POLITECNICO
MILANO 1863

MSc Computer Science
and Engineering

Software Engineering 2
ACADEMIC YEAR 2017-2018

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 29^{th} 2017
Version 1.0

Contents

1 Introduction

1.1 Purpose

Section [G.0] treats all the goals related to 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 early 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 scheduled before.

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 keeps 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 or shared transports per travel.

G.2.8 The *Travlender* can select the means he/she wants to use and deselect that 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 purchases 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 purchases 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 purchases 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 to reach the location of the events all across Milan, combining different sort of means in relation to the user preferences.

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

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

D.n.m = Domain assumption number m in section n.

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

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 purchases 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 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, dependencies and constraints

2.4.1 Assumptions

Application assumptions

A.1.1 In the minimization of carbon footprint all the public vehicles are considered like having zero-emission since the user's presence would not influence the travel mean's emission.

A.1.2 The car's trip cost is assumed to be the same for all cars, and only depends on the distance.

Domain assumptions

D.2.1 The payment credentials are verified by a reliable external service.

D.2.2 The traffic informations, coming from external services are reliable.

D.2.3 The gps positions are always accurate.

D.2.4 It is possible to keeps track of the position of your personal means through external service.

D.2.5 The cost for all car trips is given by the estimate of the kilometers per liter and euros per liter.

D.2.6 All the trams, buses, trolleybuses and metro are available for people with disabilities.

D.2.7 There are no unforseen events that can cause delay to the user (es. accidents).

3 | Specific requirements

3.1 External interface requirements

3.1.1 User interface

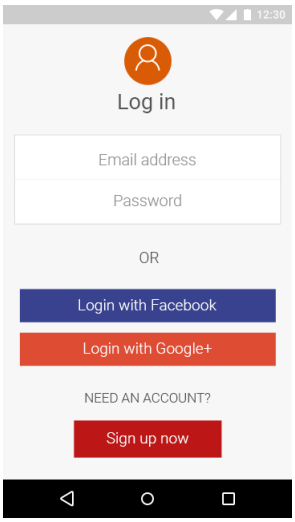


Figure 3.1: Login

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 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 will be not accessible.

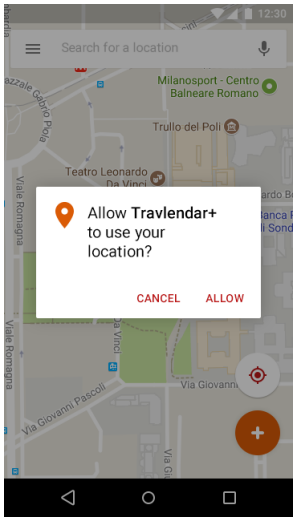


Figure 3.2: Alert

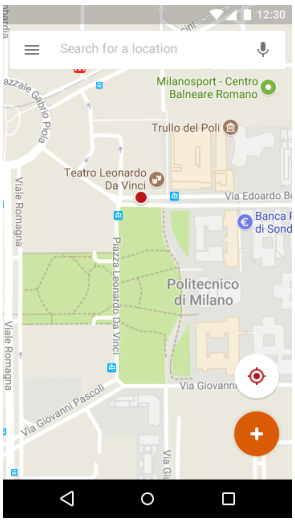


Figure 3.3: Map

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.

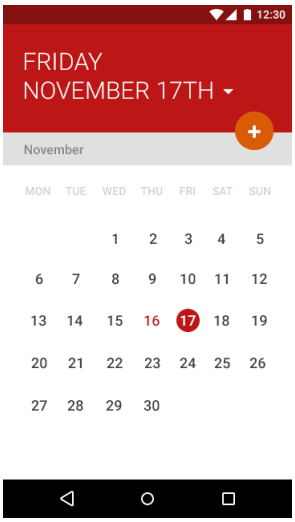
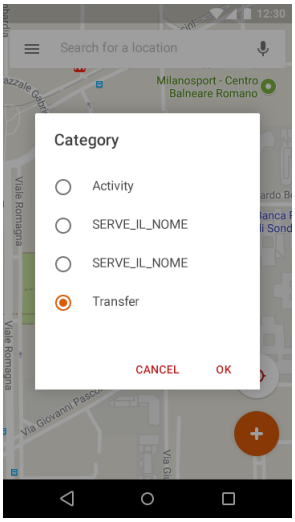


Figure 3.4: Calendar



AAA

Figure 3.5: Choice of event category

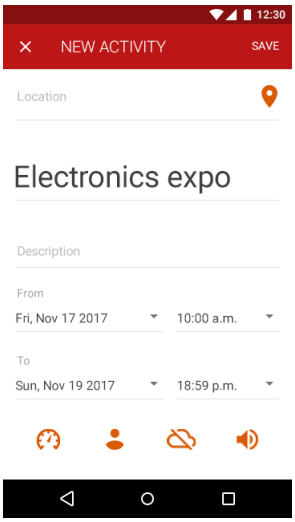
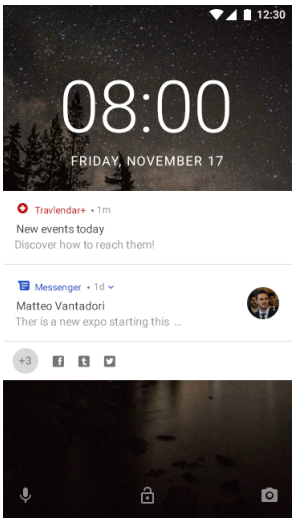


Figure 3.6: Creation of a new event

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 fourth 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 ables (or disables) the *not wet* travel. The last one ables (or disables) *notifications*.



AAA

Figure 3.7: Lock screen

AAA

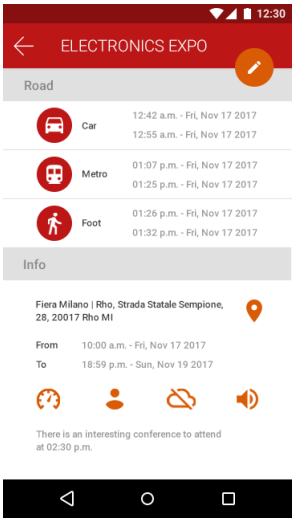


Figure 3.8: Summary of the event

3.1.2 Hardware interfaces

3.1.3 Software interfaces

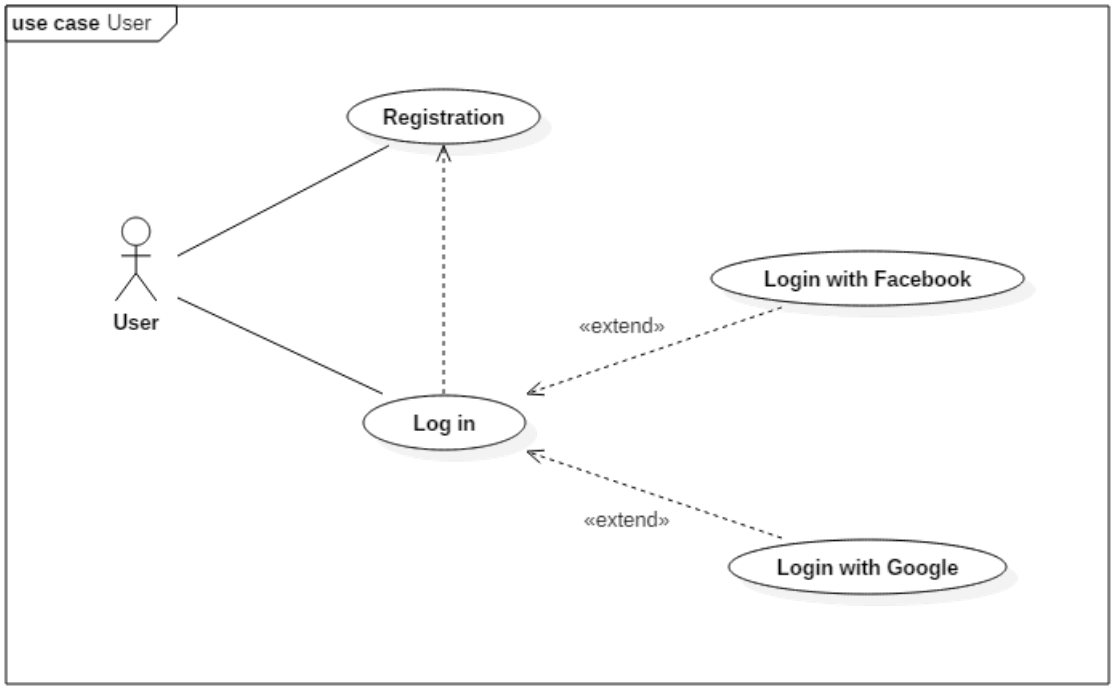
3.1.4 Communication interfaces

3.2 Functional requirements

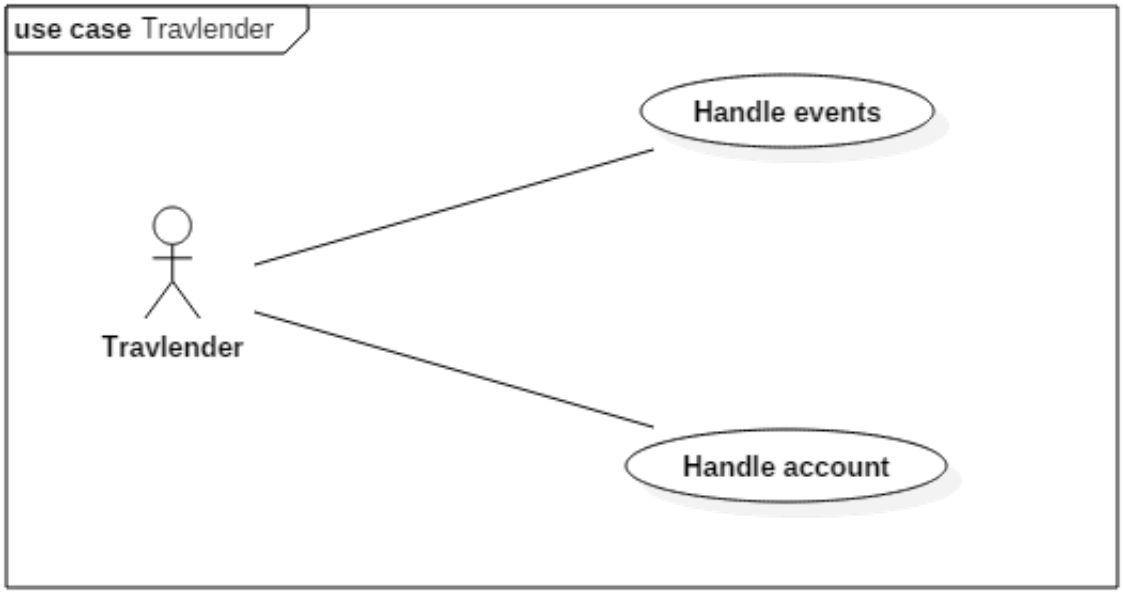
INSERIRE DEFINITION OF USE CASE DIAGRAMS!!!

3.2.1 Use case diagrams

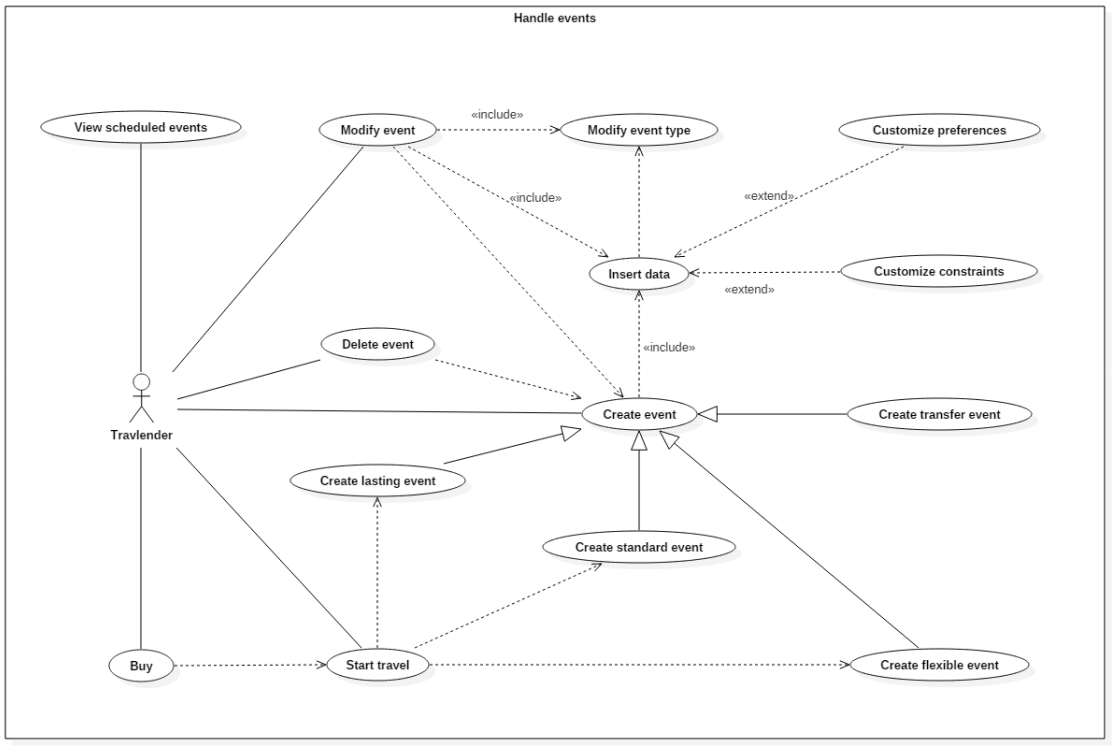
User



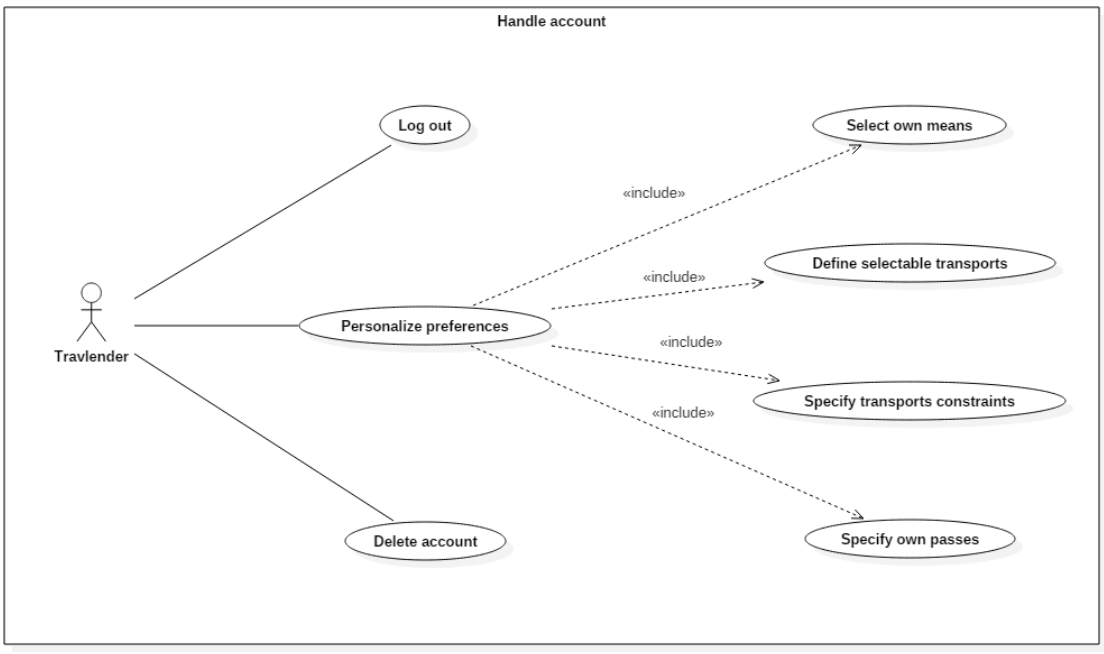
Travlender



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 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 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 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 Travlendar 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 Lasting 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 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 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 must exist at least an event
Events flow	1. The system warns the Travlender that an event is about to starts 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 reache 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

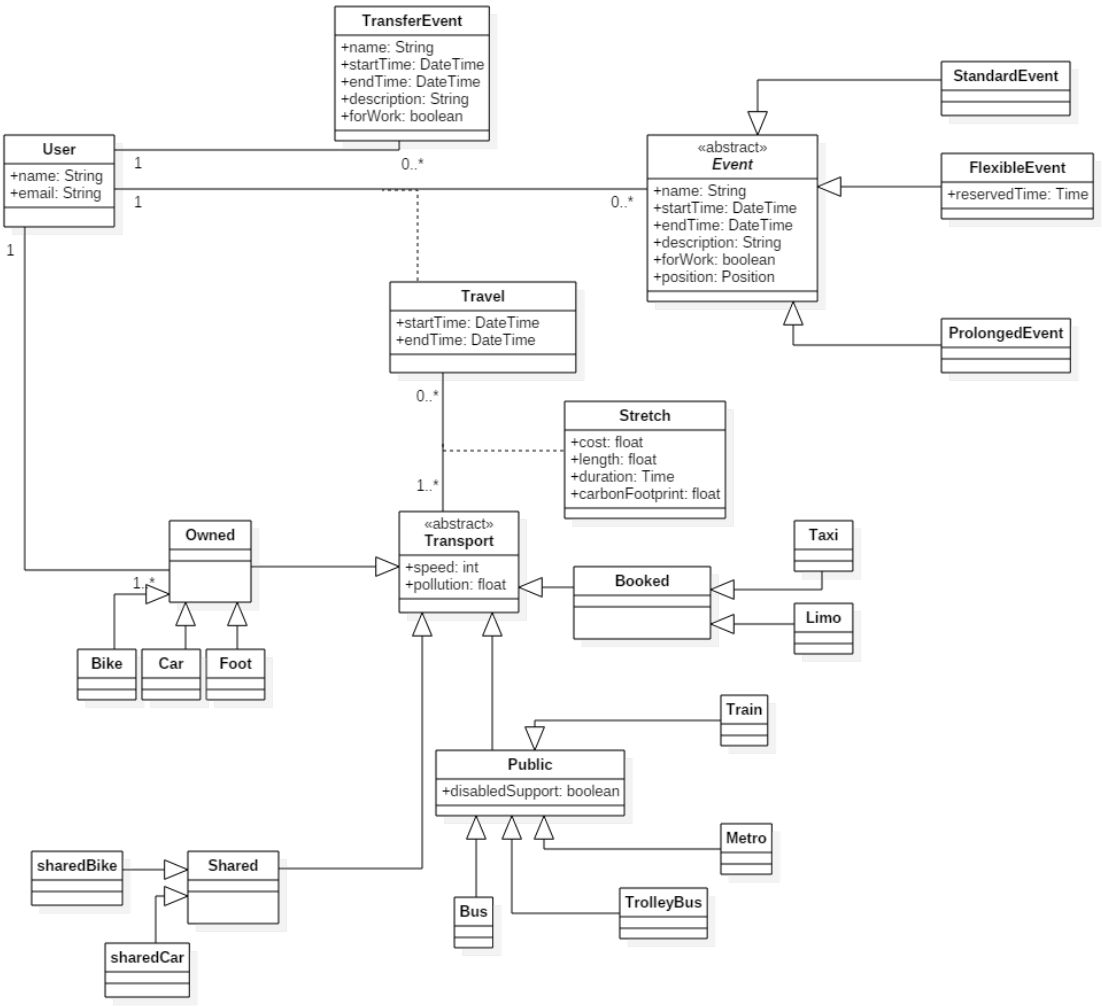
Specify transports constraints

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

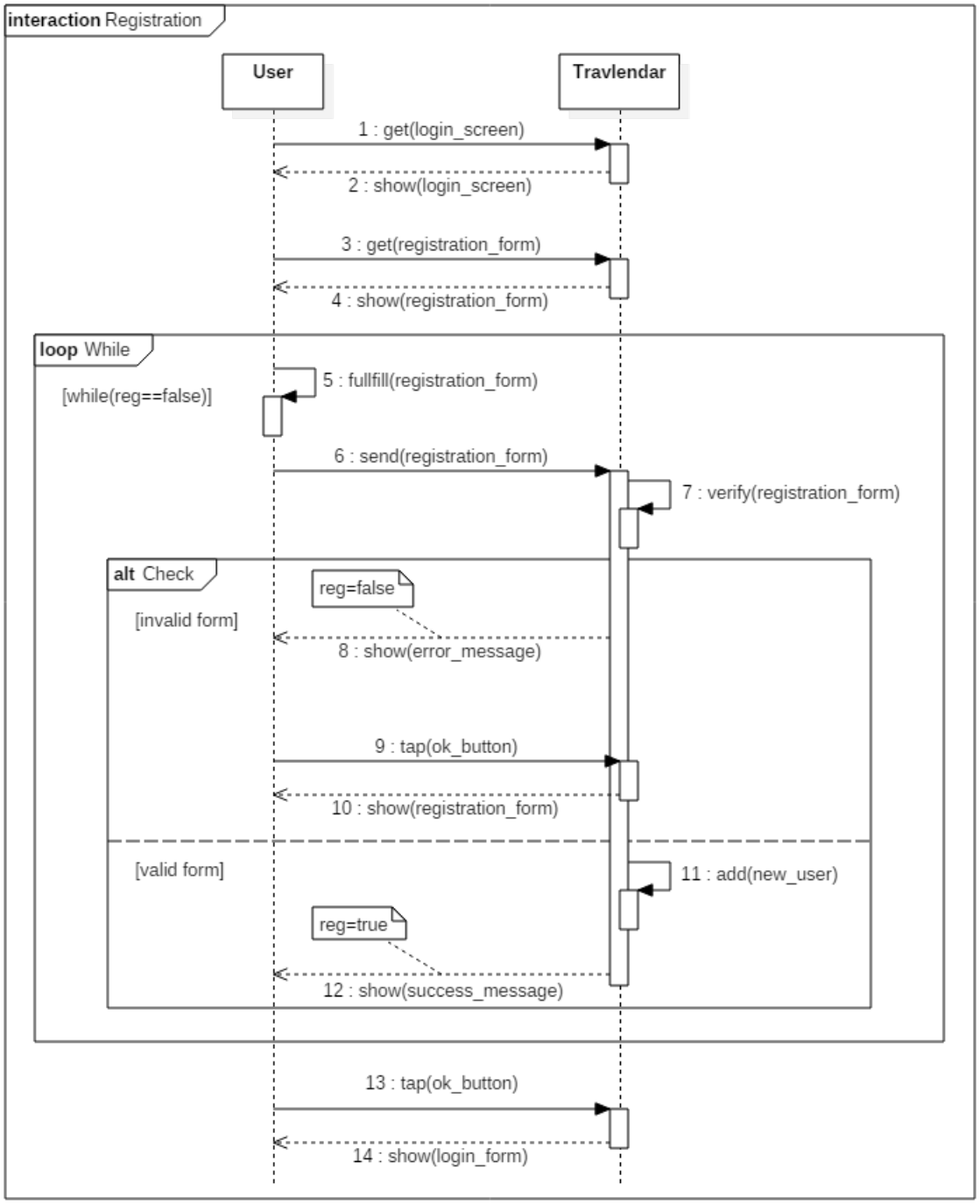
Name	Specify own passes
Actors	Travlender
Goals	G.3.1
Input conditions	The user must be logged
Events flow	1. The Travlender swipes the screen on the left and a menu appears 2. The Travlender taps Personalize 3. The Travlender taps Passes 4. The Travlender select the passess he has from a list 5. The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

3.2.3 Class diagram

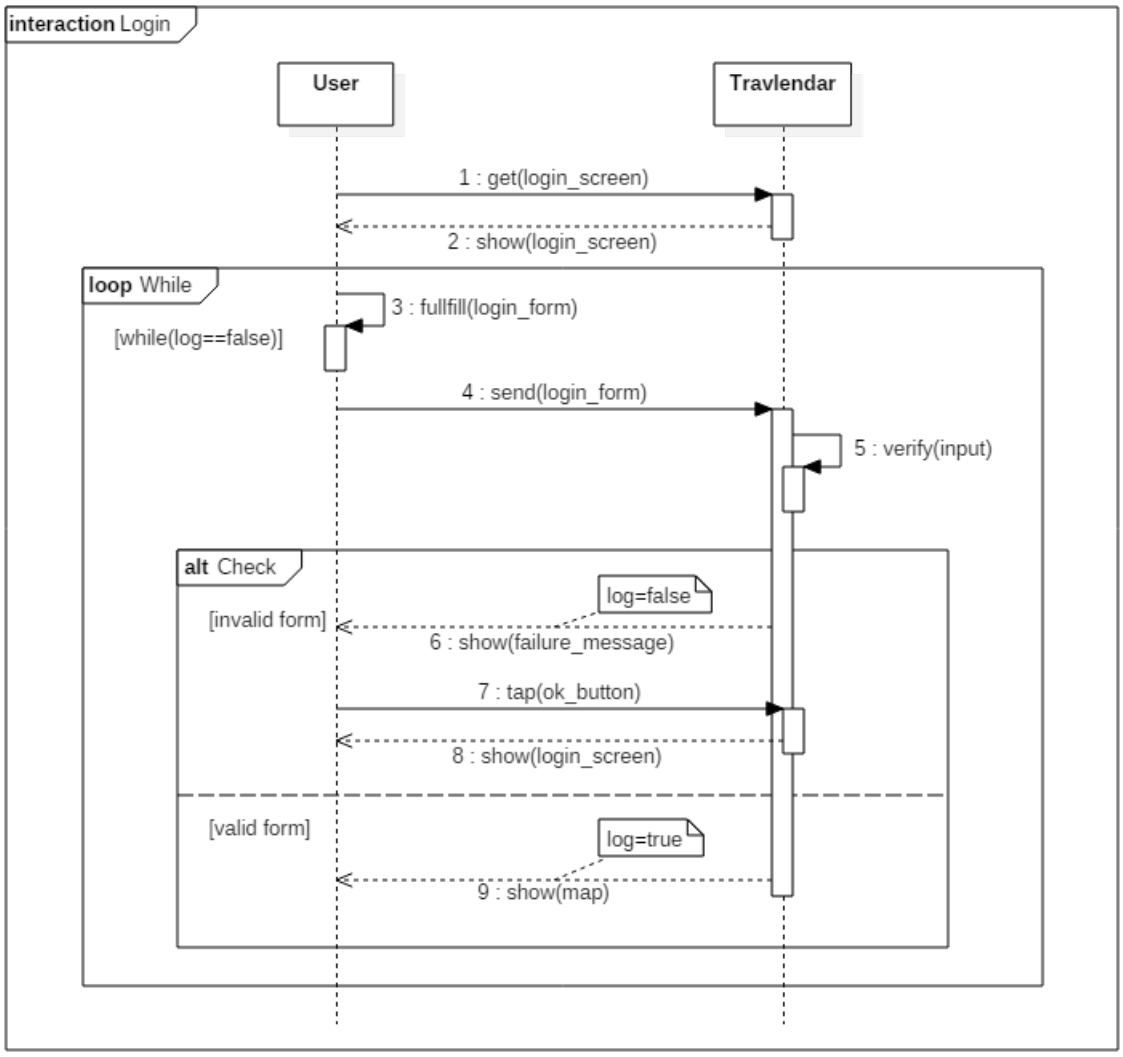


3.2.4 Sequence diagrams

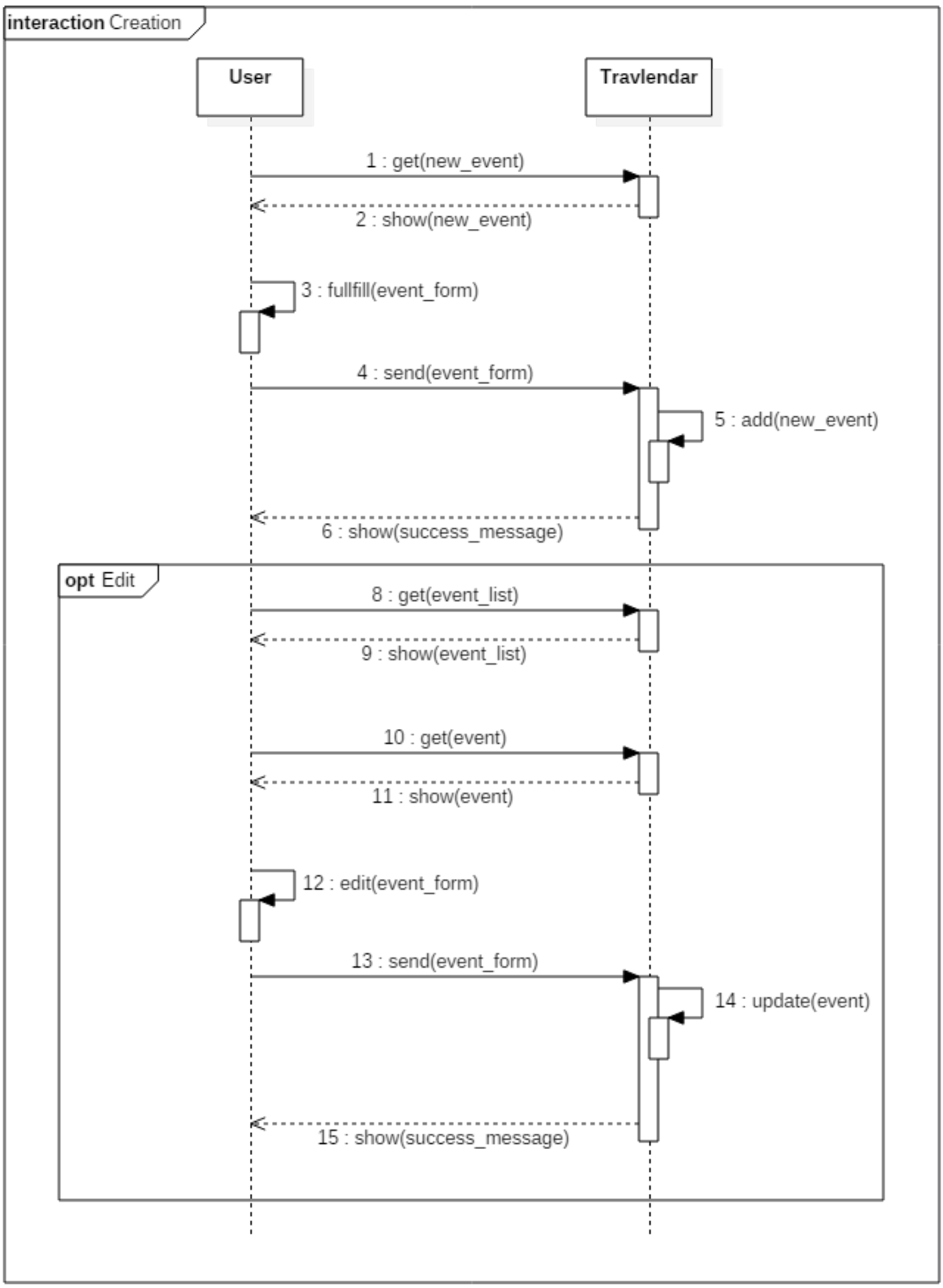
User registration



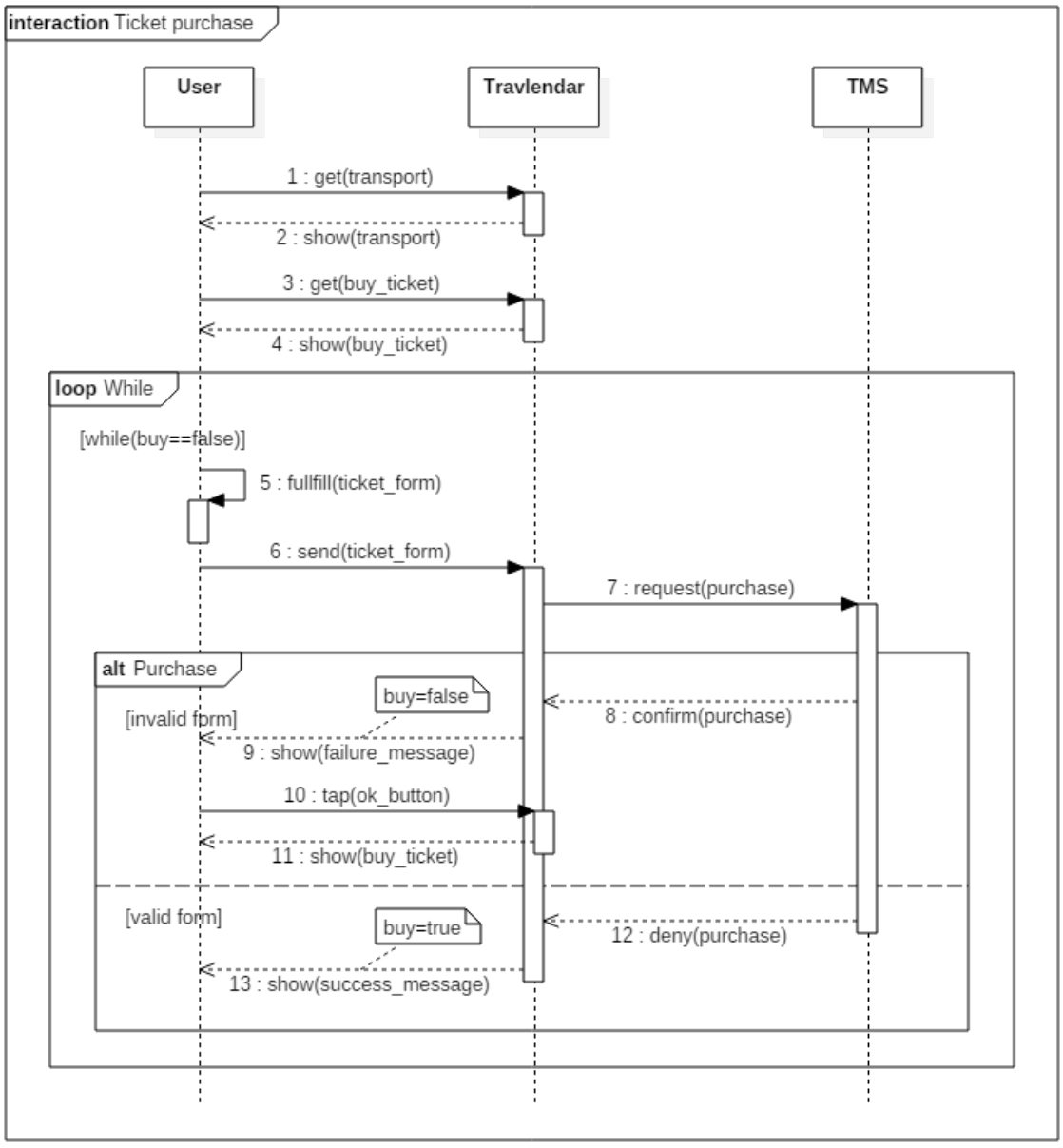
User login


interaction Login
User
Travlendar
1 : get(login_screen)
2 : show(login_screen)
loop While
3 : fullfill(login_form)
[while(log==false)]
4 : send(login_form)
5 : verify(input)
alt Check
log=false
[invalid form]
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 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 entered 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 application login:

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

R.1.2 The system must guarantee the login of the user if and only if the inserted credentials results 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 long enough for the travel to the new location.

R.2.3 The system must control wheter it is possible for the user leaving from the previous event's location to arrive on time.

R.2.4 The system must control wheter it is possible for the user leaving the event to be able to arrive on time at the following event's location.

R.2.5 The system must control wheter given the current position the user can be able to arrive at the location on time for the appointment.

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

R.2.7 The system must suggest possible solutions to arrange conflicting events: either postponing the beginning time of the later event or anticipating the ending time of the earlier one.

Section [R.4] treats all the requimerements related to the public means reservation:

R.4.1 The system must be able to locate the nearest bike to bike sharing system in the city.

R.4.2 The system must be able to locate the nearest car sharing system in the city.

R.4.3 The system must be able to locate the public trasportation areas in the city.

Section [R.5] treats all the requimerements related to outside phenomena:

R.5.1 The system must notify the user in occurance of strikes.

R.5.2 The system must notify the user in occurance of days of bad weather.

Section [R.6] treats all the requimerements related to the customization of the preferences:

3.3 Performance requirements

3.4 Design constrains

3.4.1 Standards compliance

3.4.2 Hardware limitations

3.4.3 Any other constraint

3.5 Software system attributes

3.5.1 Reliability

3.5.2 Availability

3.5.3 Security

3.5.4 Maintainability

3.5.5 Portability

4 | Formal analysis using alloy

5 | Effort spent

6 | References