POLITECNICO
MILANO 1863

Software Engineering II

Travlendar+

Implementation & Testing Document

Authors:

Edoardo D'Amico
Gabbolini Giovanni
Parroni Federico

Git Repository: `https://github.com/keyblade95/DamicoGabboliniParroni`

1st December 2017

Indice

1 Introduction **2**

2 Project Analized **3**
2.1 Name of Authors of the analyzed project 3
2.2 Link to the repository 3
2.3 Main reference document considered 3

3 Installation Instruction **4**

4 Tests **5**
4.1 Test of the implemented functionalities 5
4.1.1 Insertion of events and free times 5
Test1 8
Test2 9
Test3 10
Test4 10
Test5 10
4.1.2 Notification System 11
4.1.3 Setting Preferences 12
4.1.4 Persistance Database to store events and free times 13
Test 1 14
Test 2 15
4.1.5 BikeMI API 16
4.1.6 Wheather API 16
4.1.7 Preferences Management 16
Test 1 16

5 Additional Points **18**

6 Effort Spent **19**

Chapter 1

Project Analized

1.1 Name of Authors of the analyzed project

- Michele De Pascalis,
- Gianmarco Dello Preite Castro,
- Amin Mahboubi.

1.2 Link to the repository

`https://github.com/Gianmarcodpc/DePascalisDelloPreiteCastroMahboubi`

1.3 Main reference document considered

The Main document considered is the ITD.

Chapter 2

Installation Instruction

The software has been installed trough the apk whitin the deliverables folder in the github repository how said from the group in the ITD document. The application has been installed on two different virtual devices, the nexus 5 emulator, with android version 8.0(Oreo) and in the nexsus 5x with the same version of android, that has been done for test how the application can adapt itsel to different display size, 5 for the first one and 5.5 for the second one. There were no inchorency on the installation instruction released on the ITD apart for the presence of the application in the google playstore, in fact the application can't be downloaded from it.

Chapter 3

Tests

3.1 Test of the implemented functionalities

The tests have been done on the implemented functionalities declared on the ITD document released by the other team, for each of them at least on test has been carried out.

3.1.1 Insertion of events and free times

The insertion of an event can be done by clicking the button the main page of the application

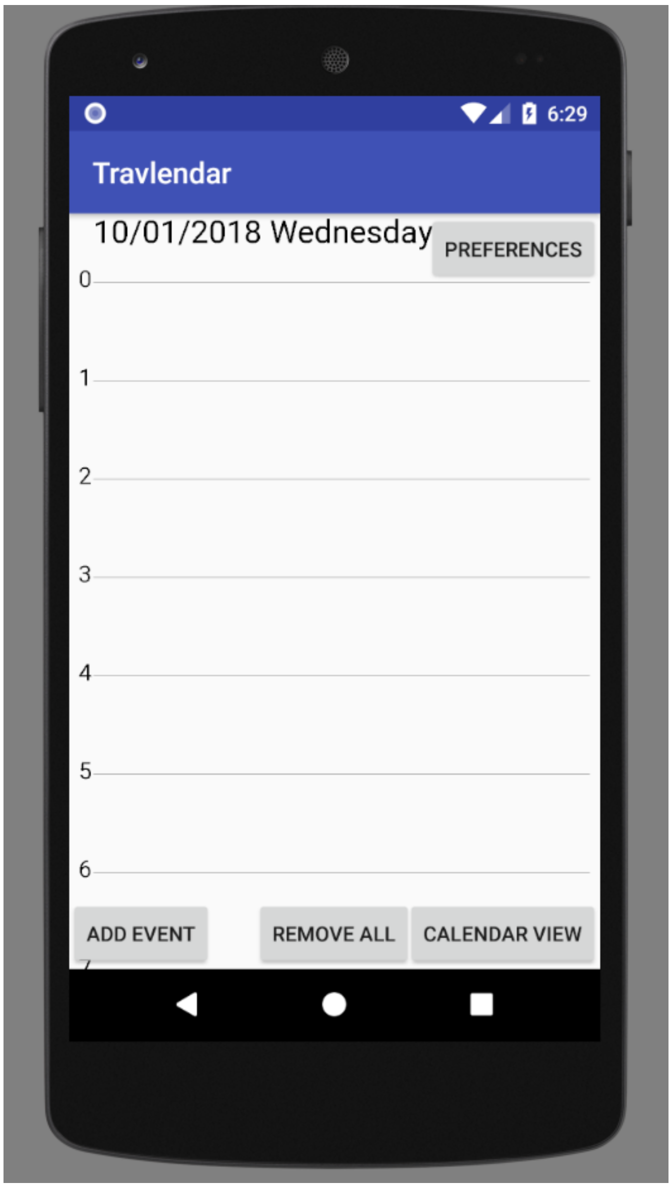


Figura 3.1: Main page view

The user is redirect to an event creation view with some fields to fill as shown below.

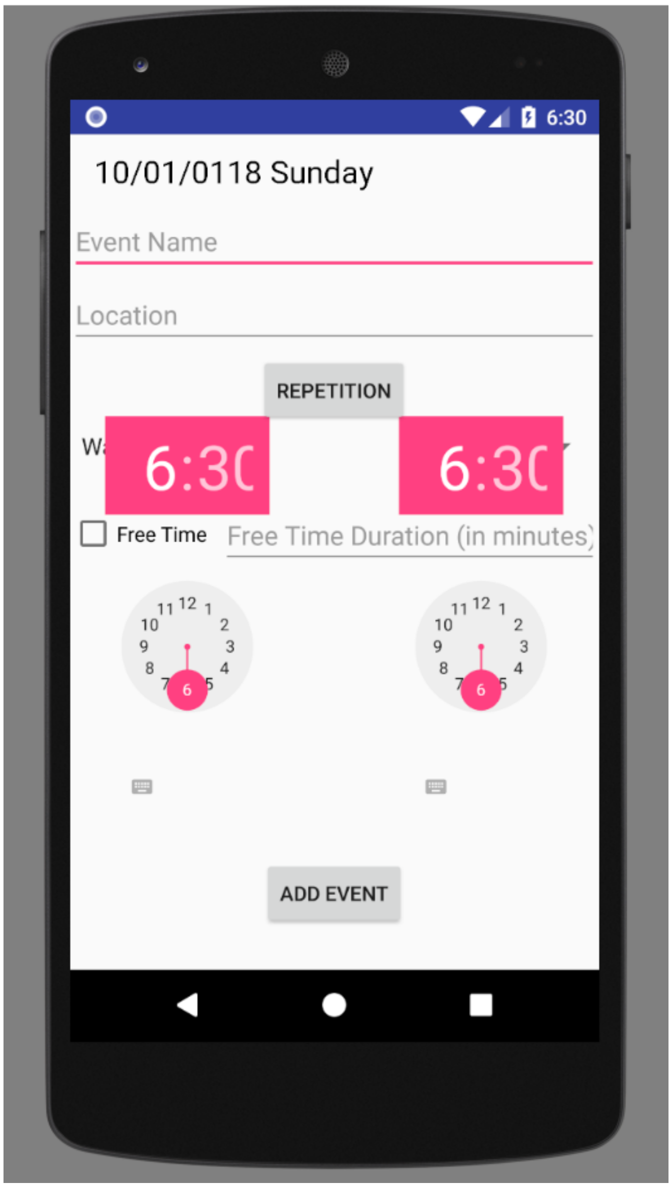


Figura 3.2: Event Creation View

The view has some missing graphic constraints that put the two clock picker in the middle of the view hiding the bar that made possible the selection of a travel mean for the event that is currently being created.
Another small issue is that the year of the selected day is wrong(is 0118 instead of 2018).
There isn't any check for correctness of the location inserted.

Test1

A first test has been carried out with a normal insertion of an event without free time and without repetition. After the click of the add event button the user is redirected to the main page but the appointment has not been added,although with a change of date on the calendar (change date and after rechange on the current date) the appointment is added, as shown in the image below,probably a little problem in refreshing the view after the insertion.

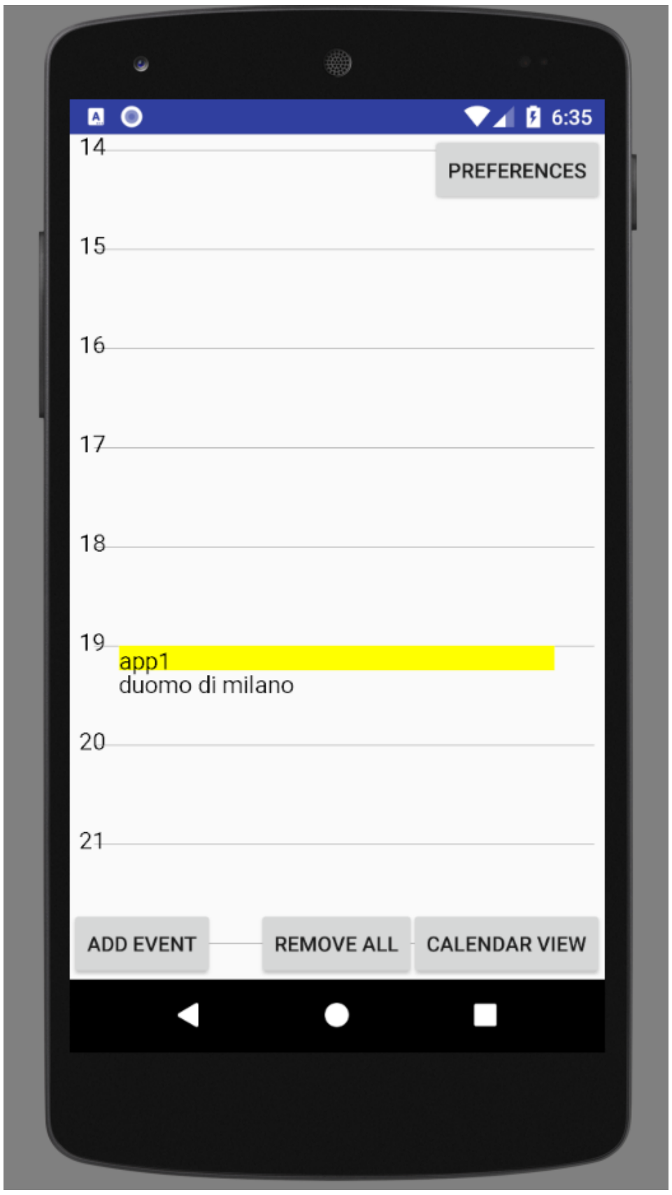
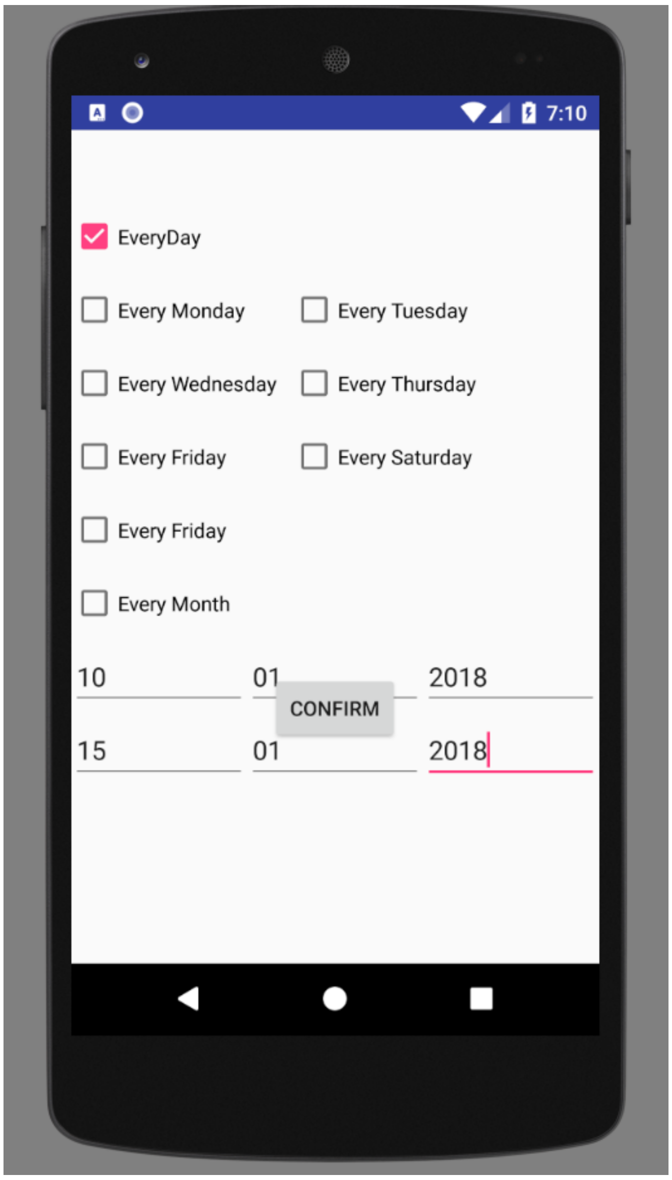


Figura 3.3: Insertion of an event

Another issue is represented by the insertion of an appointment with ending time which is previous than the starting time. In this case the application crashes

Test2

Here is tryed to add an event with repetition that is possible pushing the button on the event creation view named "repetition" the user is redirected to the following view



☑ EveryDay

☐ Every Monday ☐ Every Tuesday

☐ Every Wednesday ☐ Every Thursday

☐ Every Friday ☐ Every Saturday

☐ Every Friday

☐ Every Month

10	01	2018
15	01	2018

Figura 3.4: Repetition view

The confirm button has some missign constraints so it stay on the middle of the field for the insertion of the date.

After the click of the confirm button the user is redirected to the main page after the refresh of the view the appointment is visible only on the first date of the slot selected for the repetition, on the other days in which the appointment should be shown nothing has been added.

Test3

Here is tried to add two overlapping appointments, in this case after the insertion of the second one the application crashes.

Test4

Here is tried to add a free time, when the save button is clicked the application crashes.

Test5

As the other team has reported in theire **DD**, after the insertion of any appointment, the temporal feasibility of that appointment should be checked, meaning that the application should calculate the traveling times and therefore check if the user will be able to attend two consequent appointments without being late. This functionality actually is not working: the application lets the user insert two appointments that differ for one minute, even if they are located in very far places, like Via Golgi 42 and Duomo di Milano.

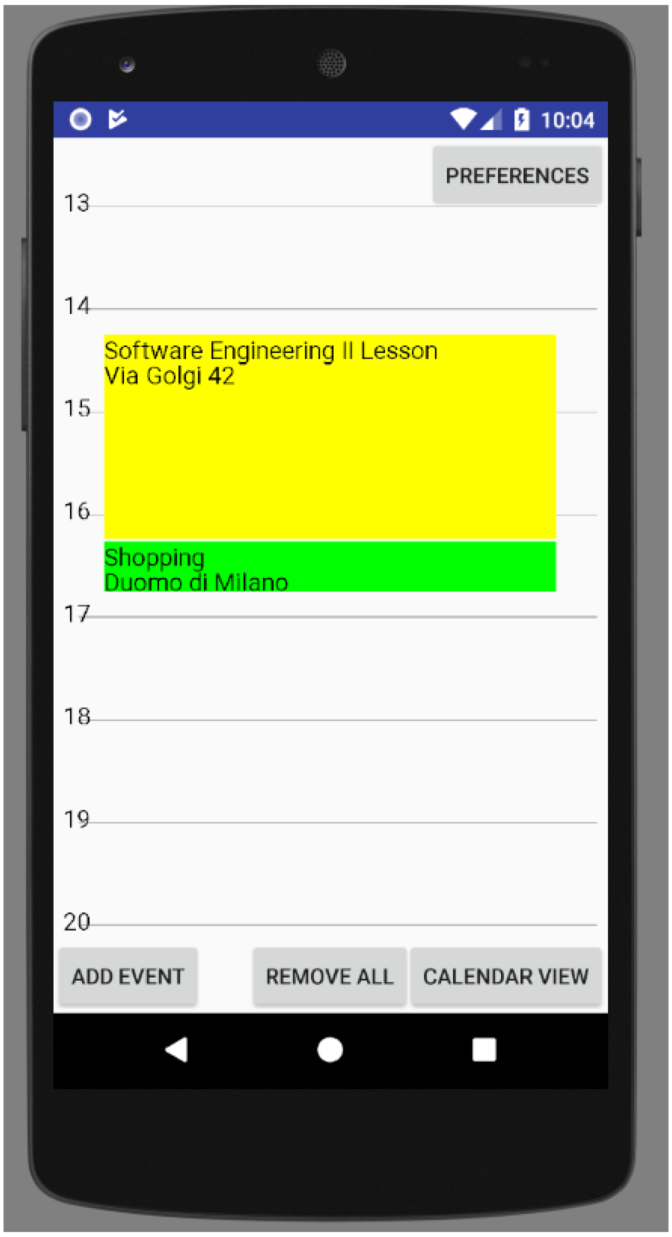


Figura 3.5: Temporaly close appointments

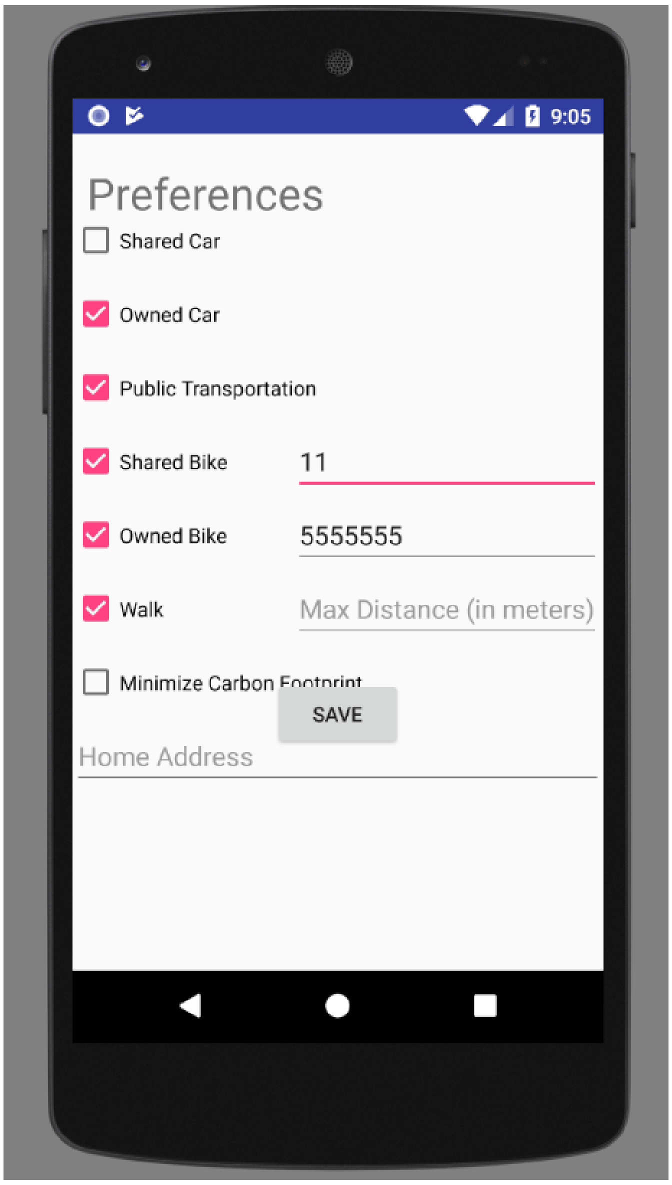
3.1.2 Notification System

This functionality is declared to be implemented on the ITD document but it isn't working, no notification are sent to the final user.

3.1.3 Setting Preferences

By clicking the preferences button on the main page we are redirected to the preferences page, where it's possible to set some user attribute, such as the ownage of travel means and of a transit travel pass, other than preferences on how many Kms the user wants to travel with some travel means.

One can argue that this view isn't complete at all: it's not possible to set the distance contraints for all the kind of travel means (for instace, for the public travel means). Moreover, it's possible to set as an optimization criteria just the carbon footprint emitted by the means taken, but not the cost optimization or the time optimization, as stated on the assignment given us.


9:05

Preferences

- [ ] Shared Car
- [x] Owned Car
- [x] Public Transportation
- [x] Shared Bike 11
- [x] Owned Bike 5555555
- [x] Walk Max Distance (in meters)
- [ ] Minimize Carbon Footprint

SAVE

Home Address

Figura 3.6: Preferences view

However, for the settings considered, the application works properly, letting the user save it's characteristics.

3.1.4 Persistance Database to store events and free times

The user preferences and the user events are declared to be saved inside the device in a persistence database.

Test 1

Here it's tested the persistence of an user event. First an event (let's say, a Software Engineering II lesson, located in Via Golgi 42, lasting 2 hours, from 14.15 to 16.15) it's added. After having closed the application, when it is opened again we can notice that the event it's still present.

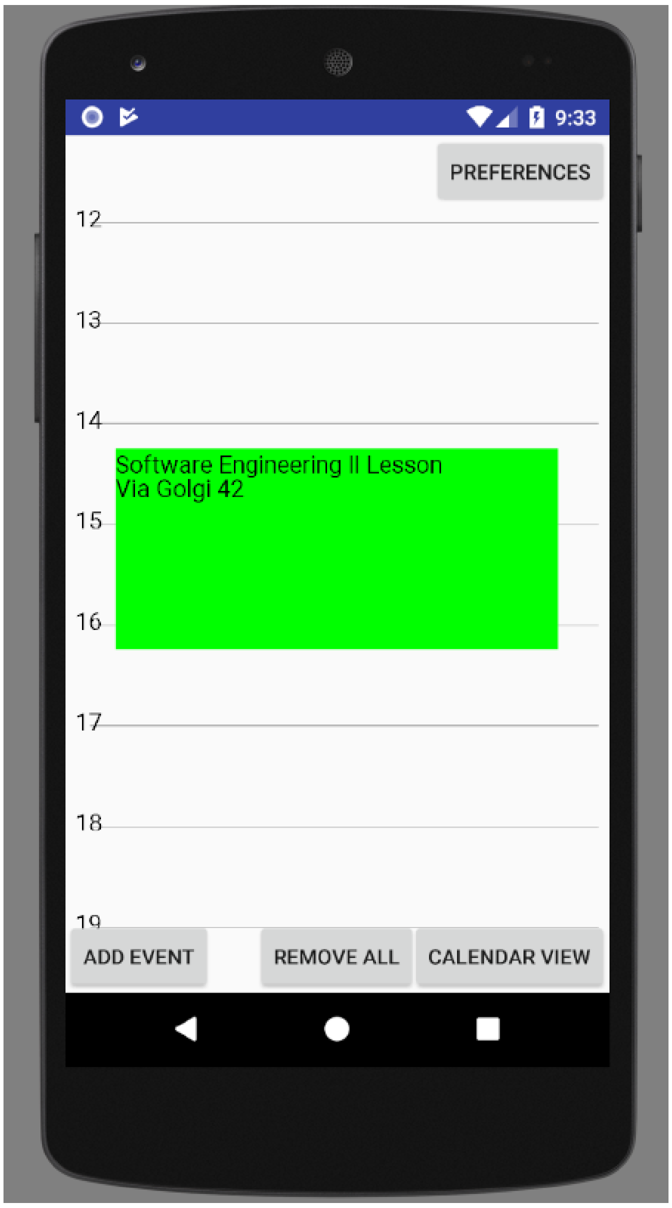
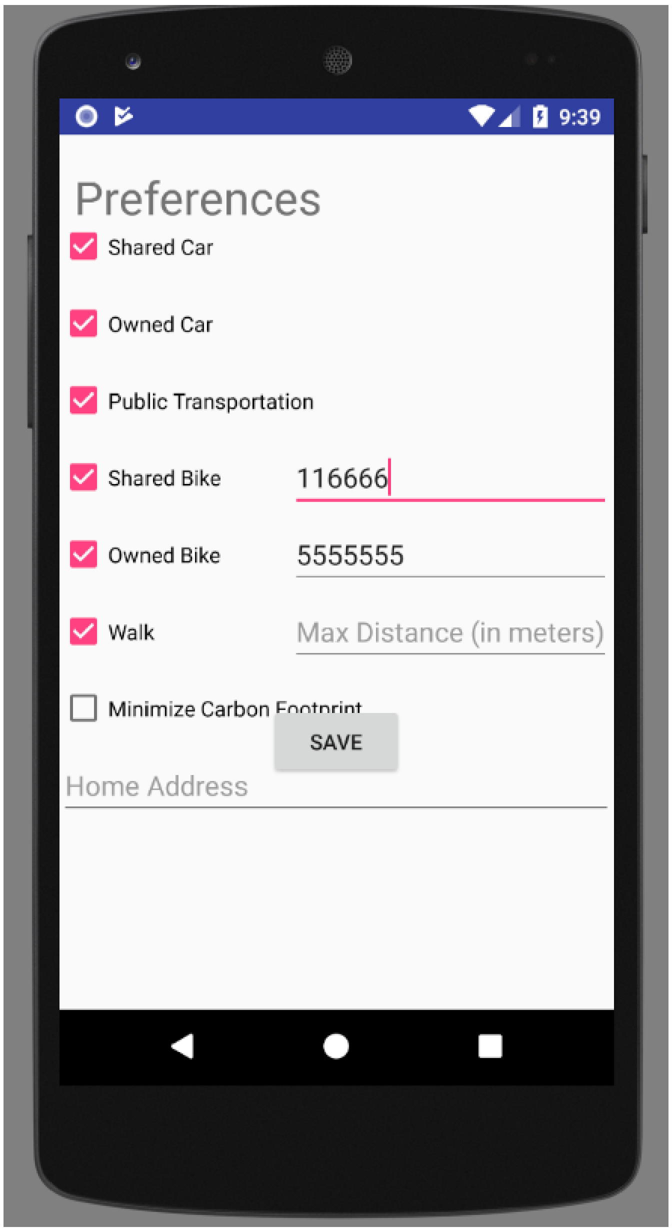


Figura 3.7: Persistence appointment

Test 2

Here it's tested the persistence of an user preference. First the ownage of a car is set to true. After having closed the application, when it is opened again we can notice that the car ownage checkbox it's still checked.



9:39

Preferences

- ✓ Shared Car
- ✓ Owned Car
- ✓ Public Transportation
- ✓ Shared Bike 116666
- ✓ Owned Bike 5555555
- ✓ Walk Max Distance (in meters)
- ☐ Minimize Carbon Footprint

SAVE

Home Address

Figura 3.8: Persistence appointment

3.1.5 BikeMI API

This functionality is declared to be implemented on the ITD document but it isn't working, no calls to this service are performed at all during the execution of the application.

3.1.6 Wheather API

This functionality is declared to be implemented on the ITD document but it isn't working, the weather conditions aren't checked.

3.1.7 Preferences Management

As declared, the user preferences are used during the computations to avoid some behaviour or to enhance others.

Test 1

This section it's not easy to test, infact most of the characteristics that has are encapsulated in the code. However, the coherence between the user's owned means and the means that the application admits to reach an event can be checked: first it's set in the preferences that we don't own a car. Then is set up a new appointment and, as transportation mean, we set our owned car. The application allow the insertion of this appointment, even if this is incoherent with the preferences.

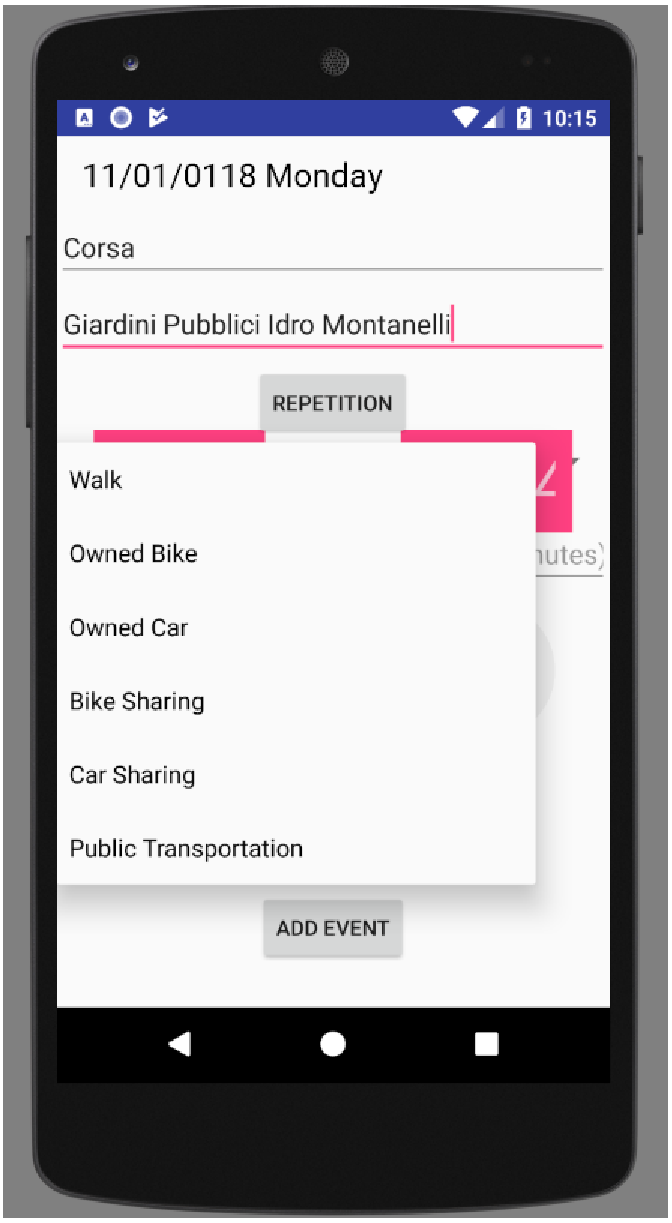


Figura 3.9: Preference Coherence

Chapter 4

Additional Points

Chapter 5

Effort Spent

- Federico Parroni: **130 hours**;
- Edoardo D'Amico: **130 hours**;
- Giovanni Gabbolini: **130 hours**.