1 One

2 Two

3 Specific Requirements

This section of the document is dedicated at giving an in-depth description of the platform's requirements, and is to be kept as reference during all future phases of development.

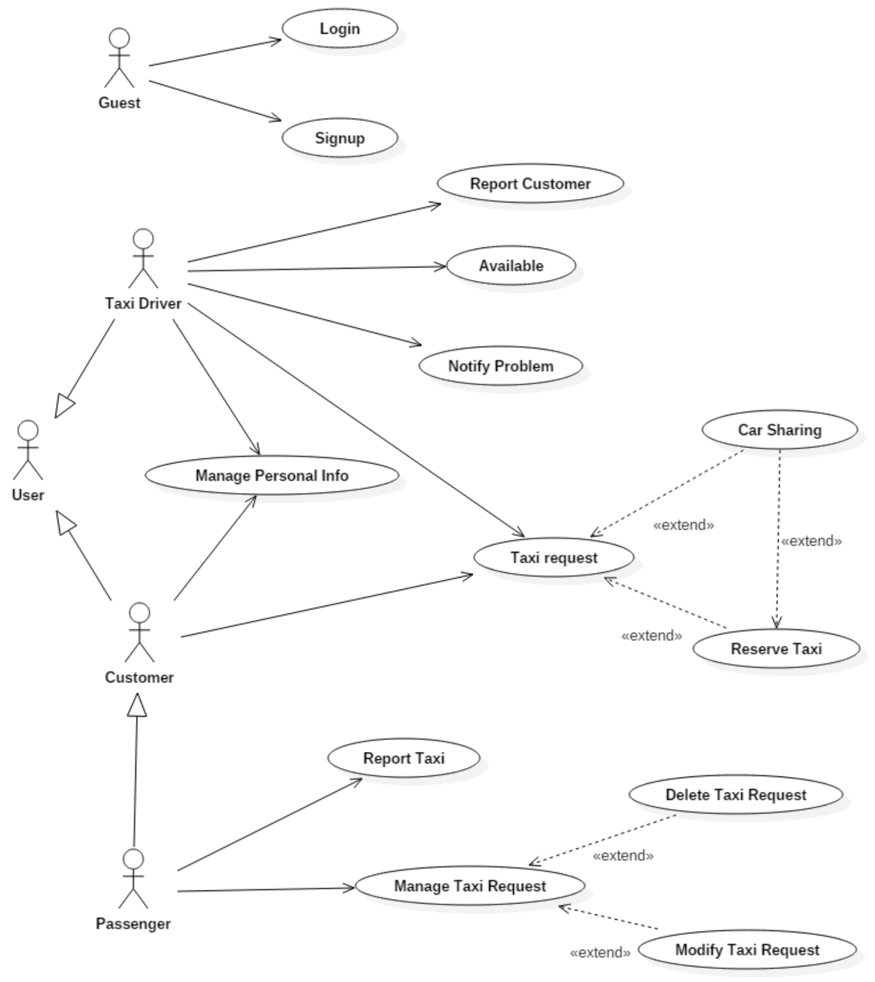
3.1 External interfaces

Being MyTaxiService a fully service-oriented platform, its only external interfaces must be those reserved for the final users; there is no need to design specific maintenance access to the back-end system as this is already fully standardized and does not need specific functionalities other than the usual system administration tools.

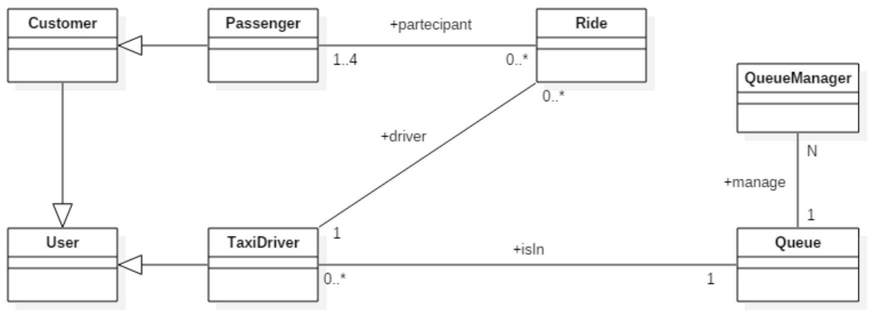
As briefly described in section **??**, the main principle that must guide the design of the external interfaces of the platform is that of business identity continuity. This section contains a set of design mock-ups that are to be kept as reference during the development of the user interfaces.

3.2 Functions

3.3 Use Cases



3.4 Class Diagram



3.5 Scenarios

To help the reader understand the above stated requirements, a brief description of how a use case might look like in the real world is given below.

In the examples, Adam and Joanne are customers who intend to request a taxi and Hector, Monica, Jim and Samuel are taxi drivers of the town.

3.5.1 Sign up

Adam has just downloaded the customer-side app and wants to sign up into the platform. He requests the customer registration page, fills the form and submit the request to the system. If Adam's e-mail and username are unique the system gives Adam a confirmation of the success of the operation and redirects Adam to the login page, otherwise it displays an error message.

3.5.2 Login

Adam, now registered, inserts the username and password in the login form and clicks the login button; the system checks the information and, if the username-pasword combination is correct, redirects Adam to his own user profile page; otherwise, an error message is shown.

3.5.3 Available

Hector, already logged into the platform, starts his working by day opening his taxi driver profile page and communicating his availability to the system. The system updates the taxi queue in Hector's zone and sends Hector a notification with his position in the queue.

3.5.4 Taxi request

Adam, now logged into the system, wants to book a taxi to go home. He opens the taxi request page on the app, and requests a taxi. The system forwards

Adam's request to the queue associated with Adam's position, and Hector, which is the first taxi driver in the queue, is notified with the request.

Unfortunately, Hector has now decided to take a break and does not want to take this ride; he refuses Adam's request by tapping a button on the app, and the system forwards the request to Monica, the taxi driver immediately after Hector in the queue.

As she accepts Adam's request, Adam receives a notification on the app with the estimated waiting time.

3.5.5 Book a Taxi

While on Monica's taxi, Adam wants to book a taxi for that evening at 6 PM, in order to go to the cinema. He opens the taxi request page of the app, and fills and submits the request form.

The system checks the information (sending eventual error notifications back to Adam) and forwards Adam's request to Jim, by using a specific selection algorithm over taxi drivers in the queue associated to Adam's zone.

Jim decides to accept Adam's booking, and will keep his schedule free for the time that Adam requested.

3.5.6 Manage Taxi Request

Later that day, Adam opens the "Manage taxi request" page from his laptop's browser to change the booking time from 6PM to 7PM. The system checks whether Adam's request is feasible (there must be at least two hours between the current time and the requested time), and eventually forwards the changes to Jim.

Jim accepts the modification and a confirmation is sent back to Adam.

3.5.7 Report Taxi

Jim picks Adam up at 7PM. During the ride Jim lights up a cigarette and is unreasonably rude towards Adam.

Adam opens the "Report taxi" page on the app, to file a complaint about Jim's behavior. The system updates Jim's profile information with the new report and confirms the success of the operation to Adam.

3.5.8 Report User

After the ride, Adam is annoyed by the behavior of Jim and refuses to pay for the ride.

Jim opens the "Report user" page, fills the complaint form and submits it to the system. The system updates Adam's profile information with the new report and confirms the success of the operation to Jim.

3.5.9 Manage Personal Information

Joanne has opened a new main email account.

She opens his profile page from the app, clicks on the edit button and changes his email address to match the new one; she then submits the new information.

The system performs a check on the information, updates Joanne's profile and notifies the success of the operation to Joanne.

3.5.10 Report Problem

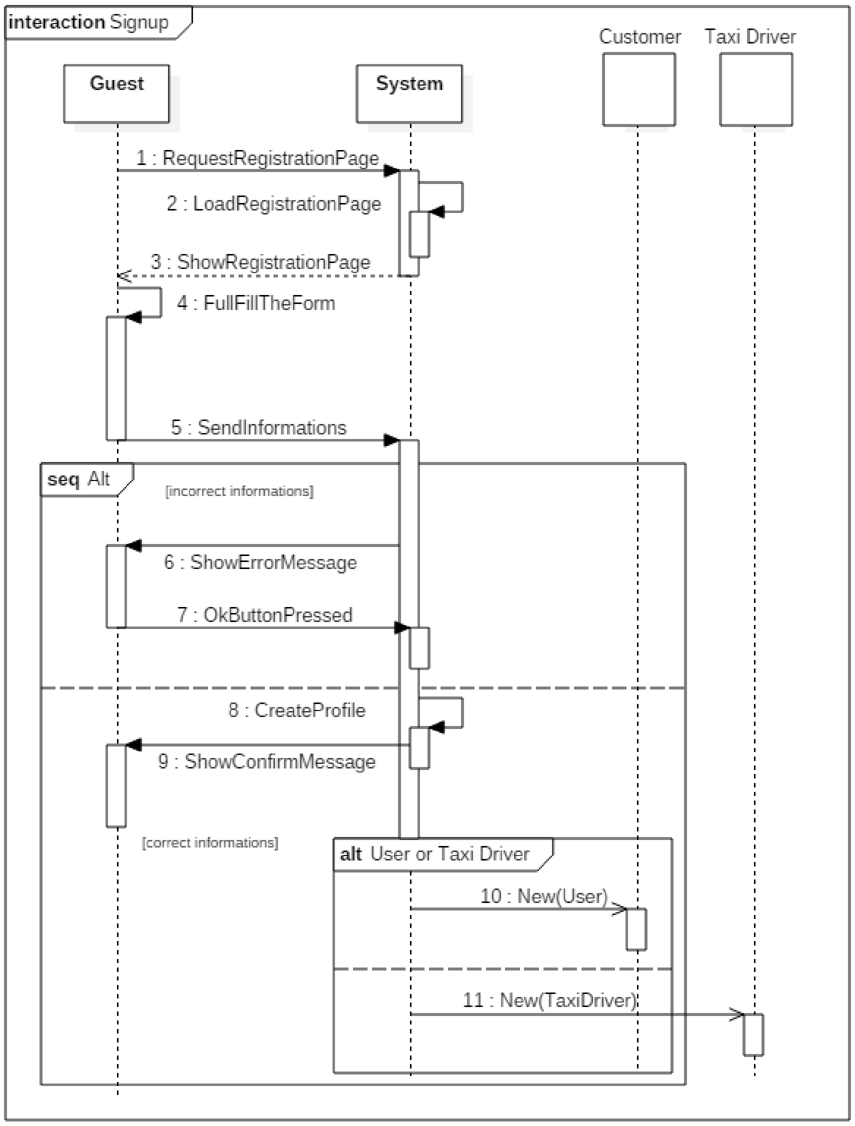
During a ride, Hector has a problem with his taxi's engine and can't bring Joanne to destination.

Through the "Report problem" page of the app, he notifies the problem to the system, by filling the form and submitting. The system acknowledges the report and asks Hector if needs a new taxi; Hector confirms, and the system forwards his request to Samuel, who is the first taxi driver in Hector and Joanne's current zone.

3.6 Flows of events

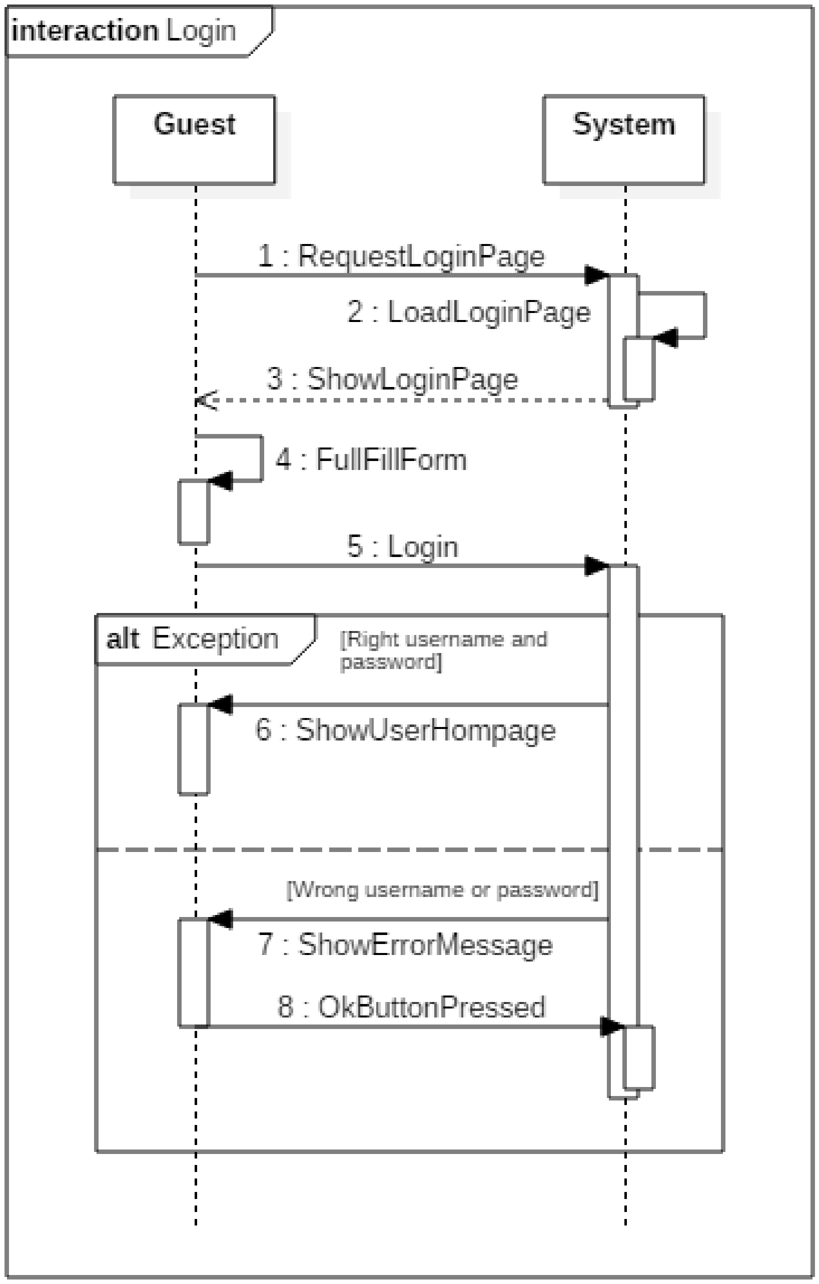
3.6.1 Sign-up

Actors	Guest
Preconditions	The guest is not registered into the system
Execution Flow	1. The guest requests the registration page 2. The system asks for the sign-up information 3. The guest fills the form and submits the request 4. The system checks the uniqueness of the username and e-mail 5. The system creates the customer (or taxi driver) profile 6. The system sends the confirmation to the guest
Postconditions	The guest is now a registered user
Exceptions	The e-mail or username are not unique or, in the case of taxi driver sign-up, the license is not valid


interaction Signup
Guest
System
Customer
Taxi Driver
1 : RequestRegistrationPage
2 : LoadRegistrationPage
3 : ShowRegistrationPage
4 : FullFillTheForm
5 : SendInformations
seq Alt
[incorrect informations]
6 : ShowErrorMessage
7 : OkButtonPressed
8 : CreateProfile
9 : ShowConfirmMessage
[correct informations]
alt User or Taxi Driver
10 : New(User)
11 : New(TaxiDriver)

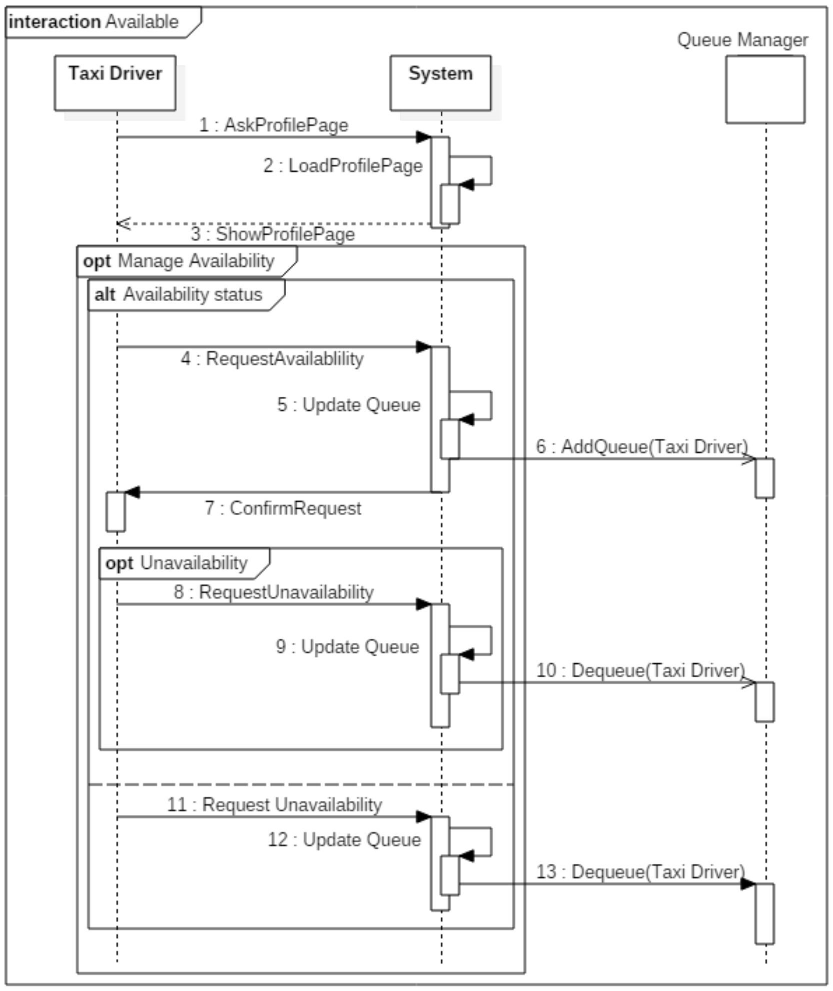
3.6.2 Login

Actors	Guest
Preconditions	The guest has already a profile into the system
Execution Flow	1. The guest requests the login page 2. The system requires the login information (Username, password) 3. The guest fills the form and submits the request 4. The system checks the username and password 5. The system sends a login confirmation 6. The guest is logged into the system 7. The guest is redirected to the user profile page
Postconditions	The guest is now a logged-in user
Exceptions	The username, password combination is incorrect, so the guest cannot log in


interaction Login
Guest
System
1 : RequestLoginPage
2 : LoadLoginPage
3 : ShowLoginPage
4 : FullFillForm
5 : Login
alt Exception
[Right username and password]
6 : ShowUserHompage
[Wrong username or password]
7 : ShowErrorMessage
8 : OkButtonPressed

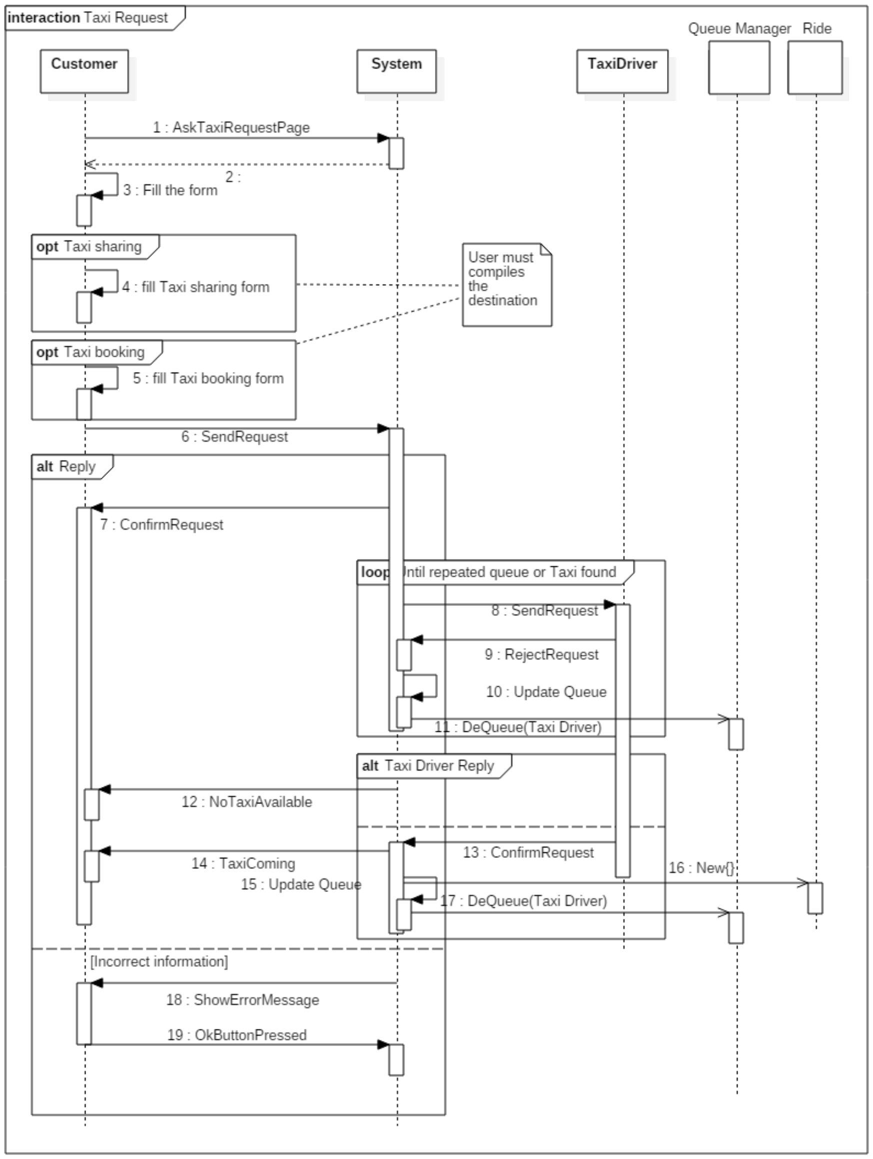
3.6.3 Available

Actors	Taxi driver
Preconditions	
Execution Flow	1. The taxi driver requests the taxi profile page 2. The system returns the personal information of the user 3. The taxi driver can choose to change the his availability, becoming available or unavailable 4. The taxi driver send the request to the system 5. The system update the queue 6. The system return a confirmation to the taxi driver
Postconditions	The taxi driver is now available
Exceptions	• The taxi driver is located in a invalid zone • The taxi driver is carrying a passenger • The taxi driver is not available


interaction Available
Taxi Driver
System
Queue Manager
1 : AskProfilePage
2 : LoadProfilePage
3 : ShowProfilePage
opt Manage Availability
alt Availability status
4 : RequestAvailablility
5 : Update Queue
6 : AddQueue(Taxi Driver)
7 : ConfirmRequest
opt Unavailability
8 : RequestUnavailability
9 : Update Queue
10 : Dequeue(Taxi Driver)
11 : Request Unavailability
12 : Update Queue
13 : Dequeue(Taxi Driver)

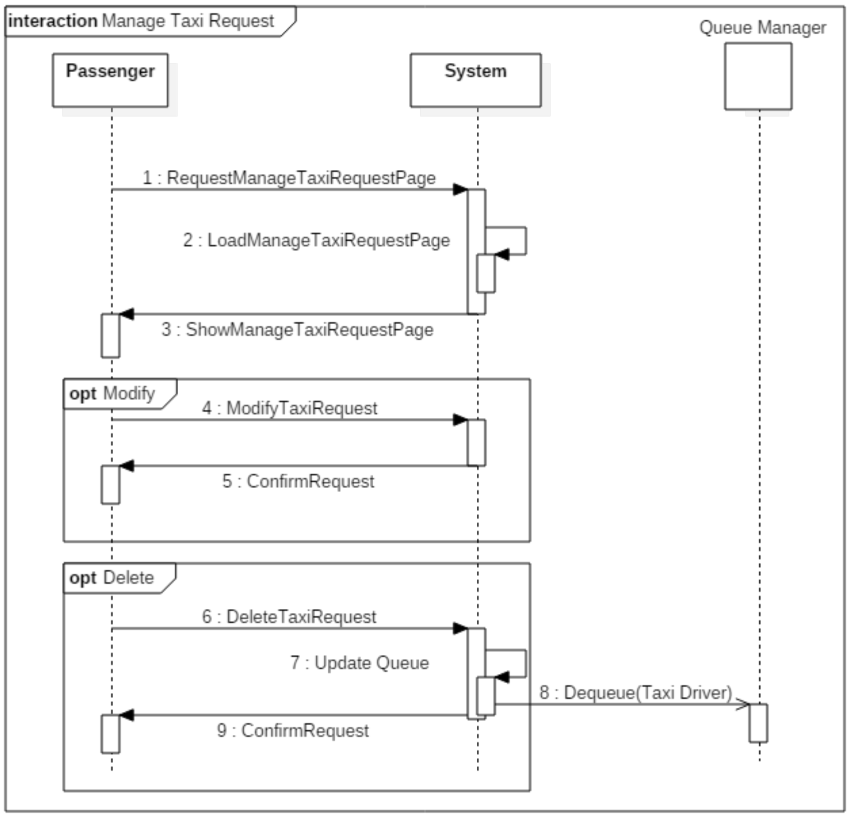
3.6.4 Taxi Request

Actors	User, Taxi driver
Preconditions	The user should not be banned
Execution Flow	1. The user requests the taxi request page 2. The system asks for the type of request that the user wants to issue 3. The user fills the request form and sends the information to the system 4. The system forwards the request to the first taxi driver of the local queue 5. If the taxi driver answers positively to the request, he takes the user in charge, otherwise he denies the request 6. If the taxi driver accepts the request, the system notifies to the user the incoming taxi and changes the availability of the taxi driver; otherwise, the system updates the queue and forwards the request to the new first taxi driver of the queue. 7. If there are no taxis available, the system notifies the user.
Postconditions	If the request is accepted by a taxi driver, the user is now a passenger
Exceptions	• The user provides incorrect information in the request form • The user is not in a valid position (*e.g. outside the town*)


interaction Taxi Request
Customer
System
TaxiDriver
Queue Manager
Ride
1 : AskTaxiRequestPage
2 :
3 : Fill the form
opt Taxi sharing
4 : fill Taxi sharing form
User must compiles the destination
opt Taxi booking
5 : fill Taxi booking form
6 : SendRequest
alt Reply
7 : ConfirmRequest
loop Until repeated queue or Taxi found
8 : SendRequest
9 : RejectRequest
10 : Update Queue
11 : DeQueue(Taxi Driver)
alt Taxi Driver Reply
12 : NoTaxiAvailable
13 : ConfirmRequest
14 : TaxiComing
15 : Update Queue
16 : New{}
17 : DeQueue(Taxi Driver)
[Incorrect information]
18 : ShowErrorMessage
19 : OkButtonPressed

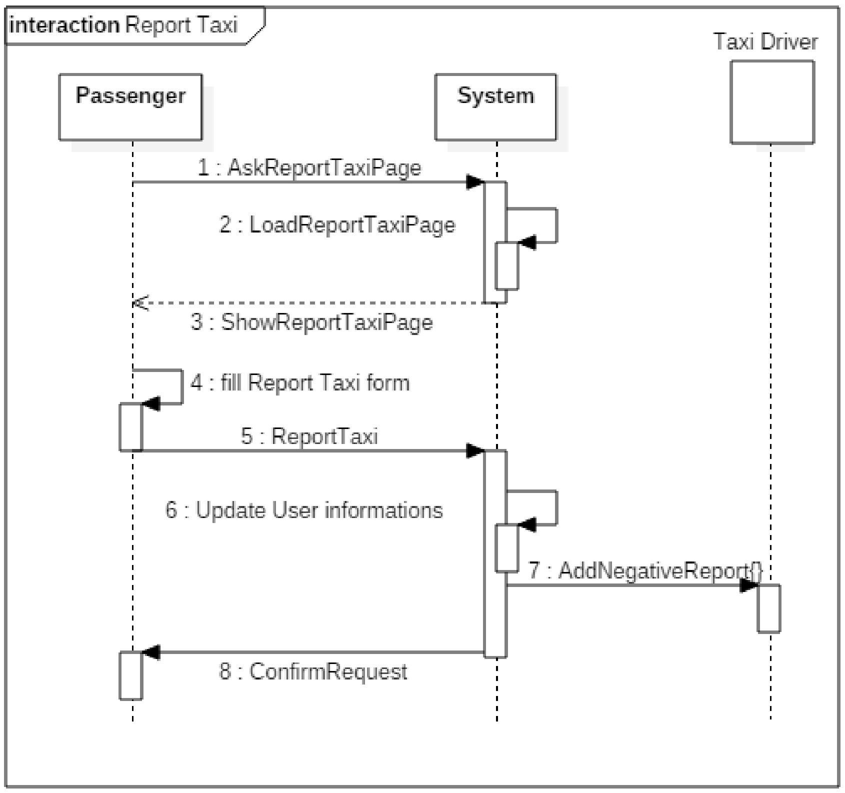
3.6.5 Manage Taxi Request

Actors	Passenger
Preconditions	
Execution Flow	1. The passenger request the taxi Request Management page 2. The system load the page and return it to the passenger 3. The passenger can modify the Request full filling the modify Request 4. The system modify the request and return a confirm to the passenger 5. The passenger can delete the request, submitting the operation to the system 6. The system update the queue and return a confirmation to the passenger
Postconditions	• If the passenger choose to modify the Request, the request is updated • If the passenger choose to delete the Request, the request is canceled and the taxi is now in queue
Exceptions	• The passenger provides incorrect information in the modify request form • The passenger cancel the request too late


interaction Manage Taxi Request
Passenger
System
Queue Manager
1 : RequestManageTaxiRequestPage
2 : LoadManageTaxiRequestPage
3 : ShowManageTaxiRequestPage
opt Modify
4 : ModifyTaxiRequest
5 : ConfirmRequest
opt Delete
6 : DeleteTaxiRequest
7 : Update Queue
8 : Dequeue(Taxi Driver)
9 : ConfirmRequest

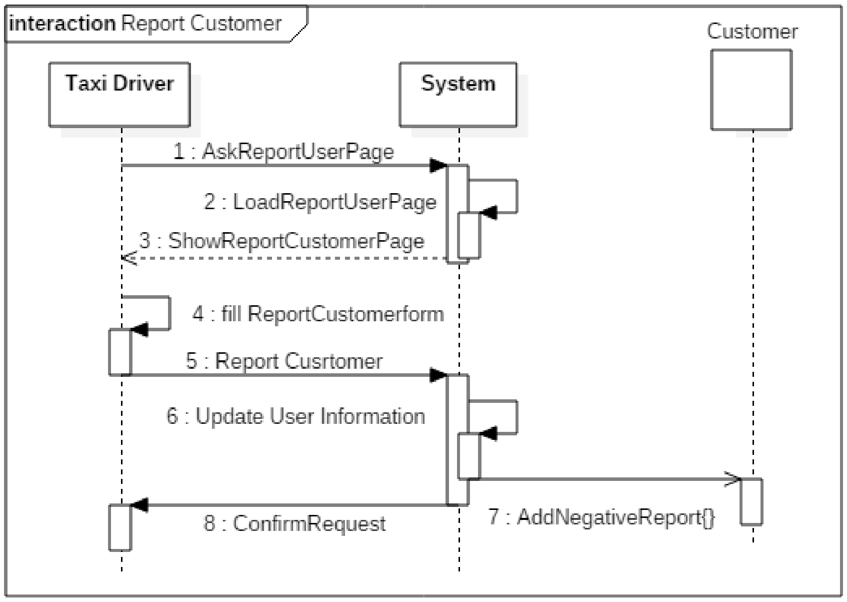
3.6.6 Report Taxi

Actors	Passenger
Preconditions	
Execution Flow	1. The passenger request the Report taxi page 2. The system elaborate the request and ask to the passenger the report information 3. The passenger fill the form and submit the report 4. The system check the data obtained 5. The system update the taxi driver information 6. The system notify to the passenger the successful of the operation
Postconditions	The taxi driver is reported by the current passenger
Exceptions	The passenger provides wrong information in the report form


interaction Report Taxi
Taxi Driver
Passenger
System
1 : AskReportTaxiPage
2 : LoadReportTaxiPage
3 : ShowReportTaxiPage
4 : fill Report Taxi form
5 : ReportTaxi
6 : Update User informations
7 : AddNegativeReport()
8 : ConfirmRequest

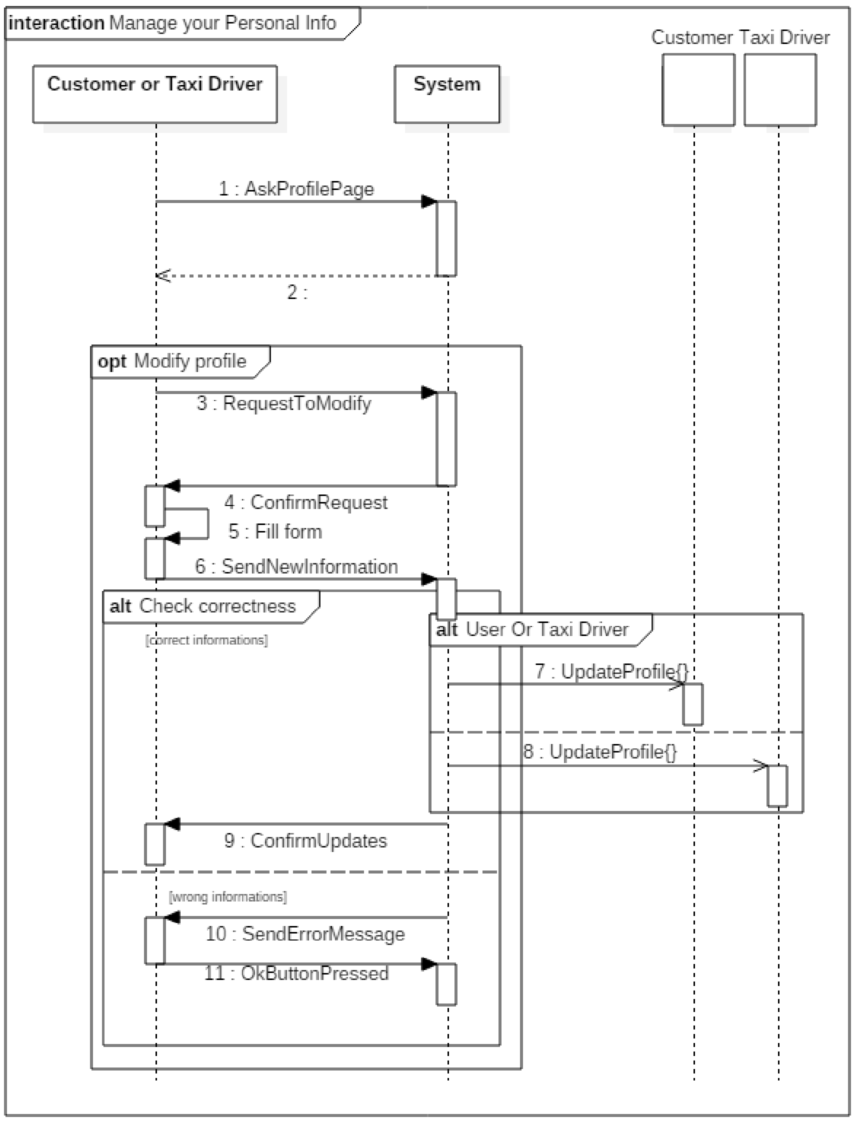
3.6.7 Report User

Actors	Taxi driver
Preconditions	The taxi driver should had carried the user past 24 hours
Execution Flow	1. The taxi driver request the Report user page 2. The system elaborate the request and ask to the taxi driver the report information 3. The taxi driver fill the form and submit the report 4. The system check the data obtained 5. The system update the user information 6. The system notify to the taxi driver the successful of the operation
Postconditions	The user is reported by the taxi driver
Exceptions	The taxi driver provides wrong information in the report form



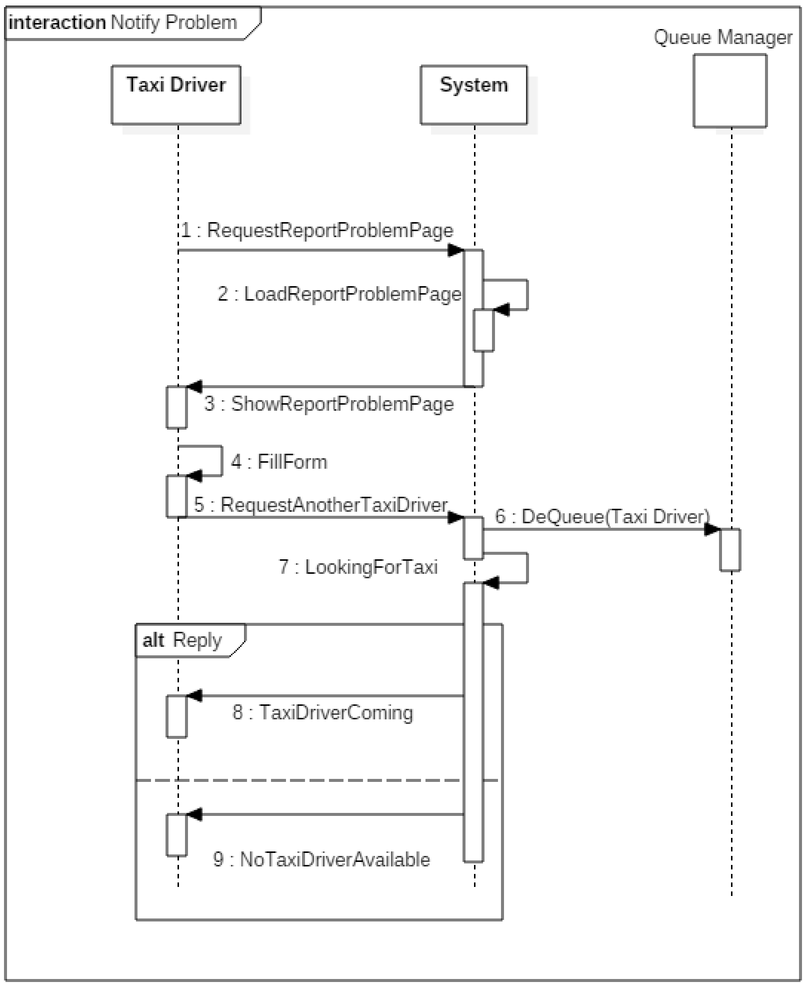
3.6.8 Manage Personal Information

Actors	User or Taxi driver
Preconditions	
Execution Flow	1. The user (or taxi driver) request the user profile page 2. The system returns the personal information of the user (or taxi driver) 3. The user (or taxi driver) can request the edit of the profile 4. The system return the editable information of the profile 5. The user (or taxi driver) can edit those data and send the changes to the system 6. The system check the correctness of the new information 7. If the information are correct send the Change Confirm to the user (or taxi driver), otherwise send back an error message
Postconditions	If the user request a modify profile and submit correct information the Profile is changed
Exceptions	The user provides wrong information


interaction Manage your Personal Info
Customer or Taxi Driver
System
Customer Taxi Driver
1 : AskProfilePage
2 :
opt Modify profile
3 : RequestToModify
4 : ConfirmRequest
5 : Fill form
6 : SendNewInformation
alt Check correctness
[correct informations]
alt User Or Taxi Driver
7 : UpdateProfile{}
8 : UpdateProfile{}
9 : ConfirmUpdates
[wrong informations]
10 : SendErrorMessage
11 : OkButtonPressed

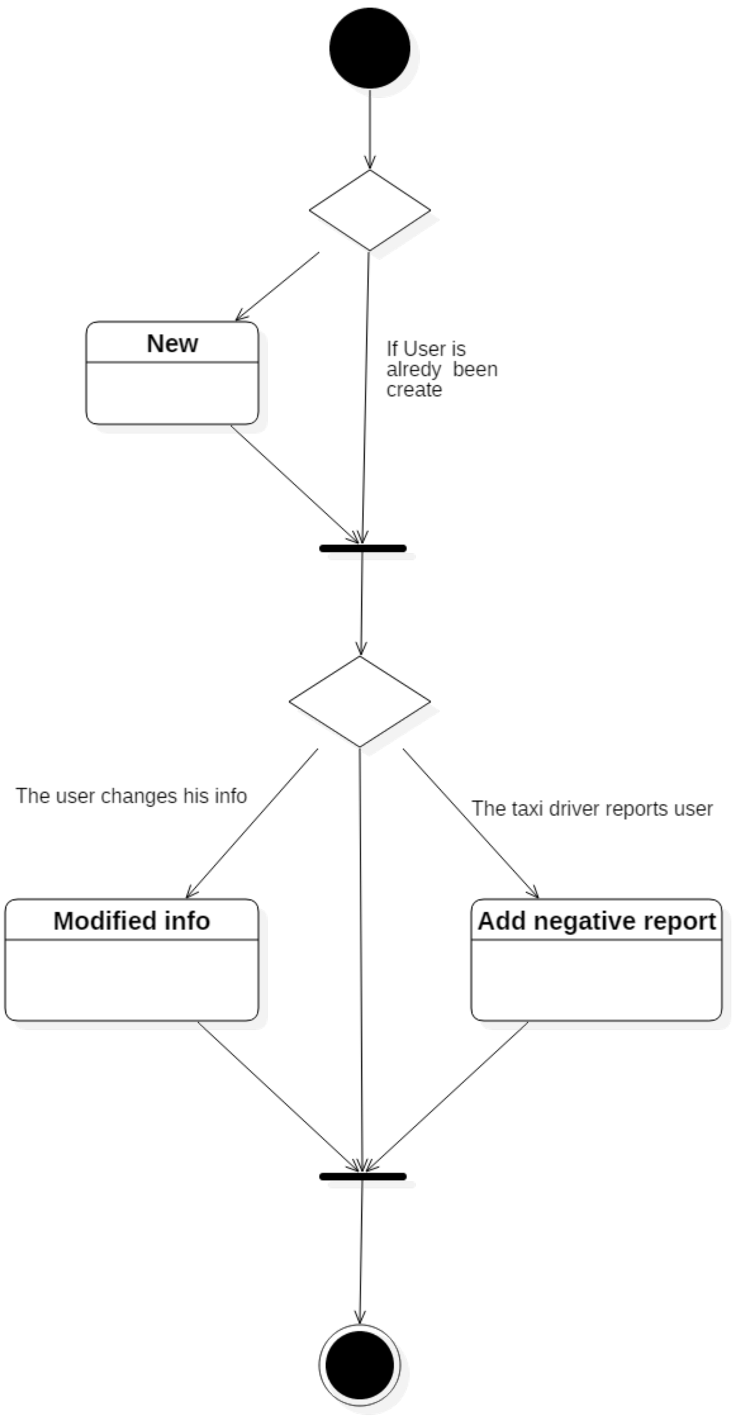
3.6.9 Report Problem

Actors	Taxi driver
Preconditions	
Execution Flow	1. The taxi driver request the Report Problem page 2. The system returns the page of the Report requiring problem information 3. The taxi driver fills the form and sends the information 4. The taxi driver can request another taxi for the user who is carrying (if is carrying one) 5. If the taxi driver request another taxi the system looks for a available taxi driver 6. If exist an available taxi change the carrier of the user with the new taxi driver and send him, otherwise notify the unavailability 7. The system return the success of the operation
Postconditions	The problem is submitted into the system
Exceptions	• The taxi driver is located in a invalid zone • The taxi driver fill the form with wrong information

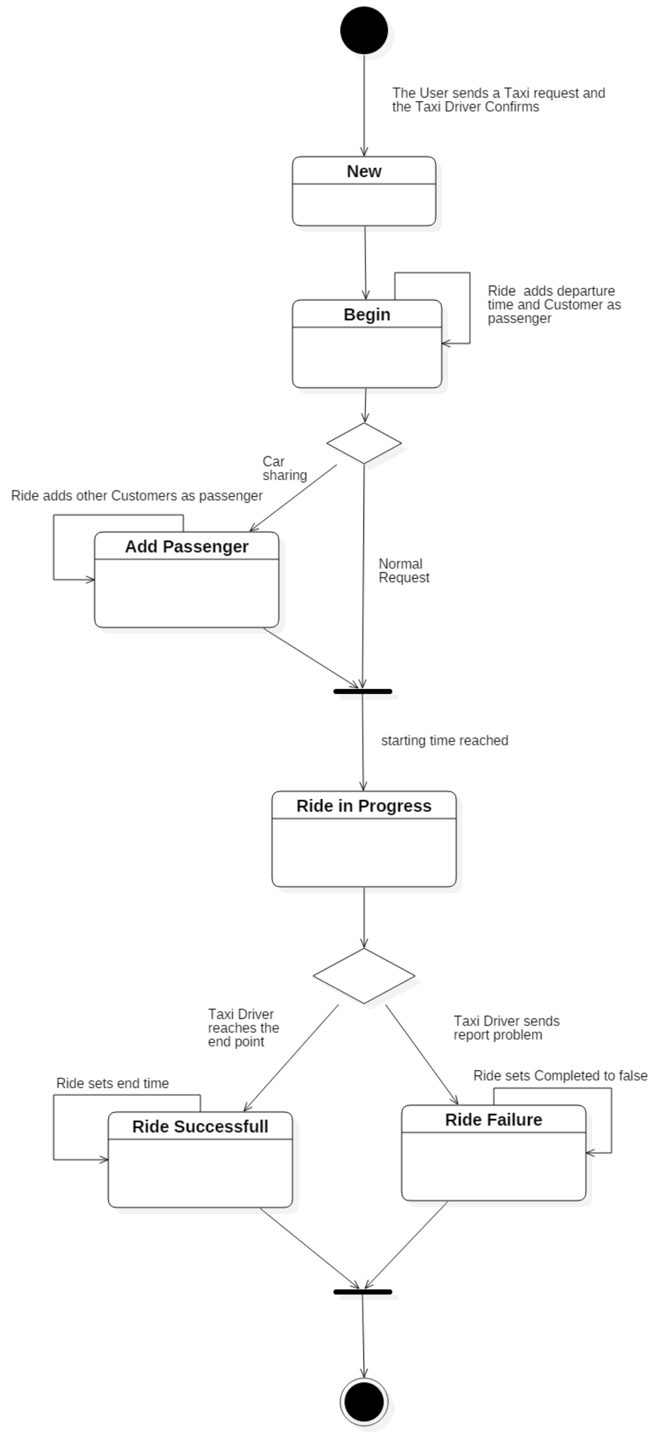

interaction Notify Problem
Queue Manager
Taxi Driver
System
1 : RequestReportProblemPage
2 : LoadReportProblemPage
3 : ShowReportProblemPage
4 : FillForm
5 : RequestAnotherTaxiDriver
6 : DeQueue(Taxi Driver)
7 : LookingForTaxi
alt Reply
8 : TaxiDriverComing
9 : NoTaxiDriverAvailable

3.7 Entities Behavior

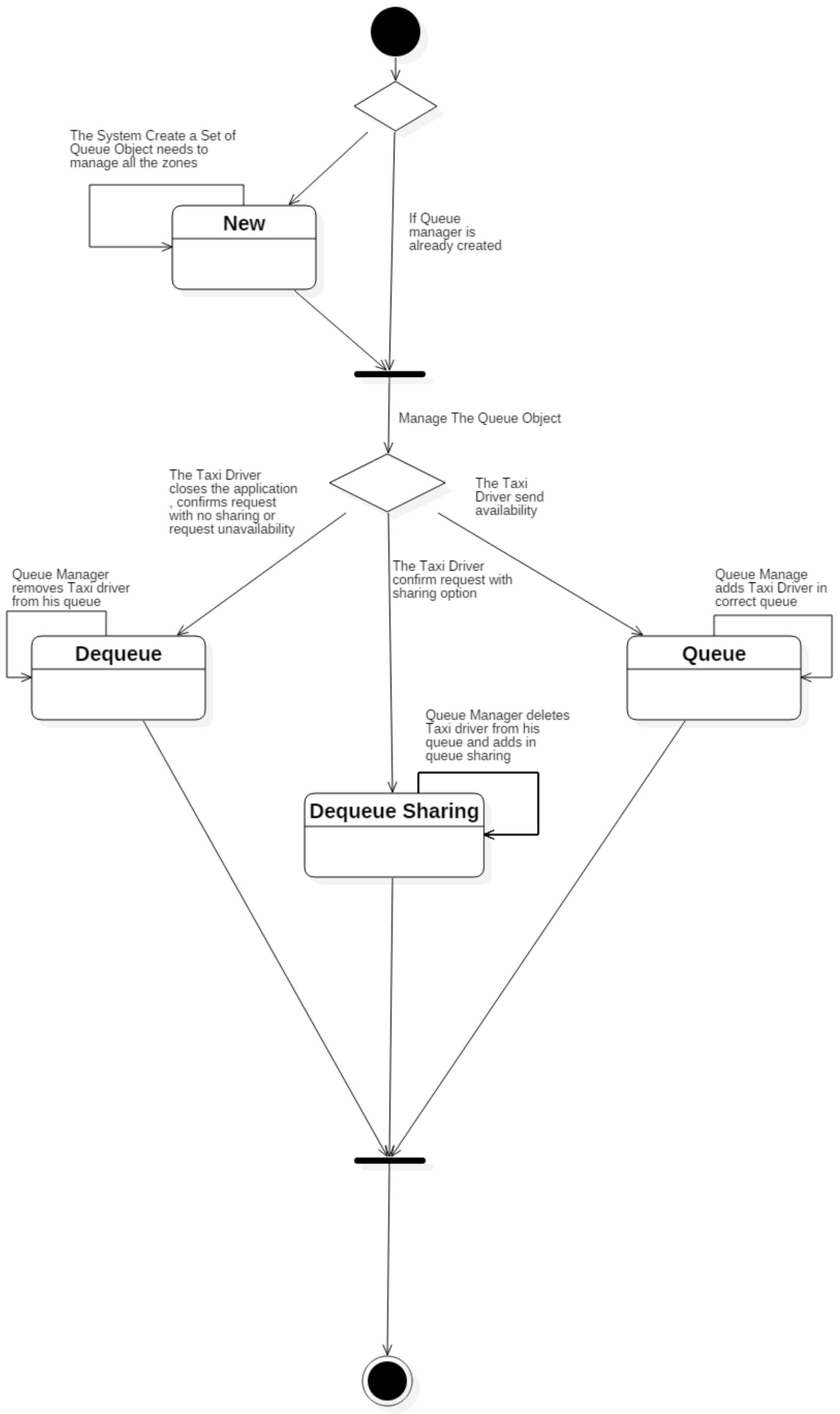
3.7.1 User


New
If User is alredy been create
The user changes his info
The taxi driver reports user
Modified info
Add negative report

3.7.2 Ride


The User sends a Taxi request and the Taxi Driver Confirms
New
Begin
Ride adds departure time and Customer as passenger
Car sharing
Ride adds other Customers as passenger
Add Passenger
Normal Request
starting time reached
Ride in Progress
Taxi Driver reaches the end point
Taxi Driver sends report problem
Ride sets end time
Ride Successfull
Ride sets Completed to false
Ride Failure

3.7.3 Queue Manager


The System Create a Set of Queue Object needs to manage all the zones
New
If Queue manager is already created
Manage The Queue Object
The Taxi Driver closes the application , confirms request with no sharing or request unavailability
The Taxi Driver send availability
The Taxi Driver confirm request with sharing option
Queue Manager removes Taxi driver from his queue
Dequeue
Queue Manage adds Taxi Driver in correct queue
Queue
Queue Manager deletes Taxi driver from his queue and adds in queue sharing
Dequeue Sharing

3.8 Performance requirements

- The system should support at least all the taxi driver registered into the Town Database
- The system should elaborate user incoming information
- The system should provide the faster path for each ride
- The system should calculate the final price of the ride with a maximum error of 10%
- The system should provide to a passenger the arrival time (maximum error 20%) and change the path in case of traffic

3.9 Logical database requirements

3.10 Design constraints

- GPS precision limitations: average 3m error
- Internet congestion

3.11 Standards compliance

3.12 Software system attributes

3.13 Reliability

3.14 Availability

3.15 Security

- SSL connection over the web
- Username and password to identify the user and taxi driver
- All the Operation are logged into the system
- The data stored inside the secondary store are encrypted
- The system databases has consistency check, data integrity check

3.16 Maintainability

3.17 Portability

3.18 Additional comments