1. INTRODUCTION
   1. PURPOSE

Our intention is to create a software that allows the users to easily manage their meetings and commitments, by providing some useful features such as finding the best means of transport to reach the appointment place, the quickest route available to be punctual and so on.

Hence, our goals consist in:

G1. Providing a calendar and the faculty to memorize events and appointments on it

G2. Automatically computing and accounting for travel time between appointments to make sure that the user is not late for them.

G3. Automatically generating a warning to notify the user whether he/she is late

G4. Automatically providing routes and travels according to user preferences about the different travel means and daily breaks

G5. Automatically generating reminders in order to prevent the users forgetting meetings

G6. Allowing the users to modify appointment schedules

G7. Automatically notifying people involved in a specific meeting if the user is late and has selected this feature previously.

G8. Providing functions to allow the customer making the travel established for him. This means to integrate functions to book a car of a sharing service, to pay the taxi, to buy buses’ ticket and so on.

On the other hand, the purpose of this paper is to define in a detailed way all the functions and requirements of our app. In doing this, we start focusing on a briefly overview to characterize the product with relevance to its interaction with the world, then we will proceed deeply in analysing which functions are relevant to be provided and which requirements are needed to the stakeholders.

* 1. SCOPE

Tired of setting up appointment reminders, opening maps and personally estimating the best route and mean of transport by considering road congestions, forecast and other factors?

With Travlendar+ having appointments has never been so easy.

Travlendar+ is your personal assistant, it provides everything you need in a single app, analysing all the aspects for you and giving you solutions according to your preferences.

Even if we are very confident about the success of our idea, initially, Travlendar+ will have a restricted domain, indeed we will experiment it only in the Italian city of Milan.

In order to provide the most complete assistance, Travlender+ will suggest different paths and a wide range of transports such as bike, even shared, your car or a shared one, taxi, bus, and also…your feet!

It goes without saying that to organize this kind of service in the most valuable way we must interact with a huge number of local institutions which provide different services in the town.

* 1. Definitions, Acronyms, Abbreviations
* “app”: this is the abbreviation for “application”, in particular this term is used meaning a mobile application.
* “delay notification function”: this phrase refers to the function which allows to notify the participants of a meeting through an email in case the user is late.
* “Route”: this term is used as a synonym of “travel”.
* “Warning”:
* Calendar
* Meeting
* Reminder
  1. Revision History
* v1.1 – after project class meeting on 11/10/17

Added lists to goal, functions and assumptions;

Revised section 1 and 2

* 1. Reference Documents
  2. Document Structure

The RASD Document is organized following a strict schema.

In the first introductory section, we give a short description both of the goals and of the environment which our app has to deal with. Moreover, we explain some notes useful to understand and read the whole paper.

The following part consists in an overall description of the software, presenting all the functions more in depth and showing the possible interactions between the user, the system and the world itself.

The third part concerns the analysis of requirements, from the technological ones, through the functional, up to design details and constraints.

Finally, we express the requirements through the Alloy model, which allows us to define the interactions, the functions and the constraints that characterize Travlendar+ using a formal language.

The document ends with a short note about the effort spent in producing it and at last you can also find useful references.

1. OVERALL DESCRIPTION
   1. Product Perspective

Our idea is to create a personal companion application to help users managing and organizing their daily life. According to this intention, we would like to realize an extremely friendly user interface and a light software in order to make Travlendar+ affordable to many people and runnable by many devices, more than once per day.

However, considering the goals of our product, it is both required that devices support GPS services and the user has some trivial skills in using them.

Moreover, Travlendar+, which offers many routes depending on different travel means, has to interact with many institutions such as the public transport agency or sharing transport providers. This aspect will affect both the software and the hardware design. Indeed, it is necessary to query data about the shared cars, bikes and even about the taxis location around the city and to retrieve information about trains and buses schedules.

Hence, our system must be very fast and dynamic to support a huge number of query in few seconds, moreover to interview non-owned databases it’s strictly required that the users have an active internet connection.

Concerning the components, we intend to have a database which contains just username and password of all our customers. For sure, it seems useless having a database for those kinds of data but in our idea this choice allows the app. to be updated and improved easily in the future, for instance saving on the database clients’ routes and meetings.

* 1. Product Functions

F1. **Signup and Login**: Travlendar+ users must sign up the first time they intend to create a meeting and further usages of the app will require a login to access all its functionalities.

F2. **Meeting creation**: This is the most important function of the app, it allows to generate an event related to an appointment. It requires the user to define all the details such as date, time, location, starting point, preferences etc.

F3. **Preferences set up**: An important feature of Travlendar+ consists in allowing the user to filter out specific routes depending on some constraints about the travel, or to set break-dedicated time slots.

F4. **Warnings management**: In case the user seems to be late, the app generates a warning to alert him. Hence, he can either ignore it or postpone the appointment.

F5. **Delays management:** If the app had noticed, according to the estimated travel time, that the user is in late, and he previously had inserted the email address of the meeting’s participants, Travlendar+ would notify them about the delay.

F6. **Route generation**: the main hidden function of Travlendar+ is to automatically compute and suggest to the user the best travel among those which fit the preferences he has selected.

F7. Reminder management:

F8. **Recurrent events management:** The smartest function Travlendar+ will offer; it consists in allowing the user to select events to be rescheduled periodically just creating one meeting. Done this choice, the app. Automatically manages to reschedule the specific meeting according to the period that the user establish, for instance one week, one month….

* 1. User characteristics

According to our idea, Travlendar+ does not have a specific customers range, it is supposed to be used by both male and female, whatever is their age.

Obviously, considering that we intend to produce a mobile application, people who want to use Travlendar+ should be familiar with a portable device like a smartphone or a tablet.

For sure, the users should respect several requirements and limits due to the travel mean they are going to take. For instance, using a car suppose the user to have a valid driver licence, taking the bus is allowed only with an appropriate ticket, cycling means that someone knows how to use a bike, finally, as far as the sharing services are concerned, probably they require the user to be subscribed.

Moreover, users interested in dealing with Travlendar+ services must have an e-mail address, primarily due to register and authenticate themselves, secondly to use the delay notification function.

* 1. Assumptions, dependencies and constraints
     1. Assumptions

A1. Signup and Login

Considering that the assignments provided do not say much generally about users without any reference to a possible signup or login, **we assume that the registration is mandatory to create the first meeting, then every access to the app requires the login to manage each event saved. Please note that login parameters could be memorized to save and recover easily a user instance.**

A2. Meeting management

According to the requirements, we want to develop a system which allows the user to set his preferences with regards to the travels. **Moreover, we decide that he can also cancel or anticipate/postpose an event, assuming a previous reschedule agreement among the participants. It goes without saying that an appointment can also be modified, this means that a user can change either the starting location or the arrival location, the hour, the date and the other details chosen during the creation of the meeting, always making the same rescheduling agreement assumption.**

A3. Warnings

Our assumptions about the warning are the following: **when the system generates a warning, the app allows the user to modify the related event that could be cancelled or delayed. In case of the user postposes the meeting, if he provided the email addresses of the other people involved in the appointment, Travlendar+ automatically will notify them that a change occurs.**

A4. Routes

Concerning the routes, we decided to manage them in this way.

**The system generates different routes according to the user preferences, it will be the user itself to decide which itinerary fits better with him among the alternatives.**

A5. Preferences

As far as the preferences are concerned, **we decided that they belong to a user instead of a meeting. This means that a user cannot define different preferences for each meeting, while they are valid for every appointment.**

* + 1. Dependencies
    2. Constraints