Sommario

[1. Introduction 4](#_Toc496309583)

[1.1. Purpose 4](#_Toc496309584)

[1.2. Scope 4](#_Toc496309585)

[1.3. Definition, Acronyms, Abbreviations 4](#_Toc496309586)

[1.4. Revision history 4](#_Toc496309587)

[1.5. Reference documents 4](#_Toc496309588)

[1.6. Document structure 4](#_Toc496309589)

[2. Overall Description 5](#_Toc496309590)

[2.1. Product perspective 5](#_Toc496309591)

[2.2. Product functions 5](#_Toc496309592)

[2.3. User characteristics 5](#_Toc496309593)

[2.4. Domain Assumption and Dependencies 5](#_Toc496309594)

[2.5. Constrains 5](#_Toc496309595)

[3. Specific requirements 6](#_Toc496309596)

[3.1. External interface requirements 6](#_Toc496309597)

[3.1.1. User interfaces 6](#_Toc496309598)

[3.1.2. Software interfaces 13](#_Toc496309599)

[3.2. Functional requirements 13](#_Toc496309600)

[3.2.1. [G1] Allow a Guest to create a registered Travlendar+ account. 13](#_Toc496309601)

[3.2.2. [G2] Allow an User to log in into his Travlendar+ account. 13](#_Toc496309602)

[3.2.3. [G3] Allow an User to create a new appointment in his calendar. 13](#_Toc496309603)

[3.2.4. [G4] Allow an User to delete an existing appointment from his calendar. 13](#_Toc496309604)

[3.2.5. [G5] Allow an User to edit an existing appointment in his calendar. 14](#_Toc496309605)

[3.2.6. [G6] Allow an User to view his appointments. 14](#_Toc496309606)

[3.2.7. [G7] Allow an User to view his Daily Schedule 14](#_Toc496309607)

[3.2.8. [G8] Allow an User to navigate and choose between different travel alternatives. 15](#_Toc496309608)

[3.2.9. [G9] Allow an User to manage alerts for each appointment. 15](#_Toc496309609)

[3.2.10. [G10] Allow an User to manage his travel preferences. 15](#_Toc496309610)

[3.2.11. [G11] Allow an User to buy public transportation tickets. 15](#_Toc496309611)

[3.3. Design constraints 16](#_Toc496309612)

[3.3.1. Standards compliance 16](#_Toc496309613)

[3.3.2. Hardware limitations 16](#_Toc496309614)

[3.4. Software System Attributes 16](#_Toc496309615)

[3.4.1. Reliability 16](#_Toc496309616)

[3.4.2. Availability 16](#_Toc496309617)

[3.4.3. Security 16](#_Toc496309618)

[3.4.4. Maintainability 16](#_Toc496309619)

[3.4.5. Portability 16](#_Toc496309620)

[4. Formal analysis using alloy 17](#_Toc496309621)

[5. Effort spent 18](#_Toc496309622)

[6. References 19](#_Toc496309623)

# 1. Introduction

## 1.1. Purpose

This document represents the Requirement Analysis and Specification Document (RASD). The goal of this document is to describe the software application and focus on all its features. Furthermore, it’s interested to describe the functional and non-functional requirements of the system.

Show the constraint, imposed by stakeholders and application environment, the limits of the software.

This document is intended to all people that are interested to the project, such as stakeholders, investors and all developer and programmer that have to implement the application.

Ecc…

## 1.2. Scope

The application to develop is a mobile application that is called Travlendar+. This software is intended to help people with many commitments to manage the calendar on their smartphone.

The only action that the user has to do is insert his daily appointments. The application should be able to organize the whole user’s day, providing advice and reminding all inserted appointment.

The application aims to be an advanced calendar management system, since it isn’t a simple appointments reminder but it has a lot functionality that allow to the user to be always well organized.

Lot are the functionality that the application provides, such as the complete transport management, that allow to compute the travel time and to identify the better travel solution basing on user’s preferences and environment information, such as weather conditions.

The user can choose if travel with own car or walk. He can decide to travel also in public transport and the application provides to the user the transport schedules and which transport choose. The system allows also the functionality to buy a ticket in-app.

Furthermore, the application is able to find the car sharing or bike sharing points nearest to the user.

It has an advices system when the appointment and the travel times overlaps.

Daily the application can set a little time window (at most half an hour) reserved for the lunch. As this functionality, the user can schedule little break that the application set in day autonomously.

## 1.3. Definition, Acronyms, Abbreviations

## 1.4. Revision history

## 1.5. Reference documents

## 1.6. Document structure

# 2. Overall Description

## 2.1. Product perspective

The product we will provide is an application distributed for any kind of device that supports Android as operative system. This application will immediately be useble as soon as you install it on a device.

It will not have any internal interface for administration but it will be only user based.

(UML e stateCharts)

## 2.2. Product functions

This application aims to provide a smart calendar, which schedules the best organization, taking account of your personal appointments, which you inserted in the calendar. The computed schedule depends on some preferences that you filled out and you can modify them when you want.

## 2.3. User characteristics

We recommend the application to a person who wants to organize easily his time in the best way. He will be able to benefit from this service in a very simple way because Travlendar+ requires only basic knowledge of a simple calendar. After registering an account, the application is ready to handle his commitments, so scheduling the best organization.

## 2.4. Domain Assumption and Dependencies

* For any day user can create unlimited number of events.
* User has only one calendar.
* There isn’t any dependence between users.
* User can choose among some alternative travel proposals.
* If an event is overlapping another one, the user must select a choice from the choices proposed.
* User can delete an event.
* User can modify an event already created.
* User can change the scheduling proposed.
* User can select in which preferences the scheduling based on.
* Notification of best proposal will be shown.
* Notification of any problem that occurs will be shown.

## 2.5. Constrains

Travlender+ requires:

• Internet connection enabled on own device

• GPS available on own device

• Login during the first access

• Initially registration with an account

• Android device

• Milano as the default city

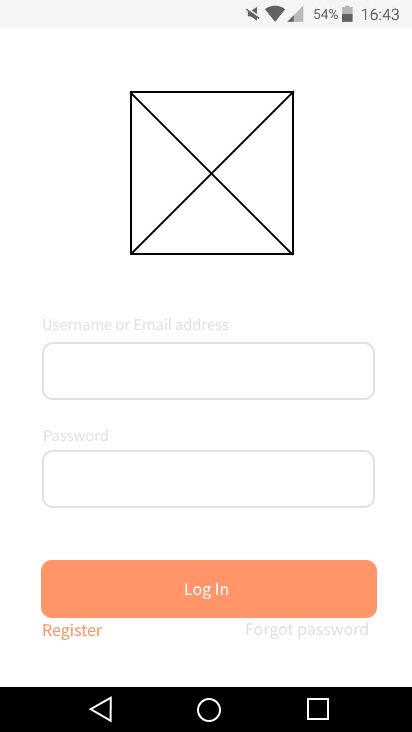
• 30 Mb(?) of storage memory available on own devise to be installed

# 3. Specific requirements

## 3.1. External interface requirements

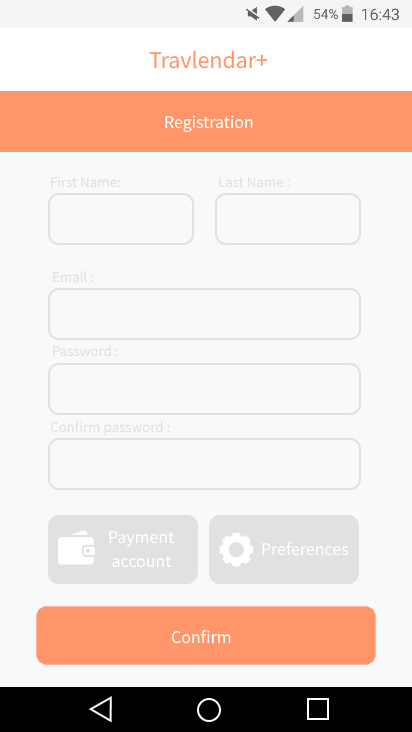
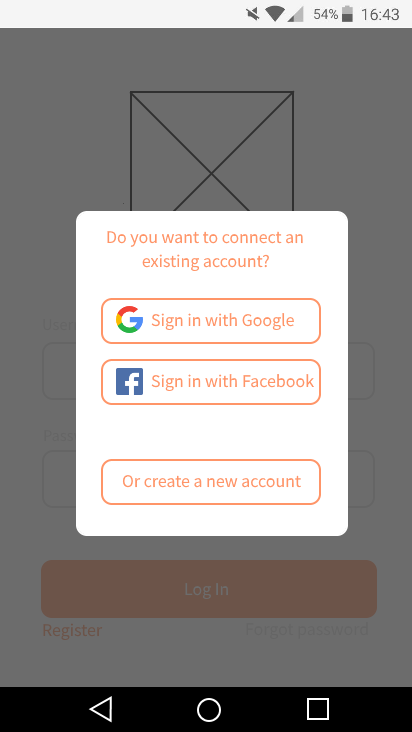
### 3.1.1. User interfaces

#### 3.1.1.1. Login



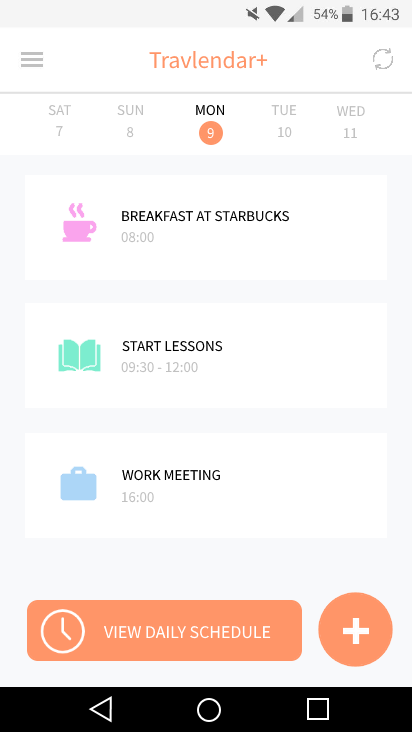
#### 3.1.1.2. Registration

#### 

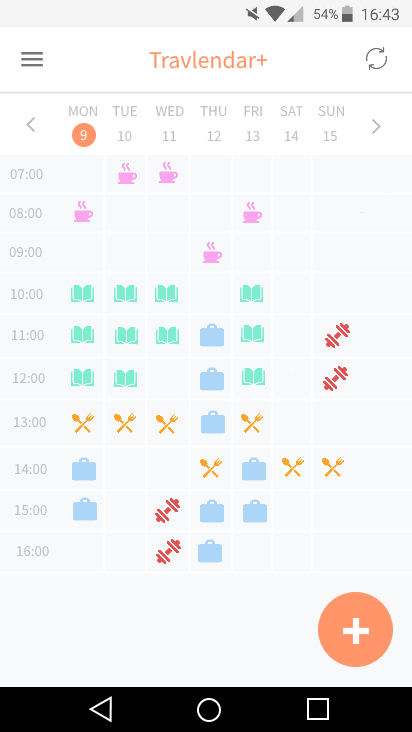


#### 3.1.1.3. Homepage

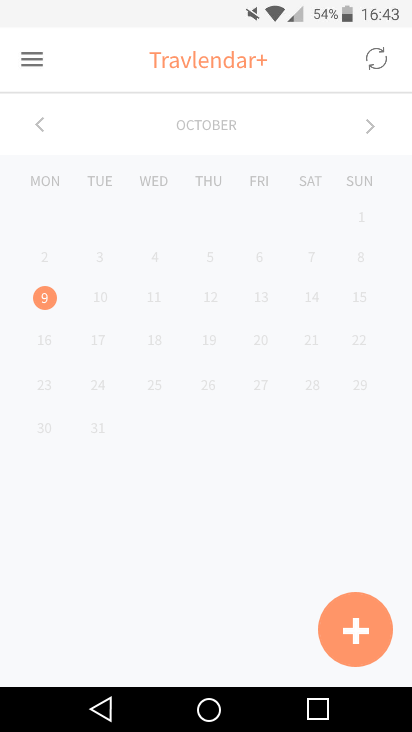
##### 3.1.1.3.1. Daily view



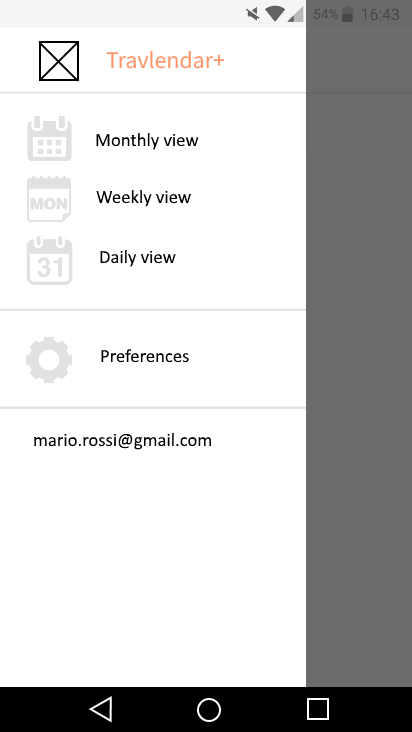
##### 3.1.1.3.1. Weekly view



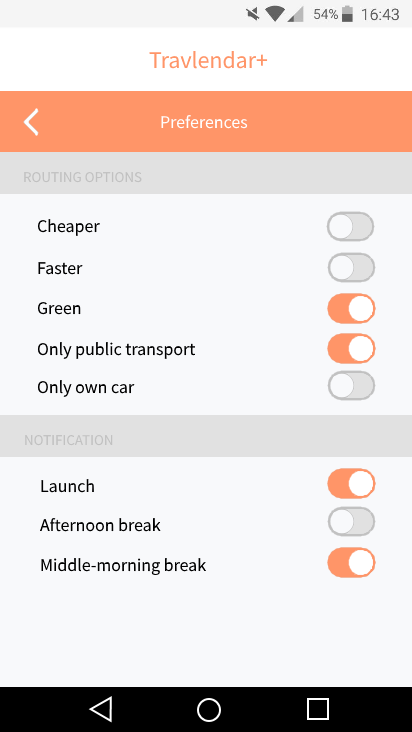
##### 3.1.1.3.1. Monthly view



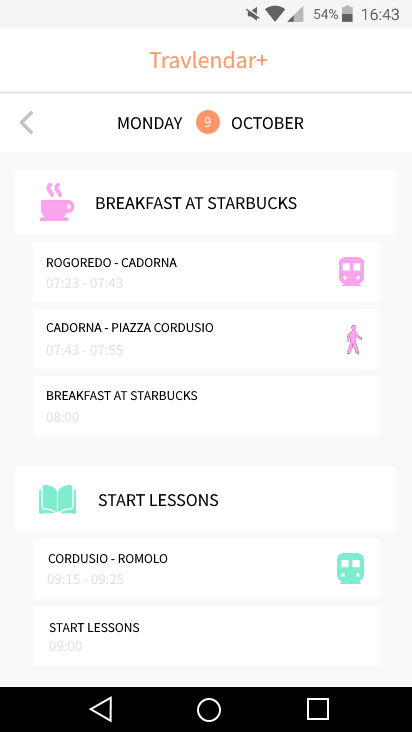
#### 3.1.1.4. Menu



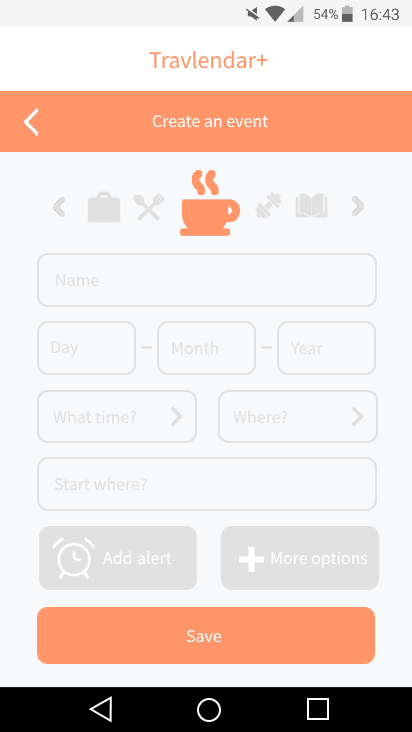
#### 3.1.1.5. Preferences



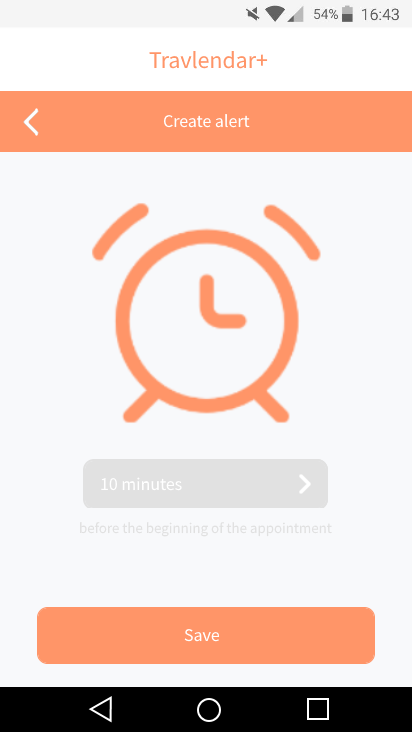
#### 3.1.1.6. Daily schedule



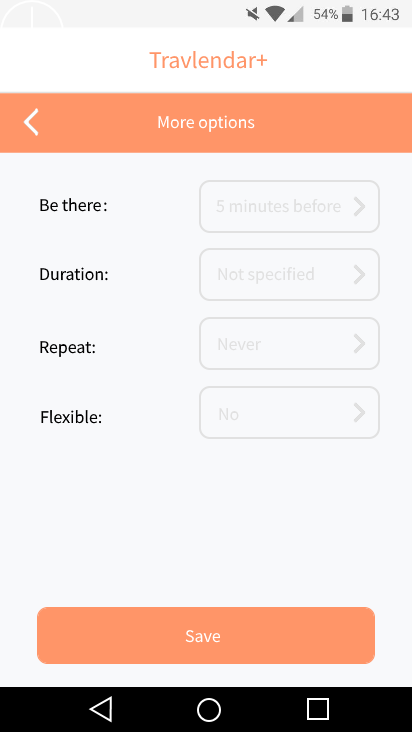
#### 3.1.1.7. Event creation



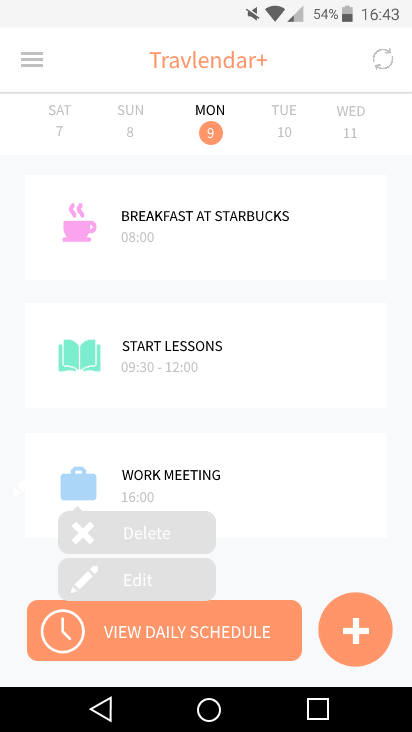
#### 3.1.1.8. Alarm creation



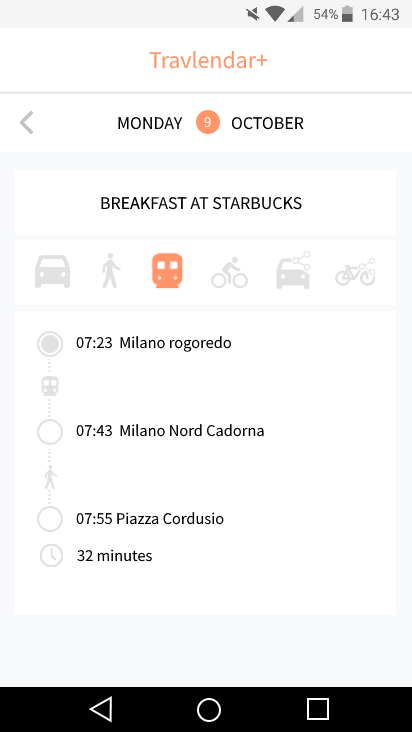
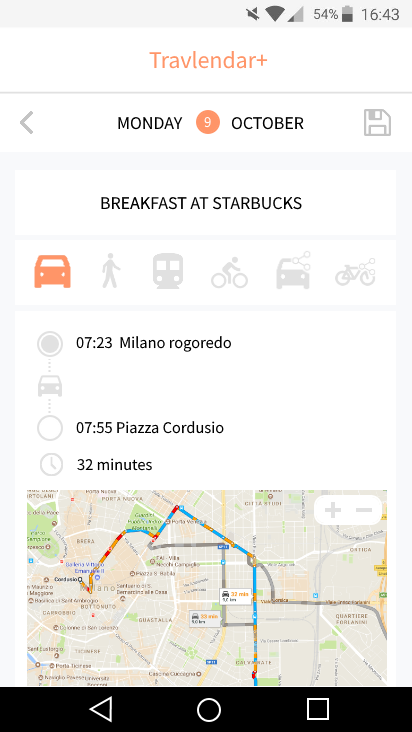
##### 3.1.1.9. Option panel



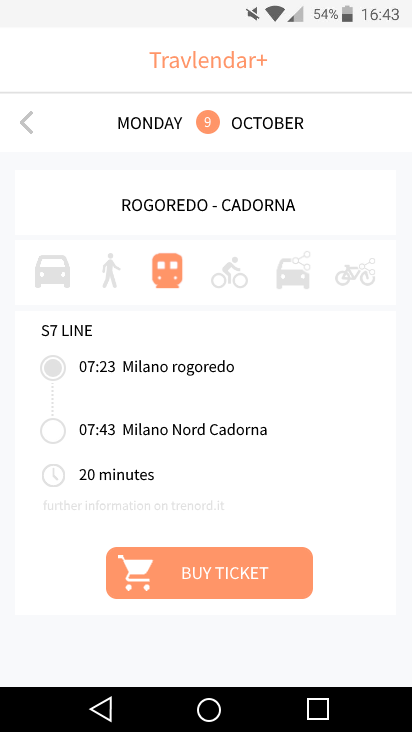
#### 3.1.1.10. Delete/Edit panel



#### 3.1.1.11. Travel details



#### 3.1.1.12. Movement details



### 3.1.2. Software interfaces

Inside application are used different API (Application programming interface):

* Weather API: <https://openweathermap.org/api>
* Google Maps API: <https://developers.google.com/maps/>
* Trenord API: <https://github.com/bluviolin/TrainMonitor/wiki/API-del-sistema-Viaggiatreno>
* Car2go API: <https://github.com/car2go/openAPI>
* Enjoy API: <https://github.com/mattiaongit/enjoy/blob/master/enjoy.py>
* BikeMi API: <https://github.com/pierlauro/bikemi-unofficial-api>
* MoBike API: <https://github.com/ubahnverleih/WoBike>

## 3.2. Functional requirements

### 3.2.1. [G1] Allow a Guest to create a registered Travlendar+ account.

* [R1] The system must ask the user to provide credentials to perform registration process.
* [R2] The system must require a valid email address and password or a link with an existing Facebook / Google account for the registration.
* [R3] The system must check that the provided email/account for the registration is not already in use from a different Travlendar+ account.
* [D1] Data used for the registration must be formally correct.

### 3.2.2. [G2] Allow an User to log in into his Travlendar+ account.

* [R5] The user must be already registered to perform login process.
* [R6] The user must be able to insert his credentials to log into the system.
* [R7] The system must be able to check the correctness of the credentials provided by user for the login.
* [R8] The system must let the user log in only if data provided are correct.

### 3.2.3. [G3] Allow an User to create a new appointment in his calendar.

* [R9] The user must be logged into the system to access application features.
* [R10] The system must be able to provide the user with an overview of his calendar and the user must be able to view all appointments fixed in a certain period.
* [R11] The user must be able to pick a chosen day from the overview of his calendar.
* [R12] The user must be able to choose the option of creating a new appointment.
* [R13] The system must ask the user to provide all information needed for the creation of a new appointment, such as place and time of start and overall duration.
* [R14] The system must check if an appointment overlaps with other events and must eventually notify it to the user.
* [R15] The user must confirm the creation of the new appointment.
* [R16] The system must save the user modifications in memory and the calendar must be updated.
* [D2] All information provided by the user in the process of appointment creation or modification must be formally corrected.

### 3.2.4. [G4] Allow an User to delete an existing appointment from his calendar.

* [R17] The appointment intended to be modified must have been previously successfully created and not already deleted.
* [R9] The user must be logged into the system to access application features.
* [R10] The system must be able to provide the user with an overview of his calendar and the user must be able to view all appointments fixed in a certain period.
* [R11] The user must be able to pick a chosen day from the overview of his calendar.
* [R18] The user must be able to choose the option of deleting the appointment.
* [R19] The user must confirm the deletion.
* [R20] The system must remove a deleted appointment from the memory and cancel every alert related to it.
* [R16] The system must save the user modifications in memory and the calendar must be updated.
* [R21] Deleting process is not reversible.

### 3.2.5. [G5] Allow an User to edit an existing appointment in his calendar.

* [R17] The appointment intended to be modified must have been previously successfully created and not already deleted.
* [R9] The user must be logged into the system to access application features.
* [R10] The system must be able to provide the user with an overview of his calendar and the user must be able to view all appointments fixed in a certain period.
* [R22] The user must be able to select a specific appointment in his calendar.
* [R23] The system must give the user access to all details of a selected appointment and the user must be allowed to edit the information needed.
* [R14] The system must check if an appointment overlaps with other events and must eventually notify it to the user.
* [R24] The user must confirm the modifications.
* [R16] The system must save the user modifications in memory and the calendar must be updated.
* [D2] All information provided by the user in the process of appointment creation or modification must be formally corrected.

### 3.2.6. [G6] Allow an User to view his appointments.

* [R9] The user must be logged into the system to access application features.
* [R10] The system must be able to provide the user with an overview of his calendar and the user must be able to view all appointments fixed in a certain period.
* [R25] The user must be able to switch between different possible calendar, such as daily calendar, weekly calendar and monthly calendar.

### 3.2.7. [G7] Allow an User to view his Daily Schedule

* [R9] The user must be logged into the system to access application features.
* [R26] The user must be able to select a specific day from his calendar.
* [R27] The system must be able to provide detailed information about the scheduled travels for a chosen day, showing the trace route and the estimated time required from each movement.
* [R28] The system must be able to choose a route between the possible travel alternatives according to the preferences expressed in the user profile settings and the information about external weather.
* [D3] Travel data are provided by an external agent.
* [D4] Information about weather are provided by an external agent.
* [D5] If the system displays a travel alternative, it means that it’s actually possible to successfully perform that travel in the way and in the time displayed.

### 3.2.8. [G8] Allow an User to navigate and choose between different travel alternatives.

* [R9] The user must be logged into the system to access application features.
* [R26] The user must be able to select a specific day from his calendar.
* [R27] The user must be able to select a specific travel in the chosen day.
* [R28] The system must be able to provide the user with an overview of the possible travel alternatives for the chosen travel, specifying all details for each one.
* [R29] The user must be able to filter the travel alternatives furnished by the system according to defined parameters, such as time of travelling or overall cost.
* [R30] The user must be able to choose a favourite travel option different from the displayed default one.
* [R31] The system must update the daily schedule according to the travel option chosen by the user and the user must be able to see the new updated schedule.
* [D5] If the system displays a travel alternative, it means that it’s actually possible to successfully perform the travel in the way and in the time displayed.

### 3.2.9. [G9] Allow an User to manage alerts for each appointment.

* [R32] The system must give the user the possibility of adding an alert to an appointment while it is being created or modified.
* [R33] The user must be able to choose a desired interval of time for the warning alert.
* [R34] The user must confirm the alert creation and the system must save the insertion in the memory.
* [R35] The user must be able to modify or remove the inserted alert when needed.
* [R36] In case of any alert modification made by the user, the user must confirm the modification and the system must save all changes.
* [D6] If a user creates a new alert, he must receive the notification after the specified amount of time.

### 3.2.10. [G10] Allow an User to manage his travel preferences.

* [R9] The user must be logged into the system to access application features.
* [R39] The user must be able to access the preferences panel of his account.
* [R40] The system must give the user the possibility of setting various preferences, such as owned and preferred travel means, address of Home and other general travel preferences.
* [R41] The user must be able to edit the provided preferences when needed.

### 3.2.11. [G11] Allow an User to buy public transportation tickets.

* [R9] The user must be already logged into the system to access his calendar.
* [R26] The user must be able to select a specific day from his calendar.
* [R27] The user must be able to select a specific travel in the chosen day.
* [R37] The system must give to the user the possibility of buying the ticket for the selected travel.
* [R38] The system must save a copy of the bought tickets and the user must be able to view them when needed.
* [D7] The payment process and ticket acquisition is made by an external public transport service.

## 3.3. Design constraints

### 3.3.1. Standards compliance

The application must require to the user different permissions:

* Access to the calendar;
* Get his position with GPS;
* Access to device storage.

## 3.3.2. Hardware limitations

The application, at the moment, runs only on Android 4.0.3 version or newer.

The device needs:

* Internet connection;
* GPS;
* Space for save application in memory.

Actual devices on the market satisfy all these requirements.

## 3.4. Software System Attributes

### 3.4.1. Reliability

The system must guarantee a 24/7 service.

### 3.4.2. Availability

The system requires a GPS service and internet connection in order to work properly. When the connection is down the system works with the last updated information available in the device memory.

### 3.4.3. Security

The application must provide secure storage for all sensitive data inserted by the user. One way to achieve it is the use of cryptographical techniques.

### 3.4.4. Maintainability

The application now is in beta version, this means that can present some bugs. Certainly, the application will be periodically upgraded and each release allow to remove known bugs and add more functionality.

Periodically all information that are stored inside the application must be backed up, in order to reduce danger of lost information in case of malfunction of the application.

### 3.4.5. Portability

Now the application has been developed only for android device (more specifically only for android version 4.0.3 Ice Cream Sandwich or newer versions).

Further developments will lead this application also in iOS devices.

Another possible development is the creation of a web site that is synchronized with application, and allow to the user to control their appointment also on desktop pc and laptop.

# 4. Formal analysis using alloy

# 5. Effort spent

# 6. References