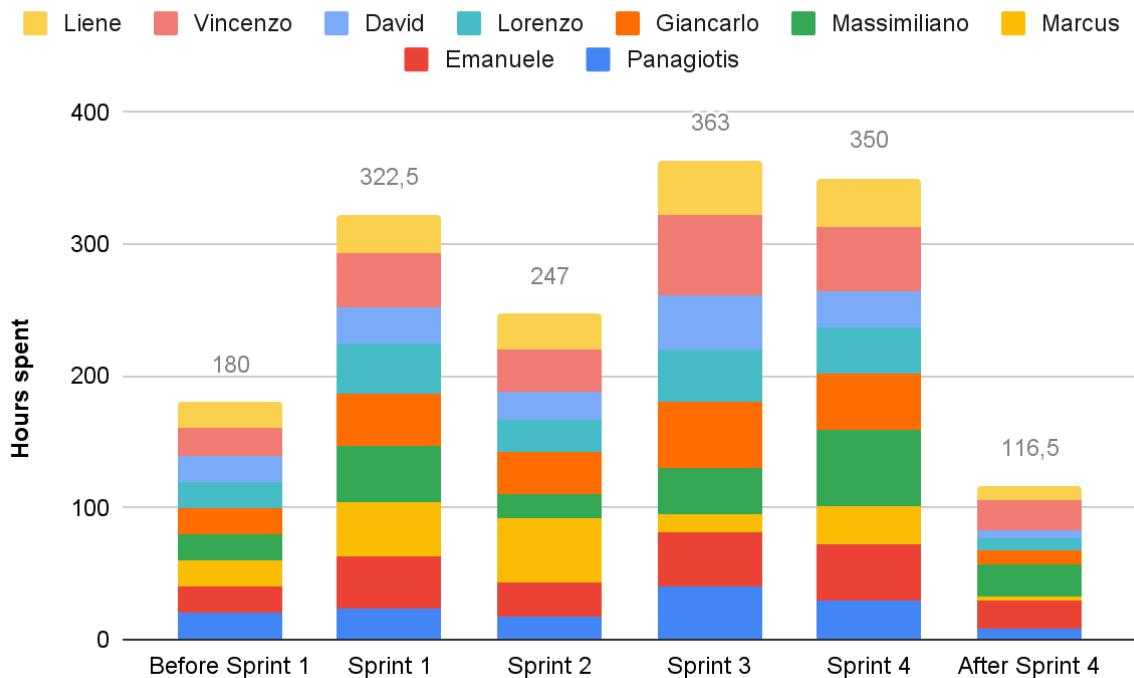


Graph 2: Effort distribution (in percentage) of each team member across the different phases of the project



Graph 3: Hours spent by each member during the different phases of the project

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14



Graph 4: Total hours required by the different phases of the project

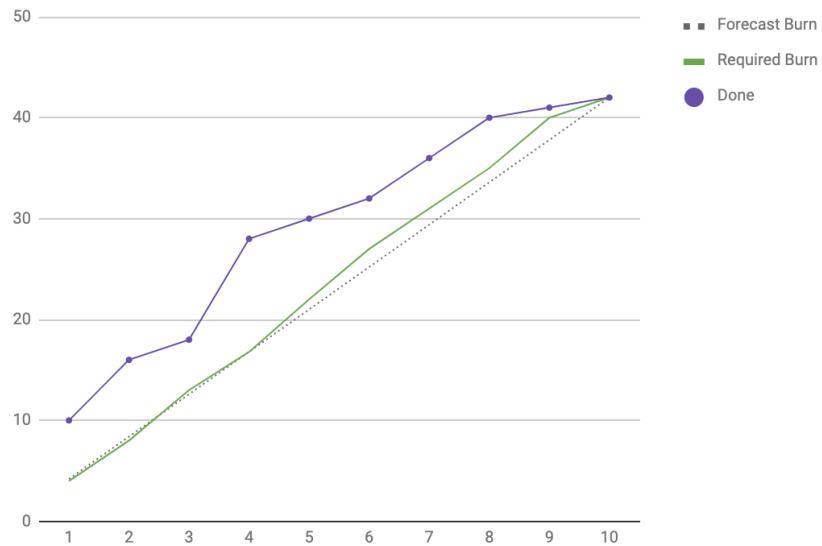
4.3 Sprints analysis

This section better analyzes the 4 sprints done. As the team workload is very dynamic due to the lack of expertise and experience in development with the adopted technology stack, user stories and sub-tasks were not assigned to a static group of people.

The first sprint was focused on laying the basis for the mobile application and the backend. It also included a consistent formative part on the technology stack that had to be used.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

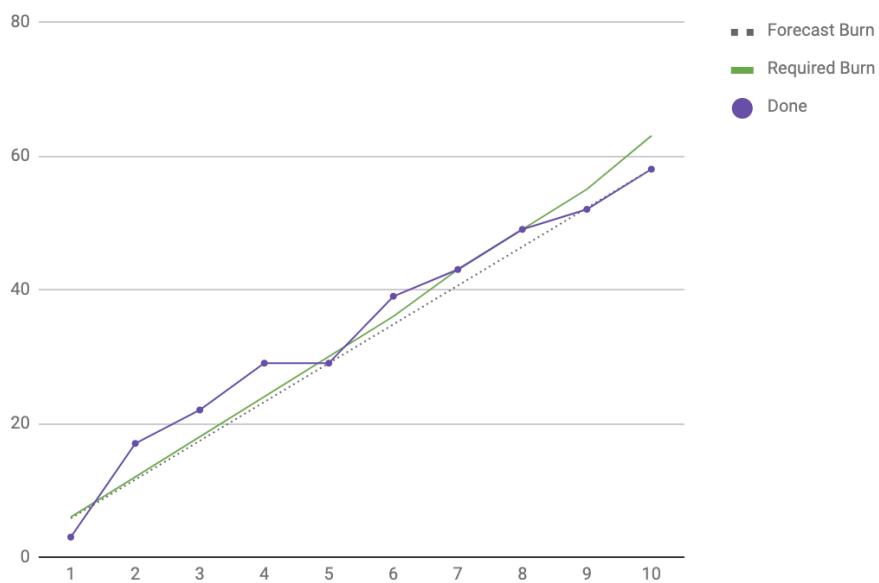
Sprint 1 Points Burn Up



Burn Down Chart 1: evolution of sprint 1 (total points done) per working day

Most of sprint 2 was spent implementing the map and ride recording features and establishing a working communication between the front-end and back-end. It was the slowest sprint, due to many problems with the map and due to implementation getting more complex for beginners in Flutter development. It also included the handling of teams across the entire system.

Sprint 2 Points Burn Up

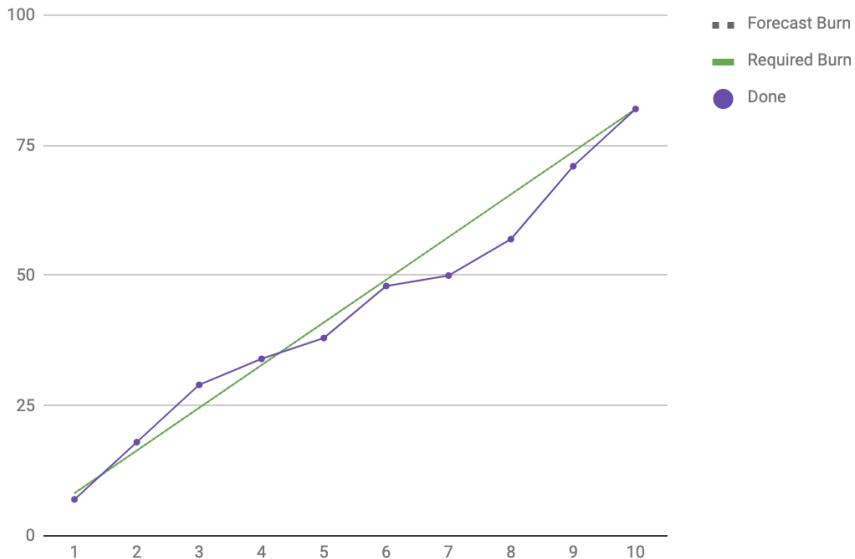


Burn Down Chart 2: evolution of sprint 2 (total points done) per working day

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

Sprint 3 was a huge sprint in terms of functionalities to implement. It started by dealing with uncompleted tasks from sprint 2. Then, the gamification part of PedalaMi was the core of the sprint, which ended with the implementation of the first event-related features, especially backend-side.

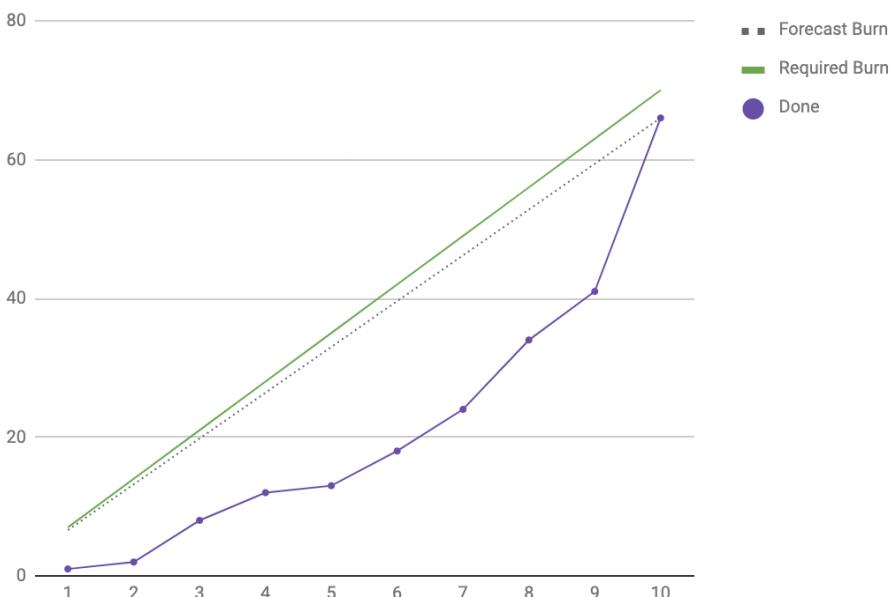
Sprint 3 Points Burn Up



Burn Down Chart 3: evolution of sprint 3 (total points done) per working day

Sprint 4 was the last one, and focused on completing all the events features. However, it also covered the implementation of most of the optional features and of the web dashboard. Finally, a consistent amount of effort was put into refinements and bug fixing.

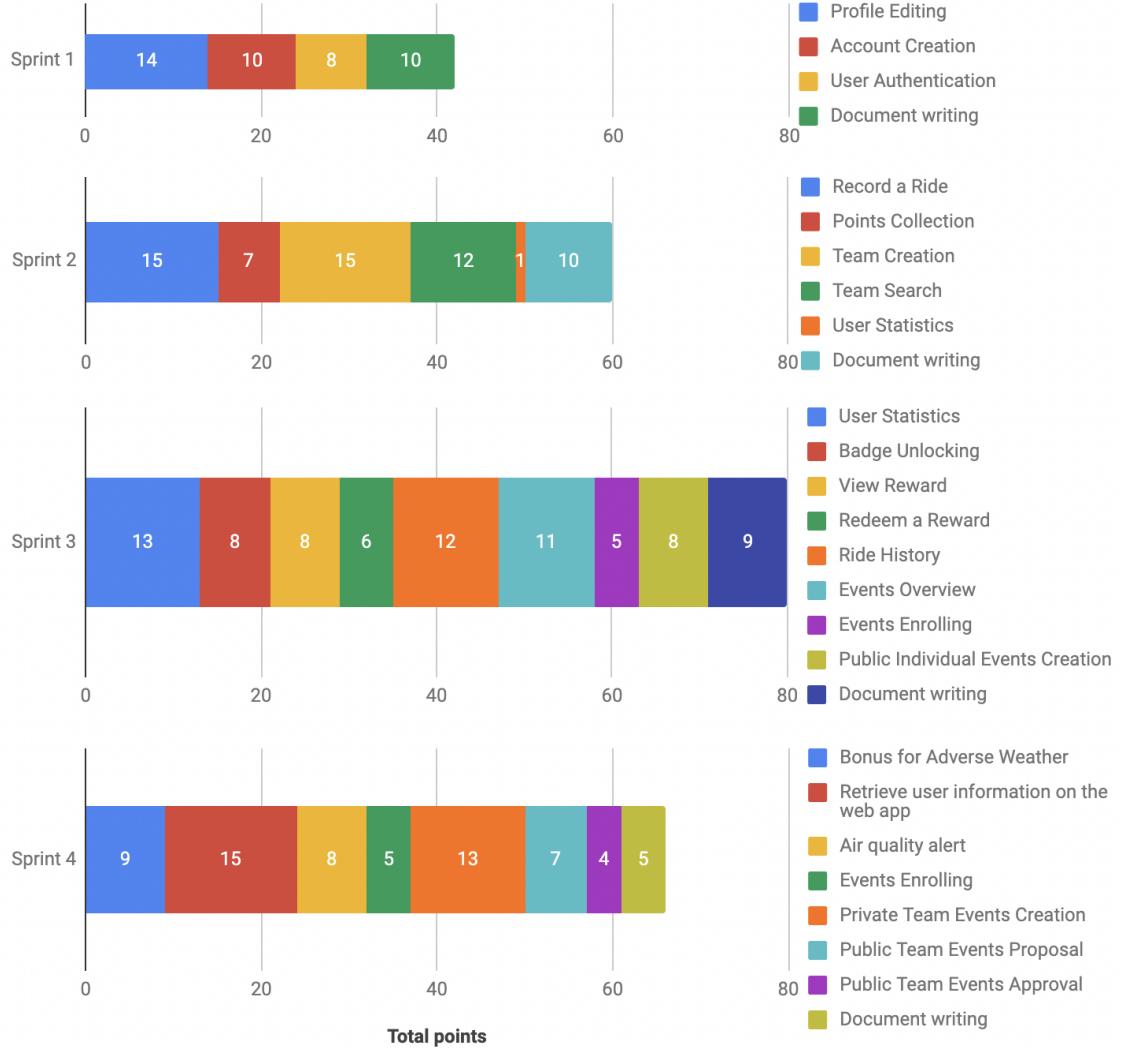
Sprint 4 Points Burn Up



Burn Down Chart 4: evolution of sprint 4 (total points done) per working day

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

Graph 5 details each sprint composition and progress, in terms of user stories and points.



Graph 5: Distribution of the user stories across the 4 sprints

4.4 Positive experiences

Despite being a demanding experience, it was a very positive one for many reasons.

Generally, the entire team did a great job on the project, in a professional and committed fashion. Each member contributed a lot to the project, regardless of the assigned role, and soon started to participate in team discussions: major decisions were always made together.

Even if the Scrum methodology was unknown to most of the team, it was quickly adopted and followed, becoming one of the most formative parts of the process.

Furthermore, each team member strictly followed our branching and code reviewing strategy, together with some common rules, learning a lot in terms of best-practices to follow when developing with other people.

PedalaMi	Version: 1.0
Final Report	Date: 2022-01-14

Being part of a large team, made by students of different backgrounds and skills, allowed each of us to share much more than technical knowledge: improving our English, our presentation, organization and communication skills was definitely a positive income for the whole team.

Moreover, we became familiar with new technologies (Flutter, Dart) and improved our ability to assess the work to do and the required time to complete it.

The experience of learning how to work only remotely proved to be a great learning experience, since it is where the world is going nowadays. Also, we had the great opportunity to improve our soft skills, and in particular how to communicate more effectively and professionally.

4.5 Possible improvements

While a number of good experiences were accomplished in the project, routines and some processes could still be improved.

While communication was generally direct and consistent, some misunderstandings arose sometimes, mainly due to an inaccurate overview of the project vision. We found, after some time, that it is very important to read the main documents over and over, multiple times, because otherwise it might result in uncertainty of what to do, or in an inappropriate implementation of features.

Initially during the implementation phase, we split up the project into two teams – backend and frontend. However, we didn't establish a policy regarding branching and git policies. In retrospect, it should have been done earlier, but after that change the workflow improved a lot since everyone followed the rules we agreed to.

As far as it concerns contributions, at the beginning of the second sprint we realized that one of the team members was not putting into the project the minimal effort required. At the beginning, we decided to give him more time to reach our pace. Perhaps we were too optimistic and we waited too long hoping for a change of direction by his side, which unfortunately never happened.