POLITECNICO MILANO 1863

MSc Computer Science
and Engineering

Software Engineering 2
ACADEMIC YEAR 2017-2018

TRAVLENDAR+

Requirements Analysis and Specification Document

894135
Franklin Onwu
franklinchinedu.onwu@mail.polimi.it

899318
Ivan Sanzeni
ivan.sanzeni@mail.polimi.it

884021
Matteo Vantadori
matteo.vantadori@mail.polimi.it

Related professor:
Prof. Matteo Giovanni Rossi

Release Date: October 29th 2017
Version 1.0

Contents

1 | Introduction

1.1 Purpose

Section [G.0] treats all the goals related to the registration to the application:

G.0.1 The unregistered user can sign up to the *Travlendar+* services.

G.0.2 The registered user can sign in to *Travlendar+*.

G.0.3 The registered user can sign out from *Travlendar+*.

G.0.4 The registered user can delete his/her account from the *Travlendar+* services.

Section [G.1] treats all the goals related to the creation and personalization of an event:

G.1.1 The *Travlender* can schedule a new event adding name, time slot, location, type of event and (eventually) a description.

G.1.2 The *Travlender* can modify the name of the event.

G.1.3 The *Travlender* can modify the location of the event.

G.1.4 The *Travlender* can modify the description of the event.

G.1.5 The *Travlender* can modify the starting time of the event.

G.1.6 The *Travlender* can modify the ending time of the event.

G.1.7 The *Travlender* can modify his/her event from a work event to a personal event or viceversa.

G.1.8 The *Travlender* can insert the description at any later time and modify it at any moment.

G.1.9 The *Travlender* can choose how many minutes earlier arrive to his/her destination.

G.1.10 The *Travlender* can delete an existing event.

G.1.11 The *Travlender* can see all the events he/she has already scheduled.

Section [G.2] treats all the goals related to the customization of the *Travlender* preferences:

G.2.1 The *Travlender* can decide to choose the quickest way as default.

G.2.2 The *Travlender* can decide to choose the cheapest way as default.

G.2.3 The *Travlender* can decide to choose the most ecological way as default.

G.2.4 The *Travlender* can decide to reach the location choosing means that keep him/her out of adverse weather conditions.

G.2.5 The *Travlender* can add constrains on the transports range of time, restricting their use only in a chosen time slot.

G.2.6 The *Travlender* can restrict the use of transports, setting a maximum distance per travel.

G.2.7 The *Travlender* can set a maximum amount of money to spend in public, shared or non-shared transports per travel.

G.2.8 The *Travlender* can select the means he/she wants to use and deselect those he/she doesn't want to.

Section [G.3] treats all the goals related to the customization of the *Travlender*'s settings:

G.3.1 The *Travlender* can add his/her public transport's tickets or passes.

G.3.2 The *Travlender* can select all his/her owned means.

G.3.3 The *Travlender* can decide to allow transports accessible by people with disabilities as the only way to travel.

Section [G.4] treats all goals related to the purchase of *non-shared transports*:

G.4.1 The *Travlender* can book in-app a taxi.

G.4.2 The *Travlender* can book in-app a limousine.

Section [G.5] treats all the goals related to the purchase of *public transports*:

G.5.1 The *Travlender* can buy in-app a ticket for the metro.

G.5.2 The *Travlender* can buy in-app a ticket for the bus.

G.5.3 The *Travlender* can buy in-app a ticket for the trolleybus.

G.5.4 The *Travlender* can buy in-app a ticket for the tram.

G.5.5 The *Travlender* can buy in-app a ticket for the train.

Section [G.6] treats all the goals related to the purchase of *shared transports*:

G.6.1 The *Travlender* can take a bike from a bike sharing service.

G.6.2 The *Travlender* can take a car from a car sharing service.

Section [G.7] treats all the goals related to the special event categories:

G.7.1 The *Travlender* can create an event with a flexible time occupation.

G.7.2 The *Travlender* can create a non-reserved time event.

G.7.3 The *Travlender* can select a location outside Milan for his event.

Section [G.8] treats all the goals related to the travel:

G.8.1 The *Travlender* can modify his/her preferences for a single travel.

G.8.2 The *Travlender* can get the route for his/her event location all over Milan.

1.2 Scope

Travlendar+ is a calendar-based application designed to schelude any kind of event, supporting the user in reaching the location of the events all across Milan, combining different sort of means in relation to the user preferences.

The application is designed to match the user needs to personalize each event in every respect. So the user can easly customize each event assigning it a category and distinguishing it between work or personal reasons, deciding means and constraints to reach it and buying tickets or booking means in-app, if necessary.

The main application goal is to lead the user to handle each kind of event with *Travlendar+*: from a lunch with friends to a job interview, from an interesting expo to an out of town meeting.

1.3 Definitions, acronyms, abbreviations

1.3.1 Definitions

Cheap = with this preference the application chooses the cheapest way to reach the location.

Eco = with this preference the application chooses the most ecological way to reach the location.

Flexible event = kind of event that provides calendar, reminder and street direction supports and can be overlapped with activities as long as exists a minimum amount of time fixed by the user.

Lasting event = kind of event that provides calendar, reminder and street direction supports and can be overlapped with activities.

Non-shared transports = limousine, taxi.

Not wet = with this preferences the application chooses only means that keeps the user out of adverse weather conditions to reach the location.

Personal event = the user specifies that the event has personal purposes.

Public transports = bus, metro, train, tram, trolleybus.

Quick = with this preference the application chooses the quickest way to reach the location.

Shared transports = bike sharing, car sharing.

Standard event = kind of event that provides calendar, reminder and street direction supports and cannot be overlapped with other activities.

Transfer event = kind of event that provides calendar and reminder supports and cannot be overlapped with other activities. It is used for events that take place outside Milan.

Travlendar+ = the name of the application.

Travlender = a registered and logged user of Travlendar+.

Work event = the user specifies that the event has work purposes.

1.3.2 Acronyms

API = Application Programming Interface.

GPS = Global Positioning System.

MMS = Mapping Managing System.

RASD = Requirements Analysis and Specification Document.

TMS = Transporting Managing System.

1.3.3 Abbreviations

A.n = Application assumption number n.

G.n.m = Goal number m in section n.

D.n = Domain assumption number n.

R.n.m = Requirement number m in section n.

T.n = Text assumption number n.

1.4 Revision history

29th October 2017
Version 1.0 - Document delivery.

1.5 Reference documents

https://standards.ieee.org/findstds/standard
IEEE standard for requirements documents.

https://developers.google.com/maps
Reference point for the third-party *MMS* considered in this project.

https://citymapper.com/milano
Reference point for the third-party *TMS* considered in this project.

RASD Sample from A.Y. 2015-2016.pdf
First RASD document example from Software Engineering 2 directory, on BEEP.

RASD Sample from A.Y. 2016-2017.pdf
Second RASD document example from Software Engineering 2 directory, on BEEP.

1.6 Document structure

In the following parts we will introduce the application that allows the user to reach the 1.1 section goals, in order to soddisfy in 1.2 section problem. The document is subdivided in other five parts, besides the introduction:

Overall description
A general description of *Travlendar+*, that includes a list of the external system interfaces, an explanation of the major system functions, a description of user characteristics in detail and all our assumptions and constraints during the app creation.

Specific requirements
A detailed description of all the *Travlendar+* requirements according to the IEEE standard: from the external interface to the functional requirements, from performance requirements to the design constraints, from the software system attributes to any other requirement.

Formal analysis using alloy
The complete description of all the goals, domains and requirements using the Alloy model.

Effort spent
A complete table of all the hours spent by each team member during the project.

References
All the reference documents we lean on during the document draft.

2 | Overall description

2.1 Product perspective

The *Travlender* interacts with the system using an application on his/her smartphone. The user interface is designed for Android 7.1.1 (Nougat) or above. The application leans on a third-party *Transport Managing System* to handle all the payments related with public vehicles tickets and shared or non-shared vehicles books. It also leans on a third-party *Mapping Managing System* to handle the map, the path calculation algorithms and all the traffic or meteorological information.

2.2 Product functions

The application handles four typologies of event:

1. *Standard*: which provides calendar, reminder and street direction supports and cannot be overlapped with other activities.
2. *Lasting*: which provides calendar, reminder and street direction supports and can be overlapped with activities.
3. *Flexible*: which provides calendar, reminder and street direction supports and can be overlapped with activities as long as exists a minimum amount of time fixed by the user.
4. *Transfer*: which provides calendar and reminder supports and cannot be overlapped with other activities. It is used for events that take place outside Milan.

Travlendar+ also provides in-app purchase for public transports tickets (metro, bus, trolleybus, tram, train) in Milan and bookings for shared (bike sharing, car sharing) and non-shared transports (taxi, limousine) services.

Travlendar+ takes account of different user preferences, like the opportunity to travel by owned means (*car*, *bike* or *foot*), the possibility to choose different algorithms to set the course (*quick*, *cheap* or *eco*) and offers the opportunity to reach the location choosing means that keeps the *Travlender* out of adverse weather conditions.

2.3 User characteristics

User

A generic unregistered user, or a registered but unlogged user. At the application startup he/she can only tap on *Become a Travlender* or fullfill the login fields (in this case, he/she can alternatively tap on *Login with Facebook* or *Login with Google+*).

Travlender

A registered and logged user. He/She has access to all the application functions.

2.4 Assumptions

2.4.1 Application assumptions

A.1 For the minimization of carbon footprint, all the public transports are considered like being zero-emission, since the user presence would not influence the vehicle emissions during its travel.

A.2 The cost of public trasports depends only on the tickets price and it is considered free if the user has already a pass.

A.3 The cost of a car travel is supposed to be the same for all cars and depends only on the distance.

A.4 For the same vehicle, the fastest way is even the most ecologic.

A.5 All the vehicles belonging to the same class have the same features (speed, pollution level, fuel cost, etc.).

2.4.2 Domain assumptions

D.1 The payment details are verified by a reliable third-party service.

D.2 The third-party payment service is always available.

D.3 The traffic and meteorological information comes from a third-party service always reliable.

D.4 The GPS position is always accurate as much as possible.

D.5 It is possible to keep track of the position of user means through a third-party service.

D.6 The cost for each car travel is given by the estimate of the kilometers per liter and euros per liter.

D.7 All the trams, busses, trolleybusses and metros are available for people with disabilities.

D.8 There are no unpredictable events that can cause any delay to the user (accidents, strikes, etc.).

D.9 Taxis follow the fastest, cheapest or most ecologic way, depending on the customer request.

D.10 Public transports are never full.

D.11 The owned bikes are considered to be handy and always available for the user.

2.4.3 Text assumptions

T.1 We consider a city as application's range of functionality, in particular the full support is given for the whole city of Milan.

T.2 The constrains about the time limition of public means are chosen from the users and depends on their preferences. The availability of the the travel means are considered separately.

3 | Specific requirements

3.1 External interface requirements

3.1.1 User interfaces

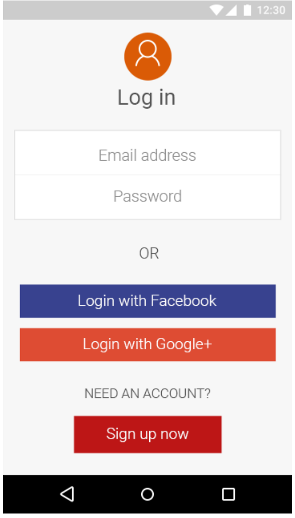


Figure 3.1: Login

The unregistered or unlogged user can log in to his/her account (if he/she has already got one) or create a new one. The application allows to log in with his/her own Facebook or Google+ account.

The first time the *Travlender* starts the application a pop-up appears, in which *Travlendar+* asks for permission to use the user's location. The user can tap on *allow* to give it, or *cancel* to refuse. In the second case, many application functions won't be accessible.

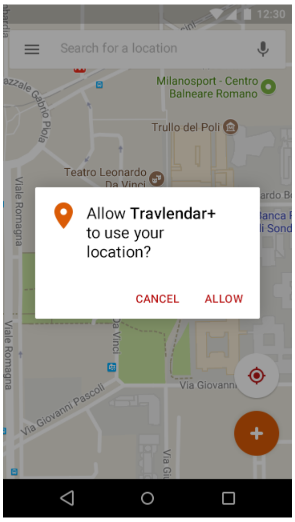

Allow Travlendar+ to use your location?

Figure 3.2: Alert

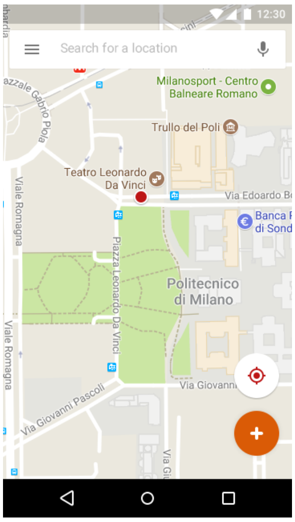
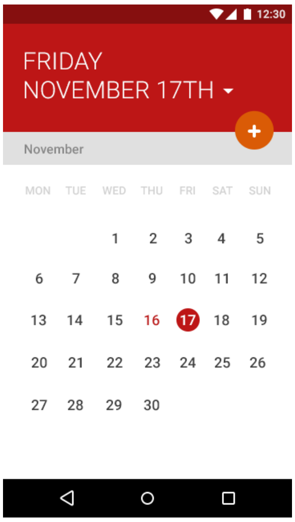


Figure 3.3: Map

If the *Travlender* decides to use the map view, he can see two buttons on the right. The first one centers the map on his/her position, the second one allows him/her to schedule a new event.

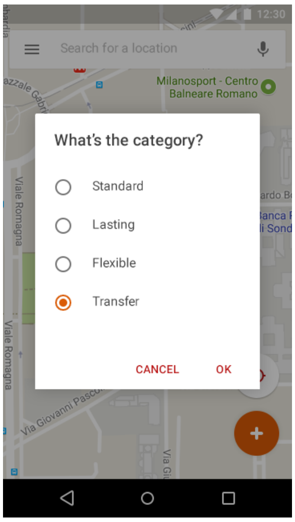
If the *Travlender* decides to use the calendar view instead, he can see only one button on the right that allows him/her to schedule a new event. The *Travlender* can go to the previous or next month swiping on the left or on the right, respectively.


FRIDAY
NOVEMBER 17TH

November

MON	TUE	WED	THU	FRI	SAT	SUN
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Figure 3.4: Calendar

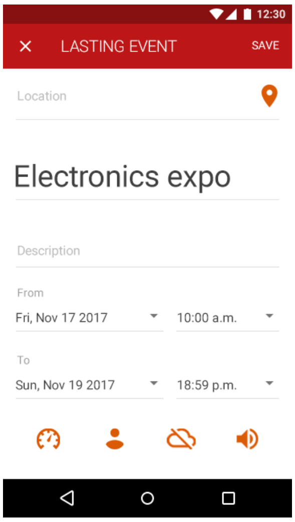

What's the category?

○ Standard
○ Lasting
○ Flexible
● Transfer

Figure 3.5: Choice of event category

When the *Travlender* decides to create a new event a pop-up appears, in which *Travlendar+* asks for the event category.

Now a new screen appears, in which the user can insert the location of the event, its name, a description, the starting and ending date and time and can modify the default settings, tapping on the four buttons on the bottom. The first one permits to change the travel preference from *quick* to *cheap*, or *eco*. The second one permits to change the event type from *work* to *personal*. The last but one enables (or disables) the *not wet* travel. The last one enables (or disables) *notifications*.



Location

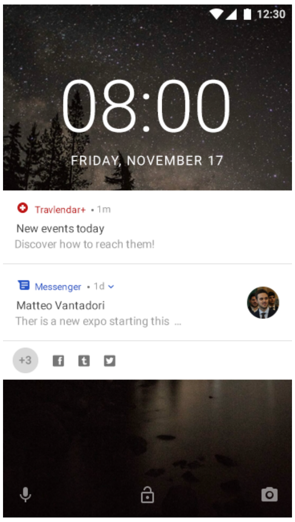
Electronics expo

Description

From
Fri, Nov 17 2017 10:00 a.m.

To
Sun, Nov 19 2017 18:59 p.m.

Figure 3.6: Creation of a new event



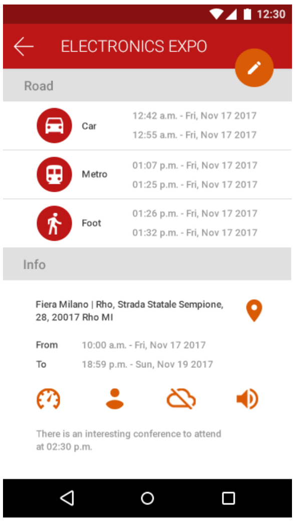
Travlendar+ • 1m
New events today
Discover how to reach them!

Messenger • 1d
Matteo Vantadori
Ther is a new expo starting this ...

Figure 3.7: Lock screen

If there is at least a lasting event, a low-priority notification appears on the lock screen once a day.

The summary permits to modify the event information with the top-right button. The button on the right permits to change the location. Also, the *Travlender* can modify the default settings, tapping on the four buttons on the bottom. The first one is the *quick/cheap/eco* button. The second one is the *work/personal* button. The last but one is the *not wet* travel button. The last one is the *notifications* button.



Car 12:42 a.m. - Fri, Nov 17 2017
12:55 a.m. - Fri, Nov 17 2017

Metro 01:07 p.m. - Fri, Nov 17 2017
01:25 p.m. - Fri, Nov 17 2017

Foot 01:26 p.m. - Fri, Nov 17 2017
01:32 p.m. - Fri, Nov 17 2017

Fiera Milano | Rho, Strada Statale Sempione, 28, 20017 Rho MI

From 10:00 a.m. - Fri, Nov 17 2017
To 18:59 p.m. - Sun, Nov 19 2017

There is an interesting conference to attend at 02:30 p.m.

Figure 3.8: Summary of the event

3.1.2 Hardware interfaces

Travlendar+ doesn't need any hardware interface.

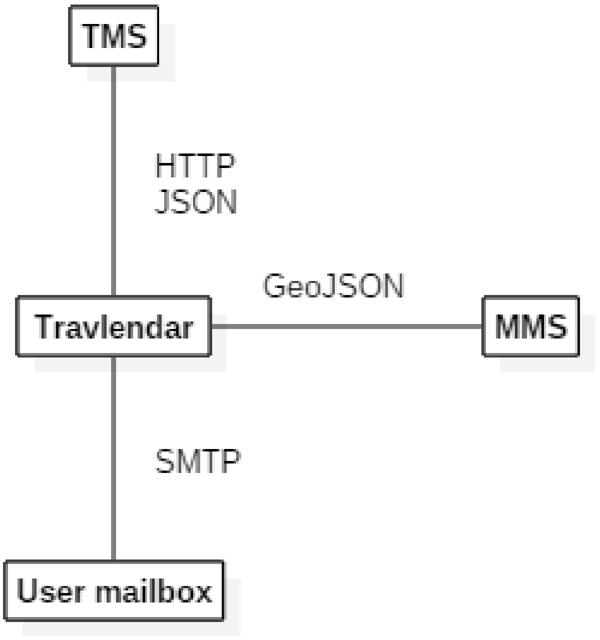
3.1.3 Software interfaces

Android 7.1.1 (or above)
Required operative system of user smartphone.

TMS
Third-party system which handles all the public, shared and non-shared vehicles purchase.

MMS
Third-party system which handles all the traffic and meteorological information and all the travel algorithms.

3.1.4 Communication interfaces



TMS

JSON The system sends to the TMS the payment details in a JSON request body, the TMS sends back a JSON response body with the details.

HTTP Required for the *201 Created* status code.

MMS

GeoJSON The system sends to the MMS the user coordinates with the GeoJSON encoding format.

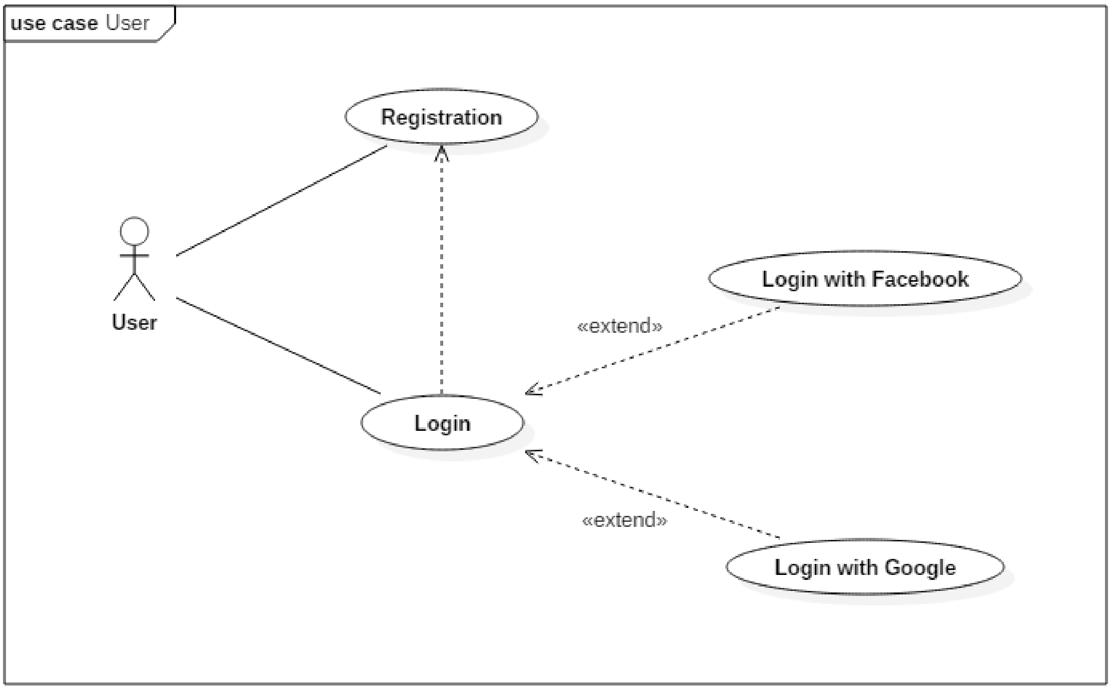
User mailbox

SMTP After the user registration, the system sends an email confirmation to the user mailbox.

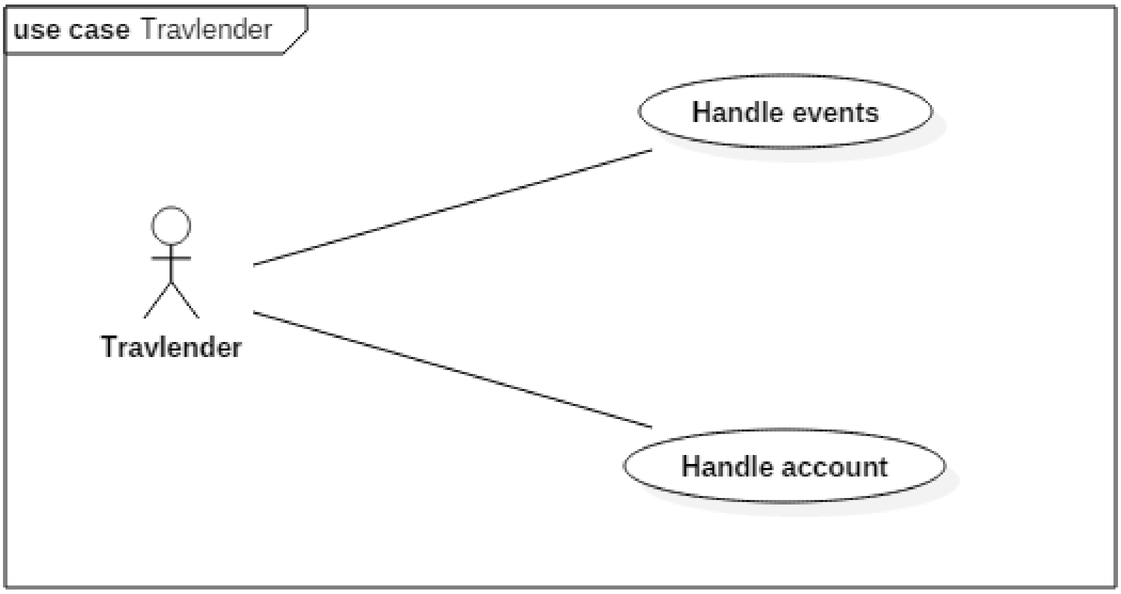
3.2 Functional requirements

3.2.1 Use case diagrams

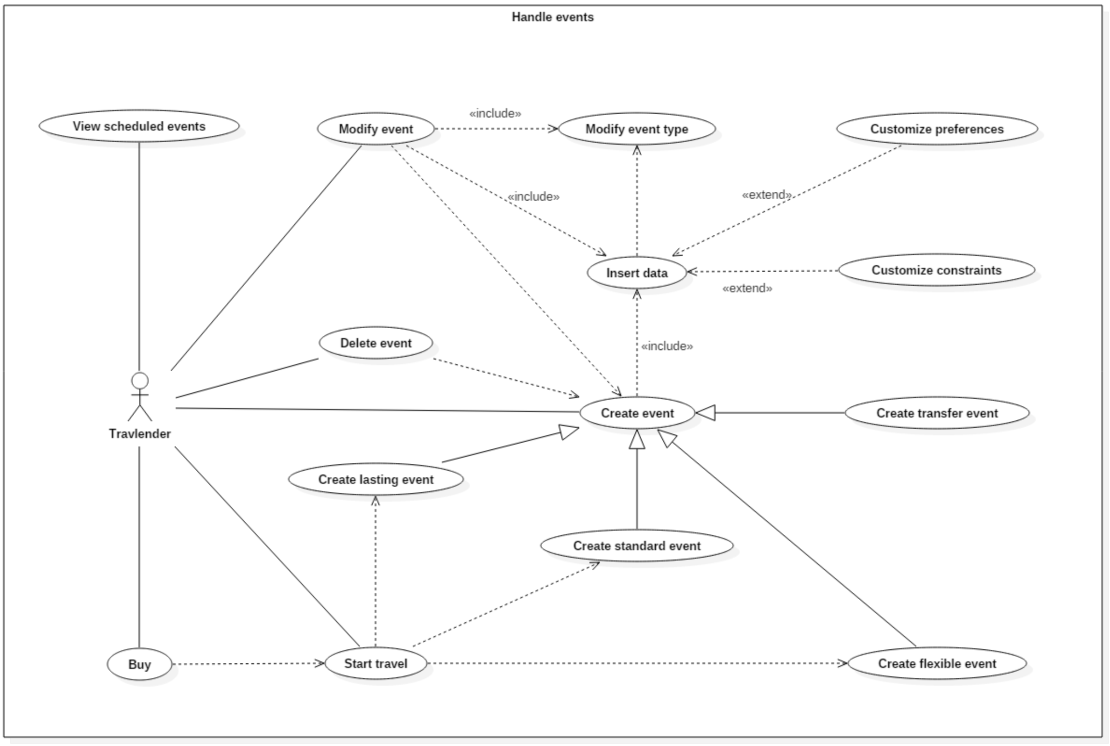
User



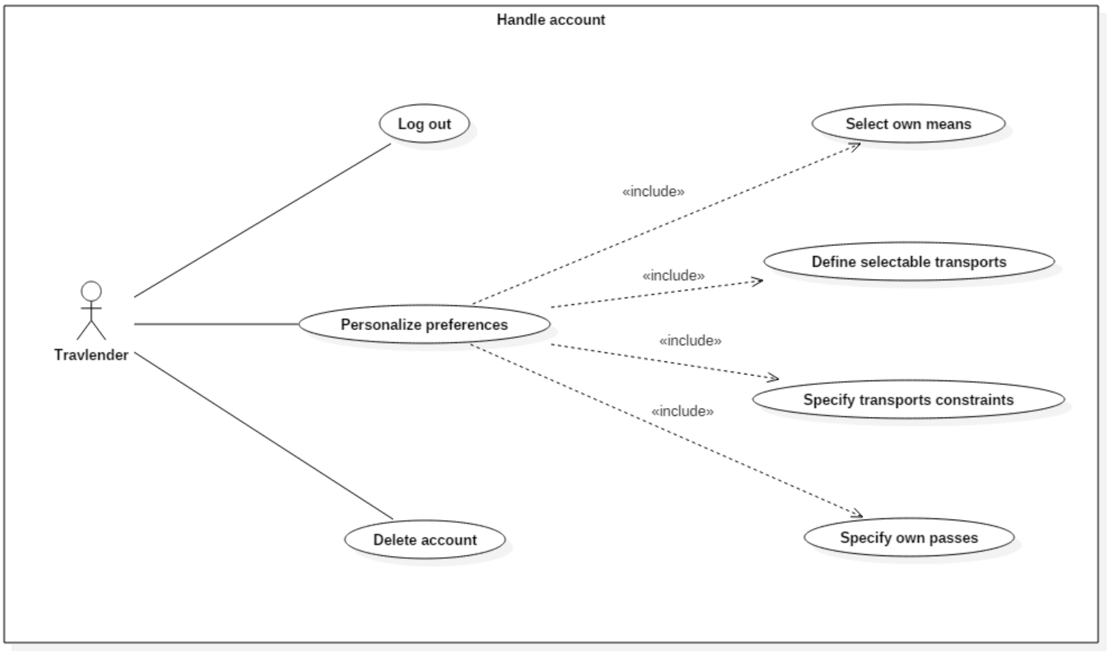
Travlender



Handle events



Handle account



3.2.2 Use case tables

User registration

Name	Registration
Actors	User
Goals	G.0.1
Input conditions	The user must have downloaded the application and opened it on the login screen
Events flow	1. The user taps *Become a Travlender* 2. The user fullfills the registration form and taps *Send* 3. The system verifies the data 4. The system shows a confirm message on the screen
Output conditions	The user is registered
Exceptions	The user makes a mistake in the registration form: 4.a. The system shows an error message on the screen Then the flow restarts from point 2

User login

Name	Login
Actors	User
Goals	G.0.2
Input conditions	The user must be registered and have opened the appliation on the login screen
Events flow	1. The user fills out *Email address* and *Password* fields and taps *Enter* 2. The system verifies the account 3. The system shows a confirm message on the screen
Output conditions	The user is logged
Exceptions	The user inserts an invalid email address or password: 3.a. The system shows an error message on the screen Then the flow restars from point 1

User login with Facebook

Name	Login with Facebook
Actors	User
Goals	G.0.3
Input conditions	The user must be registered and have opened the appliation on the login screen
Events flow	1. The user taps *Login with Facebook* 2. The system verifies the account 3. The system shows a confirm message on the screen
Output conditions	The user is logged
Exceptions	The user hasn't go an associated Facebook account: 3.a. The system shows an error message on the screen Then the flow restars from point 1

User login with Google+

Name	Login with Google+
Actors	User
Goals	G.0.4
Input conditions	The user must be registered and have opened the appliation on the login screen
Events flow	1. The user taps *Login with Google+* 2. The system verifies the account 3. The system shows a confirm message on the screen
Output conditions	The user is logged
Exceptions	The user hasn't go an associated Google+ account: 3.a. The system shows an error message on the screen Then the flow restars from point 1

View scheduled events

Name	View scheduled events
Actors	Travlender
Goals	G.1.11
Input conditions	The user must be logged
Events flow	1. The Travlender swipes the screen on the left and a menu appears 2. The *Travlender* taps Scheduled events
Output conditions	The *Travlender* can view the scheduled events
Exceptions	There are no exception cases

Create a new standard event

Name	Create standard event
Actors	Travlender
Goals	G.1.1 G.1.9
Input conditions	The user must be logged
Events flow	1. The *Travlender* taps the *+* button 2. The *Travlender* taps the *Standard event* category 3. The *Travlender* inserts the name 4. The *Travlender* inserts starting time and ending time 5. The *Travlender* inserts a description (optional) 6. The *Travlender* modifies the constraints (optional) 7. The *Travlender* adds a location 8. The *Travlender* taps *Save*
Output conditions	A new standard event is created
Exceptions	The *Travlender* inserts a time slot overlapped with another standard event: 5.a. The system shows an error message on the screen Then the flow restars from point 4
	The *Travlender* inserts a non-existent location: 8.b. The system shows an error message on the screen Then the flow restars from point 7

Create a new lasting event

Name	Create lasting event
Actors	Travlender
Goals	G.1.1 G.1.9 G.7.2
Input conditions	The user must be logged
Events flow	1. The *Travlender* taps the + button 2. The *Travlender* taps the *Lasting event* category 3. The *Travlender* inserts the name 4. The *Travlender* inserts starting time and ending time 5. The *Travlender* inserts a description (optional) 6. The *Travlender* modifies the constraints (optional) 7. The *Travlender* adds a location 8. The *Travlender* taps *Save*
Output conditions	A new lasting event is created
Exceptions	The *Travlender* inserts a non-existent location: 8.a The system shows an error message on the screen Then the flow restars from point 7

Create a new flexible event

Name	Create flexible event
Actors	Travlender
Goals	G.1.1 G.1.9 G.7.1
Input conditions	The user must be logged
Events flow	1. The *Travlender* taps the + button 2. The *Travlender* taps the *Flexible event* category 3. The *Travlender* inserts the name 4. The *Travlender* inserts starting time and ending time 5. The *Travlender* inserts a minimum time slot 6. The *Travlender* inserts a description (optional) 7. The *Travlender* modifies the constraints (optional) 8. The *Travlender* adds a location 9. The *Travlender* taps *Save*
Output conditions	A new flexible event is created
Exceptions	The *Travlender* inserts a time slot bigger than the event duration: 6.a The system shows an error message on the screen Then the flow restars from point 5
	The *Travlender* inserts a non-existent location: 9.b The system shows an error message on the screen Then the flow restars from point 8

Create a new transfer event

Name	Create transfer event
Actors	Travlender
Goals	G.1.1 G.7.3
Input conditions	The user must be logged
Events flow	1. The *Travlender* taps the + button 2. The *Travlender* taps the *Transfer event* category 3. The *Travlender* inserts the name 4. The *Travlender* inserts starting time and ending time 5. The *Travlender* inserts a description (optional) 6. The *Travlender* adds a location 7. The *Travlender* taps Save
Output conditions	A new transfer event is created
Exceptions	The *Travlender* inserts a non-existent location: 7.a The system shows an error message on the screen Then the flow restars from point 6

Delete an existent event

Name	Delete event
Actors	Travlender
Goals	G.1.10
Input conditions	The user must be logged and must exist at least an event
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Scheduled events* 3. The *Travlender* taps the event he wants to delete 4. The *Travlender* taps *Delete*
Output conditions	A new transfer event is created
Exceptions	There are no exception cases

Modify an existent event

Name	Modify event
Actors	Travlender
Goals	[G.1.2 G.1.3 G.1.4 G.1.5 G.1.6 G.1.7 G.1.8
Input conditions	The user must be logged and at least an event must exist
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Scheduled events* 3. The *Travlender* taps the event he wants to modify 4. The *Travlender* modifies the desired fields 5. The *Travlender* taps *Save*
Output conditions	A new transfer event is created
Exceptions	The *Travlender* inserts a non-valid field: 5.a The system shows an error message on the screen Then the flow restars from point 4

Start travel of an existing event

Name	Start travel
Actors	Travlender
Goals	G.8.1 G.8.2
Input conditions	The user must be logged and at least an event must exist
Events flow	1. The system warns the *Travlender* that an event is about to start 2. The *Travlender* taps *Go*
Output conditions	The event starts
Exceptions	The *Travlender* taps Go when he can no longer reach the event before the start, but he can still reach it before the end: 2.a. The system warns the *Travlender* he/she is late Then the flow restarts from point 2
	The *Travlender* taps *Go* when he can no longer reach the event before the end: 2.b The system warns the *Travlender* he/she can no longer reach the event before the end Then the flow ends

Buy a ticket or book a mean

Name	Buy
Actors	Travlender
Goals	G.4.1 G.4.2 G.5.1 G.5.2 G.5.3 G.5.4 G.5.5 G.6.1 G.6.2
Input conditions	The user must be logged and the event must be started
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Scheduled events* 3. The *Travlender* taps the ongoing event 4. The *Travlender* taps the public mean he/she wants to buy a ticket or the mean he/she wants to book 5. The *Travlender* taps *Buy/Book*
Output conditions	The *Travlender* buys the ticket or books the mean
Exceptions	There are no exception cases

Select own means

Name	Select own means
Actors	Travlender
Goals	G.3.2
Input conditions	The user must be logged
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Personalize* 3. The *Travlender* taps *Own means* 4. The *Travlender* selects his/her own means from a list 5. The *Travlender* taps *Save*
Output conditions	*Travlendar+* knows the user own means
Exceptions	There are no exception cases

Define selectable transports

Name	Define selectable transports
Actors	Travlender
Goals	G.2.8
Input conditions	The user must be logged
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Personalize* 3. The *Travlender* taps *Selectables* 4. The *Travlender* selects the means the app has to consider in his/her travels 5. The *Travlender* taps *Save*
Output conditions	*Travlendar+* knows the means to consider in travels
Exceptions	There are no exception cases

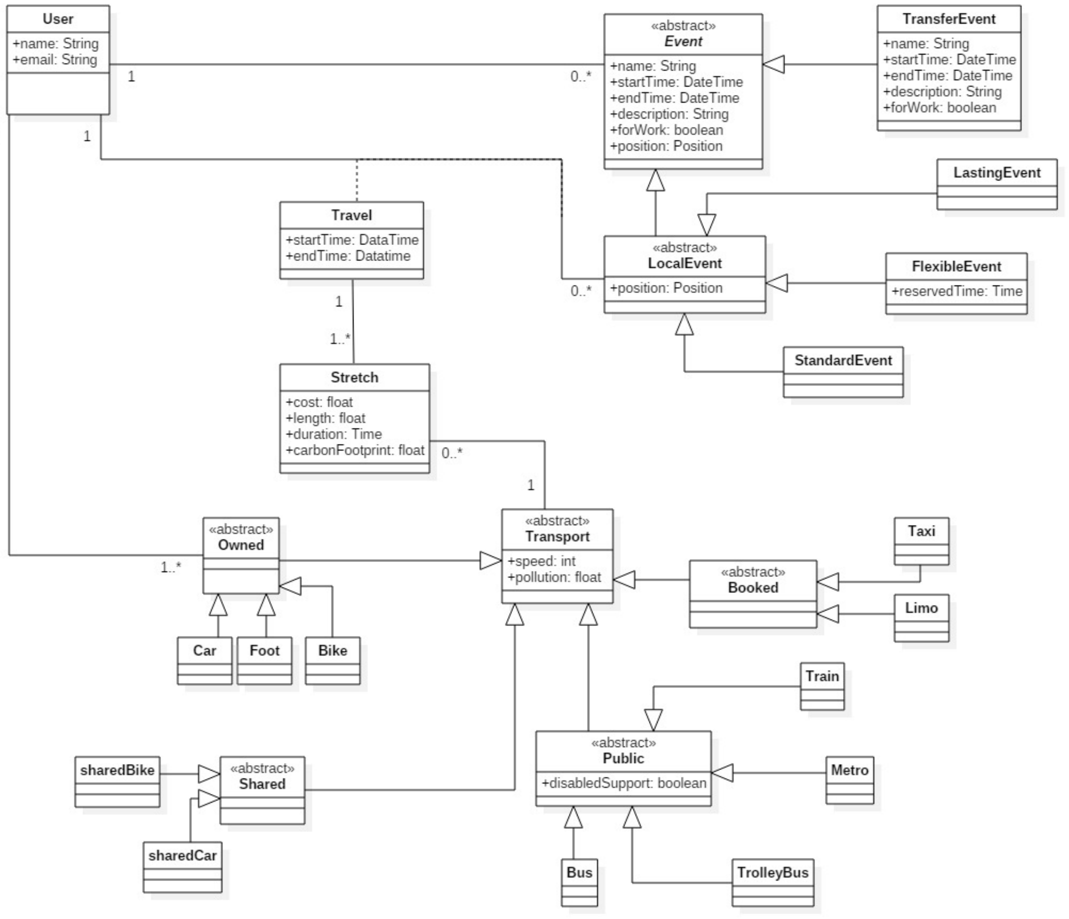
Specify transports constraints

Name	Specify transports constraints
Actors	Travlender
Goals	G.2.1 G.2.2 G.2.3 G.2.4 G.2.5 G.2.6 G.2.7 G.3.3
Input conditions	The user must be logged
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Personalize* 3. The *Travlender* taps *Constraints* 4. The *Travlender* taps the vehicle desired from a list 5. The *Travlender* inserts the constraints in the field 6. The *Travlender* taps *Save*
Output conditions	*Travlendar+* knows the means constraints to consider in travels
Exceptions	There are no exception cases

Specify own public vehicle passes

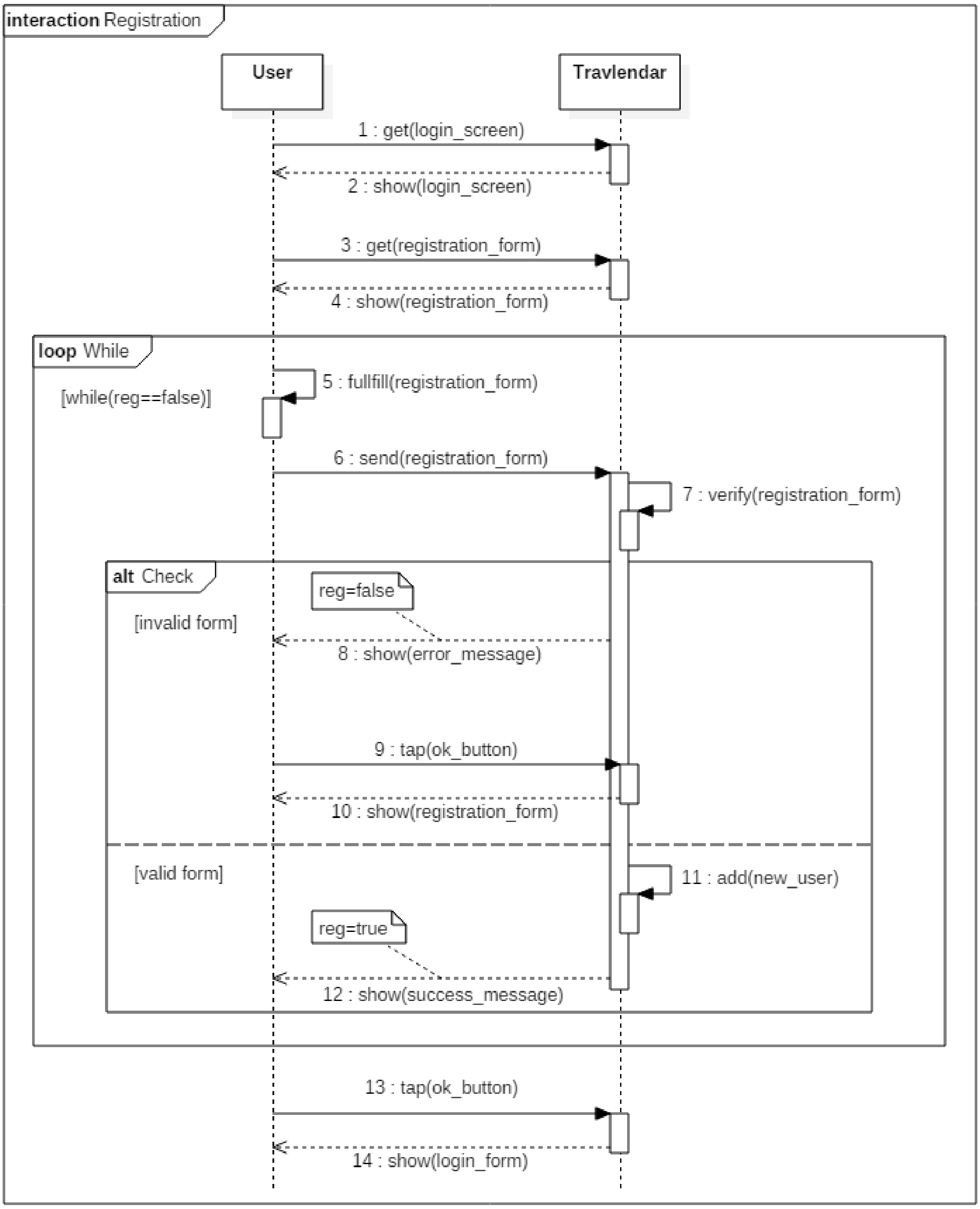
Name	Specify own passes
Actors	Travlender
Goals	G.3.1
Input conditions	The user must be logged
Events flow	1. The *Travlender* swipes the screen on the left and a menu appears 2. The *Travlender* taps *Personalize* 3. The *Travlender* taps *Passes* 4. The *Travlender* select the passess he has from a list 5. The *Travlender* taps *Save*
Output conditions	*Travlendar+* knows which passes the user has
Exceptions	There are no exception cases

3.2.3 Class diagram

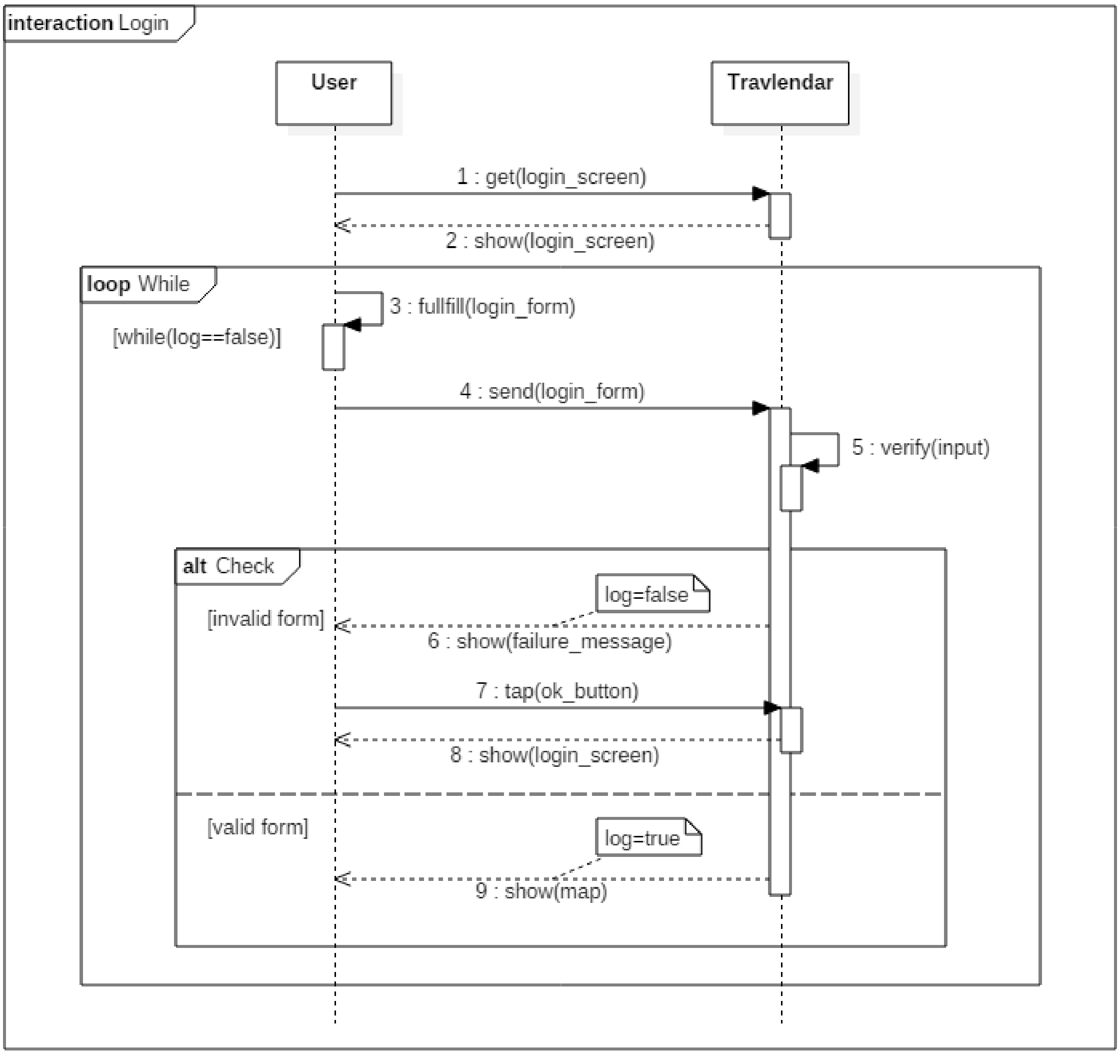


3.2.4 Sequence diagrams

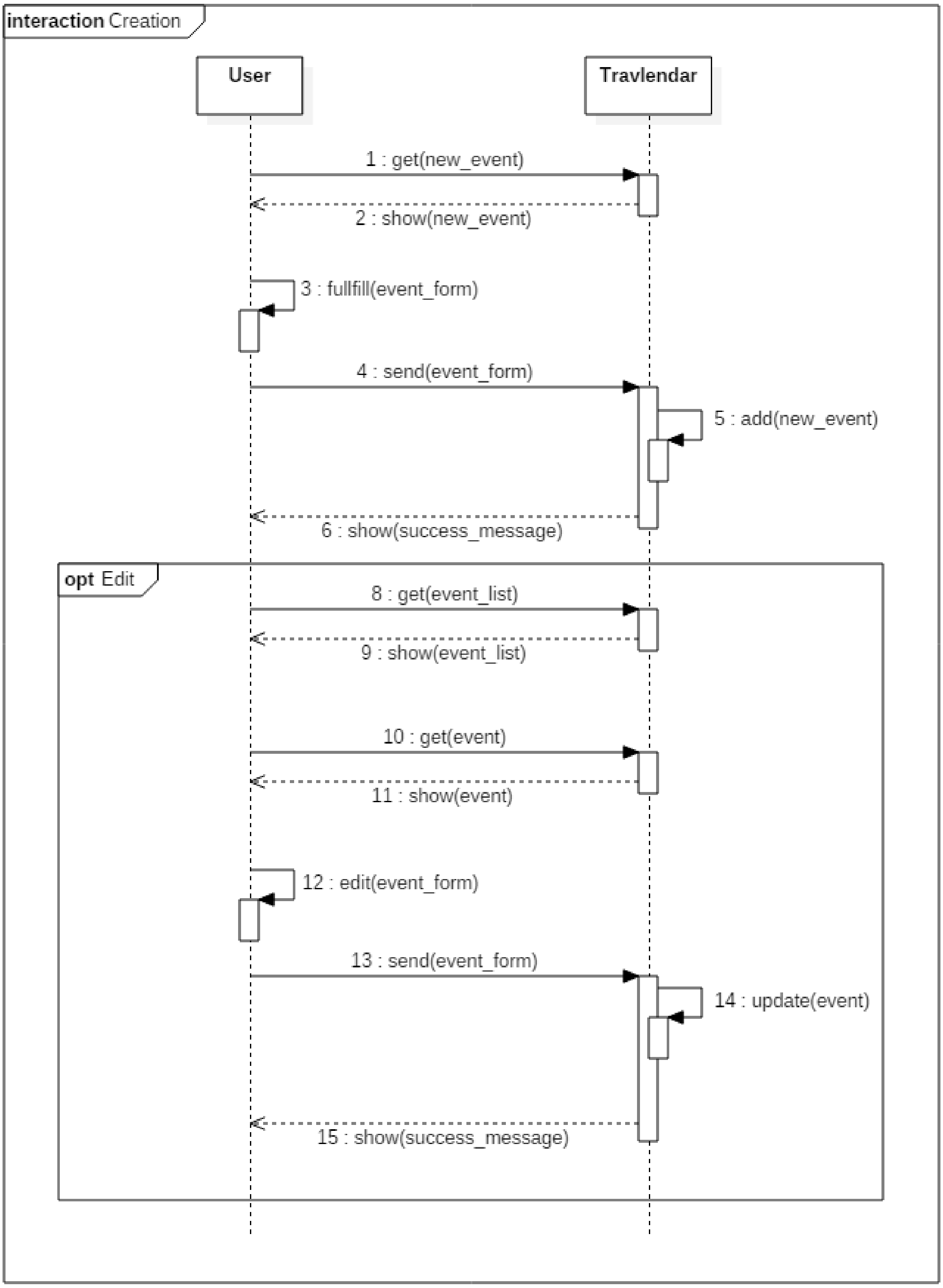
User registration



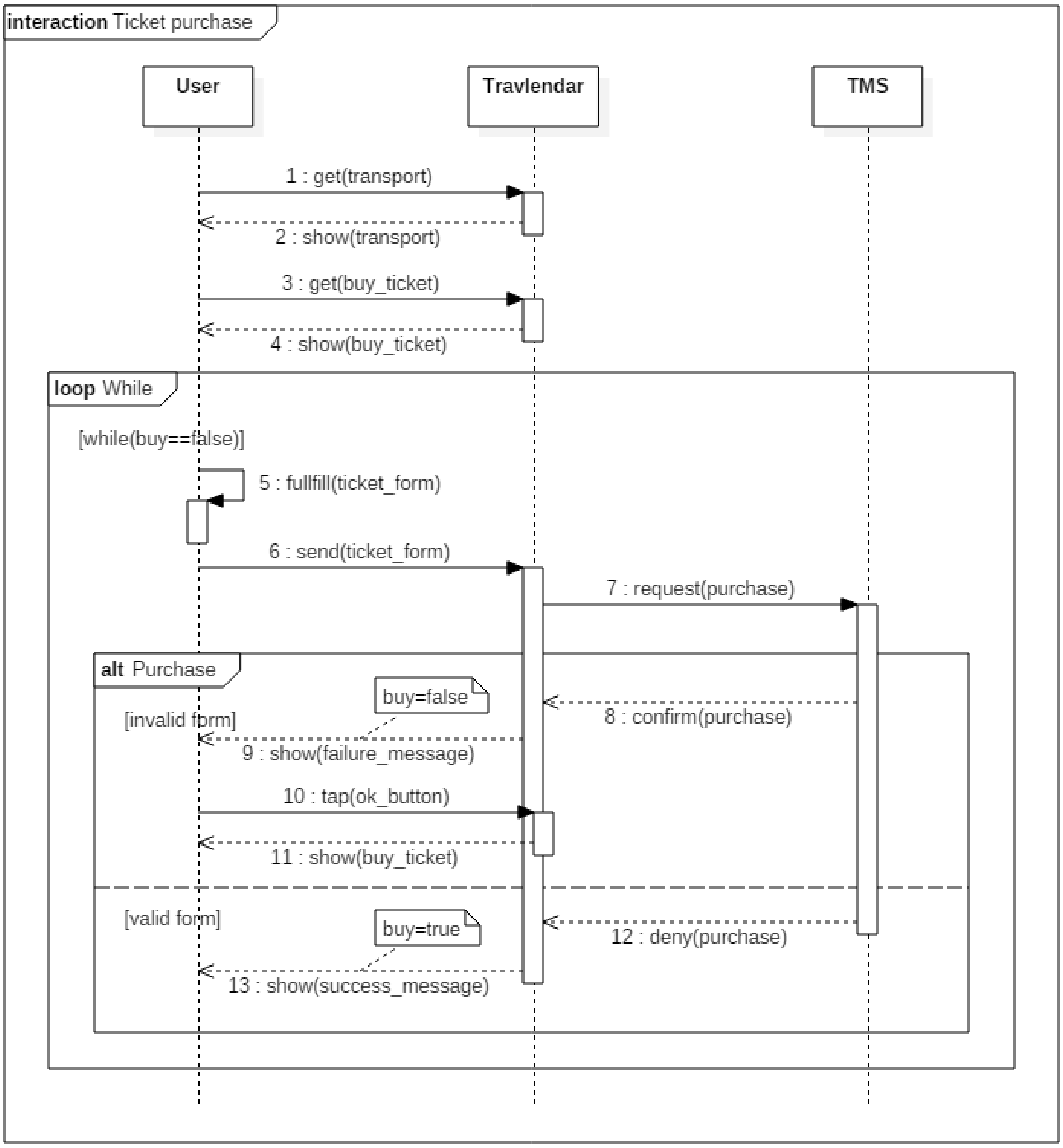
User login


interaction Login
User
Travlendar
1 : get(login_screen)
2 : show(login_screen)
loop While
[while(log==false)]
3 : fullfill(login_form)
4 : send(login_form)
5 : verify(input)
alt Check
[invalid form]
log=false
6 : show(failure_message)
7 : tap(ok_button)
8 : show(login_screen)
[valid form]
log=true
9 : show(map)

Event creation and edit



Ticket purchase



Section [R.0] treats all the requimerements related to the registration to the application:

R.0.1 The system must provide a registration form to the user.

R.0.2 The system must verify whether the user filled all the required data.

R.0.3 The system must check whether the user results already registered

Section [R.1] treats all the requimerements related to the login:

R.1.1 The system must check the input data and control if they correspond to an existing account.

R.1.2 The system must guarantee the user login if and only if the inserted credentials result verified.

Section [R.2] treats all the requimerements related to the creation or modification of events:

R.2.1 The system must verify that the user is available for the duration of the event.

R.2.2 The system must verify that the free time preceeding the event is enough to travel to the new location.

R.2.3 The system must control whether it is possible for the user to leave from the previous event location to arrive on time.

R.2.4 The system must control whether it is possible for the user to leave the event in order to arrive on time at the following event location.

R.2.5 The system must control whether, since the current position is given, the user is able to arrive at the event location on time.

R.2.6 The system must inform the user when it is not possible to schedule his/her event in the allotted time.

R.2.7 The system must suggest possible solutions to arrange conflicting events, either postponing the starting time of the next event or anticipating the ending time of the previous one.

Section [R.3] treats all the requimerements related to the shared and non-shared means:

R.3.1 The system must be able to locate the nearest bike to bike sharing system in the city.

R.3.2 The system must be able to locate the nearest car sharing system in the city.

R.3.3 The system must be able to locate the public transports areas in the city.

Section [R.4] treats all the requimerements related to the travel:

R.4.1 The system must control if the (eventually) required tickets are already owned by the user.

R.4.2 The system must notify the user in occurrence of events scheduled on days of adverse weather conditions.

R.4.3 The system must establish the best route plan for the user.

R.4.4 The system must not suggest the user to use owned car or bike in his travels from locations where those are not placed.

Section [R.5] treats all the requimerements related to the special events:

R.5.1 When the user requires to start to travel from a lasting event the system must control whether it is possible for him/her to arrive at the event location and reach the next one on time.

3.3 Design constrains

3.3.1 Hardware limitations

1. An Android (7.1.1 or above) is recommended.
2. GPS has to be always aviable.
3. The system requires at least a 3G wireless technology.
4. A minimum space for the application download is required.

3.4 Software system attributes

3.4.1 Reliability

The MMS has to offer a precise GPS service.

3.4.2 Availability

Travlendar+ is designed to be a 24/7 service.

3.4.3 Security

The TMS has to guarantee a secure service.

3.4.4 Maintainability

Short maintenance periods are allowed.

4 | Scenarios

4.1 Scenario 1

Registration and calendar function
Tom is a student who recently moved to Milan for his studies. He is a messy boy and always had his events scattared on post-its all over the house. This condition often made him miss appointments. His roommate, tired of the disorder presented to him Travlendar+, a new born application suitable for situations as Tom's. Tom downloaded the application and after a quick registration via facebook he transferred all the events he had hanging around in his Tlavlendar+ app. He was alerted when events where set in an overlapping moment and always notified in time for his events. In few days he gladly noticed how having all his events always under control brought an improvement in his daily routine organization. Even Carlo was happy, he could finally throw away all the post-it notes.

4.2 Scenario 2

Flexible event
Lucy is a mother who enjoys cooking but she lately spends much time complaining about her old pots. Since her birthday is close the son Mark decided to buy her a new set of pots. He recently started working, and with his other daily commitments he knows he does not have much free time. He realizes that his only chance may be on Wednesday, when there is the open market in the next street therefore he opens Travlendar+ and try to fit the new commitment in his day schedule. Knowing that the market starts at 8.00 a.m. and lasts until 5.00 p.m. he schedules a new flexible event in the mentioned hours and sets an occupation time of thirty minutes, time he will need to use to find the best pots for his mum. On wednesday the idea of the present was already out of his mind when, after a busy morning Travlendar+ notified him that he had the free time to spend at the market. Mark went to buy the pots and after giving them to the mother Lucy was so happy she decided to inaugurate them with a cake.

4.3 Scenario 3

Transfer event, preferences edit for single travel
George was invited by his friend Lucas to have lunch together at a restaurant in Monza. George opened Travlendar+ and exploring the map he found the restaurant and added his commitment from the map screen. The new event was automatically scheduled as a transfer event because Travlendar+ recognized the location as outside the Milan city. George was therefore warned that since it is a transfer Travlendar+ would not give full support for the travel: it wouldn't be able to suggest any route for the destination and during all the time spent outside the city it wouldn't manage to compute any travel time for future appointments. George was a bit worried, since he had had the application he always relied to it completely but fortunately he is a long-standing client of Travlendar+ so he knows all the various features of the application well so before leaving he controlled his next appointment: a dinner, that same day at his grandma's place in Milan. He estimates that with the car the expected travel time from the restaurant to the grandma's house is of one hour so he just opens the event set at grandma's and modify the event's preferences

setting a one hour reminder. Now he can leave the city without worries, aware that the everning Travlendar+ will notify him in time.

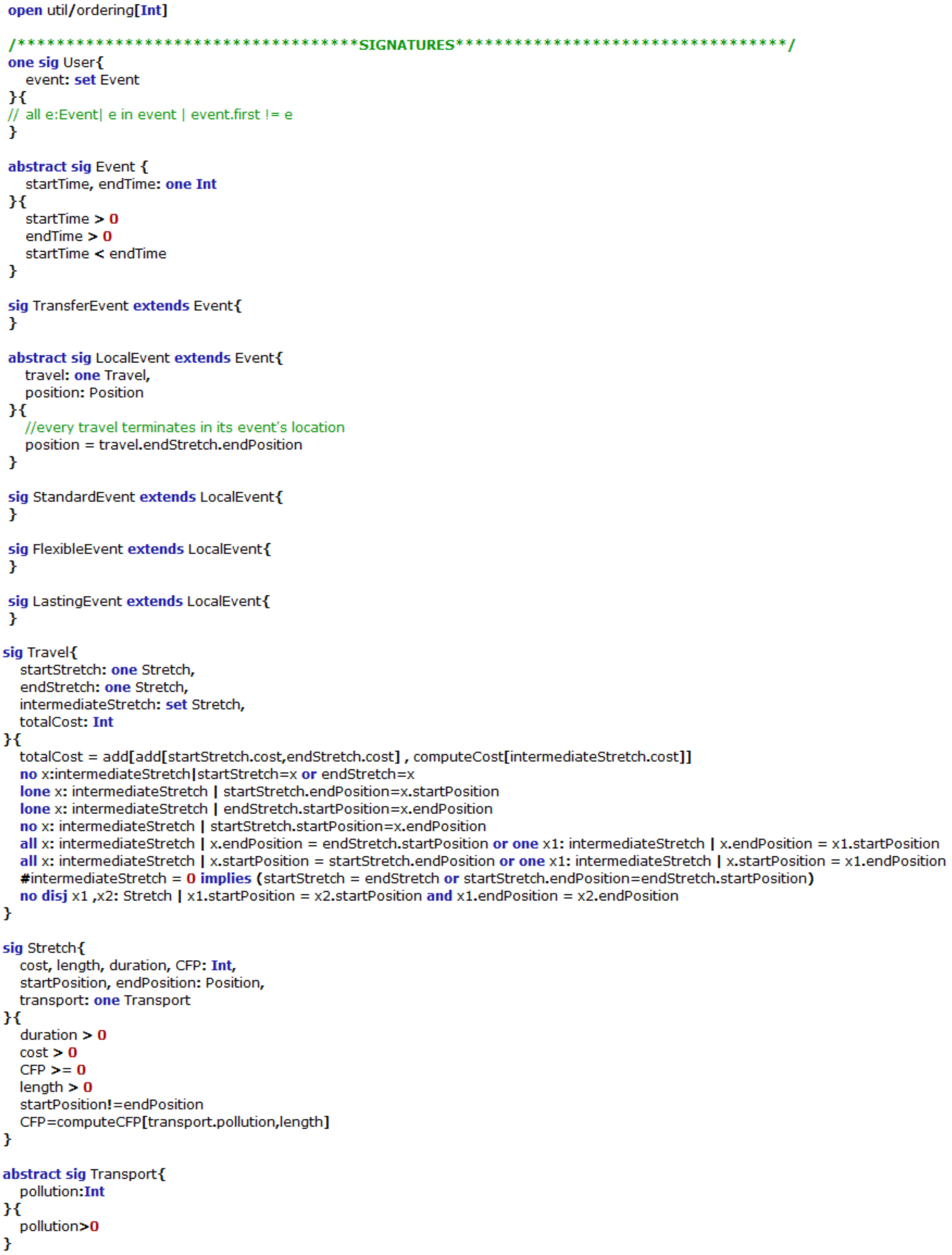
4.4 Scenario 4

Lasting event, start trip button, event editing
Every year in Milan there is an electronics fair where you can find all kind of electronics products and services. Mike is passionate about electronics so he would happily attended some stands if he finds the occasion and this year may be the right one since the expo has the special duration of one week. Mike schedules the occation as a lasting event indicating the usual informations where he specifies the days of opening and closing of the fair. The day the fair started a notification popped up on Mike's phone to reminding him of the event. Mike was excited because he discovered that there was going to be a presentation about virtual reality he wanted to attend so as soon as he finished his work shift he decides to move to the fair. He opens the event and taps the start trip button. Unfortunately the application warns him that it is not possible to start the trip because the time needed to get there won't allow him to arrive on time at the next event. Indeed Mike was forgetting that that night he had appointment at the theatre with his friend Carl. After a moment of reflection he decides to give the priority to the fair so he calls his friend to arrange a new appointment. They confirmed the decision of a theatre but had to change the theatre because the former one didn't have interesting shows that day. Mike modifies the theatre event, changes the time and place of the meeting and adds a description as a reminder not to miss the appointment. After the confirmation of edit Mike could start his journey to the electronic fair. The next day, in the morning Mike sees a low-priority notification that informs him there are ongoing events: it's still the electronic fair. This time he just ignores it because he cannot disappoint his friend.

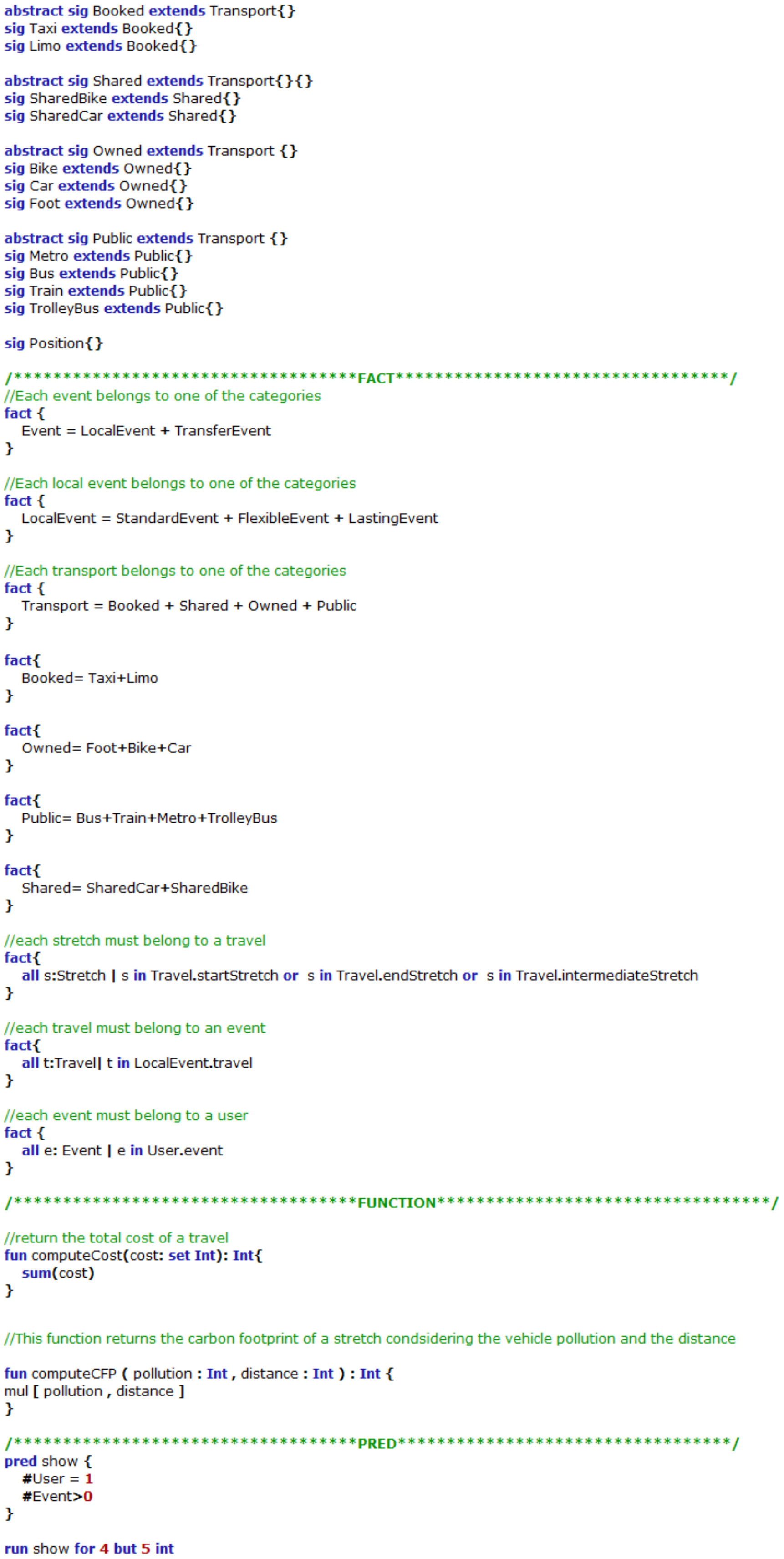
4.5 Scenario 5

Multi transport function, preferences customization, ticket purchase, travelling constrains
Carl is an environmental advocate, and has an important conference about the sustainable development. During his speech he had an excursus about how the Travlendar+ application helped him get to the converence location that same night. He talked about how he valued the possibility that the application gave him that morning in letting him choose the most echological route. He took a 1 hour long trip, as suggested by Travlendar+. As he confirmed to follow the suggested route Carl is notified that a metro ticket is needed for the travel. He is unlucky because if the transporto would have been a train tickets were not to be a problem since he has, and added in his Travlendar+ preferences a train subscription, valid a whole year. Still there is no big deal because Travlendar+ offers the oppurtunity to buy the tickets in-app and so he does, in just a few seconds thanks to some valid payment services used by the application he receives a mail notifying the success of the operation. He then starts the journey on his bike, and after having pedaled for 10 minutes he gets to the nearest tram stop. He got to the terminus of the tram after 30 minutes, and while returning on his bike for the last stretch, as the app suggested he saw another tram, leaving that same stop with the conference venue displayed as destination. He was not upset though, because he does not like to take trams at that time of the night because of bad experiences he had. Only later he remembered that he himself was the one that imposed the time constrain on transport.

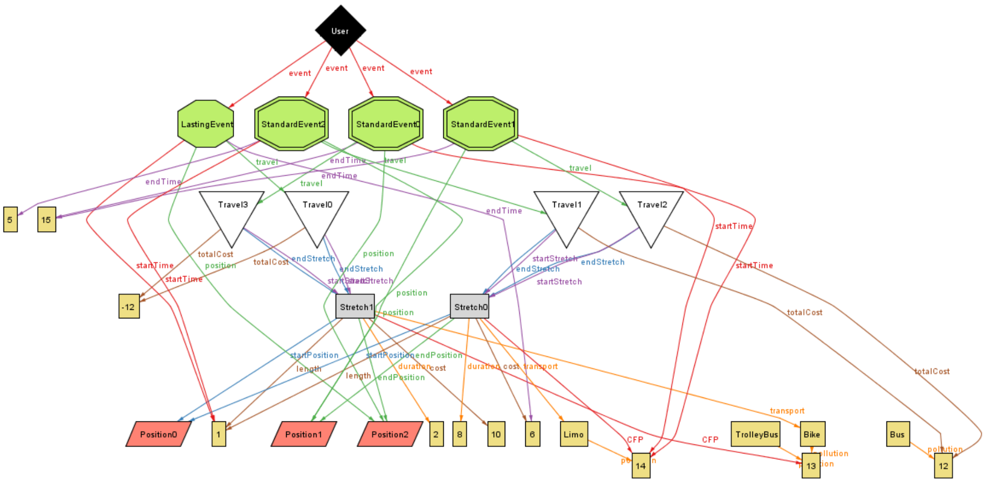
5 | Formal analysis using alloy



```
open util/ordering[Int]

/**********************************SIGNATURES*********************************/
one sig User{
   event: set Event
}{
// all e:Event| e in event | event.first != e
}

abstract sig Event {
   startTime, endTime: one Int
}{
   startTime > 0
   endTime > 0
   startTime < endTime
}

sig TransferEvent extends Event{
}

abstract sig LocalEvent extends Event{
   travel: one Travel,
   position: Position
}{
   //every travel terminates in its event's location
   position = travel.endStretch.endPosition
}

sig StandardEvent extends LocalEvent{
}

sig FlexibleEvent extends LocalEvent{
}

sig LastingEvent extends LocalEvent{
}

sig Travel{
   startStretch: one Stretch,
   endStretch: one Stretch,
   intermediateStretch: set Stretch,
   totalCost: Int
}{
   totalCost = add[add[startStretch.cost,endStretch.cost] , computeCost[intermediateStretch.cost]]
   no x:intermediateStretch|startStretch=x or endStretch=x
   lone x: intermediateStretch | startStretch.endPosition=x.startPosition
   lone x: intermediateStretch | endStretch.startPosition=x.endPosition
   no x: intermediateStretch | startStretch.startPosition=x.endPosition
   all x: intermediateStretch | x.endPosition = endStretch.startPosition or one x1: intermediateStretch | x.endPosition = x1.startPosition
   all x: intermediateStretch | x.startPosition = startStretch.endPosition or one x1: intermediateStretch | x.startPosition = x1.endPosition
   #intermediateStretch = 0 implies (startStretch = endStretch or startStretch.endPosition=endStretch.startPosition)
   no disj x1 ,x2: Stretch | x1.startPosition = x2.startPosition and x1.endPosition = x2.endPosition
}

sig Stretch{
   cost, length, duration, CFP: Int,
   startPosition, endPosition: Position,
   transport: one Transport
}{
   duration > 0
   cost > 0
   CFP >= 0
   length > 0
   startPosition!=endPosition
   CFP=computeCFP[transport.pollution,length]
}

abstract sig Transport{
   pollution:Int
}{
   pollution>0
}
```



```
abstract sig Booked extends Transport{}
sig Taxi extends Booked{}
sig Limo extends Booked{}

abstract sig Shared extends Transport{}{}
sig SharedBike extends Shared{}
sig SharedCar extends Shared{}

abstract sig Owned extends Transport {}
sig Bike extends Owned{}
sig Car extends Owned{}
sig Foot extends Owned{}

abstract sig Public extends Transport {}
sig Metro extends Public{}
sig Bus extends Public{}
sig Train extends Public{}
sig TrolleyBus extends Public{}

sig Position{}

/************************************FACT***********************************/
//Each event belongs to one of the categories
fact {
   Event = LocalEvent + TransferEvent
}

//Each local event belongs to one of the categories
fact {
   LocalEvent = StandardEvent + FlexibleEvent + LastingEvent
}

//Each transport belongs to one of the categories
fact {
   Transport = Booked + Shared + Owned + Public
}

fact{
   Booked= Taxi+Limo
}

fact{
   Owned= Foot+Bike+Car
}

fact{
   Public= Bus+Train+Metro+TrolleyBus
}

fact{
   Shared= SharedCar+SharedBike
}

//each stretch must belong to a travel
fact{
   all s:Stretch | s in Travel.startStretch or  s in Travel.endStretch or  s in Travel.intermediateStretch
}

//each travel must belong to an event
fact{
   all t:Travel| t in LocalEvent.travel
}

//each event must belong to a user
fact {
   all e: Event | e in User.event
}

/************************************FUNCTION***********************************/

//return the total cost of a travel
fun computeCost(cost: set Int): Int{
   sum(cost)
}


//This function returns the carbon footprint of a stretch condsidering the vehicle pollution and the distance

fun computeCFP ( pollution : Int , distance : Int ) : Int {
mul [ pollution , distance ]
}

/************************************PRED***********************************/
pred show {
   #User = 1
   #Event>0
}

run show for 4 but 5 int
```


User
event
event
event
event
LastingEvent
StandardEvent2
StandardEvent0
StandardEvent1
travel
travel
travel
travel
endTime
endTime
endTime
endTime
Travel3
Travel0
Travel1
Travel2
5
15
-12
startTime
startTime
startTime
startTime
totalCost
totalCost
totalCost
totalCost
position
position
position
position
endStretch
endStretch
endStretch
endStretch
startStretch
startStretch
startStretch
startStretch
Stretch1
Stretch0
startPosition
startPosition
endPosition
endPosition
length
length
duration
duration
cost
cost
transport
transport
CFP
CFP
Position0
1
Position1
Position2
2
8
10
6
Limo
TrolleyBus
Bike
Bus
14
13
12
pollution
pollution
pollution

6 | Effort spent

14-oct	15:00-21:00 00:00-02:00	8 hours
15-oct	16:00-18:00 00:00-01:00	3 hours
16-oct	10:00-12:00 22:00-01:00	5 hours
17-oct	16:00-18:00 22:00-02:00	6 hours
18-oct	15:00-18:00 00:00-02:00	5 hours
19-oct	22:00-01:00	3 hours
20-oct	22:00-02:00	4 hours
21-oct	16:00-19:00 22:00-02:00	7 hours
22-oct	22:00-02:00	4 hours
23-oct	22:00-03:00	5 hours
24-oct	17:00-19:00 00:00-02:00	4 hours
25-oct	16:00-18:00 00:00-03:00	5 hours
26-oct	16:00-19:00 22:00-04:00	9 hours
27-oct	16:00-19:00 00:00-03:00	6 hours
28-oct	16:00-19:00 20:00-00:00	7 hours
29-oct	10:00-13:00 15:00-23:00	11 hours