POLITECNICO DI MILANO
A.Y 2015/2016

Requirements Analysis and Specification Document

Software Engineering 2 project: myTaxiService

Krishnan Ranjithkumar
Korda Petar

Contents

1 Introduction

1.1 Purpose

This document represents the Requirement Analysis and Specification Document (RASD) for the software system myTaxiService, and the main goal of this document is to completely describe the system in terms of functional and non-functional requirements, analyze the needs of the customer, and represent the constraints and the limitations of the software. Also typical use cases and scenarios will be presented.

This document is intended to be used by developers, programmer and testers, who will implement and test the system, system analysts and requirement analysts for inter related systems, project managers, customers and users of the system.

1.2 Scope

Software system myTaxiService is a project meant to optimize the taxi services of the city, by providing the mobile and web application that will allow users to request a taxi, or book the ride in advance and cancel the ride. It will also provide a fair management of taxi queues for the taxi drivers, and maximize the profit for the city.

In particular, city is divided into taxi zones, and each zone is assigned to a taxi queue (the system automatically calculates the distribution of taxis in zones based on the GPS information it receives from taxis). If the taxi is available (this information is provided by the taxi driver, who informs the system about his/hers availability through a mobile application), its identifier is stored in a queue of taxis in the corresponding zone. When a user requests a taxi from a zone (through a mobile or web application), the system forwards it to the first taxi queuing in that zone. Taxi driver can then confirm or deny the request through the mobile application. If the request is denied system will forward the request to a next taxi in the queue (and move the first taxi to the end of the queue), or if the queue is empty it will make sure that another taxi gets a request. When the taxi driver confirms the request, the passenger will be informed by the system about the taxi identifier and the waiting time. The passenger can also book a taxi, specifying the starting point and the destination, and the system then allocates a taxi 10 minutes before the meeting time with the passenger.

The system will also provide programmatic interface to enable development of additional services on top of the basic one.

1.3 Actors

- **Visitor** - A person that is not registered for the service, or a registered user who is not logged in to the application. Visitor can only see the home page, registration from and log in form.
- **Registered users** - A person that has an account for myTaxiService.
- **User** - A logged in registered user.
- **Passenger** - A user that uses myTaxiService for the means of requesting and reserving a taxi ride.
- **Driver** - A user of the mobile application, that drives a taxi, and uses the application for receiving the requests for the ride.
- **Administrator** - Supervisor, manages databases, and also takes care of drivers responsibility.

1.4 Goals

List of the goals for myTaxiService:

- **[G1]** Allow visitors to register to the application.
- **[G2]** Allow registered users to log in to the application, with same credentials for both the web and the mobile application.
- **[G3]** Passenger can request a taxi ride by specifying his starting point and his destination.
- **[G4]** The passenger will be notified on request with the taxi code, waiting time and the fare amount.
- **[G5]** The passenger can either accept or cancel the taxi proposal.
- **[G6]** If the taxi proposal is accepted, taxi is dispatched and will arrive at the requested location in 15 minutes maximum.

- **[G7]** Passenger can reserve a taxi ride specifying his starting point, his destination and the desired time, at least two hours before the ride

- **[G8]** When the passenger reserves a taxi ride, he will be informed with a code of the incoming taxi 10 minutes before the specified time of the ride.

- **[G9]** Passenger can cancel his request.

- **[G10]** Passenger can cancel his reservation

- **[G11]** Available driver can confirm or reject the certain request for the ride.

1.5 Definitions, Acronyms, Abbreviations

1.5.1 Definitions

- **Available driver** - is a driver who informed the system that he is available for the rides.

- **Proposed taxi** - is a taxi, that is not yet allocated by the system, but accepted the request.

1.5.2 Acronyms

- **RASD** - Requirements Analysis and Specification Document

- **DBMS** - Data Base management system

- **DB** - Data Base

- **API** - Application Programming Interface

- **OS** - Operating System

1.5.3 Abbreviations

- **[Gn]** - n-th Goal

- **[Rn]** - n-th Functional Requirement

- **[Dn]** - n-th Domain Assumption

1.6 References

- Specification Document: myTaxiService Project, AA 2015-2016

.

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.
- Alloy tutorial on alloy.mit.edu

1.7 Document Overview

This document is structured in four parts:

- **Section 1** - Introduction. It gives a description of document and some basic information about software.
- **Section 2** - Overall Description. Gives general information about the software product with more focus about constraints and assumptions.
- **Section 3** - Specific Requirements. This part lists requirements, typical scenarios and use cases. To give an easy way to understand all functionality of this software, in this section UML diagrams are given.
- **Section 4** - Appendix. This part contains some information about the attached .als file and some described screenshot of software used to generate it.

2 Overall Description

2.1 Product Perspective

The system myTaxiService is a project being developed for the government of city. It will be released as a new, self-contained product, not integrated with other existing systems. Users will be able to access the service through a web and mobile application. The application will have a internal interface for administration, and it will provide API to enable the development of additional services on top of the basic one.

2.2 Constraints

2.2.1 Regulatory Policies

- Privacy Policy - It fulfills a legal requirement to protect a customer or client's privacy.
- Online Payment Policy - satisfies a safe online transactions.

2.2.2 Hardware Limitations

- ➢ myTaxiService doesn’t have any hardware limitations.

2.2.4 Parallel operations

myTaxiService must support parallel operations from different users when working with database and with all operation done by the user after connection.

2.2.5 Documents related

- Requirements and Analysis Specification Document (RASD).
- Design Document (DD).
- User’s Manual.
- Testing report.

2.3 Assumptions and Dependencies

- This system will only be used only for government taxi service.
- Taxi service owns the vehicles, and the drivers work for the government.
- Taxi service is available 24 hours a day.
- Each taxi has a mobile phone, GPS system and a driver.
- Passenger has to pay if he accepts the taxi proposal.
- If a passenger cancels a taxi ride, he will be refunded 50% of the amount he paid.
- If a passenger cancels a taxi reservation before 48 hours, he will be refunded 70% of the amount he paid.
- If the passenger is not available at the pick up point, taxi waits up to 15 minutes, and after that the reservation is canceled, and the passenger is refunded 30% of the amount he paid.
- Passenger can reserve a taxi at least 2 hours before the desired time, and at most 1 week before the desired time.

2.4 Future Possible Implementation

- **Taxi Sharing** - if several people from the same zone are requesting a taxi, and going to the same direction, they can share it and split the expenses.

- **myTaxiService Credits** - possibility to pay for the rides with credits (e-money), that you can charge at the taxi stations, with your card or at some shops.

3 Specific requirements

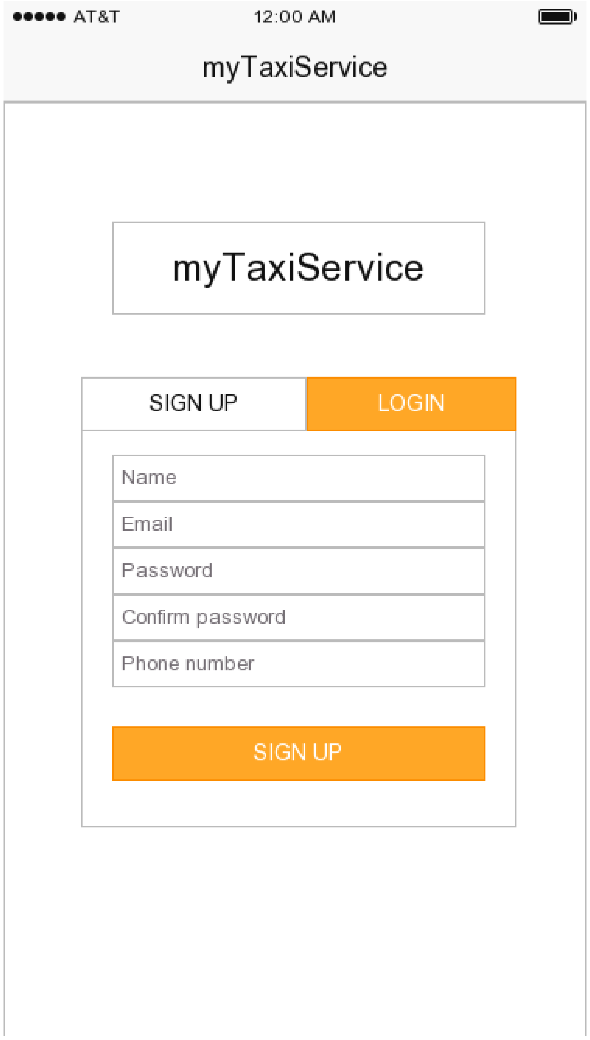
3.1 External Interface Requirements

3.1.1 User Interfaces

In the following sections, mockups of the graphical user interface for the mobile application will be presented. Mockups for the web application are not presented here, but the layout will be the same for both mobile and web application.

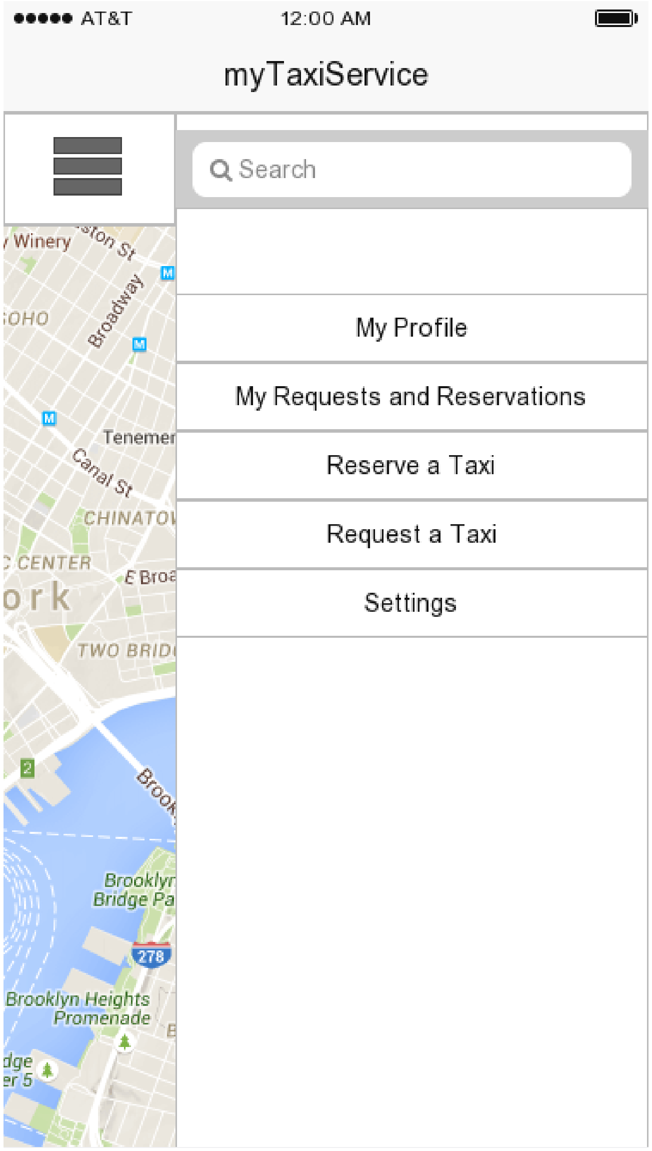
3.1.1.1 Registration and Log in

This is the page that a visitor sees. Visitors can register or log in with their credentials here.



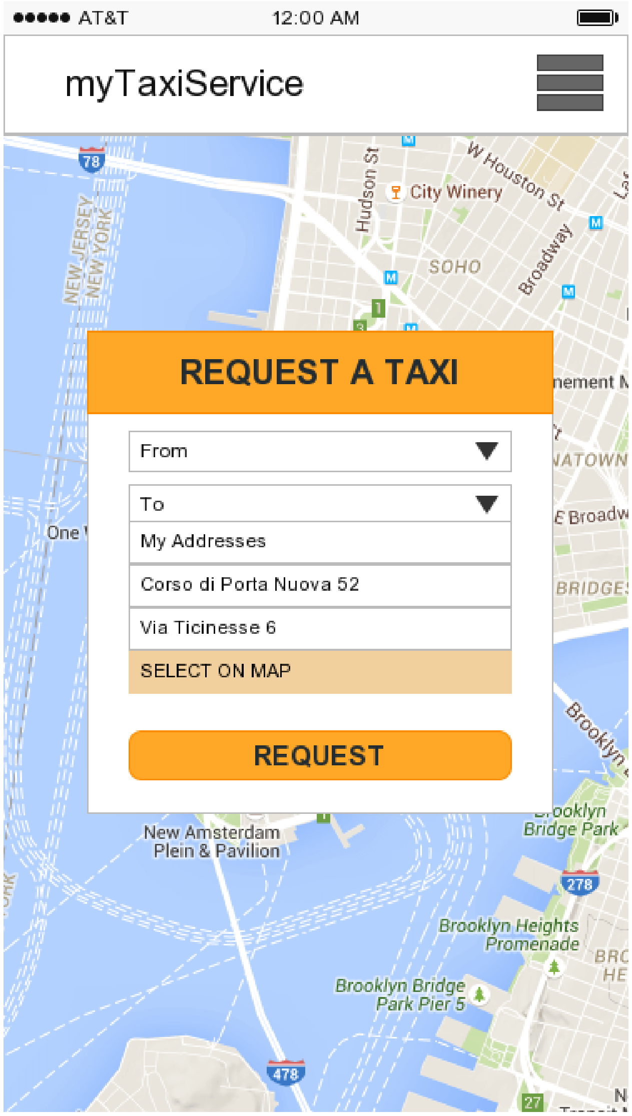
3.1.1.2 Side Bar

In the side bar user can find all of the relevant link to the application services, including requesting and reserving the taxi, and updating he's profile information or reviewing his request and reservations.



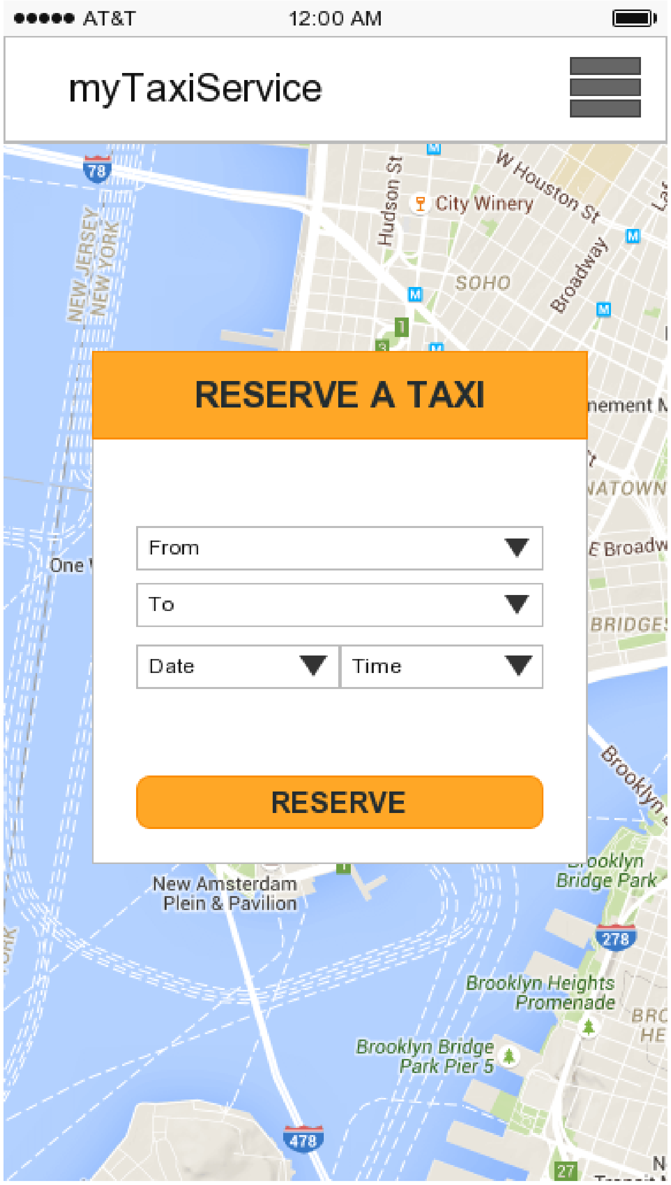
3.1.1.3 Request a Taxi

On this page user can request a taxi specifying his starting point and destination, either on the map or entering the addresses in the text fields, or selecting one of his saved addresses, or using his GPS location.



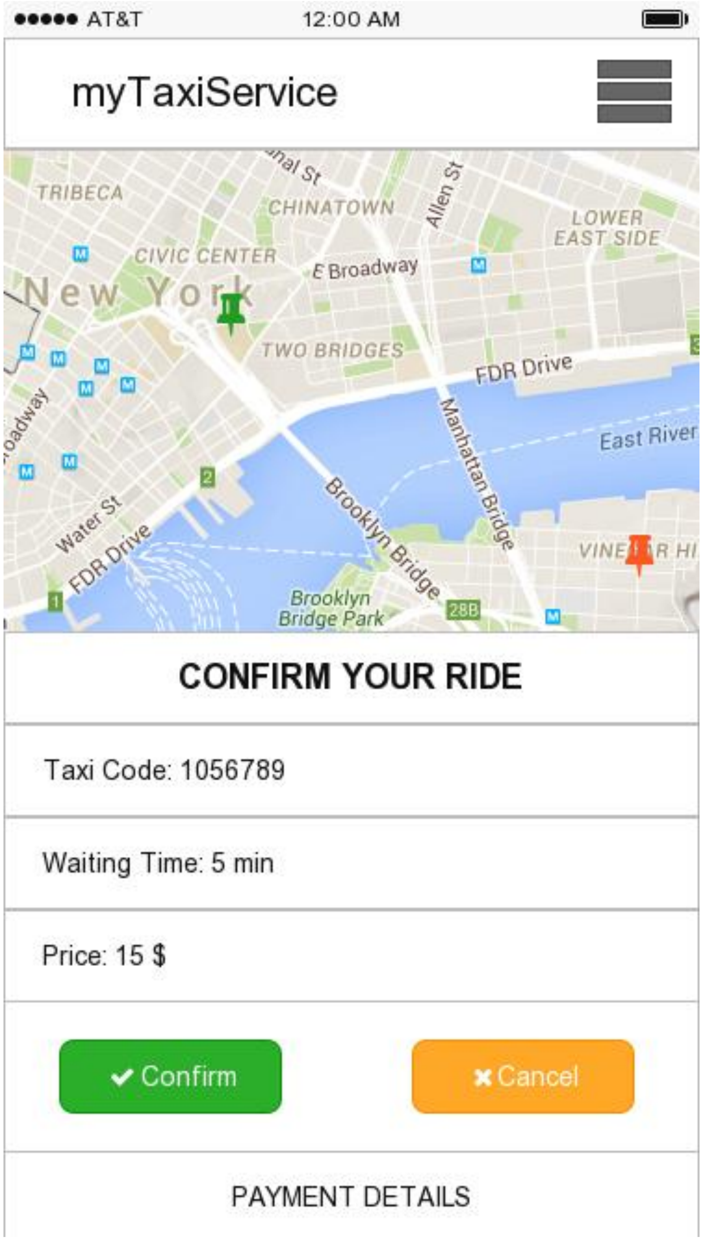
3.1.1.4 Reserving a Taxi

On this page user can reserve a taxi in advance, specifying his starting point and destination as well as the time, either on the map or entering the addresses in the text fields, or selecting one of his saved addresses.



3.1.1.5 Taxi ride confirmation page

This is the notification page with all the details for the ride, when the system finds an available taxi. User is prompted to confirm, and to pay for the ride.



CONFIRM YOUR RIDE

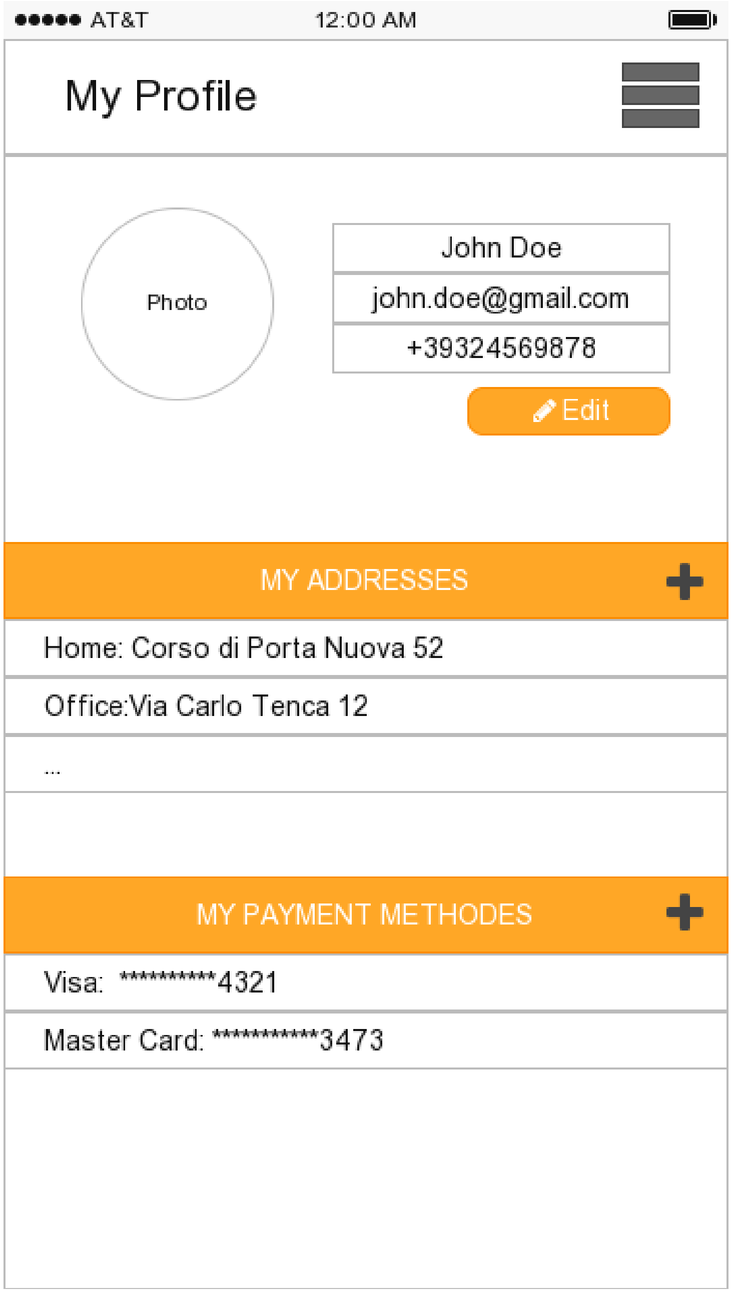
Taxi Code: 1056789

Waiting Time: 5 min

Price: 15 $

3.1.1.6 Profile

This is the profile page of the user, with all his personal information, that he can edit, as well as the list of his frequently used addresses and payment methods.



John Doe

john.doe@gmail.com

+39324569878

Home: Corso di Porta Nuova 52

Office:Via Carlo Tenca 12

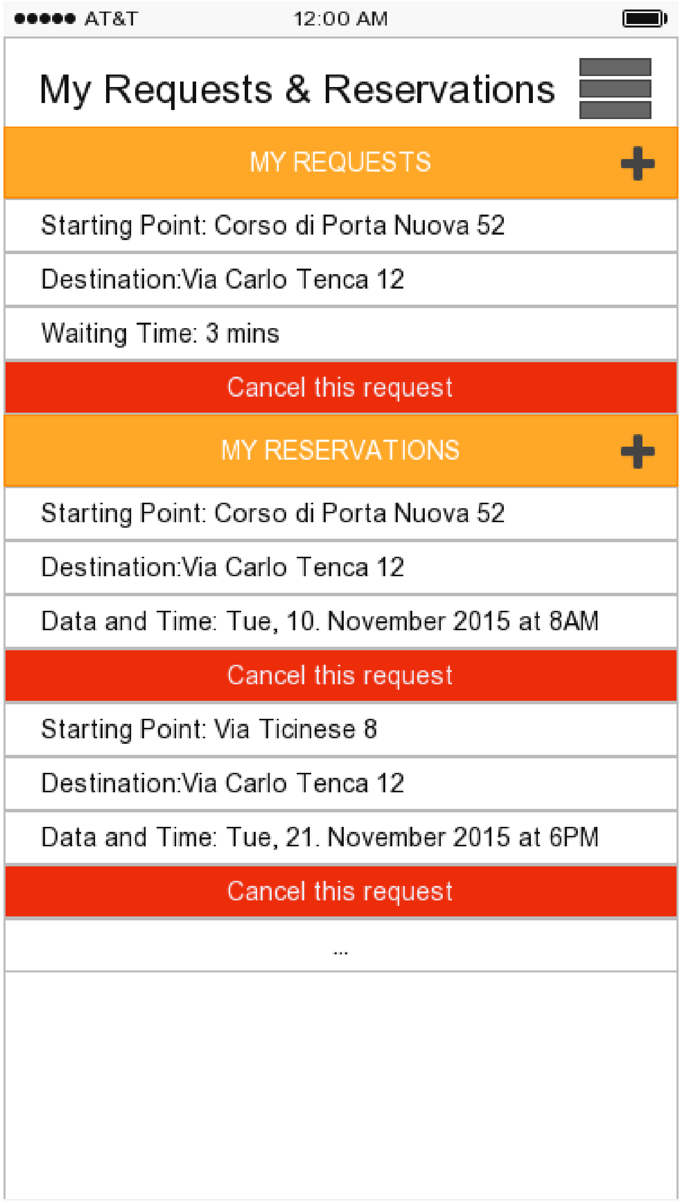
...

Visa: ***********4321

Master Card: ************3473

3.1.1.7 My Request and Reservations

This is the page where all the taxi requests and reservations of the user are displayed. From this page user can cancel his request or reservations, or generate a new request or reservation.



Starting Point: Corso di Porta Nuova 52

Destination:Via Carlo Tenca 12

Waiting Time: 3 mins

Starting Point: Corso di Porta Nuova 52

Destination:Via Carlo Tenca 12

Data and Time: Tue, 10. November 2015 at 8AM

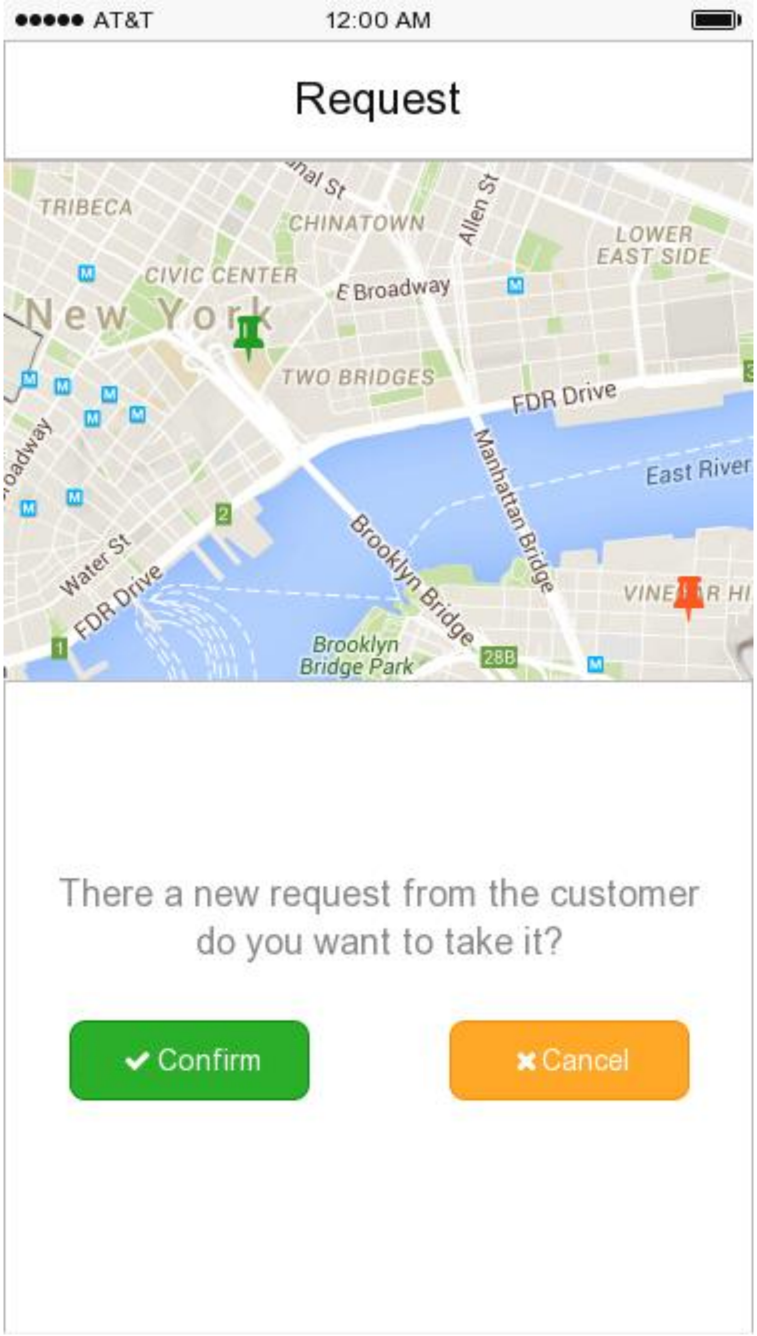
Starting Point: Via Ticinese 8

Destination:Via Carlo Tenca 12

Data and Time: Tue, 21. November 2015 at 6PM

3.1.1.8 Drivers Request Screen

This is the drivers screen from where he can receive requests for the rides



There a new request from the customer do you want to take it?

3.1.2 API Interface

myTaxiService will use two APIs:

- **Google Maps API** - for the maps, address seletction, route visualization.

- **PayPal** - the application will support payment through PayPal, for taxi rides, and refund of customer payment in cases of cancelation.

- **Areapay API** - The service enables payment from a user's mobile device directly to an application provider.

3.1.3 Software Interfaces

- **Data Base** - The database that the system is going to use is postgreSQL, version 9.4
- **Python** - For the backend, Python version 3.4 will be used.
- **Nginx** - For the web server
- **Operating System** - For the server ubuntu 14.04 will be used.
- **jQuery** - JavaScript library. Will be used for client side scripting
- **Java** - Will be used to develop the Android App
- **Objective C** - for the iOS app

3.1.4 Communications Interfaces

- **HTTPS** - 443
- **HTTP** - 80
- **DB** (For PosgreSQL) - 5432
- **FTP** (Data Transfer and Control) - 20, 21
- **SSH** (Remote Login) - 22

3.2 Functional Requirements

3.2.1 [G1] Allow visitors to register to the application.

[R1] Visitor has to fill and submit the registration form (Name, Email, Password and Phone number).
[R2] The Name, Email and Phone number should not match any other users data.
[D1] The email and phone number should be valid.

3.2.2 [G2] Allow registered users to log in to the application, with same credentials for both the web and the mobile application.

[R1] User has to type in his email/phone number and password in order to log in
[R2] User's credentials should match with the ones submited at the time of registration.
[R2] If the user wants to log in from his phone, he need to have myTaxiService application installed in his phone

[D1] System will implement the forget password functionality.
[D2] Wrong credentials will not grant access to the application

3.2.3 [G3] Passenger can request a taxi ride.

[R1] Passenger has to be logged in to the mobile or web application with his credentials.
[R2] Passenger has to specify his starting point and destination and submit a request.
[R3] Passenger can specify his starting point and destination by using the map, explicitly typing in the addresses, or using his GPS location (only for the starting point).

3.2.4 [G4] The system will notify the passenger with the taxi code, fare amount and waiting time, when a taxi driver accepts the system's request.

[R1] System has to receive the request from the passenger.
[R2] System has to forward the passenger's request to the very first taxi in the queue in zone where the passenger's starting point is.
[R3] Taxi driver has to accept the system's request.
[D1] All the taxis in the queue are free.

3.2.5 [G5] The passenger can either accept or cancel the taxi proposal.//has to pay the fare amount.

[R1]he has to receive the taxi proposal
[R2]if he accepts the taxi proposal he has to pay through the application
[R3]if he rejects the taxi proposal, the request will be canceled.
[D1]within 2 minutes he has to either accept or reject the taxi proposal
[D2]if he doesn't do anything within 2 minutes, that the request will be automatically canceled by the system .

3.2.6 [G6] Taxi is dispatched and will arrive at the requested location.

[R1]Taxi proposal needs to be accepted by the passenger
[R2]Taxi fare needs to be paid by the passenger
[R3]The system will allocate the requested taxi to the customer.
[D1]The passenger will be acknowledged by the system for his payment.
[D2] taxi will arrive within the specified waiting time

3.2.7 [G7] Passenger can reserve a taxi ride specifying his starting point, his destination and the desired time

[R1] Passenger has to be logged in to the mobile or web application with his credentials.
[R2] Passenger has to specify his starting point, destination, date and time and submit a request.

[R3] Passenger can specify his starting point and destination by using the map, explicitly typing in the addresses, or using his GPS location (only for the starting point).
[D] The reservation must occur at least 2 hours before the desired time, and at most 1 week before the desired time

3.2.8 [G8] When the passenger reserves a taxi ride, he will be informed with a code of the incoming taxi 10 minutes before the specified time of the ride.

[R1]The passenger will receive the remainder message from the system on his day of travel before two hours from his pickup time and he is asked to either accepts or cancel the reservation
[R2] if he accepts it, taxi will be allocated by the system.
[R3] if he rejects it, he will be refunded only 50% of his payment.

3.2.9 [G9] The passenger can cancel his request.

[R1]The passenger should have accepted the taxi proposal and paid the fare.
[R2]If he cancels the request, he will be refunded only 50% of his payment
[D1] If he doesn't show up in the starting point within 15 minutes from the pick up time, the system will cancel the request automatically, and the passenger will not be refunded any money.

3.2.10 [G10] The passenger can cancel his reservation.

[R1] The passenger should have accepted the taxi proposal for reservation and paid the fare.
[R2] If he cancels the reservation before 48h of the specified time, he will be refunded 70% of his payment
[R3]If he cancels the reservation before 1 hour of the specified time, he will be refunded 50% of his payment
[D1] If he doesn't show up in the starting point within 15 minutes from the pick up time, the system will cancel the reservation automatically, and the passenger will not be refunded any money

3.2.11 [G11] Available driver can confirm or reject the certain request for the ride.

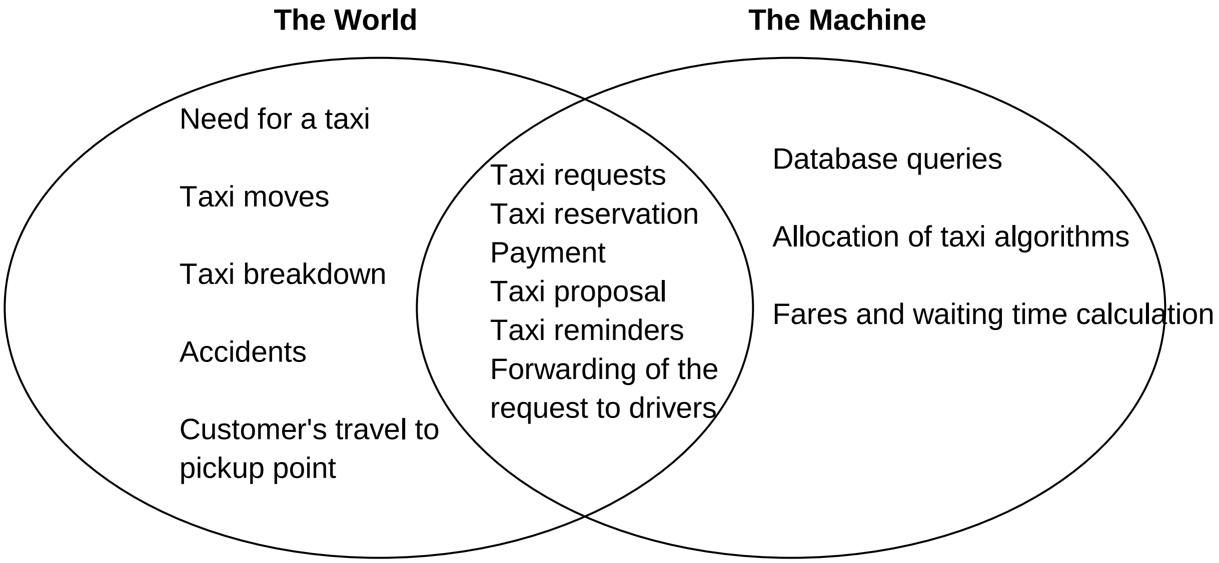
[R1] The request should have been forwarded to the taxi driver
[R2] If the driver rejects the request, he will be reallocated to the end of the queue
[R3] If the driver confirms the request, the taxi proposal will be put together and sent to the passenger
[D1] If the driver rejected the request, he has to conway his reasons to the HR department of the taxi service.

3.3 The World and The Machine

The world has the components that doesn’t have a interaction with a machine. For example, taxi breakdown and accidents, it happens due to external influence nothing to do with machine. The machine has the components that doesn’t have interaction with the world. For example, allocation of taxi algorithms, it is programmed and implemented in the system and fares and waiting time calculation, it calculates the fare depends on the distance between the starting point and destination specified in the request nothing to do with external world.

But, there are some components that needs an interaction between the World and the Machine. For example:

- Payment - user has to enter his credit card details and payment has to be done with the help of the system .
- Taxi Reminders - The system has to send the taxi reminders to the user and he/she has to accept or reject it.
- Forwarding the request to the drivers - The system has to forward the request to the driver and he has to accept or reject it.



3.4 Scenarios

3.4.1 Scenario 1

John an employer in need of taxi to go back to his home from office.He downloaded the mobile app myTaxiservice ,as a new user he has to register a new account so he clicked the Create New Account button and filled his personal details such as Name,email,mobile number,password in the registration form and click create. After created his own account,john logged in to his account using the registered credentials and clicked on request a taxi button and filled the request form with the details such as Starting point and destination and clicked the request button.Now the system received the request and forwaded it to the very first taxi in the queue in the zone and the driver accepts the request.After that,system send a taxi proposal consists of taxi code,fare and waiting time.John accepts the proposal within two minutes and the system asked him to pay his taxi fare.John clicked on the payment button entered all his credit card details such as card number,expiry date,CVV and secret code and click the button pay.The system acknowledeges his payment by sending a message and allocates the previously requested taxi to john.The taxi driver gets there in time to pickup John and he reached his home on time.

3.4.2 Scenario 2

George wants taxi to attend his friend bachelor's party,so he used myTaxi service web app since George is registered user ,he logged in to his account using his credentials and clicked on request a taxi button and filled in the request form with the details such as Starting point and destination and clicked the request button.Now the system received the request but unfortunately taxi queue in the zone is empty so the system forwaded it to the very first taxi in the queue in the next nearest zone.The taxi driver accepts the request and system send a taxi proposal consists of taxi code,fare and waiting time.George accepts the proposal within two minutes and the system asked him to pay his taxi fare.george clicked on the payment button entered all his credit card details such as card number,expiry date,CVV and secret code and click the button pay.The system acknowledeges his payment by sending a message and allocates the previously requested taxi to George.The taxi driver gets there in time to pickup George and he attend his bachelor's party happily.

3.4.3 Scenario 3

Mathew an exprimental physists in need of taxi to attend the seminar.He used the mobile app myTaxiservice,as a new user he has to register a new account so he clicked the Create New Account button and filled his personal details such as Name,email,mobile number,password in the registration form and also he entered his additional details such as home address ,office address and click create.After created his own account,Mathew logged in to his account using the registered credentials and clicked on request a taxi button and filled the request form by choosing home address as Starting point and mention the destination using GPS and clicked the request button.Now the system received the request and forwaded it to the very first taxi in the queue in the zone and the driver denied the request so the system forwaded the request to next taxi in the queue and the taxi driver accepts it,in the mean time the taxi driver who denied the system request has to convey the reason to supervisor.After that,system send a taxi proposal consists of taxi code,fare and waiting time.Mathew accepts the proposal within two minutes and the system asked him to pay his taxi fare.Mathew clicked on the payment button entered all his credit card details such as card number,expiry date,CVV and secret code and click the button pay.The system acknowledeges his payment by sending a message and allocates the previously requested taxi to Mathew.The taxi driver gets there in time to pickup Mathew and he reached the place to attend the seminar on time.

3.4.4 Scenario 4

Liza an engineer wants taxi to go to her office,so she used myTaxi service web app since Liza is a registered user ,she logged in to her account using her credentials and clicked on request a taxi button and filled in the request form with the details such as Starting point and destination and clicked the request button.Now the system received the request but unfortunately taxi queue in the zone is empty so the system forwaded it to the very first taxi in the queue in the next nearest zone.The taxi driver accepts the request and system send a taxi proposal consists of taxi code,fare and waiting time.Liza doesnot satisfy with the taxi proposal so she denied the proposal.The system asked her feedback about why she denied the taxi proposal.She gives her feedback as waiting time is too long.The system stores her feedback for future improvements.

3.4.5 Scenario 5

Franklin wants taxi to take his wife to beach.He used the mobile app myTaxiservice,as a new user he has to register a new account so he clicked the Create New Account button and filled his personal details such as Name,email,mobile number,password in the registration form and click create.After created his own account,Franklin logged in to his account using the registered credentials and clicked on request a taxi button and filled the request form by choosing home address as Starting point and mention the destination using GPS and clicked the request button.Now the system received the request and forwaded it to the very first taxi in the queue in the zone and the taxi driver accepts it,.After that,system send a taxi proposal consists of taxi code,fare and waiting time.Franklin accepts the proposal within two minutes and the system asked him to pay his taxi fare.Franklin clicked on the payment button entered all his credit card details such as card number,expiry date,CVV and secret code and click the button pay.The system acknowledeges his payment by sending a message and allocates the previously requested taxi to Franklin.Before the taxi arrives to pickup Franklin,he cancelled his request by clicking on the Cancel Request button in the app.The system acknowledges his cancellation ans sends a message that only 50% of the amount will be refunded.The system refunded the amount to his credit card with in two hours from cancellation.

3.4.6 Scenario 6

Harry the manager of private company is going to US for some business meeting so he wants to reserve a taxi to taxi to take him from his home to airport.He used myTaxi service web app since he is a registered user he logged in to her account with his credentials and fills the booking form by mentioning Pickup point,Time,Date and destination and sends a request by clicking on the reserve button.The system received his request and calculates the time remaining to his pickup time to put him in the priority queue then the system sends a taxi proposal consist of fare amount and his position in the queue .Harry accepts the taxi proposal then the system asked him to pay his fare.Harry click on the make payment button entered his credit card details such as card no,expiry date,CVV and secret code and click on the button pay.The system sends an acknoweldgement message stating that your payment is received.On the date of travel before two hours from the pickup time system sends the remainder to harry about his taxi reservation and he is asked to accept or cancel it within one hour.Once he accepted the system allocates the taxi to harry and sends taxicode,expected arrival time to harry.The taxi gets in time to the pickup point and drop him in airport on time to catch his Flight to US.

3.4.7 Scenario 7

Zara a house wife wants to take his children to her parents home on vacation so she wants to reserve a taxi to taxi to take her from her home to her parents home.She used myTaxi service mobile app as a new user she has to register a new account so she clicked the Create New Account button and filled his personal details such as Name,email,mobile number,password in the registration form and click create.After created her own account ,she logged in to her account with her credentials and fills the booking form by mentioning Pickup point and destination using GPS,fills the Time,Date and sends a request by clicking on the reserve button.The system received her request and calculates the time remaining to her pickup time to put her in the priority queue then the system sends a taxi proposal consist of fare amount and her position in the queue.Zara accepts the taxi proposal then the system asked her to pay her fare.Zara click on the make payment button entered her credit card details such as card no,expiry date,CVV and secret code and click on the button pay.The system sends an acknoweldgement message stating that your payment is received.On the date of travel before two hours from the pickup time system sends the remainder to zara about his taxi reservation and she is asked to accept or cancel it within one hour.But she denies the request because one of her children got sick.The system acknowledges her cancellation and sends her a message only 50% percent of her payment will be refunded.The system refunded the amount to her credit card with in two hours from cancellation.

3.4.8 Scenario 8

Harry the manager of private company is returning from US after attended some business meeting so he wants to reserve a taxi to taxi to take him from airport to home.He used myTaxi service web app since he is a registered user he logged in to her account with his credentials and fills the booking form by mentioning Pickup point,Time,Date and destination and sends a request by clicking on the reserve button.The system received his request and calculates the time remaining to his pickup time to put him in the priority queue then the system sends a taxi proposal consist of fare amount and his position in the queue .Harry accepts the taxi proposal then the system asked him to pay his fare.He clicks on the make payment button since Harry stored his credit card details he clicks on the credit card name and click on the button pay.The system sends an acknoweldgement message stating that your payment is received.But before 48 hours from the pickup time harry's flight is cancelled due to some weather problem so he cancelled his taxi reservation by clicking cancel reserved button.The system acknowledges his cancellation and sends him a message only 70% of his payment will be refunded.The system refunded the amount to his credit card with in two hours from cancellation.

3.4.9 Scenario 9

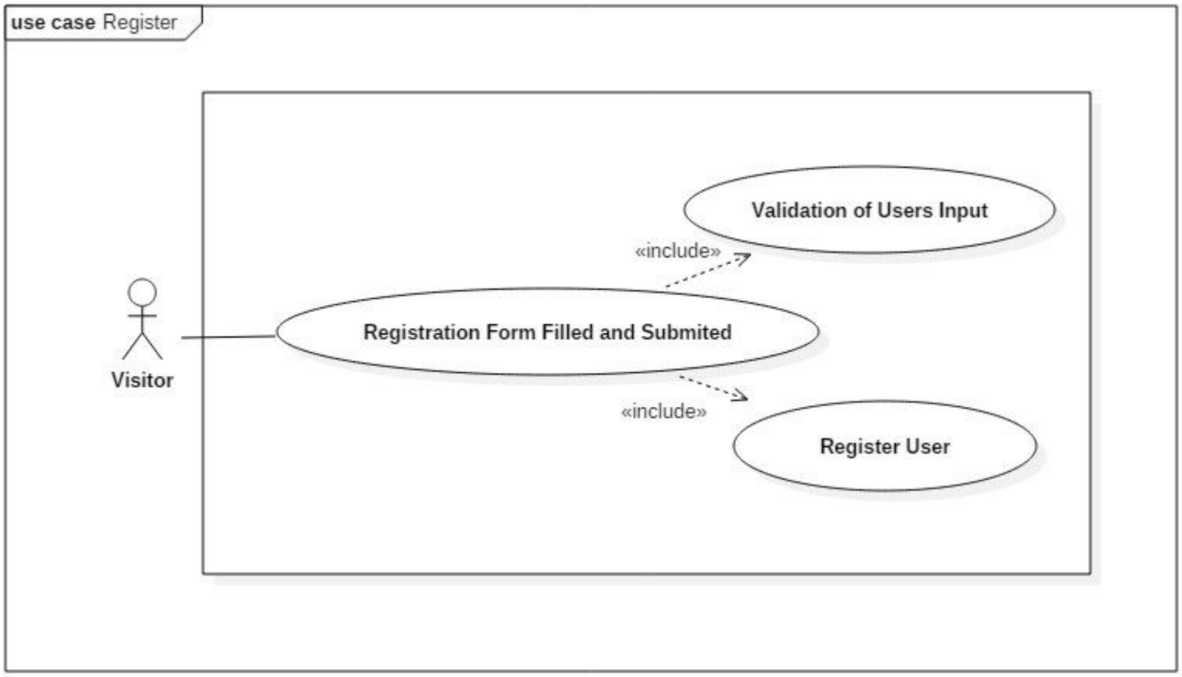
Zara a house wife wants to take his children to her parents home on vacation so she wants to reserve a taxi to taxi to take her from her home to her parents home.She used myTaxi service mobile app as a new user she has to register a new account so she clicked the Create New Account button and filled his personal details such as Name,email,mobile number,password in the registration form and click create.After created her own account ,she logged in to her account with her credentials and fills the booking form by mentioning Pickup point and destination using GPS,fills the Time,Date and sends a request by clicking on the reserve button.The system received her request and calculates the time remaining to her pickup time to put her in the priority queue then the system sends a taxi proposal consist of fare amount and her position in the queue.Zara denies the taxi proposal so the system asked her to give optional feedback.Zara gives her feedback as postion in the queue is too high.The system stores her feedback for futures improvements.

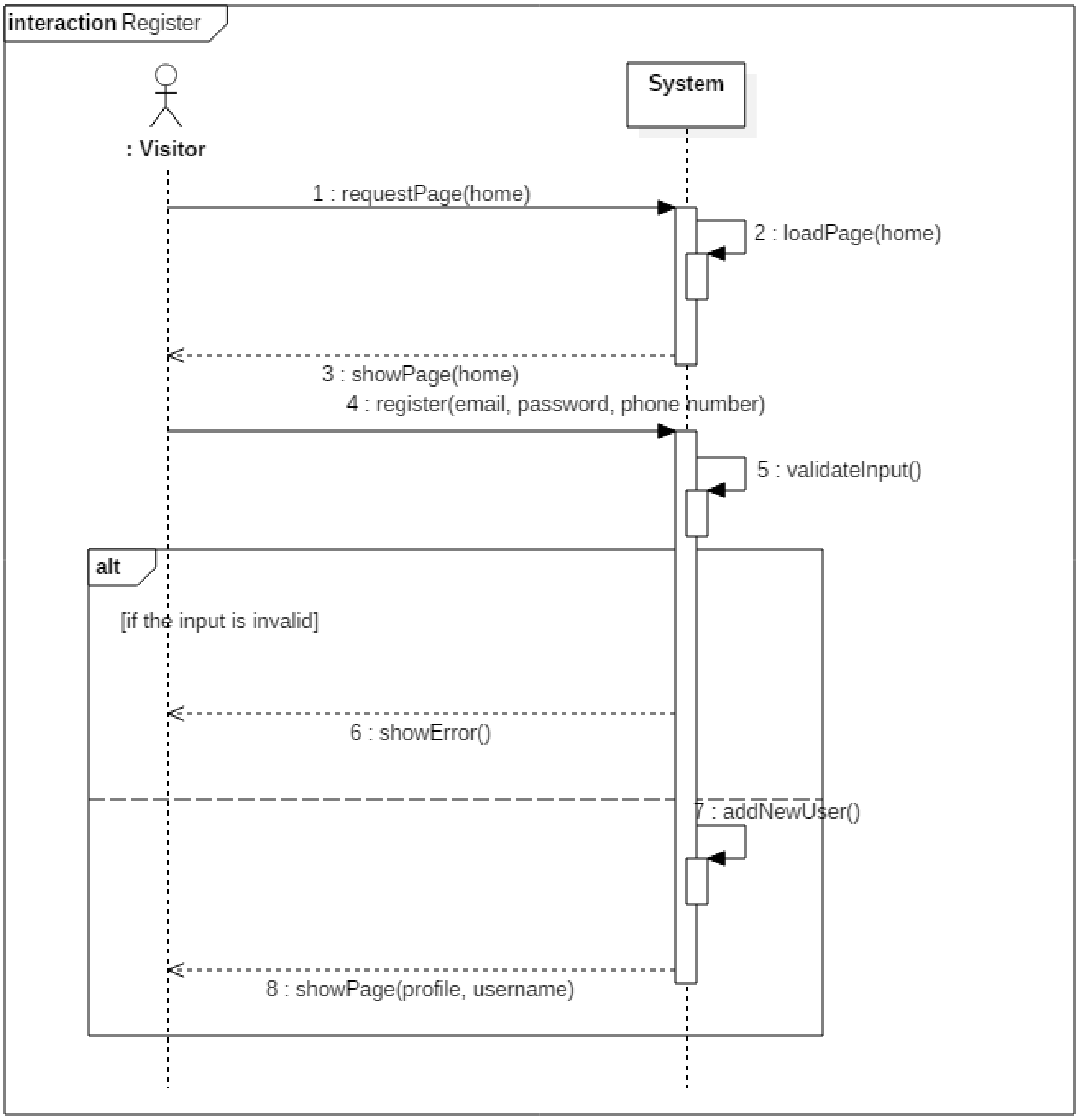
3.5 UML MODELS

3.5.1 USE CASES

3.5.1.1Vistors register to the application

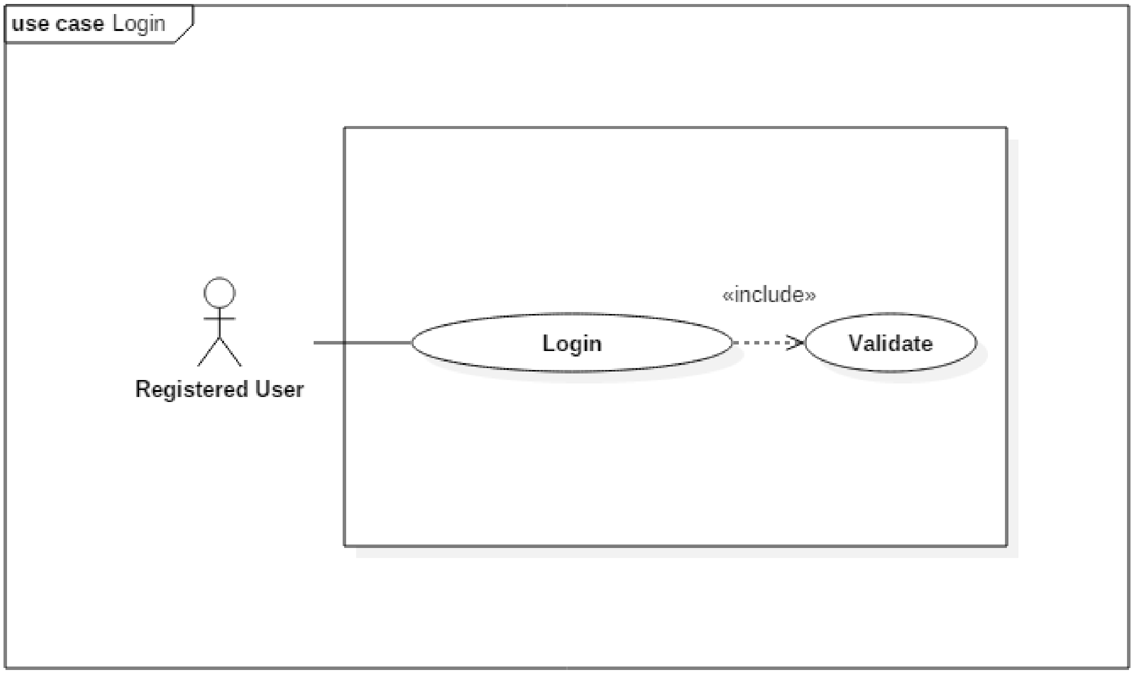
Actor	Visitors
Goal	[G1]
Input condition	Visitor should be on the webpage of the application or have the mobile application installed
Event flow	1.Visitors should click on the sign up button. 2.Visitors fills in all the mandatory fields in the registration form. 3.Visitors then click on the register button to submit. 4.Then system will save the account data in to the database.
Output conditions	Visitors becomes registered users and automatically logged in and transferred to his profile page.
Exceptions	1.Visitors already registered. 2.Failed to fill all the mandatory fields. 3. Invalid fields. 4.Email chosen is already associated with another user. 5.Phone number chosen is already associated with another user

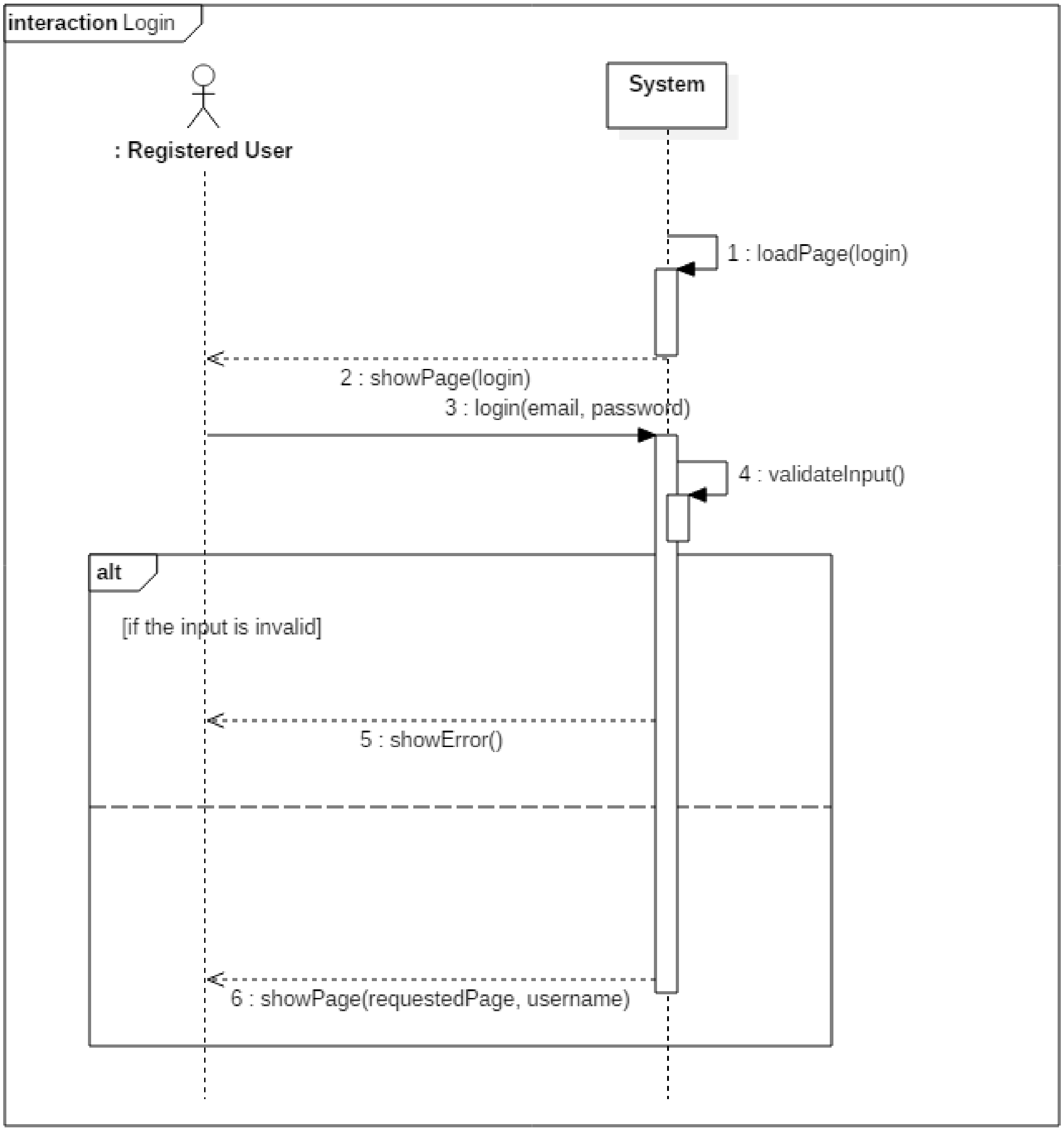

use case Register
Validation of Users Input
«include»
Visitor
Registration Form Filled and Submited
«include»
Register User


interaction Register
: Visitor
System
1 : requestPage(home)
2 : loadPage(home)
3 : showPage(home)
4 : register(email, password, phone number)
5 : validateInput()
alt
[if the input is invalid]
6 : showError()
7 : addNewUser()
8 : showPage(profile, username)

3.5.1.2 Login for registered users

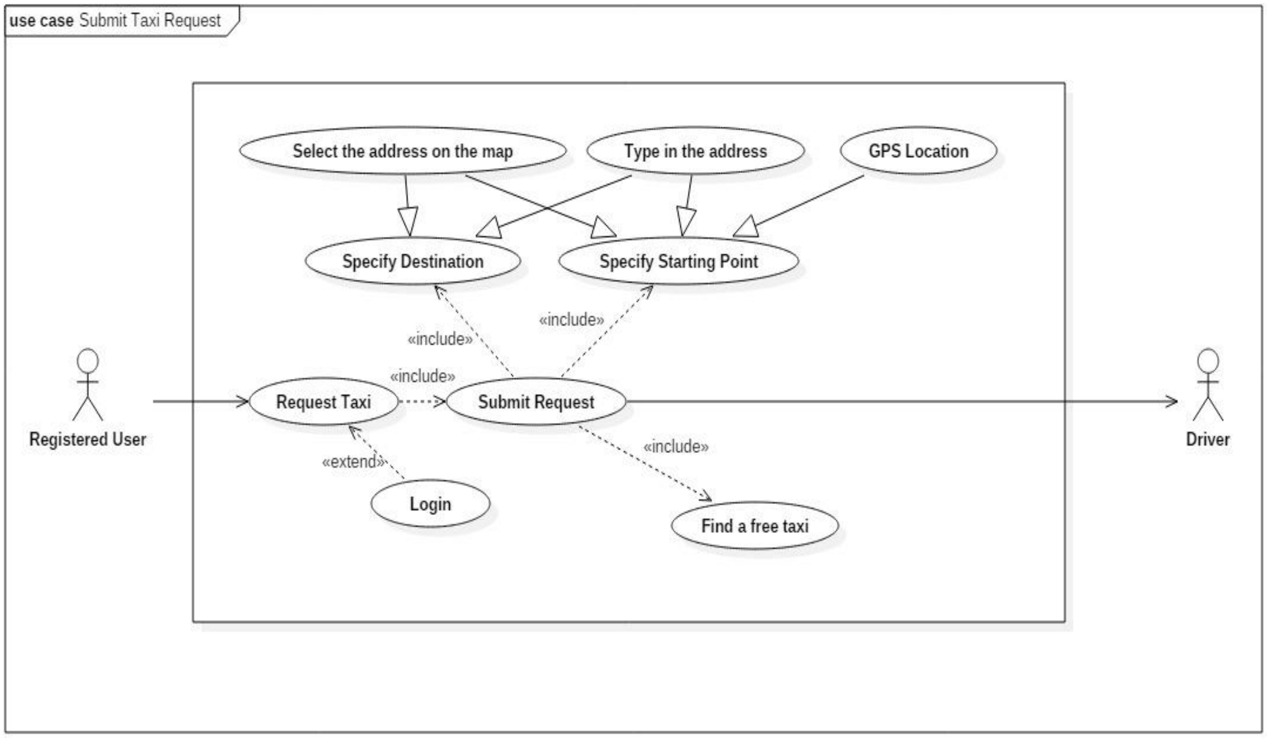
Actor	Registered users
Goal	[G2]
Input condition	Null
Event flow	1.User has to click the login button. 2.User has to enter his email or phone Number and his password. 3.User has to click login to submit his credentials 4.The system will verifies his credentials with database records.
Output conditions	User will be logged in and transferred to the main page. .
Exceptions	1.Failed to fill all the mandatory fields . 2. Invalid fields. 3.User credentials doesn’t match with database records.

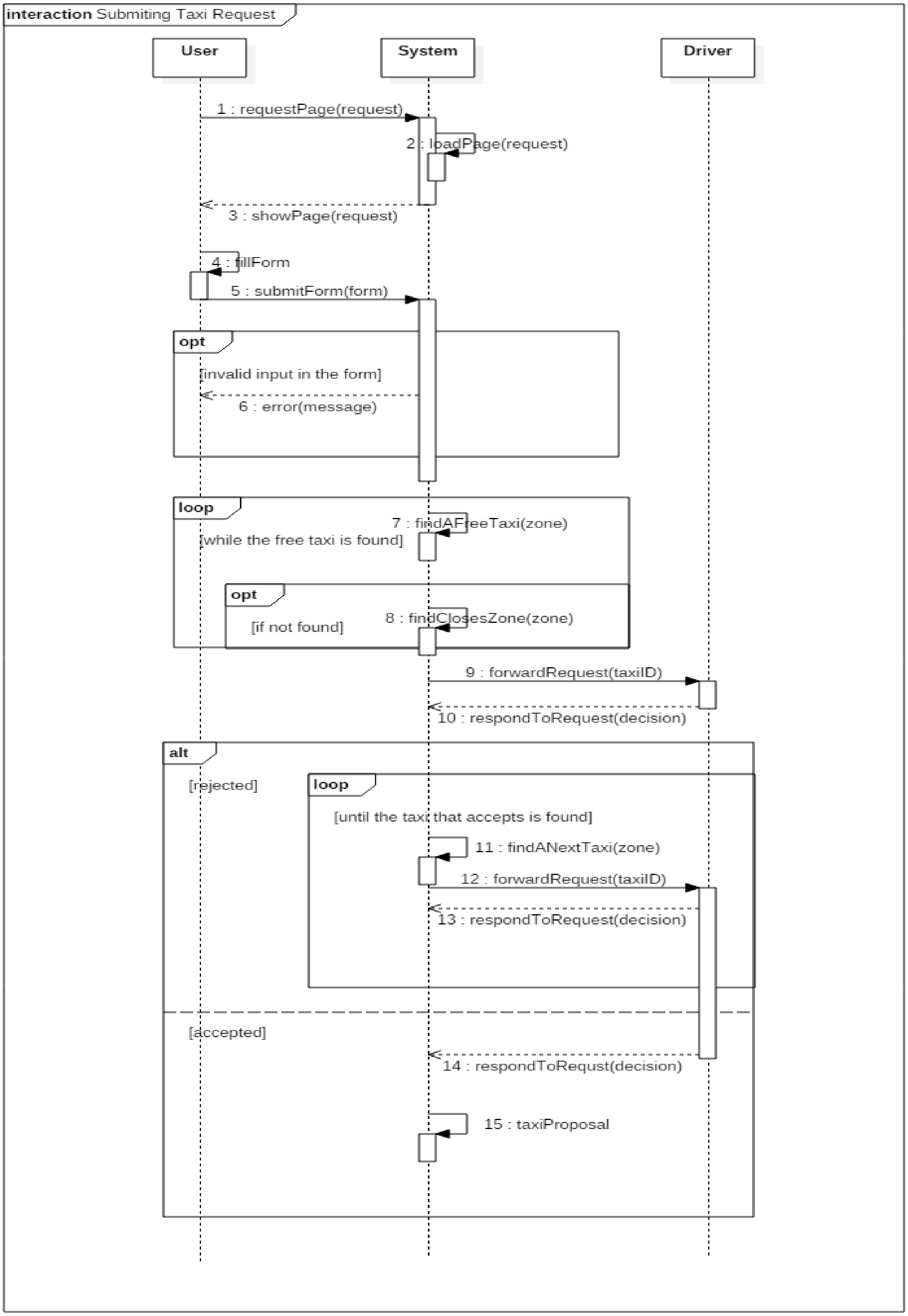

use case Login
Registered User
Login
«include»
Validate


interaction Login
: Registered User
System
1 : loadPage(login)
2 : showPage(login)
3 : login(email, password)
4 : validateInput()
alt
[if the input is invalid]
5 : showError()
6 : showPage(requestedPage, username)

3.5.1.3 Submitting a Taxi request

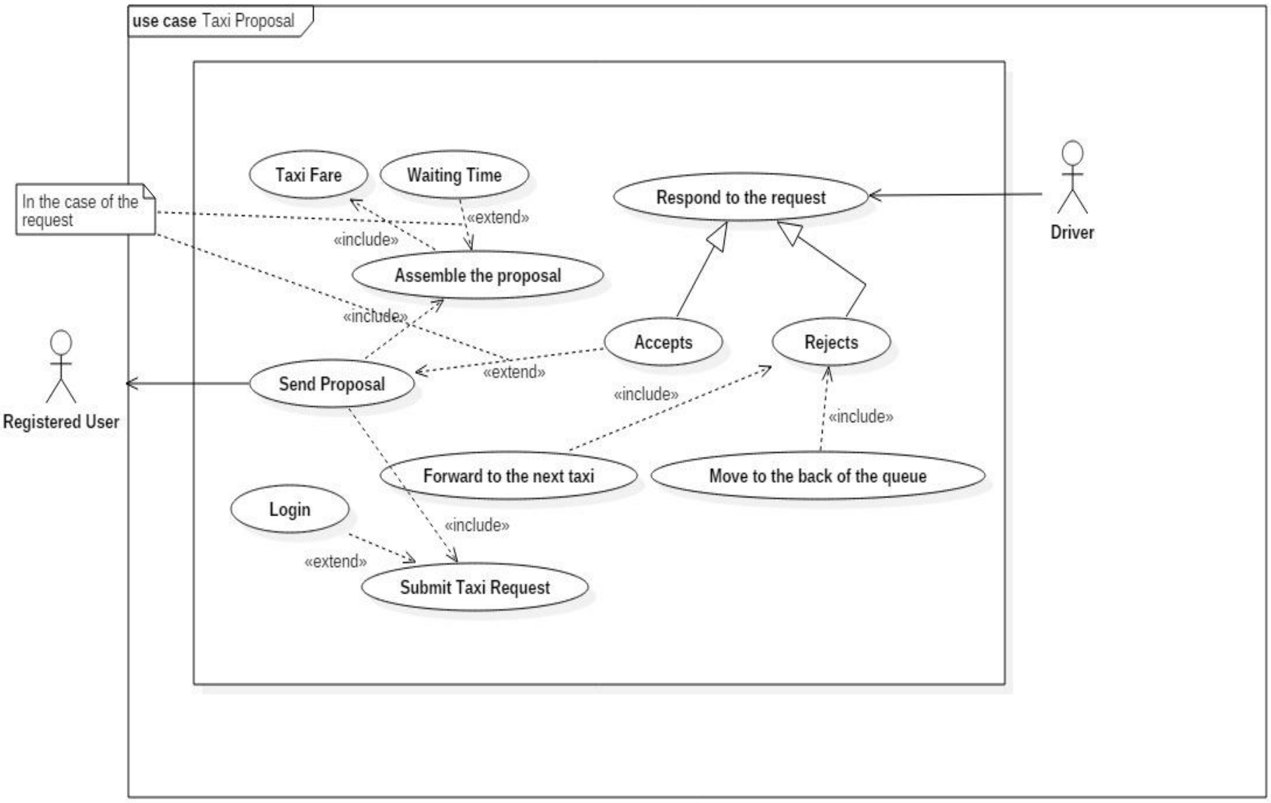
Actor	Registered users, Driver
Goal	[G3]
Input condition	User must be logged in
Event flow	1.User has to click the request a taxi button. 2.User has to specify his starting point and destination. 3.User has to click request buttton to submit his request. .
Output conditions	1.The system will forward this request to the first taxi driver in the queue in that zone if the queue is empty the request will be forwarded to the first driver in the next nearest zone. .
Exceptions	1.Failed to fill all the mandatory fields . 2. Invalid fields.(Address). 3.If the starting point or destination of the user not with in any of the zones.

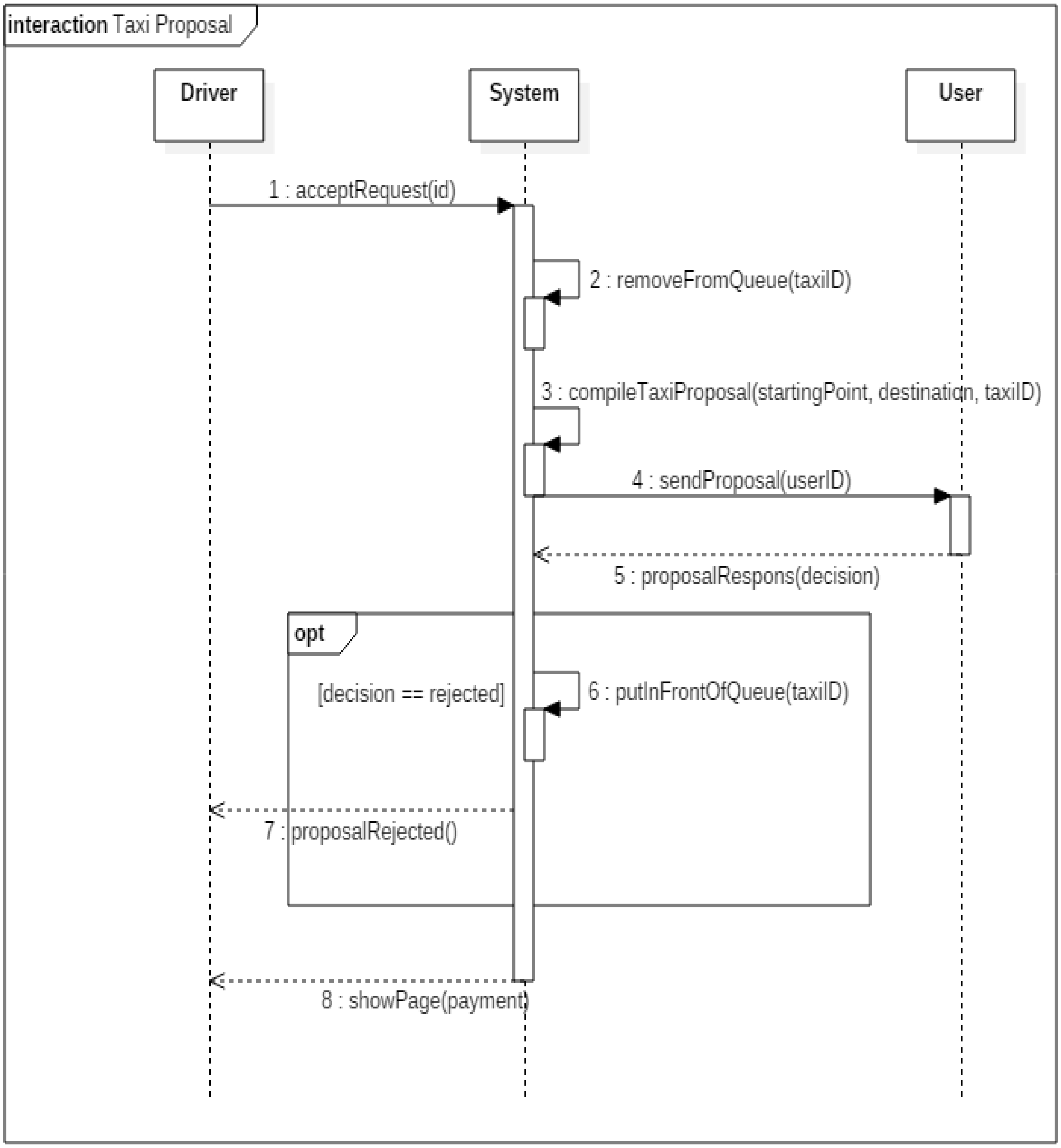

use case Submit Taxi Request
Select the address on the map
Type in the address
GPS Location
Specify Destination
Specify Starting Point
«include»
«include»
«include»
Registered User
Request Taxi
Submit Request
Driver
«extend»
«include»
Login
Find a free taxi


interaction Submiting Taxi Request
User
System
Driver
1 : requestPage(request)
2 : loadPage(request)
3 : showPage(request)
4 : fillForm
5 : submitForm(form)
opt
[invalid input in the form]
6 : error(message)
loop
7 : findAFreeTaxi(zone)
[while the free taxi is found]
opt
[if not found]
8 : findClosesZone(zone)
9 : forwardRequest(taxiID)
10 : respondToRequest(decision)
alt
[rejected]
loop
[until the taxi that accepts is found]
11 : findANextTaxi(zone)
12 : forwardRequest(taxiID)
13 : respondToRequest(decision)
[accepted]
14 : respondToRequst(decision)
15 : taxiProposal

3.5.1.4 Taxi proposal

Actor	Registered users, Driver
Goal	[G4]
Input condition	User must be logged in. User must have submitted the request or reservation. The driver has to accept the forwarded request. (only in case of the request)
Event flow	1.System calculates the Fare amount based on the distance between starting point and destination. 2.The system has to calculate the approximate waiting time based on the starting point and drivers locations (only in case of the request) 3.The system will generate the Taxi proposal by putting together Taxi code ,waiting time (only in case of the request) and fare amount and forwarded it to the user.
Output conditions	User has to receive the taxi proposal..
Exceptions	Null

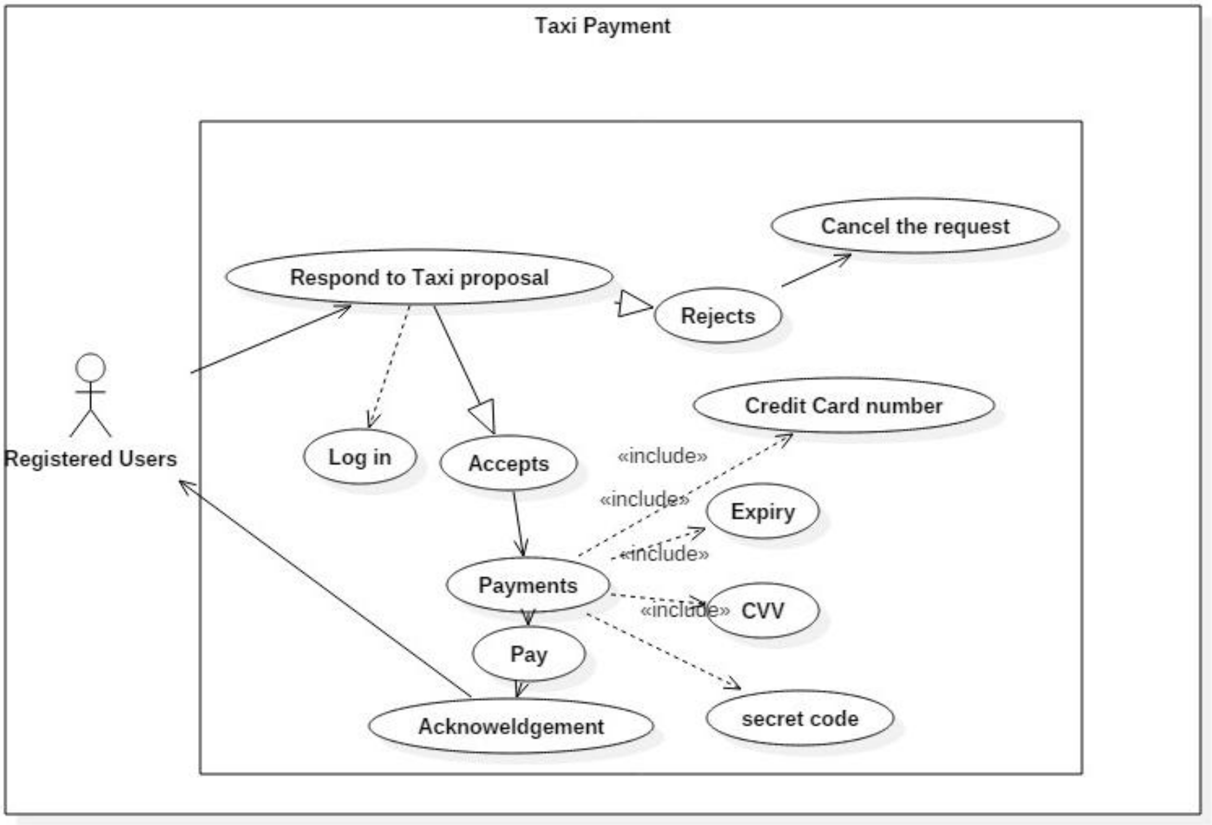

use case Taxi Proposal
In the case of the request
Taxi Fare
Waiting Time
Respond to the request
Driver
«extend»
«include»
Assemble the proposal
«include»
Accepts
Rejects
Registered User
Send Proposal
«extend»
«include»
«include»
Forward to the next taxi
Move to the back of the queue
Login
«include»
«extend»
Submit Taxi Request

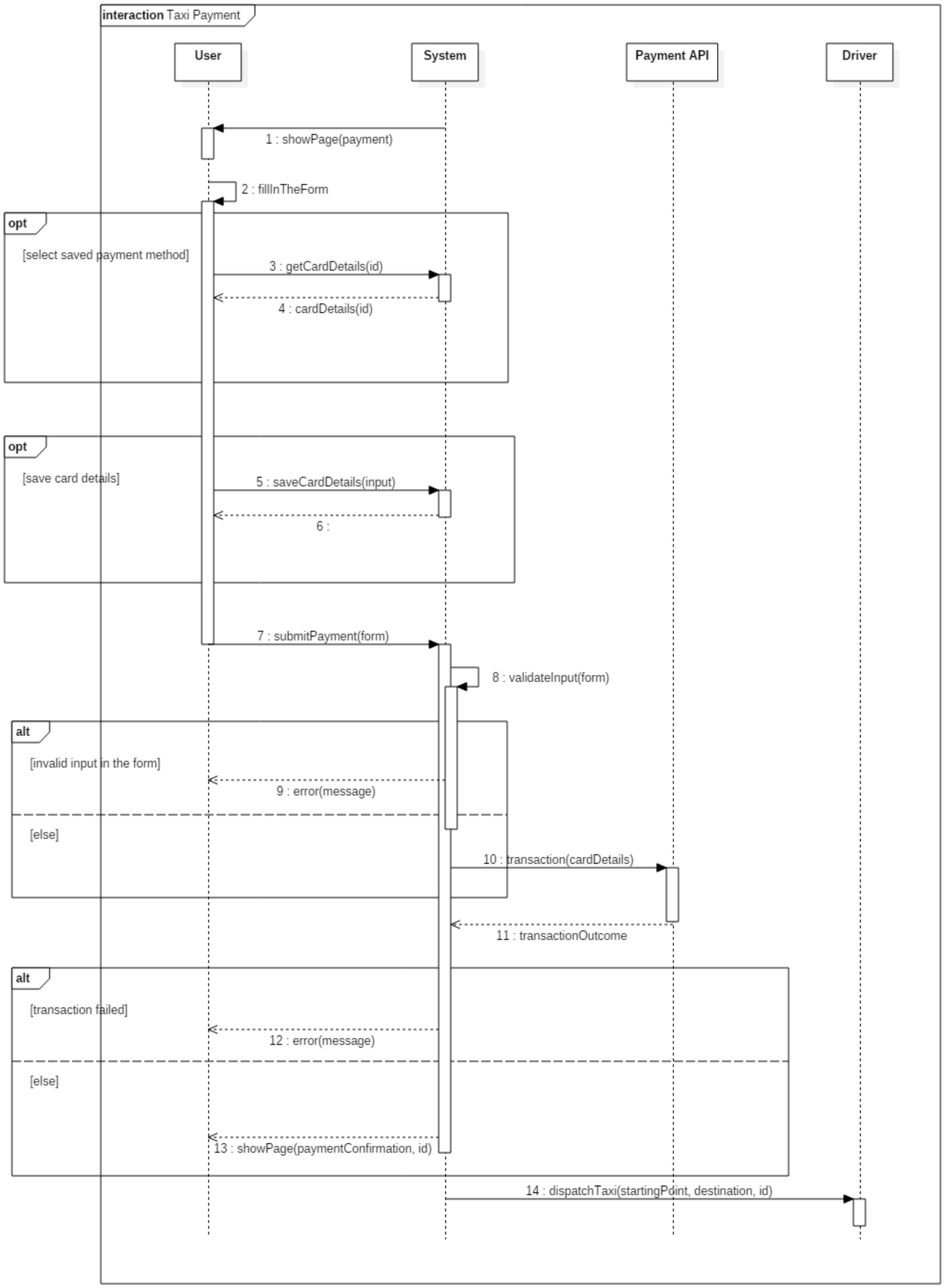

interaction Taxi Proposal
Driver
System
User
1 : acceptRequest(id)
2 : removeFromQueue(taxiID)
3 : compileTaxiProposal(startingPoint, destination, taxiID)
4 : sendProposal(userID)
5 : proposalRespons(decision)
opt
[decision == rejected]
6 : putInFrontOfQueue(taxiID)
7 : proposalRejected()
8 : showPage(payment)

3.5.1.5 Taxi payment

Actor	Registered users
Goal	[G5]
Input condition	User must be logged in. User must have submitted the request or reservation. The driver has to accept the forwarded request. (only in case of the request) User must received the taxi proposal..
Event flow	1.If user accepts the taxi proposal ,then he has to be forwarded to the payment form. 2.User has to fill in his credit card details such as card number, expiry date, CVV, secret code and click pay button. In this form it is possible for user to save his card details by save button, or to choose his previously saved payment method. 3. Transaction is completed through an API for the payment 4. Taxi request or reservation will be saved to the database (Displayed on the My Requests & Reservation page)
Output conditions	Acknowledgement message will be send to the user
Exceptions	If user rejects the taxi proposal, the request will be canceled and the taxi will be reallocated to the first place in the queue. (in case of the request) If the user neither accepts nor cancel the

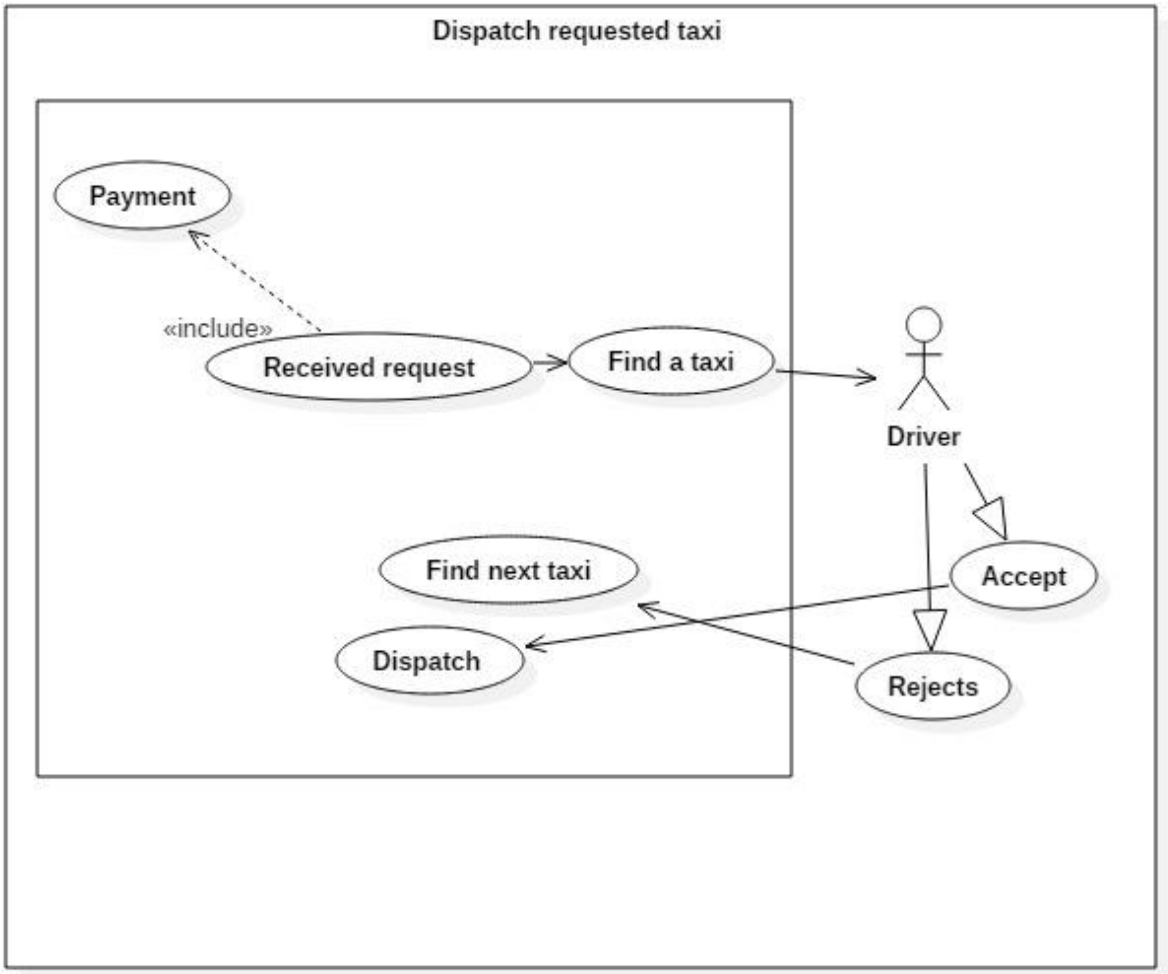
	proposal within two minutes, the request will be automatically canceled, and the taxi will be reallocated to the first place in the queue . If the transaction was unsuccessful

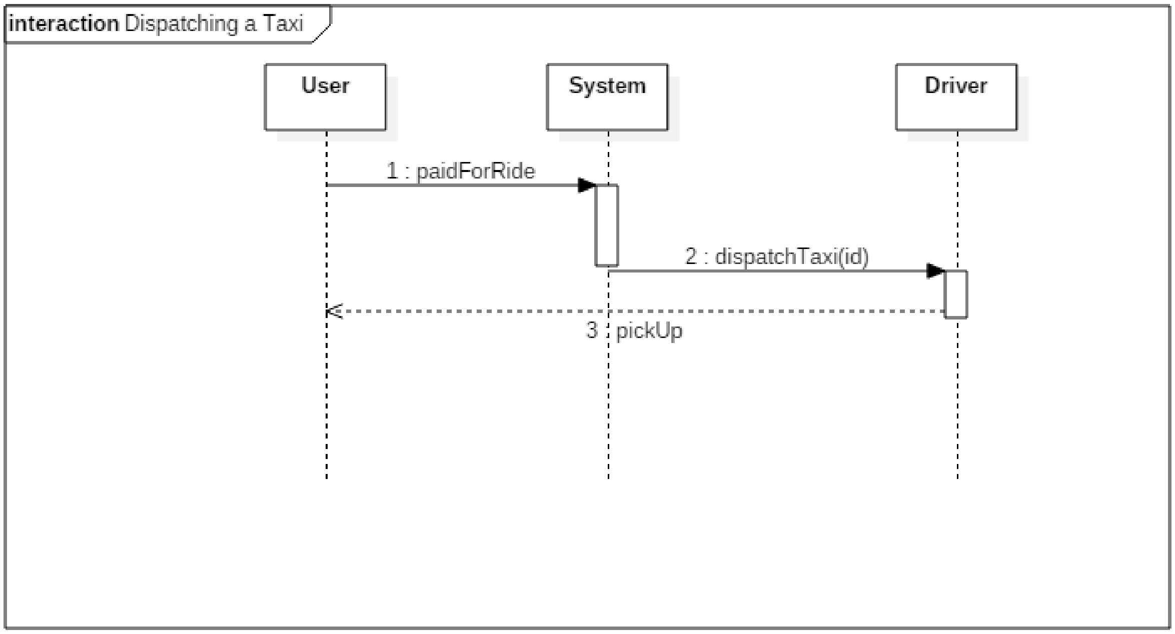




3.5.1.6 Dispatching the taxi

Actors	Registered users,
Goal	[G6]
Input condition	User must be logged in (delete this probably) Taxi proposal needs to be accepted by the passenger Taxi fare needs to be paid by the passenger
Event flow	The system will allocate the requested taxi to the customer. The taxi will be dispatched to the users pickup point
Output conditions	Taxi will arrive in specified starting point within the specified waiting time to pick up the passenger.
Exceptions	If the traffic is high. (taxi may be late) If the taxi gets breakdown or meets with accident. If the passenger does not show up at the pickup point within 15 minutes from the specified time, the request is automaticaly canceled and the passenger will not be refunded any money

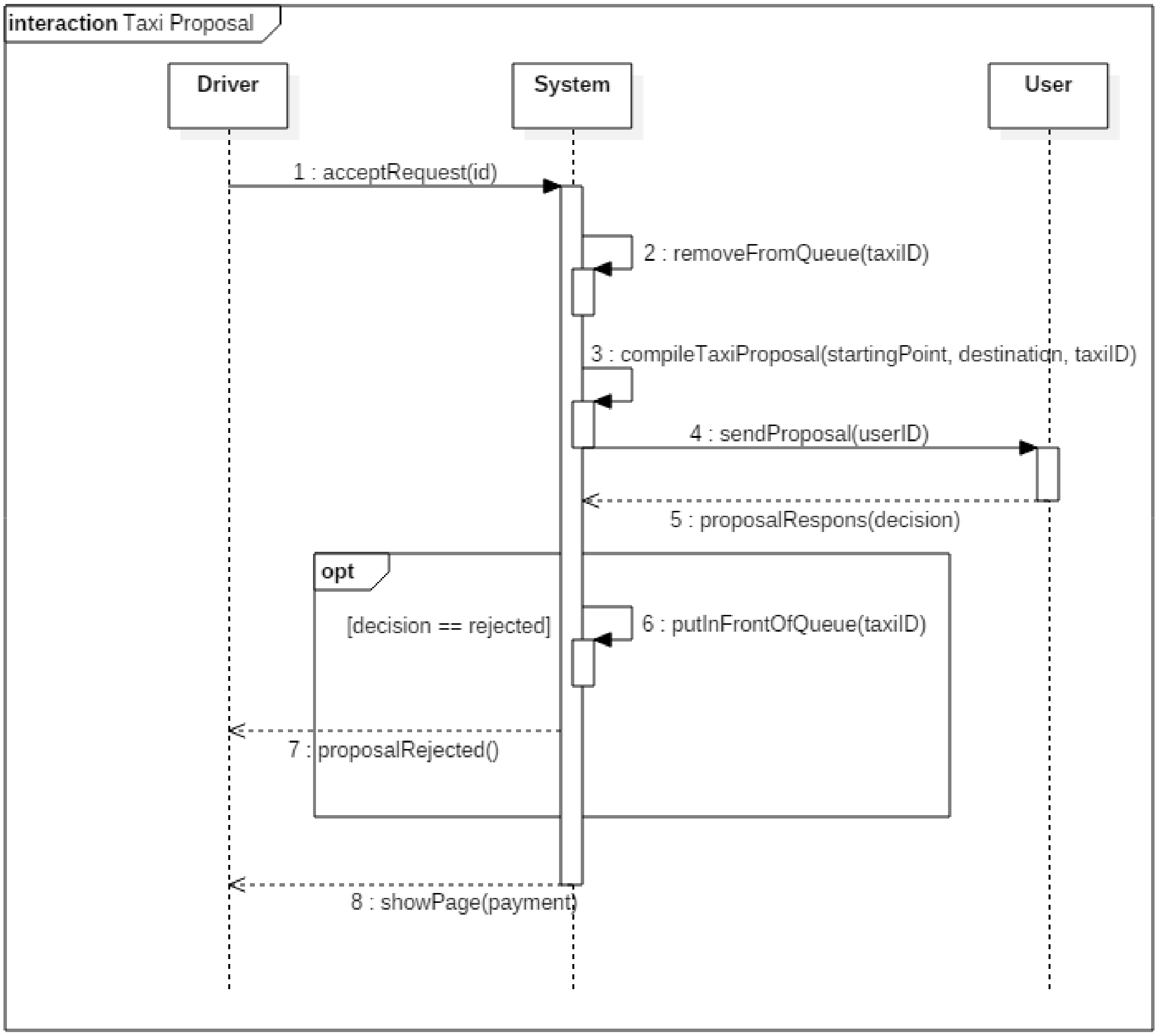

Dispatch requested taxi
Payment
«include»
Received request
Find a taxi
Driver
Find next taxi
Accept
Dispatch
Rejects



3.5.1.7 Taxi reservation

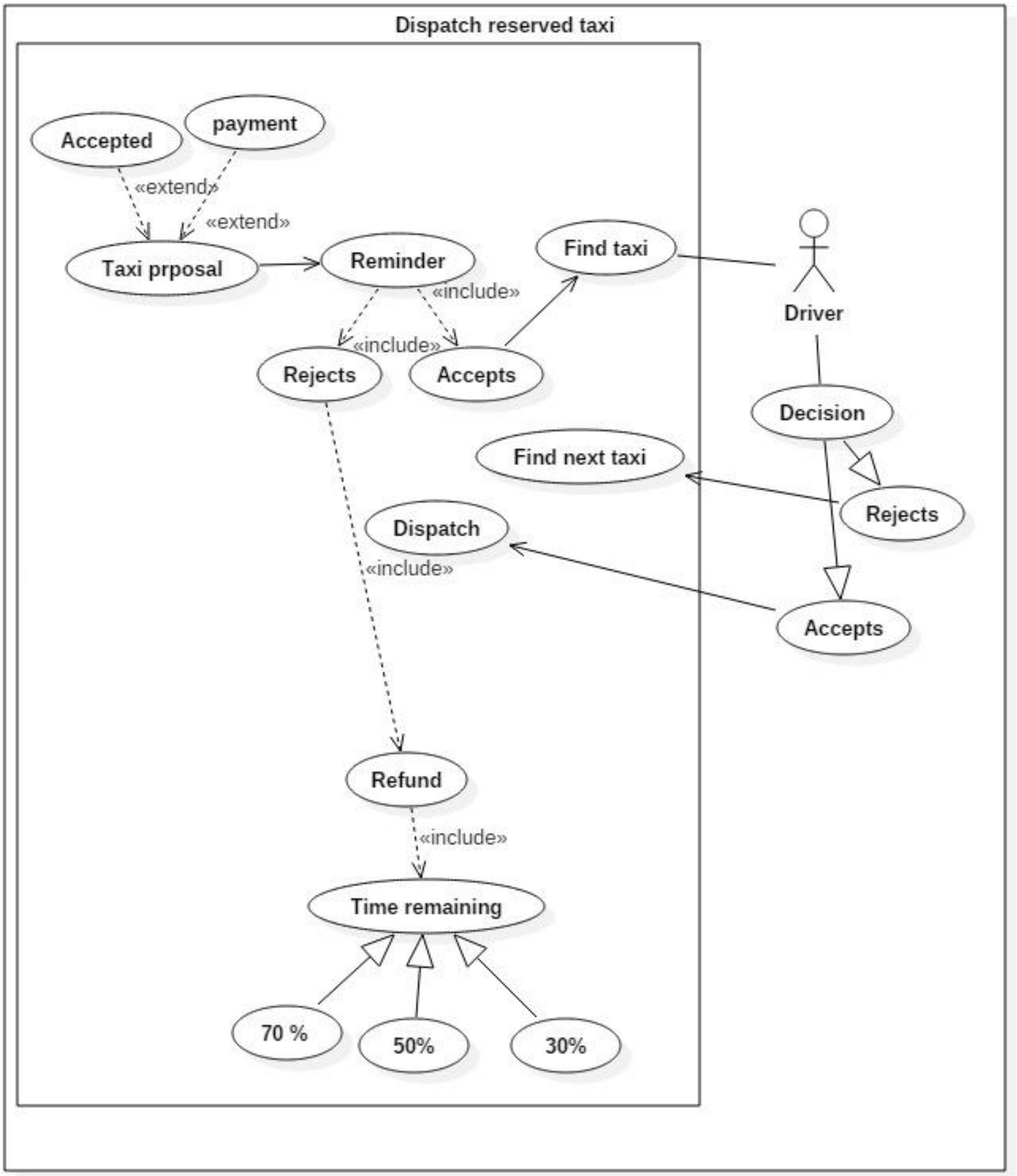
Actor	Registered users
Goal	[G7]
Input condition	User must be logged in.
Event flow	1.User has to click the reserve a taxi button. 2.User has to specify his starting point, date, time and destination. 3.User has to click reserve button to submit his request.
Output conditions	1.The system sends the taxi proposal to the user.

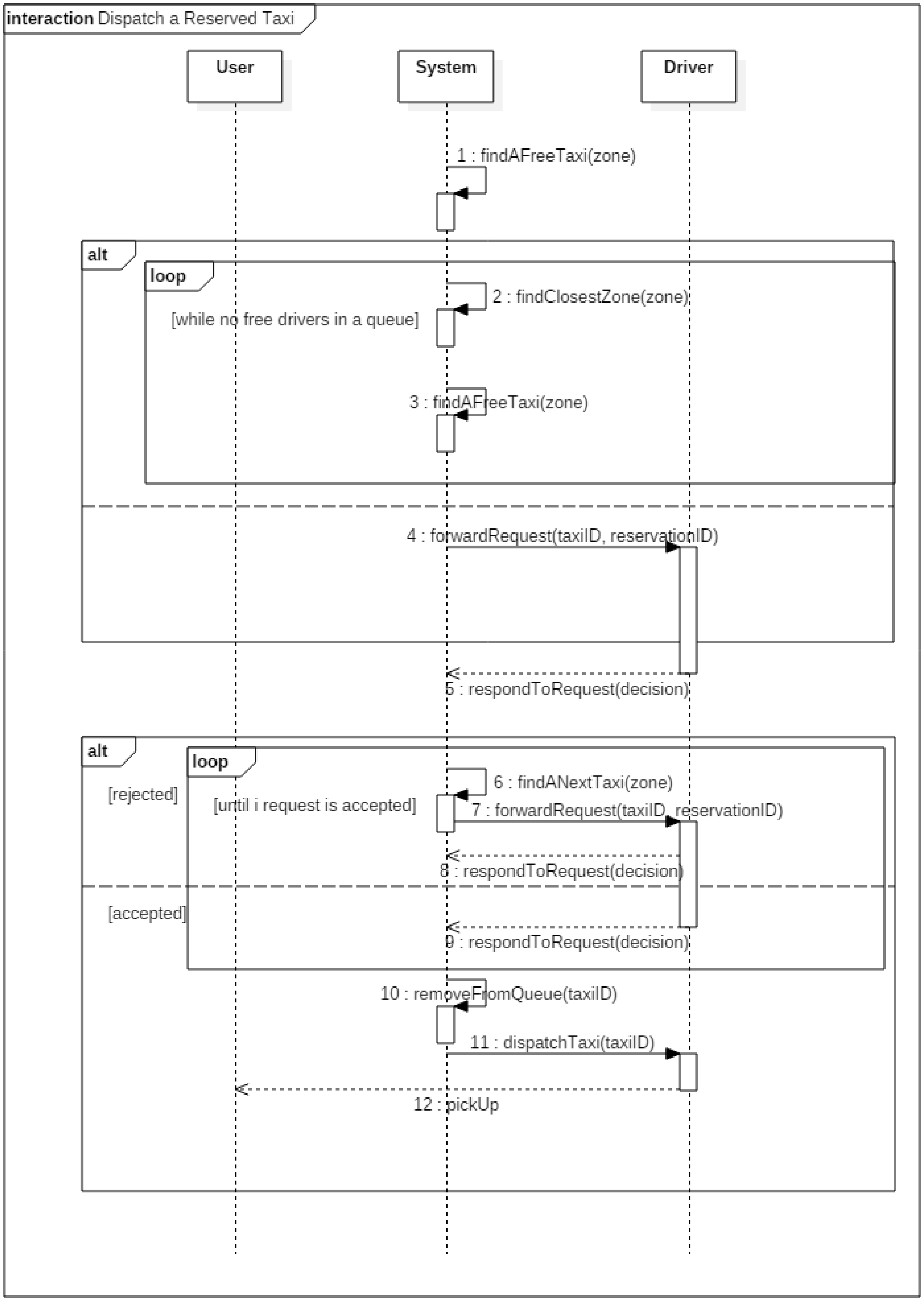
Exceptions	1.Failed to fill all the mandatory fields . 2. Invalid fields.(Address). 3.If the starting point of the user not with in any of the zones. 4. The reservation must occur at least 2 hours before the desired time, and at most 1


interaction Taxi Proposal
Driver
System
User
1 : acceptRequest(id)
2 : removeFromQueue(taxiID)
3 : compileTaxiProposal(startingPoint, destination, taxiID)
4 : sendProposal(userID)
5 : proposalRespons(decision)
opt
[decision == rejected]
6 : putInFrontOfQueue(taxiID)
7 : proposalRejected()
8 : showPage(payment)

3.5.1.8 Dispatching a reserved taxi

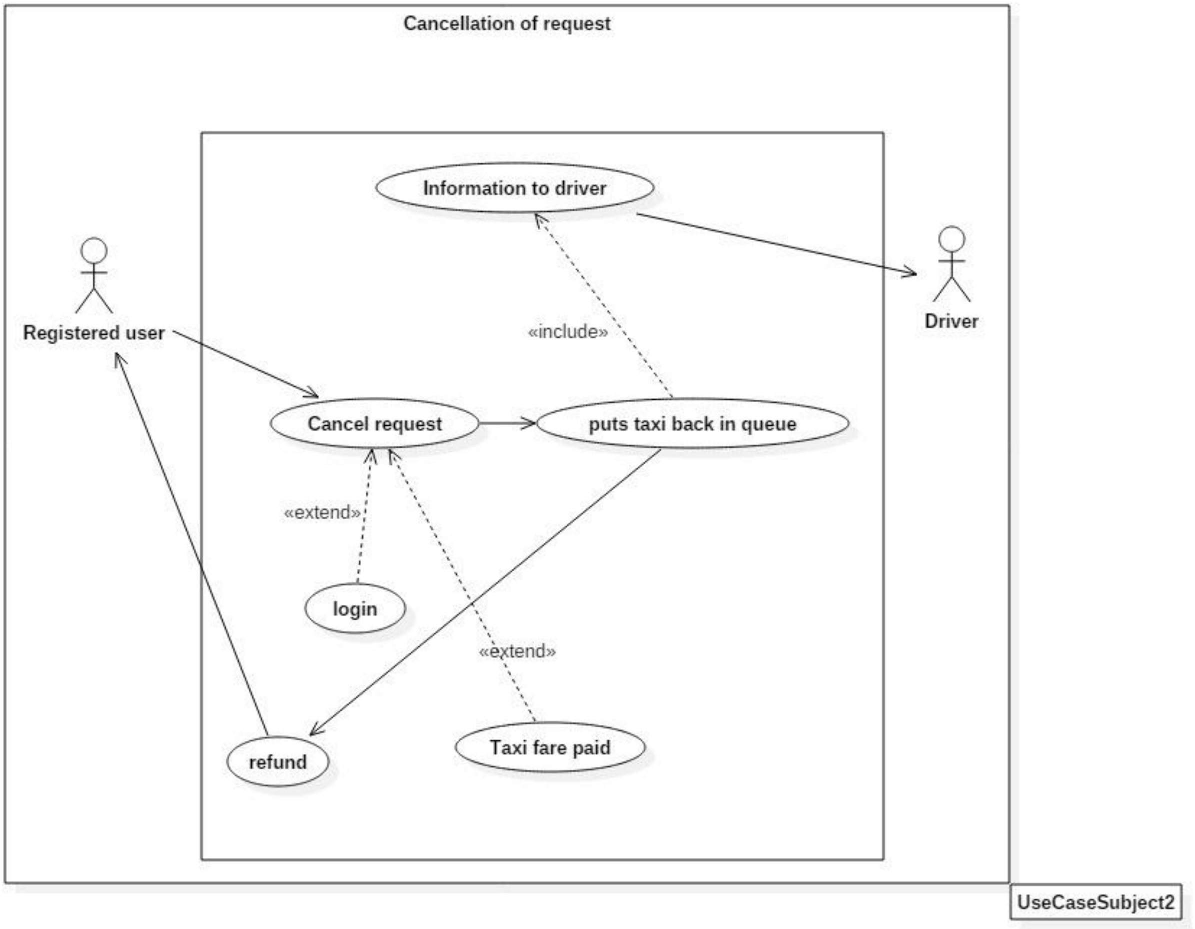
Actors	Registered users, Driver
Goal	[G9]
Input condition	User must be logged in Taxi proposal for reservation needs to be accepted by the passenger Taxi fare needs to be paid by the passenger Reservation must not be canceled previously
Event flow	1. The system will allocate the first taxi queuing in the zones queue to the customer 10 minutes before the specified time. If the queue is empty the request will be forwaded to the first driver in the next nearest zone. 2.The taxi will be dispatched to the users pickup point.
Output conditions	Taxi will arrive in specified starting point at the specified time.
Exceptions	If the traffic is high. If the taxi gets breakdown or meet with accident. If the passenger does not show up at the pickup point within 15 minutes from the specified time, the reservation is automatically canceled and the passenger will not be refunded any money

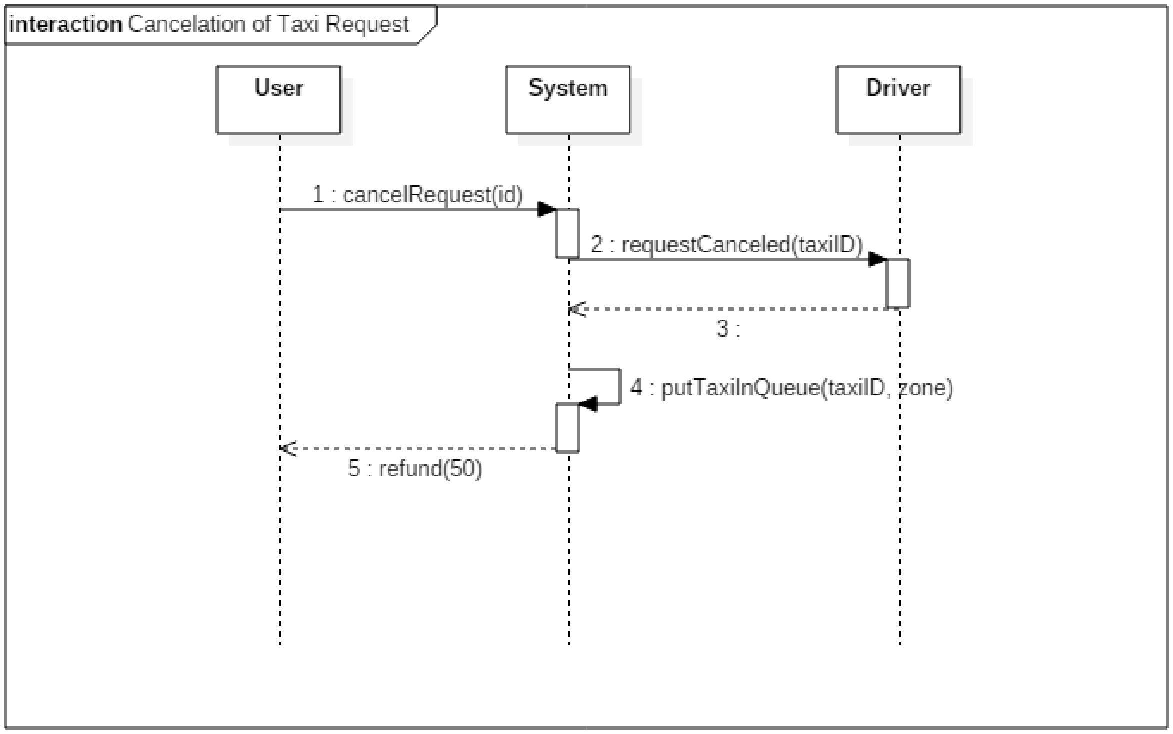

Dispatch reserved taxi
Accepted
payment
«extend»
«extend»
Taxi prposal
Reminder
«include»
«include»
Find taxi
Driver
Rejects
Accepts
Decision
Find next taxi
Rejects
Dispatch
«include»
Accepts
Refund
«include»
Time remaining
70 %
50%
30%


interaction Dispatch a Reserved Taxi
User
System
Driver
1 : findAFreeTaxi(zone)
alt
loop
[while no free drivers in a queue]
2 : findClosestZone(zone)
3 : findAFreeTaxi(zone)
4 : forwardRequest(taxiID, reservationID)
5 : respondToRequest(decision)
alt
loop
[rejected]
[until i request is accepted]
6 : findANextTaxi(zone)
7 : forwardRequest(taxiID, reservationID)
8 : respondToRequest(decision)
[accepted]
9 : respondToRequest(decision)
10 : removeFromQueue(taxiID)
11 : dispatchTaxi(taxiID)
12 : pickUp

3.5.1.9 *Cancelation of taxi request*

Actors	Registered users, Driver
Goal	[G10]
Input condition	User must be logged in Taxi proposal for request needs to be accepted by the passenger Taxi fare needs to be paid by the passenger
Event flow	1. User clicks on the cancel request button in his requests page. 2. System notifies the dispatched taxi driver that the request has been canceled 3. System puts the taxi in the first position of the queue.
Output conditions	User will be refunded 50% of the amount he paid for the ride and notified about his cancelation.
Exceptions	If the transaction was unsuccessful

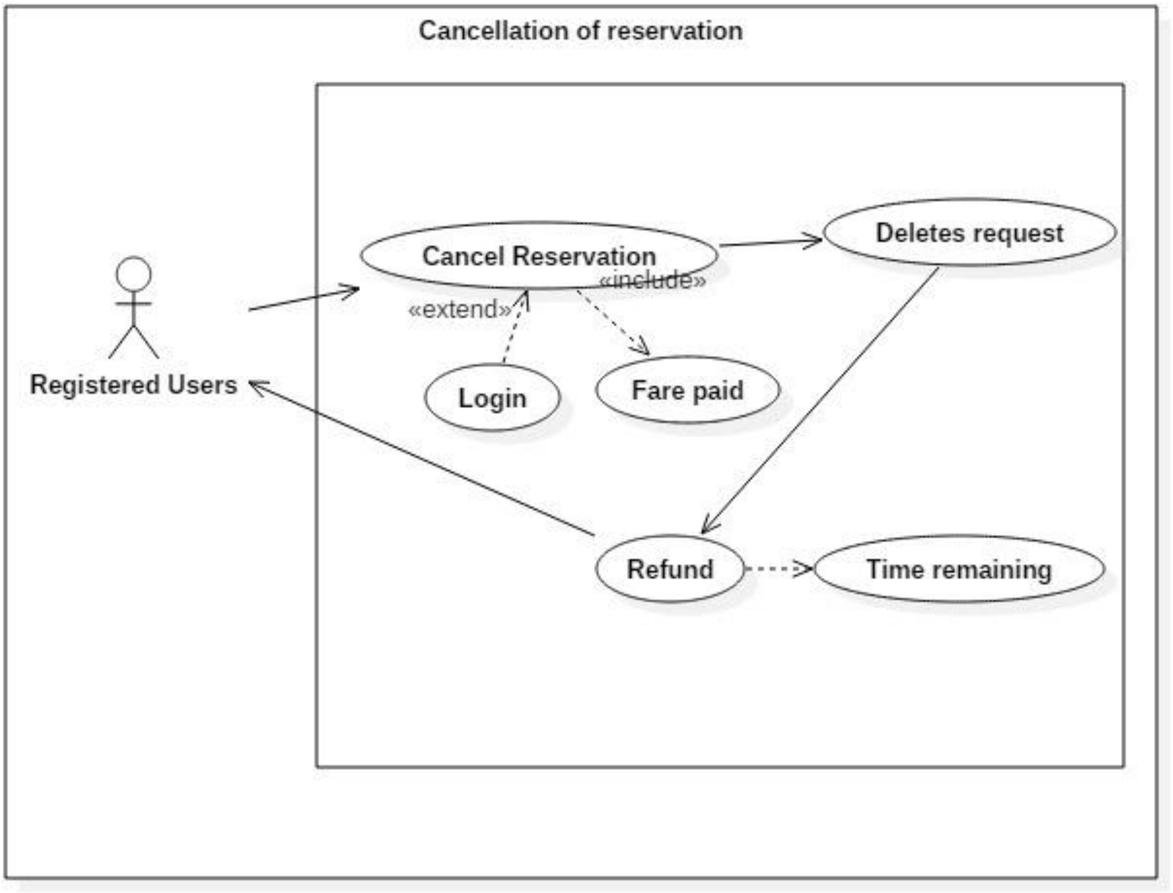

Cancellation of request
Information to driver
Registered user
Driver
«include»
Cancel request
puts taxi back in queue
«extend»
login
«extend»
refund
Taxi fare paid
UseCaseSubject2

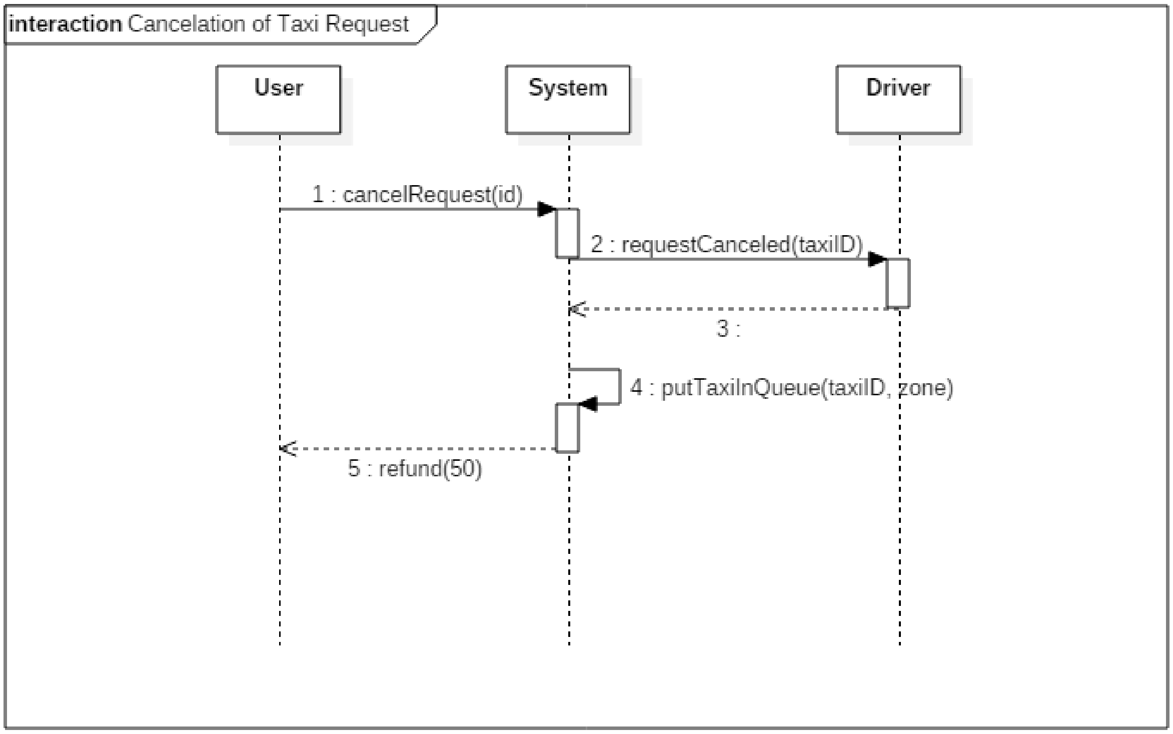

interaction Cancelation of Taxi Request
User
System
Driver
1 : cancelRequest(id)
2 : requestCanceled(taxiID)
3 :
4 : putTaxiInQueue(taxiID, zone)
5 : refund(50)

3.5.1.10 Cancelation of the taxi reservation

Actors	Registered users
Goal	[G11]
Input condition	User must be logged in Taxi proposal for reservation needs to be accepted by the passenger Taxi fare needs to be paid by the passenger
Event flow	1. User clicks on the cancel reservation button in his reservation page. 2. System deletes the reservation from the database

Output conditions	If the user cancels the reservation before 48h of the specified time,he will be refunded 70% of his payment. If he cancels the reservation before 1 hour of the specified time,he will be refunded 50% of his payment. User will be notified about his cancelation.
Exceptions	If the transaction was unsuccessful

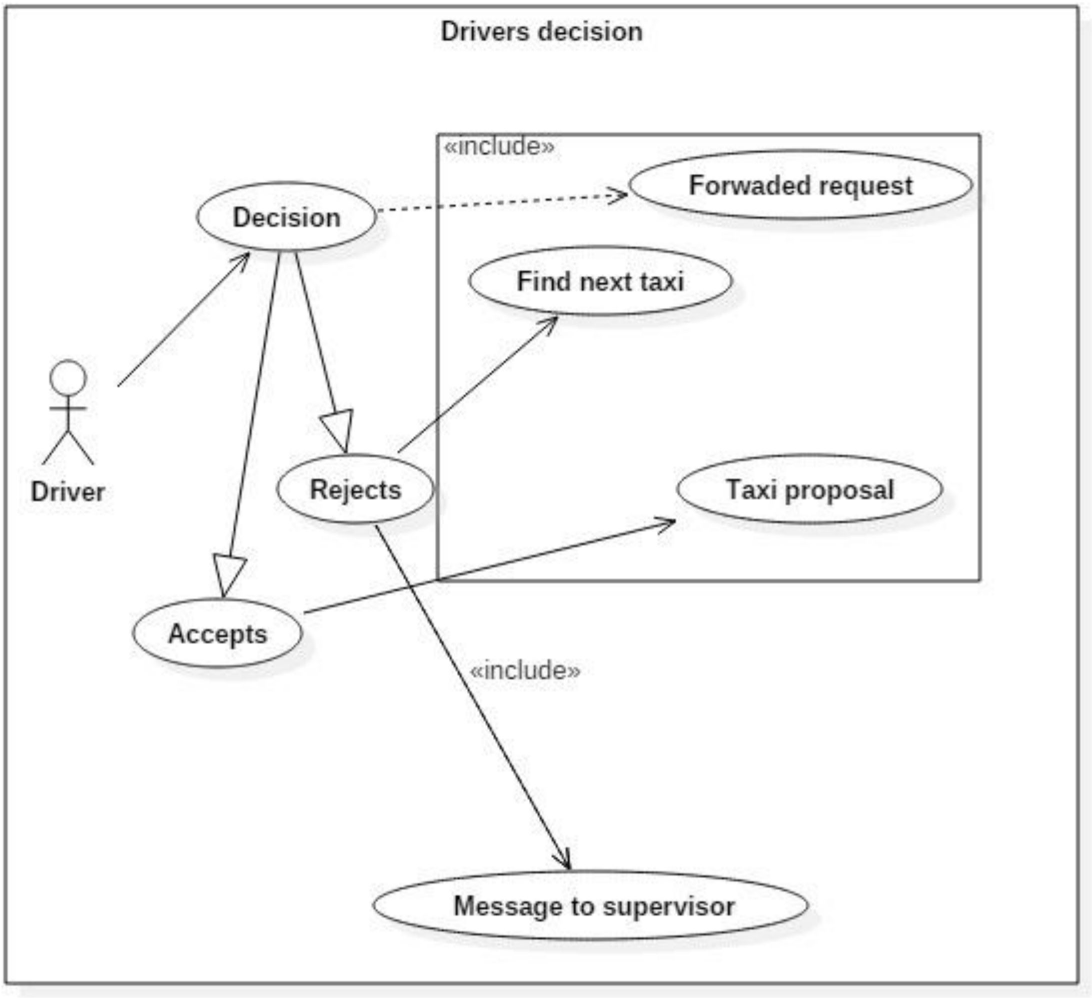


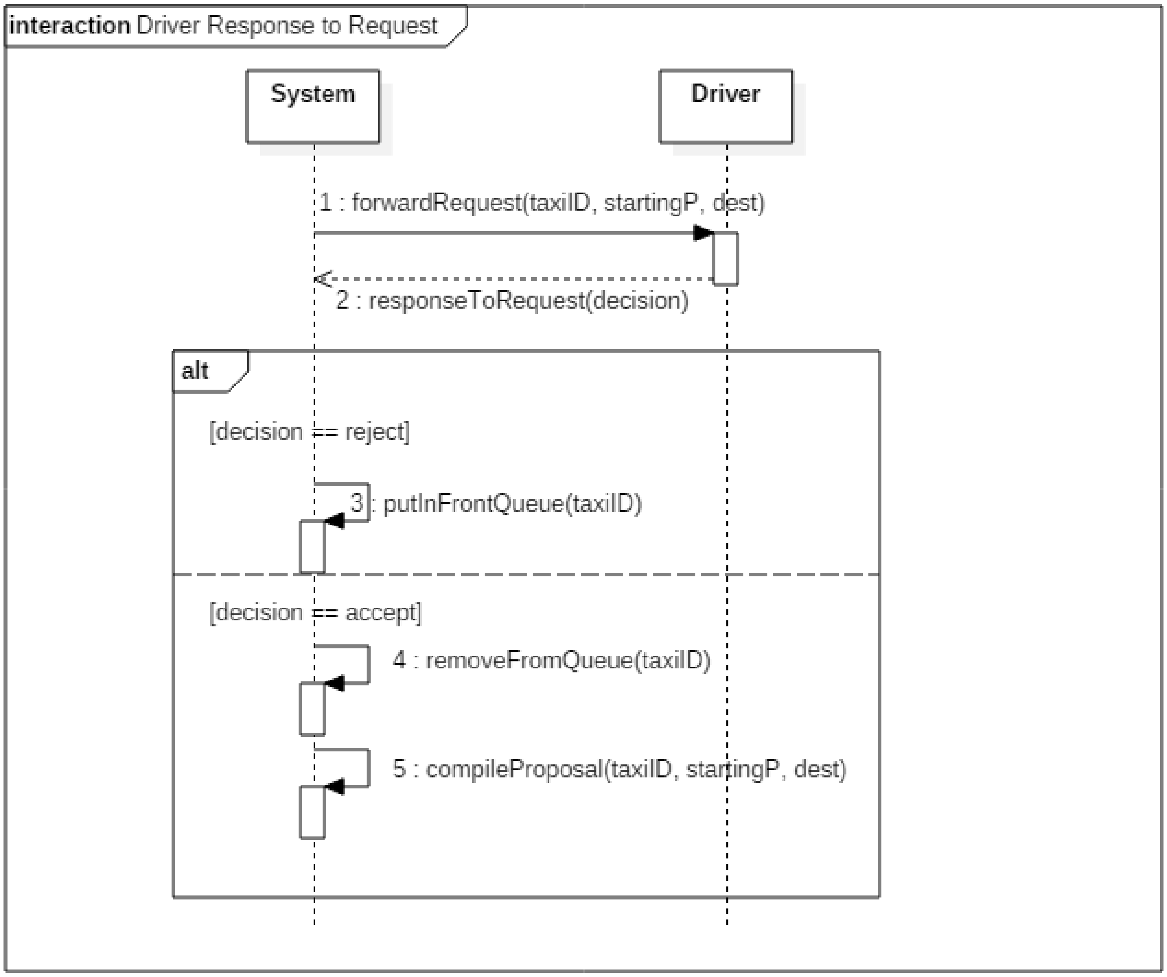

interaction Cancelation of Taxi Request
User
System
Driver
1 : cancelRequest(id)
2 : requestCanceled(taxiID)
3 :
4 : putTaxiInQueue(taxiID, zone)
5 : refund(50)

3.5.1.11 Driver confirmation or rejection of the request

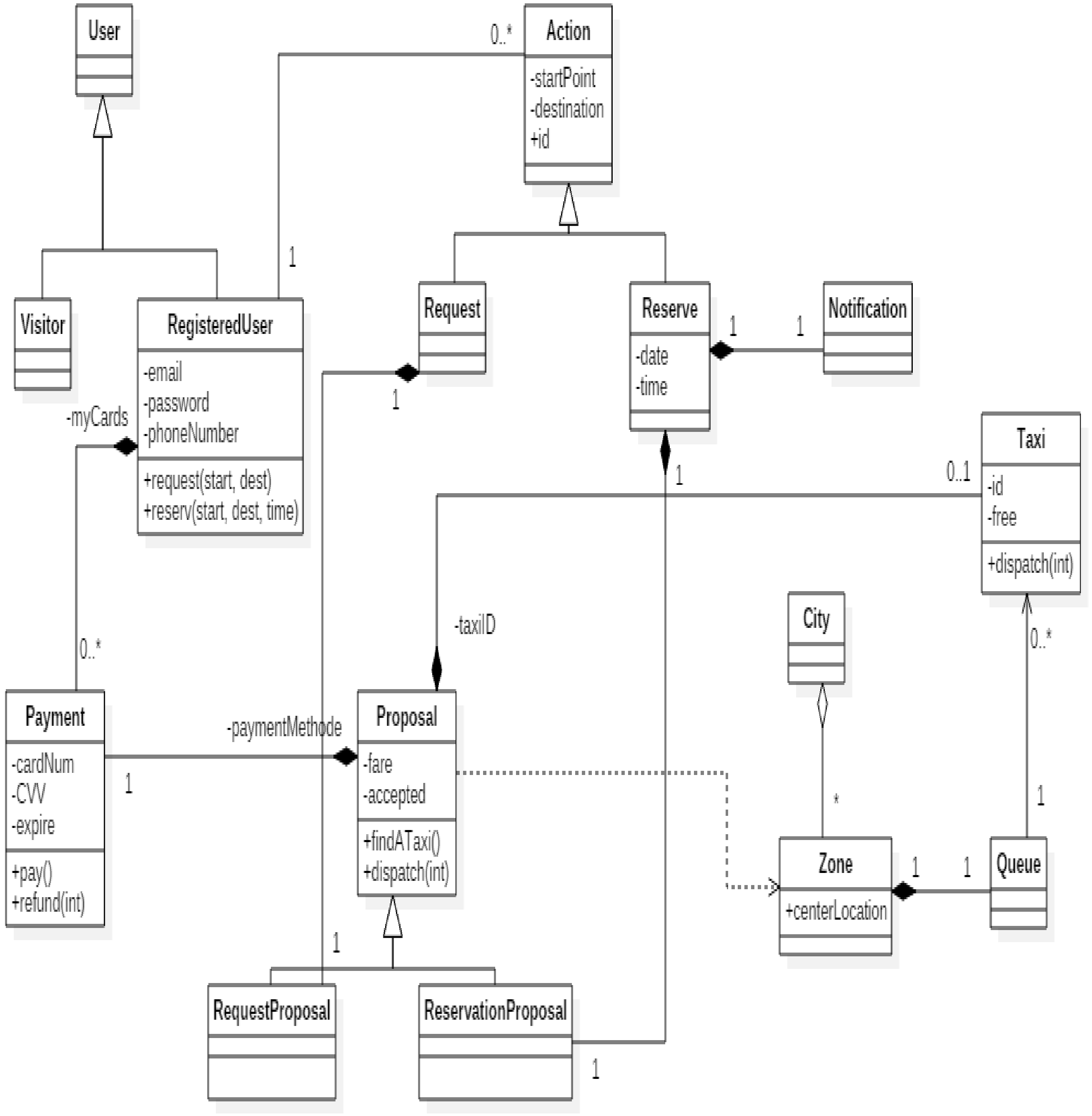
Actors	Driver
Goal	[G12]
Input condition	Taxi should be available and queuing in the zone Taxi request should have been forwarded to the driver
Event flow	1. Driver receives the request with the starting point and destination 2. Driver can choose weather he wants to take the request or reject it
Output conditions	If the driver rejected the request, he will be put to the end of the queue in that zone and the request will be forwarded to the next taxi in the queue.

	If the driver confirms the request, taxi will be removed from the queue and the system proceeds with the taxi proposal.
Exceptions	Null



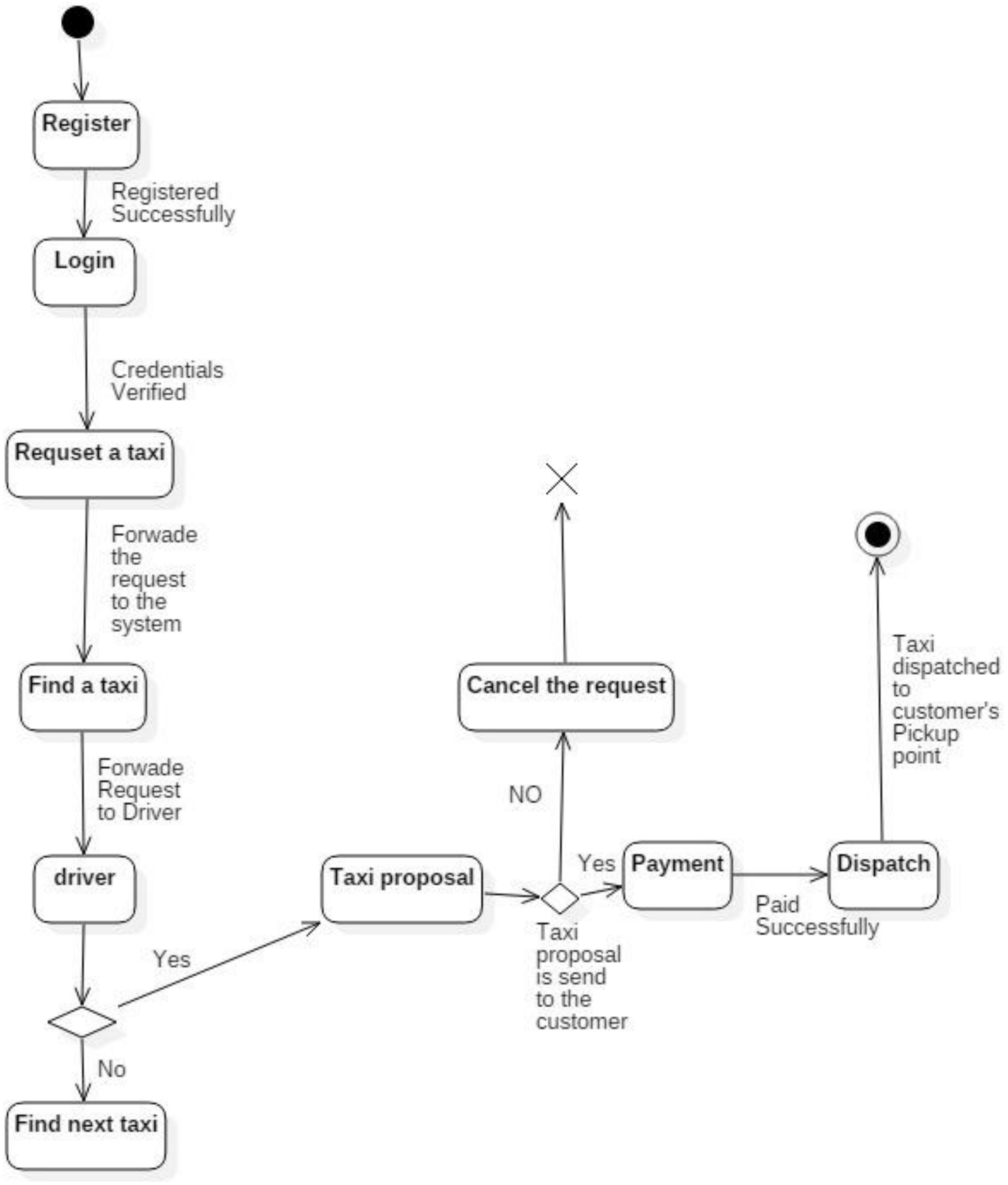

interaction Driver Response to Request
System
Driver
1 : forwardRequest(taxiID, startingP, dest)
2 : responseToRequest(decision)
alt
[decision == reject]
3 : putInFrontQueue(taxiID)
[decision == accept]
4 : removeFromQueue(taxiID)
5 : compileProposal(taxiID, startingP, dest)

3.5.2 Class Diagram

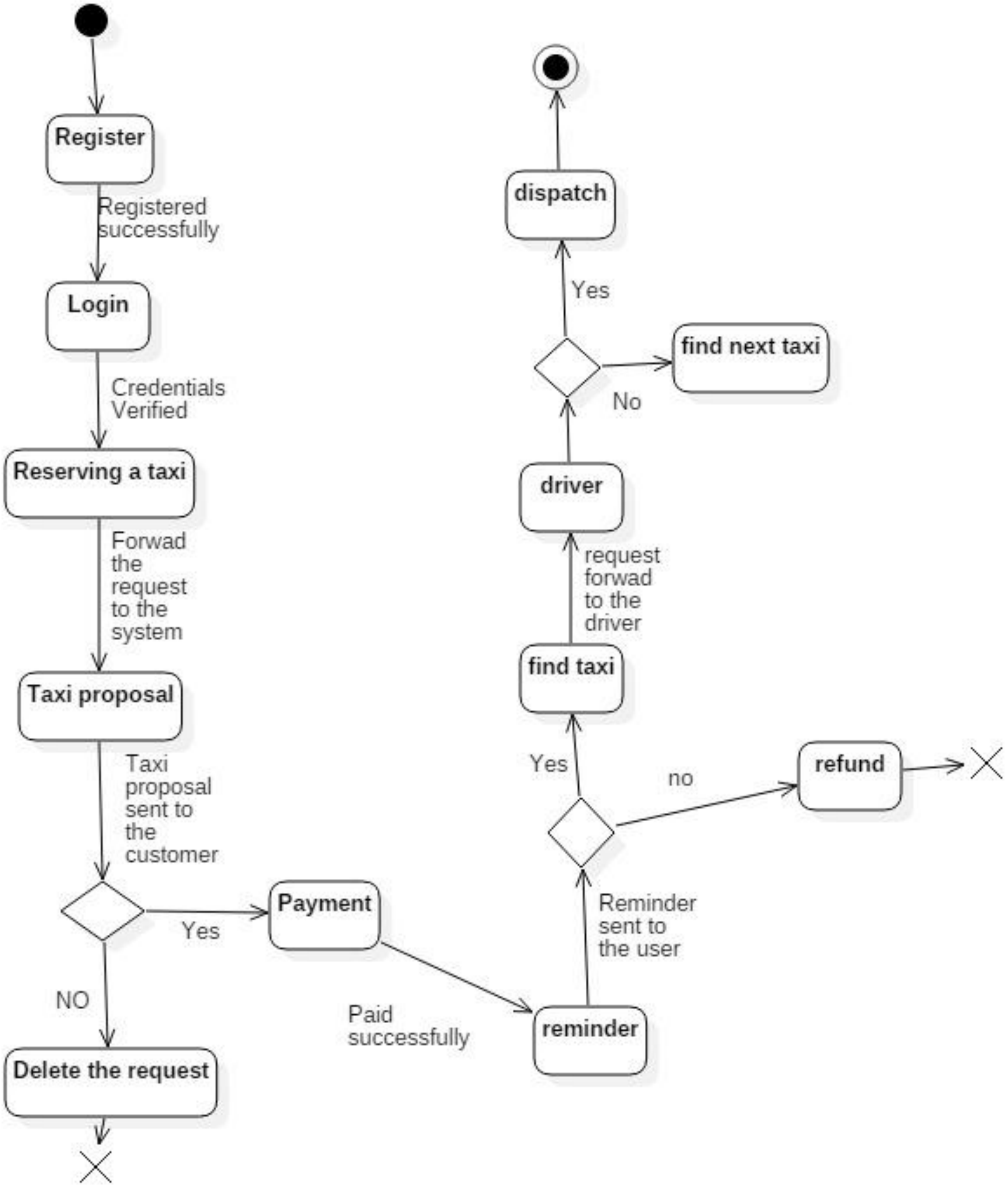


3.5.3 State Machine Diagrams

3.5.3.1 Taxi Request



3.5.3.2 Taxi Reservation



3.6 Other Nonfunctional Requirements

3.6.1 Security

Security should be provided in several points in the system:

- **Password** - The password should contain at least 8 characters, of with there is at least one uppercase letter and one number character. Passwords are hashed and stored in the database.
- **Payment** - Payments are done through an secure payment API
- **HTTPS** - Provide secure HTTP connection
- **Forms protection** - Secure the system from SQL injection attacks

3.6.2 Privacy

The system should provide:

- **Privacy policies**
- **Online Payment policies**
- **User's anonymity**

3.6.3 Performance requirements

The system is assumed to have zero response time but it literally depends on the internet connectivity of the user. It is also assumed that any number of user can send request to the system, so the through put of the system will be maximum.

3.6.4 Usability

For better understanding of the user, simple and well arranged graphical user interface is used (user friendly). For the first time users, tutorial will be provided. In addition to that Informative error messages and help facilities are provided whenever it is needed.

3.6.5 Correctness

It is assumed that system will do things properly since it is an automated system unless it is disturbed externally.

3.6.6 Availability

For the better service ,the system is made available 24 - 7 ,365 days irrespective of external influences.

4 Appendix

4.1 Alloy

4.1.1 Signatures

```
sig Address {}
sig DateTime {}
sig Amount {}
sig Email {}
sig PhoneNum {}

sig RegisteredUser {
        email: one Email,
        phone: one PhoneNum,
        actions: set Action
}

abstract sig Action {
        start: one Address,
        destination: one Address
}

sig Request extends Action {
        proposal: one RequestProposal
}

sig Reserve extends Action {
        datetime: one DateTime,
        proposal: one ReservationProposal
}

abstract sig Proposal {
        fare: one Amount,
        taxi: lone Taxi
}

sig RequestProposal extends Proposal {}

sig ReservationProposal extends Proposal {}

sig City {
        zones: set Zone
} {
        one City
        #zones >= 1
}

sig Zone {
        queue: one Queue
}

sig Taxi {}

sig Queue {
        waiting: set Taxi
}
```

4.1.2 Facts

```
//user constraints
fact UserProperties {
        //no two registered users can have the same email
        all disj u1,u2: RegisteredUser | (u1.email != u2.email)

        //no two registered users can have the same phone number
        all disj u1,u2: RegisteredUser | (u1.phone != u2.phone)
}

//taxi can not wait in more than one queue
fact TaxiQueueing {
        all t: Taxi | lone q: Queue | t in q.waiting
}


fact ZoneProperties {
        //every queue has to be asociated with exactly one zone
        all q: Queue | (one z: Zone | z.queue = q)

        //each zone has to be part of the city
        one c: City | c.zones = Zone
}

fact TaxiAndProposal {
        //if a taxi accepts proposal it is not in the queue
        all r: Proposal, q: Queue | no (r.taxi & q.waiting)

        //taxi can't accept more than one proposal at the time
        all t: Taxi | lone r: Proposal | t in r.taxi

        //differen proposal should be generated for each request/reservation
        all p: RequestProposal | one r: Request | p in r.proposal
        all p: ReservationProposal | one r: Reserve | p in r.proposal

        //amount should only exist if associated within proposal
        all a: Amount | some p: Proposal | p.fare = a

        //for one user there cannot be more than one taxi associated with a proposal at the time
```

```
	//(these are the current proposals)
	no disj r1, r2: Request | one u: RegisteredUser | (r1 in u.actions) and (r2 in u.actions) implies (#r1.proposal + #r2.proposal) = 1

}


fact ActionPropreties {
	//every action should be done by the user
	all a: Action | one u: RegisteredUser | a in u.actions

	//starting point and destination must be different
	all a: Action | a.start != a.destination

	//datetime must always be associated with reservation
	all d: DateTime | one r: Reserve | d in r.datetime

	//user can not reserve multiple taxies at the same time
	all disj r1,r2: Reserve | (r1.datetime = r2.datetime) implies no u: RegisteredUser | (r1 in u.actions) and (r2 in u.actions)
}
```

4.1.3 Assertions

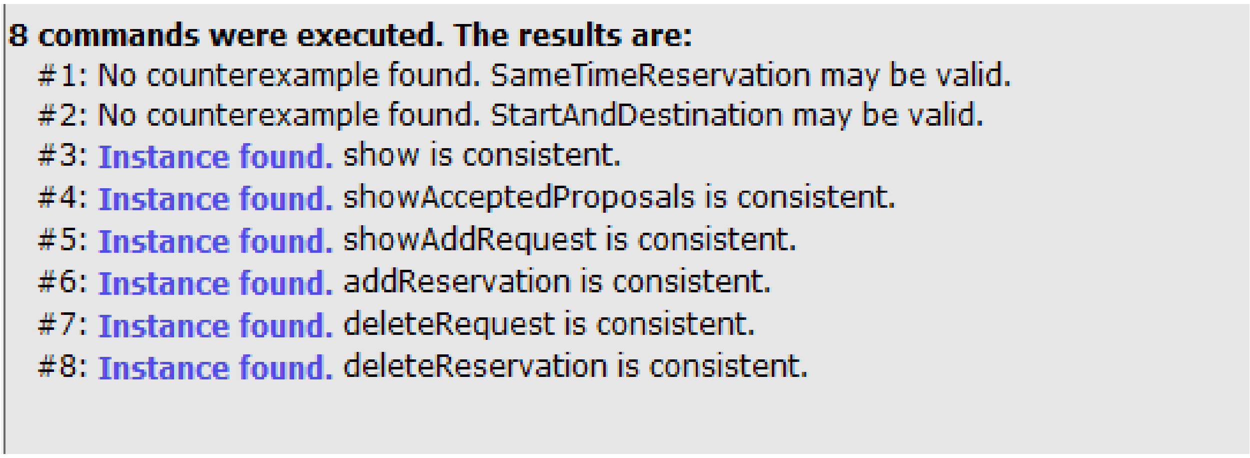
```
//two reservations can't occure in the same time for the same passenger
assert SameTimeReservation {
	no disj r1, r2: Reserve | one u: RegisteredUser | (r1.datetime = r2.datetime) and ((r1 in u.actions) and (r2 in u.actions))
}

check SameTimeReservation for 10

//starting point and destination have to be different
assert StartAndDestination {
	all a: Action | a.start != a.destination
}

check StartAndDestination for 10
```

4.1.4 Predicate

```
pred show {
	#Request>1
	#Reserve>1
	#Taxi>1
}

run show for 4
```

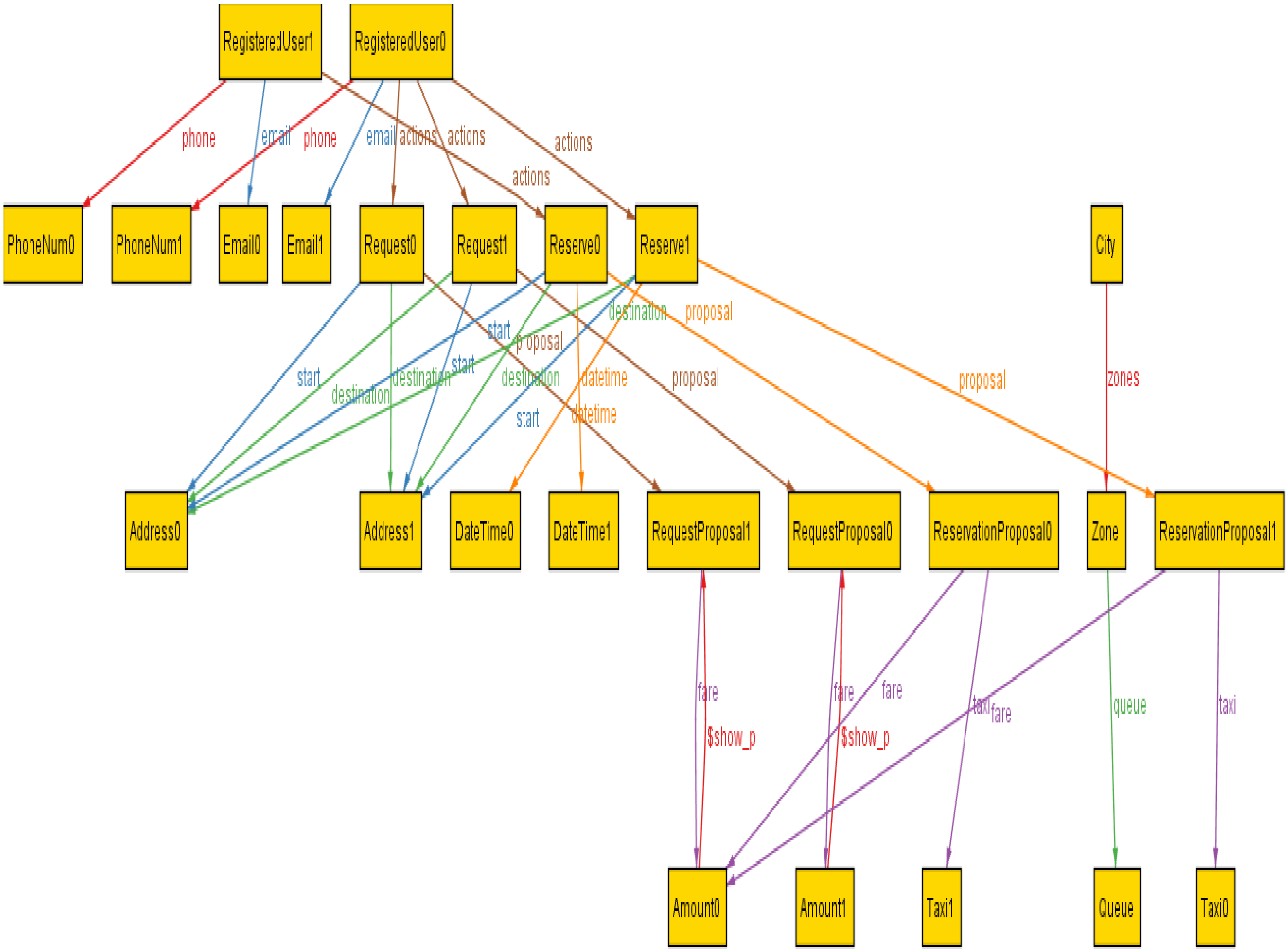
```
//Every forwarded action is accepted by the different taxi
pred showAcceptedProposals {
        all p: Proposal | one t: Taxi | p.taxi = t
        #Proposal > 2
}

run showAcceptedProposals for 5

//add a request
pred addRequest [u1,u2: RegisteredUser, r: Request] {
        (r not in u1.actions) implies u2.actions = u1.actions + r
}

pred showAddRequest [u1,u2: RegisteredUser, r: Request] {
        addRequest[u1, u2, r]
}
run showAddRequest

//add a reservation
pred addReservation [u1,u2: RegisteredUser, r: Reserve] {
        (r not in u1.actions) implies u2.actions = u1.actions + r
}

run addReservation

//delete a request
pred deleteRequest [u1,u2: RegisteredUser, r: Request] {
        (r in u1.actions) implies u2.actions = u1.actions - r
}

run deleteRequest

//delete a reservation
pred deleteReservation [u1,u2: RegisteredUser, r: Reserve] {
        (r in u1.actions) implies u2.actions = u1.actions - r
}

run deleteReservation
```

4.1.5 Result



```
8 commands were executed. The results are:
   #1: No counterexample found. SameTimeReservation may be valid.
   #2: No counterexample found. StartAndDestination may be valid.
   #3: Instance found. show is consistent.
   #4: Instance found. showAcceptedProposals is consistent.
   #5: Instance found. showAddRequest is consistent.
   #6: Instance found. addReservation is consistent.
   #7: Instance found. deleteRequest is consistent.
   #8: Instance found. deleteReservation is consistent.
```

4.1.6 Generated World



4.2 Software and tool used

- **MockingBot –**To create mockup’s.
- **StarUml**-To create use case,sequence,class and state machine diagrams.
- **AlloyAnalyzer**-To analyze our model.
- Git-Team collaborations and Version control.

4.3 Hours of Works

Krishnan Ranjithkumar ~ 30 hrs
Korda Petar ~ 30hrs