POLITECNICO
MILANO 1863

Travlendar$^+$

Requirement Analysis and Specification Document

Mirko Salaris, Piervincenzo Ventrella, Pietro Cassarino

October 24, 2017

Deliverable:	RASD
Title:	Requirement Analysis and Verification Document
Authors:	Mirko Salaris, Piervincenzo Ventrella, Pietro Cassarino
Version:	1.0
Date:	27-October-2017
Download page:	https://github.com/mirkosalaris/CassarinoSalarisVentrella/
Copyright:	Copyright 2017, M. Salaris, P. Ventrella, P. Cassarino All rights reserved

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This document is the Requirement Analysis and Specification Document for the Travlendar+ application. Its aim is to inform about what the application offers, about requirements and goals that the system must present. This document offers also an analysis of the world and of the shared phenomena regarding Travlendar+. RASD contains class diagram to show domain model and other diagrams which illustrate, with more details, transactions of the functionality of the application.

1.2 Scope

Travlendar+ is an application that allows people to organize and to track their appointments and meeting by registering them into the application. A person becomes a User of Travlendar+ by registering himself to the app. After this phase, users can start to use basic functionalities of the app (e.g. register appointments and organize meetings).

The app allows users to create appointments, eventually inviting other users of the app, facilitating communication issues. The goal of Travlendar+ is to organize in the best way all daily commitments of its users considering all the possible problems that can influence travels and trips (e.g. weather conditions, strikes, etc.).

When a User creates an event, he can add, for the travel, eventual passengers or baggage, so that the app can suggest better trip choices. Travlendar+ allows users to visualize their planned schedule too.

The dynamicity of the software allows users to set some personal preferences, for example, to set a flexible break window time for having free time or lunch, to choose eco-friendly solutions for his trips, to deactivate some transportation means.

The system interfaces with other firms (e.g. transportation companies, territory maps companies, sharing transportation companies) to offer a more comprehensive user experience. In this way, a User has the possibility to buy tickets and passes for transportation means. Moreover, the system offers a location service for vehicles of affiliated shared transport companies but to proceed with the renting the User is redirected to the company's app.

1.2.1 Goals

The system should:

[G1] allow a Person to use the app in his/her device language, if the language is one of the supported languages, English otherwise

[G2] allow every Person to have his/her own account, after providing credentials

[G3] allow a Person to log in to his/her own account, and only in that

[G4] allow User to use his/her own credit card

[G5] allow a User to set personal preferences

- [G5]#1 specify his/her preference for eco-friendly solution
- [G5]#2 set break time windows, either flexible or fixed
- [G5]#3 set time slot in which the use of specific transportation means should be avoided
- [G5]#4 set a minimum distance below which a specific transportation mean should be avoided
- [G5]#5 set a maximum distance beyond which a specific transportation mean should be avoided
- [G5]#6 set a specific transportation mean as permanently disabled

[G6] allow User to create a new appointment, specifying its details

[G7] allow User to visualize his/her appointments, eventually updating its details

[G8] during the specification of details of an appointment, allow User to visualize suggested travel solutions and to choose one of them

[G8]#1 during the specification of details of an appointment, before the choice, allow User to set a preferred transportation mean

[G9] during the specification of details of an appointment, allow User to invite other persons to the appointment

[G10] during the specification of details of an appointment, if for the selected travel solution a ticket is expected, allow User to specify if he/she owns a ticket, either ordinary or a pass (eventually specifying the deadline)

[G10]#1 if the user does not own a ticket and the transportation company is affiliated with Travlendar+, he/she can buy it

[G11] allow User to visualize daily/weekly schedule

[G12] allow User to delete an existing appointment

[G13] allow User to locate nearest vehicle of a vehicle sharing system, if that is the transportation mean of choice of an incoming appointment

[G14] allow User to being redirected to the company's app to proceed with the renting of the selected shared vehicle

[G15] show suggestions based on: traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers

[G16] notify the User about incoming appointments

[G16]#1 asking for confirmation on the transportation means previously planned

[G16]#2 in case no transportation mean was selected, suggesting one

[G16]#3 communicating eventually complications (bad weather, traffic) and suggesting more feasible solutions

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- ***welcome page*:** after completing the sign up process,when the user logs in for the first time, he is redirected to this page where sequentially the app asks him to insert credit-card data, driving licence data and set his preferences the user can skip any of this phases and complete them in a second time. After this process, the initial settings configuration is completed

- ***personal preferences*:** with this term we mean that here the user can:
  - specify break time windows;
  - specify the interest or not for eco-friendly solutions;
  - specify constraints, such as avoiding bike, on the travel means solutions).

- ***supported languages*:**

- ***valid credentials*:**

- ***appointment details*:**

- ***incoming appointment*:**

- ***affiliated company*:**

1.3.2 Acronyms

- ***S2B*:** System to Be;

- ***API*:** Application Programming Interface.

1.3.3 Abbreviations

- ***[Gn]*:** n-th goal. Apart when it is actually defined, it is always a reference to the definition of the goal
- ***[Dn]*:** n-th domain assumption. Apart when it is actually defined, it is always a reference to the definition of the domain assumption
- ***[Rn]*:** n-th requirement

1.4 Revision history

1.5 Document Structure

2 Overall Description

2.1 Product perspective

2.2 Product functions

Here we provide several scenarios to better delineate the purposes for which the app should be designed, the situations the S2B will deal with and more generally to have a better comprehension of the associated environment

2.2.1 Scenario 1

Mario is the director of his company and he has seen an ad of Travlendar+, so he wants to try it to organize the weekly meetings with his employees. After the installation, he has to register to the app. The first thing he creates is his weekly program. We work from Monday to Friday, from 8 am to 16 pm. There is an actual meeting coming up on Friday at 8 pm, so he creates a new meeting in the app. After setting the time and the day, he invites his employees to join the meeting. He will remain in his office, so he will not need to use any travel means.

Giovanni is one of Mario's employee and he is registered to Travlendar+. He receives a notification and accepts the invitation to the meeting. He chooses to reach the location by walk because he will already be nearby.

Alex is another employee and does not have an account on Travlendar. He receives an email with an invitation link to register to the app. After the registration, the app redirects him to the meeting's invitation and he will proceed by accepting it and choosing to go by car. Then he explores the app and decides to add his weekly program. The app finds out that he will be in the caf near the metro at 6:40 pm, so it suggests Alex to take the metro instead of the car. He accepts the suggestion.

2.2.2 Scenario 2

Paolo, a resident of Bergamo, has recently registered to Travlendar+ and during the initial setup has specified a flexible launch break from 11:00 am to 13:00 am, with at least 40 minutes of break.

Tomorrow he is going to have an audition at 12:30 pm, in Monza. He inserts the event in the app and after having specified that the audition will end at about 13:30pm, he looks at the suggestions of the app on the travel means to take: the app suggests him some travel solutions, but he does not specify which he's going to take because he wants to think about it overnight.

The next morning, the app sends Paolo a reminder with two travel solutions:

- go by car, leaving at about 11.45am, arriving at 12.27 pm;
- take the bus, passing at 10:49 am and arriving at 11:38 am.

He chooses the second option to avoid being late at the audition.

2.2.3 Scenario 3

Alex is a professor of Bologna University, he has a short memory and is very badly organized, so he decides to rely on Travlendar+. Alex downloads it on occasion of a work trip. He signs up and decides to insert his credit card data for an eventual purchase from the app.

He needs to reach the University of Parma to hold a conference. Alex sets up the app to arrive in Parma by train. Travlendar+ asks Alex the kind of event and he specifies it is a formal work meeting. The app asks him which transport means he wants to take in order to get to the university from the railway station. Alex opts to go by bicycle, although the app suggest not to, because of the formal type of meeting.

The departure day Alex is in a shopping center with his family, he has completely forgotten that he has a train to take, but Travlendar+ solves the problem by notifying him of the appointment. At that point, Alex has no more time and chooses to buy the train tickets using the app. While he is on the train, Travlendar+ suggests him to choose another transport means (instead of a bike) because of the bad weather conditions. Alex accepts the advice and decides to take a bus.

2.2.4 Scenario 4

Luca is a meteorologist who works for a laboratory in Venice. He knows very well all the climatic problem that humans are creating in their Country. Luca finds Travlendar+ very appropriate to help in solving this problematic. He likes to opt for an Eco-friendly solution by setting up this preference in the app settings. In this way he can avoid, at least in this aspect, further damaging the environment. His favorite functionality is bike sharing because of its innovative localization system and its low environmental impact.

2.2.5 Scenario 5

Mark, son of Lucas, asks to his father to bring him to the basketball tournament of Sunday morning.
Lucas checks the daily schedule for Sunday and he notices he already has an appointment with the hairdresser but, of course, spending time with his son is more important, so he decides to delete the previous event on the agenda and set a new one.
Mark asks to the father if also his team mate Mike can come. Of course Mark and Mike have to bring with them the bags with the jersey and the basketball shoes, so Lucas, creating the event on Travlendar+, after specifying the location of the basketball court, specifies also that he will bring with him baggage and passengers.
Unfortunately his car is broken, so Lucas use the app to look for alternative solutions.
Travlendar+, taking into account the constraints previously settled by the man, suggests to him to use Enjoy or SmartToGo, two well known car sharing companies that will solve his problem.
Lucas accepts the suggested solution and proceeds with the creation of the event.

2.2.6 Scenario 6

Mary, John's wife, one week ago, asked to her husband to pick the children up to school on Monday at 13.00 and, because she knows John, forced him to take note of that with Travlendar+.
So John planned this event on the app specifying that he will use the car to do it. He specified also the location of the school.
On Monday morning, as usual, Travlndar+ shows to John his daily program reminding him about his children and showing the previous travel mean planned.
John, still intentioned to pick the children up with the car, does not modify the plan, closes the app and goes to work in the other side of Milan.
At 12.00 Travlendar+, according to the GPS position of the man, suggests him to leave in 15 minutes.
Travel+ also suggests to avoid to go through Viale Gioia because of the traffic and take the SS1.
Thanks to Travlendar+ John manages to be on time, collect the children and make his wife proud of him.

2.3 User characteristics

2.3.1 Actors

- Person: a person that doesn't have a registered account. The only thing that he/she can do is to proceed with the Sign Up process;
- User:a person passed through a successful registration process and now able to use all the TravLendar+ services. He/she can login to the system and, after that, use all the platform's functionalities.
- Credit Institution: the institution that checks the credit-card validity and reports it to the S2B;
- Licence Istitution:the institution that checks the drive license validity and reports it to the S2B;
- Google: the system with whom the S2B retrieves the maps and related information about routes, real-time traffic situations, estimated travel time and weather conditions;
- Transport Company System: the system of the affiliated companies with whom the S2B interacts to allow the user to buy the tickets for the associated travel mean;

- Shared vehicle Company System: the system with whom the S2B interacts to allow the user to visualize the map of the available vehicles to rent and locate the nearest one. Vehicle Company System provides the GPS locations of the vehicles. To proceed with the renting the user is redirected to the vehicle Company System/App. The latter, to autenticate the user, uses the same credentials provided to the S2B.
- Manager: the person able to manage the system. His activities consists in adding or deleting affiliated companies (both transport companies or shared vehicle companies) and adding new type of tickets according to the services provided by affiliated transport companies.

2.4 Assumptions, dependencies and constraints

2.4.1 Text Assumptions

- credit cards do not have an expiration date

2.4.2 Domain Assumptions

[D1] the user's device should allow the app to retrieve the language settings

[D2] when the registration process begin, the User always inserts his/her credentials

[D3] when the S2B sends an email, it is always received by the receiver

[D4] every Person has an email address

[D5] the User shall remember his password

[D6] the User's device has a working GPS installed, to which the app has access

[D7] the affiliated companies provide a localization service APIs

[D8] Google Maps services take traffic into consideration

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Here we provide some basic mockups to show how the interface should appear to the user:

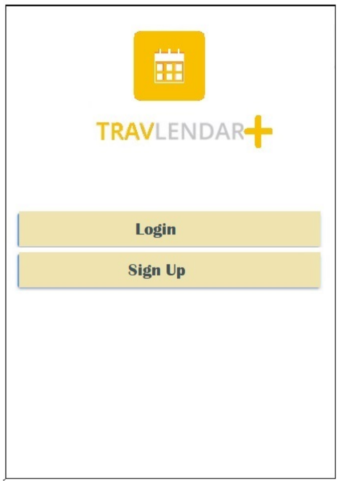
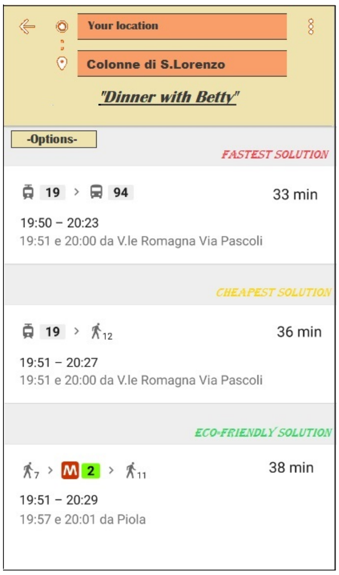


Figure 1: Login



Your location

Colonne di S.Lorenzo

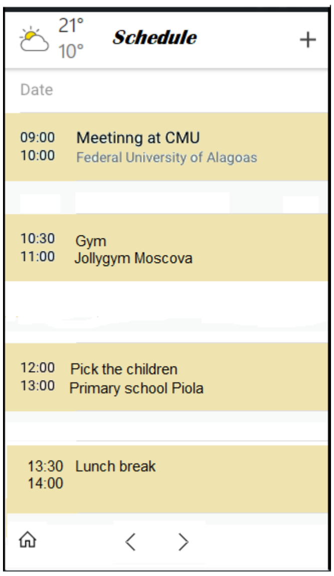
"Dinner with Betty"

19 > 94 33 min
19:50 – 20:23
19:51 e 20:00 da V.le Romagna Via Pascoli

19 > 12 36 min
19:51 – 20:27
19:51 e 20:00 da V.le Romagna Via Pascoli

7 > M 2 > 11 38 min
19:51 – 20:29
19:57 e 20:01 da Piola

Figure 2: Select solution



09:00 10:00 Meetinng at CMU
Federal University of Alagoas

10:30 11:00 Gym
Jollygym Moscova

12:00 13:00 Pick the children
Primary school Piola

13:30 14:00 Lunch break

Figure 3: Visualize schedule

3.1.2 Hardware Interfaces

The main hardware interface of the system consists in the access to the GPS data in the mobile application. The application also requires Internet connectivity and internal storage access.

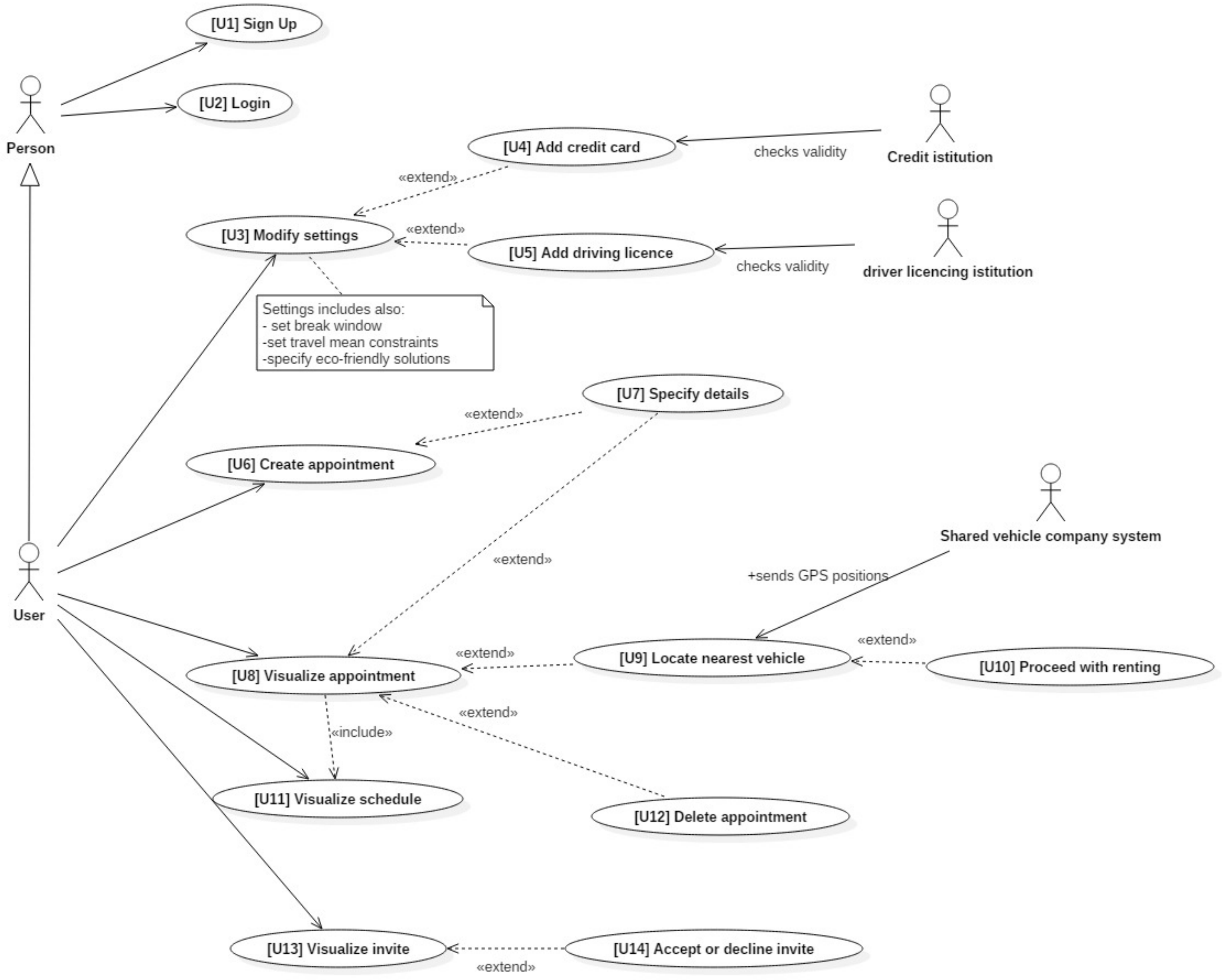
3.1.3 Software Interfaces

The mobile application must support Android,iOS and the remaining main OSs (further details are discussed in paragraph 3.6.5 Portability). The web application works on any web server that supports Java. The back-end stores its data in a DBMS and can run on every platform that supports the JVM.

3.1.4 Communication Interfaces

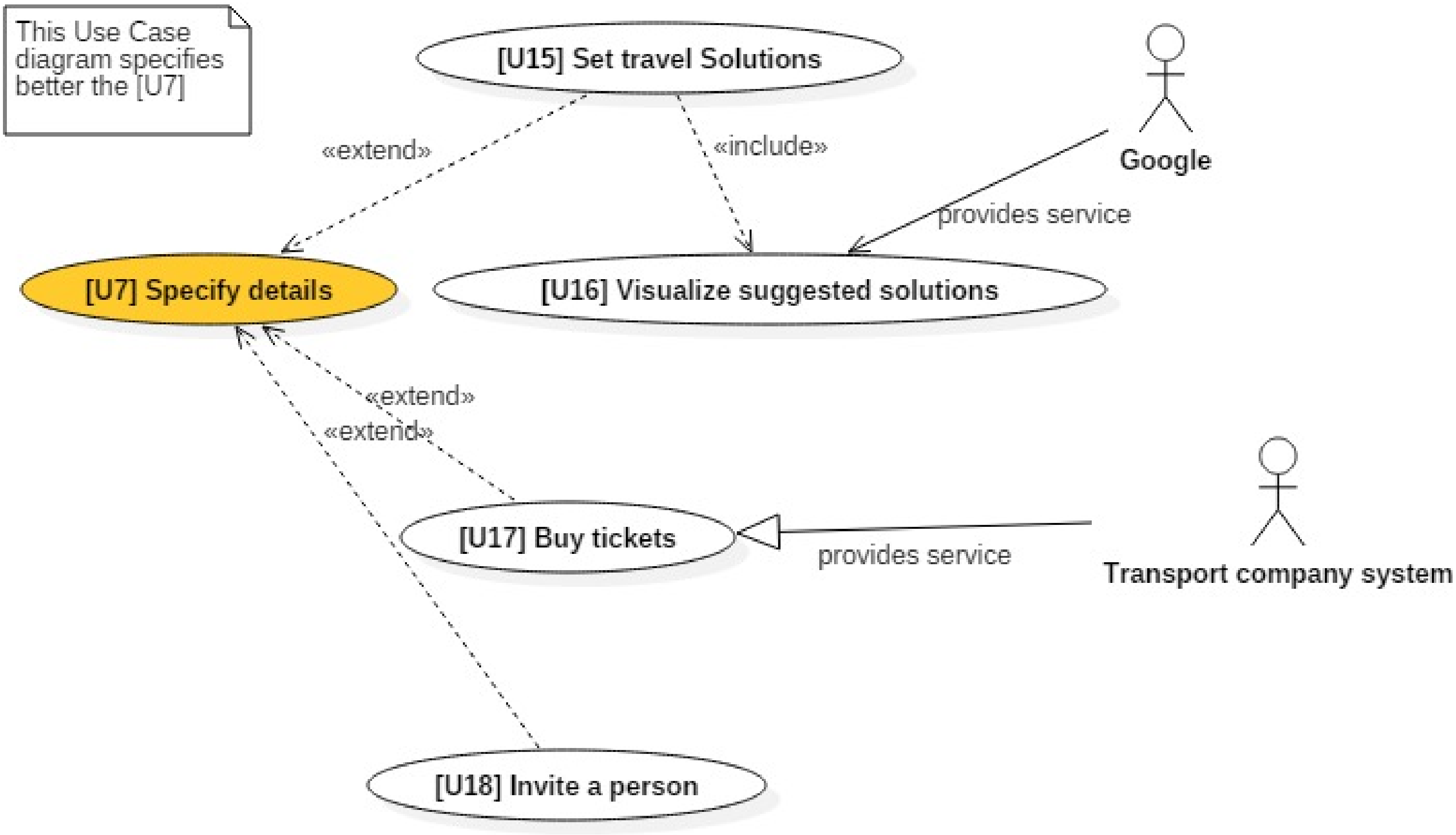
3.2 UML modeling

3.2.1 Use Case diagram



-1

Figure 4: User Use Case diagram



-1

Figure 5: specify details

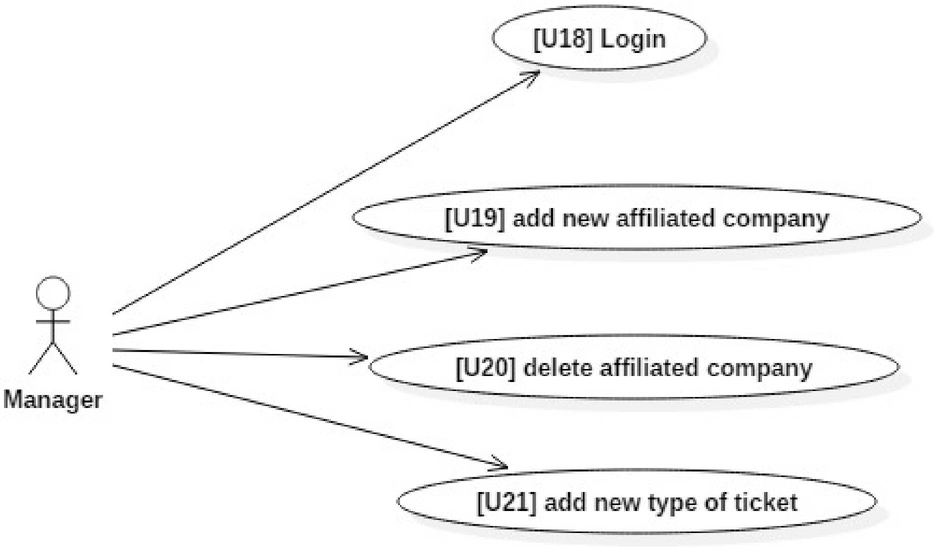

[U18] Login
[U19] add new affiliated company
Manager
[U20] delete affiliated company
[U21] add new type of ticket

Figure 6: Manager Use Case diagram

User creates appointment

Use case:	User creates an appointment
Actors:	User
Entry condition:	The user must be logged
Flow of events:	The user creates an event (a meeting, appointment or generic event) giving it a name; User specifies the location of the event; User specifies details such as passengers or baggage; User selects a travel mean taking to account apps suggestion; The app takes note of the settings and send a confirmation; The app redirects the user to the main page.
Secondary flows:	User does not specify a travel mean and let it blank; The app takes anyway note of the setting and alert the user of the missing information; The app redirects the user to the main page.
Exceptions:	Warnings messages are created in the following cases: • User creates an event that overlaps another event; • User creates an event with a location that is unreachable in the allocated time; • User creates an event that violates the set constraints about the break windows.
Post conditions:	The user is successfully redirected to the main page.

Sign Up

Use case:	Sign Up
Actors:	Person
Entry condition:	user must be not already logged;
Flow of events:	the person inserts full name and email contact; the app send an email with the confirmation link; the person give the confirmation trough the link on the mail; the app give the welcome to the new user.
Secondary flows:	none.
Exceptions:	the person inserts a wrong email contact; the sign up cannot proceed.
Post conditions:	the person is successfully signed up and become an actual logged user.

Initial settings configuration

Use case:	Initial settings configuration
Actors:	User
Entry condition:	the User just has just completed the sign up process; user must be logged.
Flow of events:	User insert sequentially the following information: • Credit card; • Driving licence; • Break time windows; • Interest for Eco-friendly solutions on the travel means. • Constraints on travel means. The app, for each step, check the info and send a confirmation; The app redirects the user to the main page.
Secondary flows:	User skips to specify one or more information that could be specified later in the settings. The app notifies the user about the missing information and redirects anyway the user to the main page.
Exceptions:	user inserts inconsistent information (incorrect credid-card/licence information, break time shorter than 30 minutes); The app allerts the user and asks him to insert again the info.
Post conditions:	the set configurations are successfully saved and the user is redirected to the main page.

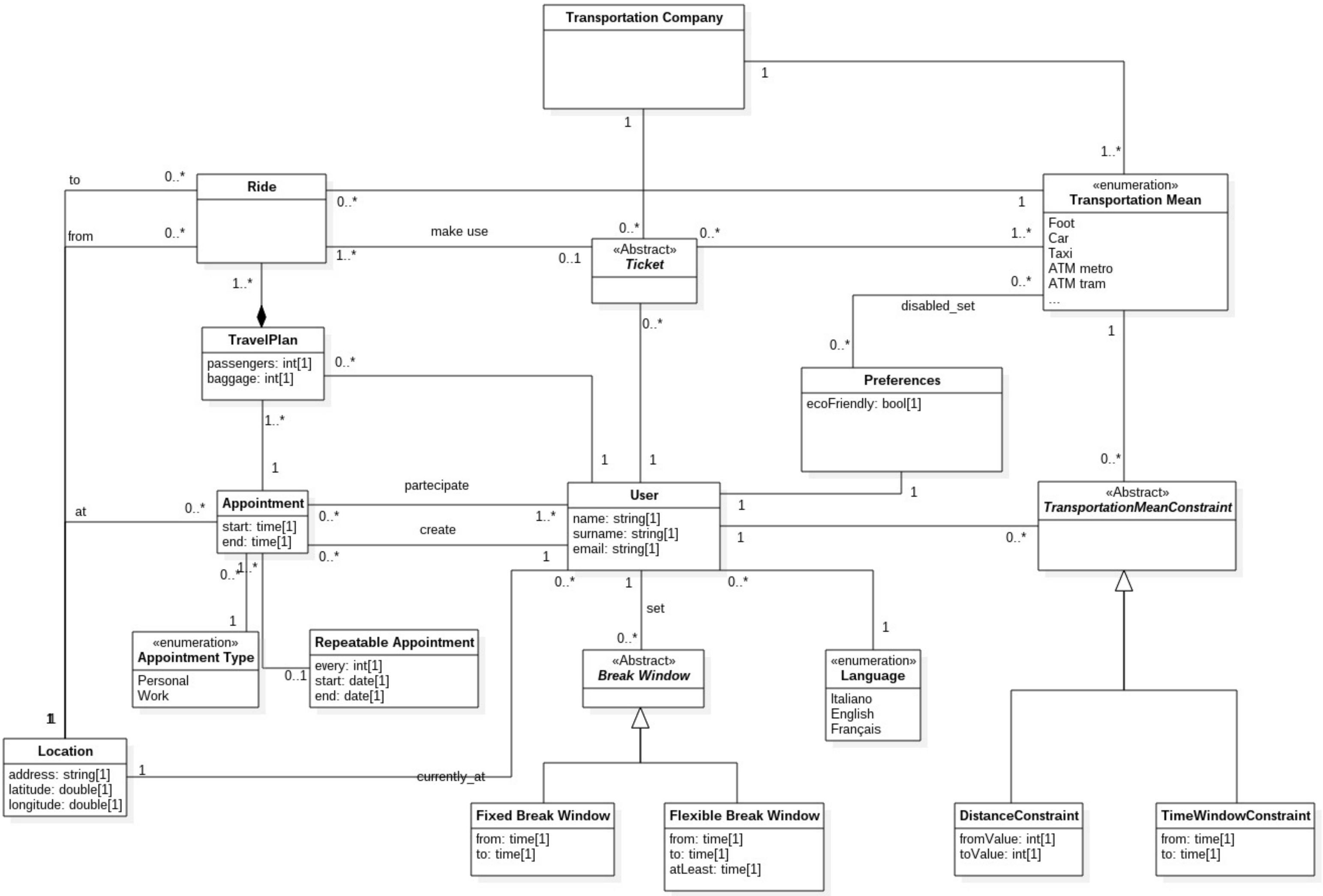
User visualizes the appointment

Use case:	User visualizes the appointment.
Actors:	User
Entry condition:	user must be logged; the user must have selected the appointment from the schedule;
Description:	user visualize an appointment and eventually: • If not specified yet, set a travel mean; • Modify the previous travel mean or other detail such as location, baggage or passengers; • Buy a ticket for the travel mean; • Rent a shared vehicle if is the proper time to do that and locate the nearest one; • Delete the appointment.

User buy travel ticket

Use case:	User buy travel ticket
Actors:	User
Entry condition:	user must be logged; user must have added a payment card .
Flow of events:	User select an event; User, through the app, searches for tickets for the specified travel mean; User select the tickets option and picks one; User proceeds with the payment; The payment operation ends successfully; The app send a confirmation and redirects the user to the main page;
Secondary flows:	none
Exceptions:	The payment is rejected (not enough credit, expired card, ..); The app notifies the user; The app redirect the user to the home page;
Post conditions:	User successfully books the tickets and is redirected to the main page.

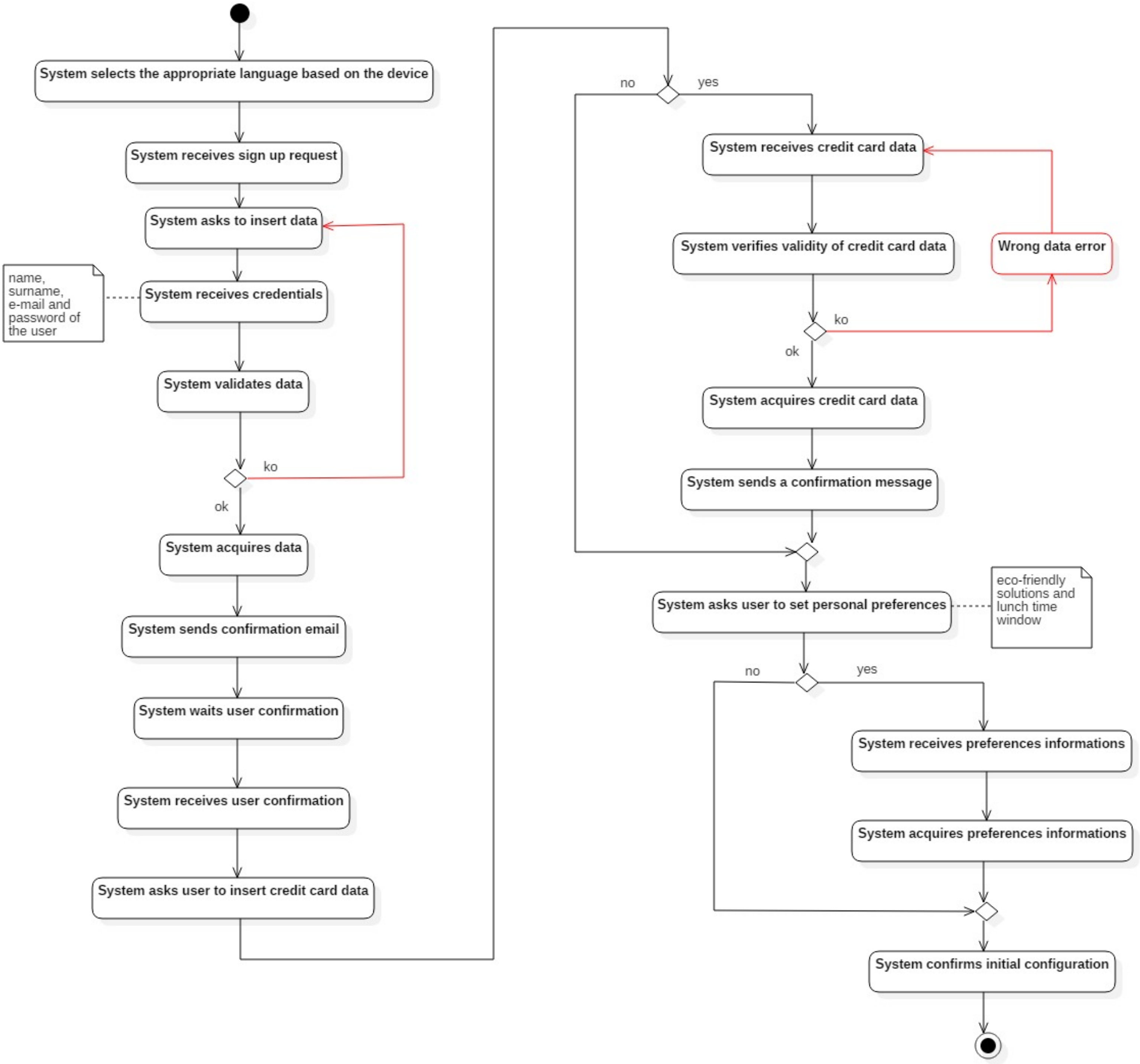
3.2.2 Class diagram



-1

Figure 7: General class diagram

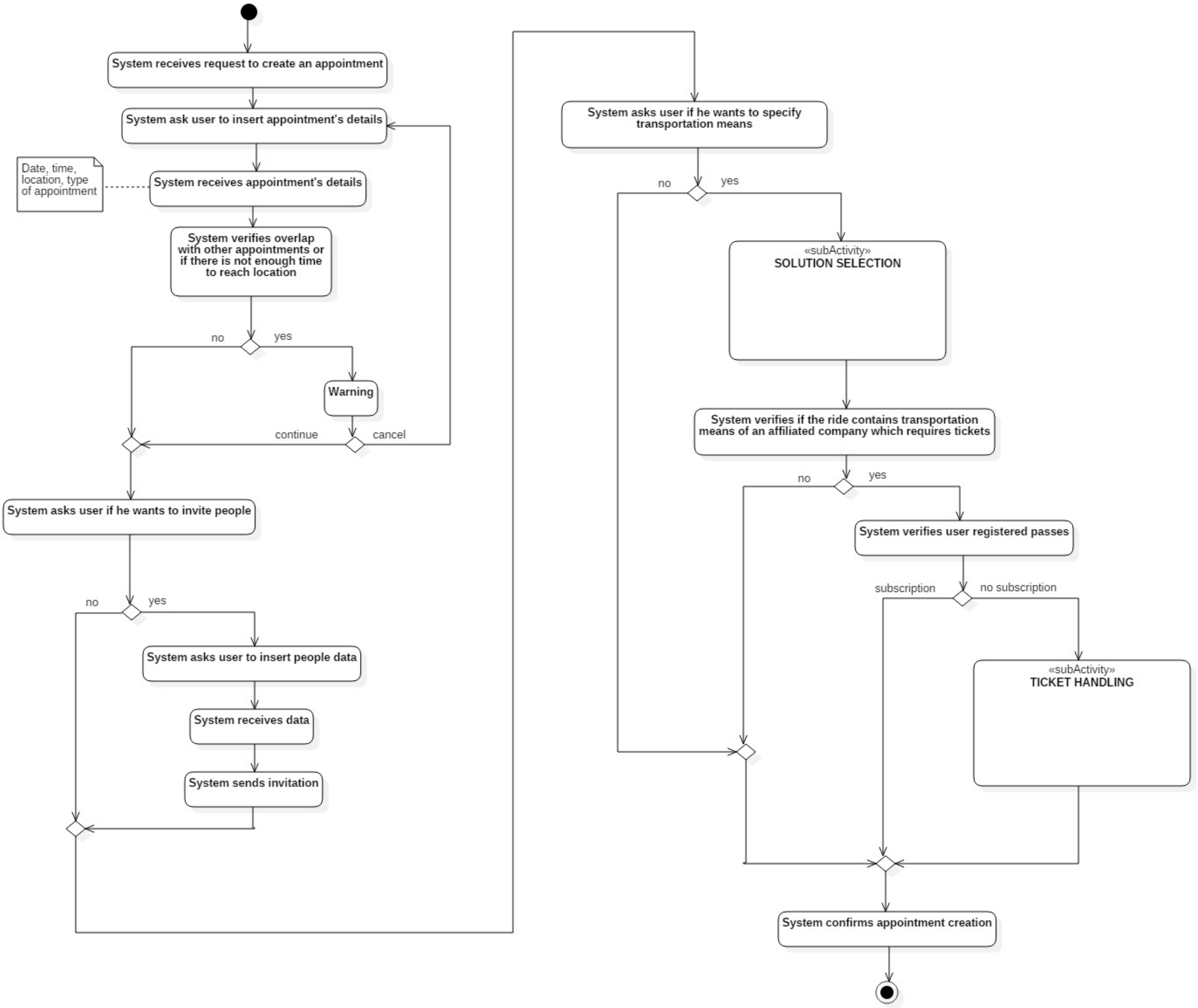
3.2.3 Activity diagrams



System selects the appropriate language based on the device
System receives sign up request
System asks to insert data
name, surname, e-mail and password of the user
System receives credentials
System validates data
ko
ok
System acquires data
System sends confirmation email
System waits user confirmation
System receives user confirmation
System asks user to insert credit card data
no
yes
System receives credit card data
System verifies validity of credit card data
Wrong data error
ko
ok
System acquires credit card data
System sends a confirmation message
System asks user to set personal preferences
eco-friendly solutions and lunch time window
no
yes
System receives preferences informations
System acquires preferences informations
System confirms initial configuration

-1

Figure 8: Registration activity diagram


System receives request to create an appointment
System ask user to insert appointment's details
Date, time, location, type of appointment
System receives appointment's details
System verifies overlap with other appointments or if there is not enough time to reach location
no
yes
Warning
continue
cancel
System asks user if he wants to invite people
no
yes
System asks user to insert people data
System receives data
System sends invitation
System asks user if he wants to specify transportation means
no
yes
«subActivity»
SOLUTION SELECTION
System verifies if the ride contains transportation means of an affiliated company which requires tickets
no
yes
System verifies user registered passes
subscription
no subscription
«subActivity»
TICKET HANDLING
System confirms appointment creation

-1

Figure 9: Creation appointment diagram

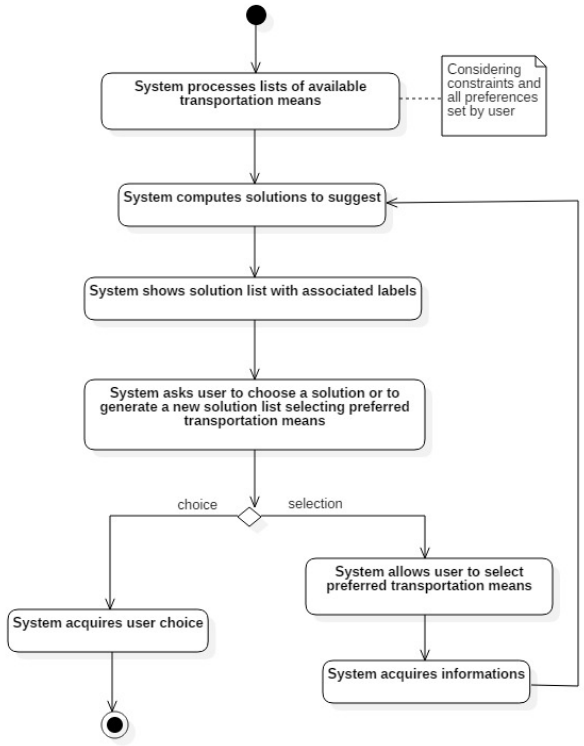

System processes lists of available transportation means
Considering constraints and all preferences set by user
System computes solutions to suggest
System shows solution list with associated labels
System asks user to choose a solution or to generate a new solution list selecting preferred transportation means
choice
selection
System allows user to select preferred transportation means
System acquires user choice
System acquires informations

Figure 10: Solution selection sub-activity

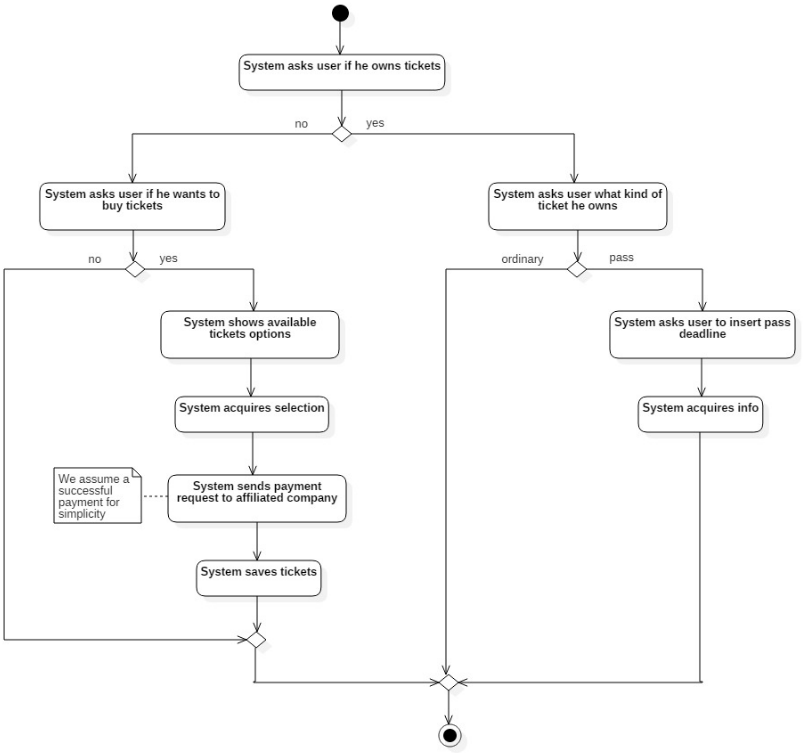

System asks user if he owns tickets
no
yes
System asks user if he wants to buy tickets
System asks user what kind of ticket he owns
no
yes
ordinary
pass
System shows available tickets options
System asks user to insert pass deadline
System acquires selection
System acquires info
We assume a successful payment for simplicity
System sends payment request to affiliated company
System saves tickets

Figure 11: Ticket handling sub-activity

3.2.4 Sequence Diagram

We chose to omit the Sequence Diagram considering that the Activity diagram in this phase will fit better.

3.3 Functional Requirements

[G1] allow a Person to use the app in his/her device language, if the language is one of the supported languages, English otherwise

[R1] the app must retrieve the language settings from the device

[R2] the app should be able to display its content in all the supported languages

[D1] the user's device should allow the app to retrieve the language settings

[G2] allow every Person to have his/her own account, after providing credentials

[R3] the S2B must provide a way to begin the registration process

[R4] after the insertion of the credentials and their validation, the S2B has to send to the provided address an email with an activation link

[R5] the registration fails if the inserted email is already associated to an account

[R6] when the Person confirms through the activation link, he/she becomes a User

[R7] in case of non valid credentials, the system must reject them and restart the registration process

[D2] when the registration process begin, the User always inserts his/her credentials

[D3] when the S2B sends an email, it is always received by the receiver

[D4] every Person has an email address

[G3] allow a Person to log in to his/her own account, and only in that

[R8] for each User, the S2B must remember the pair email-password inserted during the registration process

[R9] the S2B must grant access to the User if and only if the User inserts an existing email and the associated password

[D5] the User shall remember his password

[G4] allow User to use his/her own credit card

[R10] the S2B must accept credit card data from the User

[R11] the S2B must forward the credit card data to a credit institution to validate them

[R12] the S2B must let the User use the credit card if and only if the inserted credit card data are valid

[G5] allow a User to set personal preferences

[R13] for each type of preference, the S2B must provide to the User the possibility to change and store the preference value(s)

[G6] allow User to create a new appointment, specifying its details

[R14] the system must provide a way to start the creation of a new appointment

[R15] during the process the user shall insert the appointment details

[G7] allow User to visualize his/her appointments, eventually updating its details

[R16] the system provides a way to visualize created appointments

[R17] after visualizing an appointment, a user can choose to edit its details

[G8] during the specification of details of an appointment, allow User to visualize suggested travel solutions and to choose one of them

[R18] when time and location of the current appointment are set, the S2B produces a list of travel solutions with associated suggestions

[R19] the S2B provide the user the possibility to choose one of the travel solutions or leave the travel plan unspecified

[R20] the S2B also provide the possibility to choose a preferred transportation mean

[R21] when a new preferred transportation mean is selected the S2B has to recompute the list of solutions according to the new preference

[G9] during the specification of details of an appointment, allow User to invite other persons to the appointment

[R22] when time and location of the current appointment are set, the S2B offers the possibility to invite other users or persons, through their emails.

[R23] when a User or a Person is invited, the S2B will inform him/her sending an email

[D3] when the S2B sends an email, it is always received by the receiver

[G10] during the specification of details of an appointment, if for the selected travel solution a ticket is expected, allow User to specify if he/she owns a ticket, either ordinary or a pass (eventually specifying the deadline)

[R24] when the user selects a travel solution for which a ticket is expected, the S2B asks the User specify if he/she owns a ticket (either ordinary or a pass, in which case the deadline has to be inserted)

[R25] if the user has selected a travel solution for which a ticket is expected and the User said to not own a ticket, the S2B asks him/her to buy a ticket (options available only for transportation means of affiliated companies)

[G11] allow User to visualize daily/weekly schedule

[R26] existing appointments can be viewed together as a schedule view

[R27] the schedule can be daily or weekly

[G12] allow User to delete an existing appointment

[R28] in the schedule view the User can select one or more appointments

[R29] in the schedule view, selected appointments can be deleted

[G13] allow User to locate nearest vehicle of a vehicle sharing system, if that is the transportation mean of choice of an incoming appointment

[R30] when an user visualize an incoming appointment for which a vehicle sharing system has to be used, the S2B provides a localization service

[D6] the User's device has a working GPS installed, to which the app has access

[D7] the affiliated companies provide a localization service APIs

[G14] allow User to being redirected to the company's app to proceed with the renting of the selected shared vehicle

[R31] the S2B offers the possibility to rent a vehicle

[R32] when the user selects a vehicle to rent, the app redirects him/her to the right company's app or site

[G15] show suggestions based on: traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers

[R33] if weather forecast are bad: foot, bicycle motorbike are discouraged

[R34] if strikes have been announced, public transport is discouraged

[R35] in case of baggage or passengers a car is recommended

[R36] in case of a work appointment, bicycle is discouraged

[D8] Google Maps services take traffic into consideration

[G16] notify the User about incoming appointments

[R37] when an appointment becomes incoming, a notification is sent to the user

[R38] if a travel plan has not already been choosen, the notification suggests one

[R39] if a travel plan has already been choosen but at the moment better travel plans are available (according to new weather forecast and news on traffic and strikes), the notification suggests a new feasible solution

3.4 Performance Requirements

The system has to be able to respond to a possibly great number of simultaneous requests and more generally to a great number of request throughout the day. The S2B, at least for the start, will only be available for the Lombardy region. Based on demographic analysis (number of inhabitants, number of people under the age of 60, number of smartphones sold over the past 2 years), it was decided to design the S2B to support 100,000 users simultaneously, but scalability needs to be guaranteed.

3.5 Design Constraints

3.5.1 Standard compliance

To ensure interoperability the S2B will follow the W3C web standard and will be as adherent as possible to coding practices in relation to the use of HTML/XHTM, CSS and Java programming language Moreover the use of private libraries will be avoided.

3.5.2 Hardware limitations

- Mobile App:
  * 3G connection
  * GPS
  * Space for app package
- Web App:
  * Modern browser with AJAX support

3.5.3 Any other constraint

Regulatory policies The system will ask for users' payment informations and obviously, in addition to store them safely, will use them only for fees and rides payments. Moreover, the system will have to ask for users' permission in order to retrieve and use their positions. Email addresses won't be used for commercial uses.

3.6 Software System Attributes

3.6.1 Reliability

The system must guarantee a 24/7 service. Very small deviations from this requirement will be obviously acceptable.

3.6.2 Availability

The S2B must guarantees a 3-nines availability (99.9 percent) with a downtime not greater than 8 hours per year.

3.6.3 Security

User credentials and payment information will be stored. Data confidentiality is a primary concern. In addition,when the user wants to buy tickets or rent a shared vehicle, the stored information must be sent to affiliated transport Company systems or shared vehicle Company systems . To ensure the security and the confidentiality of this information, the S2B must be able to adopt access managment protocols and comunication protocols able to prevent not granted access and/or Sniffing/Spoofing activities performed by third.

3.6.4 Maintainability

The S2B must be designed in the way to easly correct defects or their cause, repair or replace faulty or worn-out components without having to replace still working parts, prevent unexpected working condition, maximize its useful life, maximize efficiency, reliability, and safety, meet new requirements, make future maintenance and cope with a changed environment.

3.6.5 Portability

The S2B must be able to run in all main mobile OS(Android, iOS, Nokia OS, blackBerry OS, windowsphone OS) and being supported by all the main Web Browser(Google Crome, Safari,FireFox, Microsoft Edge, Internet Explorer, ...)

4 Formal Analysis Using Alloy

```
//primitive signatures
sig Name {}

sig Surname {}

sig Email {}

sig Address {}

sig Double {}

sig Time {}

sig Date {}

enum Bool {
        True,
        False
        }

//signatures
sig Ride {
        makeUseTicket: lone Ticket,
        byTranMean: TransportationMean,
        fromLocation: Location,
        toLocation: Location
        }

sig TransportationCompany {}

abstract sig Ticket {
        usedFor: some TransportationMean,
        providedByCompany: TransportationCompany
        }

sig User {
        name: Name,
        surname: Surname,
        email: Email,
        ownsTicket: set Ticket,
        setPreferences: Preferences,
        hasConstraints: set TransportationMeanConstraint,
        speaksLanguage: Language,
        setBreakWindows: set BreakWindow,
        createsAppointment: set Appointment,
        partecipatesToAppointment: set Appointment,
        hasTravelPlan: set TravelPlan,
        currentlyAtLoc: Location
        }

sig Appointment {
        start: Time,
        end: Time,
        atLocation: Location,
        hasType: AppointmentType,
        isRepeatable: lone RepeatableAppointment
        }

sig Location {
        address: Address,
        latitude: Double,
        longitude: Double
        }

enum AppointmentType {
        Personal,
        Work
        }

sig RepeatableAppointment {
        every: Int,
        start: Date,
        end: Date
        }

sig TravelPlan {
        passengers: Int,
        baggages: Int,
        hasRide: some Ride,
        forAppointment: Appointment
        }

abstract sig BreakWindow {}

sig FixedBreakWindow extends BreakWindow {
        from: Time,
        to: Time
        }

sig FlexibleBreakWindow extends BreakWindow {
        from: Time,
        to: Time,
        atLeast: Time
        }

enum Language {
        Italiano,
        English,
        Francais
        }

abstract sig TransportationMean {
        belongsToCompany: TransportationCompany
        }

one sig Foot, Car, Taxi, Metro, Tram extends TransportationMean {}

sig Preferences {
        ecoFriendly: Bool,
        disabledTranMean: set TransportationMean
        }

```

```
abstract sig TransportationMeanConstraint {
        associatedToTranMean: TransportationMean
        }

sig DistanceConstraint extends TransportationMeanConstraint {
        fromValue: Int,
        toValue: Int
        }

sig TimeWindowConstraint extends TransportationMeanConstraint{
        from: Time,
        to: Time
        }


pred show {}

run show
```

5 Effort Spent

6 References

6.1 Software and Tools

- LaTeX for typesetting document
- GitHub for version control and team work
- Alloy latex highlighting package: https://github.com/Angtrim/alloy-latex-highlighting
- StarUML for uml models