1 - INTRODUCTION

1.1 - PURPOSE

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.

1.2 - 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.

1.2.3 – Goals **(TODO, rifare lista con G1 ecc…)**

Users should be able to:

1. Sign up into application;
2. Log into application;
3. Create meetings with location and scheduled time;
4. Modify existing meetings;
5. Globally activate or deactivate each travel mean;
6. Provide constraints on different travel means;
7. Select combinations of transportation means that minimize carbon footprint;
8. Buy public transportation tickets or day/week/season pass basing on their needs;
9. Specify breaks’ time and their minimum duration flexibly;

The system should be able to:

1. Compute and account for travel time between appointment;
2. Identify the best mobility option based on external variables like strikes, weather etc.;
3. Locate the nearest vehicle of a vehicle sharing system;
4. Create a warning when the user cannot reach a location at a certain time;
5. 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 and can discover the functionality of the application through an information screen.
* 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
* Meeting creator: a registered user that creates meetings.
* Meeting: an appointment that happens in the real world among the meeting creator and other people. It can be created, modified and deleted by the creator.
* Trip: it indicates the route and the travel means chosen, based on user’s preferences.

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

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

2.4 ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

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

3.3 PERFORMANCE REQUIREMENTS

3.4 DESIGN CONSTRAINTS

3.4.1 STANDARDS COMPLIANCE

3.4.2 HARDWARE LIMITATION

3.4.3 ANY OTHER CONSTRAINT

3.5 SOFTWARE SYSTEM ATTRIBUTES

3.5.1 RELIABILITY

3.5.2 AVAILABILITY

3.5.3 SECURITY

3.5.4 MAINTENABILITY

3.5.5 PORTABILITY

4 FORMAL ANALYSIS USING ALLOY

5 EFFORT SPENT

6 REFERENCES