**Abstract**

The main task of this document is to give a specification of the requirements that our system has to fulfil adopting the IEEE-STD-830-1993 standard for RASD documentation . It also introduces the functional and non-functional requirements via UML diagrams and a high level specification of the system. In the last part of this document it presents the formal model of the specification using Alloy analysis.

The information in this document are intended for the stakeholders and the developers of the project. For the stakeholders this document presents a description useful to understand the project development, meanwhile for the developers it’s an useful way to show the matching between the stakeholders’ requests and the developed solution.

**Introduction**

In this section we will explain which are the main scopes of the Travlendar+ application and we will provide a general overview of all the features.

**Purpose**

The main purpose of the RASD document (Requirement Analysis and Specification Document) is to highlight the domain in which the system will work and the primary use cases.

The document below specifies who will use the system, the reason for which the system is developed, what services will be provided and the environment in which the system finds its use.

We will need to define the functional and not functional requirements of the system.

**2.1 Product Perspective**

We are going to release a cross-platform web application which will be able to run on every device. This application won't provide an interface for the service administrator, because it is possible from the server side. As we are developing a WebApp, we are willing to release custom API for any future application in order to facilitate further implementations.

**World and Machine model interpretation**

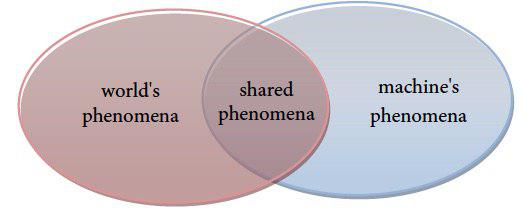
****

Figure 2.1: Relation between world and machine phenomena.

From now on we will refer to world as everything that not concern our system and machine as everything about our system. Therefore world's phenomena are all the external events happening in the world and machine's phenomena are the events related to the system. There are as well some shared phenomena which are observable by both the parts.

In our case the world's phenomena are:

* Bus out of service or other mechanical failure in public transports
* Closed roads due to a public demonstration

In our case the machine's phenomena are:

* Estimation and managing of users' free time
* Database query
* User's registration

In our case the shared phenomena are:

* Tracking of public transport's means
* Strikes and public services manifestation

**Product Functions**

The product provide to users a simple and user-friendly interface to schedule their events and help them to organize their journey. In particular the system is expected to be able to:

* Let the guest register and log in as users
* Schedule several events for every user
* Computate the best possible path for the daily journey
* Notify the user if the current journey is not realizable and why
* Interact with third part and sharing services
* Let the user buy tickets for the ride
* Support the user during the whole travel
* Allow the user to insert his preferred choice for the used transports
* Gives a real time evaluation of the environmental conditions (like strikes and
* weather situation) and uses it to nd the best choice for the journey.

**External Interfaces Requirements**

**User Interface**

The user interface must be user-friendly volted to guarantee to the nal user an easy way to interact with the application. The development of the front end is realized due to HTML and CSS to provide a responsive interface for all devices.

**Hardware Interface**

The system must be able to acquire the correct position of the means of transport and sharing services' vehicles. In order to acquire such data, every mean of transport must have a GPS localizer and internet connection to communicate its position to their server.

**Software Interface**

The system in order to be developed, both on the front-end side and back-end side, needs the use of the following programming, markup and representation languages:

* JavaScript
* HTML
* CSS
* Python
* SQL
* JSON

There are no particular limitation about the OS of the user's devices, as the application is cross-platform and can be also used with modern browsers.

**Communication Interface**

The communication between client and server happens using HTTPS by TCP protocol using port 443. A crypted channel is also used in order to perform secure transactions with third part sellers. On the other hand the connection which allow to download public informations from third part API provided by major services like Google uses HTTP with port 80.

**3.2 Functional Requirements**

The user must be able to:

Register to Travlendar+ application.

The system:

* Provides a form to insert all the personal information like name, surname, e-mail...
* Veries the genuineness of the e-mail address by sending an e-mail with a confirm request

2. Create a daily schedule specifying the location in which the event will take place and its timetable.

The system:

* Provides a form to be fulllled with all the information about the event
* Checks that all the relevant elds were compiled in order to approve the request
* Provides a method to prove that the request was conrmed
* Deletes all unsaved changes to the planning if the internet connection is absent or the application server is down
* Finds the best route solution starting from some user's parameters
* Noties the user whenever the given parameters lead to an impossible solution
* Takes in account the weather condition when looking for the best mean of transport and provides the options that can be selected by the user
* Track the user initial position throught the GPS or an inserted address
* Noties the user if there is a lack of internet connection and therefore a restriction to the application's services

3. Manage his own data.

The system:

* Provides a method to change all the user's data like e-mail, telephone number..
* Requires another e-mail verication in case the original e-mail was changed
* Let the user modify his preferences about the desired means of transport
* Let the user set or modify his preferences about the desired break time interval and how much it lasts

4. Retrieve informations about the weather forecast in a specied range of dates

The system:

* Allows the user to select the date to check the weather in
* Retrieves the datas about the weather forecast
* Uses the GPS position in order to retrieve the informations about the city the user is requesting the forecast for

5. Buy tickets and subscription for public transports.

The system:

* Provides the user all possible choices to perform a payment (like PayPal, Credit Card and other kinds of money transfer)
* Noties the user, whenever a fail occurs, the reason why it failed
* Runs along with third parts applications to fullll the request
* Memorizes all the bought or already owned tickets or subscriptions
* Provides an easy and accessible way to visualize already bought tickets
* Noties the user if any of the subscription are evaluated

6. Share his own daily schedule with other users.

The system:

* Provides an easy way to share datas with other users
* Saves the received shared schedule to be seen later

7. Modify an already compiled schedule for a certain day.

The system:

* Provides a method to allow the user to reschedule the timetable with new preferences
* Noties the user whenever his changes lead to a non optimal or impossibile solution
* Let the user delete an already created schedule

8. Choose his preferred time for having a meal.

The system:

* Let the user decide the range of time when he wants to eat and the preferred time

9. Receive information about the current journey.

The system:

* Provides all the informations about the selected journey showing the way to travel
* Shows any information about the current trac jam and road works
* Informs the user about the waiting time for public transports