

**POLITECNICO DI MILANO**  
**Computer Science and Engineering**  
**Project of Software Engineering 2**



**myTaxiService**

# **Requirements Analysis and Specification Document**

Authors: Simone Rosmini (853949)  
Vincenzo Viscusi (858689)  
Matteo Zambelli (776162)

Reference Professor: Mirandola Raffaella

# INDEX

<b><u>1. INTRODUCTION</u></b>	<b>4</b>
1.1 DESCRIPTION OF THE GIVEN PROBLEM	
1.2 APPLICATION SCOPE	
1.3 GOALS	
1.4 DOMAIN PROPERTIES	
1.5 GLOSSARY	
1.6 PROPOSED SYSTEM	
1.7 IDENTIFY STAKEHOLDERS	
<b><u>2. ACTORS IDENTIFYING</u></b>	<b>7</b>
<b><u>3. REQUIREMENTS</u></b>	<b>8</b>
3.1 FUNCTIONAL REQUIREMENTS	
3.2 NON-FUNCTIONAL REQUIREMENTS	
3.2.1 USER INTERFACE	
<b><u>4. SCENARIO IDENTIFYING</u></b>	<b>16</b>
<b><u>5. UML MODELS</u></b>	<b>17</b>
5.1 USE CASE LIST	
5.2 USE CASE DIAGRAM	
5.3 USE CASE DESCRIPTION AND RELATIVE SEQUENCE DIAGRAM	

#### 5.4 CLASS DIAGRAM

#### 5.5 STATE CHART DIAGRAMS

### **6. ALLOY MODELING.....38**

#### 6.1 ALLOY CODE

#### 6.2 ALLOY WORLDS

### **7. DOCUMENT PERSPECTIVE.....46**

#### 7.1 REFERENCE DOCUMENTS

#### 7.2 FUTURE IMPLEMENTATION

#### 7.3 HARDWARE CONSTRAINTS

### **8. USED TOOLS.....46**

## **1. INTRODUCTION**

This document is a Requirement Analysis and Specification Document (RASD) and his main goal is to provide a specification of requirements that must be satisfied by the system.

This document presents a description of the functional requirements and non-functional, using various UML diagrams and an high level specification. There is a part where it is instead presented a formal model of specifications and the results of its analysis, made by Alloy.

This document is addressed to myTaxiService project developers and all other stakeholders of the project. Regarding the firsts, the document it becomes important when you need to establish a coherence between the development and the requests processed. For stakeholders, this document is a tride of the analytical work done by developers, useful for interpreting their development work.

### **1.1 DESCRIPTION OF THE GIVEN PROBLEM**

The problem is the one to try to develop a taxi service which is more efficient than the actual system by using a mobile application or a web application.

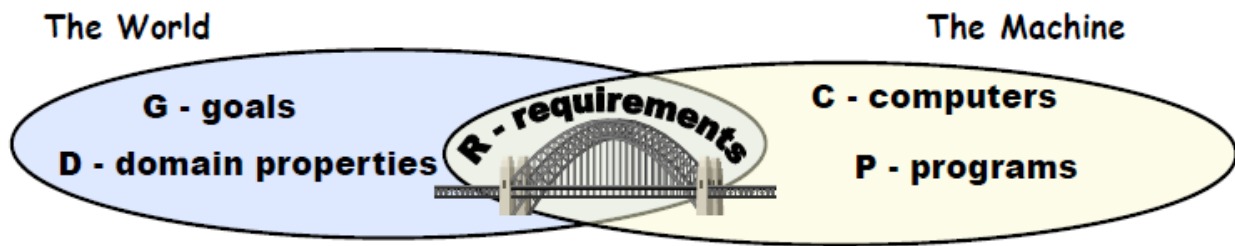
Using the new taxi services the user should be able to call a taxi to start a ride from his actual position or make a reservation of a run.

In order to save money, users can choose to make a reservation for a share ride without extend the duration of the ride more than they really want by inserting the maximum delay time they can accept. In this way the system joins only the compatible runs allowing the users to arrive where they want saving money and never being late.

### **1.2 APPLICATION SCOPE**

The main scope of the product is allow people to call taxies more easily and guarantee a fair management of taxi queues. User, once registered, should be able to request a taxi make a reservation, in a single mode or in a shared mode, or modify an already done ones.

On the taxi driver's view he can communicate at the system when is ready for a taxi ride, when is busy or when he is not ready for other problems, if he is ready and receive a taxi call from a user he can decide at the moment if accept or not the call with the application.



### 1.3 GOALS

- G1: Allow the users to request a real time ride
- G2: Allow the user to reserve a ride
- G3: Reserve a ride shared among more users
- G4: Allow the user to sign up to the taxi service
- G5: Allow the login of a subscribed user
- G6: Keep constant, as much as possible, the number of taxi for each zone
- G7: Allocate a taxi
- G8: Provide to the taxi administrator some statistics to choose the number of taxi for each zone.
- G9: Allow the user to create a calendar with the scheduled rides

### 1.4 DOMAIN PROPERTIES

- A Shared call can be joined in the same ride only with other shared call
- If a run is serving a single call then it isn't serving other calls.
- If a run is serving a real time call then it isn't serving other calls.
- Each call is associated to a single ride.
- A user can be a passenger of a taxi only if the taxi is serving a call made by the user.
- All the shared calls associated to the same ride have the same starting zone.
- A shared ride must stop to each start address and end address specified by the call that it is serving.
- Each taxi is associated to a single queue according with its position.
- Each ride is associated to a single taxi.
- Each taxi can serve only one ride at the same time.
- Two calls are shared in the same ride only if the new estimated arrival time for each call doesn't exceed the maximum delay accepted from the corresponding user.
- Two calls are shared in the same ride only if they start from the same zone.
- A taxi receives a request for a ride only if its state is free

## 1.5 ASSUMPTIONS

- if a user of a taxi shared call cancel its ride, all the other users remain in the taxi shared call and the path change into the shortest one without the ride deleted
- we assume that the taxis don't have any type of problem during the ride
- we assume that the taxi driver manage by itself the taxi fuel when he is in the don't available state
- we assume that the users cannot modify the destination address during a ride
- When a taxi is mobilized, it will reach the source location in the shortest possible time
- Accurate taxis locations are known by GPS
- We assume that the users pay with a credit card or paypal when a taxi ride is confirmed and so we don't need to introduce any type of forfeit

## 1.6 GLOSSARY

Here there are explained the meaning of the terms used in this RASD :

**USER:** for user we mean a person already registered in the system. A user has a profile that includes all these information: Name, Surname, Email, Username, Password, Payment Card And optionally a Picture. A user has the capability of doing everything that we mentioned in the goals, making exception for accept or decline the calls.

**TAXI DRIVER :** a taxi driver is a person that work for taxi's company and drive the taxi during his worktime. A taxi driver has the capability of accept or decline the calls.

**GUEST :** a guest is a person who hasn't signed up yet. Guests have less power in the system than users, they don't have the users' skills, and the only function they can use is to sign up.

**CALL:** Each request done by a user for a ride (real time, shared or single)

**TAXI Ride:** a taxi ride is the path between the source and the destination (final or intermediate)

**TAXI SHIFT :** we call taxi shift the movement of a taxi that change its zone for compensate the movement of another taxi in the reverse path

**(PROCESS OF ) TAXI ALLOCATION:** when a user call a taxi with the app the taxi driver first in the queue of his zone receive a notification and have to accept or decline

the call, if accept the call finish when taxi driver bring the user at the destination, but if he decline, the call pass through the queue until one taxi driver accept it.

**TAXI DRIVER'S STATES** : a taxi driver with his application can set its status on three modes :

1. Available : when he is waiting a call and can accept it
2. Busy : when he is driving for another call and can't serve another one
3. Not available : when for personal reasons he can't serve a taxi call

## **1.7 PROPOSED SYSTEM**

We suppose that the actual system that provides the service is the traditional call to the Taxi centerline. Now we are developing a new service that will substitute the actual system and it will be based on web app service.

The system that we will implement is composed by a central server that works 24hperday and communicates with clients from a web application (on the website or on a smartphone), the information about the traffic are taken from Google Maps that also provide the GPS information about the taxi and the virtual space where we store and compute the information about the taxi's ride.

## **1.8 IDENTIFY STAKEHOLDERS**

Our main stakeholder is the professor who gave us the delivery of the project, the person who will use the system when it will be delivered on the market (user, guest and taxi driver) and Google because give us the main information about the street and provide a system for analyze and compute the GPS information about the taxi.

## **2. ACTORS IDENTYFING**

The Actors of our system are three:

**GUEST**: is someone who has not signed up yet in the system and can only see main features and signing up.

**USER** : is someone who have already signed up in the system and can see and do the main features of the system like **CALL** a taxi ride for the same moment or **RESERVE** a taxi ride for another day (shared or not)

TAXI DRIVER : is the person who can drive a taxi and communicate with the system with a screen placed in the taxi and through it he/she can accept or decline the request that arrive at he/she

### **3. REQUIREMENTS**

#### **3.1 FUNCTIONAL REQUIREMENTS**

-G1: Allow the users to request a real time run (shared btw world and system)

- R1: The system should require the user to fill out the following fields:
  - Meeting address
  - Destination address
- R3: The system should allocate a taxi for the request
- R6: The system should create a path for the ride and send it to the taxi
- R5: The system should send to the user the code of the taxi that has taken on the ride and the waiting time

-G2: Allow the user to reserve a run (shared btw world and system)

- R1: The system, to create a new reservation, should require the user to fill out the following fields:
  - Meeting Time
  - Meeting address
  - Destination Address
  - Travel mode
  - Repetition (No/Daily/Weekly/Monthly)
- R2: The system should compute the duration of a ride and the expected arrival time
- R3: The system accepts the reservation only if there is an interval time of at least 15 minutes between the expected arrival time of other scheduled reservations and the meeting time of the new reservation
- R4: The system accept the reservation only if there is an interval time of at least 15 minutes between the expected arrival time of the new reservation and the meeting time of other scheduled reservations
- R5: The system should associate a tracking number (ID\_Reservation) to the reservation
- R6: The details of the new reservation are stored only if the reservation is accepted
- R7: The system accepts the reservation only if the user has enough money to pay the ride.
- R8: The system 10 minutes before de meeting time mast allocate a taxi for the reservation.



- G3: Reserve a ride shared among more users:
  - R1: The system should ask to each user that wants to share a ride, the maximum delay on the estimate arrival time that he wants to accept for sharing the ride.
  - R2: The system must be able to join different paths to create a single paths that passes in all the start and destination addresses of each path.
  
- G4: Allow the user to sign up to the taxi service (shared btw world and system)
  - R1: The system should require the user to fill out the following fields:
    - E-mail with confirmation
    - Password with confirmation
    - Surname and Name
    - Preferred payment way(credit card or paypal)
  - R2: The system should require the user to insert a list of the zone from he often takes a taxi
  - R3: The system uses the email as the user identifier
  - R4: The subscription is accepted only if all the fields have been fulfilled and the email doesn't already exists in the database of the registered users
  - R6: The system stores all the user information only if the subscription is accepted
  - R5: The system should send a welcome email to the user with all the necessary information to use the service.
  
- G5: Allow the login of a subscribed user
  - R1: The system must require the user to fill out the following fields:
    - E-mail used in the subscription phase
    - Password used in the subscription phase
  - R2: The login is accepted only if the password inserted corresponds with the email in the database of the registered users.
  - R3: The user can have access to the service only if the login is accepted
  
- G9: Keep constant, as much as possible, the number of taxi for each zone (world):
  - R1: Every time a taxi starts a ride from a zone A to a zone B then a taxi in the zone B moves towards the zone A.
  
- G3: Allocate a taxi (shared btw world and system)
  - R1: the system should send the details of a ride request to the first taxi in the queue of the zone that contains the start address of the ride:
    - Tracking number of the ride.
    - Path of the ride.
    - Path to reach the start address
    - Call type (Real Time or Reservation).
    - Travel Mode (Single or Shared).
  - R2: The taxi driver taxes on the ride only if he accept the request within 30 seconds from its arrival.

- R3: The system should pass the ride request to the next taxi in the queue only if the taxi driver has refused the request or it passes more than 30 seconds from the arrival of the request.
- R4: If the taxi driver refuses the request or he takes more than 30 seconds to accept his taxi is removed from the head of the queue and placed in the tail.
- R5: If no taxis are available in the zone the request is passed to the first taxi in the queue of a near zone.

-G6: Provide to the taxi administrator some statistics to choose the number of taxi for each zone:

- The system should calculate the average number of run request per hour for each zone (request rate).
- The system should be able to show to the taxi administrator a table with the following information:
  - Actual number of taxi for each zone
  - Priority of the zones
  - Request rate for each zone.

-G8: Allow the user to create a calendar with the scheduled rides

- R1: The system should display to the user a calendar with the scheduled reservation.
- R2: The system should ask the user to select the day in which he wants to schedule a new reservation or see the details of once already scheduled
- The system, to create a new reservation, should require the user to fill out the following fields:
  - Meeting Time
  - Meeting address
  - Destination Address
  - Travel mode
  - Repetition (No/Daily/Weekly/Monthly)
- R3: The system should compute the duration of a ride and the expected arrival time
- R4: The system accepts the reservation only if there is an interval time of at least 15 minutes between the expected arrival time of other scheduled reservations and the meeting time of the new reservation
- R5: The system accept the reservation only if there is an interval time of at least 15 minutes between the expected arrival time of the new reservation and the meeting time of other scheduled reservations
- R6: The details of the new reservation are stored only if the reservation is accepted

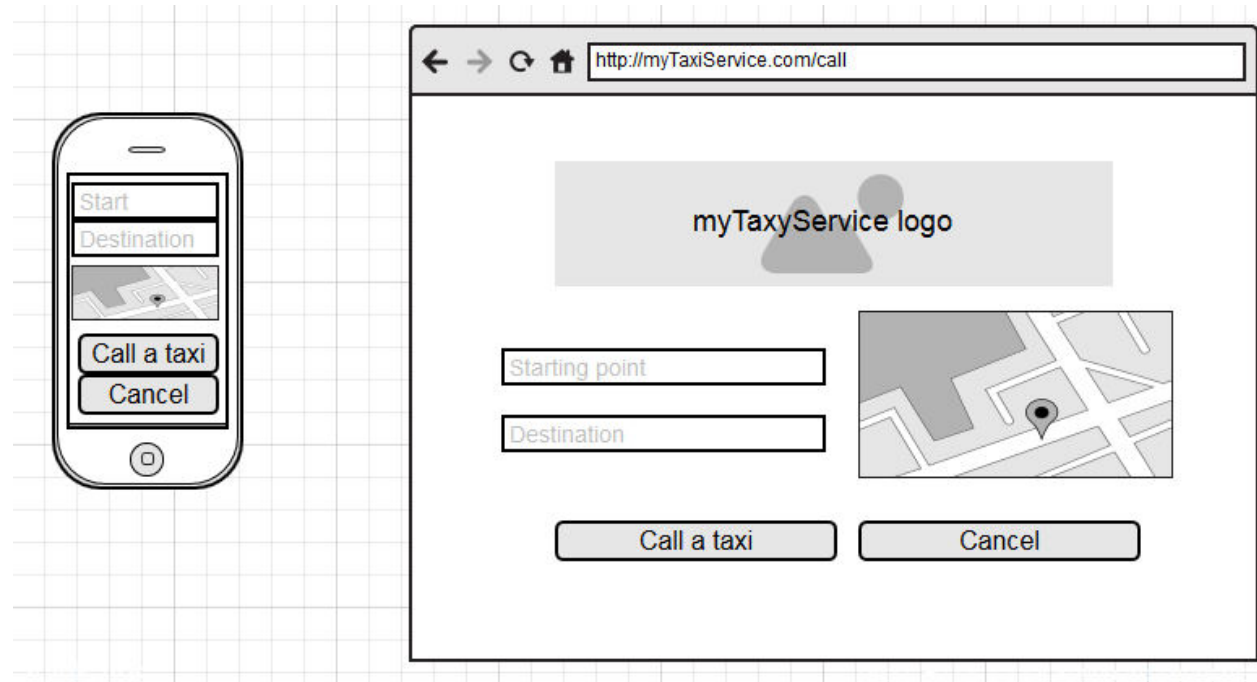
### 3.2 NON-FUNCTIONAL REQUIREMENTS

- The service must be available 24 hours per day
- The system must choose always the fastest path for the moving of the taxis
- The system must be available through web application and mobile application

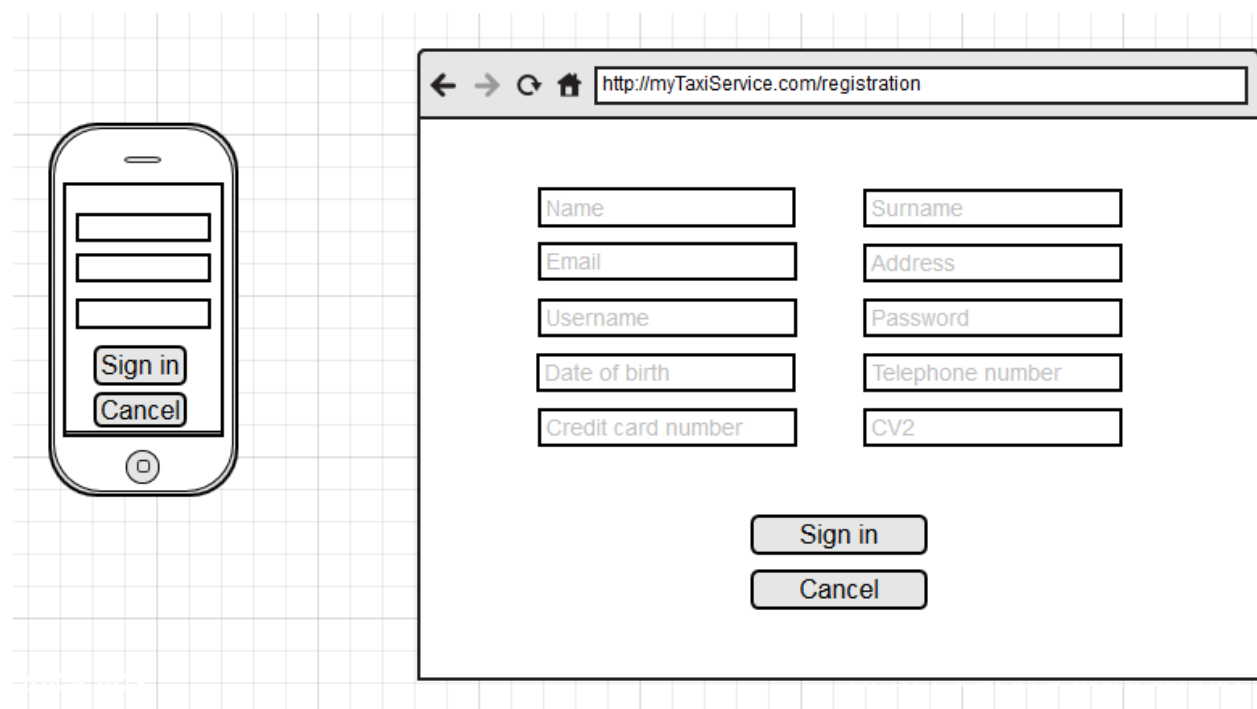
### 3.2.1 User Interface

Here are presented some mockup that represent an idea of the structure of the application pages:

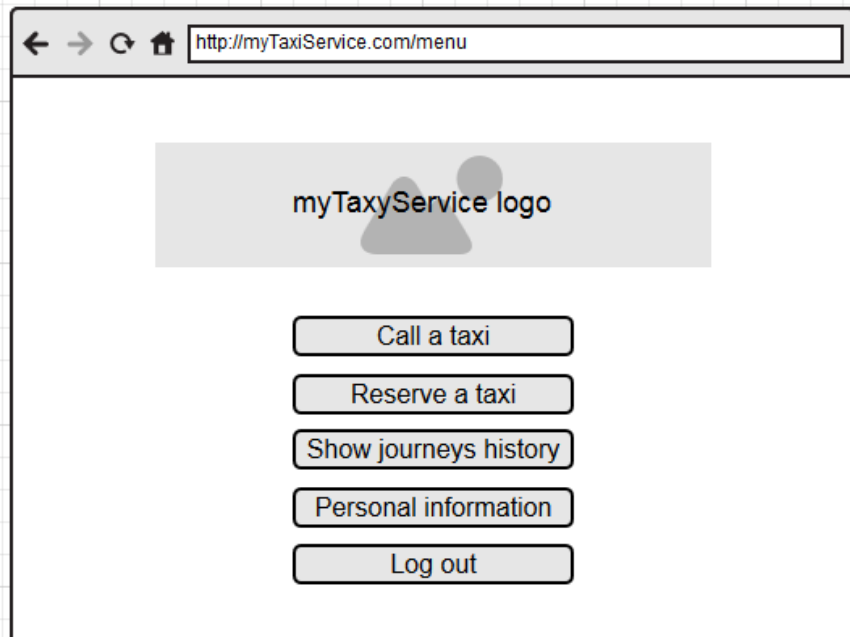
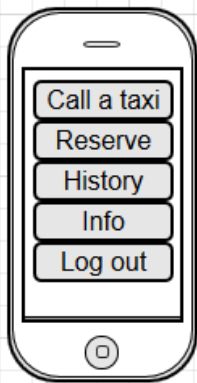
-Login



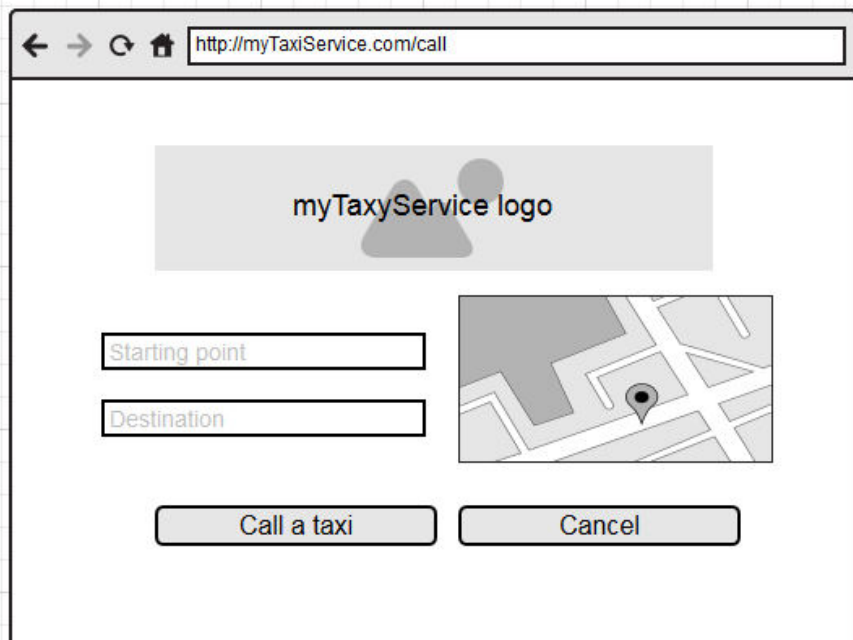
-Registration



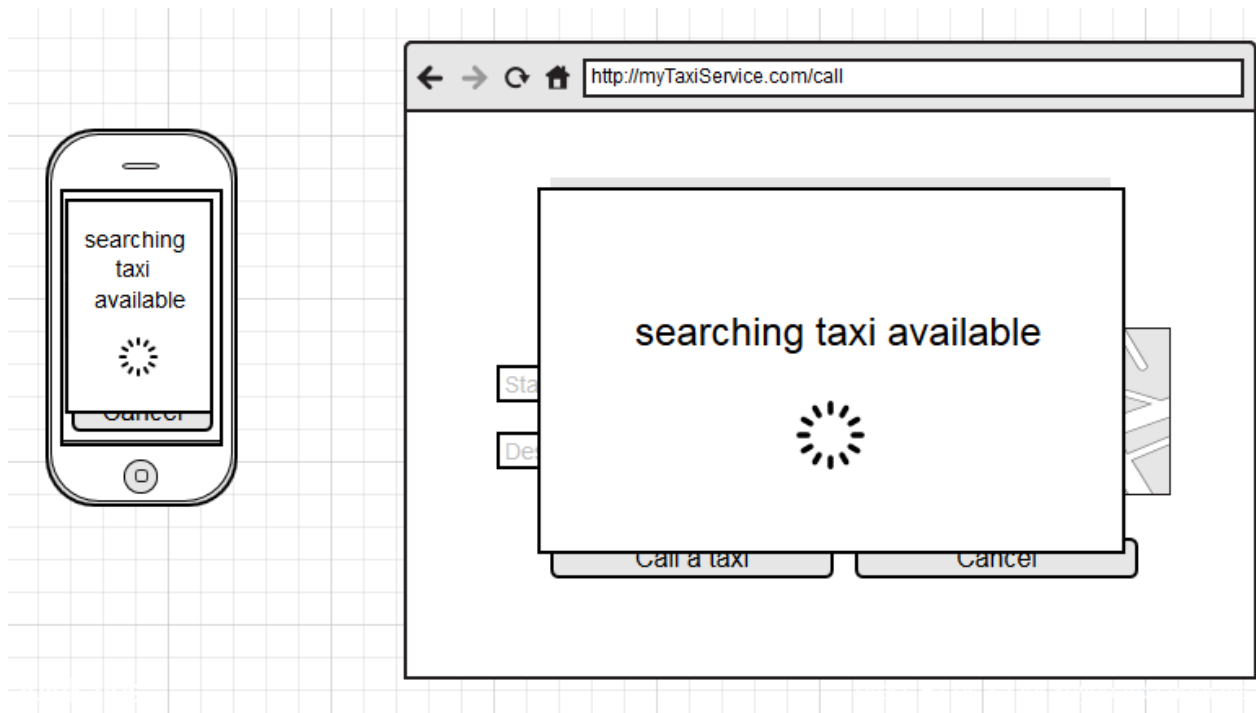
## -Menu



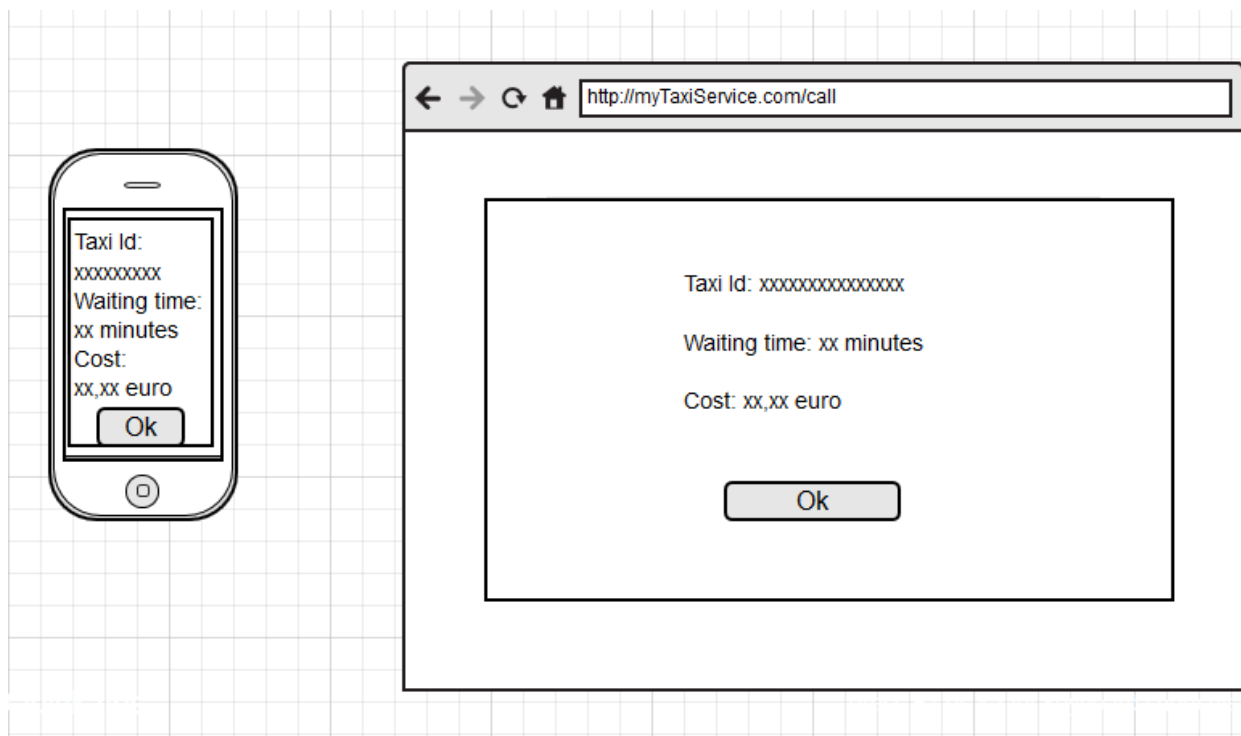
## -Calling of a taxi



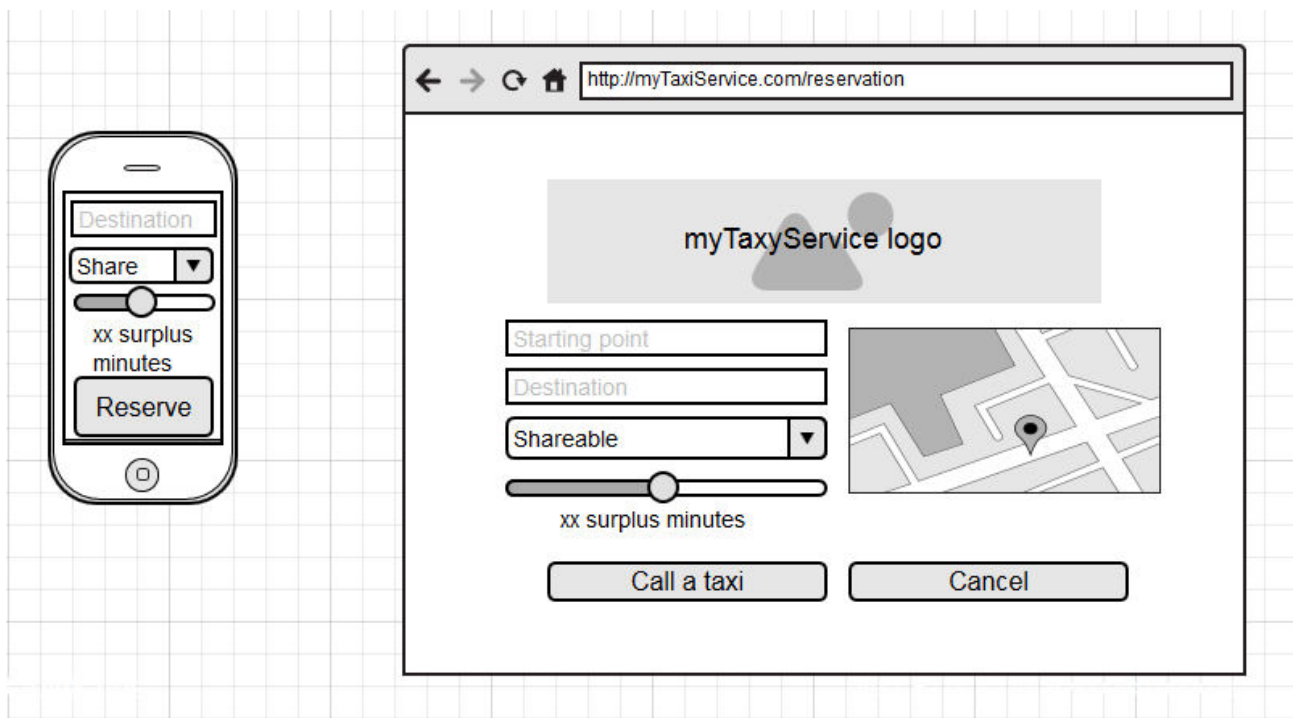
-Waiting



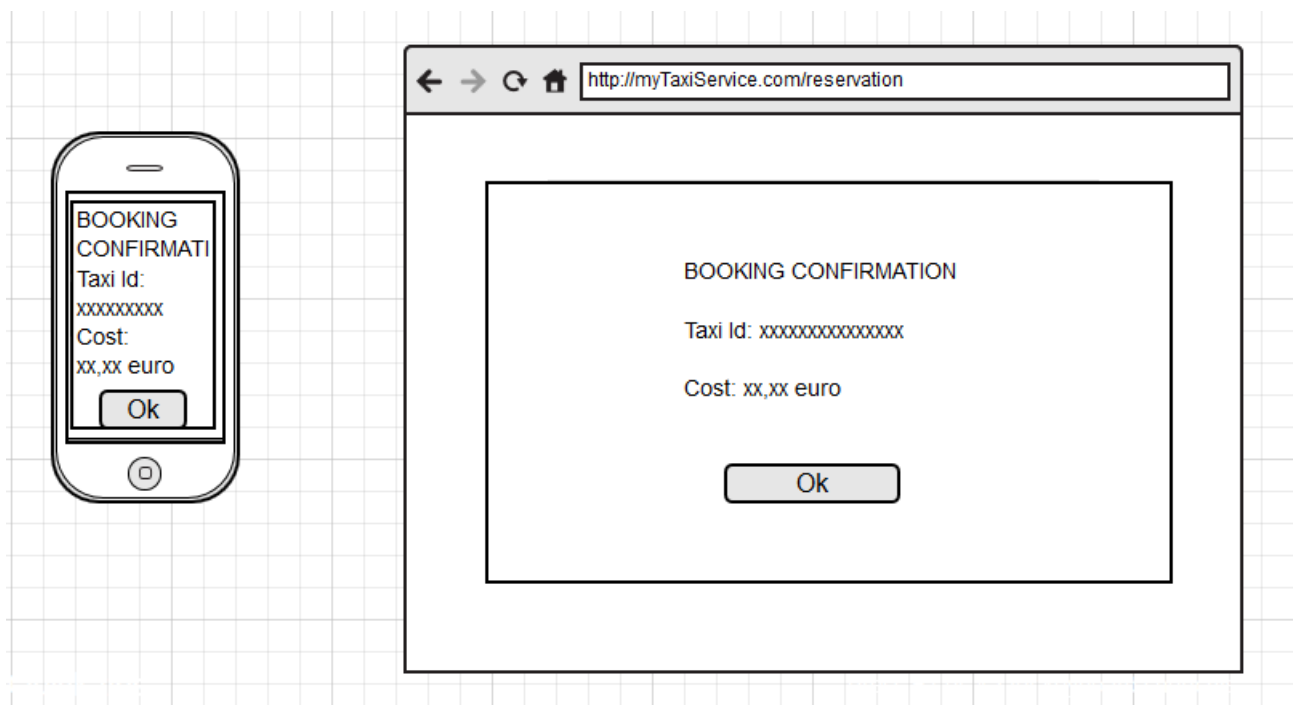
-Confirm call



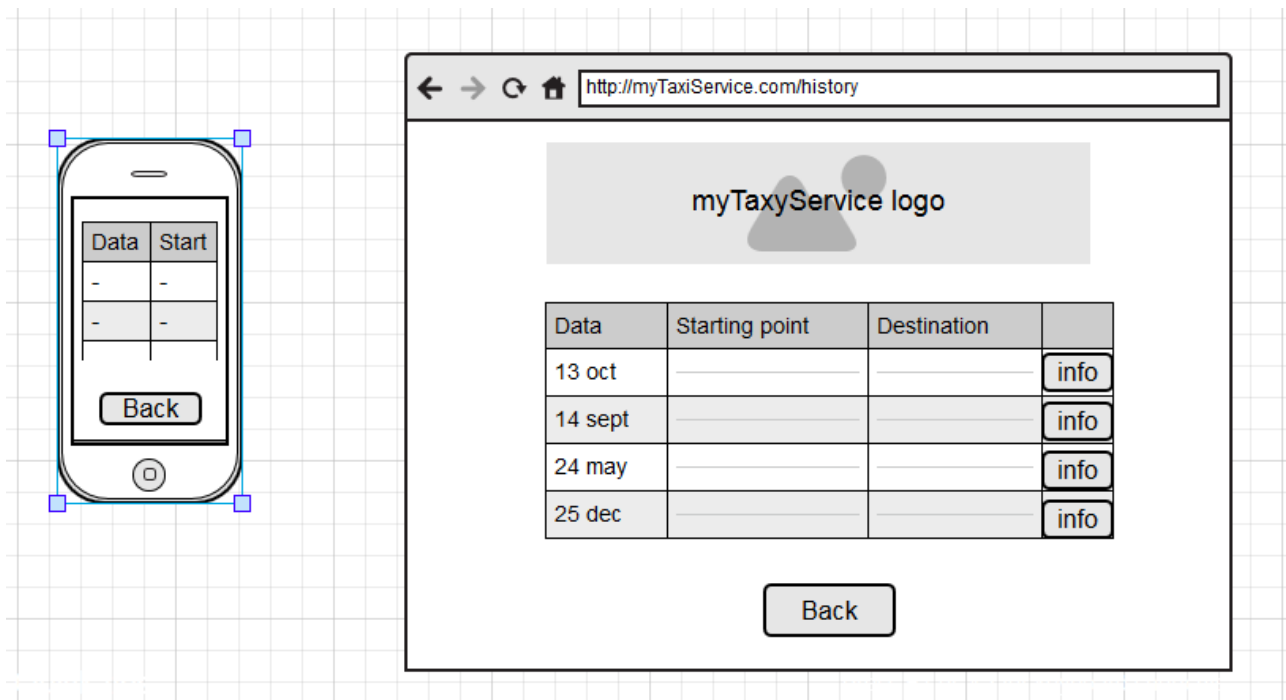
## -Reservation of a taxi



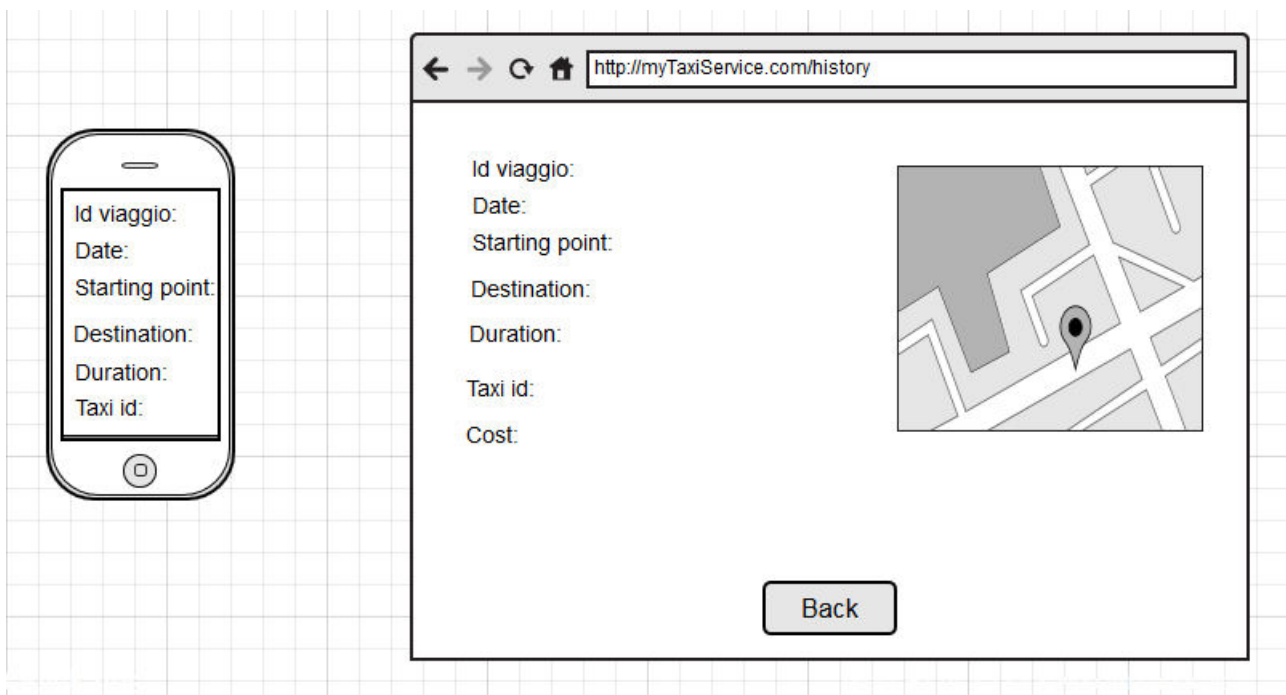
## -reservation confirm



## -Journeys history



## -Information of a journey



## **4. SCENARIO IDENTIFYING**

### **Scenario 1**

Simone as every day is doing his job as a taxi driver for myTaxiService. He is on duty for two hours but has not yet received calls.

Around noon the integrated system in the car displays a request of a customer. He accept and heads towards the position of the passenger.

As soon as Vincenzo has settled in the car, Simone starts moving toward the destination.

When Vincenzo leaves the car, Simone informs the system and becomes available

### **Scenario 2**

Matteo as every year decides to visit his grandparents in Milan during the Christmas holidays.

Knowing that their house is not accessible by public transport, this year Matteo decided to sign up on myTaxiService, a service that allows to call and book a taxi with a simple mobile app.

He, arrived to the central station, logs in with the app and immediately calls a taxi to take him to the grandparents' address.

He waits for a few seconds until the system tells him the taxi MIT278 will come to pick him up after 5 minutes.

A few minutes later therefore Matteo takes the taxi of Pasquale that will arrive directly in front of his grandparents' house.

After thanking Pasquale, Matteo descends from the taxi and heads for home.

### **Scenario 3**

Luca must participate in a working meeting to be held at a hotel across town in the afternoon. He decides to book a taxi with the app myTaxiService. He completes the reservation form with the time and place of the meeting and to spend less decides to sharing the journey with others by specifying with the appropriate bar an extension of time up to 15 minutes.

10 minutes before the meeting, Luca receives the identifier of the taxi that is coming. After a few minutes after departure, the taxi of Massimo stops to pick up Elisa. During the journey the two passengers talking to each other and Elisa tells Luca that is heading to a clinic for a medical examination.

Reached her destination, Elisa greets Luca, thanks Massimo and gets out of the taxi, Luca instead continues until his destination.

Once arrived, he thanks Massimo and heads for the door of the hotel.



## Scenario 4

Gianluigi is a taxi driver of myTaxiService and he is in the stadium area. He just brought a passenger to the destination and now is in a parking to rest since it has a big headache.

Not even 10 minutes from when he turned off the car, he gets a call request from the system. The passenger should be brought into the area Lambrate which is across town.

He thinks for a few seconds before putting safety before the money, because he does not feel in a position to do that distance at the moment.

Gianluigi then refuses the request, knowing that he will be sent to the back of the queue.

Gianluigi puts himself as " not available " to try to recover.

## Scenario 5

Adalberto wants to go see an exhibition in the Navigli area in the afternoon. Not being a lover of public transport such as metro and tram, he decides to use myTaxiService that he has already used another time remaining satisfied.

He logs in with the app with the smartphone and inserts the travel informations.

After lunch he decides to take one hour of rest before going to the meeting place with the taxi. At one point he is awakened by the sound of a notification about the confirmation of booking of the taxi that will come to the meeting point after 10 minutes.

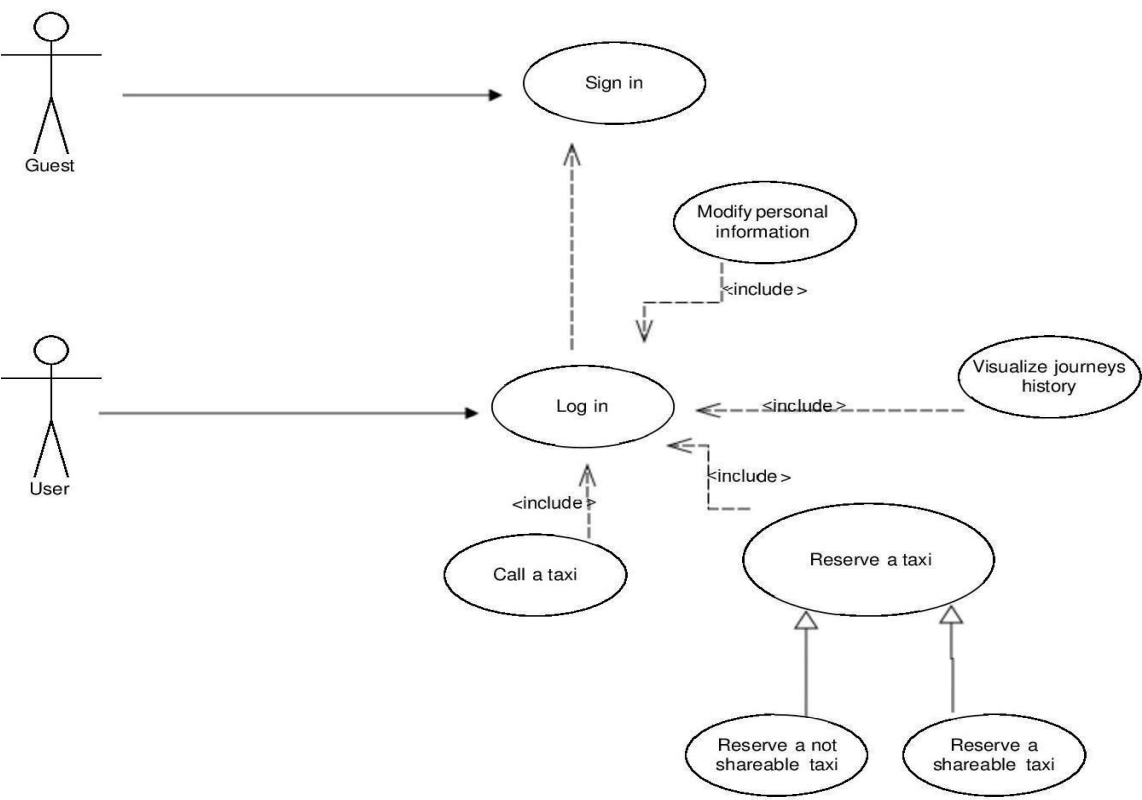
He prepares in a hurry but still reaches the starting point with 20 minutes late and not even a tride of a taxi. Cursing himself for being lying on the couch after dinner, he decides to call another taxi, aware that he will have to repay the journey.

## **5. UML MODELS**

### **5.1 USE CASE LIST**

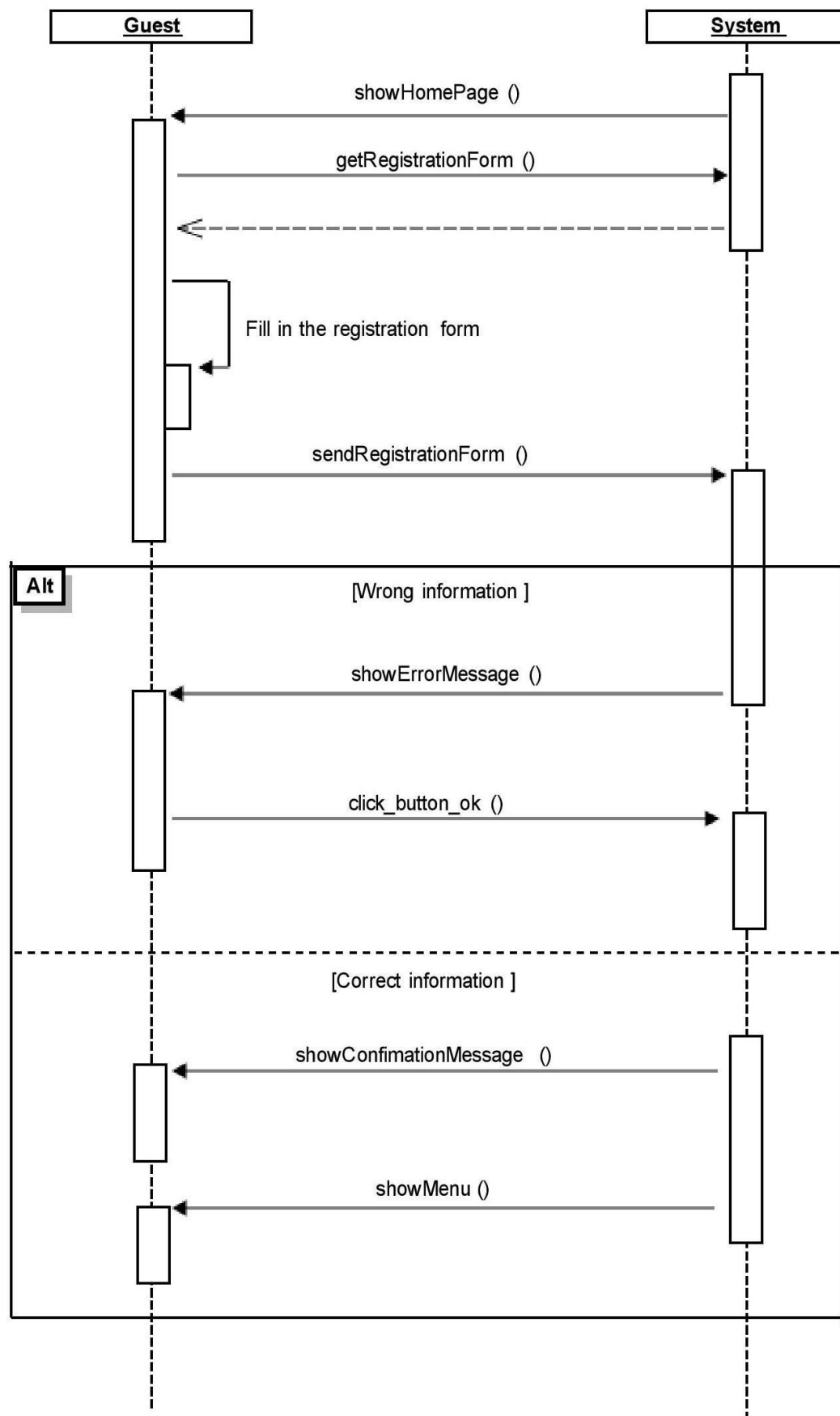
- Sign up
- Log in
- Calling of a taxi with acceptance of the taxi driver
- Calling of a taxi with the driver's refusal
- Reservation of a not shareable taxi
- Reservation of a shareable taxi
- Display information of a journey

5.2 USE CASE DIAGRAM

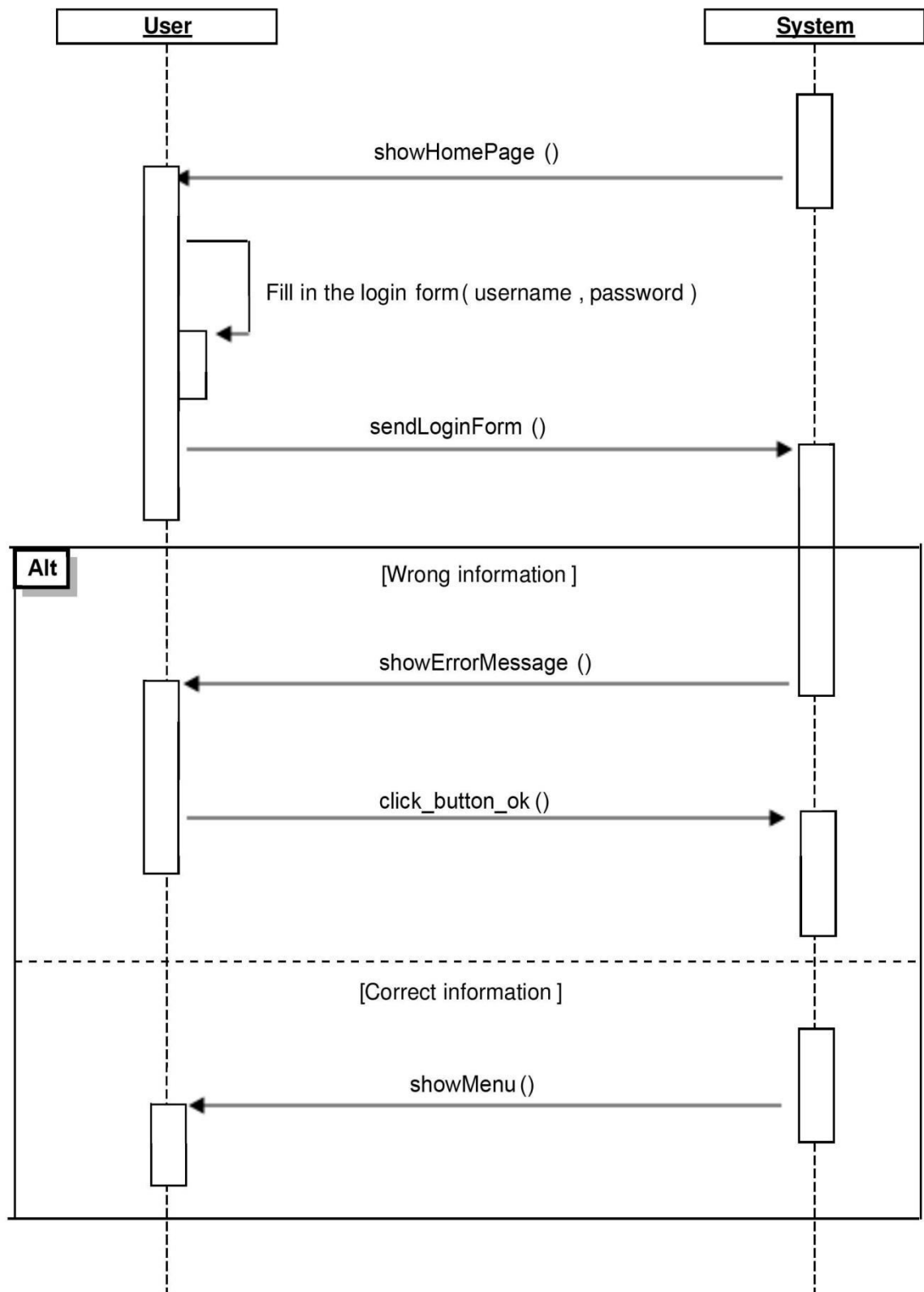


### 5.3 USE CASE DESCRIPTION AND RELATIVE SEQUENCE DIAGRAM

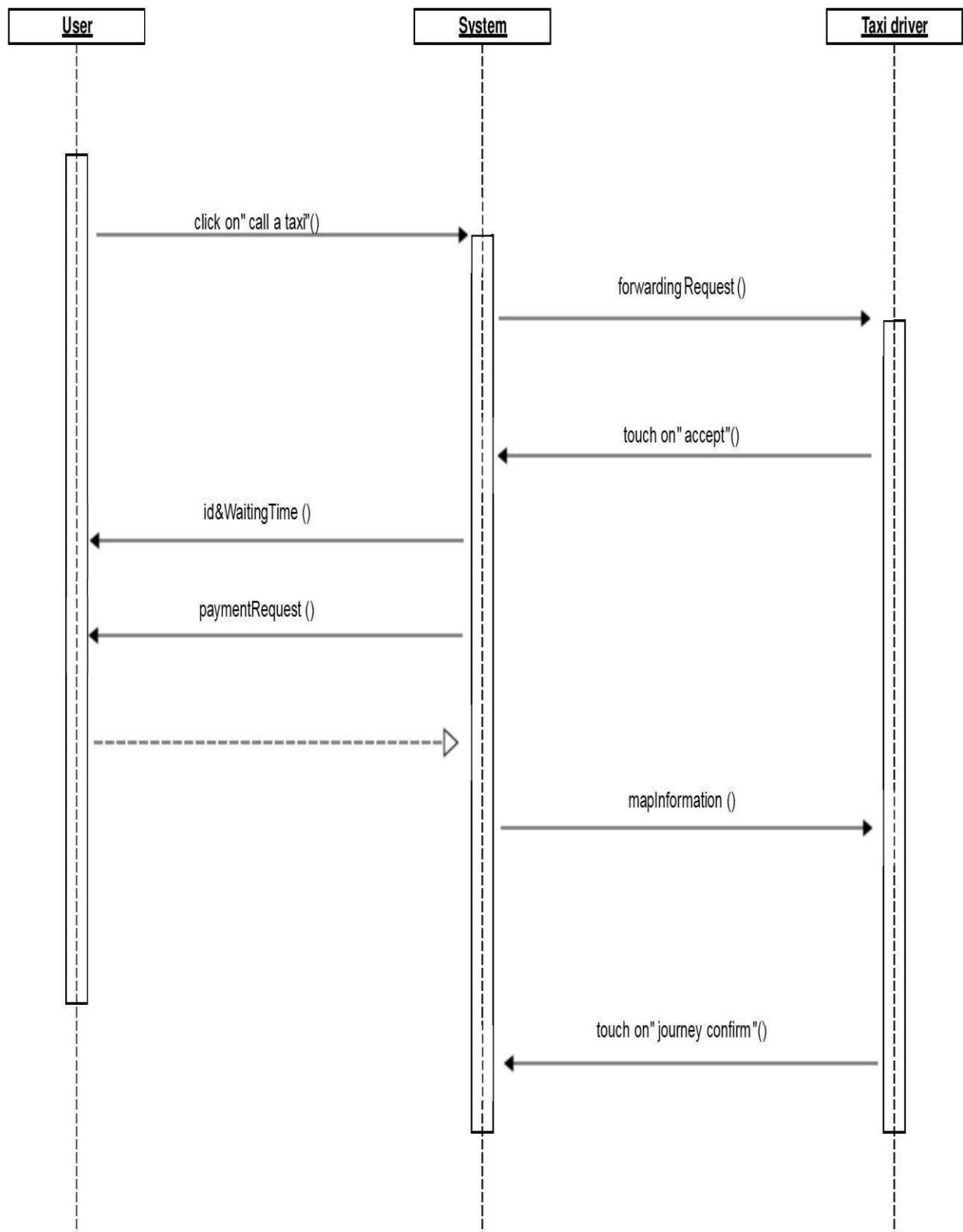
Name	Sign up
Actors	Guest
Entry Conditions	The guest isn't registered.
Flow of events	<ul style="list-style-type: none"><li>• The guest enters the website or the app;</li><li>• The guest clicks on the "SIGN UP" button;</li><li>• The guest fills in the form where he has to write:<ul style="list-style-type: none"><li>-Name</li><li>-Surname</li><li>-Email</li><li>-Address</li><li>-Username</li><li>-Password</li><li>-Date of birth</li><li>-Telephone number</li></ul></li><li>•The guest inserts the number of credit card and the CV2 security code.</li><li>• The guest clicks the "DONE" button;</li><li>• The system shows him the menu.</li></ul>
Exit conditions	Registration successfully done.
Exceptions	An exception can be caused if the username the guest inserts already exists or if some field aren't filled.



Name	Log in
Actors	User
Entry Conditions	User has successfully signed up to the system.
Flow of events	<ul style="list-style-type: none"> <li>• The user enters the web site or the app.</li> <li>• The user fills in the text fields username and password.</li> <li>• The user clicks on the “LOG IN” button.</li> </ul>
Exit conditions	The system shows the user the menu.
Exceptions	The password and/or username inserted by the user are wrong. The System shows an error message to the user.



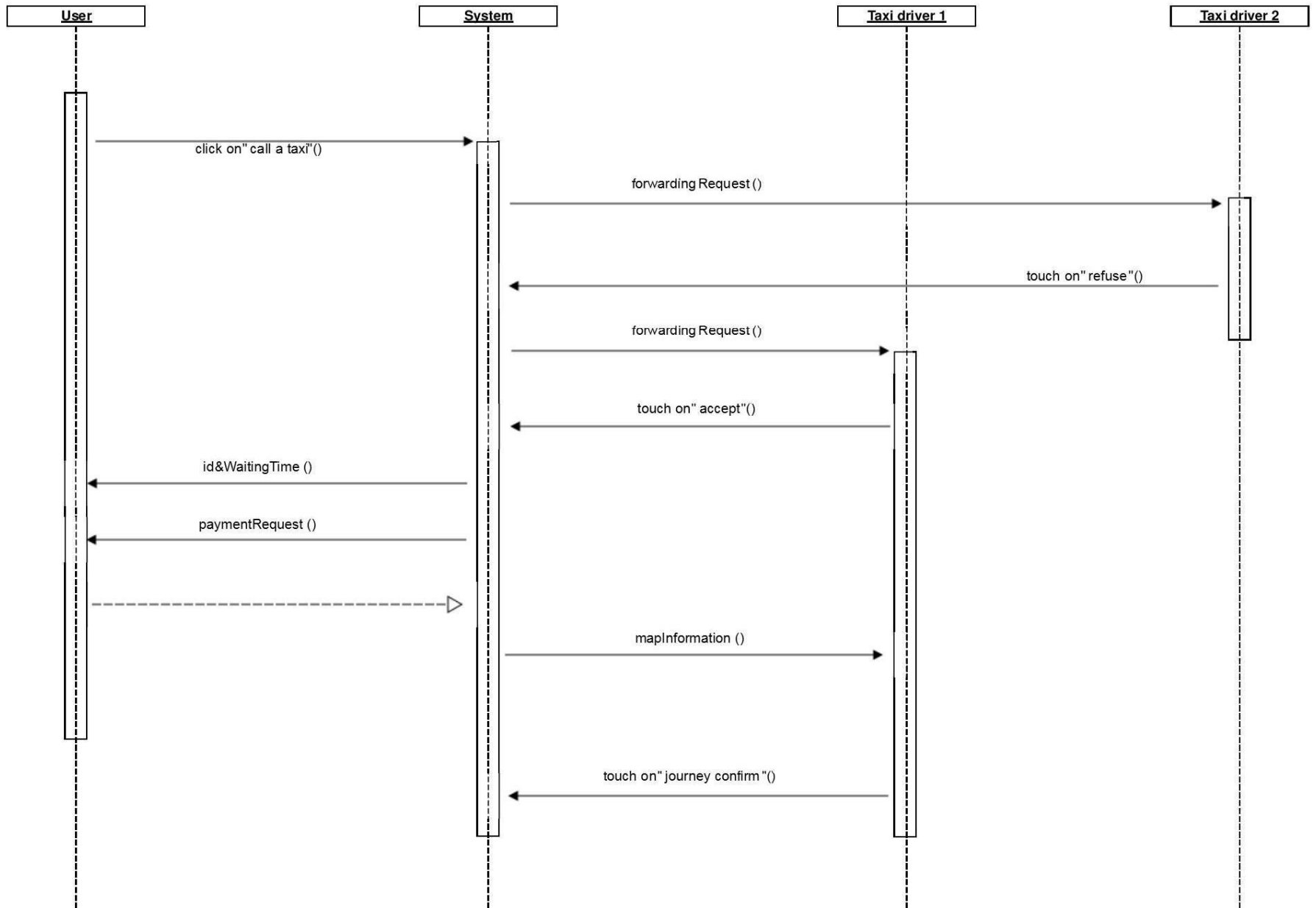
Name	Calling of a taxi with acceptance of the taxi driver
Actors	User, taxi driver
Entry Conditions	The user must be logged in The taxi driver must be in service
Flow of events	<ul style="list-style-type: none"> <li>• The user clicks on the button "Call a taxi" in the menu.</li> <li>• The user inserts the origin and destination places.</li> <li>• The user waits while the system looks for an available taxi.</li> <li>• The taxi driver receives the call request with the position of the passenger and the destination and has 20 seconds to choose to accept or not.</li> <li>• The taxi driver touches "accept" on the screen in the car and heads for the passenger.</li> <li>• The user receives the identifier of the taxi and the waiting time.</li> <li>• The system takes from the user's credit card the amount due for the journey.</li> <li>• When the passenger leaves the taxi at his destination, the driver touches "journey confirm" on the screen and becomes available again.</li> </ul>
Exit conditions	The user is where he asked to go.
Exceptions	The user could not be present, in this case the taxi driver notifies, by touching "user absent", the system that has already taken upfront the payment for the journey.





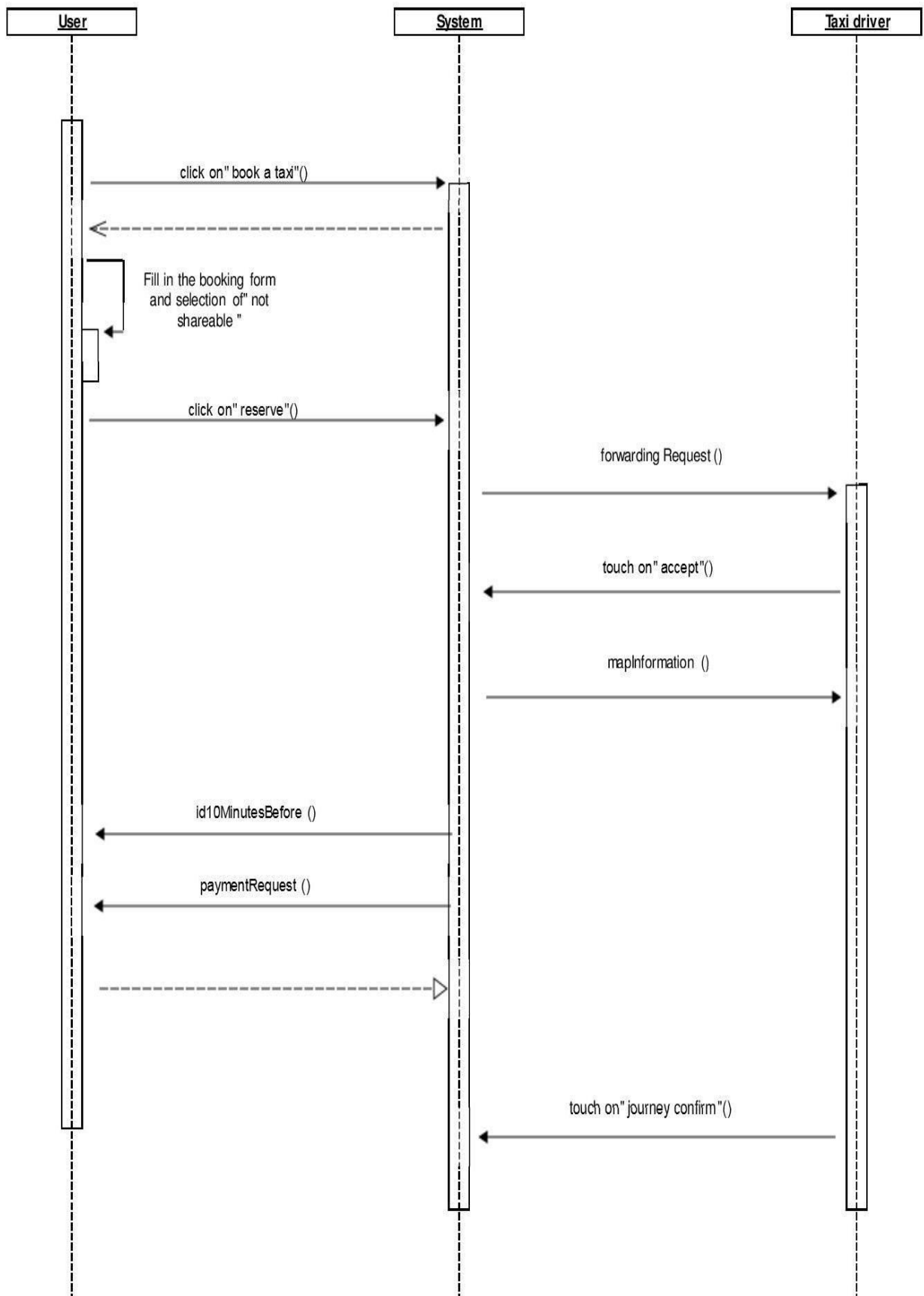
Name	Calling of a taxi with the driver's refusal
Actors	User, taxi drivers
Entry Conditions	The user must be logged in The taxi driver must be in service
Flow of events	<ul style="list-style-type: none"> <li>• The user clicks on the button "Call a taxi" in the menu.</li> <li>• The user inserts the origin and destination places.</li> <li>• The user waits while the system looks for an available taxi.</li> <li>• The first taxi driver in the queue receives the call request with the position of the passenger and the destination, and has 20 seconds to choose to accept or not.</li> <li>• The driver is not available to go and take the user, so he touches "refuse" on the screen.</li> <li>• The system moves the taxi driver in the back of the queue and contact the next.</li> <li>• If also the next taxi driver isn't available, the situation is repeated until someone is available to accept the request.</li> <li>• The available taxi driver touches "accept" on the screen in the car and heads for the passenger.</li> <li>• The user receives the identifier of the taxi and the waiting time.</li> <li>• The system takes from the user's credit card the amount due for the journey.</li> <li>• When the passenger leaves the taxi at his destination, the driver touches "journey confirm" on the screen and becomes available again.</li> </ul>

Exit conditions	The user is where he asked to go.
Exceptions	The user could not be present, in this case the taxi driver notifies the system, by touching "user absent", that has already taken upfront the payment for the journey.



Name	Reservation of a not shareable taxi
Actors	User, taxi driver
Entry Conditions	The user must be logged in The taxi driver must be in service. There are at least two hours before the user wants to take the taxi.
Flow of events	<ul style="list-style-type: none"> <li>• The user clicks "book a taxi".</li> <li>• The user completes the booking form with departure, destination and time of meeting.</li> <li>• The user selects "not shareable" in the appropriate field and click "reserve".</li> <li>• The first taxi driver in the queue receives the call request with the position of the passenger and the destination, and has 20 seconds to choose to accept or not.</li> <li>• The driver is not available to go and take the user, so he touches "refuse" on the screen.</li> <li>• The system moves the taxi driver in the back of the queue and contact the next.</li> <li>• If also the next taxi driver isn't available, the situation is repeated until someone is available to accept the request.</li> <li>• The available taxi driver touches "accept" on the screen in the car and he arranges to be present at the time received by the user.</li> <li>• 10 minutes before the meeting, the user receives the booking confirmation and the identification of the taxi.</li> <li>• The system takes from the user's credit card the amount due for the</li> </ul>

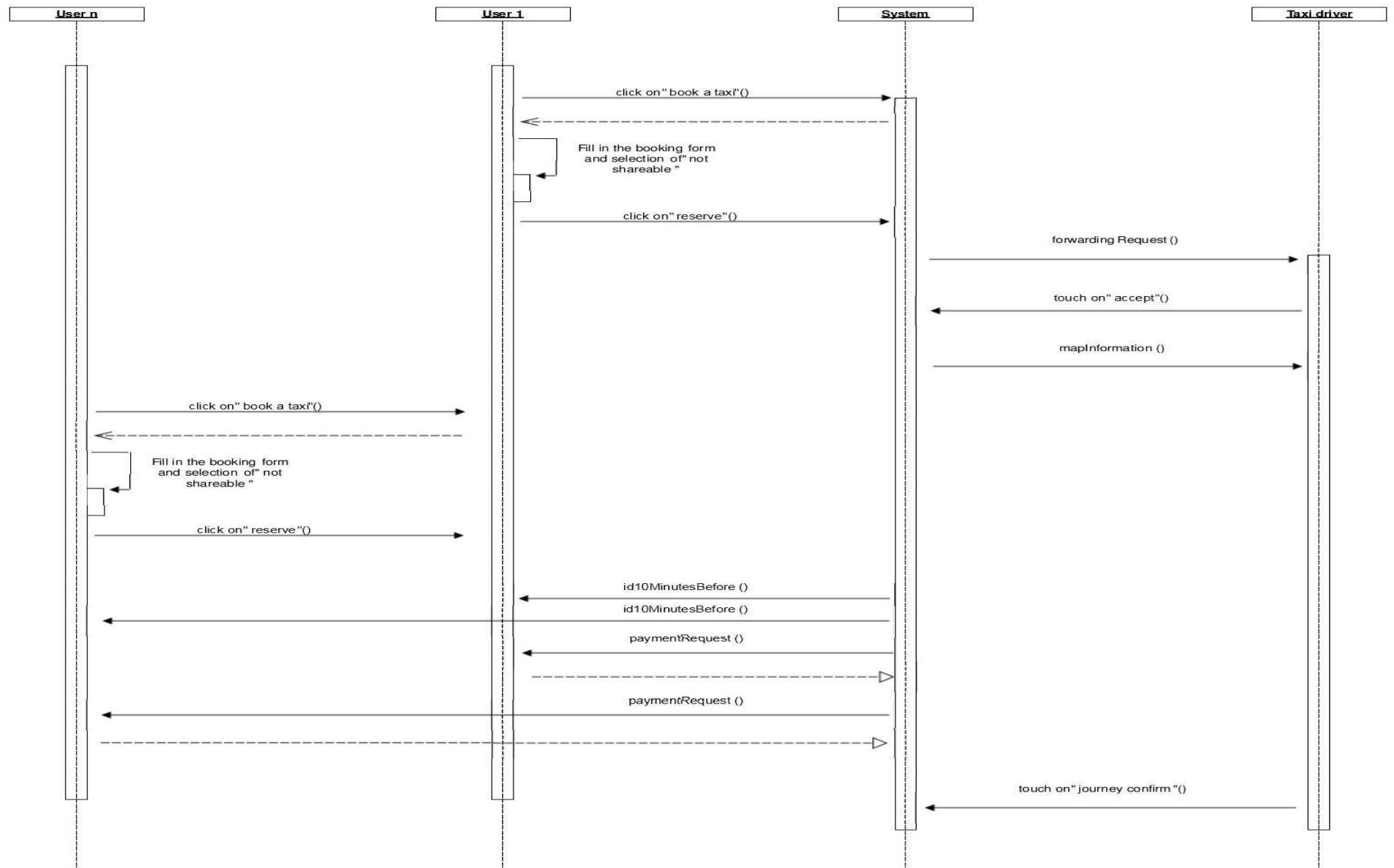
	<p>journey.</p> <ul style="list-style-type: none"> <li>• When the passenger leaves the taxi at his destination, the driver touches "journey confirm" on the screen and becomes available again.</li> </ul>
Exit conditions	The user is where he asked to go.
Exceptions	The user could not be present, in this case the taxi driver notifies, by touching "user absent", the system that has already taken upfront the payment for the journey.



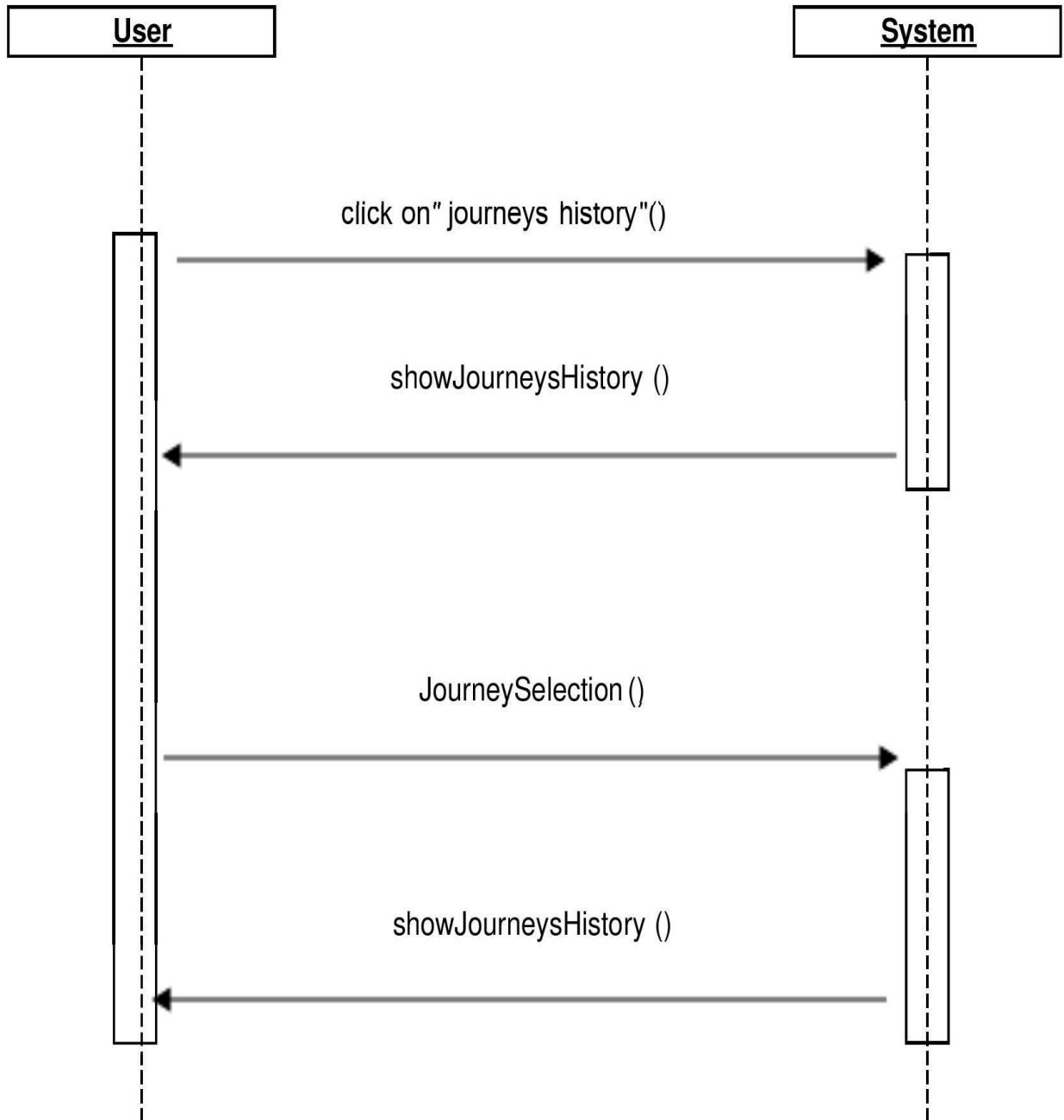
Name	Reservation of a shareable taxi
Actors	Users, taxi driver
Entry Conditions	<p>The user must be logged in</p> <p>The taxi driver must be in service</p> <p>There are at least two hours before the user wants to take the taxi.</p>
Flow of events	<ul style="list-style-type: none"> <li>• The user clicks "book a taxi".</li> <li>• The user completes the booking form with departure, destination and time of meeting.</li> <li>• The user selects "shareable" in the appropriate field, indicates the acceptable range of extension time of the journey, the number of people and then click "reserve".</li> <li>• The first taxi driver in the queue receives the call request with the position of the passenger and the destination, and has 20 seconds to choose to accept or not.</li> <li>• The driver is not available to go and take the user, so he touches "refuse" on the screen.</li> <li>• The system moves the taxi driver in the back of the queue and contact the next.</li> <li>• If also the next taxi driver isn't available, the situation is repeated until someone is available to accept the request.</li> <li>• The available taxi driver touches "accept" on the screen in the car and he arranges to be present at the time received by the user.</li> <li>• The situation is repeated if other users want to make a journey</li> </ul>

	<p>shareable that the system detects compatible with the range selected by all users. In this case the taxi driver is obliged to accept and therefore receives meeting time and place of new users.</p> <ul style="list-style-type: none"> <li>• 10 minutes before the meeting, users receive the booking confirmation and the identification of the taxi.</li> <li>• The system takes from users' credit card the amount due for the journey.</li> <li>• When a passenger leaves the taxi at his destination, the driver touches "journey confirm" on the screen and continues the journey.</li> <li>• When the last passenger has arrived, the taxi driver becomes available again.</li> </ul>
Exit conditions	Users are where they asked to go.
Exceptions	A user could not be present, in this case the taxi driver notifies, by touching "user absent", the system that has already taken upfront the payment for the journey.

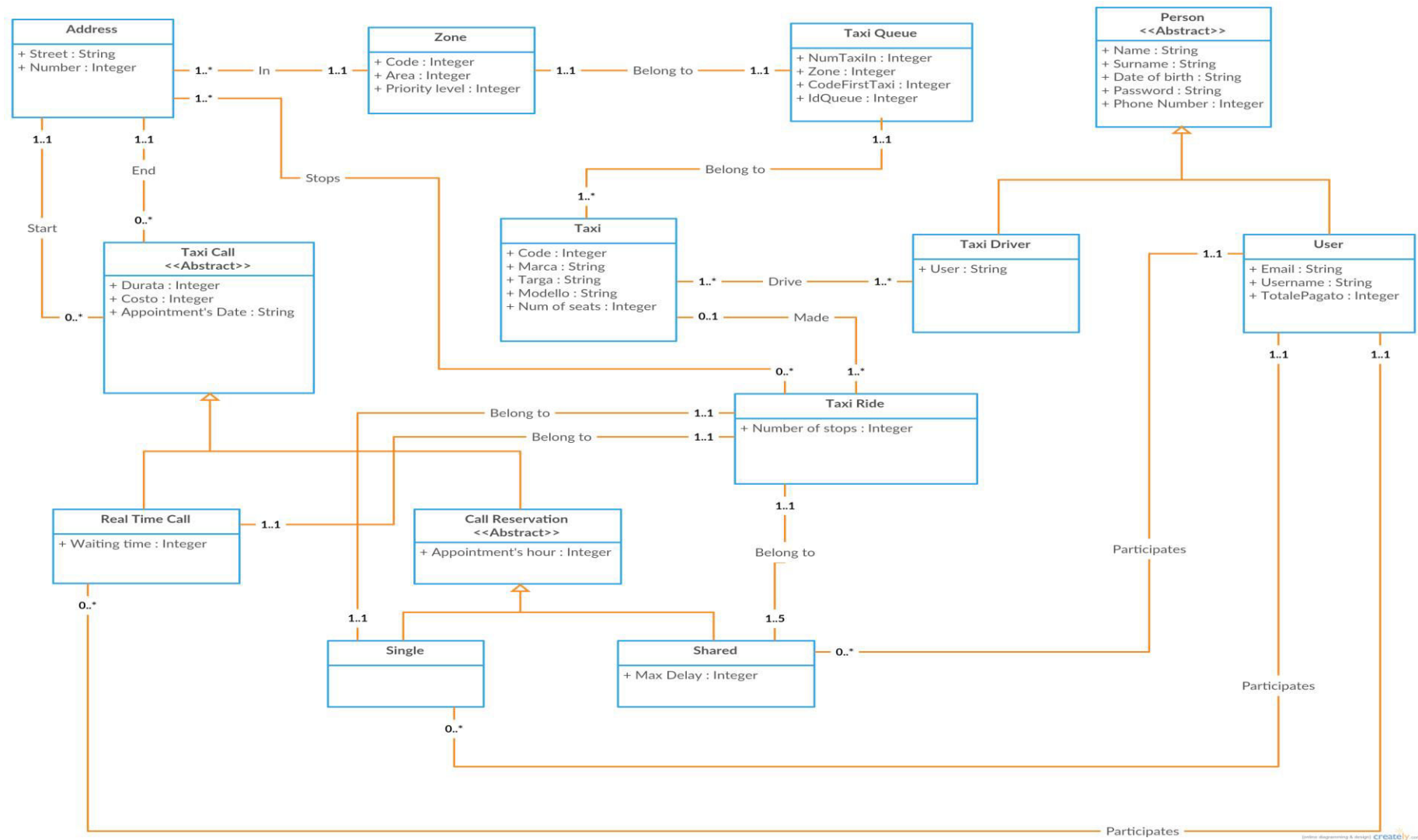




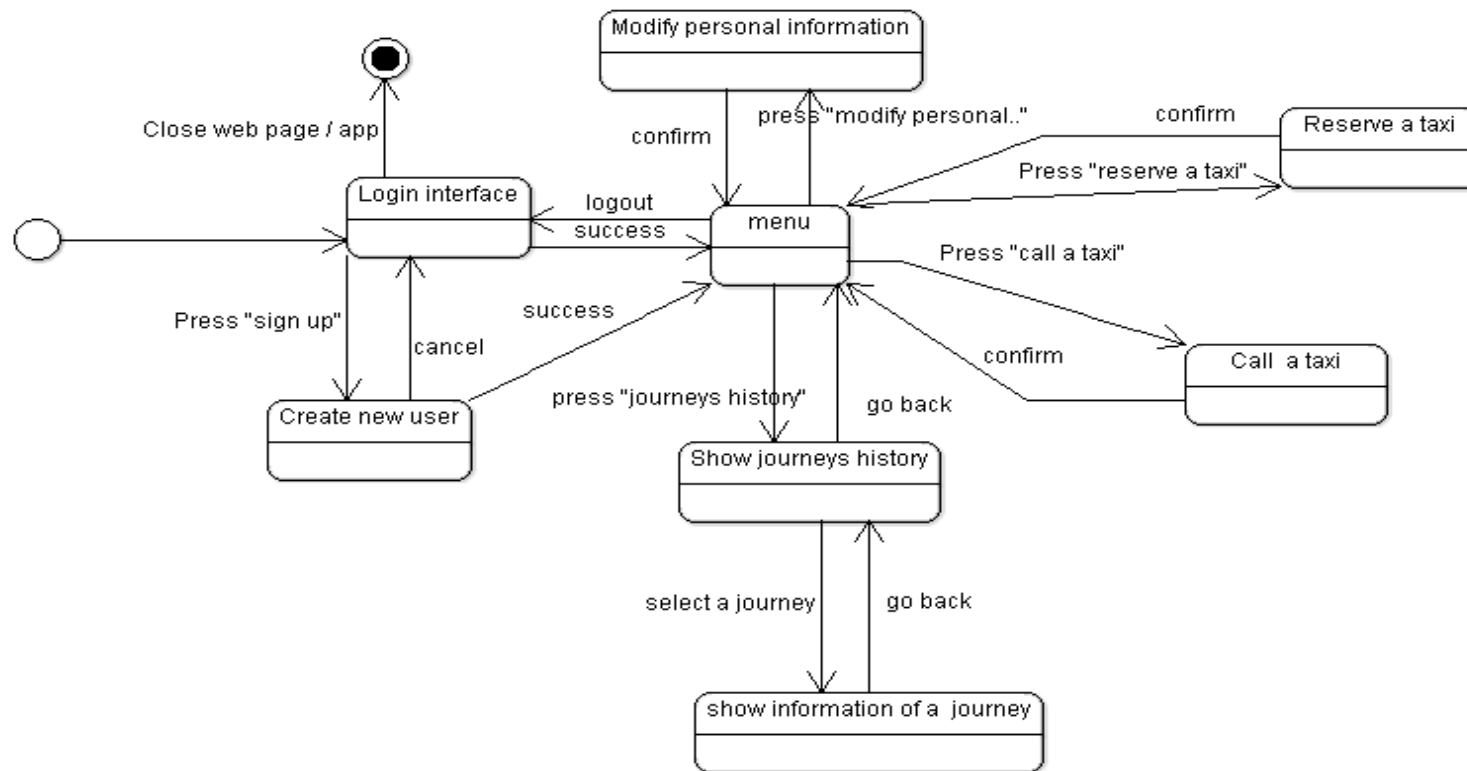
Name	Display information of a journey
Actors	User
Entry Conditions	The user must be logged in
Flow of events	<ul style="list-style-type: none"> <li>• The user visualizes a list of journeys done by clicking on "journeys history"</li> <li>• The user selects a journey and a window appears with the following information: <ul style="list-style-type: none"> <li>- identification of the taxi;</li> <li>- places of origin and destination;</li> <li>- arrival and departure times;</li> <li>- way;</li> <li>- duration of the journey;</li> <li>- fare paid;</li> </ul> </li> </ul>
Exit conditions	The system shows the user the list of the selected journey' information.
Exceptions	No exceptions.



## 5.4 CLASS DIAGRAM



## 5.5 STATE CHART DIAGRAMS



## **6. ALLOY MODELING**

### **6.1 ALLOY CODE**

```
sig User{  
    calls: set Call,  
}
```

```
sig TaxiDriver{  
    ownTaxi: Taxi,  
}
```

```
sig Taxi{  
    driver: TaxiDriver,  
    taxiRun: Run,  
    ownQueue: Queue,  
    position: Addr,  
}
```

```
sig Queue{  
    queueZone: Zone,  
    taxiList: set Taxi,  
}
```

```
sig Zone{  
    zoneQueue: Queue,  
    priority: Integer,  
}
```

```
sig Addr{  
    addrZone: Zone,  
}
```

```
abstract sig Call{  
    callUser: User,  
    callRun: Run,  
    startAddr: Addr,  
    endAddr: Addr,  
}  
  
    startAddr != endAddr  
}
```

```
sig RealTime extends Call{  
  
}
```

```
abstract sig Reservation extends Call{}
```

**sig SingleRes extends Reservation{**

**}**

**sig SharedRes extends Reservation{**

**}**

**sig Run{**

**runTaxi: Taxi,**

**runPassengers: set User,**

**callList: set Call,**

**stops: set Addr,**

**{**

**#runPassengers > 0**

**#runPassengers <= 5**

**#callList >0**

**#callList <= 5**

**}**

**FACT:**

**fact{**

**no disj t1,t2 : Taxi | t1.taxiRun = t2.taxiRun**

**all d : TaxiDriver, t: Taxi | d.ownTaxi = t iff t.driver = d**

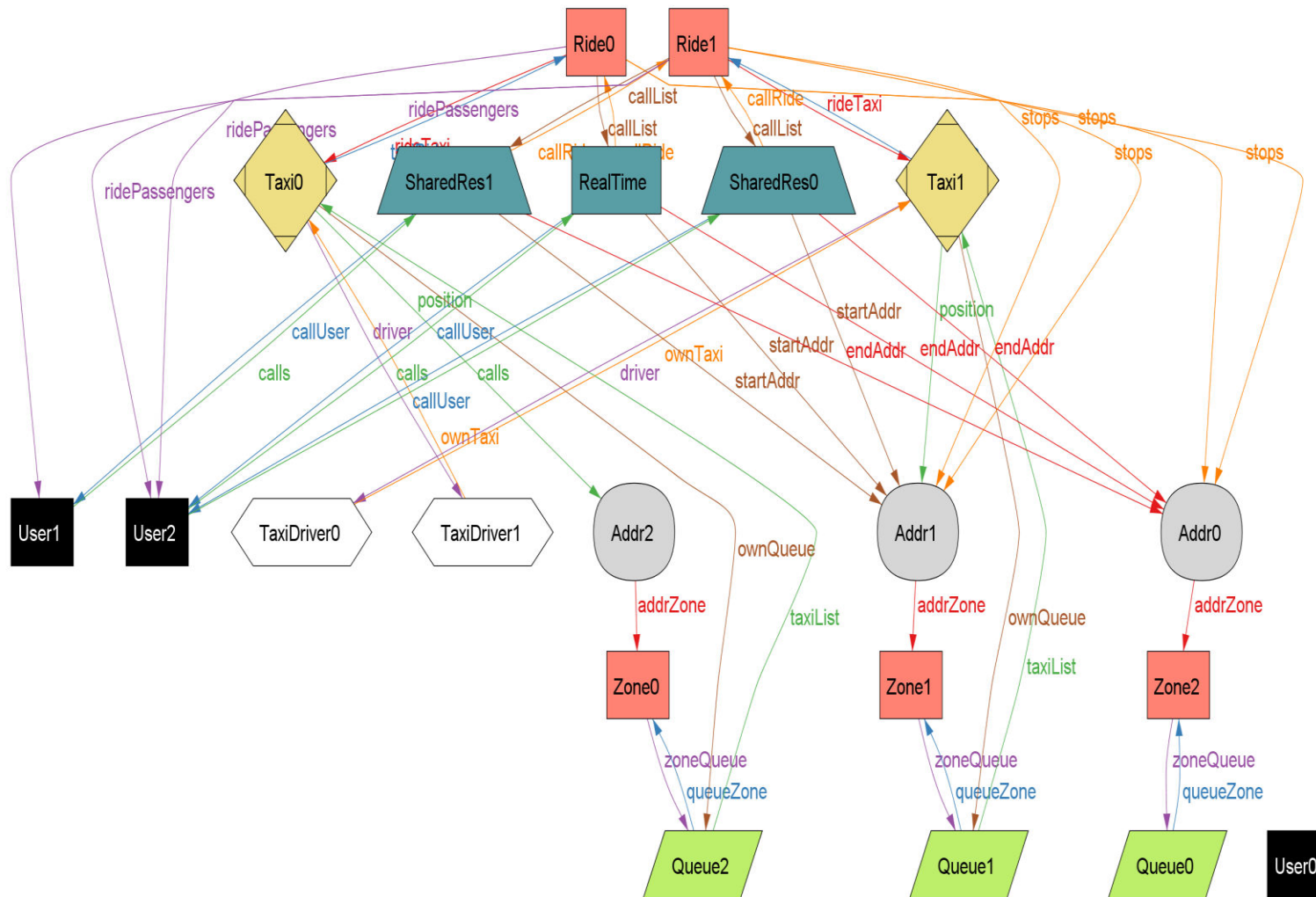
**all q : Queue , z : Zone | q.queueZone = z iff z.zoneQueue = q**

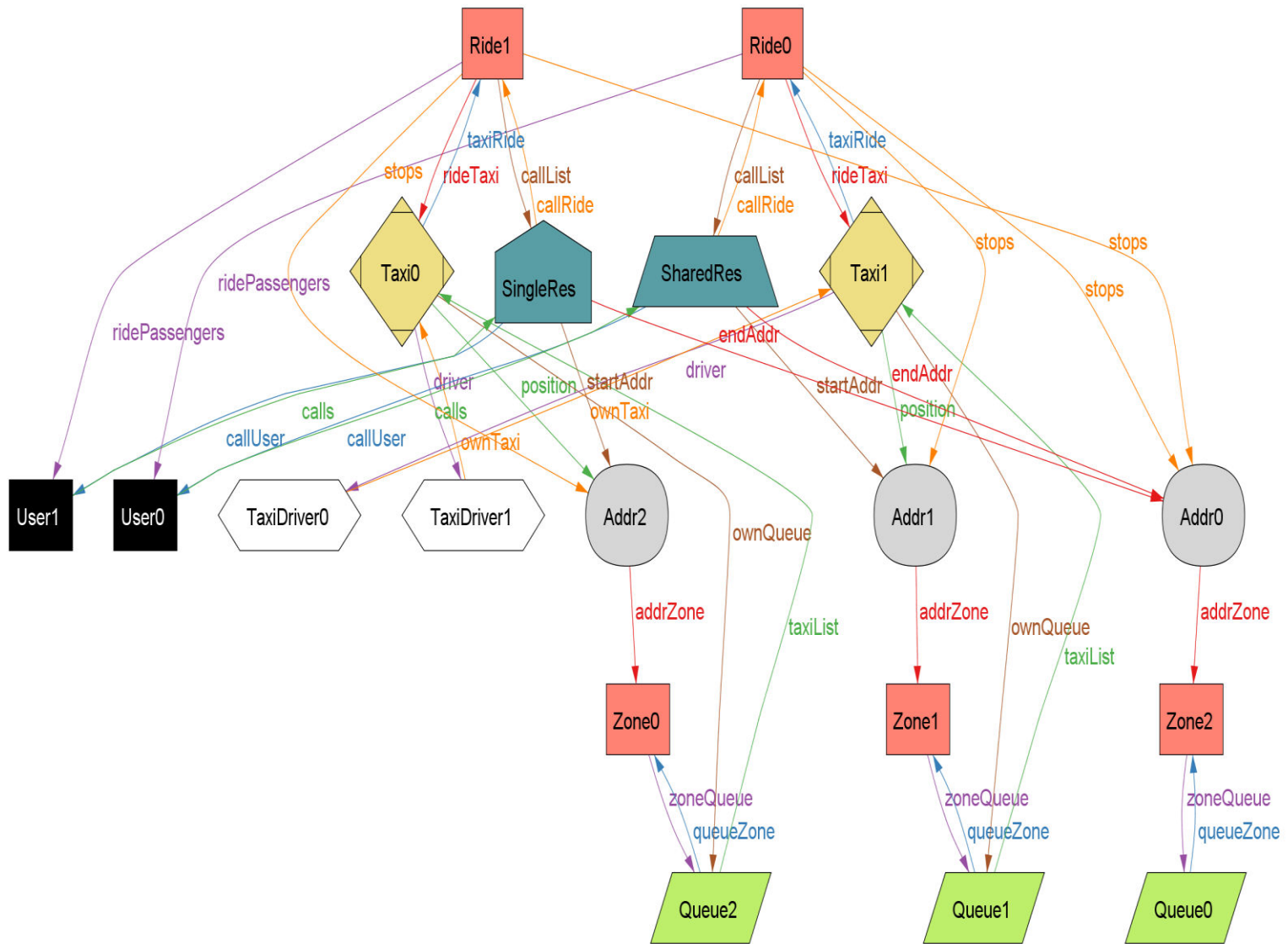


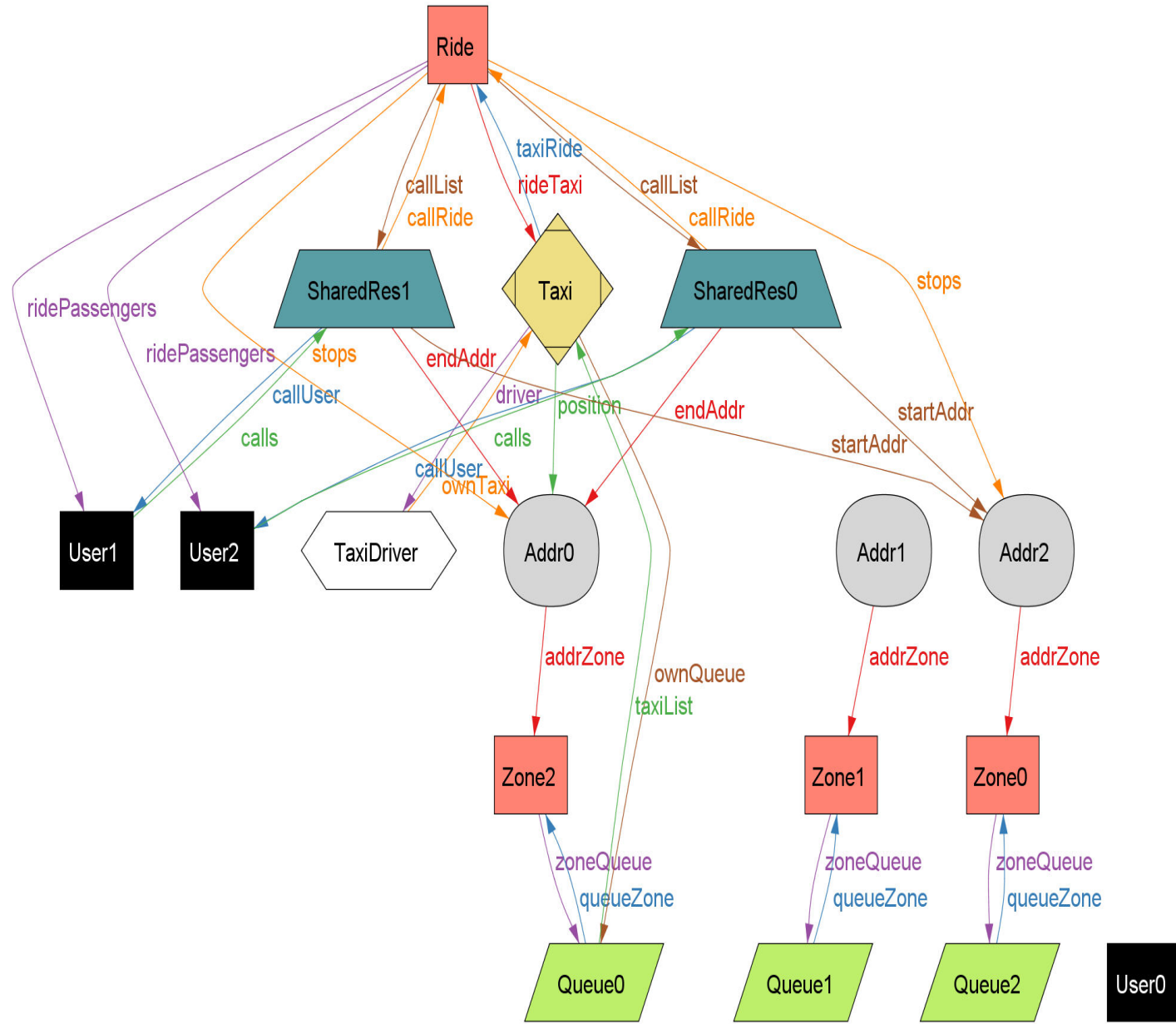
all t : Taxi, q : Queue | t.ownQueue = q iff t in q.taxiList  
 no disj r1,r2 : Run | r1.runTaxi = r2.runTaxi  
 all r : Run, t : Taxi | t.taxiRun = r iff r.runTaxi = t  
 all u: User, c: Call | c.callUser = u iff c in u.calls  
 no c1,c2: SharedRes | c1 !=c2 and c1.callUser = c2.callUser and c1.callRun = c2.callRun  
 no c1: SingleRes, c2: Call | c1 != c2 and (c1.callRun = c2.callRun or one r : Run | c1 in r.callList and c2 in r.callList)  
 no c1: RealTime, c2: Call | c1 != c2 and (c1.callRun = c2.callRun or one r : Run | c1 in r.callList and c2 in r.callList)  
 no disj r1,r2:Run, c: Call | c in r1.callList and c in r2.callList  
 all c: Call, r:Run | c in r.callList iff c.callRun = r  
 all u: User, r: Run | u in r.runPassengers iff one c : Call | c in u.calls and c in r.callList  
 all c1,c2: SharedRes | c1.callRun = c2.callRun implies c1.startAddr.addrZone = c2.startAddr.addrZone  
 all t: Taxi | t.ownQueue = t.position.addrZone.zoneQueue  
 all r: Run, c: Call | c in r.callList implies (c.startAddr in r.stops and c.endAddr in r.stops)

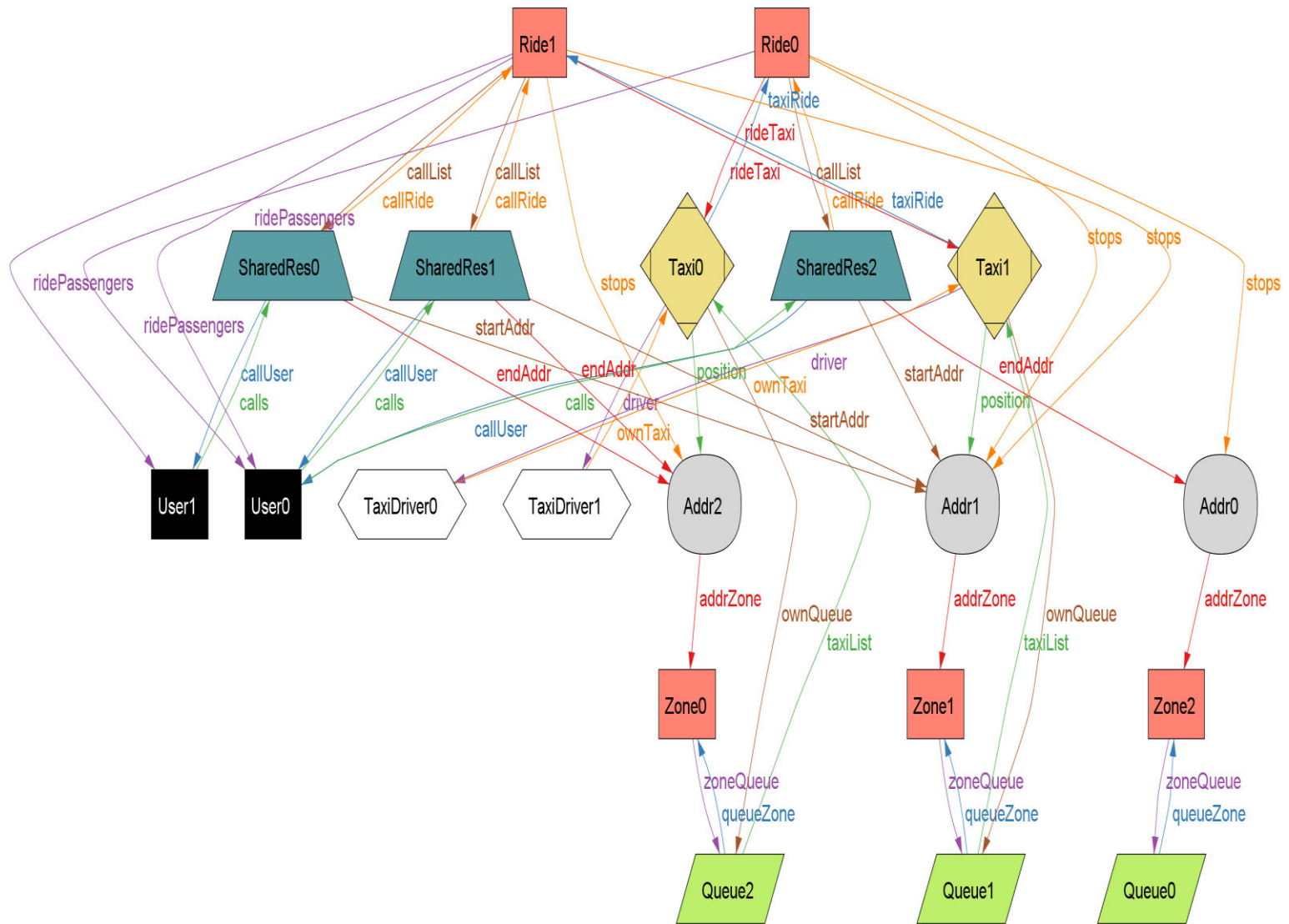
}

## 6.2 ALLOY WORLDS









## **7. DOCUMENT PERSPECTIVE**

### **7.1 REFERENCE DOCUMENTS**

- Specification Document: Assignments 1 and 2.pdf
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications

### **7.2 FUTURE IMPLEMENTATION**

- Improve the security
- New functions like the possibility of made a feedback on the taxi drivers
- Introduce statistics to better meet the demands of the different areas

### **7.3 HARDWARE CONSTRAINTS**

For the server there are no limits because is all maintained from Google, for the smartphone application its size is not over 100MB and is portable over the three main S.O. android, Microsoft and IOS with no constraints about the hardware (RAM,CPU).

## **8. USED TOOLS**

Microsoft Office: to write this document.  
Creately tool: to draw the Class Diagram.  
Gliffy.com and ArgoUML: to create the other UML diagrams.  
Alloy Analyzer 4.2: to prove the consistency of our model.

**Hours of work:** Vincenzo about 25 hours

Simone about 25 hours

Matteo about 25 hours