TRAVLENDAR+

**RASD**

*Requirements Analysis and Specification Document*

Version 1.0

*Greco Sonia*

*Guzzo Francesco*

Release Date: xx.10.2017

***TABLE OF CONTENTS***

**[1. INTRODUCTION](#_Toc495419656)** [4](#_Toc495419656)

[*1.1 PURPOSE* 4](#_Toc495419657)

[*1.2 SCOPE* 4](#_Toc495419658)

[*1.3 DEFINITIONS* 4](#_Toc495419659)

[*(1.4 REVISION HISTORY)* 4](#_Toc495419660)

[*1.5 REFERENCE DOCUMENT* 4](#_Toc495419661)

[*1.6 DOCUMENT STRUCTURE* 4](#_Toc495419662)

[**2. OVERALL DESCRIPTION** 5](#_Toc495419663)

[*2.1 PRODUCT PERSPECIVE* 5](#_Toc495419664)

[*2.2 PRODUCT FUNCTIONS* 5](#_Toc495419665)

[*2.3 USER CHARACTERISTICS* 5](#_Toc495419666)

[*2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS* 5](#_Toc495419667)

[**3. SPECIFIC REQUIREMENTS** 7](#_Toc495419668)

[*3.1 EXTERNAL INTERFACE REQUIREMENTS* 7](#_Toc495419669)

[*3.1.1 User interfaces* 7](#_Toc495419670)

[*3.1.2 Hardware interfaces* 7](#_Toc495419671)

[*3.1.3 Software interfaces* 7](#_Toc495419672)

[*3.1.4 Communication interfaces* 7](#_Toc495419673)

[*3.2 FUNCTIONAL REQUIREMENTS* 7](#_Toc495419674)

[*3.2.1 [G1] Registration.* 7](#_Toc495419675)

[*3.2.2 [G2] Login.* 7](#_Toc495419676)

[*3.2.3 [G3] Allow an User to create/delete meetings.* 7](#_Toc495419677)

[*3.2.4 [G4] Allow an User to notify a public transport strike.* 8](#_Toc495419678)

[*3.2.5 [G5] Allow an User to insert personal preferences to modify the calculation of the best path.* 8](#_Toc495419679)

[*3.2.6 [G6] Allow an User to acknowledge the best path to follow to reach the daily meetings.* 8](#_Toc495419680)

[*3.2.7 [G7] Allow the application to notify an User if there are changes on the pre-defined route.* 8](#_Toc495419681)

[*3.3 PERFORMANCE REQUIREMENTS* 9](#_Toc495419682)

[*3.4 DESIGN CONSTRAINTS* 9](#_Toc495419683)

[*3.4.1 Standards compliance* 9](#_Toc495419684)

[*3.4.2 Hardware limitations* 9](#_Toc495419685)

[*3.4.3 Other constraints* 9](#_Toc495419686)

[*3.5 SOFTWARE SYSTEMS ATTRIBUTES* 9](#_Toc495419687)

[*3.5.1 Reliability* 9](#_Toc495419688)

[*3.5.2 Availability* 9](#_Toc495419689)

[*3.5.3 Security* 9](#_Toc495419690)

[*3.5.4 Maintainability* 9](#_Toc495419691)

[*3.5.5 Portability* 9](#_Toc495419692)

[**4. FORMAL ANALYSIS USING ALLOY** 10](#_Toc495419693)

[**5. EFFORT SPENT** 11](#_Toc495419694)

[**6. REFERENCES** 11](#_Toc495419695)

# **1. INTRODUCTION**

## *1.1 PURPOSE*

Travlendar+ is a project which aims to create a calendar interface that automatically computes and handles the appointments of a person in order to make sure that the user is never late for his/her meetings and everyday activities.

Travlendar+ also ensures that the user utilizes the best mobility option available at current time, including walking, taking into account possible disabilities of the user, strikes and weather conditions.

The user will be able to give personal constraints (e.g. It won’t be suggested to use the car if the user doesn’t have a driving license; walking distances should be less than a given threshold.. ) by activating or deactivating each travel means.

The application will:

- compute times between meetings and prevent them to overlap.

- keep a spot during the day if the user wants to have lunch in a given slot of time.

## *1.2 SCOPE*

Travlendar+ gives a useful support to users’ life. Achieving the features explained above, the application helps the user making the best decision every day. It doesn’t just keep track of all the appointments and meetings of a person, but it also makes sure that you can actually reach the different locations in time, it tells the optimal way to reach them according to various aspects (weather, personal disabilities, owned mobility means, time of the day..). When the user creates meetings that are unreachable in the allotted time, a warning is created.

It also allows the user to buy tickets for the chosen mobility mean or localize the nearest bike of a specified sharing system.

## *1.3 DEFINITIONS*

## *(1.4 REVISION HISTORY)*

## *1.5 REFERENCE DOCUMENT*

## *1.6 DOCUMENT STRUCTURE*

# **2. OVERALL DESCRIPTION**

## *2.1 PRODUCT PERSPECIVE*

*here we include further details on the shared phenomena and a domain model (class diagrams and statecharts)*

## *2.2 PRODUCT FUNCTIONS*

The most important requirements of the project are:

- the computed way from the starting point to the goal must be optimal:

• it should be the shortest

• it should choose the best mobility mean, taking into account the weather forecast

• it should consider also other World related phenomena.

- the optimal way suggested, should also consider user’s personal preferences, such as minimizing the carbon footprint, etc.

## *2.3 USER CHARACTERISTICS*

The user of Travlendar+ is a person who needs help to schedule his/her daily appointments and to find the best way to reach them, according to a series of variables.

## *2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS*

We suppose that these *domain properties* hold in the considered World:

• [D1] The Username chosen during the registration must be unique.

• [D2] The login process needs User’s Username or E-mail, and the corresponding password.

• [D3] The GPS of user’s device is always on and always gives the correct position.

• [D4] Time, Date and Address are necessary to define uniquely a meeting.

• [D5] A public transport strike implies the impossibility to use public transport means.

• [D6] The application should allow the user to register with his/her personal data, preferences and owned mobility means.

• [D7] If there’s a meeting in a given day, there must be at least an optimal way to reach the goal.

• [D8] The application refers to Google API in order to faithfully represent delays, arrivals and departures of every public transport mean, possible works on the roads or traffic jams at current time.

• [D5] The application should take into account the weather forecast when suggest the optimal way to reach the goal with a given mobility mean.

• [D6] During the day, there is a fixed slot of time kept for breakfast/lunch time, chosen by the user.

We assume that these *domain assumptions* hold in the considered World:

- there is one and only one calendar corresponding to one user.

- The user entered his/her personal data in the profile area.

- Meetings and appointments have already been defined by the user before the computation of the best path.

- The application recalculate dynamically during the day the optimal solution only to travel between the appointments already defined.

- The application doesn’t recomputed the journey if the user moves out of the track already defined.

We suppose that these *constraints* hold in the considered world:

- Regulatory policies: the system must require to user the permission to get his position and he has to manage sensible data (position, credentials, e-mail) respecting the privacy law. Furthermore, the system must not use notifications to send SPAM respecting the privacy law.

- It is not possible to create meetings covering the all slot of time reserved for lunch break.

- Hardware limitations:

- USER:

\* 3G connection

\* GPS

\* space for app package

- Interfaces to other applications

- interface with Google API to get information about transport means, traffic jams, mobility schedules, etc.

- interface with Weather Forecast

- the user shouldn’t create overlapping meetings or unreachable ones. A meeting can be unreachable if it not possible to go there in a given slot of time (e.g. it is too far or there are no available transport means).

# **3. SPECIFIC REQUIREMENTS**

## *3.1 EXTERNAL INTERFACE REQUIREMENTS*

### *3.1.1 User interfaces*

### *3.1.2 Hardware interfaces*

### *3.1.3 Software interfaces*

### *3.1.4 Communication interfaces*

## *3.2 FUNCTIONAL REQUIREMENTS*

### *3.2.1 [G1] Registration.*

• [R1] A visitor must be able to begin the registration process.

• [R2] The password chosen for the registration must be at least of 7 characters and it must include at least one number and one capital letter.

• [D1] The Username chosen during the registration must be unique.

### *3.2.2 [G2] Login.*

• [R3] The visitor must be already registered to go through the login.

• [R4] A registered User must be able to login to the system using his/her credentials.

• [D2] The logic process needs User’s Username or E-mail, and the corresponding password.

### *3.2.3 [G3] Allow an User to create/delete meetings.*

• [R4] A registered User must be able to login to the system using his/her credentials.

• [R5] The system must ask the User to insert Date, Time and Address of the meeting he/she wants to create.

• [R6] An User must be able to delete a meeting if and only if it is already in the calendar.

• [R7] The system must notify the User with a warning if the meeting he/she wants to create overlaps with another one already in the calendar.

• [D3] The GPS of user’s device is always on and always gives the correct position.

• [D4] Time, Date and Address are necessary to define uniquely a meeting.

• [D10] During the day, there is a fixed slot of time kept for breakfast/lunch time, chosen by the user.

### *3.2.4 [G4] Allow an User to notify a public transport strike.*

• [R4] A registered User must be able to login to the system using his/her credentials.

• [R8] The User must be able to insert a public transport strike in the calendar to influence the calculation of the best path for those days.

• [D5] A public transport strike implies the impossibility to use public transport means.

### *3.2.5 [G5] Allow an User to insert personal preferences to modify the calculation of the best path.*

• [R4] A registered User must be able to login to the system using his/her credentials.

• [R9] The User must be able to insert personal preferences and constraints:

- A maximum amount of time for walking

- Minimize carbon footprint

- Insert owned mobility means

- Choose not to use public transport means after a certain hour of the day.

• [D6] The application should allow the user to register with his/her personal data, preferences and owned mobility means.

### *3.2.6 [G6] Allow an User to acknowledge the best path to follow to reach the daily meetings.*

• [R4] A registered User must be able to login to the system using his/her credentials.

• [R10] There must be at least a meeting during the day.

• [D3] The GPS of user’s device is always on and always gives the correct position.

• [D7] If there’s a meeting in a given day, there must be at least an optimal way to reach the goal.

• [D8] The application refers to Google API in order to faithfully represent delays, arrivals and departures of every public transport mean, possible works on the roads or traffic jams at current time.

• [D9] The application should take into account the weather forecast when suggest the optimal way to reach the goal with a given mobility mean.

### *3.2.7 [G7] Allow the application to notify an User if there are changes on the pre-defined route.*

• [R4] A registered User must be able to login to the system using his/her credentials.

• [R10] There must be at least a meeting during the day.

• [R11] The system must have already computed the best route at least once.

• [D3] The GPS of user’s device is always on and always gives the correct position.

• [D7] If there’s a meeting in a given day, there must be at least an optimal way to reach the goal.

• [D8] The application refers to Google API in order to faithfully represent delays, arrivals and departures of every public transport mean, possible works on the roads or traffic jams at current time.

• [D9] The application should take into account the weather forecast when suggest the optimal way to reach the goal with a given mobility mean.

## *3.3 PERFORMANCE REQUIREMENTS*

## *3.4 DESIGN CONSTRAINTS*

### *3.4.1 Standards compliance*

### *3.4.2 Hardware limitations*

### *3.4.3 Other constraints*

## *3.5 SOFTWARE SYSTEMS ATTRIBUTES*

### *3.5.1 Reliability*

### *3.5.2 Availability*

### *3.5.3 Security*

### *3.5.4 Maintainability*

### *3.5.5 Portability*

# **4. FORMAL ANALYSIS USING ALLOY**

*in this section you will include your Alloy model. We require*

*you to comment on the model by discussing the purpose of the model, what you can prove*

*with it and why what you prove is important given the problem at hand. You are also*

*required to show one or more worlds obtained by running your model.*

# **5. EFFORT SPENT**

# **6. REFERENCES**