

Course Title: System Analysis and Design

Course Code: IS 309

Course Instructor:

Project Title:

Online Ordering Cabs System



Online Ordering Cabs System

Table of Content

1. Introduction 4

2. Business Requirement 4

3. Use Case Diagram 7

4. Use Case Description 7

5. Activity Diagram 12

6. Sequence Diagram 14

7. Class Diagram 17

8. Behavior State Machine 18

9. Method Specification 20

10. User Interfaces 25

11. Deployment Diagram 28

1. Introduction

Online Ordering Cabs System (Riyadh Taxi) provides solutions to the disadvantages of ordinary cabs such as the cab availability in Non-main Street and in public places is poor. Also, we want to provide different functionality that ordinary cab does not have (e.g. the customer can determine the car model).

This system basically concerns about offering people in Riyadh city to book a cab and track it in real time. The customer first search for cabs. Then the system displays all cabs in a list so the customer can select preferred one based on time, car model and location. The customer can cancel or modify the reservation, choose the payments method (visa or cash), and the customer is able to evaluate the service (ex. the driver and time to get the cab). The customer must have an account to get the system's services.

The "Request taxi" button triggers the scanner, which finds partner cabs within certain radius of a location. The nearby cab drivers, equipped with smartphones, can accept a ride or reject it. Apart from accepting rides, he have to provide the customer with the approximate time of arrival to his/her desired destination using GPS system, and the driver can send arrival alerts.

The system administrator keeps track of available cabs and their locations and stores them in a database, assign cabs based on the customer's request to choose the suitable one from them, and maintaining the entire system. The system will be available twenty four seven.

We expects the number of customers would increase through meeting their needs.
Thus, the profits would increase.

2. Business Requirements

2.1 Functional Requirements

2.1.1 Administrator Requirements

- 1.1 The administrator shall be able to log in and out the system.
- 1.2 The administrator shall be able to access the database.
- 1.3 The administrator shall be able to access customer information.
- 1.4 The administrator shall be able to monitor the taxis.
- 1.5 The administrator shall be able to maintain the entire system.
- 1.6 The administrator shall be able to add new driver.
- 1.7 The administrator shall be able to update the list of available cabs.
- 1.8 The administrator shall be able to store all the cabs in a database.

2.1.2 Customer Requirements

- 2.1 The customer shall be able to log in and out the system.
- 2.2 The customer shall be able to edit his/her reservation.
- 2.3 The customer shall be able to contact the driver.
- 2.4 The customer shall be able to evaluate the service.
- 2.5 The customer shall be able to create an account.
- 2.6 The customer shall be able to retrieve their password.
- 2.7 The customer shall be able to search for cabs.
 - 2.7.1 The system shall be able to display to the customer the nearest available cabs.
- 2.8 The customer shall be able to pay for his reservation.
- 2.9 The customer shall be able to modify his cab reservation.
- 2.10 The customer shall be able to choose the payment method for his reservation.
- 2.11 The customer shall be able to cancel his cab reservation.
- 2.12 The customer shall be able to select a cab.

2.1.3 Driver Requirements

- 3.1 The Driver shall be able to log in and out the system.
- 3.2 The Driver shall be able to contact the customer.
- 3.3 The Driver shall be able to report under any emergency.
- 3.4 The system shall allow the driver to report the customer.
- 3.5 The system shall allow the driver to report the car.

- 3.6 The driver shall be able to accept/reject a ride.
- 3.7 The driver shall be able track the customer location.
- 3.8 The driver shall be able to send arrival alter to the user.
- 3.9 The driver shall be able to accept the change from customer of the request.
- 3.10 The driver should have a phone.

2.2 Non-Functional Requirements

1. Performance

- 2.1.1 The system shall increase the response time.
- 2.1.2 The system shall consider the throughput of the system.
- 2.1.3 The system shall be able to handle up to 10000 concurrent users when satisfying all their requirements and up to 25000 concurrent users with browsing capabilities.

2. Availability

- 2.2.1 The system shall be available twenty four seven.
- 2.2.2 The system shall not be unavailable more than 1 hour per 1000 hours of operation.

3. Testability

- 2.3.1 The system shall be easy to test.
- 2.3.2 The system shall be easy to report errors.
- 2.3.3 The delivered system shall include unit tests that ensure 100% branch coverage.

4. Portability

- 2.4.1 The system shall work on Android platform.
- 2.4.2 The system shall work on iOS platform.
- 2.4.3 The system shall work on Windows platform.
- 2.4.4 No more than 5% of the system implementation shall be specific to the operating system.
- 2.4.5 The meantime needed to replace the current Relational Database System with another Relational Database System shall not exceed 2 hours. No data loss should ensue.

5. Recoverability

- 2.5.1 The system should be recoverable.
- 2.5.2 The estimated loss of data in case of a disk crash shall be less than 0.01%.
- 2.5.3 The system shall backup every 24 hours.

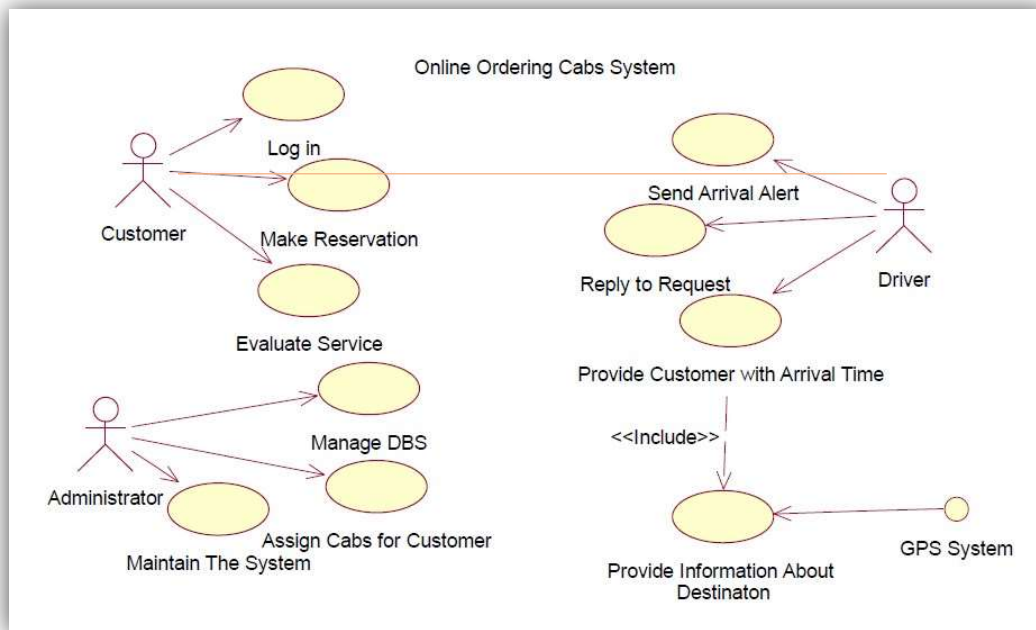
6. Maintainability

- 2.6.1 The system shall allow the functionalities changes.
- 2.6.2 The system should be easy to repairs.

- 2.6.3 Installation of a new version shall leave all database contents and all personal settings unchanged.
- 7. Reliability
 - 2.7.1 The system defect rate shall be less than 1 failure per 1000 hours of operation.
 - 2.7.2 No more than 1 per 1000000 transactions shall result in a failure requiring a system restart.
- 8. Usability
 - 2.8.1 Novice users shall perform tasks X and Y in 15 minutes.
 - 2.8.2 Experienced users shall perform tasks X and Y in 2 minutes.
 - 2.8.3 At least 80% of customers polled after a 3 months usage period shall rate their satisfaction with the system at 7 and more on a scale of 1 to 10.
- 9. Security
 - 2.9.1 The system shall identify all of its client before allowing them to use its capabilities.
 - 2.9.2 The system shall resistance to known attacks.
 - 2.9.3 The system shall encrypt the secure information.
 - 2.9.4 At least 99% of intrusions shall be detected within 10 seconds.

3. Use-case Diagram

This use case diagram describe online ordering cabs system, these are the primary function of the system for the Customer; the person who wants to order a cab through our software, the Administrator; the charged person of monitoring the system and other functions, and the Driver; the person in charge of serving customers by driving them from/to their destinations. GPS actor system: which is the main system for providing locations information.



4. Use-case description

1. Log in use case description

Use Case Name: Log in	ID:1	Importance Level: High
Primary Actor: Customer	Use Case Type: Detail, Essential	

Stakeholders and Interest:

Customer: Wants to enter the system to use its services.

Brief Description: This use case describes how the customer log into the system.

Trigger: Customer press "log in" button.

Type: External

Relationships:

Association: Customer

Include:

Extend:

Generalization:

Normal Flow of Events:

1. The Customer presses "Log in" button.

2. The Customer enters his email.
3. The Customer enters his password.
4. The Customer presses "Log in" button.

Sub Flow:

S-1 : Create account:

1. The Customer enters his email.
2. The Customer enters his full name.
3. The Customer enters his phone number.
4. The Customer create a secure password.
5. The Customer presses "Create" button.

S-2 : Update Account

1. The Customer modify his profile.
2. The Customer presses "Save" button.

Alternate/Exceptional Flows:

2. Make reservation use case description

Use Case Name: make reservation	ID:2	Importance Level: High
Primary Actor: Customer	Use Case Type: Detail, Essential	
Stakeholders and Interests:		

Customer : Wants to reservation taxi.

Brief Description: This use case describes how the customer are making reservation.

Trigger: login in the app.

Type: External

Relationships:

Association: Customer

Include:

Extend: Modify reservation, Cancel reservation

Generalization:

Normal Flow of Events:

1. The customer select the "login" button .
2. The customer enter his / her username and password.
3. The customer select the "Enter" button .
4. The system will validates the username and password.
5. The customer enter the model of car he wanted .

6. The customer enter his location.
7. The customer choose the way payment.
8. The customer select the "search" button .
9. The system display the nearest cabs based on his / her request .
10. The customer chooses the car he / she wanted .
11. The customer select the "confirm" button .
12. The system will validates the request .
13. The system send message to driver .

Sub Flows:

S-1 :Modify reservation :

1. The customer performs the S-2 cancel reservation.
2. The customer performs the new reservation .

S-2 :Cancel reservation :

1. The system view all the customer's reservation .
2. The customer selects the reservation he wants to cancel .

Alternate/Exceptional Flows:

4a.The username and password is invalid the login is rejected .

3. Evaluate Service use case

Use Case Name: Evaluate service	ID: 3	Importance Level: High
Primary Actor: Customer	Use Case Type: Detail, Essential	

Stakeholders and Interest:

Customer: wants to evaluate the application service.

Brief Description: This use case describes how customers will evaluate service.

Trigger: Customer experience the service.

Type: External

Relationships:

Association: Customer

Include:

Extend:

Generalization:

Normal Flow of Events:

1. The customer experience the service.
 2. The system send message to the customer ask him to evaluate service.
 3. The customer receive message from the system.
 4. The customer evaluate the service by answering some questions.
 5. The customer click "submit" button.
 6. The system accept the customer message.
-

Sub Flow:

Alternate/Exceptional Flows:

4. Reply to customer request use case

Use Case Name: Reply to Customer Request	ID 4:	Importance Level: High
Primary Actor: Driver	Use Case Type: Detail, Essential	

Stakeholders and Interests :

Driver – Wants to decide whether he wants to accept or reject a ride

Customer – Wants to get serve by a driver

Brief Description:This use case describes how the driver accept or reject the customer request

.

Trigger :Customer selects a cab to serve him

Type :External

Relationships :

Association:Driver **Include:**

Extend:

Generalization:

Normal Flow of Events :

1. The customer selects the suitable cab for his ride .
2. The customer verifies his request.

3. The system notify the selected cab.
4. The system send customer's location to the driver.
5. The driver accepts the ride.
6. The driver serve the customer.

Sub Flows :

Alternate/Exceptional Flows :

6a. The driver reject the ride. The customer select another cab .

5. Assign cabs for customer

Use Case Name: Assign Cabs For Customer	ID: 2	Importance Level: High
Primary Actor: Administrator	Use Case Type: Detail, Essential	

Stakeholders and Interest:

Administrator: Wants to assign a cabs for customer based on his/her needs

Customer: Wants to search for a cab after entering his/her preference

Brief Description: This use case describes how the administrator assign cabs for customer.

Trigger: Customer press "Search" button after entering his/her preference.

Type: External

Relationships:

Association: Administrator

Include:

Extend:

Generalization:

Normal Flow of Events:

1. The Customer press "Search" button after entering his/her preference.
2. The Administrator starts matching the customer's request with the available cabs in the cab database.
3. The Administrator displays the nearby matching cabs into the customer's device.
4. The Customer views the nearby matching cabs.

Sub Flow:

Alternate/Exceptional Flows:

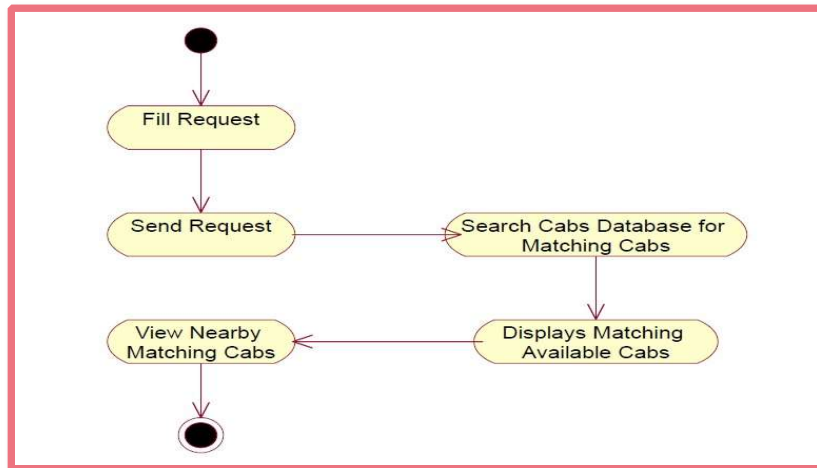
4a. There is no matching cabs, the Administrator send an apologizing message.

(Repeat 1 through 4)

5. Activity Diagram

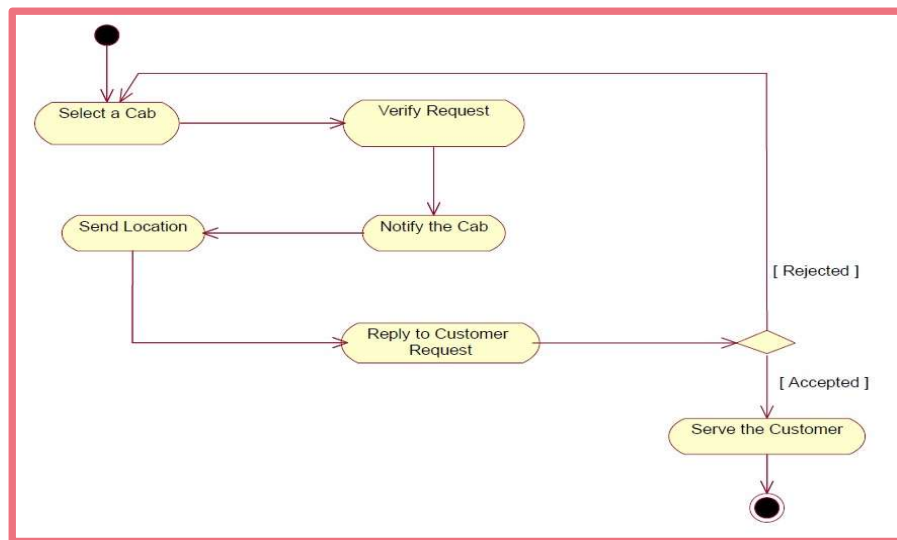
1. Assign Cabs for Customers

This diagram shows the sequence of activities for "assign cabs for customers" use case, which start after customer fills the request with his preference to make a cab reservation then send it by clicking the "search" button, then the cabs database starts to find a matching available cabs for the customer request, after finding some the administrator is responsible to display the matching cabs into the customer interface so he/she can view the available cabs that match his/her preference.



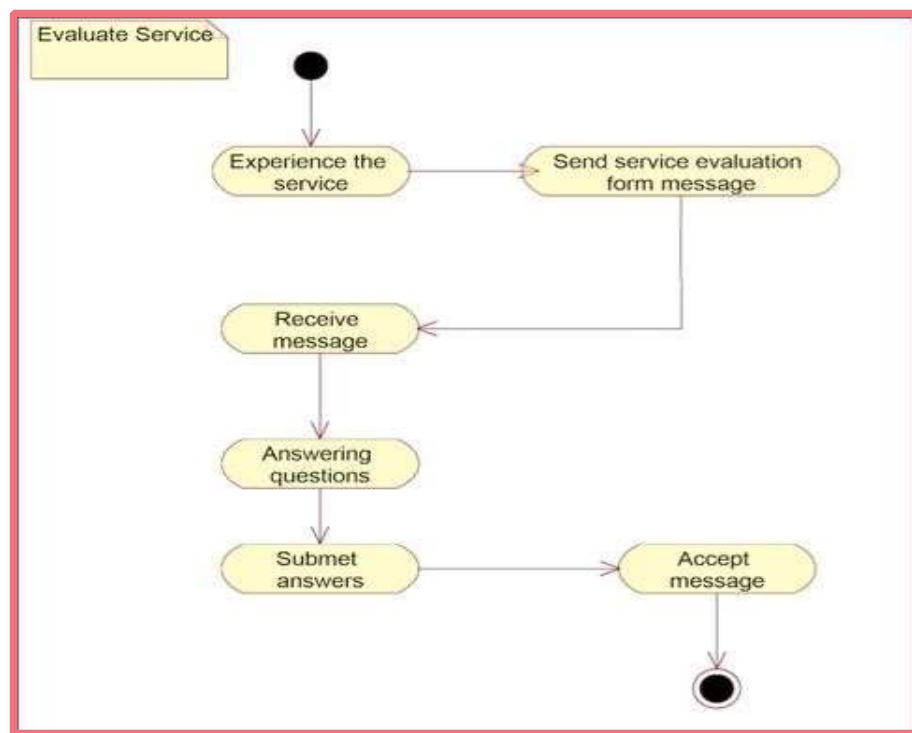
2. Replay to Customer Request

This activity diagram illustrates the sequence of activities for "reply to customer request" use case, which begins by customer selecting a suitable cab and verifying his/her request. Then, the selected cab will be notified for the customer request, and the customer location will be sent to the driver who drive the selected cab. After that, the driver will have to reply either by accepting or rejecting the ride. The customer will be served only if the driver accepts the ride. However, if the driver reject the ride, the customer will have to select another cab and the same sequence will be executed.



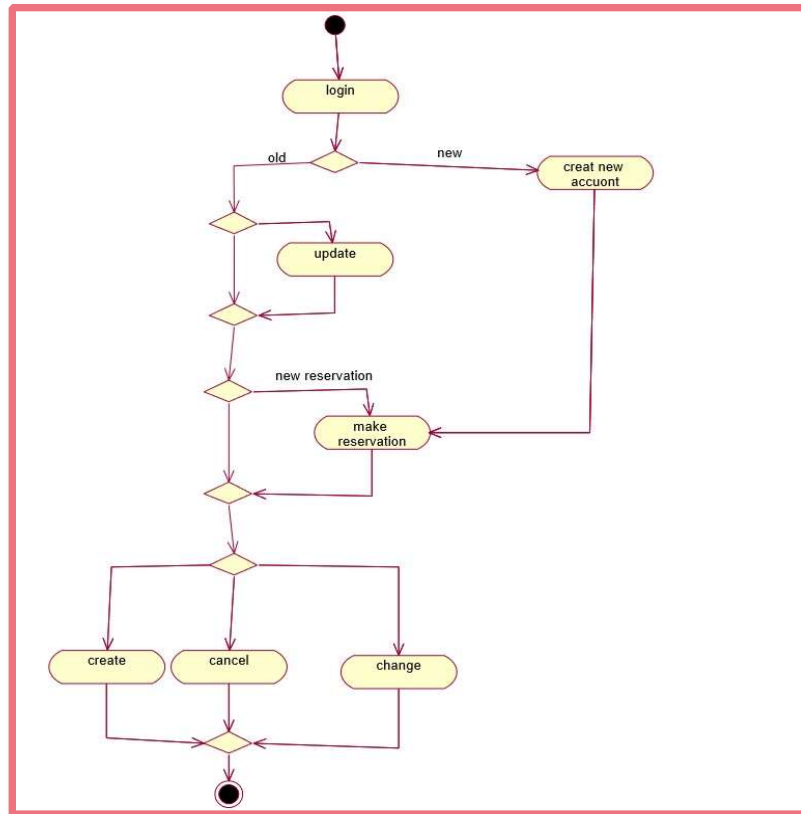
3. Evaluate Service

This diagram shows the sequence of activities for "Evaluate Service" use case, which start after customer experience the service, then the system send message to customer ask him to evaluate service, then the customer receive message from the system and answering question, then submit answer to system, then the system accept message.



4. Make Reservation

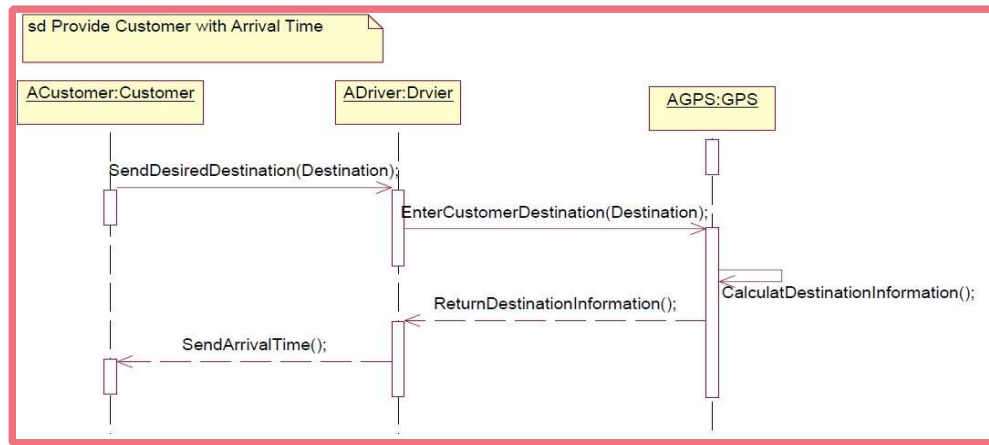
This diagram shows the sequence of activities for “make reservation” use case, which start by clicking “login” button, IF customer is new they will create a new account then you can make a reservation, if customer is old they can update her/his account or make reservation without update. Make reservation they have three selections, first selection customer create new reservation, second selection customer can cancel her/his reservation, third selection customer can modify her/his reservation



6. Sequence Diagram

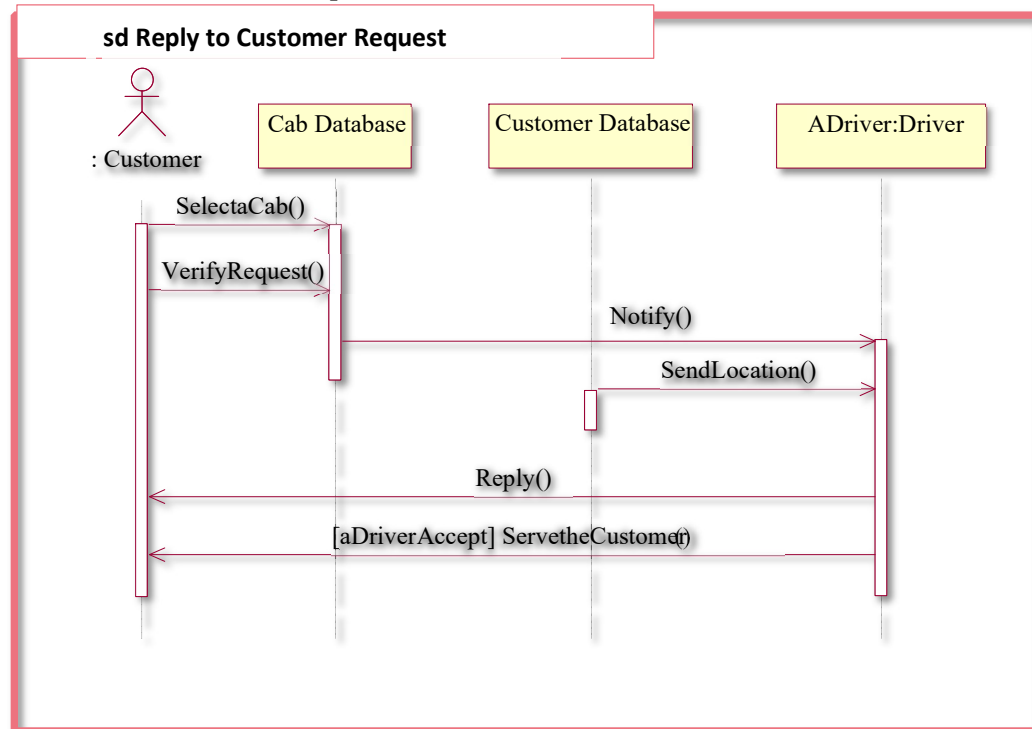
1. Provide Customer with Arrival Time

This diagram shows the sequence of messages for "provide customer with arrival time" use-case, first the customer will send his desired destination to the driver through the system, the driver will enter the destination into the GPS system he has in the car, then GPS system will calculate and collect information about the location, then display all the information to the driver, the driver then send the approximant time for arriving to the customer destination.



2. Reply to Customer Request

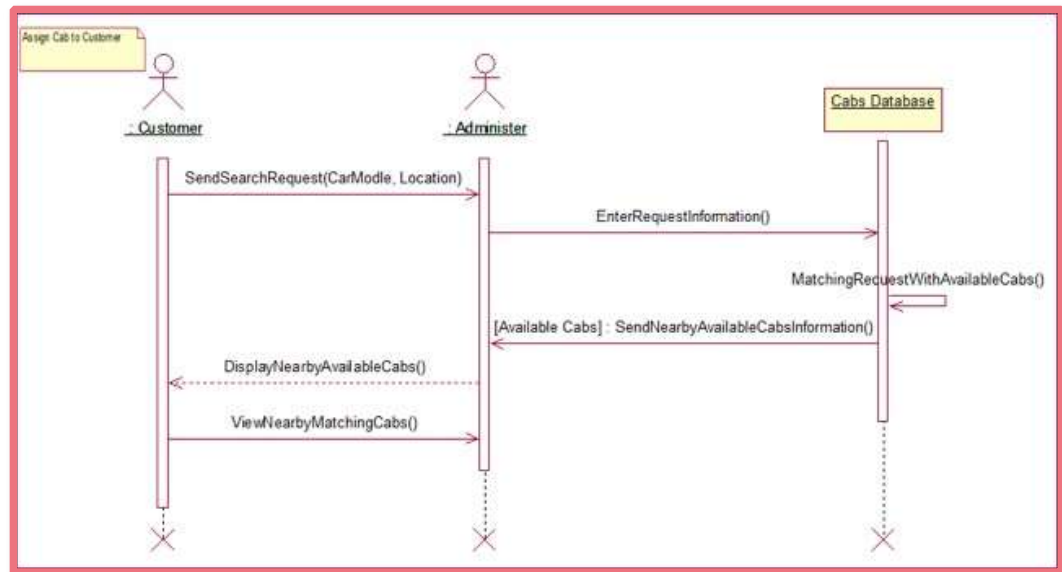
This sequence diagram illustrates the objects that participate in "Reply to customer request" use case, and shows the sequence of messages that pass between them. First, the customer will select a cab from the cab database, and verifies his/her request. Then, the cab database notifies the driver, and the location of the customer will be sent to the driver from the customer database. After that, the driver will have to reply to the customer. If he accepts the ride, he will then serve the customer.



3. Assign Cabs to Customer

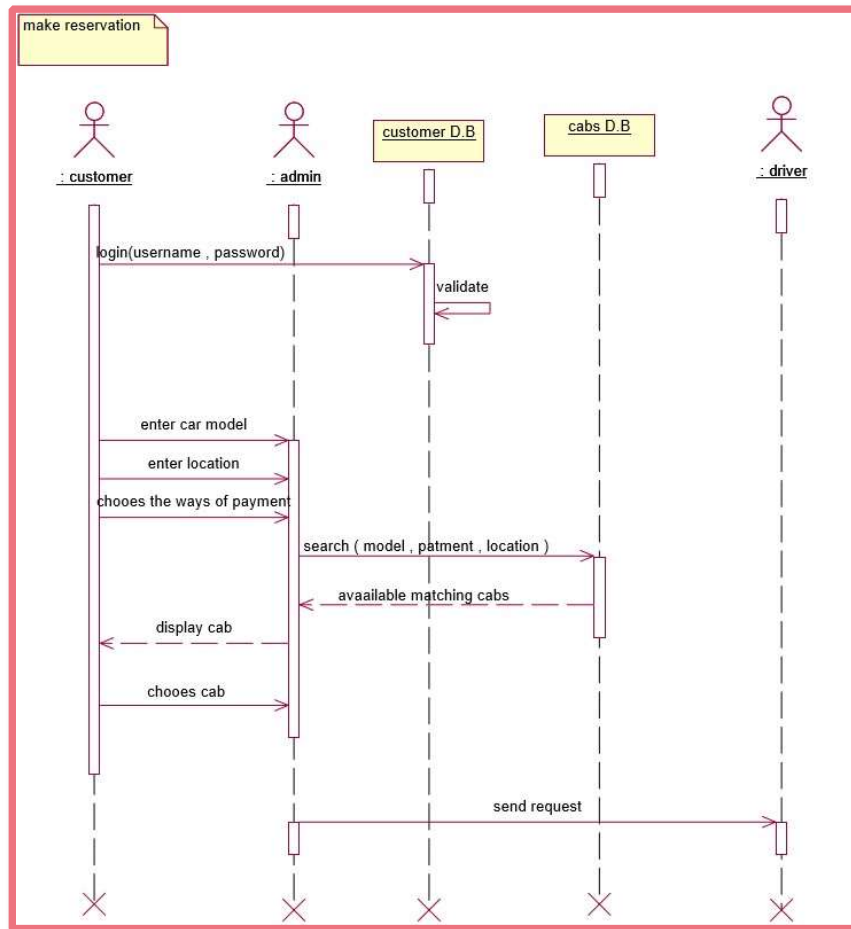
This diagram shows the sequence of messages for "Assign cab to customer" usecase, first the customer will send search cab request to administrate, then the administrate send request information to the cabs database, the cabs database start

matching request with the available cabs, when it has available cabs the cabs database send nearby available cabs information to administrate, the administer display nearby available cabs to customer, then the customer can view nearby matching cabs.



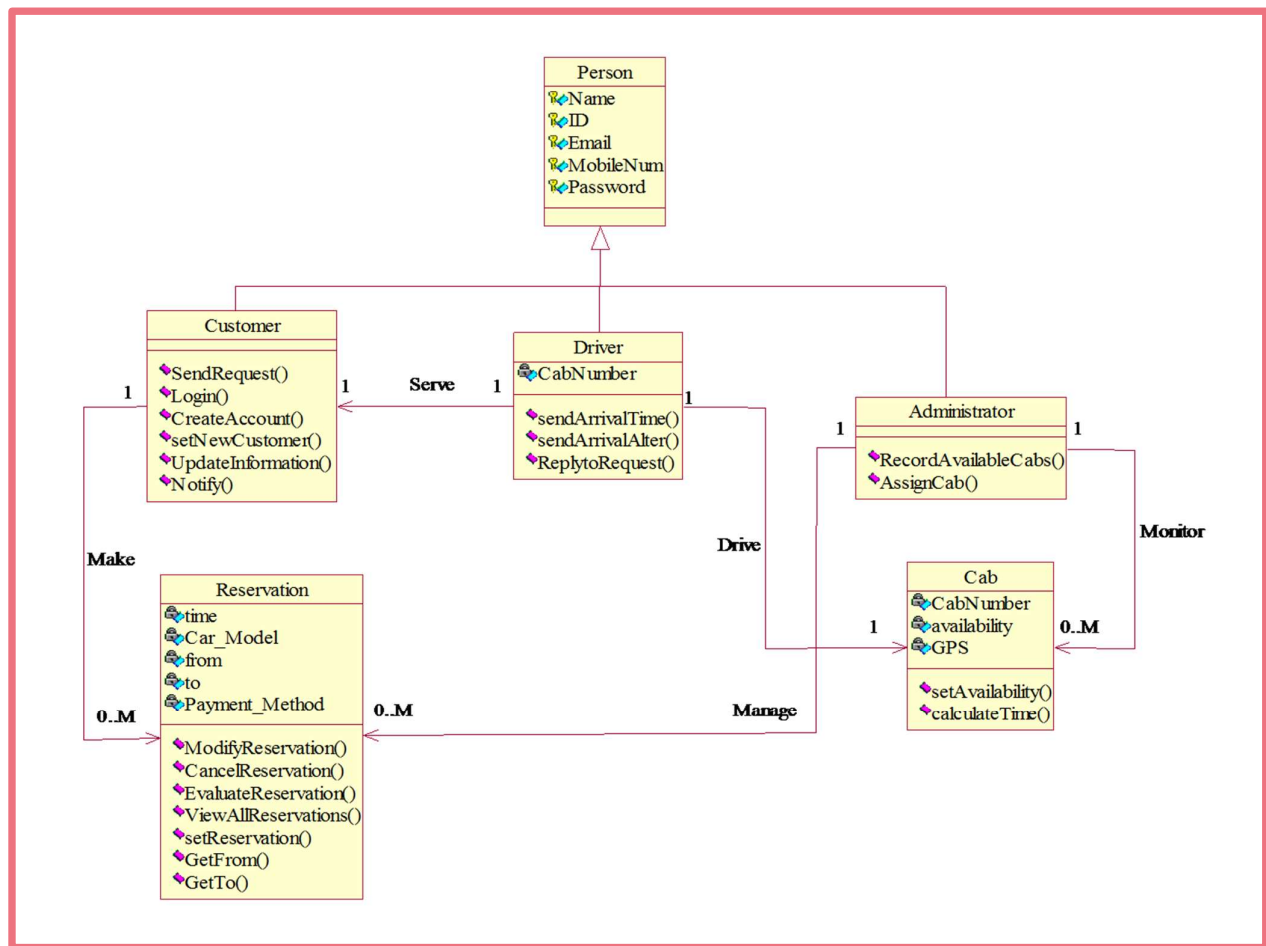
4. Make Reservation

This diagram shows the sequence of activities for “make reservation” use case, first the customer will login with username and password to customer D.B through the system, then customer D.B will validate her/his username and password, then customer will enter the car model, location, and choose the way of payment, then send to admin through the system, the admin will search customer requirement in cabs D.B, then cabs D.B will send the available matching cabs to admin, then the admin display nearest cab for customer, then the customer will send her/his cab choose to admin, then the admin will send the request to driver.



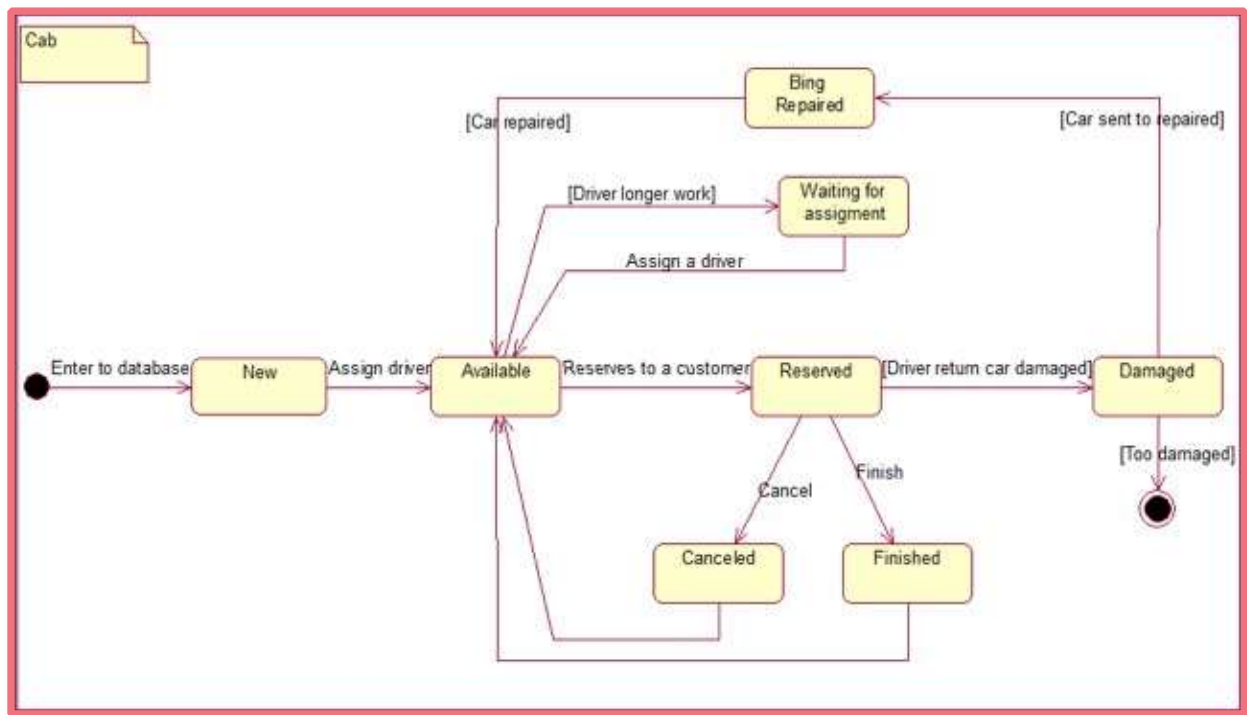
7. Class Diagram

This is the class diagram that shows classes and their methods and attributes, we ended up with 6 different classes that describe either a person, place or a thing on our system.



8.Behavior State Machine

Here is the behavior state machine for a cab in our system, a cab in online ordering system can go through different stages, which we try here to describe it. They are: New: when the cab car first been entered to the cab database. Available: when the cab been assign to a particular driver and have or been reserved yet. Idle: when its driver happen to resigns and it is waiting to be assigned to another driver. Reserved: when a cab is now reserved by a customer and it is not available in the meantime. Damaged: when a cab is damaged and cannot be reserved. Being repaired: when a damage cab can be repaired it will be sent to be fixed and it will be in Repair state till it is fixed then return to the system as available cab.



9. Method Specification

1. Log in:

Method Name: login()	Class Name: Customer	ID: 1
Contract ID: 133	Programmer: A.Taher	Date Due: 2/12/16
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input checked="" type="checkbox"/> C++ <input type="checkbox"/> Java		
Triggers/Events: 'Log in' button. Customer press 'l'		
Arguments Received: Data Type:	Notes:	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
Argument Returned: Data Type:	Notes:	
Algorithm Specification: Get Email Get Password IF FILE EXIST THEN READ Password FROM FILE IF FILE.Password == Entered Password THEN Login succesful ELSE PRINT "incorrect username or password" ELSE PRINT "incorrect username or password"		
Misc.Notes: None		

2. Record Available Cabs:

Method Name:	Class Name: Administrator	ID: 2
--------------	---------------------------	-------

RecordAvailableCabs()		
Contract ID: 123	Programmer: A.Taher	Date Due: 2/1/16
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input checked="" type="checkbox"/> C++ <input type="checkbox"/> Java		
Triggers/Events: Cab becomes available.		
Arguments Received: Data Type:	Notes:	
Cab	The cab that became free.	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
Cab.setAvailability()	Cab	
Argument Returned: Data Type:	Notes:	
Void		
Algorithm Specification: IF c is available THEN c.setAvailability(TRUE) AND Add c to list		
Misc.Notes: None		

3. Replay to Request:

Method Name: Replay to request ()	Class Name: Driver	ID: 3
Contract ID: 434	Programmer: S.George	Date Due: 2/1/16
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input checked="" type="checkbox"/> C++ <input type="checkbox"/> Java		
Triggers/Events: Administrator send request to driver		

Arguments Received: Data Type:	Notes:	
Time	Time that send from administrator	
Location	Location that send from administrator	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
Cab.Set.availabiliti()	Cab	
Argument Returned: Data Type:	Notes:	
True	If accept request	
False	If refusal request	
Algorithm Specification: IF time is OK THEN IF location is OK THEN RETURN TRUE THEN c.setAvailability(FALSE) ELES RETURN FALSE EXIT		
Misc.Notes: None		

4. Send Arrival Time:

Method Name: SendArrivalTime()	Class Name: Driver	ID: 4
Contract ID: 136	Programmer: A.Ahmed	Date Due: 2/12/2016
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input checked="" type="checkbox"/> C++ <input type="checkbox"/> Java		
Triggers/Events: Customer Send Desired Destination. 		

Arguments Received: Data Type:	Notes:	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
Reservation.GetFrom()	Reservation	
Reservation.GetTo()	Reservation	
Self.CalculateTime()	Cab	
Customer.Notify()	Customer	
Argument Returned: Data Type:	Notes:	
Time		
Algorithm Specification: 1.For all True Do 1. NewReservation.GetFrom(); 2.NewReservation.GetTo(); 3.Return 1. CalculateTime(From,To); 4.NewCustomer.Notify(Destination_information); 2..Exit		
Misc.Notes: None		

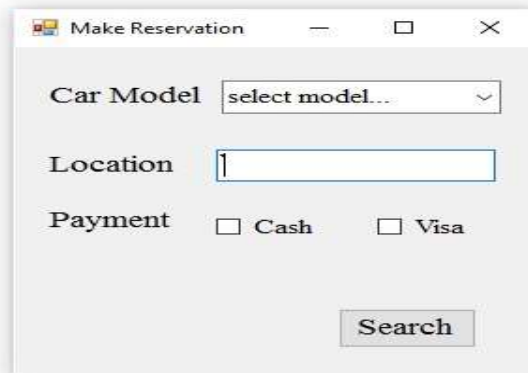
5. Send Request:

Method Name: SendRequest	Class Name: Customer	ID: 5
Contract ID: 15	Programmer: Y. Baker	Date Due: 5/1/16
Programming Language: <input type="checkbox"/> Visual Basic <input type="checkbox"/> Smalltalk <input checked="" type="checkbox"/> C++ <input type="checkbox"/> Java		
Triggers/Events: Customer send request to reserve a cab		

Arguments Received: Data Type:	Notes:	
Messages Sent & Arguments Passed: ClassName.MethodName:	Data Type:	Notes:
self.SetNewCustomer()	Customer	
self.UpdateInformation()	Customer	
Reservation.SetReservation()	Reservation	
Reservation.ModifyReservation()	Reservation	
Reservation.CancelReservation()	Reservation	
Arguments Returned: Data Type:	Notes:	
Void		
Algorithm Specification: Create new Reservation from Customer IF Customer Not in the Customer Database THEN SetNewCustomer(Name,ID, email,MobileNo>Password) IF Customer wants to Update Customer Information THEN UpdateInformation(Name,ID, email,MobileNo>Password) WHILE Customer wants to Make New Reservation NewReservation.SetReservation(CarModel,Location,PaymentMethod) IF Customer wants to Modify Reservation THEN NewReservation.ModifyReservation(CarModel,Location,PaymentMethod) IF Customer wants to cancel Reservation THEN NewReservation.CancelReservation(ReservationNo)		
Misc. Notes:		
None.		

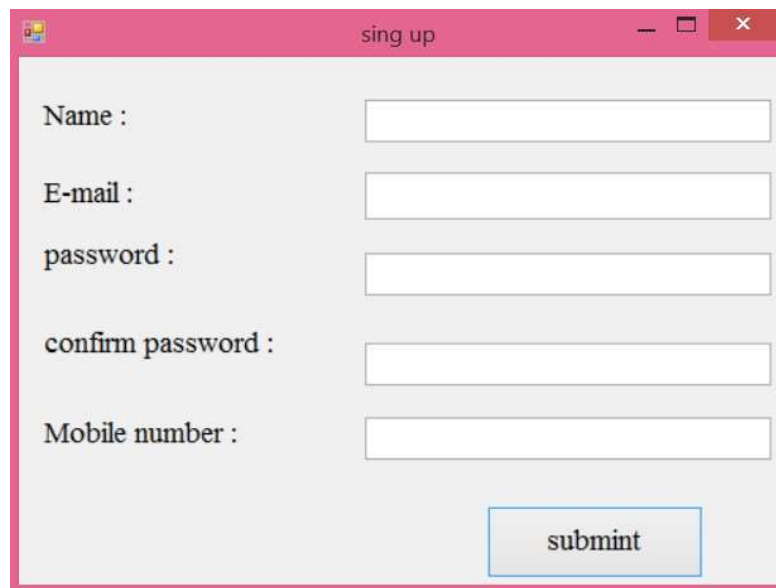
10. User Interfaces

1. Make Reservation:



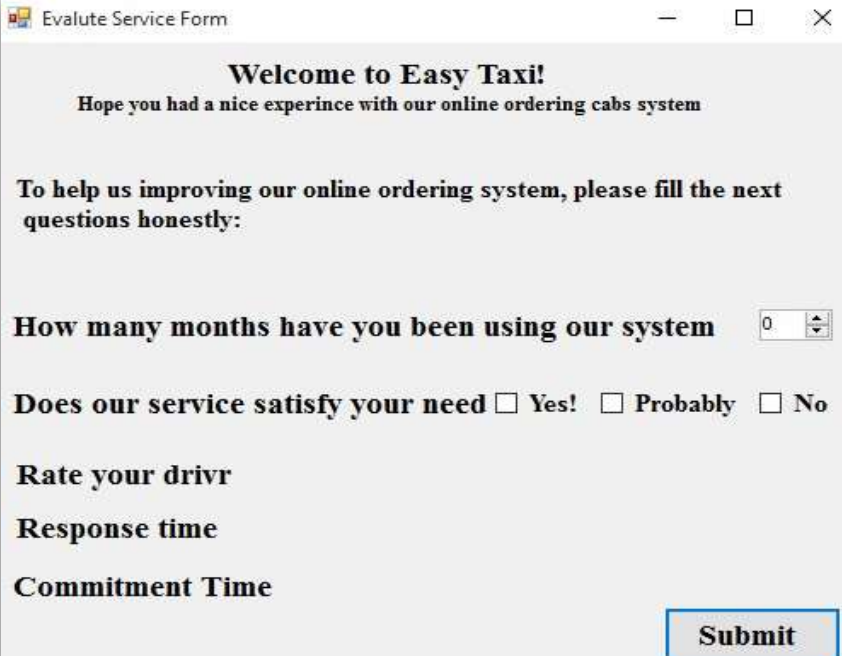
A screenshot of a 'Make Reservation' window. It features a title bar with a standard icon, a minus button, a maximize button, and a close button. The window contains three input fields: 'Car Model' with a dropdown menu showing 'select model...', 'Location' with a text box containing the letter 'l', and 'Payment' with two checkboxes labeled 'Cash' and 'Visa'. A 'Search' button is located at the bottom right.

2. Sign up:



A screenshot of a 'sing up' window. It has a pink title bar with a standard icon, a minus button, a maximize button, and a close button. The window contains five input fields: 'Name :', 'E-mail :', 'password :', 'confirm password :', and 'Mobile number :'. A 'submint' button is located at the bottom right.

3. Evaluate Service:



Welcome to Easy Taxi!
Hope you had a nice experince with our online ordering cabs system

To help us improving our online ordering system, please fill the next questions honestly:

How many months have you been using our system

Does our service satisfy your need ☐ Yes! ☐ Probably ☐ No

Rate your drivr

Response time

Commitment Time

Submit

4. Modify/Cancel Reservation:



You have two reservations:

Car Model: Black Ford 2013
Location: Daqlah Street 6811 - Riyadh 13315
Desired Location: Alsail Alkabeer Street 8799 - Riyadh 15589
Way of Payment: Cash

Modify **Cancel**

Car Model: Grey Hondai 2014
Location: Al Habary Street 7895 -Riyadh 12467
Desired Location: Al Saadah Street 4365 -Riyadh 17925
Way of Payment: Cash

Modify **Cancel**

5. Reply to Request:



11. Deployment Diagram

