

QUESTO E' IL TITOLO DELLA MIA TESI

Pinco Pallino

20 ottobre 2017

# Indice

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.1.1	Description of the RASD . . . . .	1
1.1.2	Purpose of the application . . . . .	1
1.2	Scope . . . . .	1
1.2.1	Description of the given problem . . . . .	1
1.2.2	Actual system . . . . .	2
1.2.3	Goals . . . . .	2
1.2.4	Actors . . . . .	3
1.2.5	Class diagram . . . . .	4
1.2.6	Statechart diagram . . . . .	4
1.3	Definitions, acronyms, abbreviations . . . . .	4
1.3.1	Definitions . . . . .	4
1.3.2	Acronyms . . . . .	4
1.3.3	Abbreviations . . . . .	5
1.4	Revision history . . . . .	5
1.5	Reference documents . . . . .	5
1.6	Document structure . . . . .	5
<b>2</b>	<b>Overall description</b>	<b>6</b>
2.1	Product perspective . . . . .	6
2.2	Product functions . . . . .	6
2.3	User characteristics . . . . .	7
2.4	Assumptions, dependencies and constraints . . . . .	8
2.5	The World and The Machine . . . . .	8
<b>3</b>	<b>Specific requirements</b>	<b>9</b>
3.1	External interface requirements . . . . .	9
3.1.1	User interfaces . . . . .	9
3.1.2	Hardware interfaces . . . . .	9
3.1.3	Software interfaces . . . . .	9
3.1.4	Communication interfaces . . . . .	9
3.2	Functional requirements . . . . .	9
3.2.1	Scenarios . . . . .	11
3.2.2	Use case diagrams . . . . .	15
3.3	Performance requirements . . . . .	18

3.4	Design constraints . . . . .	18
3.4.1	Standards compliance . . . . .	18
3.4.2	Hardware limitation . . . . .	19
3.4.3	Any other constraint . . . . .	19
3.5	Software system attributes . . . . .	19
3.5.1	Reliability . . . . .	19
3.5.2	Availability . . . . .	19
3.5.3	Security . . . . .	19
3.5.4	Maintenability . . . . .	20
3.5.5	Portability . . . . .	20
<b>4</b>	<b>Formal analysis using alloy</b>	<b>21</b>
<b>5</b>	<b>Effort spent</b>	<b>22</b>
<b>6</b>	<b>References</b>	<b>23</b>

# **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 customer, it serves as an input to the design specification, as a product validation check and as a contractual basis between the customer 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 analysts, 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 and to specify the maximum walking distance. 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<sub>1</sub>] Users should be able to sign up into application;
- [G<sub>2</sub>] Users should be able to log into application;
- [G<sub>3</sub>] Users should be able to see their personal calendar;
- [G<sub>4</sub>] Users should be able to see the map on which meetings are showed;
- [G<sub>5</sub>] Users should be able to see their daily planner;
- [G<sub>6</sub>] Users should be able to create meetings with location and scheduled time;
- [G<sub>7</sub>] Users should be able to modify existing activities;
- [G<sub>8</sub>] Users should be able to delete activities;
- [G<sub>9</sub>] Users should be able to globally activate or deactivate each travel mean;

- [G<sub>10</sub>] Users should be able to provide constraints on different travel means;
- [G<sub>11</sub>] Users should be able to select combinations of transportation means that minimize carbon footprint;
- [G<sub>12</sub>] Users should be able to buy public transportation tickets or day/week/season pass basing on their needs;
- [G<sub>13</sub>] Users should be able to globally specify breaks' time and their minimum duration flexibly;
- [G<sub>14</sub>] User should be able to add new breaks in the schedule with their duration;
- [G<sub>15</sub>] User should be able to select the maximum walking distance;
- [G<sub>16</sub>] The system should be able to save user's username and password if he/she wants to;
- [G<sub>17</sub>] The system should be able to compute and account for travel time between appointment;
- [G<sub>18</sub>] The system should be able to identify the best mobility option based on external variables like strikes, weather etc.;
- [G<sub>19</sub>] The system should be able to locate the nearest vehicle of a vehicle sharing system exploiting its application;
- [G<sub>20</sub>] The system should be able to create a warning when the user cannot reach a location at a certain time;
- [G<sub>21</sub>] The system should be able to support different travel means;
- [G<sub>22</sub>] The system should be able to create a warning when a strike is announced in a day with activities.

#### **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.

#### **1.2.5 Class diagram**

#### **1.2.6 Statechart diagram**

### **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.
- Blocked travel means: it is a travel means that the user has selected as unwanted.

#### **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

## 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

We already identified the goals that characterized functionalities of Travlendar+. Instead, in this section, we offer a specific list of the main features of the application in order to explain better its utility.

**Offline mode** Travlendar+ could be used in offline mode simply as a basic calendar. Users' can visualize in the main screen all their activities for the day, by clicking them they can see all details about the location and the time. In this mode it isn't possible to see the map with the computed trips. Users' can also insert new activities by pressing the plus button, whereas if they click on the calendar icon, they are able to visualize the whole calendar.

**Map** The application offers the possibility to visualize the map of the interested area. If there is any activity saved in the calendar for the current day, an indicator is shown in its location. On the map a blue line highlights the trip between the different meetings of the day. By pressing an activity appears also the chosen path to reach it from the actual location. By pressing "GO" it can be visualized the whole path of the day.

**Personalization** Travlendar+ offers a lot of options that can be customized, for example the user can:

- Specify a minimum duration of the lunch time;
- Specify everyday lunch time;
- Activate or deactivate travel means;
- Insert the maximum walking distance;
- Select combinations of transportation means that minimize carbon footprint;
- Deactivate public transportation in a certain time zone.

**The best route's computation** Once a meeting is created Travlendar+ suggests the best route between the previous, the current and the next activity that satisfies the constraints of the user. If the user is not satisfied with the proposed trip, he/she can choose another one between those suggested by the app.

**Real time informations** Every time the system is updated with the information about the traffic, the weather, strikes etc. The user can either receive the respective notification or visualize all the warnings in the main screen. All this warnings influence the choice of the travel means for the trip.

**Possible future implementations** TO DO

## 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

## 2.5 The World and The Machine

## 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 Functional requirements

[G<sub>1</sub>] Users should be able to sign up into application;

[R<sub>1</sub>] The system should allow the visitor to begin the registration process by asking him/her all data that are necessary: name, surname, date of birth, email, username, password;

[R<sub>2</sub>] The system should be able to extract data from Facebook or Twitter if the visitor sign in with that application;

[R<sub>3</sub>] The system must not allow a registered user to perform registration process;

- [R<sub>4</sub>] The system must allow visitors only to see the login page and the registration form;
- [D<sub>1</sub>] The username must be unique; TODO VA IN R O IN D?? VEDI ESEMPI
- [D<sub>2</sub>] The password must be at least of 8 characters.
- [D<sub>3</sub>] The email address must be correct.
- [G<sub>2</sub>] Users should be able to log into application;
  - [R<sub>1</sub>] The system should allow only registered users to login;
  - [R<sub>2</sub>] Username and password inserted must be correct to perform the login;
  - [R<sub>3</sub>] Visitors cannot access to the calendar and to the map without login;
- [G<sub>3</sub>] Users should be able to see their personal calendar;
  - [R<sub>1</sub>] The system must allow users to see their calendar with all their activities
- [G<sub>4</sub>] Users should be able to see the map on which meetings are showed;
- [G<sub>5</sub>] Users should be able to see their daily planner;
- [G<sub>6</sub>] Users should be able to create meetings with location and scheduled time;
- [G<sub>7</sub>] Users should be able to modify existing activities;
- [G<sub>8</sub>] Users should be able to delete activities;
- [G<sub>9</sub>] Users should be able to globally activate or deactivate each travel mean;
- [G<sub>10</sub>] Users should be able to provide constraints on different travel means;
- [G<sub>11</sub>] Users should be able to select combinations of transportation means that minimize carbon footprint;
- [G<sub>12</sub>] Users should be able to buy public transportation tickets or day/week/season pass basing on their needs;

- [G<sub>13</sub>] Users should be able to globally specify breaks' time and their minimum duration flexibly;
- [G<sub>14</sub>] User should be able to add new breaks in the schedule with their duration;
- [G<sub>15</sub>] User should be able to select the maximum walking distance;
- [G<sub>16</sub>] The system should be able to save user's username and password if he/she wants to;
- [G<sub>17</sub>] The system should be able to compute and account for travel time between appointment;
- [G<sub>18</sub>] The system should be able to identify the best mobility option based on external variables like strikes, weather etc.;
- [G<sub>19</sub>] The system should be able to locate the nearest vehicle of a vehicle sharing system exploiting its application;
- [G<sub>20</sub>] The system should be able to create a warning when the user cannot reach a location at a certain time;
- [G<sub>21</sub>] The system should be able to support different travel means;
- [G<sub>22</sub>] The system should be able to create a warning when a strike is announced in a day with activities.

### 3.2.1 Scenarios

**Scenario 1** Dybala is a very busy football player that lives in Turin. Tomorrow morning he is going to have an important meeting with one of the main sponsor of the Juventus team, the Adidas, in Milan, and in the evening he has the Champions League's final. It is the most important match of the year, so he needs to organize his day at the best! He decides to use Travlendar+. He inserts the two activities in the list of meetings: "Meeting with Adidas", on 20th of May, from 11 am to 1 pm, in Corso Buenos Aires 40 (Milan) and "Match prep for the Final", on 20th of May, from 3 pm to 7 pm, at the Allianz Stadium (Turin). Since he wants to preserve energies for the Final, he selects "Walking" and "Bike" as blocked travel means in the Preferences' area. The application suggests him a trip with the train

from Turin to Milan in the early morning because the estimated time for the route by car or taxi is reported to be slower due to the traffic. The application redirects him to the Trenitalia's website to buy the ticket. Then, for coming back, Travlendar+ suggests him to take a taxi because public transport timetables for the given time are not convenient.

**Scenario 2** Giuseppe is a student at Politecnico of Milan. In view of the exams session, he has few time but he does not want to give up a good lunch with his friends. Therefore, he specifies in the "My Account" section of Travlendar+ that everyday he needs at least 30 minutes free to have a comfortable lunch between 11.30 am and 2.30 pm. He inserts the schedule of all his lessons. After having inserted the Formal Languages and Compilers's lesson of Wednesday, a warning appears to report an overlap with the fixed time reserved for the lunch. He can ignore the warning and decide what he prefers to do.

**Scenario 3** Carlo Cracco is a famous chef that has two restaurants in Milan. He would like to open a new restaurant in Verona. So he has to go to Verona to search for a place that can be restored as he likes and to meet architects and designers to decide all details of the new restaurant. He would like to have a coffee break between the afternoon appointments to restore himself. Therefore he uses Travlendar+ to organize the full day. He inserts all the meetings and he also inserts the coffee break: pressing the plus symbol in the main screen, Carlo selects the type "Break", choose the "Coffee break", he inserts the minimum duration for the break and he is redirected to the website "TripAdvisor". In this way the chef can see what coffee is opened near the two appointments that he has before and after the break and he can choose the best one.

**Scenario 4** Lorenzo Fragola is a young boy with a strong passion for singing, so he decides to go to an audition for XFactor, the most famous music talent show in Italy. The audition will take place on the 30th of July in Milan, at Arena Civica. Lorenzo lives in Catania and decides to use Travlendar+ to program the trip for the audition. He inserts the meeting in the system, with the date, the location and the time, and visualizes the trip. The two available means suggested by the application are the train, which takes 13 hours, and the car, which takes 14 hours. Since both take too long, he

decides to organize himself the trip with the plane. Once he will be in Milan, he will use the application to go to the Arena Civica. The trip suggested is by autobus from the airport to the railway station, and than by metro to the Arena. Unfortunately, one week before the audition, a warning appears on his phone: on the day of the audition there will be a strike of ATM! At first time Lorenzo is a little bit worried, but immediately sees that Travlendar+ has suggested him an alternative trip by car (taxi or car sharing) from the railway station to the Arena. He will choose at that day the best option. When the moment arrives, he selects car sharing as favourite means, and the application redirects him on the playstore to download the Enjoy's application (the car sharing service). Through Enjoy's application he can see that there is a car near the station and he can book it.

**Scenario 5** Travlendar+ is become a very well known application. A lot of people start to use it and find Travlendar+ the most useful application of 2017. Also the USA president has become curious because of the fame of this new type of smart calendar. He decides to try it: he has an Iphone X and he goes to the AppStore and install it. Once it is installed, Trump opens it and he finds out that he has to sign in. He clicked the "Sign in" button and a new screen appears. He has to insert a lot of personal informations: name Donald, surname Trump, date of birth 14/06/1946, address 1600 Pennsylvania Ave NW, Washington, username Tump, password \*\*\*\*\* and at the end the telephone number +1 202-456-1111. Once he finishes to compile, he clicks "Sign in" and the screen for the registered users appears: there is the screen with the schedule of the day that is obviously empty for new users.

**Scenario 6** Leonardo DiCaprio is a famous actor that has always been active in fighting climate change. To raise awareness among the people about the environmental problems, he has the idea to live stream one of his ordinary day to show people the importance of using means of transport that minimize carbon footprint. Leonardo decides to use Travlendar+ in order to arrange his trips in the day. Once the application is installed and he creates his personal account, Leonardo inserts three activities (all in New York) to be performed during the day. The application now suggests the ideal route that minimize travel durations, but in order to pursue his goal Leonardo enter the "My Account" section of the app and then he enables, in the preferences' area, the option "Minimize footprint". The route suggested



by the application seems to be more eco-friendly, for example Travlendar+ proposes to him to rent a bike or to use a service of electric car sharing.

**Scenario 7** Claire is going to have a marry of her best friend on Sunday morning. Since after the ceremony there will be a big party with food and drink, she decides not to drive. To discover the best option to reach the church, she uses Travlendar+. The application suggests her to take the metro and then to walk for ten minutes from the underground station to to the destination. It's all perfect for her, but unfortunately she wakes up on Sunday morning and it is raining. She can not affords to walk under the rain because she has just gone to the hairdresser! She immediately logs in to the application to check an alternative trip. Travlendar+, that has an updated weather forecast, has already changed the trip: it suggests her to take the tram once exited from the underground station instead of walking. She notices looking at the tram schedule in the activity's details that she will not take much more to go there with this alternative, because trams of line 34 stops every 13 minutes. She will reach the destination in time.

**Scenario 8** Guglielmo is a Computer science engineer. He works for a big company: he earns a lot of money. Because he often had to move, he used a lot his car. Guglielmo was very reckless, in fact he went on taking fines, even very expensive. He didn't care about it. But once the police gave him another big fine and withdrew his driving licence and confiscate his car. Since he lives in Milan and he hates public transports, he decides to try the application Travlendar+. He enrolles in the app and in the section "My Account" he deactivates every public transportations and sharing cars. He has to go to work: when the trip is suggested, he presses on bike sharing's option because he really love the speed and he has a lot of time to reach the location. Guglielmo is redirected to the Ofo's application and from that application he can see where is the nearest bike. Finally he can go fast without taking any fine!

### 3.2.2 Use case diagrams

#### Registration

Actor	Visitor
Goal	[G <sub>1</sub> ]
Input condition	NULL
Event flow	<p>The registration's process is:</p> <ol style="list-style-type: none"><li>1. The visitor tap on "sign up" on the home page to start the registration process.</li><li>2. The visitor fills in at least the mandatory fields, that are: name, surname, username, password, email, address, date of birth. The non-mandatory field is the telephone number.</li><li>3. The visitor tap on "ok".</li><li>4. The application saves the data and redirect him to the home page where he can proceed to the login.</li></ol>
Output condition	Registration process is completed. The visitor become a registered user.
Exception	<p>The possible exceptions are:</p> <ol style="list-style-type: none"><li>1. The visitor is already a registered user.</li><li>2. The visitor fills one or more fields with invalid information.</li><li>3. The visitor chooses an already existent username.</li><li>4. The visitor inserts an email that is already associated to another user.</li></ol> <p>In case of exception the invalid fields are coloured with red. The visitor can't press "ok" until all fields are correctly filled.</p>

## Login

Actor	Visitor, registered user
Goal	[G <sub>2</sub> ]
Input condition	The user is on the home page.
Event flow	<p>The login's process is:</p> <ol style="list-style-type: none"><li>1. The registered user fills the fields "username" and "password" with her/his username and password. Alternatively, a simple visitor can tap on "Login with Facebook" or "Login with twitter" in order to login directly without having completed the registration process.</li><li>2. If the access is not with Facebook or Twitter, after having inserted the username and password the registered user must tap on "Login".</li></ol>
Output condition	Travlendar+ verifies user's credentials (either they have been directly inserted by the user or obtained through Facebook or Twitter) and if they are correct shows the main screen with the map with the activities.
Exception	<p>The possible exceptions are:</p> <ol style="list-style-type: none"><li>1. The user inserts an incorrect username or password.</li><li>2. The user's data obtained by Facebook or Twitter are invalid.</li></ol> <p>In both the cases the user is notified with an error message and invited to try again.</p>

## New activity

Actor	Registered user
Goal	[G <sub>8</sub> ], [G <sub>14</sub> ]
Input condition	The registered user is already logged into Travlendar+.
Event flow	<p>The process for the creation of a new activity is:</p> <ol style="list-style-type: none"><li>1. The registered user tap on the plus icon ("new activity button").</li><li>2. Travlendar+ shows a new page with form, and the creation process starts. The user is required to specify the type of activity (meeting or break) and to insert all the attributes about the activity, such as name, date, time, location and favourite means. If the activity is of type "break", the registered user must also specify a minimum duration of the activity.</li><li>3. The registered user tap on "ok".</li></ol>
Output condition	Travlendar+ redirects the registered user on the main screen. On the map is appeared the new activity on its location.
Exception	<p>The possible exceptions are:</p> <ol style="list-style-type: none"><li>1. The user does not want to complete the creation process. In such case he/she must tap on "back".</li><li>2. The user does not fill a mandatory field or fills it with invalid data. In such cases the invalid fields are coloured with red. The user must insert the correct data in order to complete the creation of the activity.</li></ol>

### 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 Jakob 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

In order to be compatible both with Android and iOS, the application will be developed with in the following programming languages:

- Swift 4 (for the iOS version)
- Java 8 (for the Android version)

We have choose to use the last version of the programming languages to increase robustness and stability of the application, despite the fact that a lot of devices in the market will not be compatible because they're not up to date with the latest OS version.

#### 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 Maintainability**

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.

#### **3.5.5 Portability**

In order to reach the highest number of devices on the market, the application could be used on every smartphone or tablet 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”)