



**POLITECNICO**  
**MILANO 1863**

SOFTWARE ENGINEERING II

**Travlendar+**

REQUIREMENTS ANALYSIS  
AND  
SPECIFICATIONS DOCUMENT

Authors:

*Edoardo D'Amico*  
*Gabbolini Giovanni*  
*Parroni Federico*

1<sup>st</sup> October 2017

# Indice

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Purpose . . . . .	2
1.2	Scope . . . . .	2
1.2.1	World Phenomena . . . . .	3
1.2.2	Shared Phenomena . . . . .	4
1.3	Definitions, Acronyms, Abbreviations . . . . .	4
1.4	Revision history . . . . .	5
1.5	Reference documents . . . . .	5
1.6	Document structure . . . . .	5
<b>2</b>	<b>Overall Description</b>	<b>6</b>
2.1	Product Perspective . . . . .	6
2.1.1	User Model . . . . .	6
2.1.2	Appointment Model . . . . .	6
2.1.3	Schedule Model . . . . .	7
	The optimization criteria . . . . .	8
2.1.4	Constraints . . . . .	8
	Constraints on schedule . . . . .	8
	Constraints on appointment . . . . .	9
2.1.5	Class Diagram . . . . .	9
2.2	Product Functions . . . . .	9
2.2.1	User characteristics . . . . .	10
2.2.2	Assumptions, dependencies and constraints . . . . .	11

# Chapter 1

## Introduction

### 1.1 Purpose

Our team will develop Travlendar+, a calendar-based application that aims to provide a schedule of user appointments, giving a plan to organize his daily life. The main goals the app must fulfill are:

- G1** Allow the user to insert a list of appointments according to his necessities and his preferences;
- G2** The system S.P.W. to create a valid schedule of the user appointments (fare ref alla def)
- G3** The system should offer the possibility to create a new account that the owner can use for authentication
- G4** The system S.B.A to book the travel means involved in the current schedule under user approval
- G5** The system S.B.A to display the result of the scheduling
- G6** the system should offer a GUI that handles All the interactions with the user. operation through a graphical interface

### 1.2 Scope

Here we provide a brief description of the aspects of the reality of interest which the application is going to interact with.

User can receive an appointment on a certain date, time and location (over a region), that can be reached using different available travel means. The appointment can be held either at a specific time or in a time interval and lasts for a certain amount of time. An appointment can be recurrent, in other words, it repeats regularly over time (e.g., lunch, training, etc.). User can travel with someone else and can pick up or leave off these people during the day.

User can have his own travel means and a pass for public transportation. The travel means considered in this scenario can be grouped in three categories: public, shared or private.

- Public travel means: these include trains, buses, underground, taxis, trams. They have to be taken in their **appositi** stops. User must have a valid ticket in order to get on a public travel means (except for taxis, that pick up the user wherever he wants upon a call and do not require any ticket);
- Shared travel means: these include car and bike. They are located in specific places and require a reservation in order to be used by the user;
- Private travel means: vehicles owned by the user. They can be cars, bikes, motor-bikes.

Weather conditions can change during the day affecting usable travel means. At the beginning of the day, or on demand, user can request a schedule of his daily appointments, following some criteria evaluated according to their assigned priority and satisfying some constraints imposed by the user. When a new appointment is received, user creates a new item in the application and saves it in the appointment list. User can request a reschedule to the application due to unexpected changes of his plan (e.g. a cancelled appointment).

### 1.2.1 World Phenomena

- User receives a new appointment;
- User picks up a person;
- User owns private travel means and/or passes for public transportation;
- User wakes up;

- User pass expires.

### 1.2.2 Shared Phenomena

- Shared travel mean moves;
- Shared travel mean its not available anymore;
- Wheather condition changes;
- Public travel means reach a stop-place;
- Public travel means are late;
- Public travel means are not available due to a strike day;
- User requests a schedule to the machine;
- User inserts a new appointment into the application;
- User requests to book rides;
- User moves.

## 1.3 Definitions, Acronyms, Abbreviations

sinonimi: Appointment/meeting Schedule/Scheduler System/Application preferences/-  
constraint

def:

preferences: constraints on appointments or schedules

Opt Criteria: criteria followed by the scheduler in order to optimize

travel option: the combination of travel path and travel means that allow to reach  
one spot from another.

travel option data: additional information about a travel option:

- Cost;
- Traveling time;
- Carbon emission;

Schedule: a set of time-ordered and not overlapping appointments where their starting times are fixed and they're linked each other by a path travelled with a specific transportation mean

Valid Schedule: a Schedule which:

- is optimized according to the criteria chosen by the user;
- ensures that the user will be on time for all his appointments;
- respects the constraints imposed by the user

convenzioni: variables are italic states are bold

abbr: GPS GUI: graphic user interface ETA: estimated time of arrival Should provide a way: SPW S.B.A.: should be able API

## 1.4 Revision history

## 1.5 Reference documents

## 1.6 Document structure

## Chapter 2

# Overall Description

### 2.1 Product Perspective

#### 2.1.1 User Model

A user is represented within the application by a set of parameters: Some important informations about the user are held by the following ones:

- *travelPass*: indicates if the user has a pass for public transportation;
- *hasBike*;
- *hasCar*.

#### 2.1.2 Appointment Model

An appointment is represented within the application by a set of parameters:

- *duration*: the time extension of the appointment
- *startingTime* or *timeInterval*: the first should be given if the starting hour is well-known (deterministic), otherwise a time interval in which the appointment will be held it's provided.
- *location*: identifies the coordinates of the place where the appointment will be held;
- *recurrent*: specifies if the appointment will be repeated over a fixed period of time;
- *peopleVariation*: represents a variation occourring when the user picks up or leaves off someone.

The life cycle of an appointment can be represented by the following statechart:

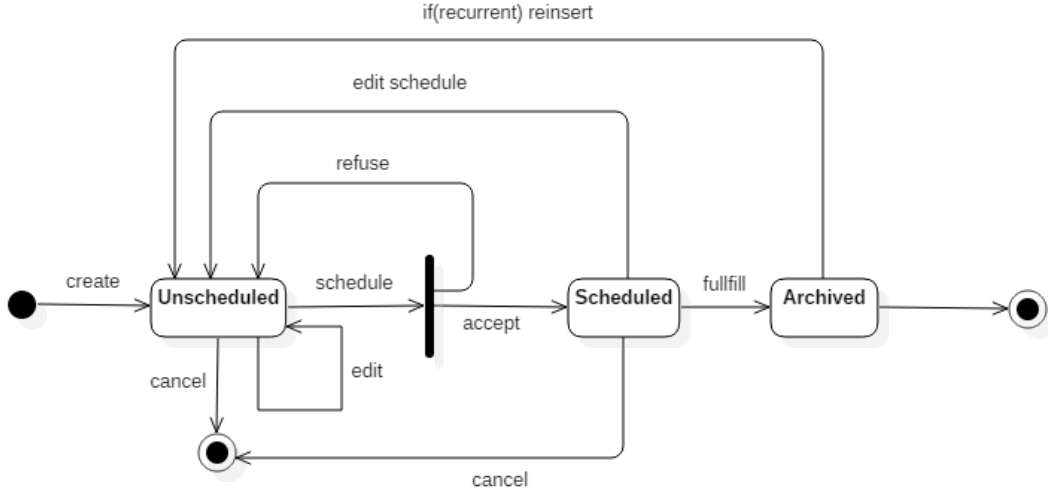


Figure 2.1: Appointment statechart

A newly-created appointment is **Unscheduled**. It could remain **Unscheduled** either when edited or there isn't a possible arrangement when a schedule is performed. Otherwise it becomes **Scheduled** if there's a feasible way to arrange it. When a scheduled appointment is edited all the appointments in that schedule return to be **Unscheduled**, because it's possible that they bring to a different schedule. When a scheduled appointment is fulfilled it becomes **Archived** and stored in the schedule history. If this last one is a recurrent appointment it must be reinsert in the list of unscheduled appointment so it will become **Unscheduled** again. The user can cancel an appointment in every moment.

### 2.1.3 Schedule Model

A schedule is a set of Appointments in a given day, ordered by the scheduler following the criteria described below. A schedule is characterized by the following variables:

- *date*;
- *startingPosition*: is the starting location of the user (e.g. user's home);



- *startingNumberOfPeople*: the number of people that must reach the first appointment.
- *wakeUpTime*: it is the starting time from which the schedule should start arranging appointments.

### The optimization criteria

The criteria that can be chosen with priority for a schedule by the user for the optimization are the following:

- *Minimize carbon footprint*: the scheduler will try to minimize the amount of kilometers travelled in polluting means;
- *Minimize money spent*: the scheduler will try to avoid expensive means and to exploit the public ones (especially if the user has a pass) or going by bike or on foot;
- *Minimize travelling time*: the scheduler will compute the quickest possible path reaching all the appointments locations.

#### 2.1.4 Constraints

Constraints are impositions on some parameters managed by the system during the process of scheduling the appointments. We can distinguish between constraints on schedule and constraints on the single appointment. These can be selected by the user when he inserts an appointment or when he requests a schedule, otherwise the constraints are initialized to default values.

#### Constraints on schedule

- *Maximum travelling distance with a specific travel mean*: the user can set a maximum amount of km to travel with a travel mean;
- *Travel means time slots*: user can specify a time interval in which a travel mean can be used;
- *User can deactivate a particular travel means*;
- *User can select which travel means uses under certain weather condition.*

## Constraints on appointment

- *User can deactivate a particular travel means.*

## 2.1.5 Class Diagram

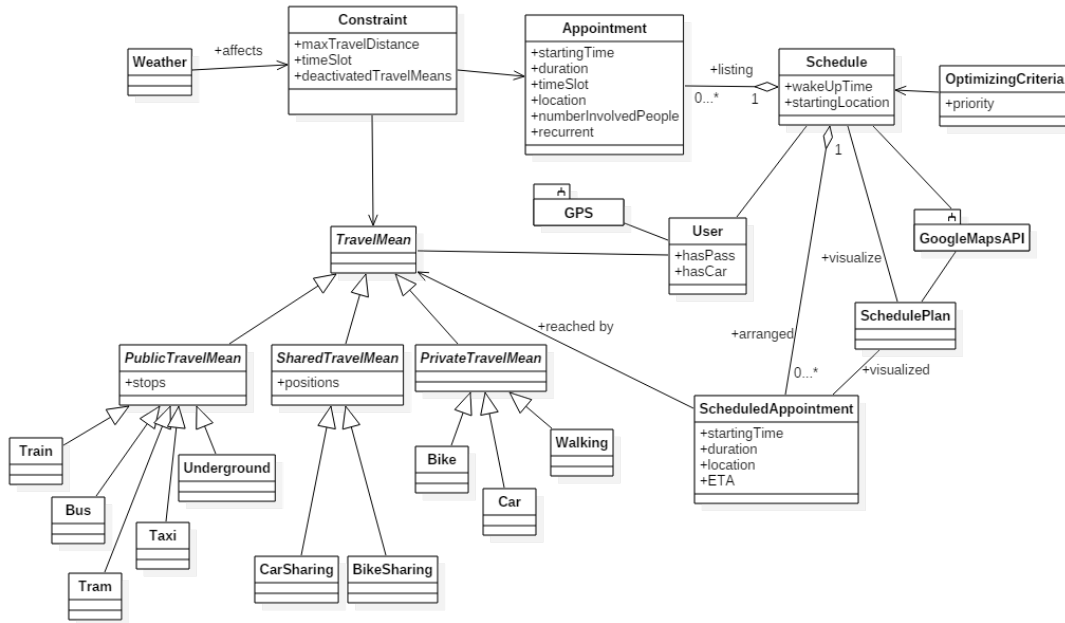


Figure 2.2: System Class Diagram

## 2.2 Product Functions

The following requirements are derived in order to fulfill the specified goals.

Requirements for **G1**:

**R1** The system SPW to create a new appointment allowing user to specify all its parameters;

**R2** The system SPW to add constraints to a previously created appointment specifying which travel means have to be avoided during the travel to the meeting location;

Requirements for **G2**:

**R3** Allow the user to set the parameters of the schedule (2.1.3), or to accept the default values;

**R4** Allow the user to select the optimization criteria (2.1.3) for the schedule;

**R5** The system S.B.A. to gather information from external APIs about:

- travel options with related travel option data;
- weather forecast;
- strike days;

**R6** The system S.B.A. to select the best travel option according to the optimization criteria taking into account:

- user constraint
- travel option data
- weather forecast
- information about strike day

**R7** The system S.B.A. to avoid a particular travel mean under a certain weather condition;

Requirements for **G4**:

**R8** The system should offer to the user a way to link all his travel service accounts into the Travlendar+ account;

**R9** The system S.B.A to book a travel mean through external API

### 2.2.1 User characteristics

Users can use our system when they want something that let them able to schedule their meetings according to their necessities and constraints. Necessary conditions for the users in order to use the system are:

- He must have an (**specificare meglio in seguito in quale dispositivo occorre farla girare**) in which run the application

This is the only requirement that is needed. Anyway additional characteristics of the user lead to the exploitation of all the system features. In fact some of them are guaranteed only after having submitted to the application some information. In this sense, welcomed user's characteristics are:

- The ownership of some travel means
- The ownership of travel passes
- The registration to sharing services

Beside these, an obviously tacit assumption that is made it's that the user has an age in which is able to move where he wants with autonomy.

### 2.2.2 Assumptions, dependencies and constraints

- The system should be able to retrieve information about public travel means. In particular:
    - information about delays;
    - information on possible strike days.
  - The system should be able to retrieve information about shared means. In particular:
    - position of the available ones;
    - prices per time unit;
  - there exist external APIs that allow to:
    - retrieve all travel options and travel option data
    - signed user to book and pay for all travel services
    - retrieve information about weather forecast
  - the device on which the application runs is connected to the internet;
- item