Questo è il titolo della mia tesi

Pinco Pallino

15 ottobre 2017

Indice

1 Introduction

1.1 Purpose

1.1.1 Description of the RASD

RASD stands for Requirements Analysis and Specification Document. The main goal of this document is to describe the system by clearly specifying functional and non-functional requirements in structured though informal form in order to provide a guideline. RASD takes into account the limits and the constraints of the problem and its possible solutions. It provides feedback to the costumer, it serves as an input to the design specification, as a product validation check and as a contractual basis between the costumer and the developer. Therefore, RASD should be a model that is directed towards ensuring that the final system conforms to client needs. The document is addressed to all customers and users, system and requirement analysists, developers and programmers who participate to the implementation of the requirements, testers and project managers.

1.1.2 Purpose of the application

The application should be useful for busy people that have to travel from a place to another one because of their engagements. It helps people by organizing their own calendar: it finds the best solution to reach a certain place in a specific time basing on user's preferences. Users are able to see

their meetings, their journey between them, their breaks and in every time, they can modify their day calendar by modifying an activity or deleting it.

1.2 Scope

1.2.1 Description of the given problem

We want to create a calendar-based application which name is Travlendar+. It is a software that provides a support to everyone that has scheduling meetings at various locations. It helps the user in finding the best option to reach the destination in the optimal conditions and at a fixed time. Once the user has registered and has inserted time and place of his/her meetings, the system automatically computes and accounts for travel time between appointments, to make sure that the client will not be late. Moreover, the user will be warned if the location is not reachable at the allotted time. Other services are provided, such as the possibility to identify the best mobility option basing on user's preferences (preference or avoidance for a determined mean) and on external conditions (weather, strikes?), the opportunity to buy public transportation tickets and to locate the nearest bike of a bike sharing system or the nearest car of a car sharing system. It is also possible to select combinations of transportations means that minimize carbon footprint. Thanks to this application the users can also organize in a customizable way their break time between and appointment and one other, for example by managing a flexible lunch.

1.2.2 Actual system

Even if there already exist applications that allow users to find the best travel solution, this is a new kind of application for the innovative idea of managing the time. Therefore, we assume that the whole system will be created by new. However, the application will exploit applications or websites to allow the user to buy transportation's tickets and to use sharing services. Moreover, it will profit by these websites to have real-time news on weather, strikes...

1.2.3 Goals

[G_1] Users should be able to sign up into application;

[G_2] Users should be able to log into application;

[G_3] Users should be able to see their personal calendar;

[G_4] Users should be able to see the map on which meetings are showed;

[G_5] Users should be able to see their daily planner;

[G_6] Users should be able to create meetings with location and scheduled time;

[G_7] Users should be able to modify existing activities;

[G_8] Users should be able to delete activities;

[G_9] Users should be able to globally activate or deactivate each travel mean;

[G_{10}] Users should be able to provide constraints on different travel means;

[G_{11}] Users should be able to select combinations of transportation means that minimize carbon footprint;

[G_{12}] Users should be able to buy public transportation tickets or day/week/season pass basing on their needs;

[G_{13}] Users should be able to globally specify breaks' time and their minimum duration flexibly;

[G_{14}] User should be able to add new breaks in the schedule with their duration;

[G_{15}] The system should be able to save user's username and password if he/she wants to.

[G_{16}] The system should be able to compute and account for travel time between appointment;

[G_{17}] The system should be able to identify the best mobility option based on external variables like strikes, weather etc.;

[G_{18}] The system should be able to locate the nearest vehicle of a vehicle sharing system exploiting its application;

[G_{19}] The system should be able to create a warning when the user cannot reach a location at a certain time;

[G_{20}] The system should be able to support different travel means; (Va aggiunto anche il fatto che si possono comprare biglietti?)

1.2.4 Actors

There are two types of actors:

- Visitors: people that download the application and that are not registered in the system but they have free access to the login page, the sign-up page.
- Registered users: they can see all the pages available to visitors and after successful login they can take advantage of all the services of the application.

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

- Visitor: a person that is not registered yet, but has the access to the application's information
- Registered user: a person that is logged in the system and can create meetings.
- Activity: an event that happens in the real world and that could be a meeting or a break.
- Meeting: an activity among the registered user and other people. It can be created, modified and deleted by the meeting's creator.
- Break: an activity that a registered user can insert in order to manage it in a customizable way
- Trip: it indicates the route and the travel means chosen, based on user's preferences.
- Creation screen: the screen of the application in which the registered user create a meeting or a break and enters its related details.

1.3.2 Acronyms

- RASD: Requirements Analysis and Specification Document

1.3.3 Abbreviations

- Gi: i-goal
- Ri: i-requirement
- Di: i-domain assumption

1.4 Revision history

1.5 Reference documents

- Specification Document: Mandatory Project Assignment.pdf;
- Requirement Engineering Part III.pdf;

1.6 Document structure

This document is structured as follows: Section 1: Introduction In this section it is described the purpose of this document, the main goals of the given problem and a brief description of its main characteristics. Section 2: Overall Description It provides further information about the application with a summary of major functions and it states all the assumptions and the constraints Section 3: Specific Requirements In this part we include more details about the requirements Section 4: Formal Analysis using Alloy This section provides the Alloy model and all the proves that it supplies. Section 5: Effort spent Here are reported the information about the hours of work spent by each member of the group by doing this project Section 6: References

2 Overall description

Riprendiamo quanto visto nella sezione 3.5.5, bla bla...

2.1 Product perspective

Our application requires a smartphone (iOs/Android) to be executed and it requires an internet connection in order to benefit from the application's main services. It also requires an active GPS connection to identify the user's position. Travlendar+ is similar to other pre-existing applications that compute the best route with the best means to reach a specific location (e.g. Moovit), and like them it is supported with updated time tables of all the travel means and with an estimation of travel time.

2.2 Product functions

2.3 User characteristics

Travlendar+ has no specific user target: everyone, that is able to use a device with an internet connection, can take advantage of this new type of calendar application. However, the user, that we expect to benefit most, is a busy person who wants to organize his/her daily activities in the best way, finding the fastest and the most comfortable way to travel between two activities and to reach the destination on time. To own a personal calendar, the user must have a device with an internet connection and register with all necessary data or sign in with Facebook or Twitter.

2.4 Assumptions, dependencies and constraints

- To become a registered user, a visitor must either insert name, surname, username, password, email, address, date of birth, telephone number or sign up with Facebook or Twitter
- A visitor can see only the log in and registration page
- Password must be at least 8-characters long for security reason
- To log in a registered user must provide the username and the password associated to him/her
- A registered user can create an unlimited number of meetings
- The calendar for each user is unique

- When meetings overlap, or they can not be reached in the allotted time, a warning is created
- In order to buy public transportation's tickets or to use sharing systems the user is redirected to websites/apps that provide those services
- Warnings are visualized inside the meeting's information screen
- When a warning is generated, a notification is sent to the user's device
- In the section "My Account" a user can modify personal data and can express global preferences (e.g. activate/deactivate each travel means, specify the minimum lunch duration)
- In the creation screen a user can specify the type of activity to be added (break or meeting)
- To create a meeting, users must insert name, location, date, starting/ending hours. Optionally they can also insert a brief description
- To create a break, users must insert the type of break

3 Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

Our application has been designed to be used through a smartphone or a tablet. The following mock-ups show the screens of the main features offered by the smartphone version.

3.1.2 Hardware interfaces

The application does not require any hardware interface.

3.1.3 Software interfaces

Travlendar+ does not provide for itself the possibility to directly buy the public transportation's tickets and the possibility to use sharing systems, but redirects the user to the corresponding website or, if it's already installed in the device, to the corresponding app.

3.1.4 Communication interfaces

The application needs an internet connection on the device in order to receive real-time information about traffic, strikes and weather. Furthermore, it's required also for the communication with third party services that are provided in Travlendar+.

3.2 Funcional requirements

3.3 Performance requirements

In order to guarantee the performance of our application, we have to specify:

- Response Time
- Workload
- Scalability
- Platform

To be reactive and able to answer to a large number of requests, we assume that the response time is close to 0 (from Jakon Nielsen book on Usability 0,1 seconds is about the limit to have the user feel that the system is reacting instantaneously), so it depends mostly on the internet connection of the platform used. We assume that there will be no problem with scalability even if it is a new software and it could suffer an unexpected growth in popularity and from an increase in workload.

3.4 Design constraints

The application will be developed with Java 8, so it will inherit all language's constraints.

3.4.1 Standards compliance

This RASD is written trying to be conformed to the IEEE Standard (ISO/IEC/IEEE 29148 dated 2011). We would like that our application's life cycle process is conformed to the IEEE Standard, in particular to ISO/IEC 12207 dated 2008.

3.4.2 Hardware limitation

Since it is a mobile application, Travlendar+ requires a smartphone or a tablet with internet connection and with the GPS to find the location of the user.

3.4.3 Any other constraint

3.5 Software system attributes

3.5.1 Reliability

The system must be active 24/7 to guarantee all the services in every occasion.

3.5.2 Availability

3.5.3 Security

External Interface Side Travlendar+ application is equipped with a login authentication to protect the information of users. Some precautions about the password are necessary to limit vulnerability and to guarantee a complete security of the user?s payment data. First, in order to avoid brute force attacks, it is necessary to develop a system that requires a strong password, for example containing at least 8 characters comprehensive of numbers and capital letters. This expedient is not sufficient against key logger attacks, against which is needed multi-factor authentication. A possible solution is a two-factor authentication, for example with a code sent by email or sms to the user.

Application side The application layer is the hardest to defend. To prevent injection attacks it is useful to employ comprehensive data sanitization or to use a web application firewall. Moreover, to avoid sensitive data exposure, such as the credit cards or authentication credentials, it is needed the implementation security measures like the encryption of the data or the definition of accessibility, secure authentication gateway (for example the use of the advanced standard security technology like SST or TSL) and a backup plan.

Server side An idea to implement the server side architecture is to strongly separate the data from application and to use firewalls to separate one zone to each others.

3.5.4 Maintenability

The application does not provide any specific API, but the whole application code will be documented to well inform future developers of how application works and how it has been developed. -¿ NON AVEVO LA PIU PALLIDA IDEA, HO PRESO LA FRASE DA UN ESEMPIO DI UN ALTRO ANNO!!!!!

3.5.5 Portability

The application could be used on every smartphone provided with iOs or Android.

4 Formal analysis using alloy

5 Effort spent

6 References

[1] `http://www.1202performance.com/` (To understand the part of performance requirements)

[2] `http://www.iso.org/standard/` (For the section "Standard Compliance")