



POLITECNICO
MILANO 1863

Travlendar⁺

Requirement Analysis and Specification Document

Mirko Salaris, Piervincenzo Ventrella, Pietro Cassarino

October 18, 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

1	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.2.1	Goals	4
1.3	Definitions, Acronyms, Abbreviations	5
1.3.1	Definitions	5
1.3.2	Acronyms	5
1.3.3	Abbreviations	5
1.4	Revision history	5
1.5	Reference documents	5
1.6	Document Structure	5
2	Overall Description	6
2.1	Product perspective	6
2.2	Product functions	6
2.2.1	Scenario 1	6
2.2.2	Scenario 2	6
2.2.3	Scenario 3	6
2.2.4	Scenario 4	6
2.2.5	Scenario 5	7
2.2.6	Scenario 6	7
2.3	User characteristics	7
2.4	Assumptions, dependencies and constraints	7
3	Specific Requirements	8
3.1	External Interface Requirements	8
3.1.1	User Interfaces	8
3.1.2	Hardware Interfaces	10
3.1.3	Software Interfaces	10
3.1.4	Communication Interfaces	10
3.2	UML modeling	10
3.2.1	Use Case diagram	10
3.2.2	Class diagram	14
3.2.3	Activity diagrams	15
3.3	Functional Requirements	16
3.4	Performance Requirements	16
3.5	Design Constraints	16
3.5.1	Standard compliance	16
3.5.2	Hardware limitations	16
3.5.3	Any other constraint	16
3.6	Software System Attributes	16
3.6.1	Reliability	16
3.6.2	Availability	16
3.6.3	Security	16
3.6.4	Maintainability	16
3.6.5	Portability	16
4	Formal Analysis Using Alloy	17
5	Effort Spent	18
6	References	19
6.1	Software and Tools	19

List of Figures

1	Login	8
2	Select solution	8
3	Visualize schedule	9
4	Use Case diagram	10
5	General class diagram	14
6	Registration diagram	15

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] allows a Person to use the app in his/her device default language
- [G2] allows a Person to become a registered User after providing credentials
- [G3] allows a Person to log in
- [G4] allows a new User to access the welcome page
- [G5] allows User to add credit card data
- [G6] allows User to add driving license
- [G7] allows a User to set personal preferences:
 - [G7]#1 specify his/her preference for eco-friendly solution
 - [G7]#2 set break time windows, either flexible or fixed
 - [G7]#3 set time slot in which the use of specific transportation means should be avoided
 - [G7]#4 set a minimum distance below which a specific transportation mean should be avoided
 - [G7]#5 set a maximum distance beyond which a specific transportation mean should be avoided
 - [G7]#6 set a specific transportation mean as permanently disabled
- [G8] allows User to create a new appointment, specifying its details
- [G9] allows User to visualize his/her appointments, updating its details

- [G10] allows User to specify the details of appointments: date and time, location, type of appointment, passengers, baggage
- [G11] during the specification of details of an appointment:
 - [G11]#1 allows User to visualize suggested travel solutions and to choose one of them
 - [G11]#2 before the choice, allows User to set a preferred transportation mean
 - [G11]#3 allows User to invite other persons to the appointment
 - [G11]#4 allows User to specify if he/she owns a ticket, either ordinary or a pass (eventually specifying the deadline)
- [G12] allows User to buy ticket for an appointment: the ticket can be either ordinary or a season pass
- [G13] allows User to delete an existing appointment
- [G14] allows User to visualize daily/weekly schedule
- [G15] allows User to invite other persons to his/her appointment
- [G16] allows User to locate nearest vehicle of a vehicle sharing system, if that is the transport means of choice of an incoming appointment
- [G17] allows User to being redirected to the company's app to proceed with the renting
- [G18] show suggestions based on: traffic, weather conditions/forecast, strikes, type of appointment, baggage, passengers
- [G19] notify the User about incoming appointments
 - [G19]#1 asking for confirmation on the transportation means previously planned
 - [G19]#2 in case no transportation mean was selected, suggesting one
 - [G19]#3 communicating eventually complications (bad weather, traffic) and suggesting more feasible solutions

1.3 Definitions, Acronyms, Abbreviations

1.3.1 Definitions

- *welcome page*:

- *personal preferences*:

1.3.2 Acronyms

1.3.3 Abbreviations

1.4 Revision history

1.5 Reference documents

1.6 Document Structure

2 Overall Description

2.1 Product perspective

2.2 Product functions

TODO: here will be included a description

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, Travlendar+ 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

2.4 Assumptions, dependencies and constraints

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

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

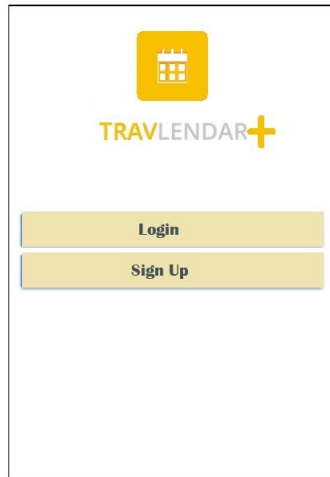


Figure 1: Login



Figure 2: Select solution

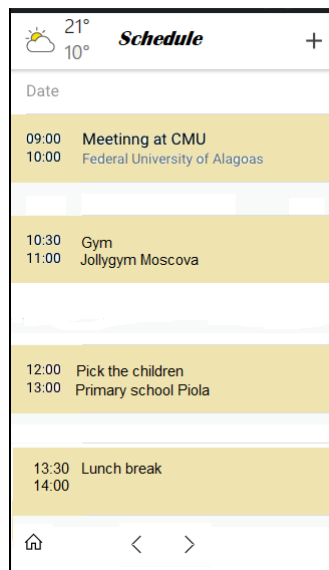


Figure 3: Visualize schedule

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communication Interfaces

3.2 UML modeling

3.2.1 Use Case diagram

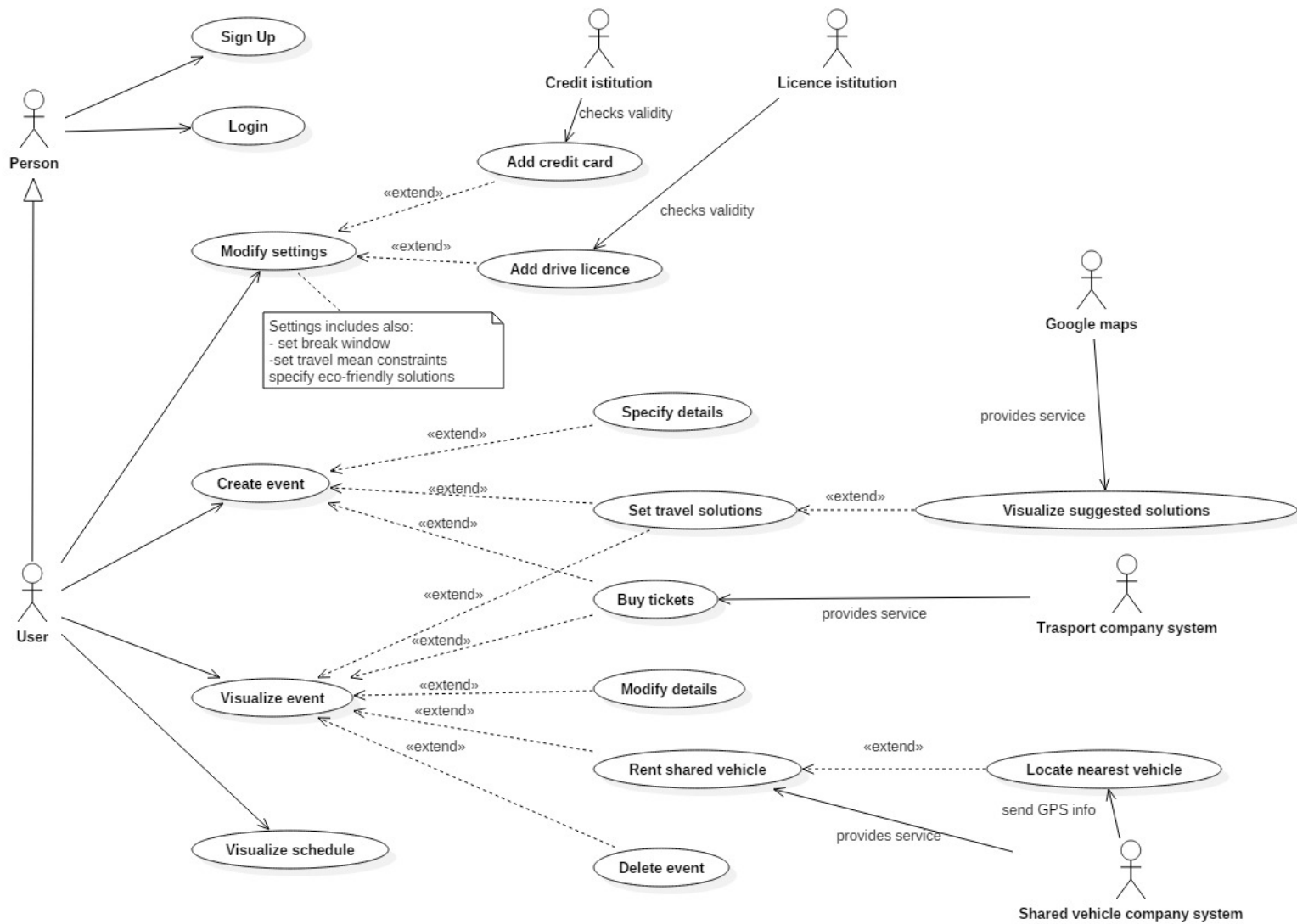


Figure 4: Use Case diagram

User creates event

Use case:	User creates an event
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: <ul style="list-style-type: none">• 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:	<p>User insert sequentially the following information:</p> <ul style="list-style-type: none">• Credit card;• Driving licence;• Break time windows;• Interest for Eco-friendly solutions on the travel means.• Constraints on travel means. <p>The app, for each step, check the info and send a confirmation; The app redirects the user to the main page.</p>
Secondary flows:	<p>User skips to specify one or more information that could be specified later in the settings.</p> <p>The app notifies the user about the missing information and redirects anyway the user to the main page.</p>
Exceptions:	<p>user inserts inconsistent information (incorrect credid-card/licence information, break time shorter than 30 minutes);</p> <p>The app allerts the user and asks him to insert again the info.</p>
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 event must exist;
Description:	<p>user visualize an appointment and eventually:</p> <ul style="list-style-type: none">• 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

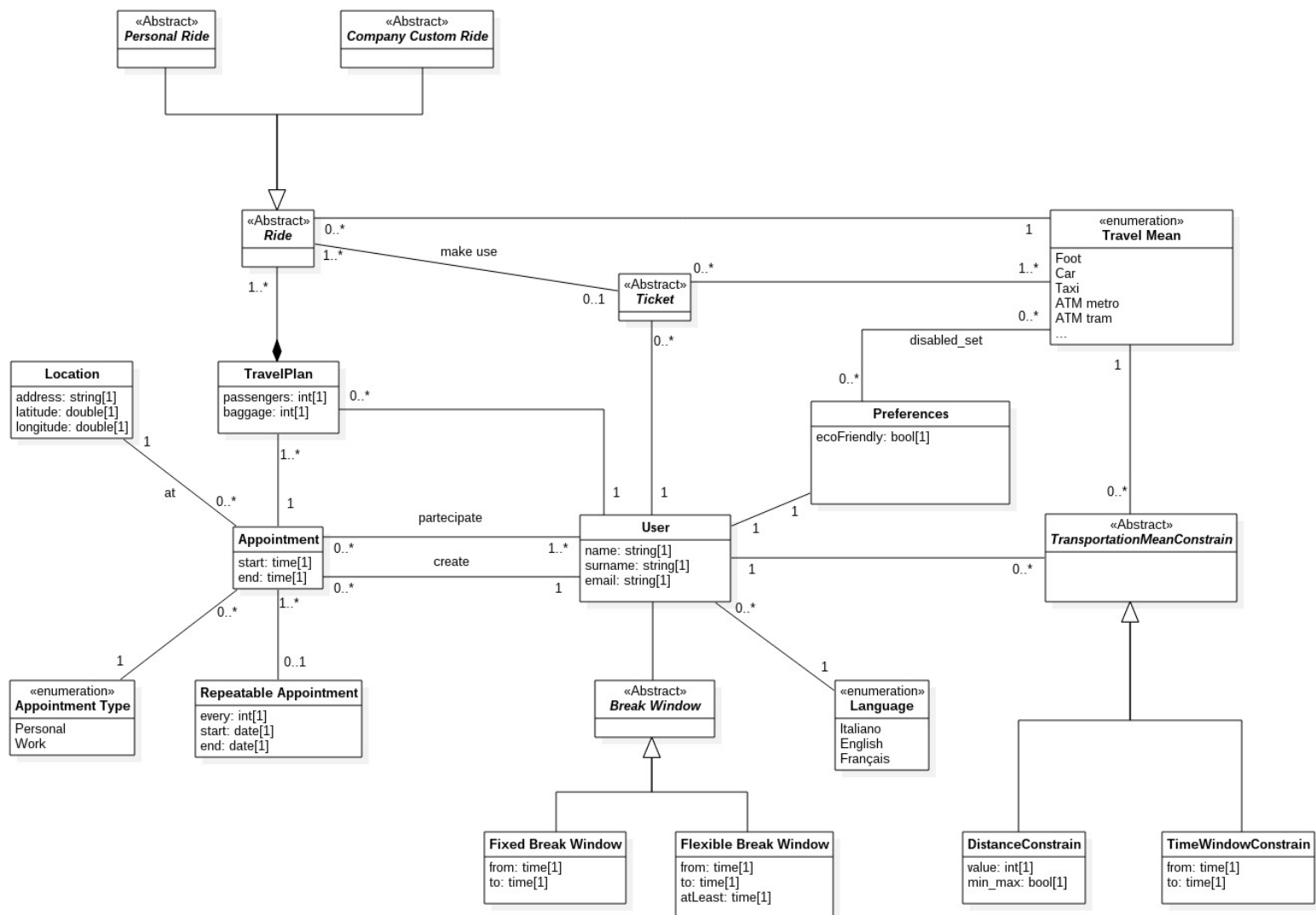


Figure 5: General class diagram

3.2.3 Activity diagrams



Figure 6: Registration activity diagram

- 3.3 Functional Requirements**
- 3.4 Performance Requirements**
- 3.5 Design Constraints**
 - 3.5.1 Standard compliance**
 - 3.5.2 Hardware limitations**
 - 3.5.3 Any other constraint**
- 3.6 Software System Attributes**
 - 3.6.1 Reliability**
 - 3.6.2 Availability**
 - 3.6.3 Security**
 - 3.6.4 Maintainability**
 - 3.6.5 Portability**

4 Formal Analysis Using Alloy

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>