[**1. Introduction**](#_fq6bsci23mbn) **2**

[1.1. Purpose](#_5qnaj7mmisar) 2

1.2. Scope 2

[1.3. Definitions, Acronyms, Abbreviations](#_unc0drc7k1ht) 2

[1.4. Revision history](#_5oy8ekhs4c4x) 3

[1.5. Reference Documents](#_2piarg89gwa) 3

[1.6. Document Structure](#_tc3s87judass) 3

[**2. Overall Description**](#_nv6ok59zvj0) **3**

[2.1. Product perspective](#_oxnf4u5ftsl8) 3

[2.2. Product functions](#_fddftzdsqcc7) 3

[2.3. User Characteristics](#_gu3a7i63ftgv) 3

[2.4. Assumptions, dependencies and constraints](#_rnczn98259w0) 3

[**3. Specific Requirements**](#_h0ah6uk65nt5) **3**

[3.1. External Interface Requirements](#_h05cvpuvbcb9) 3

[3.1.1. User Interfaces](#_tif8t0u8v24j) 3

[3.1.2. Hardware Interfaces](#_hewjmbewp83j) 3

[3.1.3. Software Interfaces](#_vo16ias7h15e) 3

[3.1.4. Communication Interfaces](#_17banybb6bs1) 3

[3.2. Functional Requirements](#_i0278imvdf9i) 3

[3.3. Performance Requirements](#_9y3xy1goc4pt) 4

[3.4. Design Constrains](#_6oqsxennspc7) 4

[3.4.1. Standards compliance](#_vbtakg8zlqy7) 4

[3.4.2. Hardware limitations](#_88592anot83u) 4

[3.4.3. Any other constraint](#_ahs376b9eypy) 4

[3.5. Software System Attributes](#_mk681p26x1qe) 4

[3.5.1. Reliability](#_an4efl2tx81y) 4

[3.5.2. Availability](#_4qsbrbrh1cl7) 4

[3.5.3. Security](#_ouedava0cl36) 4

[3.5.4. Maintainability](#_95owgo49hd95) 4

[3.5.5. Portability](#_sjrqbwoyho4d) 4

[**4. Formal Analysis using alloy**](#_x5ti2fho44gr) **4**

[**5. Effort Spent**](#_kbudkhw2qrw) **4**

[**6. References**](#_cvwdg9fbingp) **4**

# 

# **1. Introduction**

Travelling has never been so easy. Not only traditional vehicles improved, but new phenomena like car and bike sharing, or Uber appeared. Mobility itself is changing.

People started using new services to move, and when they need to visit a new place often rely completely on Google Maps. The same users need a better manage of their schedules.

Therefore this project wants to help users to schedule their meetings accounting the travel time between each appointment giving the possibility to customize each appointment and each travel path.

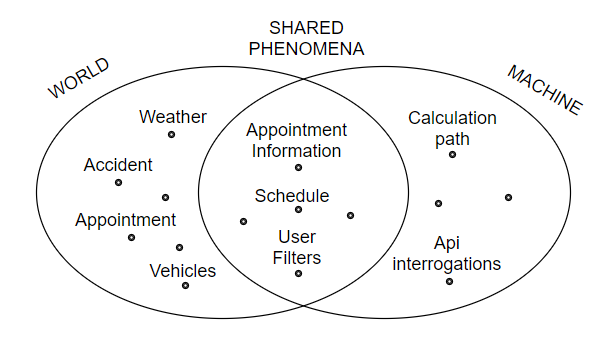
## ***1.1. Purpose***

This project wants to:

* Allow users to build a calendar where they can add appointments;
* Show the distance from one meeting to the next;
* Show all the possible travel solutions to arrive at the next appointment;
* Give the possibility to schedule particular appointment, like lunch, or private ones;
* Allow clients to filter the travel solution to select the one they prefer;
* Support users in their travel, allowing them to buy tickets directly from the platform.

## ***1.2. Scope***

The system, in order to achieve all the aforementioned goals has to interact both with users and other external services (like weather forecasting, ticket prices, etc.) which will help the application in giving a complete and reliable service.

This is an example of how the machine interacts with the environment.

Users insert their appointments and the machine calculates the best path, according to users’ preferences, making interrogations to an appropriate API, taking into account the possibility of accidents and the weather.

## ***1.3. Definitions, Acronyms, Abbreviations***

* *Vehicles****.*** All the means of transport (bike, bus, car...) that the user can use to reach the point of interest.
* *Appointment***.** An arrangement to meet someone at a particular time and place, it comprends working appointments, dates, all the events planned by the user.
* *Schedules.* A plan for carrying out a lists of intended events and times.
* *Travel**path.* Is the roadway chosen by the system based on the point the user has to go.
* *Accident.* All the event that can happen to delay the user during the travel path.
* *User**filters.* The user’s preferences on the travel path, like elements to avoid, or elements to take.

## ***1.4. Revision history***

## ***1.5. Reference Documents***

## ***1.6. Document Structure***

# **2. Overall Description**

This section describes how the system

## ***2.1. Product perspective***

## ***2.2. Product functions***

## ***2.3. User Characteristics***

## 

## ***2.4. Assumptions, dependencies and constraints***

* The system will be using various services and APIs. Therefore errors and problems may occur if something happens to these external platforms and so the continuation of the service cannot be ensured.
* The punctuality of all the public transportation cannot be guaranteed as some unpredictable accidents may happen. The time table used by the system will be the official one, provided by the managing institution of the public transportation.
* All the information about traffic and weather are in real time and represents, as much as possible, the reality
* If the user decides to buy tickets through the platform he cannot ask for a refund in any time later

# **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 Constrains***

### ***3.4.1. Standards compliance***

### ***3.4.2. Hardware limitations***

### ***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. Maintainability***

### ***3.5.5. Portability***

# **4. Formal Analysis using alloy**

# **5. Effort Spent**

Federico Amadelli 5 hrs

Alessandro Artoni 6.5 hrs

Alessio Baccelli 5 hrs

# **6. References**