

Politecnico di Milano

A.Y. 2017/2018

Software Engneering 2: ***Travlendar+***

**Requirements Analysis and Specification Document**

Matteo Biaselli - Emilio Capo - Mattia Di Fatta

v. 1.0.1

**Table of Contents**

**1. Introduction**

1.1. Document purpose..………………………………………………...………………….3

1.2. Description of the problem………………………………………………………….3

1.3. Actors………………………………………………………………………………………...3

1.4. Goals…………………………………………………………………………………………..4

1.5. Definitions, Acronyms, Abbreviations………………………………………….4

1.5.1 Definitions……………………………………………………………………….4

1.5.2 Acronyms………………………………………………………………………...4

1.5.3 Abbreviations…………………………………………………………………..4

1.6. Reference Documents…………………………………………………………………4

1.7. Document Structure……………………………………………………………………5

1.8. Revision History…………………………………………………………………………5

**2. Overall Description**

2.1. Product Perspective…...……………………………………………………………….6

2.2. User Characteristics……………………………………………………………………6

2.3. Assumptions, Dependencies and Constraints……………………………….6

2.3.1 Assumptions……………………………………………………………………6

**1. Introduction**

* 1. **Document purpose**

This document focuses on the requirements analysis for the project Travlendar+. The application’s purpose is to support users in handling out one of the most difficult nowadays’ challenges: organizaion. No previous versions of this application were developed.

This document is addressed to developers that will work on the project, possible stakeholders and final users.

* 1. **Description of the problem**

The aim of the project is to create an all-in-one system that unites services that are nowadays offered by various different applications(e.g. Calendar, Travel Scheduler). In order to use Travlendar+, final users should be registered and logged in.

Users should be able to schedule their activities directly through the application and, by taking into account travelling times, constraints and preferences expressed by the user, Travlendar+ should:

* Identify the best mobility option;
* Support the user in buying public transport tickets, if necessary;
* Locate the nearest car or bike sharing, if they represent the best solution;
* Warn the user when a place can’t be reached in the available time.

In general, Travlendar+ should make it easier to organize complex schedules, by finding the best compromises between time optimization and the users’ needs and preferences.

* 1. **Actors**
* **Guest**: a person that is not yet registered or logged in. This actor can only see the main page of Travlendar+, where he’s asked to log in or start a registration procedure.
* **User**: a person that is registered and logged in. This type of actor can access to all the features offered by Travlendar+.
* **Third part systems**: external systems that, through the use of APIs, Travlendar+ can use to provide the services it intends to offer.

*(e.g. Google Maps’ APIs can be used to calculate the time needed to move from a place to another by car, bike, public transport system and on foot).*

*(e.g. a car sharing system’s APIs can be used to locate the available cars and propose to the user to use a car sharing service)*

* 1. **Goals**

Travlendar+ features should match with the following goals:

* [G1] Users should be able to use the system properly.
* [G2] Allow users to schedule an activity.
* [G3] Allow the users to manage already existing activities
* [G4] Allow users to set constraints and preferences.
* [G5] Simplify procedures and reduce the number of operations that a user has to do to schedule an new activity.
* [G6] Allow users to be on time at their scheduled appointments
  1. **Definitions, Acronyms, Abbreviations**
     1. **Definitions**
* **User**: actor that is using the application and may want to access all functionalities.
* **Application**: with the term application we are talking about both the desktop version and mobile version of the Travlendar+ system.
* **Scheduling ad activity**: action performed by a user that is adding a new activity to his personal calendar.
  + 1. **Acronyms**
* **RASD:** Requirements Analysis and Specification Document
* **UI:** User Interface
* **API:** [Application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface)
* **UXD**: User Experience Diagram
* **UML**: Unified Modeling Language
  + 1. **Abbreviations**
* **[Gn]:** the n-th goal
* **[Rn]:** the n-th requirement
* **[NFRn]:** the n-th non-functional requirement
* **[An]:** the n-th assumption
  1. **Reference Documents**
* Mandatory project assignments for the A.Y. 2017/2018 available on the beep’s page of the Software Engineering 2 course.
* Projects examples and other documents available on the beep’s page of the Software Engineering 2 course.
  1. **Document Structure**
* **Introduction:** This is the very first part of the document.

In this section iIt’s possible to retrieve general information about the project and its goals and about the system that is going to be described with more details in the next sections. For a mater of simpicity and to avoid any possible ambiguity, Acronyms, Definitions and Abbreviations that will be used in the whole document have been specified here.

* **Overall Description:** This section will contain a more detailed explanation of the product perspective, that can be useful to visualize and contextualize the project we’re working on. Further information about the application’s functions, the user and the requirements will be clarified in the Overall Decription.
* **Specific Requirements:** This section contains more details and examples about the User Interface and specifies which are the Communication Interfaces that must be supported by the end-user devices in order to make the application work properly.

Plus, Functional requirements will be defined with more details and they will be mapped into goals.

UML diagrams will be provided in this section.

* **Formal Analysis using Alloy:** The source code of an Alloy 4.2 specification of the system.
* **Effort Spent:** Information about the number of hours each group member has spent working on the RASD document.
  1. **Revision History**
* **v. 1.0 [05 Oct 2017]**: added the whole “Introduction” section.
* **v. 1.0.1 [07 Oct 2017]**: added part of the “Overall Description” section.

**2.0 Overall Description**

This section includes a summary of the major functions provided by the system, the user characteristics, the constraints and the assumptions over the domain.

**2.1 Product Perspective**

Since the application can be used both on desktop computers and mobile devices and since a user may want to access his calendar from both kind of devices at the same time, data cannot be stored locally. The system we’re going to develop will then consist of an application for the end user and an application for the central server. On the server, users’ data, preferences and schedules will be stored in a Database.

Furthermore, our product needs to be perfectly integrated with some pre-existing systems, such as other car sharing and bike sharing systems, through their exposed APIs.

**2.2 User Characteristics**

As specified in the “Actors” section above, there is no relevant distinction that we have to make between our expected end users, simply because there are no particular kinds of users we are expecting use the application, once it is ready.

Users don’t need any specific knowledge to make a good use Travendar+, and, furthermore, users’ main life occupations are not really relevant because our application can be used to schedule every kind of appointment, from business meetings to a football match with friends.

* 1. **Assumptions, Dependencies and Constraints**
     1. **Assumptions**
* **[A1] Ubiquity:** Users cannot schedule two different activities in two different places at the same time of the same day. This means that the second activity must be scheduled on a different time of the day, or the first one must be deleted.
* **[A2] GPS:** to suggest the best mobility option, Travlendar+ has to acquire the position of the user though GPS, if the user hasn’t inserted a starting point manually. We assume that those data have a maximum error of 10 meters.
* **[A3] Email:** Users’ declared emails are supposed to be currently in use. When a user registers, his email is verified. From that moment, then, we assume that the user will not stop using and regurarly checking that mailbox.
* **[A6] Accidents:** Accidents and delays are frequent but most of the times online systems that calculate shortest mobility options get to know about them rapidly and take them into account for the solutions they propose. We assume that, if a user is following one of the mobility options given from Travlendar+ (see [A7]), unpredictable accidents that may cause delays have a rate of 10%. This means that the rest of the times (90%) users will be on time at their apointments.
* **[A7] Mobility options:** Users will follow the mobility options proposed by Travlendar+.

**5. Effort Spent**

This section will provide detailed information about the nomver of hours spent on this document.

**Matteo Biasielli**, matr. 893590

|  |  |
| --- | --- |
| **Section(s)** | **Number of hours** |
| Introduction | 3 |
| Overall Description | 1,5 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* [R2] Users should be able to log in to Travlendar+.
* [R3] Users should be able to register to Travlendar+.
* [R4] Users should be able to set their own preferences that will be taken into account and will be applied to schedules every time this is possible and reasonable.
* [R5] Users should be able to reserve a certain amount of time for lunch every day.
* [R6] When necessary, users should be supported in buying transports tickets directly on Travlendar+ and/or redirected on the correct external page.
* [R7] Users should be warned when they’re scheduling an activity that is not physically possible due to a lack of time or that overlaps with other activities.
* [R8] Solutions involving car and bike sharing systems must be taken into account and proposed to the user when they represent the optimal solution.