POLITECNICO
MILANO 1863

Travlendar+
Acceptance Test Delivery

Fumagalli Paolo, Grotti Pietro, Gullo Marco

January 14, 2018

Contents

1 Introduction

1.1 Purpose

The following document aims at showing the Acceptance Testing results of the analysis our team conducted on team D Amico - Gabbolini - Parroni s project. We were asked to go through all the documents that team has produced (RASD, DD and ITD) and verify the if their first prototype of the application was coherent with them.

1.2 Scope

We could not find inconsistencies between the RASD and the DD, that appeared very well related. ITD fitted perfectly in the project context as well.
Our team decided to focus on verifying the goals and requirements that the other team claimed to have successfully implemented, checking if errors or anomalies occurred.
Some acceptance test cases were executed, one case for each goal: in Acceptance Test Results section it is possible to see if any trouble came out in the mobile version (.apk file) provided.

2 Group Identification

2.1 Authors

- Edoardo D'Amico
- Giovanni Gabbolini
- Federico Parroni

2.2 Repository

https://github.com/keyblade95/DamicoGabboliniParroni/

2.3 Reference Documents

- RASD: Requirement Analysis and Specification Document, Version 1.1
- DD: Design Document, Version 1.1
- ITD: Implementation and Test Deliverable, Version 1.0

3 Installation Instructions

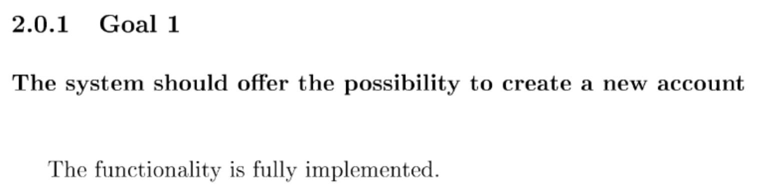
To install the software we had to download an APK from a dropbox folder provided in their ITD. The APK worked as expected, we installed the application in three different devices, a Samsung Galaxy S8, a Sony Xperia Z3 Compact and a Huawei P8 Lite 2017 and tested the application with these phones.

4 Acceptance Test Cases

Here are shown all the cases we believed as most relevant in the implementation evaluation. All the pictures below are extracted from their ITD.

4.1 Server Side

4.1.1 Registration



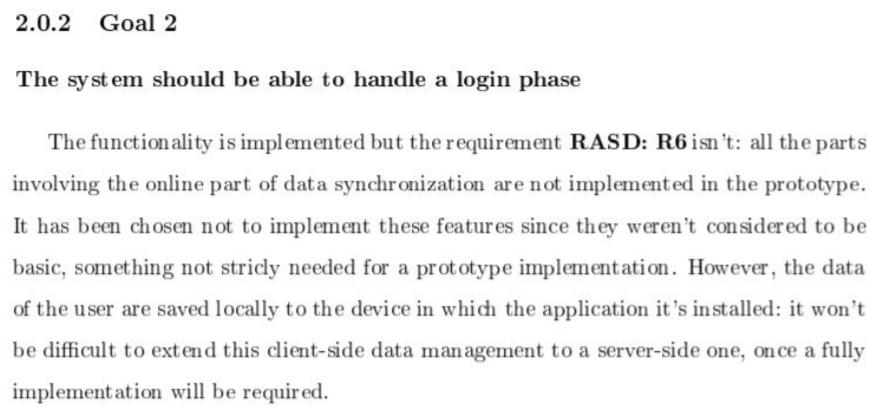
2.0.1 Goal 1

The system should offer the possibility to create a new account

The functionality is fully implemented.

We tested the registration by inserting a valid email address and a password. The result came as expected, the account was successfully created.
We also tried to register with an already used email address and the application didnt allow us to register, as it would be expected, because the email was already used.
Future development tip: the client application should wait for the server to notify the registration result, without allowing the user to type anything in the registration form.

4.1.2 Login



2.0.2 Goal 2

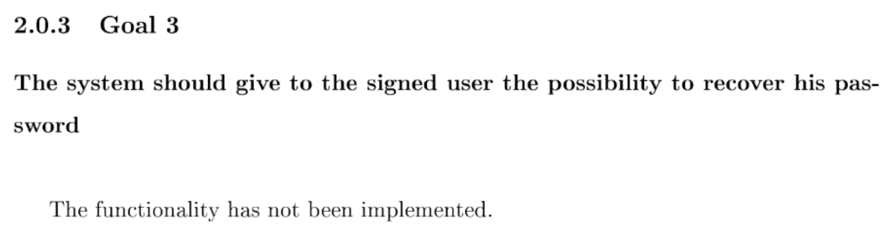
The system should be able to handle a login phase

The functionality is implemented but the requirement **RASD: R6** isn't: all the parts involving the online part of data synchronization are not implemented in the prototype. It has been chosen not to implement these features since they weren't considered to be basic, something not stricly needed for a prototype implementation. However, the data of the user are saved locally to the device in which the application it's installed: it won't be difficult to extend this client-side data management to a server-side one, once a fully implementation will be required.

The login was also successful. We tried to login with the same account on different devices and it worked, which means that the accounts are stored on the server database, but, as stated by the developers in the ITD, the appointments loaded on the application are the ones saved on the client, which means that logging from the same phone with different accounts loads the same appointments.
We also tried to insert the wrong information in the login form and the application notified us with an error message, as expected.

4.2 Client Side

4.2.1 Password Recovery



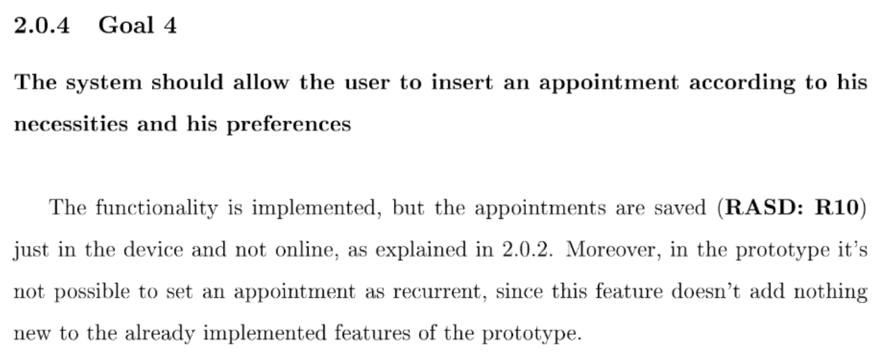
2.0.3 Goal 3

The system should give to the signed user the possibility to recover his password

The functionality has not been implemented.

This goal is not implemented, as stated in the ITD.

4.2.2 Appointment Creation



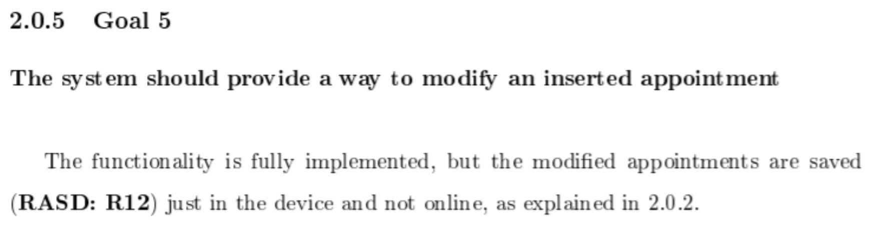
2.0.4 Goal 4

The system should allow the user to insert an appointment according to his necessities and his preferences

The functionality is implemented, but the appointments are saved (**RASD: R10**) just in the device and not online, as explained in 2.0.2. Moreover, in the prototype it's not possible to set an appointment as recurrent, since this feature doesn't add nothing new to the already implemented features of the prototype.

The appointments are successfully created if you create an event which starts at a feasible time. The location search works perfectly and the form is well structured. If you try to insert an appointment which starts before the local time the application will crash instead of sending an error notification. It also sometimes crashes if you forget to input how many people are involved in the event.

4.2.3 Editing An Appointment



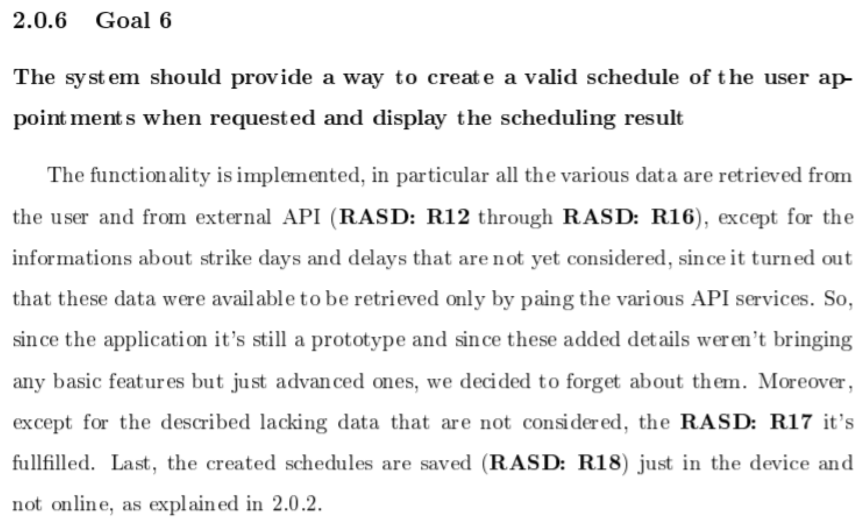
2.0.5 Goal 5

The system should provide a way to modify an inserted appointment

The functionality is fully implemented, but the modified appointments are saved (**RASD: R12**) just in the device and not online, as explained in 2.0.2.

We tried to edit multiple appointments and it always worked without problems.

4.2.4 Schedule Creation

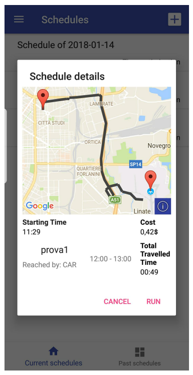


2.0.6 Goal 6

The system should provide a way to create a valid schedule of the user appointments when requested and display the scheduling result

The functionality is implemented, in particular all the various data are retrieved from the user and from external API (**RASD: R12** through **RASD: R16**), except for the informations about strike days and delays that are not yet considered, since it turned out that these data were available to be retrieved only by paing the various API services. So, since the application it's still a prototype and since these added details weren't bringing any basic features but just advanced ones, we decided to forget about them. Moreover, except for the described lacking data that are not considered, the **RASD: R17** it's fullfilled. Last, the created schedules are saved (**RASD: R18**) just in the device and not online, as explained in 2.0.2.

The schedule creation works but has some problems. For example, if a schedule is not feasible the application simply notifies the impossibility of creating a schedule, without explaining which are the problems.
It also gives some results which are not very reasonable, for example we tried to set an appointment at the Linate airport in Milan starting from Piola, and the application said that it would take us 49 minutes by car to get there, while in reality it only takes about 15 minutes and and the expected time of arrival combined with the travel time says that you will arrive late for the appointment.



Starting Time
11:29

Cost
0,42$

prova1
Reached by: CAR
12:00 - 13:00

Total Travelled Time
00:49

4.2.5 Multiple Schedules

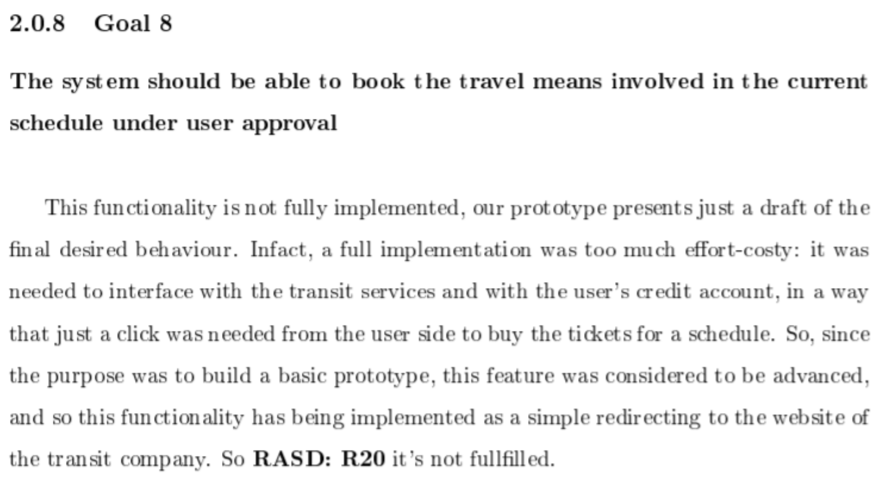
2.0.7 Goal 7

The system should let the user create valid multiple schedules and decide which one is chosen for the current day

This functionality is partially implemented. Taxis are not implemented, because there is no available API for taxis.

The creation of the schedule sometimes doesnt work properly.
It can happen that the application sends an infeasible schedule notification without a clear reason (e.g.: wake up time: 7 a.m.; appointment time: 9 p.m. (distance of few km); optimization parameter: cost). Rarely, the application freezes on the schedule creation page.
When we were able to successfully create two schedules for the same day, this goal was reached correctly.

4.2.6 Booking



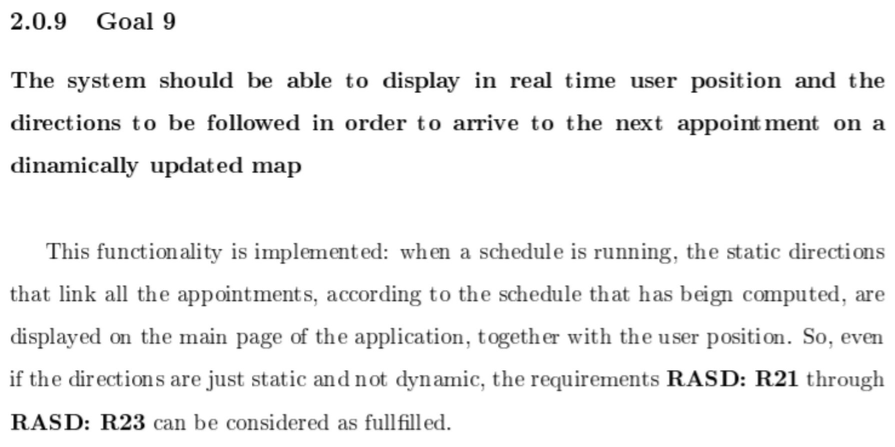
2.0.8 Goal 8

The system should be able to book the travel means involved in the current schedule under user approval

This functionality is not fully implemented, our prototype presents just a draft of the final desired behaviour. Infact, a full implementation was too much effort-costy: it was needed to interface with the transit services and with the user's credit account, in a way that just a click was needed from the user side to buy the tickets for a schedule. So, since the purpose was to build a basic prototype, this feature was considered to be advanced, and so this functionality has being implemented as a simple redirecting to the website of the transit company. So **RASD: R20** it's not fullfilled.

We were not able to access this functionality. We tried to create multiple schedules in which we had to take public transports, but, after running the schedules, there wasnt any redirection to the website of the transit company. We do not know if the application has this feature working, but we were not able to access it.

4.2.7 Real Time User Position



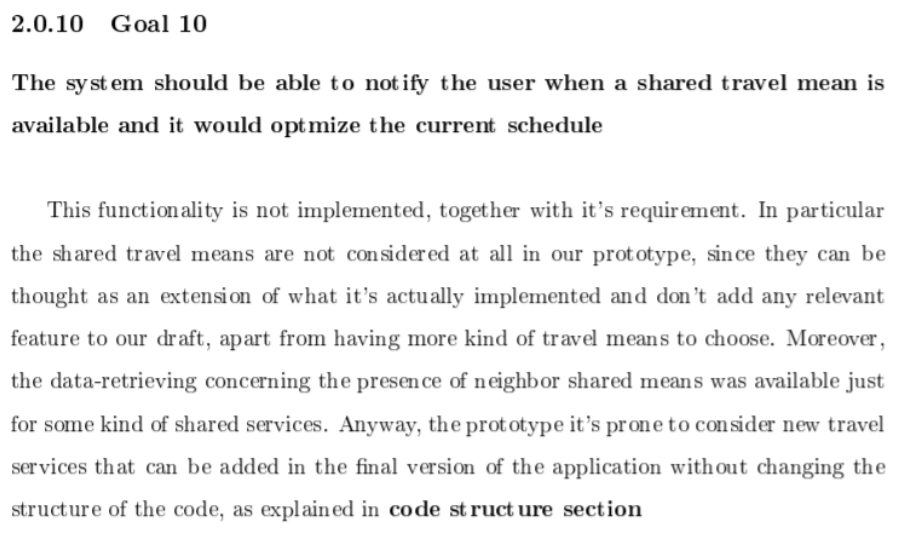
2.0.9 Goal 9

The system should be able to display in real time user position and the directions to be followed in order to arrive to the next appointment on a dinamically updated map

This functionality is implemented: when a schedule is running, the static directions that link all the appointments, according to the schedule that has beign computed, are displayed on the main page of the application, together with the user position. So, even if the directions are just static and not dynamic, the requirements **RASD: R21** through **RASD: R23** can be considered as fullfilled.

This feature works as described above. Once the schedule is created, on the homepage of the application all the directions from the starting point to the location of the appointment are shown both on the map and as text on the bottom of the screen. The directions are well detailed and easy to follow.

4.2.8 Shared Travel Means



2.0.10 Goal 10

The system should be able to notify the user when a shared travel mean is available and it would optmize the current schedule

This functionality is not implemented, together with it's requirement. In particular the shared travel means are not considered at all in our prototype, since they can be thought as an extension of what it's actually implemented and don't add any relevant feature to our draft, apart from having more kind of travel means to choose. Moreover, the data-retrieving concerning the presence of neighbor shared means was available just for some kind of shared services. Anyway, the prototype it's prone to consider new travel services that can be added in the final version of the application without changing the structure of the code, as explained in **code structure section**

This feature is not implemented.

5 Final Conclusion

The functionalities were well thought and accurately designed. However we found some problems while running the application. For example the application is not totally stable because some exceptions are not handled and, when you try to do some forbidden actions the system, instead of sending an error notification, will simply crash and you will lose all the temporary data.
Some of the options during the creation of events and the schedule were not immediately comprehensible, for example while setting constraints for a schedule it asks you to input a max distance without specifying the measure unit.
We believe that these problems at the moment are not too concerning because it is simply a prototype, but should be fixed before the official launch of the application since they could generate frustration in the users and ruin their experience with this program.

6 Effort Spent

- Teamwork: 6 hours.