My Taxi Service

Requirements **A**nalysis and **S**pecification **D**ocument

Authors:

Andrea DONATI {andrea4.donati@mail.polimi.it}

Gabriele CARASSALE {gabriele.carassale@mail.polimi.it}

Manuel DELEO {manuel.deleo@mail.polimi.it}

Prof: Elisabetta Di Nitto

Milan, November 6, 2015

Contents

1 Introduction

1.1 Purpose

The goal of this Requirement Analysis and Specification Document (RASD) is to understand, along with the stakeholders, the purposes of the myTaxiService project and the functionalities that should be implemented in the system. Moreover, this document will be used by developers in the further stages of the project, such as design, testing and implementation.

1.2 Overall description of the problem

The aim of myTaxiService project is to develop a web-based application which shall give users the opportunity to request taxi rides in a fast and easy way. The system should manage taxi queues splitting the city in taxi zones, thus guaranteeing a better allocation of taxis to users.
Passengers should be able to tell the system their position so that the system can inform them about the waiting time and the taxi code. Users can also reserve a ride, by specifying origin and destination, at least two hours earlier.
The system should allow users to share a ride with other people by arranging the route of the taxi, defining the fee for passengers and informing both the driver and the passengers.
The architecture of the system should give the possibility to implement additional features thanks to programmatic interfaces.

1.3 Goals

List of goals of myTaxiService system:

- [**G1**] Users should be able to sign up to the system;
- [**G2**] Users should be able to login to the system;
- [**G3**] Users should be able to request a taxi ride;
- [**G4**] Users should be able to know about their taxi code and the waiting time;
- [**G5**] Users should be able to reserve a ride in advance;
- [**G6**] Users should be able to start a shared ride;
- [**G7**] Taxi drivers should be able to know when they have to take care of a request and to accept or reject it;
- [**G8**] Taxi drivers should be able to confirm their availability in order to be inserted in the taxi queue;

1.4 Definitions, acronyms and abbreviations

1.4.1 Definitions

- **User - Passenger:** person who wants to make use of the system. A user should be able of taking advantage of the functionalities listed in the goals section.
- **Taxi driver:** person who owns a taxi and a license. A taxi driver should be able to use the functionalities listed in the goals section.

- **Taxi ride:** ride which goes from an origin to a destination, requested by a specific user and managed by a taxi driver.
- **Taxi zone:** portion of the city which is 2 km^2 large. Each of these zones is associated to a queue of taxis.
- **Taxi queue:** ordered list in which every available taxi in the associated zone is stored.
- **Reserved ride:** ride reserved in advance by a user.
- **Shared ride:** ride created by a user who wants to share the ride and its cost with others.

1.4.2 Acronyms

- **API:** Application Programming Interface;
- **ETA:** Estimated Time of Arrival;
- **OS:** Operating system;

1.4.3 Abbreviations

- **[Gn]:** n-th goal;
- **[Rn]:** n-th functional requirement;

1.5 References

- IEEE Standard For Requirements Specification, 20-10-1998, IEEE Computer Society;
- Specification Document: Assignments 1 and 2 (RASD and DD).pdf

1.6 Overview

In the rest of the document, an overall description of the software-to-be, along with the stakeholders, the interfaces and the assumptions which have been made, will be presented. In particular, we'll focus on the actors, the domain properties and the goals of the project and, consequently, on the requirements derived from the latter. Lastly, we'll use Use Case, Class, Sequence and State Chart Diagrams to model the software system.

2 Overall Description

2.1 Regulatory policies

The system has to respect all the laws of the nation and the city. In particular, users and taxi drivers have to read, accept and respect terms and conditions imposed by the law and by the stakeholder.
For instance, as already mentioned, taxi drivers must own a regular driving license and a regular taxi license.
Concerning this, in the following chapters we'll introduce the administrator, an actor that has the role of managing the admission and the rejection of users and taxi drivers.

2.2 Stakeholder

The stakeholder for the system is the government of a large city. Its aim is to optimize and simplify the management of taxi rides in the city. The system-to-be, in fact, will offer a web and a mobile application that will simplify the use of the taxi service, both to users and to taxi drivers. At the same time, the system will optimize the service by dividing the city in zones, creating apposite queues of taxis and computing, thanks to GPS-based information, which taxi driver shall serve a determinate user.

2.3 User characteristics

- Users of the system and taxi drivers must be able to access a web browser or the mobile application;
- Taxi drivers must have a regular driving license and a regular taxi license. They must also own a taxi and download the mobile application in order to receive notifications and send GPS information;

2.4 Domain properties

The following properties are assumed to hold in the domain:

- A user can be served only by one taxi at a time;
- The waiting time received by a user is indeed real;
- When a taxi driver confirms his availability, he doesn't leave the zone he has been assigned to;
- If a taxi rider accepts a request, he doesn't withdraw its decision;
- If a user requests a taxi, he doesn't withdraw its decision;
- When a user reserves a ride, there is always at least one taxi available at the requested time of departure;
- GPS information about taxis' position is always available;
- All taxis are equal and can carry up to four passengers;

2.5 Assumptions

The specification document wasn't complete, so we made these assumptions about the problem:

- Unregistered users should be able to sign into the system so that they don't have to specify their details every time;
- Payments are entirely managed by the system and the user is able to pay online with credit card;
- The user, at the time of registration, has to accept terms, conditions and rules imposed by the stakeholder; while using the service, the user has to respect these conditions;
- Prices of taxi rides are defined by the system both in the case of ordinary ride and in the case of a shared ride;
- If a user enables the taxi sharing option, he must also specify the time of departure;
- If a user enables the taxi sharing option, he waits for other users until the exact time of departure;

2.6 Future implementations

- ETA Update: a real-time update of the estimated time of arrival could be implemented. Following the dispatch of a request, the user could see the taxi's actual position on the map and the updated time of arrival;
- Taxi driver rating system: users could rate their trip with a specific taxi driver so that the system could present to other users how every driver works. The rating could be divided in fields, such as politeness, availability, punctuality and safety;
- Expansion of city zones: the system could be expanded in order to bring the service to the peripheral parts of the city. In this areas, city zones could be enlarged due to the smaller number of citizens who could request the service;
- Different kinds of taxis: users, at the time of request, could choose their preference in terms of car. For instance, they could request a limousine for a luxurious ride or a small taxi for a cheap one. Users could also indicate which car they own when sharing their rides;

3 Requirements

3.1 Functional Requirements

The following requirements are inferred from the domain properties, the assumptions and from the specific goals of the system:

- [**G1**] Users should be able to sign up to the system
  - [**R1**] The system has to provide a function that allows all unregistered users to sign up;
  - [**R2**] If the username selected by the guest during the registration already exists, the system must ask the user to change it
  - [**R3**] If some required fields are empty during the registration, the system must ask the user to fill them
  - [**R4**] If password and password confirmation inserted by the guest during the registration are different, the system must ask the guest to insert them again
- [**G2**] Users should be able to login to the system
  - [**R5**] The system has to provide a function that allows users to log into the web application and the mobile app
  - [**R6**] If username or password inserted by the user during login are incorrect, the system must ask the user to try again
- [**G3**] Users should be able to request a taxi ride
  - [**R7**] The system has to provide a function that allows users to request a ride by specifying the origin and the destination of the ride
  - [**R8**] If the address provided by the user is outside city boundaries where the service is active, the system must ask the user to insert a valid address
- [**G4**] Users should be able to know about their taxi code and the waiting time
  - [**R9**] The system has to send a notification, containing the taxi code and the waiting time, to users who requested a ride
  - [**R10**] The system must send the notification by email if the user accesses the system from the browser or by a mobile notification if the user accesses the system from the mobile application
- [**G5**] Users should be able to reserve a ride in advance
  - [**R11**] The system has to provide a function that allows users to reserve a ride in advance, by specifying the time of departure, the origin and the destination of the ride
  - [**R12**] If the address provided by the user is outside city boundaries where the service is active, the system must ask the user to insert a valid address
  - [**R13**] If the reservation occurs less than two hours before the ride, the system must notify the user asking to change the reservation time
  - [**R14**] The system must allocate a taxi 10 minutes before the meeting time with the user

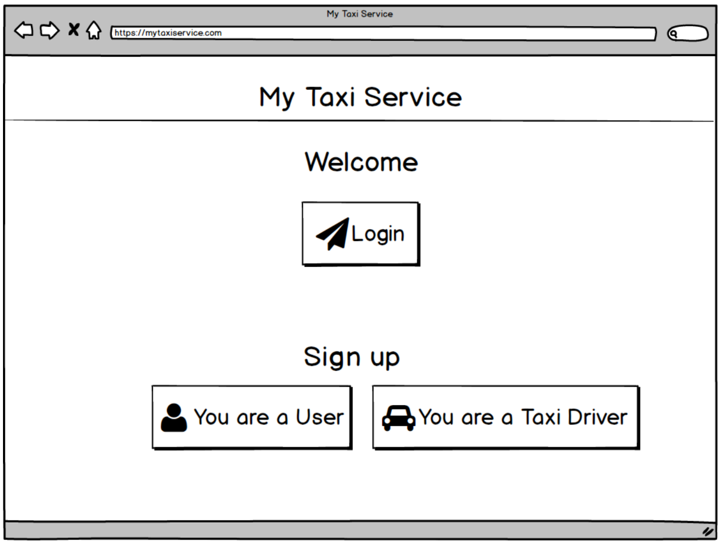
- [**G6**] Users should be able to start a shared ride
  - [**R15**] The system has to provide a function that allows users to share a ride with others, by specifying the time of departure, the origin and the destination of the ride
  - [**R16**] If the origin and destination addresses provided by the user are outside city boundaries where the service is active, the system must ask the user to insert valid addresses
- [**G7**] Taxi drivers should be able to know when they have to take care of a request and to accept or reject it
  - [**R17**] The system must send a notification to the Taxi Driver in the first position in the zone's taxi queue when a user requests a ride in that zone
- [**G8**] Taxi drivers should be able to confirm their availability in order to be inserted in the taxi queue
  - [**R18**] The system has to provide a function that allows taxi drivers to confirm their disponibility from the mobile application
  - [**R19**] If the taxi driver is outside of the city boundaries when he confirms his availability, the system must notify the error asking the driver to go in a correct zone
  - [**R20**] When the taxi driver confirms his availability, the system must insert him at the end of the taxi queue correspondent to the city zone where the driver is in that moment

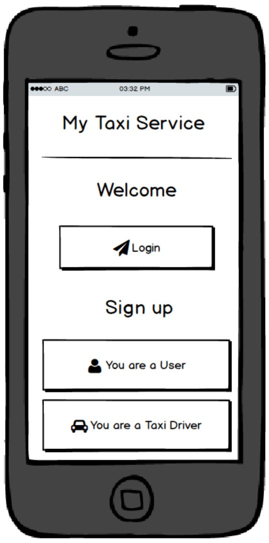
3.2 Non-Functional Requirements

3.2.1 User Interface

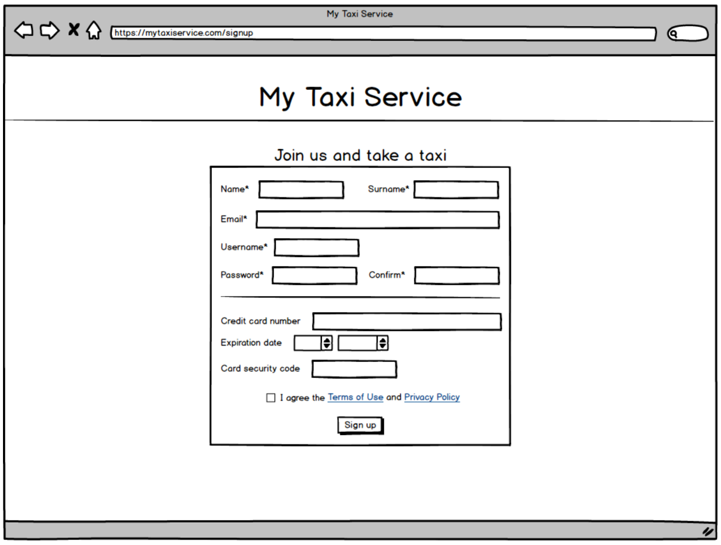
The system is implemented as a web service so that everyone with an internet connection and a web browser can take a taxi. User will be able to use both a browser and a smartphone interface, on the other hand the smartphone interface for administrator was considered useless.
For this reasons the user interface has been developed for both web browsers and smartphones, while the administrator interface was developed only for web browsers.

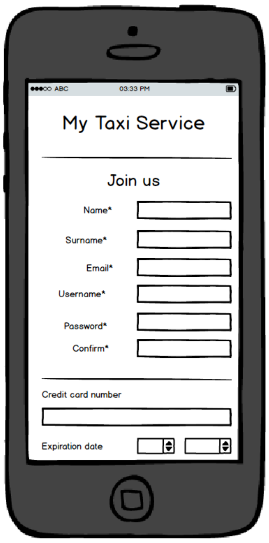
When the guest connects to the system he is shown a welcome page, from which he can decide to log in or register as a user or taxi driver.



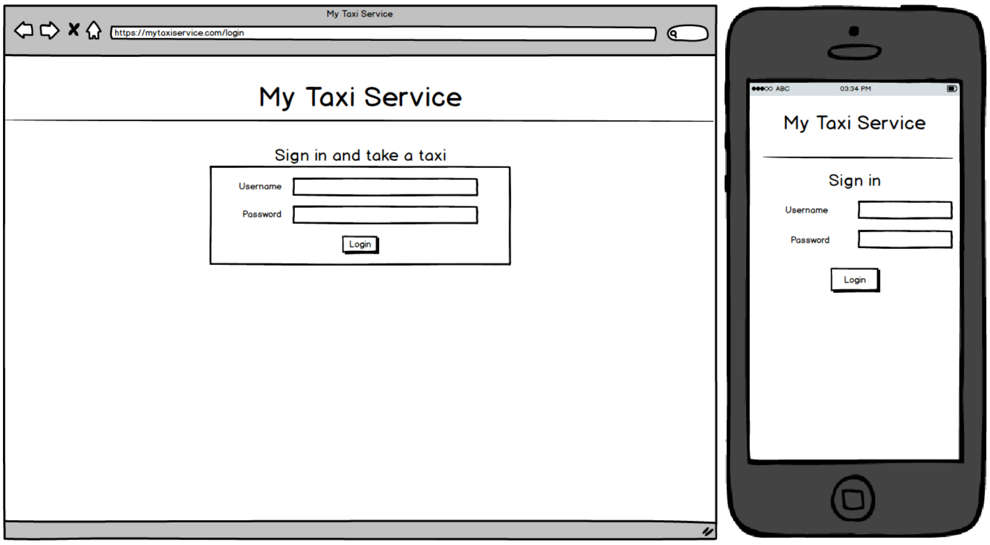


In the registration page is requested only the minimum information necessary to start to use the full service.



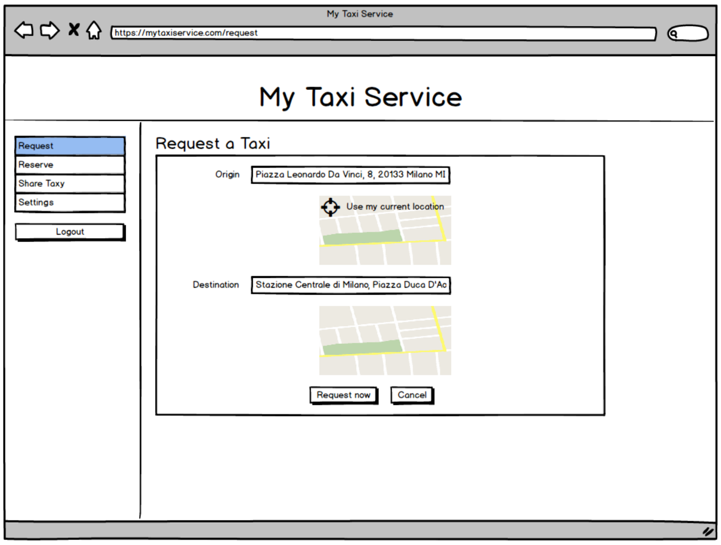


Login page will have a easier and minimal design.



(From now on, it will be shown only one interface at a time)

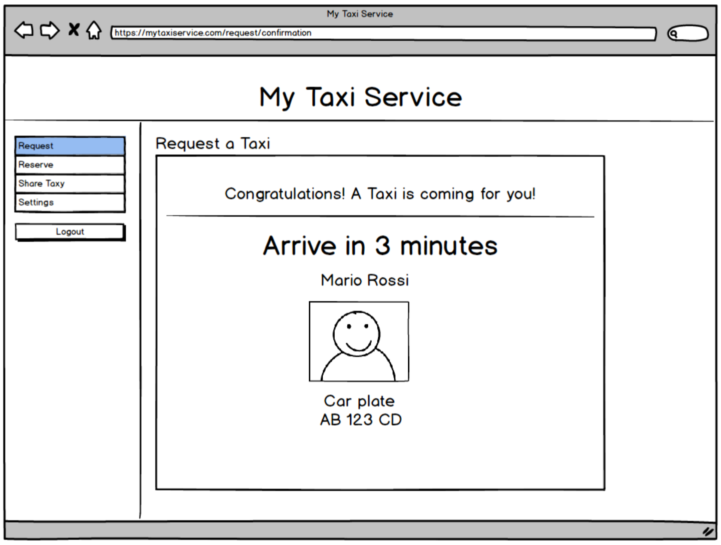
After logging in or registering, if it is a user, the system displays the page ”Request” from where you can quickly take a taxi.



Origin: Piazza Leonardo Da Vinci, 8, 20133 Milano MI

Destination: Stazione Centrale di Milano, Piazza Duca D'Ao

After pressing the button ”Request Now”, the system displays a confirmation message indicating the name of the taxi driver, the plate of the taxi and the estimated time of waiting.



My Taxi Service

Request a Taxi

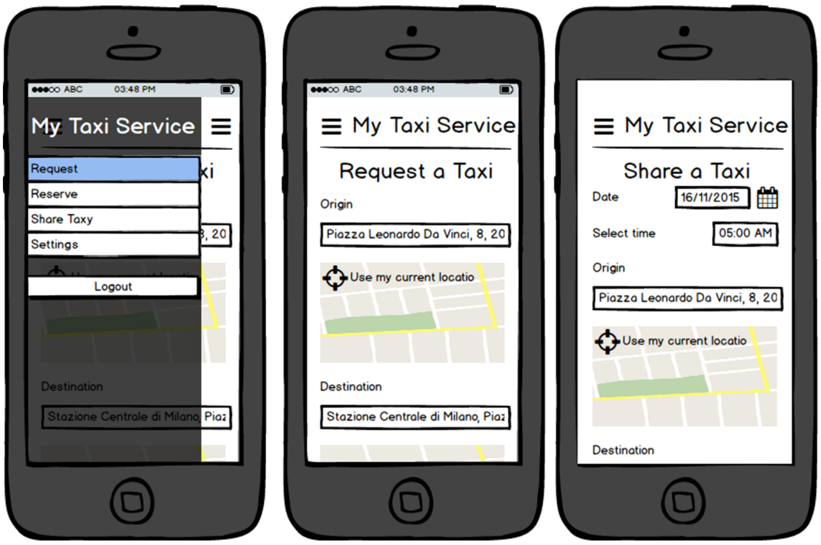
Congratulations! A Taxi is coming for you!

Arrive in 3 minutes

Mario Rossi

Car plate
AB 123 CD

From the left menu you can also access the pages "Reserve", "Share", "Settings" or logout from the system.



My Taxi Service

Request a Taxi

Origin: Piazza Leonardo Da Vinci, 8, 20

Destination: Stazione Centrale di Milano, Piaz

My Taxi Service

Share a Taxi

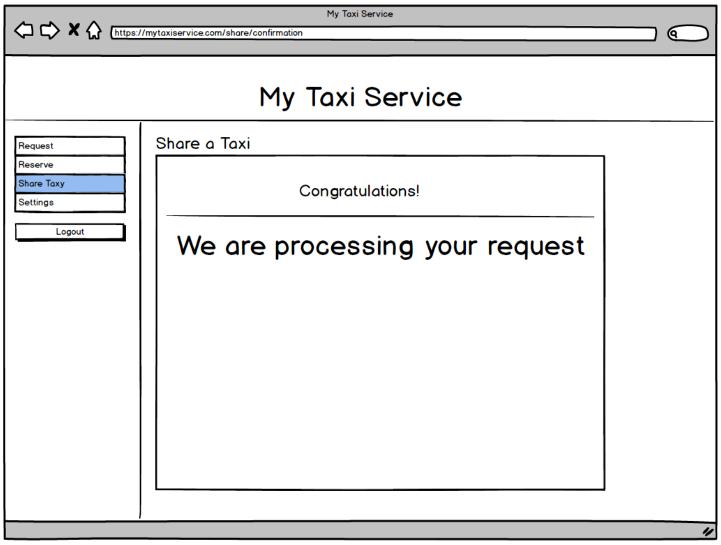
Date: 16/11/2015

Select time: 05:00 AM

Origin: Piazza Leonardo Da Vinci, 8, 20

Destination

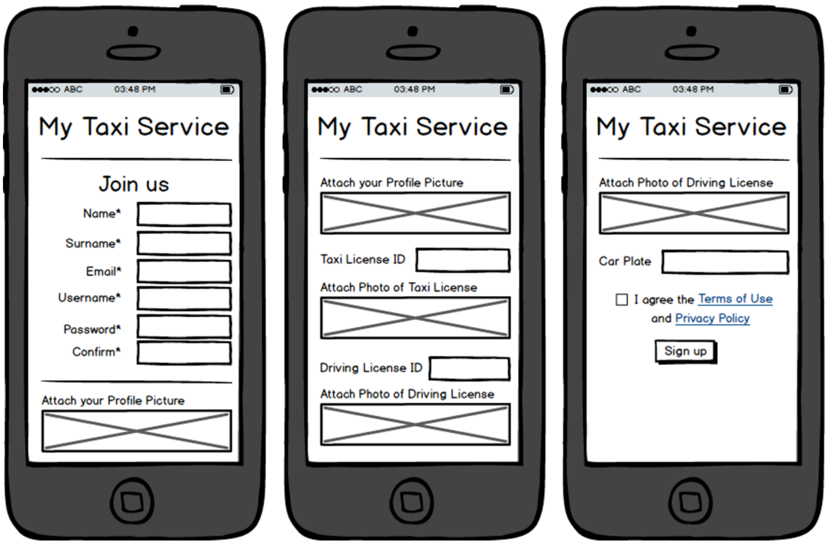
The user receives a confirmation message for the correct reservation or sharing of the ride.



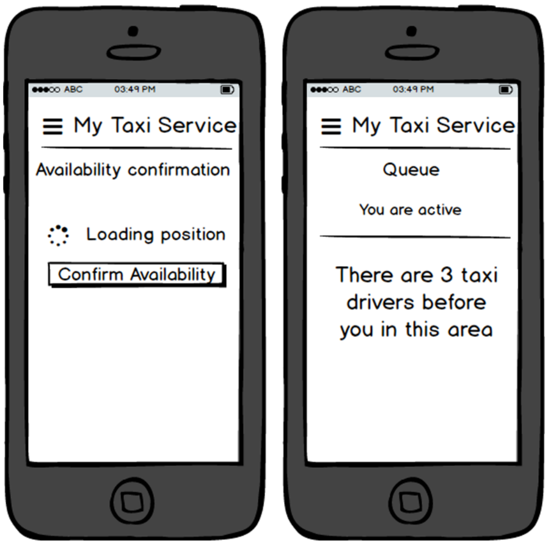
Congratulations!

We are processing your request

The registration page of the taxi driver has a different interface, which will also include the photo of the requested documents.



When the driver connects to the service, he may confirm his availability in the area by clicking on the ”Confirm Availability”. It will appear then a confirmation message with the position in the queue of that area.



●●●○○ ABC 03:49 PM

My Taxi Service

Availability confirmation

Loading position

Confirm Availability

●●●○○ ABC 03:49 PM

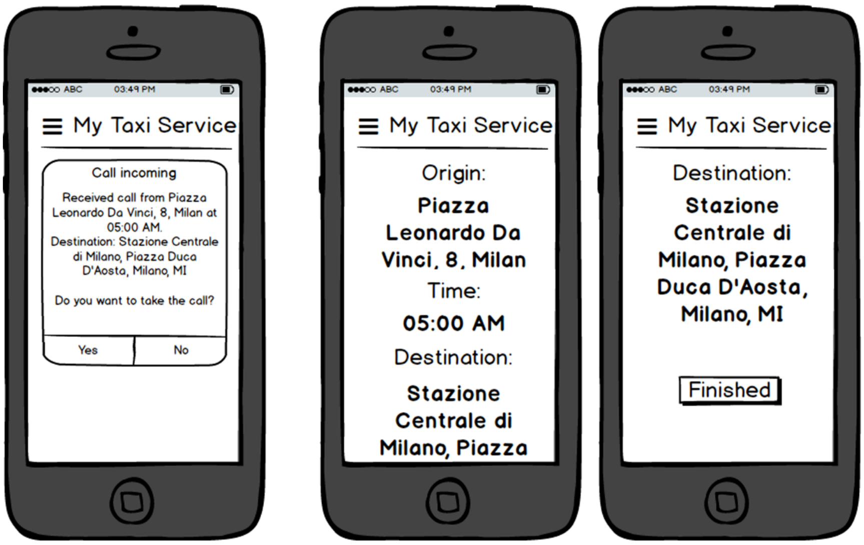
My Taxi Service

Queue

You are active

There are 3 taxi drivers before you in this area

When the driver receives a request, a dialog box will appear. If the driver accepts it, then he will be redirected to the page containing the origin and the destination addresses and, in case it was a ”Reserve” request, also the departure time.
When the ride will be finished, the driver will have to click on the ”Finished” button.



●●●○○ ABC 03:49 PM

My Taxi Service

Call incoming

Received call from Piazza Leonardo Da Vinci, 8, Milan at 05:00 AM.
Destination: Stazione Centrale di Milano, Piazza Duca D'Aosta, Milano, MI

Do you want to take the call?

Yes No

●●●○○ ABC 03:49 PM

My Taxi Service

Origin:

Piazza Leonardo Da Vinci, 8, Milan

Time:

05:00 AM

Destination:

Stazione Centrale di Milano, Piazza

●●●○○ ABC 03:49 PM

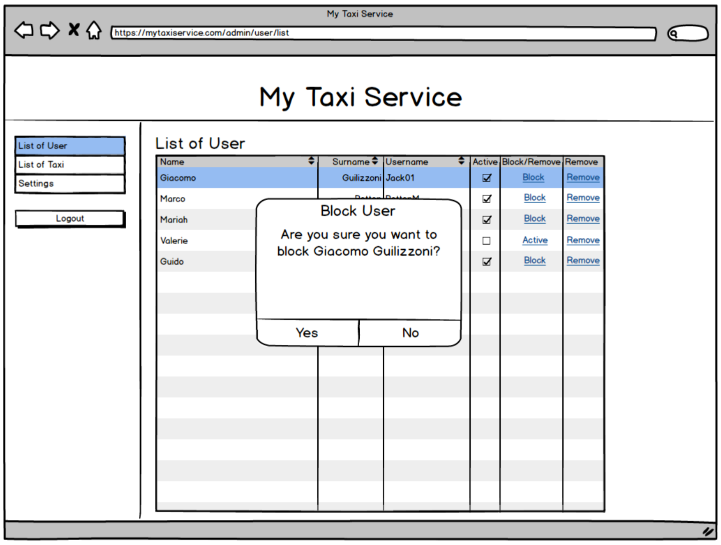
My Taxi Service

Destination:

Stazione Centrale di Milano, Piazza Duca D'Aosta, Milano, MI

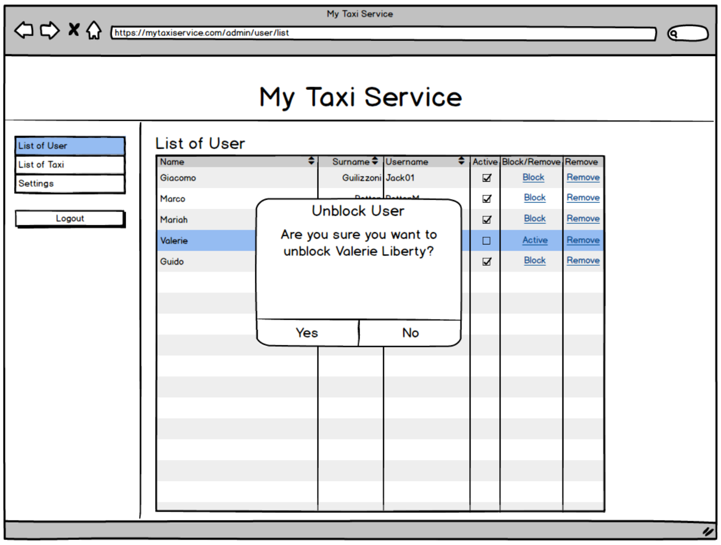
Finished

The administrator can view the list of users and taxi drivers and choose to block, unblock or remove any of them.



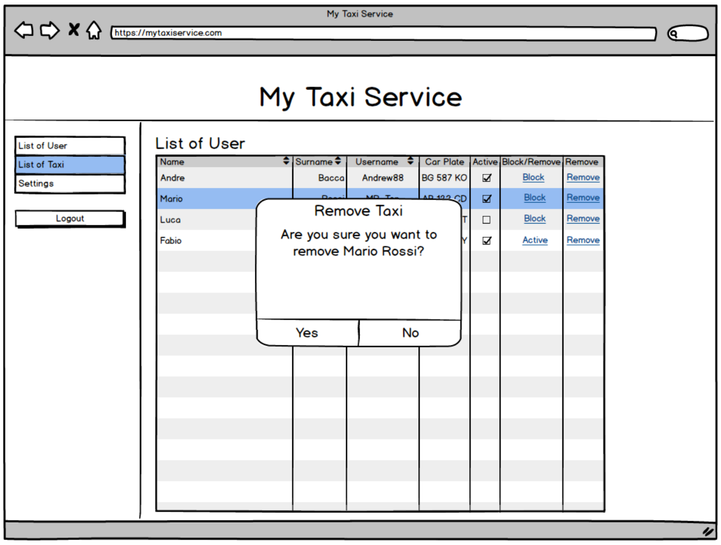
Name	Surname	Username	Active	Block/Remove	Remove
Giacomo	Guilizzoni	Jack01	☑	Block	Remove
Marco			☑	Block	Remove
Mariah			☑	Block	Remove
Valerie			□	Active	Remove
Guido			☑	Block	Remove

Are you sure you want to block Giacomo Guilizzoni?



Name	Surname	Username	Active	Block/Remove	Remove
Giacomo	Guilizzoni	Jack01	☑	Block	Remove
Marco			☑	Block	Remove
Mariah			☑	Block	Remove
Valerie			□	Active	Remove
Guido			☑	Block	Remove

Are you sure you want to unblock Valerie Liberty?



My Taxi Service

https://mytaxiservice.com

My Taxi Service

- List of User
- List of Taxi
- Settings

Logout

List of User

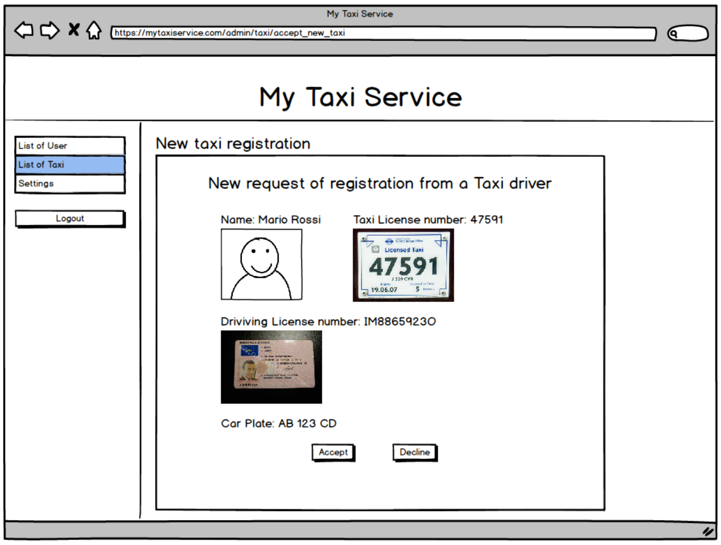
Name	Surname	Username	Car Plate	Active	Block/Remove	Remove
Andre	Bacca	Andrew88	BG 587 KO	☑	Block	Remove
Mario	Rossi	[illegible]	AB 123 CD	☑	Block	Remove
Luca			T	□	Block	Remove
Fabio			Y	☑	Active	Remove

Remove Taxi

Are you sure you want to remove Mario Rossi?

Yes No

The administrator can accept or reject the registration of a taxi driver and see the uploaded documents.



My Taxi Service

https://mytaxiservice.com/admin/taxi/accept_new_taxi

My Taxi Service

- List of User
- List of Taxi
- Settings

Logout

New taxi registration

New request of registration from a Taxi driver

Name: Mario Rossi

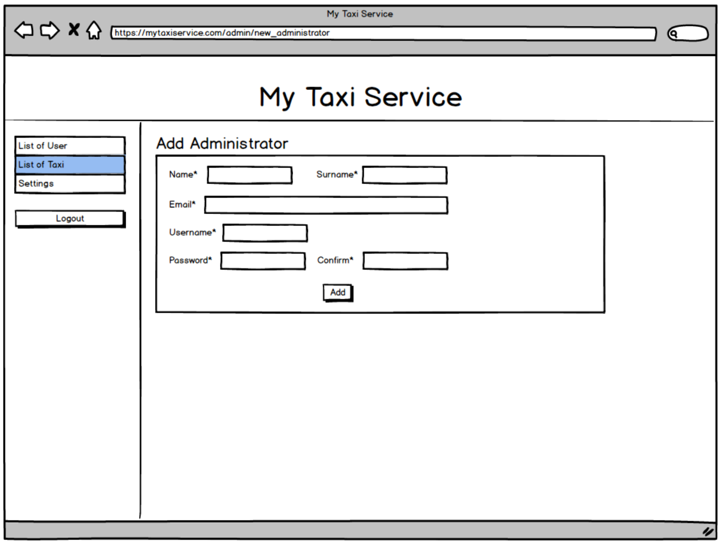
Taxi License number: 47591

Driviving License number: IM88659230

Car Plate: AB 123 CD

Accept Decline

The administrator can add another administrator.



3.2.2 API, Software and Hardware Interfaces

Users have the possibility (while taxi drivers are forced) to interface with the system through a mobile application. This implication limits the app to be developed for iOS, Android and Windows operating systems, i.e. the main OSs for mobile devices. In addition, the application's performance has to be adjusted in accordance to the hardware present on tablets and mobile devices which can actually be found on the market.
There aren't any constraints on the development of the web application accessed by the browser.
The system will make use of Google Maps API. This interface will permit the computation of shared taxi routes, the specification of precise origins and destinations based on Google's points of interest.

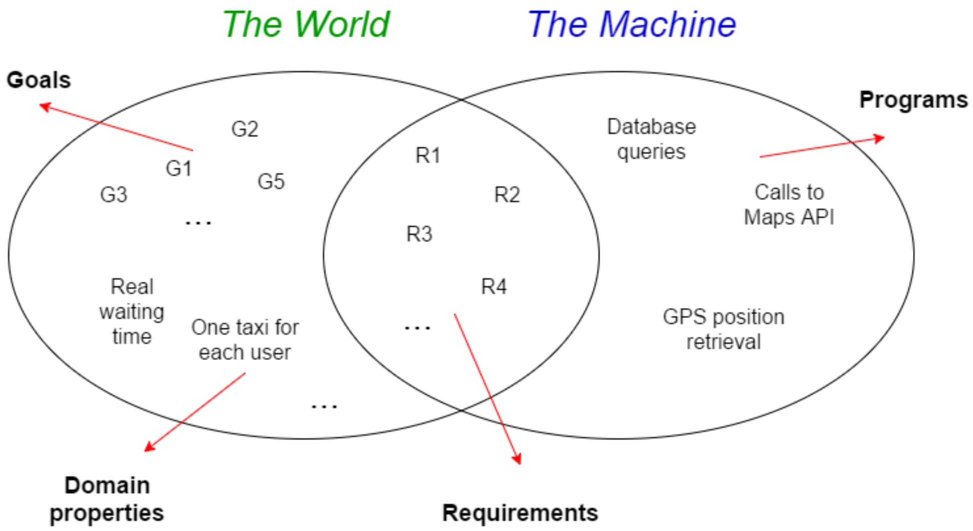
3.2.3 Other

The stakeholder requested the system to have programmatic interfaces in order to offer additional contents. An additional requirement is thus needed: administrators of the system should be able to make use of an API which gives them the possibility to access to system's data, such as the list of taxi drivers, the list of city zones and the taxi queues.

3.3 The world and the machine

We've followed the Jackson and Zave approach to create this diagram which represents world phenomena, machine phenomena and shared phenomena, using the goals (Section 1.3), domain properties (Section 2.4), programs and requirements (Section 3.1) we've found:

[ht]


The World
The Machine
Goals
G2
G1
G5
G3
...
Real waiting time
One taxi for each user
...
Domain properties
R1
R2
R3
R4
...
Requirements
Database queries
Programs
Calls to Maps API
GPS position retrieval

4 Specifications

4.1 Actors

- **Guest** This is an user not recognized by the system yet. The guest can only register or login to access the user functionalities.
- **User** Once logged, the user can access all the main services such as requesting or sharing a ride.
- **Taxi driver** He owns a taxi with a regular license.
- **Administrator** He manages the system and all the registered users and taxi drivers.

4.2 Scenarios

In order to find the requirements of the system and to meet the goals of the project, we've written down some possible scenarios in which the previously defined actors could run into.

4.2.1 Registration

Josh comes to London on vacation and doesn't know the city's transport system well. He finds myTaxiService mobile application on the Android application store and downloads it. Once opened, the app asks him to sign up, by passing his name, surname, email and payment method. Josh then chooses his username and password. He can thus start using the service just by logging in every time he wants to call for a taxi during his vacation.

4.2.2 [G1, G2] Request

Paul is always late and he has to go to the other part of the city in the shortest time possible. Thus he decides to use myTaxiService web application to call a taxi. First, he logs into the system by inserting his username and password and then he requests a taxi. After some time Paul is notified about the confirmation of the request with the code of the incoming taxi and the waiting time.

4.2.3 [G3] Reservation

Harry has to take a flight from Rome to Milan tomorrow in the early morning, but he doesn't know how to reach the airport. He could go by bus, train or underground, but, unfortunately, they are not working at 5:00 AM.

For this reason he decides to book a taxi ride on its myTaxiService mobile application so that he can reserve the taxi for the next morning specifying the origin, the destination and the time of the ride.

After some time Harry receives the confirmation of the reservation by the system and he can finally relax going to the cinema.

The next morning he finds the taxi driver waiting for him at his home at 5:00 AM.

4.2.4 [G4] Sharing

Bruce, who lives in the west side of Paris, has to visit his grandmother in the south-east part of the city at 8 PM. He thus decides to share the ride on myTaxiService mobile application so that he can find someone who keeps him company and share the ride's fee at once. Bruce opens his app and enables the taxi sharing option by clicking on the "Share taxi" button. He, then, writes the origin, the destination and the time of the departure and waits for somebody who wants to share the ride with him. At 8 PM he checks its smartphone for notifications and he notices that 3 other users want to share the ride with him. The system, then, gives him indications about the route that he has to take and about his personal fee. Finally, he starts his trip.

4.2.5 [G5, G6] Availability confirmation

William wakes up at 6 AM and goes to work. Once in his taxi he logs into myTaxiService mobile application and confirms his availability. At 6.30 AM he receives a notification from the app about an incoming request from a user called Stephen. William has to go at Stephen's and bring him to University. He decides to take care of the request and starts the ride. After the ride, William confirms his availability to the system that inserts him in the taxi waiting queue relative to the taxi zone where he is in this moment.

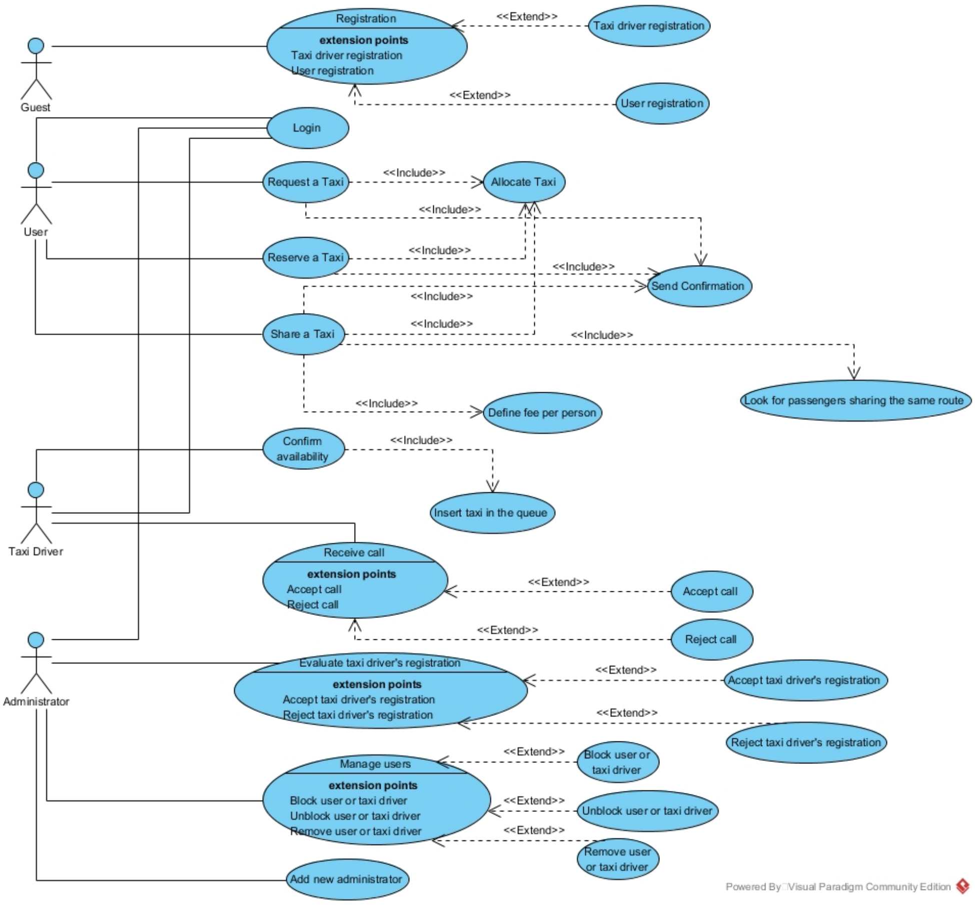
4.2.6 Administrator accept taxi driver's registration

Steven is an administrator and is working in his office. He receives a registration request from a new taxi driver, Mike. He checks his driving license, taxi license and car plate and, then, he adds him into the system.

4.2.7 Administrator block or remove user or taxi driver

Andrew is a system administrator and has received complaints about a user, Manuel, which has made use of a fake payment method. So, he removes Manuel's account from the system so that he cannot access the service anymore.

4.3 Use Case Diagram



4.3.1 User Registration

Actor	Guest
Goal	[G1]
Input Condition	The guest opens the mobile or web application and he has not an account
Flow of Events	• The welcome page is shown • The guest clicks on ’Sign Up’ button • The guest inserts: username, password, password confirmation, name, surname, email, credit card number, card’s expiration date and card’s security number • The guest must accept ”Terms of Use” and ”Privacy Policy” of the system • The guest clicks ’Submit’ • The guest is redirected to the homepage
Output Condition	Registration is successful and the system creates an account for the new user
Exceptions	• The selected username already exists and the system will prompt an error asking to try again • Some required fields are empty, so the guest will be asked to fill them • Password and password confirmation are different, so the system will prompt an error asking to insert them again

4.3.2 Taxi Driver Registration

Actor	Guest
Goal	[G1]
Input Condition	The guest opens the mobile or web application and he has not an account
Flow of Events	• The welcome page is shown • The guest clicks on ’Sign Up’ button • The guest inserts: username, password, password confirmation, name, surname, email, driving license photo, driving license ID, taxi license photo, taxi license ID, car plate ID and finally the profile picture • The guest must accept ”Terms of Use” and ”Privacy Policy” of the system • The guest clicks ’Submit’ • The guest is redirected to the homepage
Output Condition	Registration is successful and the system sends confirmation request of the registration to an administrator
Exceptions	• The selected username already exists and the system will prompt an error asking to try again • Some required fields are empty, so the guest will be asked to fill them • Password and password confirmation are different, so the system will prompt an error asking to insert them again

4.3.3 Login

Actor	User Taxi Driver Administrator
Goal	[G2]
Input Condition	The user owns an account and browses to the homepage
Flow of Events	• The welcome page is shown to the user • The user clicks the 'Login' button • The user fills in the login form with his username and password • The user clicks 'Ok' • The user is redirected to the correct homepage
Output Condition	Login is successful and the system authenticates the user. The system redirects the user to the correct homepage depending on his account type (USer, Taxi Driver, Administrator)
Exceptions	Username or password are incorrect, so the system asks the user to try again

4.3.4 Request a Taxi

Actor	User
Goal	[G3], [G4]
Input Condition	The user is logged in
Flow of Events	• The user clicks on ’Request taxi’ button • The request screen is shown • User inserts the origin and destination address of the ride • The user clicks the ’Send’ button and the request is sent to the system • The user waits for the confirmation of the request by the system • The user is notified with incoming taxi code number and ETA
Output Condition	The system allocates a taxi
Exceptions	• The address provided by the user is outside city boundaries where the service is active, so the system prompts an error asking the user to insert a new address

4.3.5 Reserve a Taxi

Actor	User
Goal	[G5], [G4]
Input Condition	The user is logged in
Flow of Events	• The user clicks on 'Reserve taxi' button • The reservation screen is shown • The user inserts the origin and the destination of the ride, and the time of the reservation • The user clicks the 'Send' button and the reservation request is sent to the system • The user waits for the confirmation of the reservation by the system • The user is notified about the success of the reservation
Output Condition	The system creates the reservation
Exceptions	• The reservation occurs less than two hours before the ride, so the system notifies the user asking to change the reservation time • The address provided by the user is outside city boundaries where the service is active, so the system prompts an error asking the user to insert a new address

4.3.6 Share a taxi

Actor	User
Goal	[G6], [G4]
Input Condition	The user is logged in
Flow of Events	• The user clicks on 'Share taxi' button • The sharing screen is shown • The user inserts the origin and the destination of the ride, and the time of departure • The user clicks the 'Send' button and the sharing request is sent to the system • The user waits for the confirmation of the operation by the system • The user is notified with his personal fee
Output Condition	The system arranges the route for the taxi driver and define the fee for all passengers sharing the taxi
Exceptions	• The destination address is outside city boundaries where the service is active • The address provided by the user is outside city boundaries where the service is active, so the system prompts an error asking the user to insert a new address

4.3.7 Confirm Availability

Actor	Taxi Driver
Goal	[G8]
Input Condition	The taxi driver is logged in
Flow of Events	• The taxi driver clicks on the 'Confirm Availability' button on his mobile application • The taxi driver is notified with a confirmation of the operation • The taxi driver visualizes the information about his position in the taxi queue related to the zone he has been assigned to
Output Condition	The taxi driver is inserted in the waiting queue correspondent to the zone where he is in that moment
Exceptions	The taxi driver is outside of the city boundaries when he confirms his availability, so the system notifies the error asking the driver to go in a correct zone

4.3.8 Accept call

Actor	Taxi Driver
Goal	[G7]
Input Condition	The taxi driver is logged in and he is available
Flow of Events	• The taxi driver is waiting for a call and he is in the homepage of the mobile application • The taxi driver receives a notification of a new ride from the system • He accepts the call • The taxi driver receives the information about the ride like the address and time • The taxi driver goes to pick up the passenger
Output Condition	The system sends a confirmation to the passenger. Taxi driver is no longer available
Exceptions	

4.3.9 Reject call

Actor	Taxi Driver
Goal	[G7]
Input Condition	The taxi driver is logged in and he is available
Flow of Events	• The taxi driver is waiting for a call and he is in the homepage of the mobile application • The taxi driver receives a notification of a new ride from the system • He rejects the call • The taxi driver receives a message saying that he has been moved to the end of the waiting queue • The taxi driver starts waiting for another call
Output Condition	Taxi driver is moved to the end of the waiting queue. Taxi driver is still available
Exceptions	

4.3.10 Accept taxi driver's registration

Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator receives a notification of a new taxi driver's registration request by the system • The administrator checks the validity of all the documents submitted by the taxi driver during the registration process • The administration accepts the registration
Output Condition	The system sends a confirmation to the taxi driver. The taxi driver is now registered in the system and he has his own account
Exceptions	

4.3.11 Reject taxi driver's registration

Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator receives a notification of a new taxi driver's registration request by the system • The administrator checks the validity of all the documents submitted by the taxi driver during the registration process • The administrator can write a message explaining the reasons of the rejection • The administration rejects the registration
Output Condition	The system sends a notification to the taxi driver with the message. The taxi driver is not registered in the system
Exceptions	

4.3.12 Block user or taxi driver

Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator clicks on 'Block Account' button • The administrator selects the account to be blocked • The administrator can write a message explaining the reasons of the block • The administrator confirms and blocks the account of the user or taxi driver
Output Condition	The system sends a notification to the blocked user or taxi driver with the message. The account of the user or taxi driver is now blocked and cannot be used by the owner
Exceptions	

4.3.13 Unblock user or taxi driver

Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator clicks on 'Unblock Account' button • The administrator selects the account to be restored • The administrator can write a message explaining the reasons of the restoration • The administrator confirms and unblocks the account of the user or taxi driver
Output Condition	The system sends a notification to the user or taxi driver with the message. The account of the user or taxi driver is now unblocked and all features are restored
Exceptions	

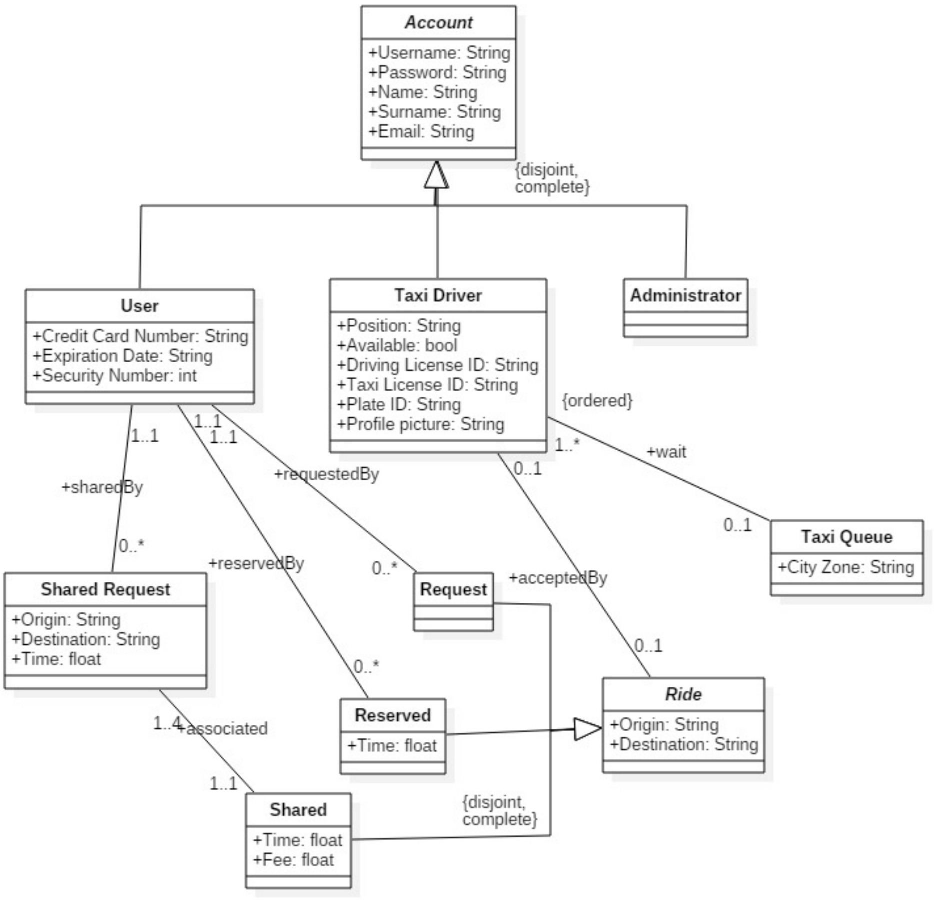
4.3.14 Remove user or taxi driver

Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator clicks on 'Remove Account' button • The administrator selects the account to be removed • The administrator can write a message explaining the reasons of the removal • The administrator confirms and removes the account of the user or taxi driver
Output Condition	The system sends a notification to the removed user or taxi driver with the message. The account of the user or taxi driver is removed and doesn't exist anymore
Exceptions	

4.3.15 Add new administrator

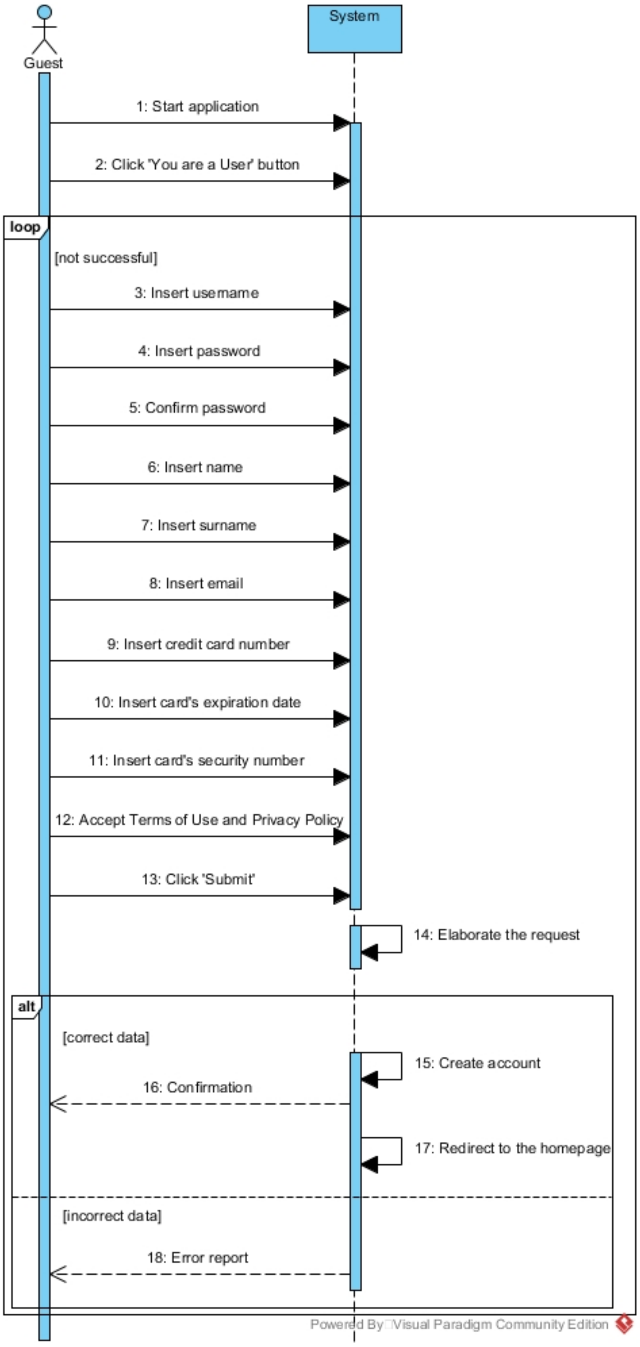
Actor	Administrator
Input Condition	The administrator is logged in
Flow of Events	• The administrator clicks on 'Add new administrator' button • The administrator fills in the registration form for the new administrator with name, surname, email, username, password and password confirmation • The administrator confirms and creates a new account
Output Condition	The system sends a notification to the new administrator with the login details. The administrator is now registered in the system and he has his own account
Exceptions	• Some required fields are empty, so the administrator will be asked to fill them • Password and password confirmation are different, so the system will prompt an error asking to insert them again

4.4 Class Diagram

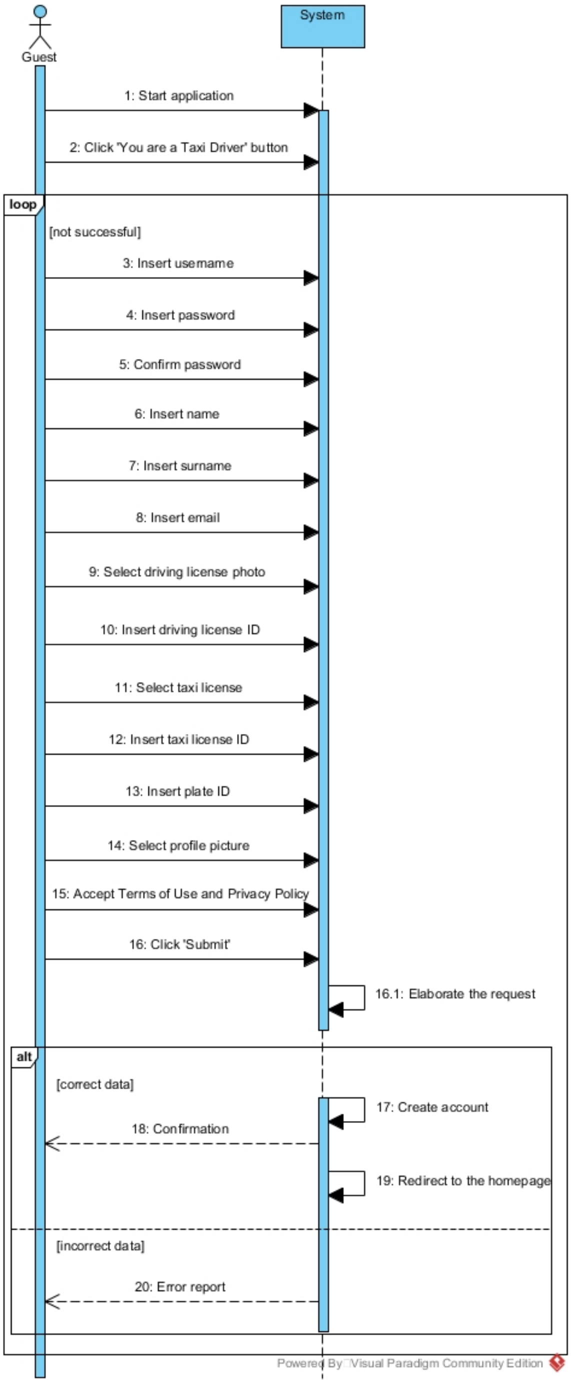


4.5 Sequence Diagram

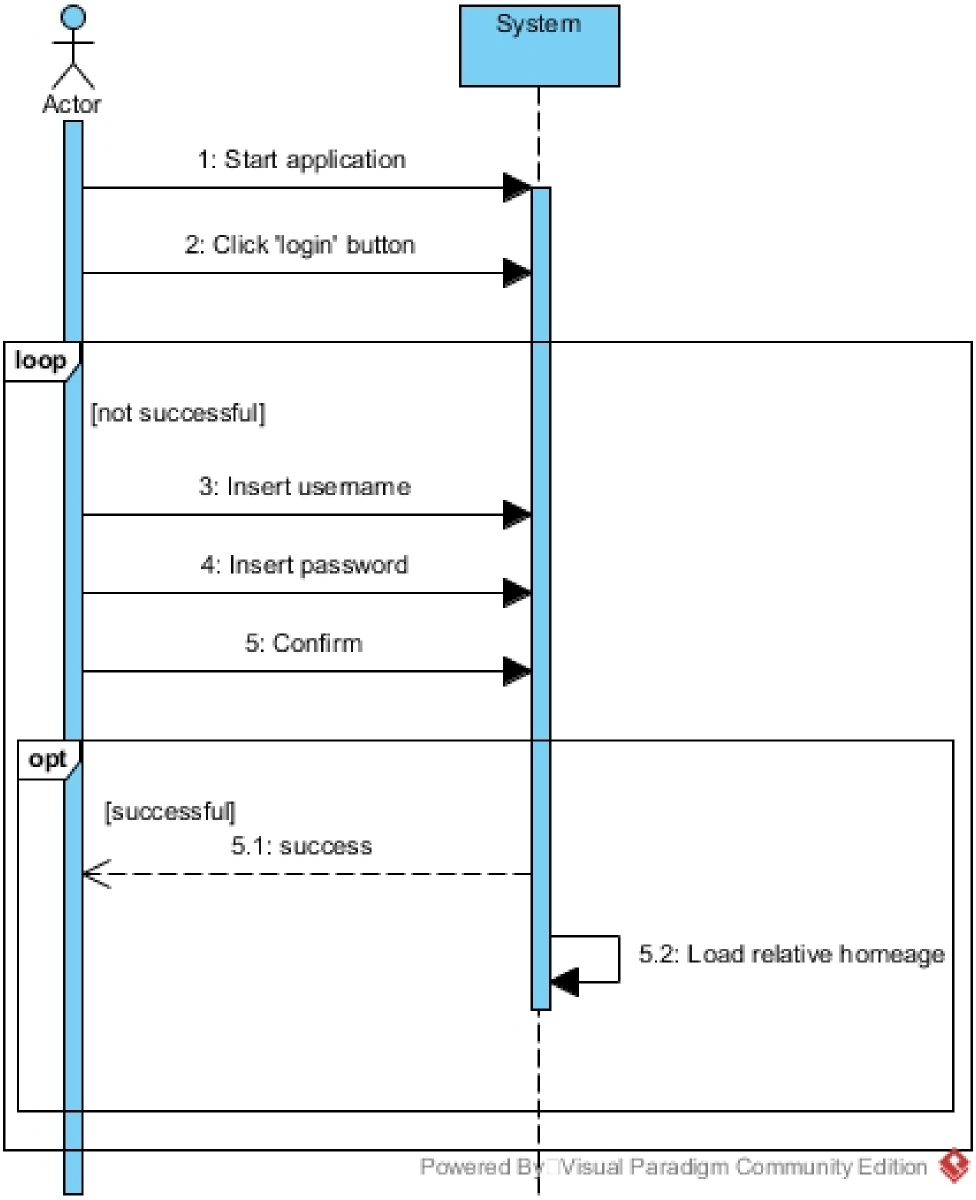
4.5.1 User Registration


Guest
System
1: Start application
2: Click 'You are a User' button
loop
[not successful]
3: Insert username
4: Insert password
5: Confirm password
6: Insert name
7: Insert surname
8: Insert email
9: Insert credit card number
10: Insert card's expiration date
11: Insert card's security number
12: Accept Terms of Use and Privacy Policy
13: Click 'Submit'
14: Elaborate the request
alt
[correct data]
15: Create account
16: Confirmation
17: Redirect to the homepage
[incorrect data]
18: Error report
Powered By Visual Paradigm Community Edition

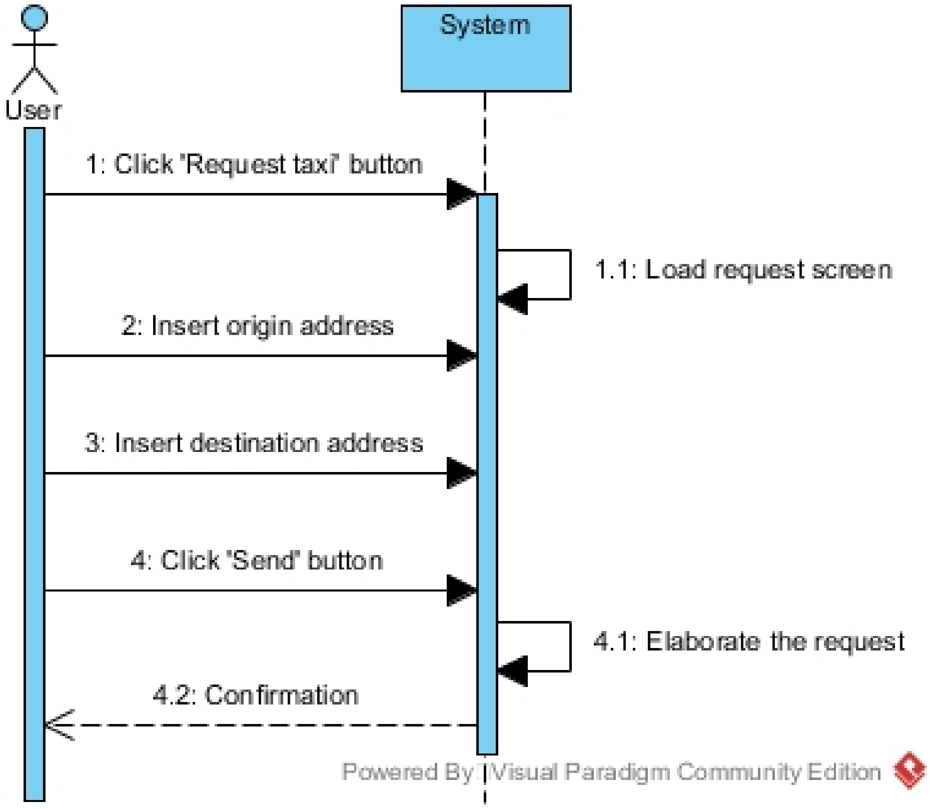
4.5.2 Taxi Driver Registration



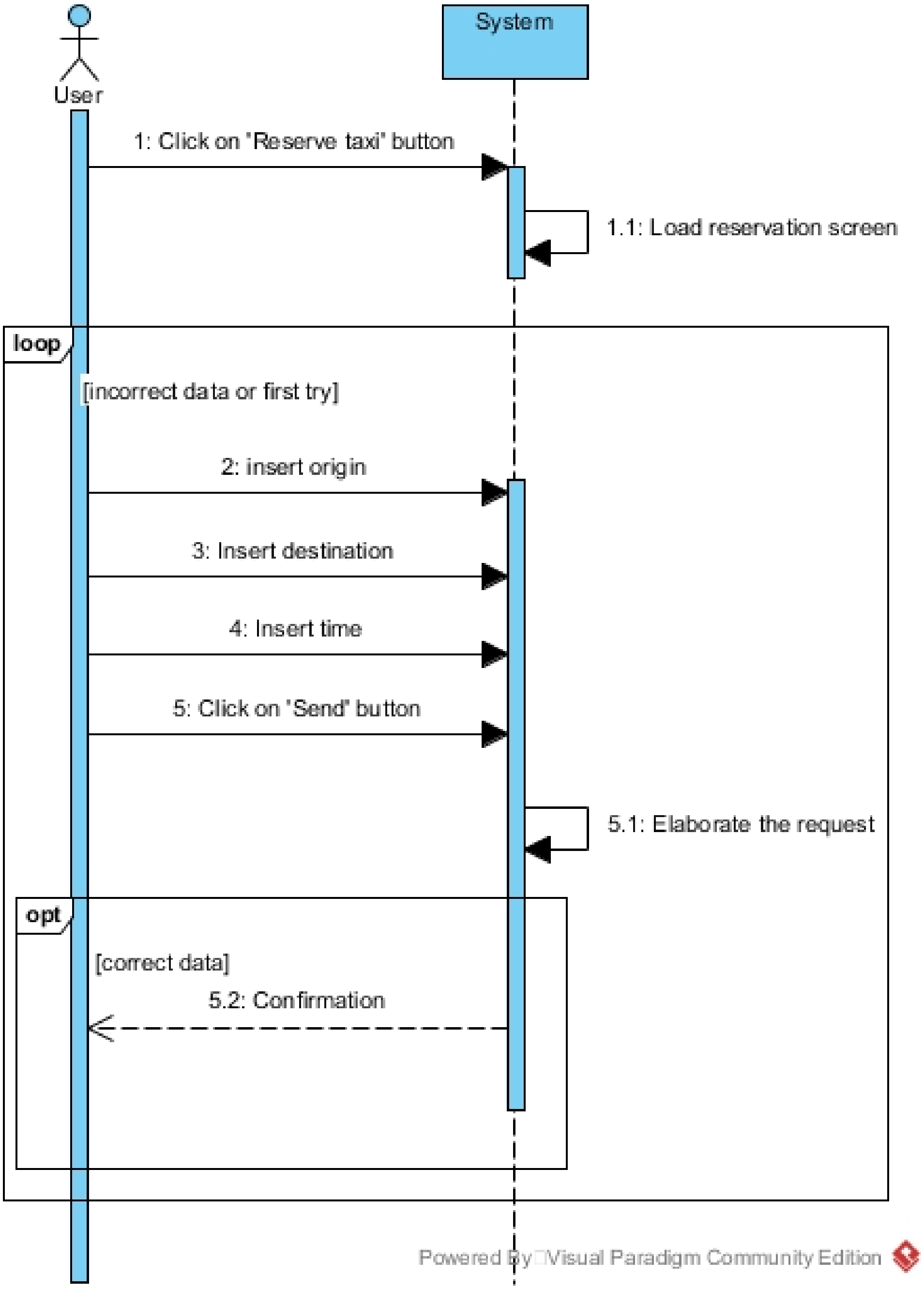
4.5.3 Login


Actor
System
1: Start application
2: Click 'login' button
loop
[not successful]
3: Insert usemame
4: Insert password
5: Confirm
opt
[successful]
5.1: success
5.2: Load relative homeage
Powered By Visual Paradigm Community Edition

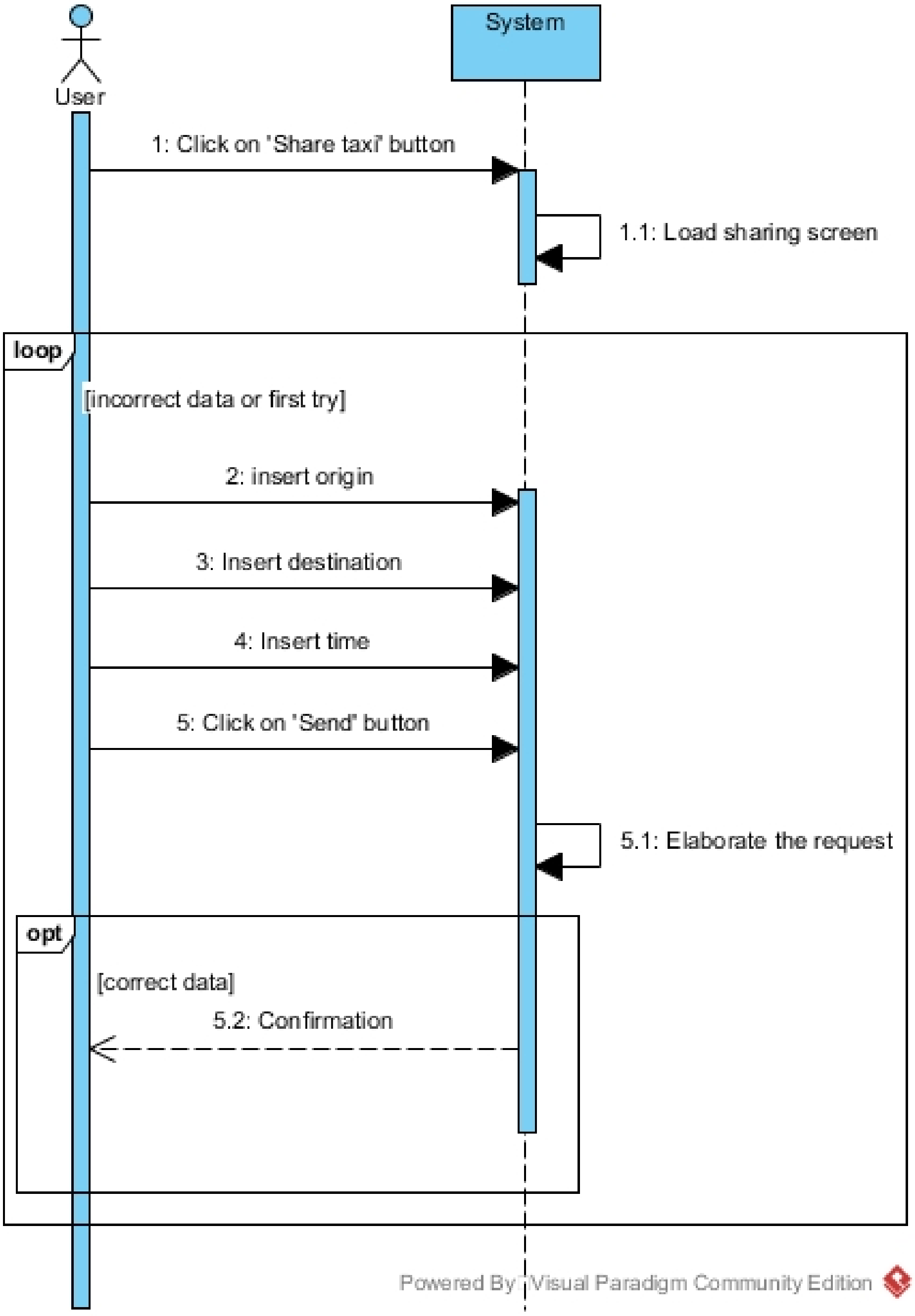
4.5.4 Request a taxi


System
User
1: Click 'Request taxi' button
1.1: Load request screen
2: Insert origin address
3: Insert destination address
4: Click 'Send' button
4.1: Elaborate the request
4.2: Confirmation
Powered By Visual Paradigm Community Edition

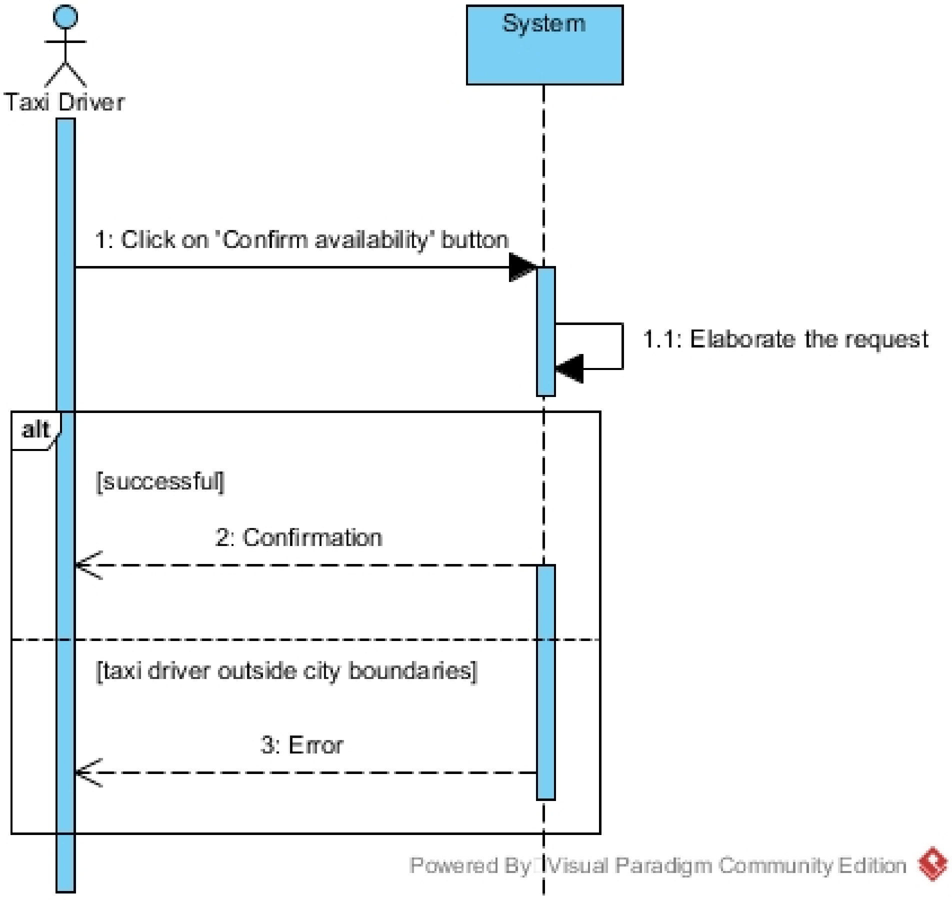
4.5.5 Reserve a taxi


System
User
1: Click on 'Reserve taxi' button
1.1: Load reservation screen
loop
[incorrect data or first try]
2: insert origin
3: Insert destination
4: Insert time
5: Click on 'Send' button
5.1: Elaborate the request
opt
[correct data]
5.2: Confirmation
Powered By Visual Paradigm Community Edition

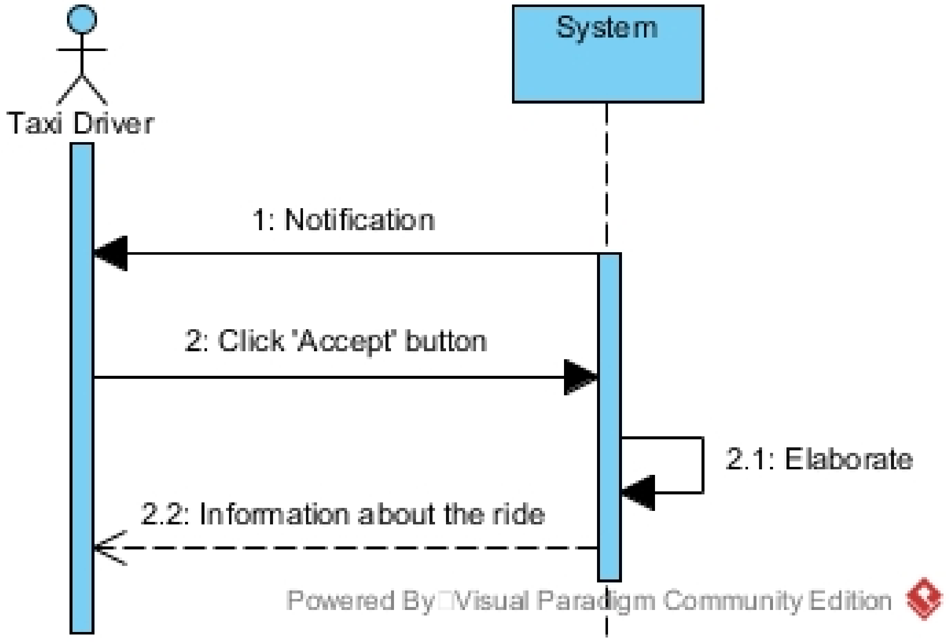
4.5.6 Share a taxi


System
User
1: Click on 'Share taxi' button
1.1: Load sharing screen
loop
[incorrect data or first try]
2: insert origin
3: Insert destination
4: Insert time
5: Click on 'Send' button
5.1: Elaborate the request
opt
[correct data]
5.2: Confirmation
Powered By Visual Paradigm Community Edition

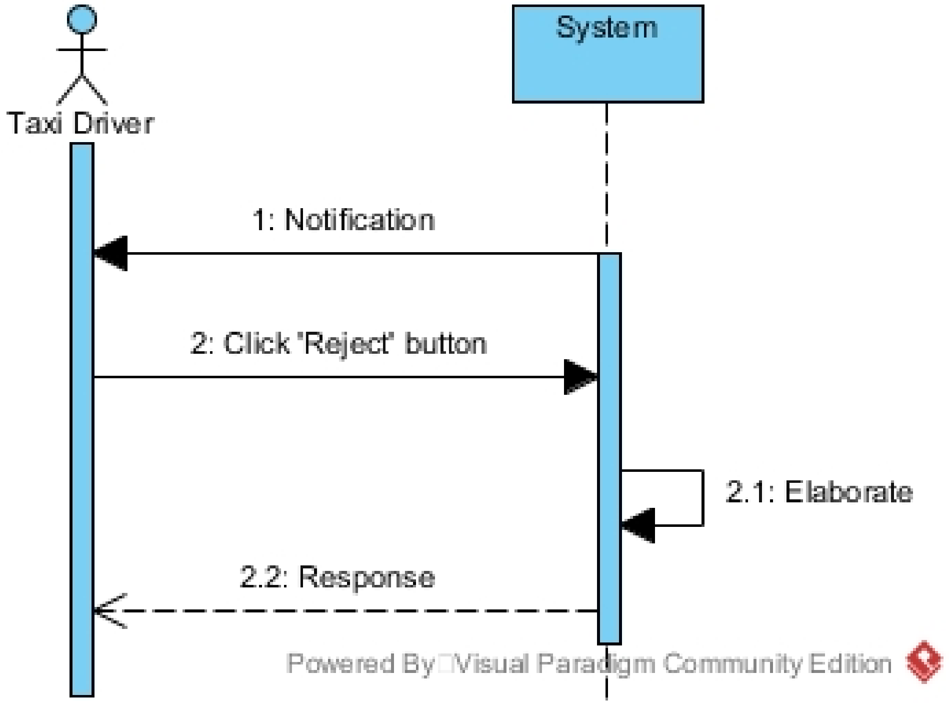
4.5.7 Confirm availability


Taxi Driver
System
1: Click on 'Confirm availability' button
1.1: Elaborate the request
alt
[successful]
2: Confirmation
[taxi driver outside city boundaries]
3: Error
Powered By Visual Paradigm Community Edition

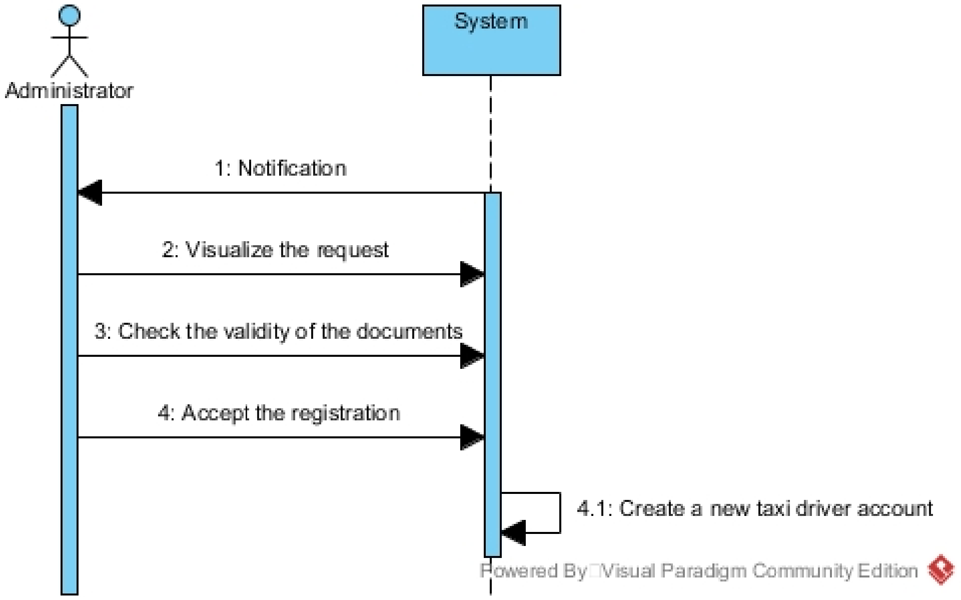
4.5.8 Accept call


Taxi Driver
System
1: Notification
2: Click 'Accept' button
2.1: Elaborate
2.2: Information about the ride
Powered By Visual Paradigm Community Edition

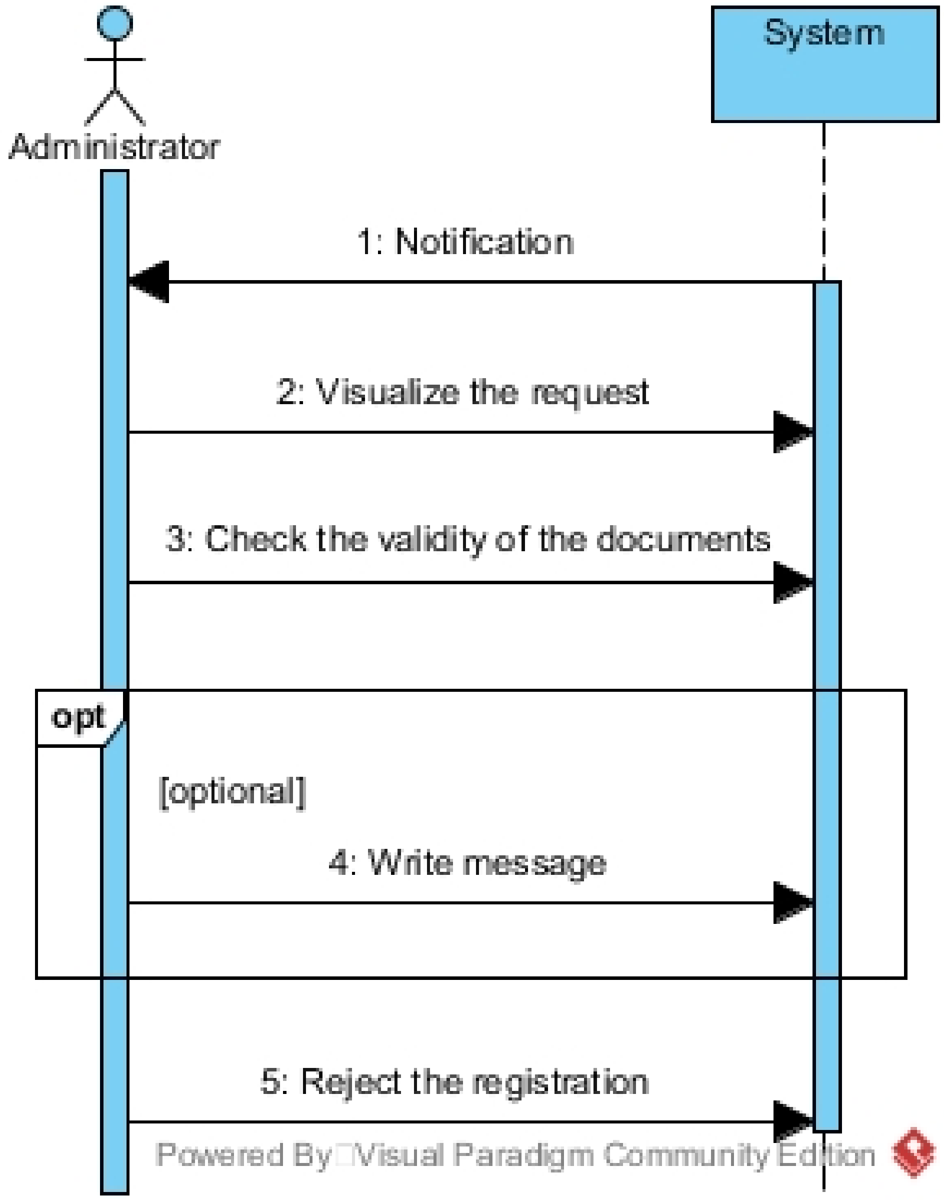
4.5.9 Reject call



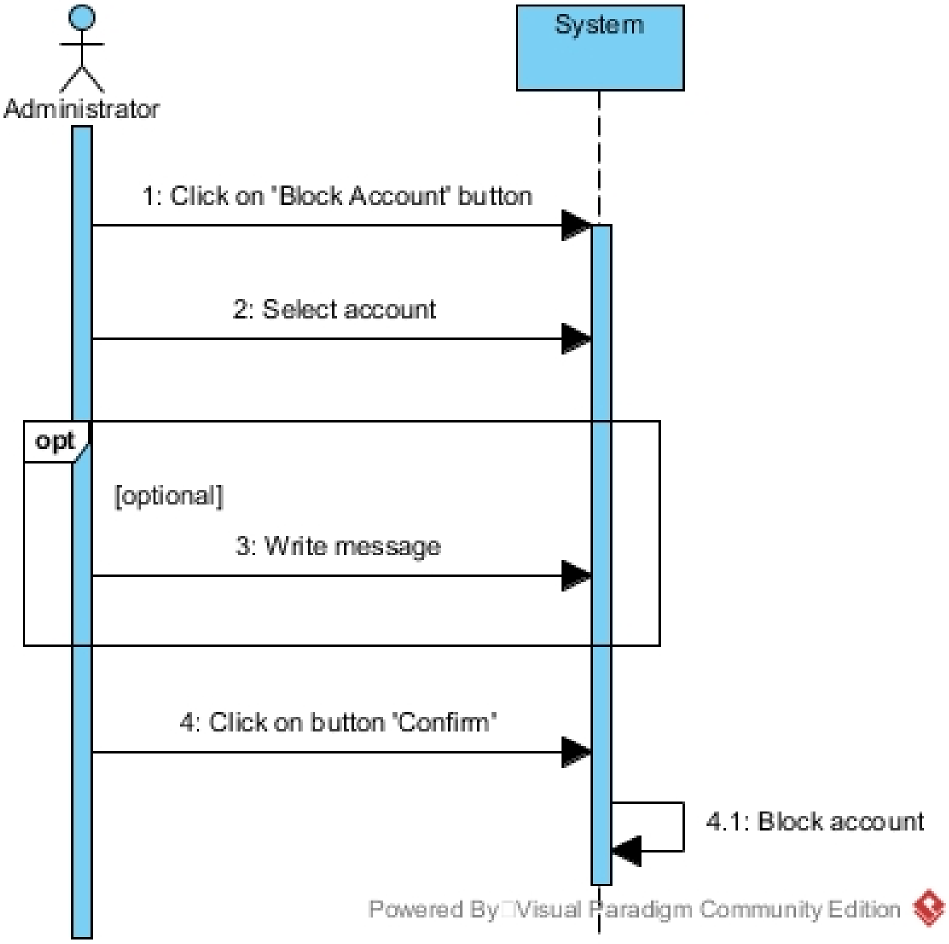
4.5.10 Accept taxi driver's registration


Administrator
System
1: Notification
2: Visualize the request
3: Check the validity of the documents
4: Accept the registration
4.1: Create a new taxi driver account
Powered By Visual Paradigm Community Edition

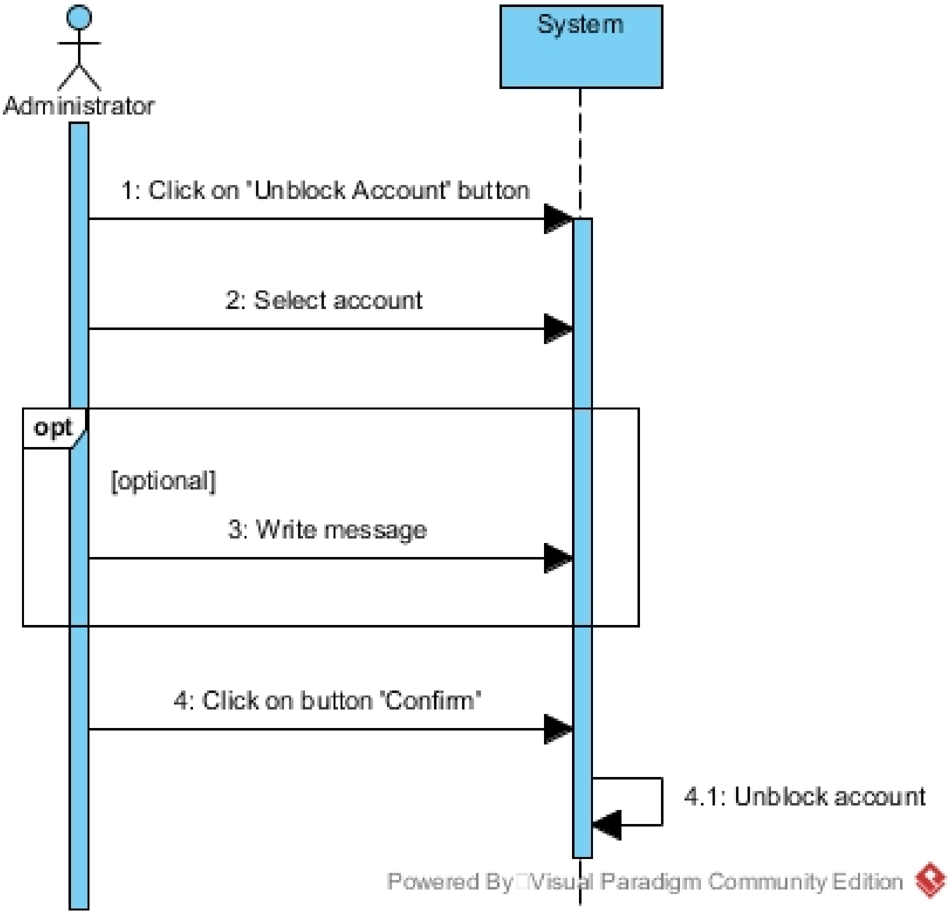
4.5.11 Reject taxi driver's registration


Administrator
System
1: Notification
2: Visualize the request
3: Check the validity of the documents
opt
[optional]
4: Write message
5: Reject the registration
Powered By Visual Paradigm Community Edition

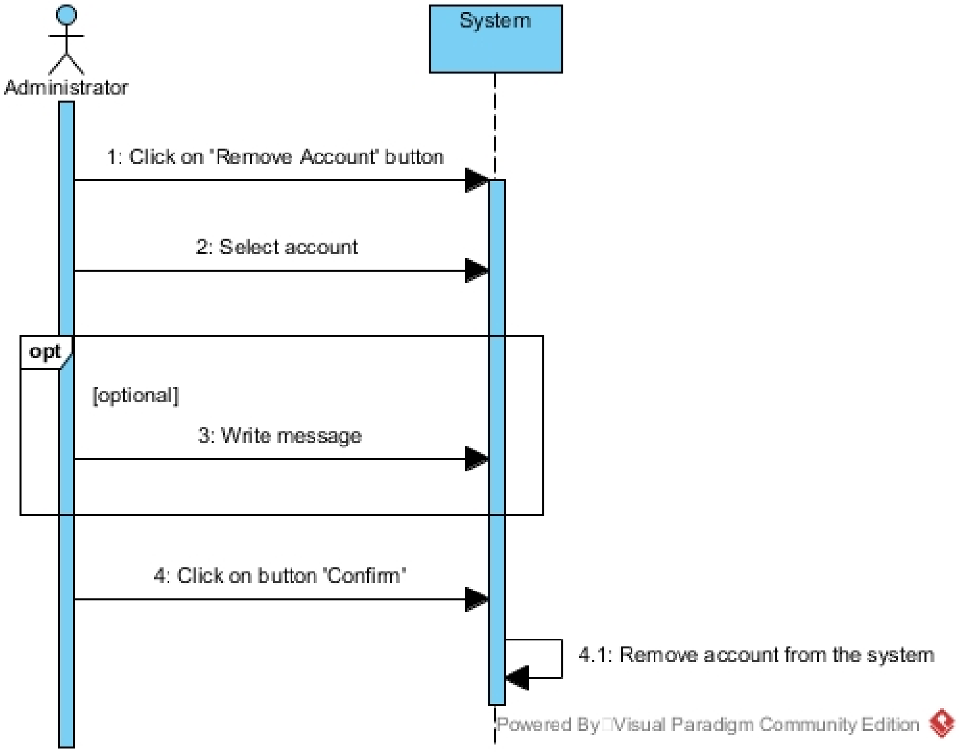
4.5.12 Block user or taxi driver


System
Administrator
1: Click on 'Block Account' button
2: Select account
opt
[optional]
3: Write message
4: Click on button 'Confirm'
4.1: Block account
Powered By Visual Paradigm Community Edition

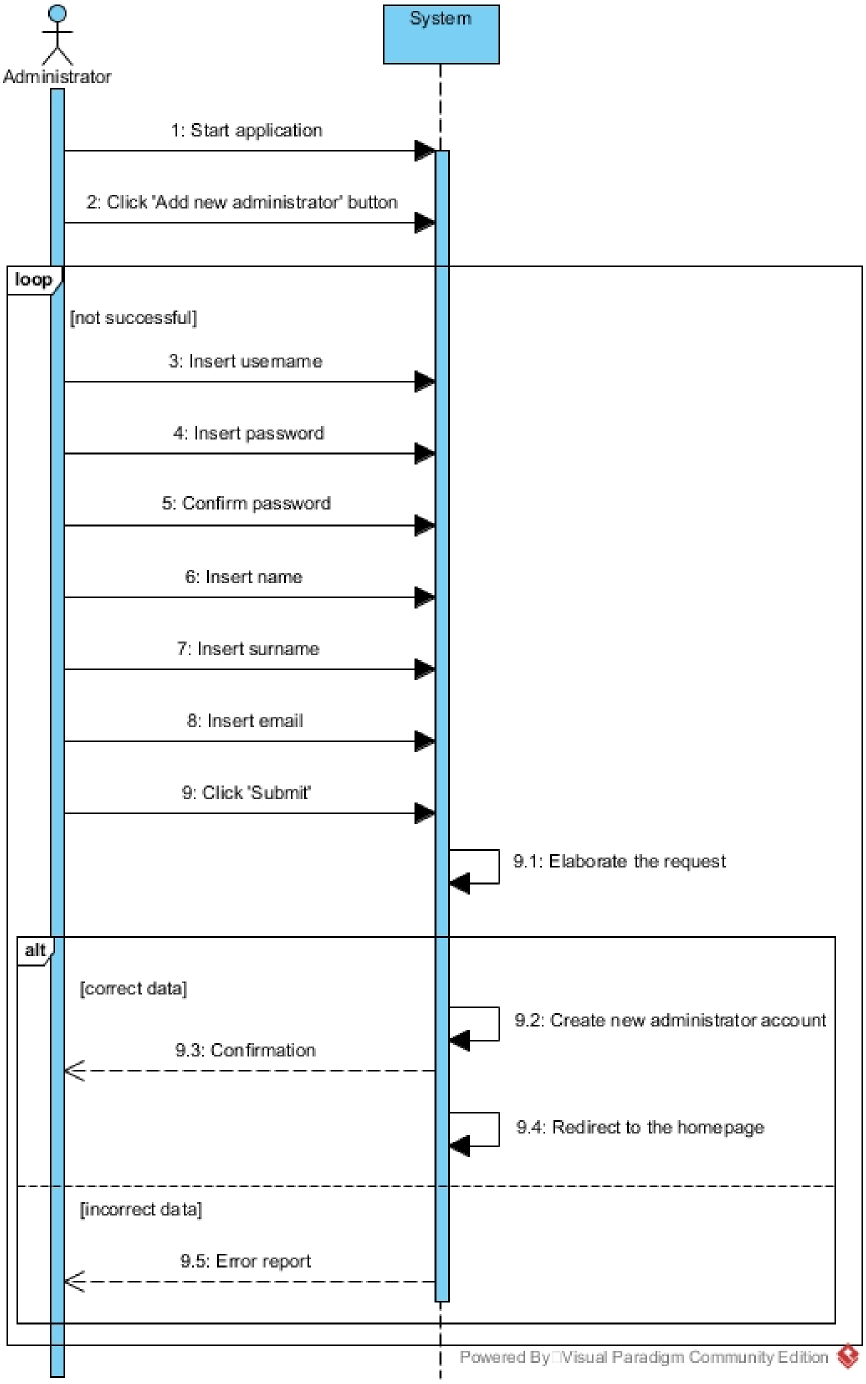
4.5.13 Unblock user or taxi driver


System
Administrator
1: Click on "Unblock Account" button
2: Select account
opt
[optional]
3: Write message
4: Click on button 'Confirm'
4.1: Unblock account
Powered By Visual Paradigm Community Edition

4.5.14 Remove user or taxi driver

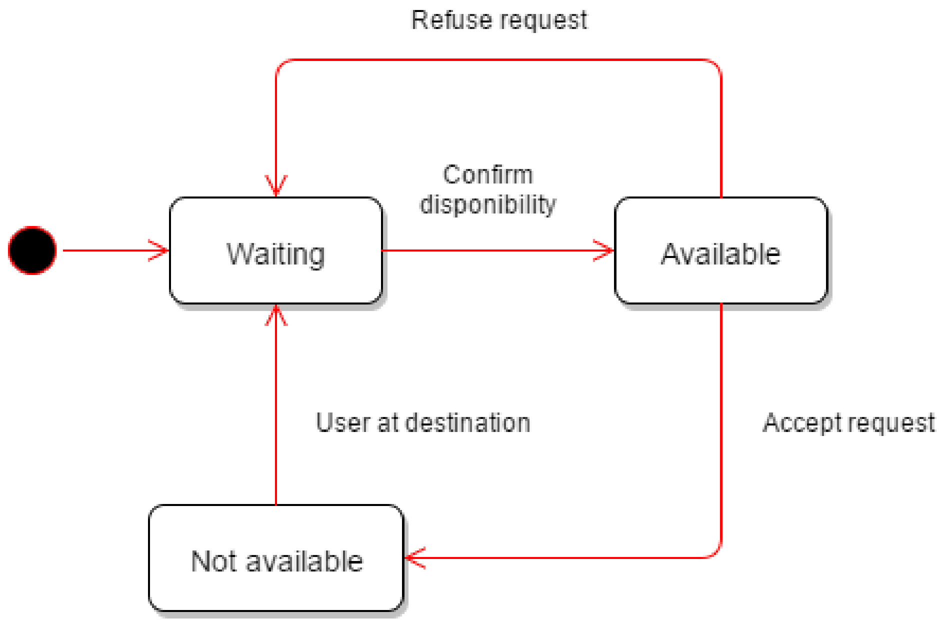

Administrator
System
1: Click on 'Remove Account' button
2: Select account
opt
[optional]
3: Write message
4: Click on button 'Confirm'
4.1: Remove account from the system
Powered By Visual Paradigm Community Edition

4.5.15 Add new administrator



4.6 State Chart Diagram

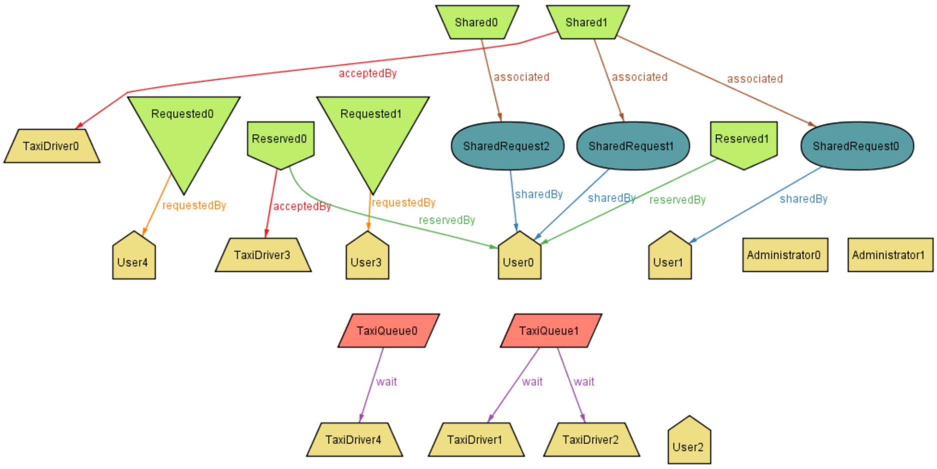
4.6.1 Taxi driver



4.7 Alloy

```
// Signatures

abstract sig Account {}

sig User extends Account {}
sig TaxiDriver extends Account {}
sig Administrator extends Account {}

abstract sig Ride {
        acceptedBy: lone TaxiDriver
}

sig Requested extends Ride {
        requestedBy: one User
}

sig Reserved extends Ride {
        reservedBy: one User
}
```

```
sig Shared extends Ride {
        associated: set SharedRequest
} {
        #associated ≥ 1
        #associated ≤ 4
}

sig SharedRequest {
        sharedBy: one User
}

sig TaxiQueue {
        wait: some TaxiDriver
}

// Facts

fact SharedRequestIsAssociatedToOneShared {
        all sr: SharedRequest — one associated.sr
}

fact UserCanRequestOneTimeTheSameShared {
        all u: User, s: Shared — lone (s.associated & sharedBy.u)
}

fact TaxiDriverCanWaitInOneOrZeroTaxiQueue {
        all t: TaxiDriver — lone wait.t
}

fact TaxiDriverCanAcceptOneOrZeroRides {
        all t: TaxiDriver — lone acceptedBy.t
}

fact TaxiDriverIsInTheQueueOrInAcceptedBy {
        all tq: TaxiQueue, r: Ride — no (tq.wait & r.acceptedBy)
}

pred show {
        #TaxiDriver = 5
        #User = 5
        #Administrator = 2

        #Requested = 2
        #requestedBy.User ≥ 2

        #Reserved = 2
        #reservedBy.User ≥ 2

        #Shared = 2
        #SharedRequest ≤ 10
        #sharedBy.User ≥ 2

        #acceptedBy.TaxiDriver ≥ 2
        #wait.TaxiDriver ≥ 2

        //#TaxiQueue = 5
}

run show for 42
```



4.8 Tools

- Alloy Analyzer 4.2
- StarUML for Class Diagram
- Visual Paradigm CE for Sequence Diagrams and Use Case Diagram
- Overleaf for LaTeX document
- Draw.io for State-Chart Diagram
- Balsamiq Mockups 3 for interface mockups

4.9 Times spent working on this document

Group Member	**Total Hours**
Andrea Donati	31
Gabriele Carassale	29
Manuel Deleo	30