Travlendar+ Requirement Analysis and Specification Document

Sinico Matteo, Taglia Andrea

Version 1.0

30 September 2017

Table of Contents

1. INTRODUCTION

the RASD does not contain items that are unrelated to the definition of requirements (e.g. design or implementation decisions).

A. Purpose:

here we include the goals of the project
WHAT
This document addresses the requirement analysis of the application Travlendar+, which will be developed in order to automatize the process of scheduling meetings at various location all across a city. No previous version of this application have been developed. The intended audience is formed by the developers of the software, the financial stakeholders and the end users.

A.1 Goals:

[G1]: Allow users to have their personal calendar;

[G2]: Allow users to edit their personal calendar;

[G3]: Allow users to edit their preferences of travel;

[G4]: Allow users to choose among a few options of day-travel proposal computed by the application based on the user preferences, the user calendar, and the weather forecast of that day;

[G5]: Allow users to check any day scheduled appointments;

[G6]: Allow users to reserve flexible slot of time for custom breaks during the day.

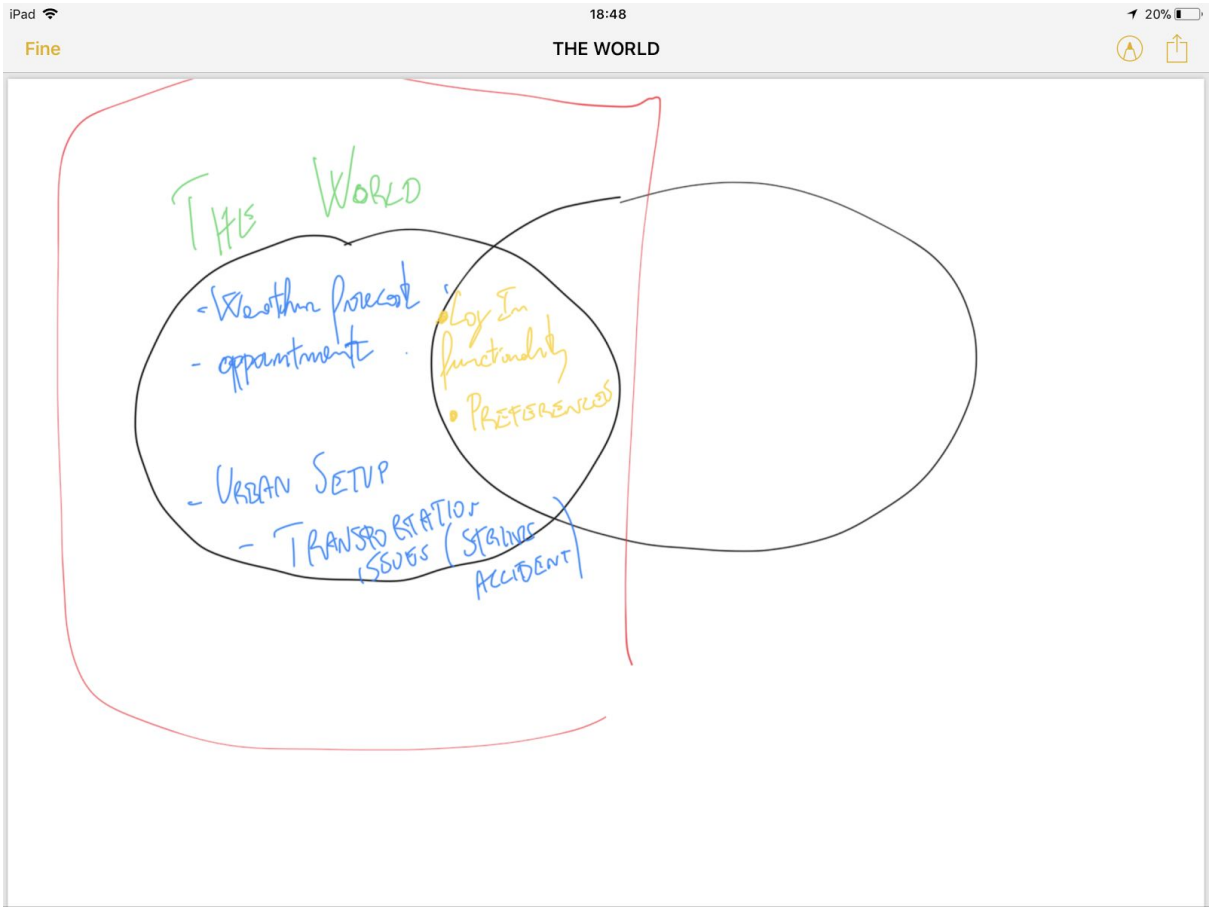
[G7]: Warning users when adding new appointment at a location which is not reachable on time.

[G8]: Allow the users to have a quick overview about travel proposal for the incoming appointment.

B. Scope:

here we include an analysis of the world and of the shared phenomena
Travlendar+ is an application which will automatize the process of scheduling the user's meeting during the day. Every day we spent a lot of time thinking and planning how to reach a location at a certain time, Travlendar+ allow users to save all this time. Users will be able to manage their own calendar by adding, updating or deleting personal appointment at various location and time during the day; they will be able to edit their preference of travel (use their own car, avoid the public transportation, use bike sharing…). Then the application

will compute a personal schedule taking into account the users calendar, preferences and the weather forecast. This will turn out into a concrete amount of saved times.
So the input that the application receive from the external world are: weather forecast, personal users appointment, transportation issues (like strike, accident or constructions), and also urban setup (streets, indications…). While what does the applications expect to been correctly specified by the users are the preferences of travel.



C. Definitions, Acronyms, Abbreviations

D. Revision history

E. Reference Documents

F. Document Structure

2. OVERALL DESCRIPTION

A. Product perspective:

here we include further details on the shared phenomena and a domain model (class diagrams and statecharts)

B. Product functions:

here we include the most important requirements
HOW

Requirements are expressed with respect to certain goals:

[G1] Allow users to have their personal calendar;

- [R1.1] A log in functionality must be provided in order to authenticate the user.
- [R1.2] User's data (calendar appointments, preferences, personal data) must be kept secret to the user himself and to anybody else.
- [R1.3] A password recovery functionality must be available in order for the user not to permanently lose access to the service.
- [R1.4] Users must be able to edit their personal data.

[G2] Allow users to edit their personal calendar;

- [R1.1] Users must be able to add an event to their personal calendar giving a title, a day, a time at which the event should be scheduled in the calendar.
- [R1.2] Users must be able to edit an existing event changing the title.
- [R1.3] Users must be able to delete an already existing event in their personal calendar.

[G3] Allow users to edit their preferences of travel;

- [R1.1] Users must be able to change their preference about car usage.
- [R1.2] Users must be able to change their preference about bike usage.
- [R1.3] Users must be able to change their preference about carbon footprint.
- [R1.4] Users must be able to change their preference about public transports.

[G4] Allow users to choose among a few options of day-travel proposal computed by the application based on the user preferences, the user calendar, and the weather forecast of that day;

- [R1.1] Computation of optimal route must take into account user preferences.
- [R1.2] Computation of optimal route should be made more times and each with different weight on preferences in order to give different day-travel proposals.
- [R1.3] If no feasible proposal can be computed then the user must be warned.
- [R1.4] There must be a way to let the user switch among different proposal.

[G5] Allow users to check any day scheduled appointments;

- [R1.1] There must be a comprehensive view over any given day.
- [R1.2] Travlendar+ application must be aware of the whole user's calendar even though a piece of it it's being showed to the user.
- [R1.3] Each appointment should have an estimated time of arrival (computed based on public transports) attached.

[G6] Allow users to reserve flexible slot of time for custom breaks during the day.

- [R1.1] Users must be able to edit preferences on break times choosing a lower and upper bound for lunch and dinner times and a minimum break time.

- [R1.2] No appointment can be successfully scheduled after another if it would not respect flexible break times.
- [R1.3] Change in break times preferences are not taken into account for previously scheduled appointments.

[G7]: Warning users when adding new appointment at a location which is not reachable on time.

- [R1.1] No overlapping appointments are allowed. A warning would show up and the user may not be able to schedule the event.
- [R1.2] Transport time estimation between appointment location being added and previous location on schedule must be done based on public transports only.
- [R1.3] The application must always be aware of the previous user location with respect to the newly added event.

[G8]: Allow the users to have a quick overview about travel proposal for the incoming appointment.

- [R1.1] The application must be aware of the current time of the day.
- [R1.2] The application must be aware of the next scheduled appointment.
- [R1.3] The application must compute a travel proposal to next event location to show to the user.

C. User characteristics:

here we include anything that is relevant to clarify their needs

D. Assumptions, dependencies and constraints:

here we include domain assumptions

3. SPECIFIC REQUIREMENTS:

Here we include more details on all aspects in Section 2 if they can be useful for the development team.

A. External Interface Requirements

A.1 User Interfaces

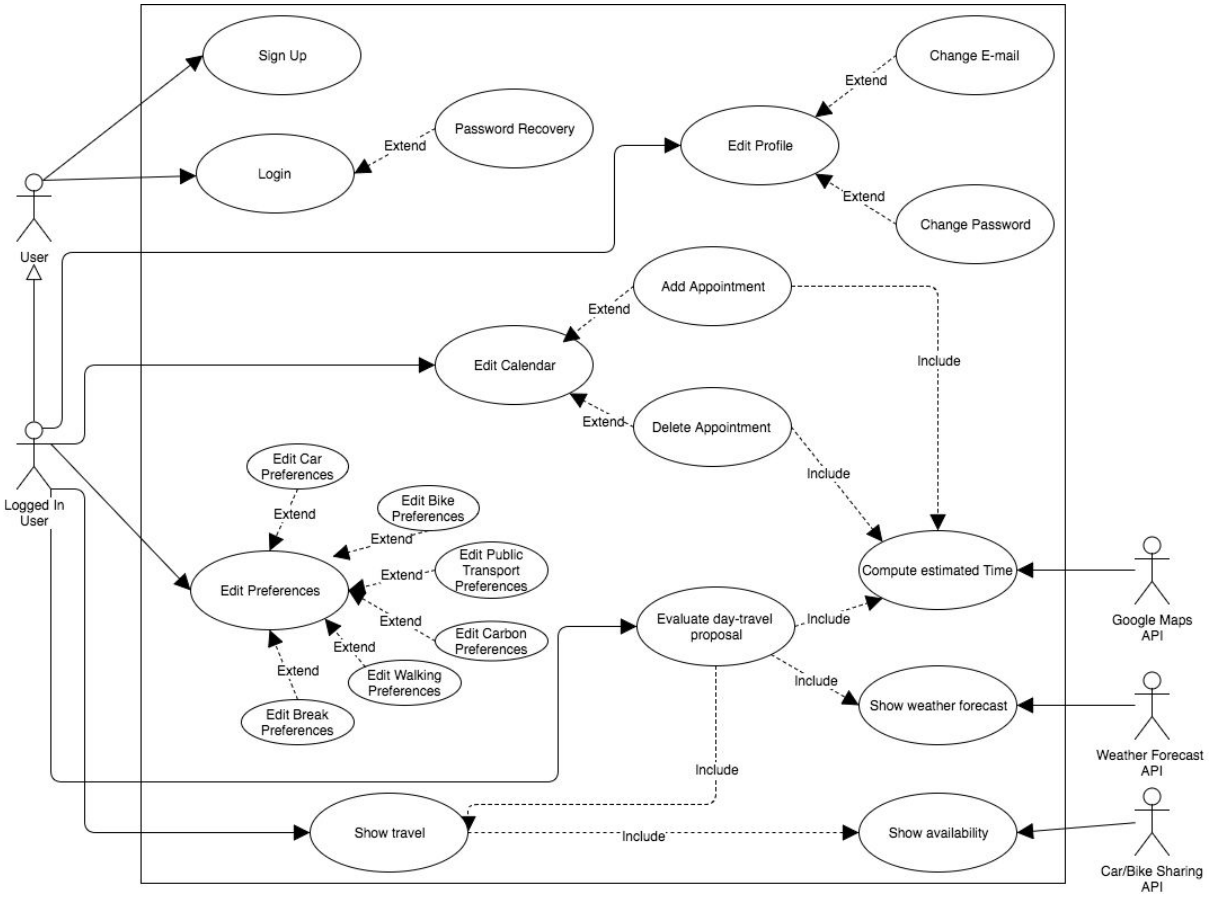
A.2 Hardware Interfaces

A.3 Software Interfaces

A.4 Communication Interfaces

B. Functional Requirements:

Definition of use case diagrams, use cases and associated sequence/activity diagrams, and mapping on requirements



[UC1]	Sign Up
Description	This use case allow the user to sign up to the system.
Actors	User
Entry conditions	The User isn't registered to the Travlendar+ application

Flow of events	• The user open the application
	• The system shows him the home page
	• He clicks on the sign up button
	• The system shows him the sign up page
	• The user fulfills the sign up form which the following information to be inserted:
	  ○ First name
	  ○ Last name
	  ○ Username
	  ○ Password
	  ○ Sex
	  ○ E-mail
	  ○ Date f birth
	• He submits the form
	• He confirms his account by clicking a link in an a confirmation e-mail previously sent automatically by the system.

Exit Conditions	Account confirmation successfully done
Exceptions	In case of missing mandatory information or username already in use or incorrect data then the system shows the user an error message highlighting the incorrect records

[UC2]	Login
Description	This use case allow the user to log in the system.
Actors	User
Entry conditions	The User is already registered to the Travlendar+ application
Flow of events	• The user open the application • The system shows the home page • The user clicks on the log in button • The system shows the log in page • The user insert his username and password • The user click on the log in button
Exit Conditions	The user successfully access the application
Exceptions	The user has forgotten his password so the system ask him if he wants to recover it by starting the password recovery use case

[UC3]

Use Case	Password Recovery
Description	This use case allow the user to recover his password in the case he has forgotten it
Actors	User
Entry conditions	The User has forgotten his password and has clicked on the password recovery button

Flow of events	• The system shows the user the password recovery page • The user fulfills the email record • The user clicks on the send code button • The user receives the security code on his e-mail account • The system shows the user the code checking page to allow him to change his password • The user fulfills the code record with the received code • The user submits it • The system generates a new password and replaces the old one • The system sends the new password to the user e-mail
Exit Conditions	The system send the new password to the user
Exceptions	If the user enters a non-registered mail or an invalid security code then the recovery procedure must be aborted by the system showing an error message. If there is a connection loss then the procedure is also aborted

Use Case	Edit Profile
Description	This use case allow the user to edit the main field of his profile
Actors	Logged in User
Entry conditions	The logged User needs to change some information about his profile and clicks the edit profile button
Flow of events	• If the user needs to change his account password or his e-mail, then he respectively clicks on the change password button which starts the Change Password use case or on the change e-mail button which starts the Change E-mail use case. • Otherwise, the user simply changes the information to be modified • The user submits the updated profile
Exit Conditions	The system successfully stores the information the user has just changed
Exceptions	If the changed data are not valid the system shows the user an error message highlighting the invalid records

Use Case	Change E-mail
Description	This use case allow the user to change the e-mail on his account
Actors	Logged in User
Entry conditions	While the logged User is editing his profile he clicks on the change e-mail button
Flow of events	• The system shows the user the change e-mail page • The user fulfills the record with his new e-mail • He clicks over the submit button to send the new information

	• The system sends him a confirmation e-mail
Exit Conditions	The user confirms his new e-mail
Exceptions	If the new e-mail is not valid the system shows an error message

Use Case	Change Password
Description	This use case allow the user to change his password
Actors	Logged in User
Entry conditions	While the logged User is editing his profile he clicks on the change password button
Flow of events	• The system shows the user the change password page • The user fulfills the records, the first one related to the old password and the second related to the new one • He clicks over the submit button to send the new information
Exit Conditions	The system stores the new password by replacing the old one
Exceptions	If the old password is wrong or the new one is invalid the system shows the user an error message highlighting the wrong record

Use Case	Edit calendar
Description	This use case allow the user to view his personal calendar
Actors	Logged in User
Entry conditions	The logged in User wants to check or to edit his personal calendar, so he clicks on the button edit calendar
Flow of events	• The system show the personal calendar page • The user select a specific day among the calendar • The system show the default day-proposal computed based on public transport times and the appointment already inserted. • If the user wants either to add an appointment or to delete an appointment or to edit an appointment already added, then he respectively clicks on the Add Appointment button which starts the **Add Appointment** use case, the Delete Appointment button which start the **Delete Appointment** use case, or he selects the appointment and he can only change the title of the appointment.
Exit Conditions	The system successfully stores the information the user has just changed
Exceptions	

Use Case	Add Appointment
Description	This use case allow the user to add an appointment to his calendar
Actors	Logged in User
Entry conditions	The User has already clicked the Add Appointment button

Flow of events	• The system show the add appointment page
	• The system ask the user to insert this following data:
	  ○ Appointment title
	  ○ Data
	  ○ Start Time
	  ○ End time
	  ○ Location address
	• The system asks a third party application to compute the estimated time based on the public transport time. This request starts the **Compute estimated time** use case.

Exit Conditions	The user successfully add an appointment to his persona calendar
Exceptions	If one of the following conditions happen: • the just added appointment isn't reachable in the estimated time • the following appointment isn't reachable in the estimated time • two or more appointment overlap the a warning is generated by the system.

Use Case	Delete Appointment
Description	This use case allow the user to delete an appointment from his personal calendar
Actors	Logged in User
Entry conditions	The User has already clicked the delete appointment button
Flow of events	• The system shows the delete appointment page. • The system asks the user to choose the appointment to delete. • The system shows the details of the appointment to delete and asks the user if is it sure to delete the following appointment. • The user clicks on the button to confirm the appointment. • The system remove the appointment • The system asks a third party application to compute the estimated time based on the public transport time. This request starts the **Compute estimated time** use case.
Exit Conditions	The user successfully delete the selected appointment.
Exceptions	

Use Case	Edit Preferences

Description	This use case allow the user to edit his travel preferences.
Actors	Logged in User
Entry conditions	The logged in User wants to check or to edit his travel preferences, so he clicks on the button edit preferences

Flow of events	• The system shows the edit preferences page
	• The user could modify the following preferences by clicking the respectively button, then the respectively use case starts:
	  ○ Car preferences
	  ○ Bike preferences
	  ○ Public transport preferences
	  ○ Walking preferences
	  ○ Carbon footprint preferences
	  ○ Break preferences

Exit Conditions	The user successfully modify his travel preferences.
Exceptions	

Use Case	Edit Car Preferences
Description	This use case allow the user to edit his car preferences.
Actors	Logged in User
Entry conditions	The logged in User has clicked on the edit Car preferences.

Flow of events	• The system shows the edit Car preferences page
	• The user could check and modify this preferences that are yes or no question.
	  ○ Disable*
	  ○ Use your own car
	  ○ Use car sharing service
	  ○ Use car for (long/short) movement**
	  ○ Avoid traffic
	  ○ Avoid motorway
	  ○ Avoid toll
	• Then the system updates the car travel preferences

Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page
Exceptions	

Use Case	Edit Bike Preferences
Description	This use case allow the user to edit his bike preferences.
Actors	Logged in User

Entry conditions	The logged in User has clicked on the edit Bike preferences.

Flow of events	• The system shows the edit Bike preferences page
	• The user could check and modify this preferences that are yes or no question:
	  ○ Disable
	  ○ Use your own bike
	  ○ Use bike sharing service
	  ○ Use bike for (long/short) movement**
	• Then the system updates the bike travel preferences

Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page
Exceptions	

Use Case	Edit Public transport Preferences
Description	This use case allow the user to edit his public transport preferences.
Actors	Logged in User
Entry conditions	The logged in User has clicked on the edit Public transport Preferences.

Flow of events	• The system shows the edit Public transport references page
	• The user could check and modify this preferences that are yes or no question:
	  ○ Disable
	  ○ Use public transports for (long/short) movement**
	  ○ avoid underground
	  ○ avoid tram
	  ○ avoid buses
	  ○ avoid trains
	• Then the system updates the public transports travel preferences

Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page
Exceptions	

Use Case	Edit Walking Preferences
Description	This use case allow the user to edit his walking preferences.
Actors	Logged in User
Entry conditions	The logged in User has clicked on the edit walking Preferences.
Flow of events	• The system shows the edit walking preferences page

	• The user could check and modify this preferences that are yes or no question:
	  ○ Disable
	  ○ Walking for (long/short) movement**
	  ○ Avoid congested streets
	• Then the system updates the walking travel preferences

Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page
Exceptions	

Use Case	Edit Carbon footprint Preferences
Description	This use case allow the user to edit his Carbon footprint references.
Actors	Logged in User
Entry conditions	The logged in User has clicked on the edit Carboon footprint preferences.
Flow of events	• The system shows the edit Carboon footprint preferences page • The user could choose among two level (low, high) of priority to associate to the importance of minimize the carbon footprint. Or could also disable the preferences clicking on the disable button. • Then the system updates the bike travel preferences
Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page
Exceptions	

Use Case	Edit Break Preferences
Description	This use case allow the user to edit his break references.
Actors	Logged in User
Entry conditions	The logged in User has clicked on the edit break preferences.

Flow of events	• The system shows the edit break preferences page
	• The user could set the following data to edit his break preferences:
	  ○ Start time (lower bound)
	  ○ End time (upper bound)
	  ○ minimum time
	or could click the disable button to disable the preferences.
	• Then the system updates the bike travel preferences

Exit Conditions	The user successfully modify his travel preferences, the system shows the edit preferences page

Exceptions	

Use Case	Evaluate day-travel proposal
Description	This use case allow the user to view a few options of the day-travel proposal, and choose one of them
Actors	Logged in User
Entry conditions	The User has clicked on the Travel Button
Flow of events	• The system request the weather forecast API the weather forecast for each appointments in the user schedule. Each request starts the **Show weather forecast** user case • Then the system, based on the user preferences, create a few day-travel proposal. • The system, request the google maps API to compute each movement that is in the schedule options. Each request starts the **Compute estimated time** use case. • Finally the system discard the solution that are not possible and choose the best ones to present to the user. • The system shows the Travel page • The user could scroll the day-travel proposal to check the solution computed by the application, or could switch to the next proposal by clicking the next button. • When the user finds the best proposal for him could accept the proposal by clicking the Accept button. • The system records the day-travel proposal accepted.
Exit Conditions	The system starts the show travel use case.
Exceptions	

Use Case	Compute Estimated Time
Description	This use case allow the google maps API to communicate with the google maps service to compute the estimated time of a movement.
Actors	google Maps API
Entry conditions	One of the following conditions has just happened: • the user has added an appointment • the user has deleted an appointment • the evaluate day-travel proposal use case request to compute a movement

Flow of events	• The google maps API create a new request to be sent to the google maps services with the following data:
	  ○ Location of the appointment A
	  ○ End time of the appointment A
	  ○ Location of the appointment B
	  ○ Means of transport selected X

	• The API send the request to the google maps services, and wait for an answer. • The services compute the estimated time to reach B from A, starting at the end time of A, with X, and answer the API.
Exit Conditions	The google maps API successfully receives the answer from the google maps service, and makes it available to the system.
Exceptions	If the API cannot receive an answer because of connection loss, sends back an error message

Use Case	Show travel
Description	This use case allow the user to see the day-travel schedule previously accepted.
Actors	Logged in User
Entry conditions	The user has already accepted one of the day-travel proposal offered by the system in the Evaluate day-travel proposal
Flow of events	• The user can scroll the day-travel schedule to see his next appointments and all the directions and informations that needs to reach them • If the day-travel schedule suggest to use a shared bike or car the user can click on the show availability button which starts the **Show availability** use case.
Exit Conditions	The user successfully see his day-travel schedule.
Exceptions	

Use Case	Show weather forecast
Description	This use case allow the weather forecast API to communicate with the weather forecast service.
Actors	Weather forecast services API
Entry conditions	The system has requested the weather forecast API the weather forecast for the current day

Flow of events	• The API request the weather forecast to the third party weather forecast services, giving it:
	  ○ the location of the appointment A
	  ○ the start time of A
	  ○ the end time of A
	  ○ the current date
	• Then waits for an answer from the weather forecast service.

Exit Conditions	The API receive an answer from the weather forecast service, and makes it available to the system.

Exceptions	If the API cannot receive an answer because of connection loss, sends back an error message

Use Case	Show availability
Description	This use case allow the car/bike sharing API to communicate with the car/bike sharing services to show the car/bike available
Actors	Car sharing API, Bike sharing API
Entry conditions	The user has clicked on the show availability button
Flow of events	• The sharing API request to the sharing service to show the current available car/bike giving it the current location, and wait for an answer.
Exit Conditions	The sharing API receive an answer from the sharing services, and makes it available to the system that shows to the user the available car/bike
Exceptions	If the API cannot receive an answer because of connection loss, sends back an error message

*If the user set the disable preferences to yes, then all the other preferences are disabled, and the algorithm doesn't take in account this category of preference to compute the day-travel proposal.
**The User can only choose among two option offered by the system (long, short) in term of distance.

C. Performance Requirements

D. Design Constraints

D.1 Standards compliance

D.2 Hardware limitations

D.3 Any other constraint

E. Software System Attributes

E.1 Reliability

E.2 Availability

E.3 Security

E.4 Maintainability

E.5 Portability

4. FORMAL ANALYSIS USING ALLOY:

in this section you will include your Alloy model. We require you to comment on the model by discussing the purpose of the model, what you can prove with it and why what you prove is important given the problem at hand. You are also required to show one or more worlds obtained by running your model.

5. EFFORT SPENT:

In this section you will include information about the number of hours each group member has worked for this document.

6. REFERENCES