**Table of Contents**

Table of Contents 2

1. Introduction 5

1.1. Purpose 5

1.2. Scope 5

1.2.1. Description of the given problem 5

1.2.2. Current System 5

1.2.3. Goals 6

1.3. Definitions, acronyms and abbreviations 6

1.3.1. Definitions 6

1.3.2. Acronyms 7

1.3.3 Abbreviations 7

1.4. References 7

1.5. Overview 7

2. Overall Description 9

2.1. Product perspective 9

2.2. Product functions 9

2.2.1. Payment management 9

2.2.2. Car maintenance 9

2.2.3. Others 9

2.3. User characteristics 10

2.3.1. Actors 10

2.4. Constraints 10

2.4.1. Regulatory policies 10

2.4.2. Hardware limitations 10

2.4.3 Interfaces to other applications 10

2.5. Assumptions and dependencies 11

2.5.1. Text assumptions 11

2.5.2. Domain assumptions 12

3. Specific Requirements 13

3.1. External interface requirements 13

3.1.1. User interfaces 13

3.2. Functional requirements 14

3.2.1. [G1] Allow a visitor to become registered User after providing credentials and payment information.  14

3.2.2. [G2] Allow an User to find locations of available cars.15

3.2.3. [G3] Allow an User to reserve a car.  15

3.2.4. [G4] Allow an User to unlock the car that he/she has registered. 16

3.2.5. [G5] Allow an User to complete and pay for a full ride with the car that he/ she has registered. 16

3.2.6. [G6] Allow an User to get discounts on prices if he/she behaves well and respects certain conditions. 16

3.2.7. [G7] Allow a System Manager to do operations on the system for updating and maintenance.

3.3. Non-functional requirements

3.3.1. Performance

3.3.2. Reliability

3.3.3 Security

3.3.4 Scalability

3.3.5. Accuracy

4. Scenarios

4.1. Scenario 1

4.2. Scenario 2

4.3. Scenario 3

4.4. Scenario 4

4.5. Scenario 5

4.6. Scenario 6

4.7. Scenario 7

4.8. Scenario 8

5. UML modeling

5.1. Use case descriptions

5.1.1. Visitor registration

5.1.2. User login

5.1.3. User reserves an available car after GPS localization

5.1.4. User reserves a car after searching it by address

5.1.5. User unlocks a car

5.1.6. User stops a ride

5.1.7. User enters in pit-stop mode

5.1.8. User plugs a car into a power grid

5.1.9. System manager login

5.1.10. System manager searches for a car

5.1.11. System manager edits system fields

5.1.12. System manager adds a new car to the system

5.1.13. System manager adds a new safe area

5.2. Use case diagrams

5.2.1. Visitor

5.2.2. User

5.2.3. System manager

5.3. Class diagram

5.4. Statechart diagram

5.5. Sequence diagrams

5.5.1. Visitor registration

5.5.2. User login

5.5.3. User reservation

5.5.4. User in a Ride

5.5.5. User enters in pit-stop mode

5.5.6. System manager adds a new car to the system

6. Alloy modeling

6.1. Signatures

6.2. Facts

6.3. Dynamic model

6.4. Results

6.4.1. Proof of consistency

6.4.2. Generated world

7. Appendix

7.1. Used tools

7.2. Hours of work

7.2.1. Bassetto Riccardo

7.2.2. Belloni Massimo

**1. Introduction**

* 1. Objectives

This document represents the Requirement Analysis and Specification Document (RASD). The main goals of this document are to analyze the users in order to model a system that meet their needs, describe the system in terms of functional and non- functional requirements, specify the constraints and the limits of the software and define the main typical use cases and users behaviors . This document is addressed to the developers who have to implement the requirements and could be used as a contractual basis.

* 1. Scope

1.2.1 Description of the given problem

In this project we are going to develop and implement an application called Travlendar. It is a calendar-based application which allows you to create a calendar according to the events you have (meetings whether for work or personal reasons) and in addition to calculating the time that the user has between appointments so that he does not arrive late, the application will suggest the user the best mobility option between the appointments and also will alert him when it is impossible to reach a specific event on time. In addition to these functionalities, the system will allow the user to buy public transportation tickets and locating the nearest point to hire another type of service (bike of a bike sharing system, car of a car-sharing service, etc). Users could define their transportation preferences, they can activate or deactivate any kind of transportation (including walking). The application will also take into account the weather in the location of the user. If it is raining at the time the user has to move to another event, the system will take this into account and will change the way of transport if it is necessary. The application will also allow the user to define breaks to eat or to develop other types of activities. In this way the system will organize the appointments of the user according to their breaks and the time they need to do this activities. Finally users should also be able, if they wish to, to select combinations of transportation means that minimize carbon footprint.

1.2.2 Cureent System

Even though there are already applications in the market in charge of the planning of meetings, none of these have the functionalities of Travelendar. These are limited to organize the events and notify the participants through the application and that these participants can be put in contact with other users. By this we can say that there is no application in the market with the qualities of Travlendar and therefore it has no competitors until now.

1.2.3Goals

* [G1] Allow an User to create a calendar with meetings at one time in a specific location.
* [G2] Allow an User to specify their own preferences according to the availability of that means of transport and to their pleasures.
* [G3] Allow an User introduce the breaks that he requires during the day and the temporal range in which he wants to do the rest.
* [G4] Users should receive an alert if, when entering the location of an event, the system notifies that there is no time to arrive on time.
* [G5] Allow an User to choose means of transport that minimize carbon emissions.
* [G6] Users should receive an alert if the weather conditions are not adequate or if there is some altercation in the route to their next appointment.
* [G7] The user must receive directions from their current location to reach the meeting.
* [G8] The system must take into account in addition to the bad weather, days of strike or in general days in which the public transport or the transport chosen by the user is reduced.
* [G10]
  1. Definitions, acronyms and abbreviations
     1. Definitions
* *Meeting:* appointment of any kind whether related to work or personal.
* + 1. Acronyms
* *RASD*: Requirement Analysis and Specification Document.
* *API*: Application Programming Interface
  + 1. Abbreviations
* [Gn]: n-goal.
* [Dn]: n-domain assumption.
* [Rn]: n-functional requirement.
  1. References
  2. Document Structure

1. In the first part of the document is defined the objectives of the document, also are identified the main goals of the application. In the same way is explained, without going into much detail, how the application works. Finally is given some information about definitions and abbreviations to better understand the rest of the document.
2. In the second part it is given an overall description of the system including the functions of the application, clarifying some concepts of the system. Also list the actors who are going to take part of the system. In the same way the contraints and limitations of the system will be defined. Finally are specified text and domain assumptions to resolve certain types of doubts that may arise by reading the document.
3. The third part of the document refers to specific requirements. Here are defined both functional and non functional. In this part of the document we go into more detail in the aspects mentioned in section 2.