Project Report for CICS520 and CICS507 – SuperRent System Design and Implementation

Prepared by

Ping Xiang
Sreenamitha Varthakavi
Dongsheng Shen
Nakisa Abbasi
Yu Yan
Manisha Sharma

April 17, 2015

Table of Contents

Chapter 1 Introduction

The purpose of this project is to develop an application for a Car Renting Company called SuperRent. Since the SuperRent is a typical car renting company, its major business is about renting its vehicles to customers, processing returned vehicles, managing reservations, processing customer registration and membership, etc. To fulfill these requirements, the application should be able to provide a framework that can organize and assist the activities performed by the company. Different menus and functions will be displayed to different users by identifying their login ID and password. Based on that, the users can work through the system with different interfaces; in this way, part of the work can be performed by the system. Considering that, the application can simplify the manual transactions performed by the Clerk or manager by automating some activities, which thereby reduces the complexity or improves time efficiency of the company's operation. For example, the clerk just needs to provide relative information, and the renting cost will be calculated automatically.

The system is designed for all branches of the SuperRent Company. It should support four kinds of users that are customer, clerk, manager and system administrator. Each type of user can access different menu and functions, so they can perform their own activities through the system. The customer can access the system through the workstations or PCs located in different SuperRent branches. Currently, web application is not considered yet, but it can be the next phase of the project. Major functions performed by this system are rent, return, payment, reservation, registration, membership, login, and so on. Even though the credit card payment is discussed in this project, the card verification system is not in the scope. Except for the major function requirements, some other requirements should be also considered. However, most of the non-functional requirements are still not solved yet. Based on this, this document is to identify the status of the project and get the first release of the software.

Chapter 2 Project Management

2.1 Agile Project Management

The management of this project is in the form of agile project management, and the Scrum approach is adopted. Considering this, we focused on the iterative development of the software rather than specific plans. Basically, there are three phases for this project. First, a general outline plan is done through several kick-off meetings. With these meetings, a project backlog of the requirements identification process is done from the stakeholders. During this process, a requirements analysis document is done to provide specific requirements as well as functions specification. Since the application is not quite complicated, the requirements are classified into several categories based on the stakeholders.

Then, it takes several sprint planning meetings to achieve some general agreements on the basic design of the entire application such as the interface and other prioritized features. Hence, the features identified are distributed to the team. Team members communicate through short daily Scrum meetings to share information, identify problems, and describe their progresses. Ping Xiang serves as the Scrum master to arrange the meeting and provide help to other team members. Since, the specific requirements are illustrated specifically in the requirements document as well as the functions. Issues for each sprint cycle are identified for each team member. Finally, the sprint review meeting is undertaken when all members give their deliverables. A final report document is written to identify the current status of the project.

Since all team members have visibility of the whole project, efficient team communication can be achieved easily. Team members can work on some common functions or methods through communication. Moreover, new requirements can be identified and added to the development process.

2.2 Milestones

Most of the milestones are identified during the sprint planning meetings at the beginning. However, the planned dates are not identified at that time because they are identified through the communication among the team members. Table

1 is a specification of the major milestones. These are achieved through the communication of the team members.

Milestones	Description	Milestone Criteria	Planned Date
M0	Start Project	Team construction	Jan 20, 2015
M1	Start Planning		Jan 25th, 2015
	Project scope and specification	ER Diagram, User case diagram	Feb 10th, 2015
	Requirements and functions specification	Requirements analysis document	
M2	Start Execution		Feb 17th, 2015
	Product separation into manageable parts	Job assigned team members	Feb 20th, 2015
	System Architecture and configuration	Tools and development environment agreement	Feb 22th, 2015
	Database construction	Tables creation and samples insertion	Feb 24th, 2015
	Major functions development	Coding of functionalities	April 2nd, 2015
M3	Confirm Execution		April 3nd, 2015
	System test	Coding of new functionalities, final database tables agreement and draft documentations	April 12nd, 2015
M4	Release Product	Final project report review and Demo presentation	April 17nd, 2015
M5	Close Project		April 18nd, 2015

Table 1 Milestones of the project

Chapter 3 Refined Requirements Analysis

3.1 Modified Use Case Diagrams

Since major functions have been modified through the communication among team members, the use case diagrams are quite different from those before. This section is to show some updates to the use case diagrams. Currently, 12 use causes are identified in this project including both top-level and sub-level cases. Figure 1 shows the top-level use case, and major functions as well as the actors are included in the diagram. Another example is a sub-level use case shown in Figure 2 for returning vehicle. The top-level returnVehicle function is specified into some detailed functions including calculate cost, process payment, handle membership and so on. Some of them are not considered in the previous requirements analysis document and are refined by the team.

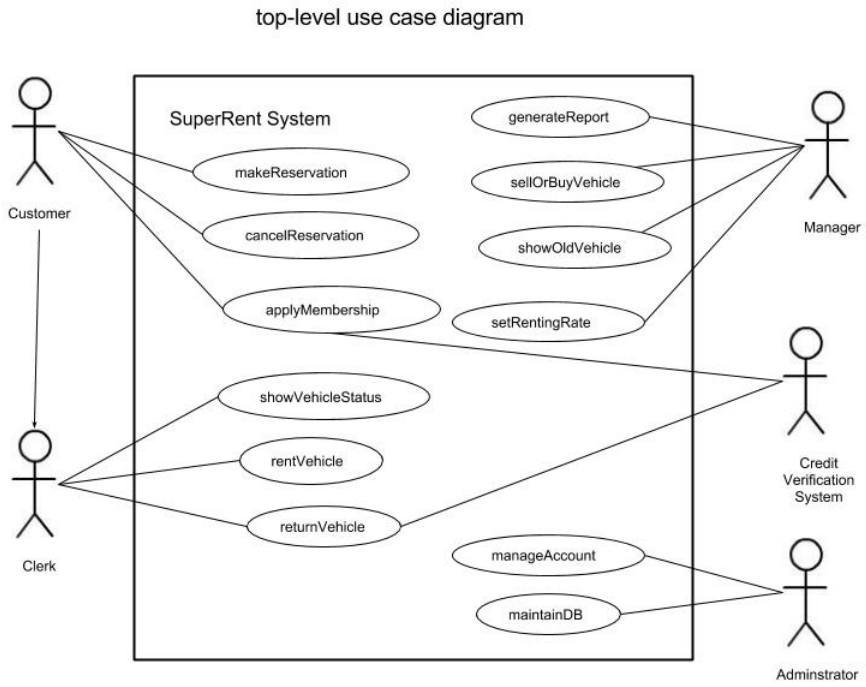


Figure 1 Top-level Use Case Diagram

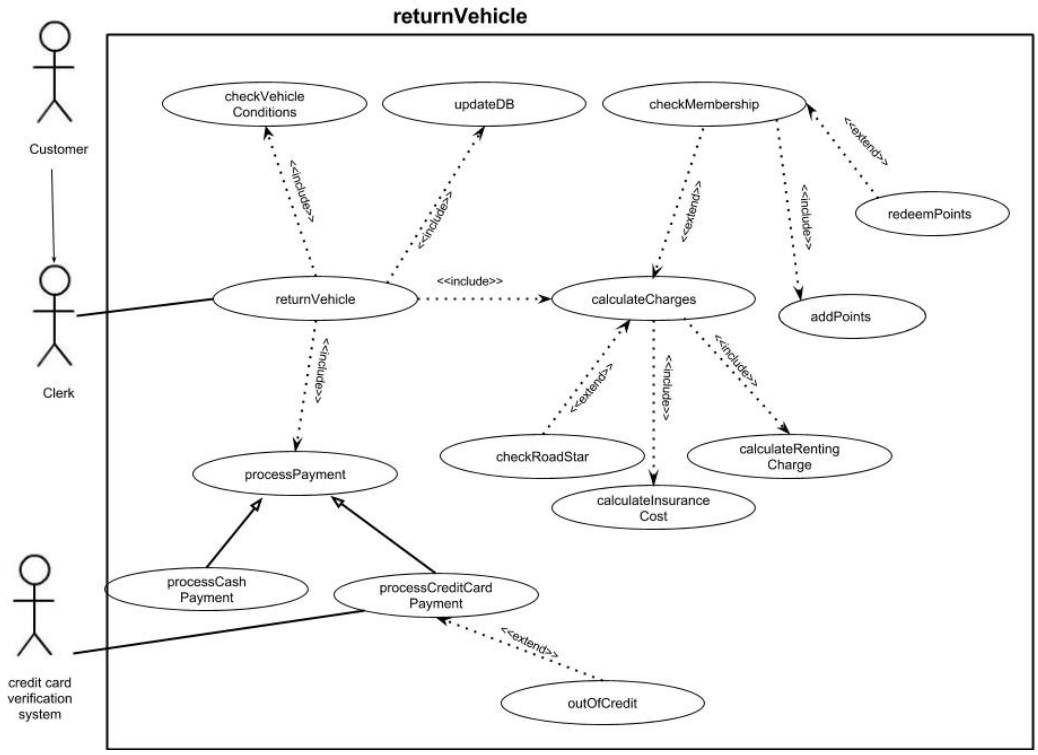


Figure 2 Sub-level Use Case Diagram for return vehicle

3.2 Refined Functional Requirements

For the functional requirements, the major functions are previously identified in the Requirements Analysis Documents. The specifications for each major function is also illustrated including entry condition, exit condition, exceptions and the flows of events. These are actually the major tasks for the team to fulfill by coding. Not all functions in the application are perfectly done since this is the first release of the application. Hence, by refining the requirements, better performance can be achieved.

For customer, previously identified functions are login as a customer, manage reservation, make a payment, apply for the Club membership, and check for points accumulated. By communicating with the users, the functions are further refined to make/check/cancel reservations, pay for or cancel membership, and edit profile information. Customer can also query the database for available vehicles to make the reservation. Some details are that the customer can get an estimation of the rental cost by identifying vehicle and time period.

For clerk, previously defined functions are login as a clerk, rent vehicles for customers, process the return of the rented vehicles, process payments of customers, and check the status of vehicles (availability, overdue). By further refinement, the major functions are classified into four categories that are reserve, rent, return and search vehicles. Functions such as payment and get status of vehicle are embedded in the four categories.

For manager, the previously defined functions are login as a manager, query the database for old vehicles, sell old vehicles and buy new vehicles, and set rates and costs. These functions are actually embedded in four categories that are manage vehicle, find vehicle, set price and generate reports by further refining the requirements with the users. While, for administrator, the previous functions such as add/remove/edit user information are done, but new functions

can be added since admin can access and manipulate the database directly such as modify the quantity of some equipment or add new columns to some tables.

3.3 Unachieved Non-Functional Requirements

The team also identifies major non-functional requirements; however, some of them are still not achieved yet. Future work should include these considerations into coding. The following are some examples of the non-functional requirements that we need to test or code in future. These are not included in the test design of this document because we only test the major functions in the application currently.

Usability

- The Clerk and Manager should be able to use the system with 4-hours of training.
- The Customer should be able to manage his/her reservation without any training.
- The system should also provide a user manual for different users.

Reliability

- The system should maintain all the customer information in encrypted form and should not keep the credit card details of the customer once he/she has returned the vehicle.
- The system shall automatically logout all customers after a period of inactivity.
- The system shall not leave any cookies on the customer's computer containing the user's password.
- The system should not lose data in case of power failure.

Performance

- The system retrieves data in less than a minute.
- The system should handle 1000 users at the same time.

Chapter 4 Design

4.1 Database Design

The entity-relational diagram below represents one single ER diagram; however, it is too large so we have to separate it into 4 parts that are displayed in the following four figures. Based on the design, 18 tables are added to the database.

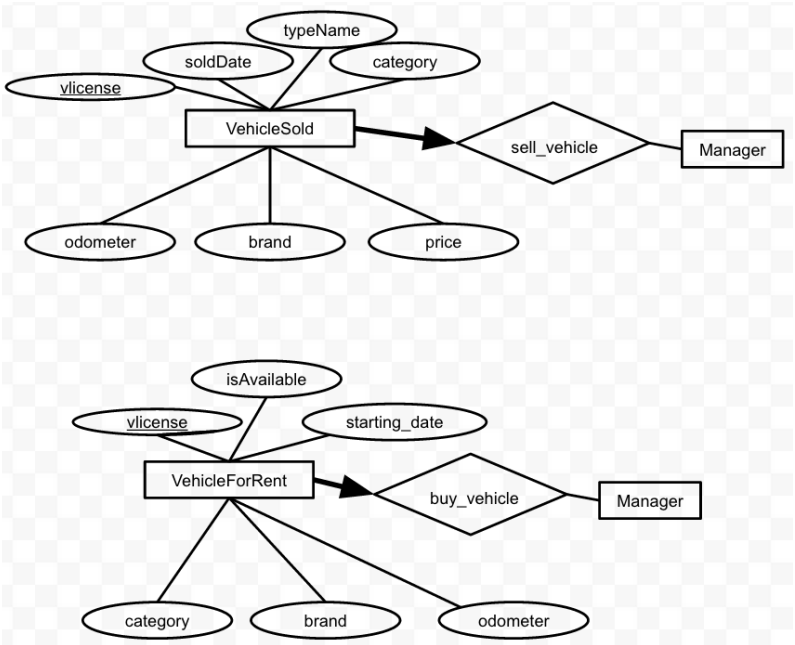


Figure 3 ER Diagram Part1

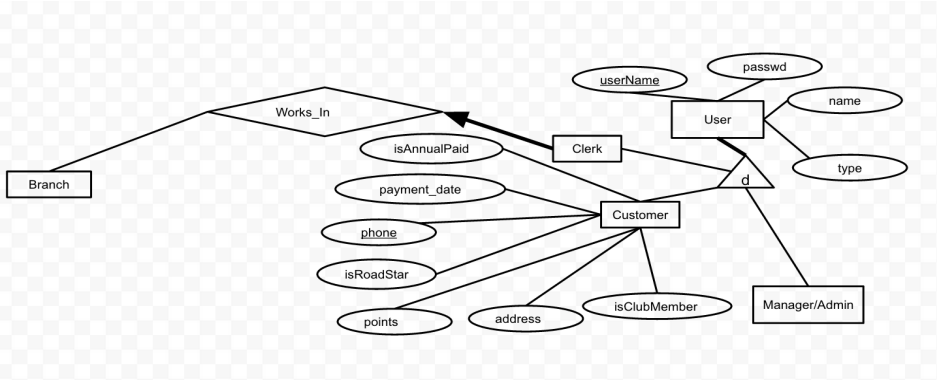


Figure 4 ER Diagram Part2

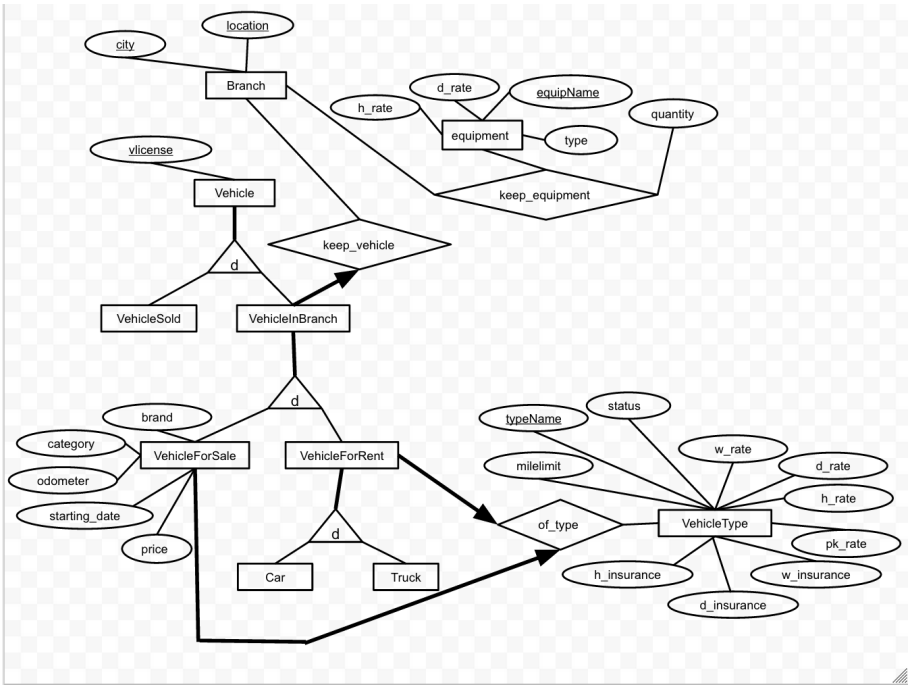


Figure 5 ER Diagram Part3

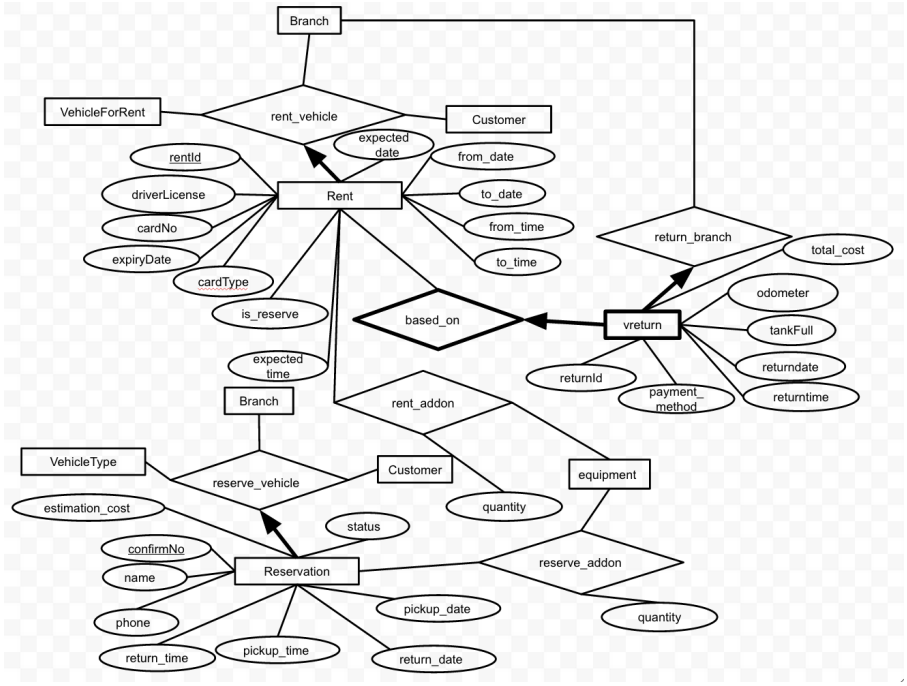


Figure 6 ER Diagram Part4

Based on the ER diagrams, the schemas for the tables are obtained as follows:

VehicleType (typeName, w_rate, d_rate, h_rate, pk_rate, w_insurance, d_insurance, h_insurance, milelimit)

Branch (city, location)

User(username, password, name, type)

Constraints: 1. Type can be clerk, customer or manager.

2. Each user has at least one type.

Customer (username, phone, address, IsRoadStar, isClubMember, point, IsAnnualPaid, payment_date)

Clerk (username, branch_city, branch_location)

Constraints: Every clerk works in exactly one branch

Manager (username)

Admin (username)

Equipment (equipName, type, d_rate, h_rate)

Vehiclesold (vlicense, price, solddate, typeName, category, brand, manager, odometer)

Constraints: Every vehiclesold has exactly one related manager.

VehicleForRent (vlicense, isAvailable, starting_date, category, vehicleType, brand, manager, type)

Constraints: 1. Every VehicleForRent has at least one related vehicleType.

2. Type can be car or truck.

3. Every VehicleForRent has exactly one related manager.

VehicleForSale (vlicense, price, starting_date, category, brand, VehicleType, odometer)

Constraints: Every VehicleForSale has exactly one related VehicleType.

Vehicleinbranch (vlicense, city, location, type)

Constraints: 1.Every vehiclebranch keeps vehicles in exactly one related branch.

2. Type can be VehicleForSale or VehicleForRent.

3. Each vehicleinbranch has at least one type.

Keep_equipment (equipName, city, location, quantity)

Reservation (confirmation_number, pickup_date, pickup_time, return_date, return_time, estimation_cost, branch_city, branch_location, customer_username, status, vehicleType)

Constraints: Every reservation has exactly one related branch, one related customer, and one related vehicle type.

Rent (rentid, driver_license, vlicense, branch_city, branch_location, customer_username, card_type, card_no, expiry_date, from_date, from_time, expected_return_date, expected_return_time)

Constraints: Every rent has exactly one related branch, one related customer, and one related vehicle.

Vreturn (returnid, rent_id, return_date, return_time, branch_city, branch_location, tank_full, odometer, total_cost, payment_method)

Constraints: 1. Every vreturn has exactly one branch to return vehicle.

2. Every vreturn has exactly one related rent.

Rent_addon (rentid, quantity, equipName)

Reserve_addon (confirmNo, quantity, equipName)

4.2 Application Design

UML diagrams are used to design the entire system including class diagram, state machine diagram, sequence diagram, components diagram, and activity diagram. These diagrams are important tools to show the application design of the project.

Class diagram:

Below are the class diagrams (Figure 7 to Figure 13). The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed.

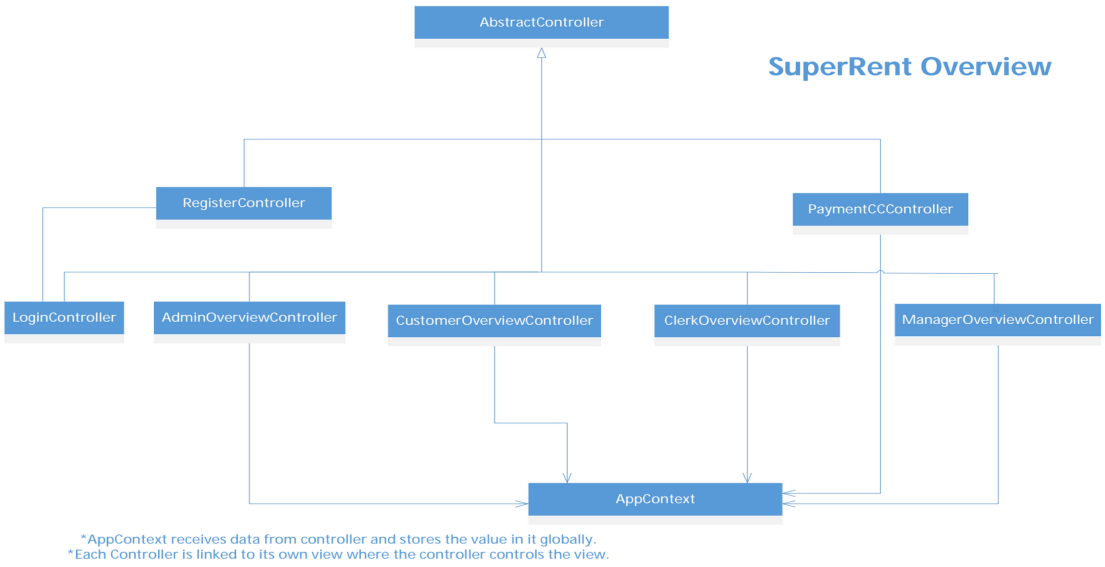


Figure 7 Class Diagram - SuperRent Overview

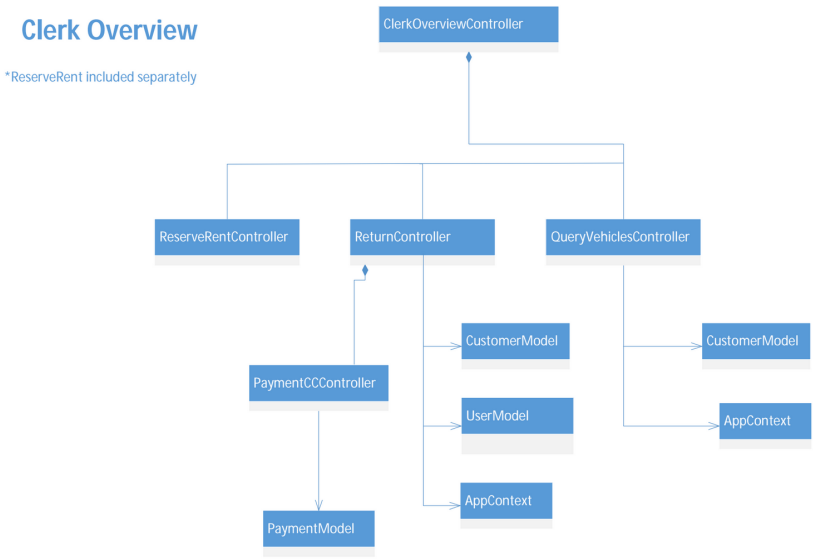


Figure 8 Class Diagram - Clerk Overview

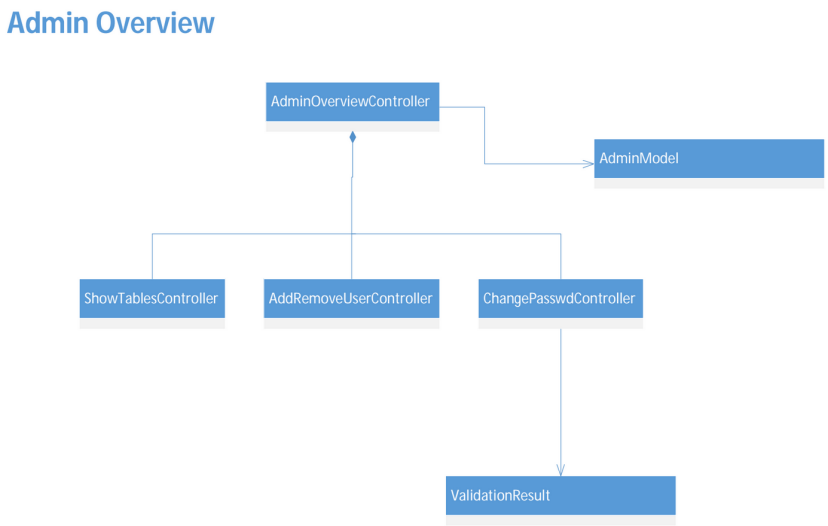


Figure 9 Class Diagram - Admin Overview

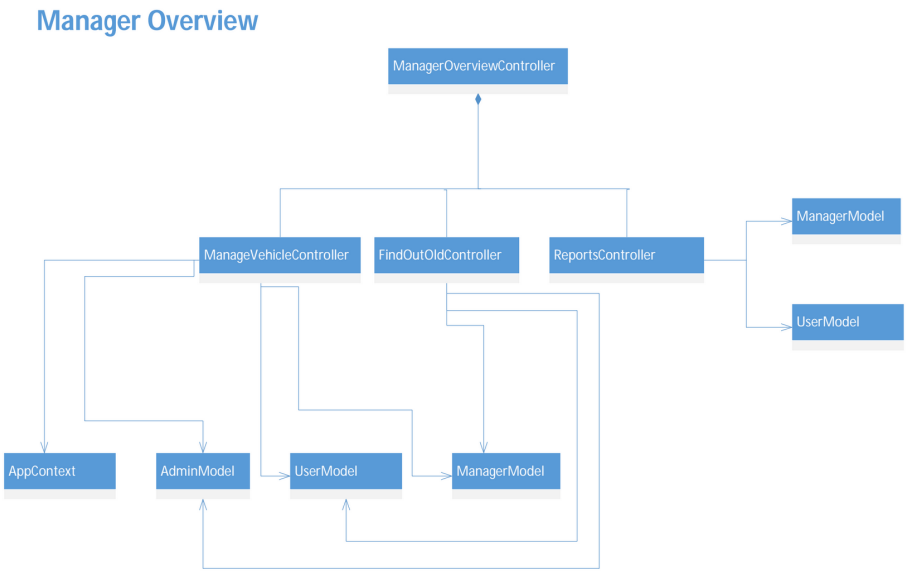


Figure 10 Class Diagram - Manager Overview

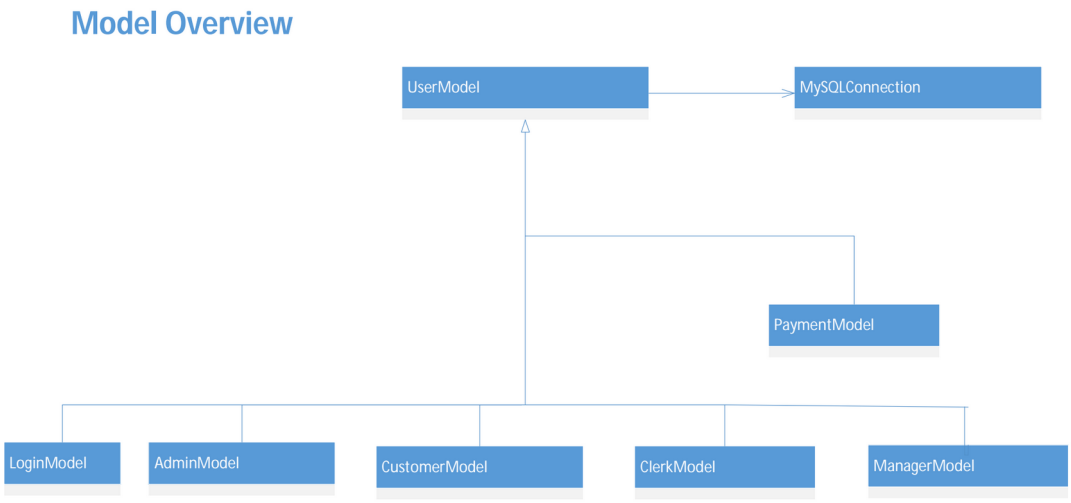


Figure 11 Class Diagram - Model Overview

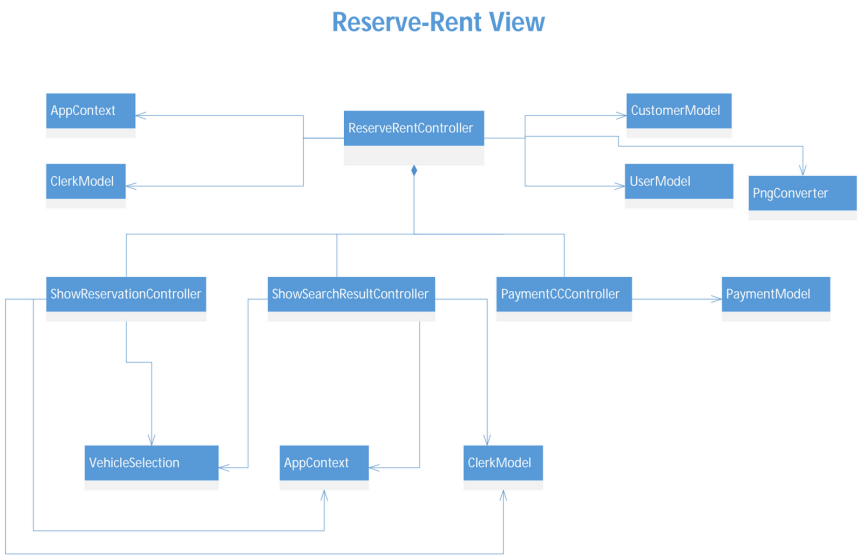


Figure 12 Class Diagram - Reserve Rent Overview

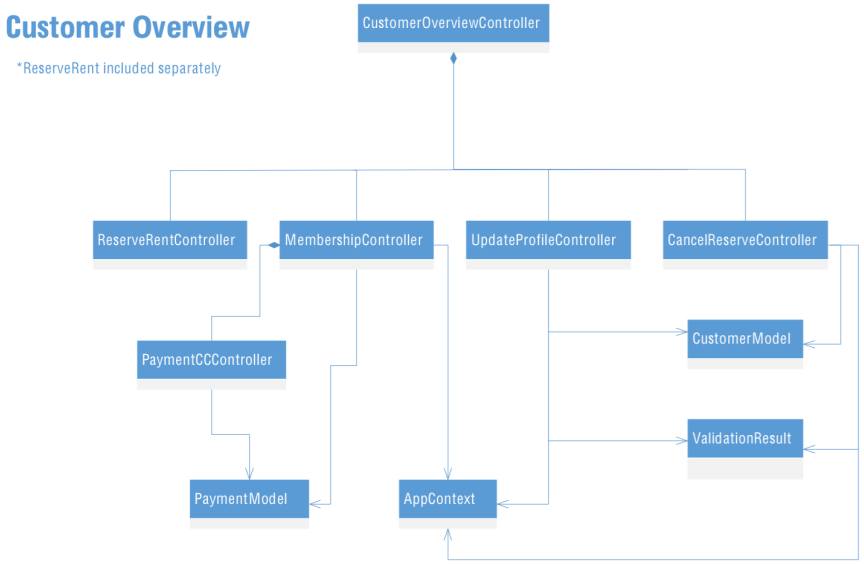


Figure 13 Class Diagram - Customer Overview

Sequence diagram:

In Figure 14, we use UML sequence diagrams to model the flow of logic within our system in a visual manner, enabling both document validate our logic, and are commonly used for both analysis and design purposes.The diagram shows the rent and reservation process. From the message arrow we can logically find the relation between the four objects. The clerk sends different messages to ReserveRent View, and then the view sends related message to ReserveRentController. The Controller gives some feedback messages to view and also sends main message to UserModel and gets some feedback. In the whole process, the alternative and option means the internal logical relation in the diagram. In our design, the opt means the choices the clerk should make. The different choices mean the different logic flow in the diagram.

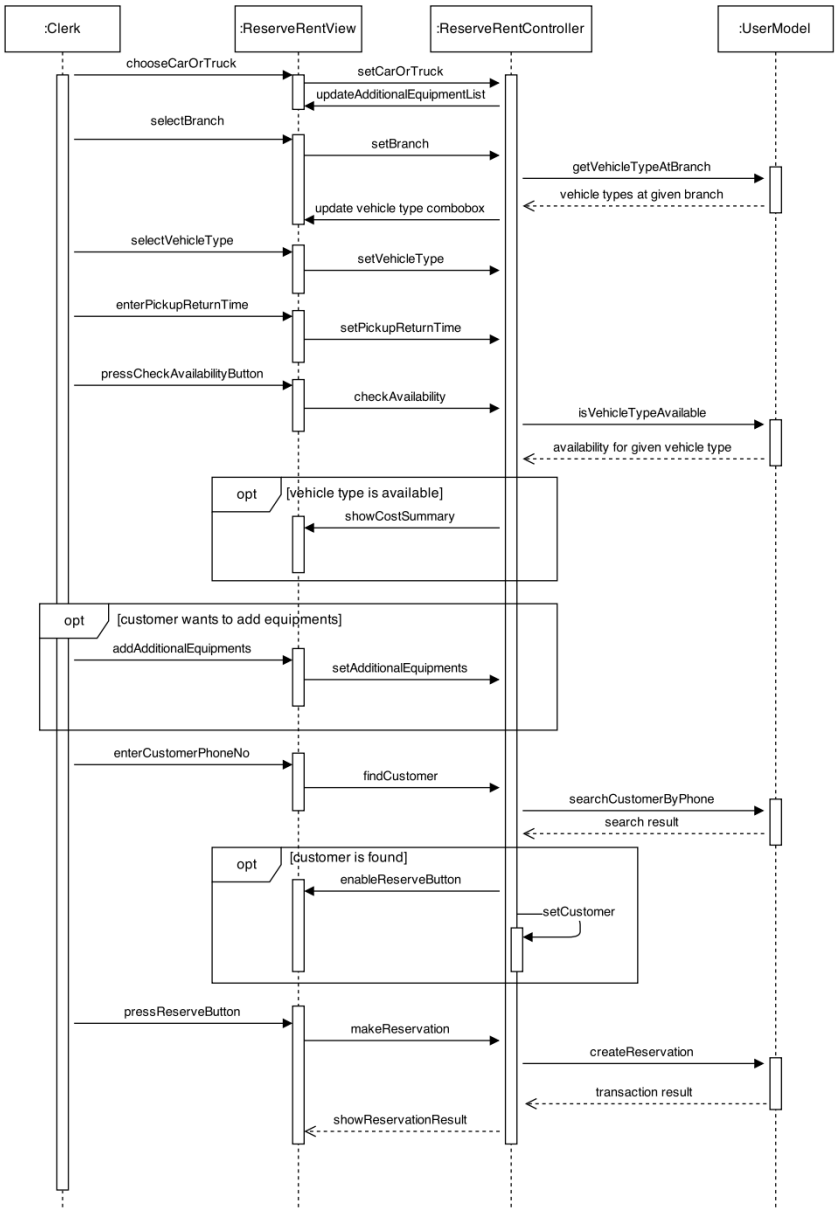


Figure 14 Sequence Diagram

Component diagram:

Component diagrams illustrate the pieces of software, embedded controllers, etc., which will make up a system. A component diagram has a higher level of abstraction than a Class Diagram, so a component is implemented by one or more classes (or objects) at runtime. They are building blocks so a component can eventually encompass a large portion of a system. The diagram shows the top level view of 5 components in our design: View , Controller, Model, DataBase, UTIL. From Figure 15, it can be seen that all of them have services exchange by provided interface and required interface.

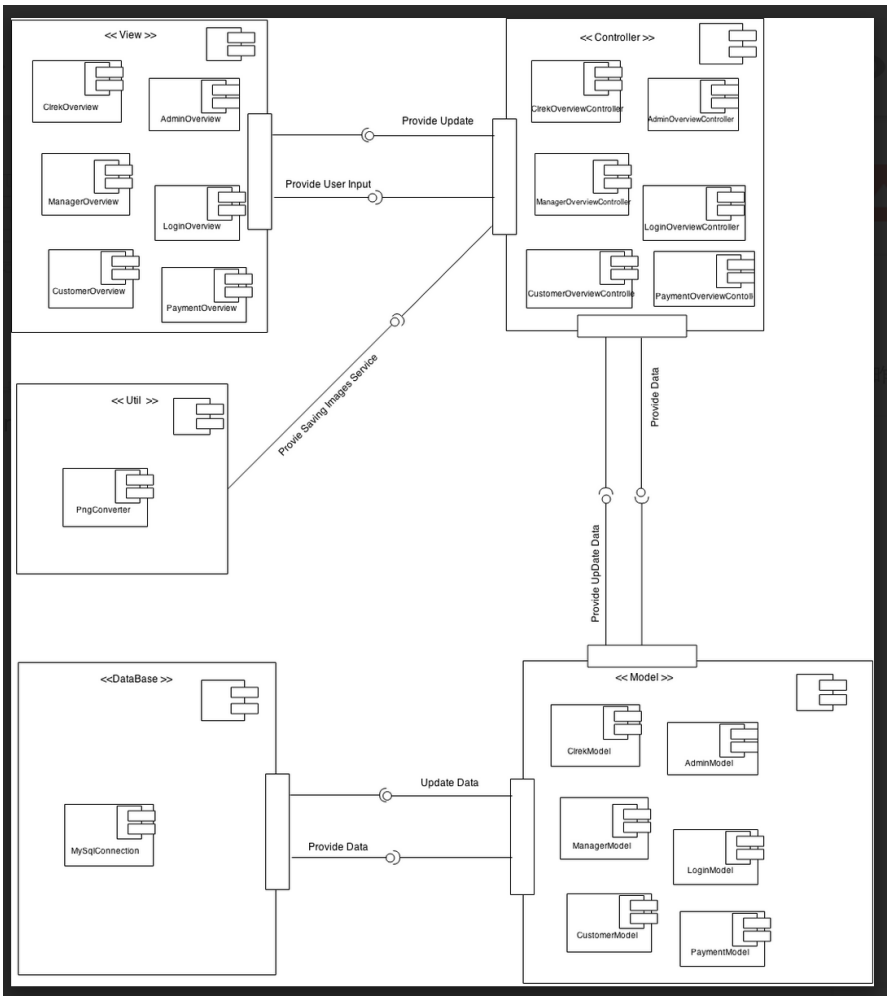


Figure 15 Components Diagram

State-machine diagram:

State-machine diagram specifyies the sequence of events that an object goes through during its lifetime in response to events. This diagram indicates the

transition between log in and log out of four different kinds of users in our design that is shown in Figure 16.

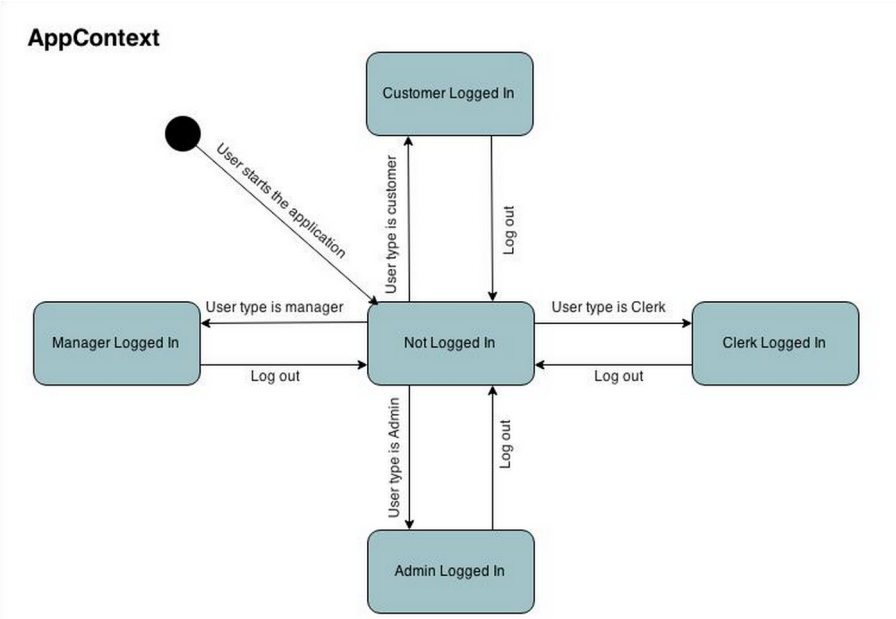


Figure 16 State Machine Diagram

Deployment diagram:

The Deployment Diagram in Figure 17 helps to model the physical aspect of an Object-Oriented software system. It models the run-time configuration in a static view and visualizes the distribution of components in an application. In

most cases, it involves modeling the hardware configurations together with the software components that lived on. In our design, this diagram shows the relationship in hardware level. User Interface Layer, Business Logic Layer, and Data Access Layer are in the Application Server. The Application Server is also in SuperRent. The SuperRent connects with PC and DB Sever. In DB Sever,we use our own MSS DB.

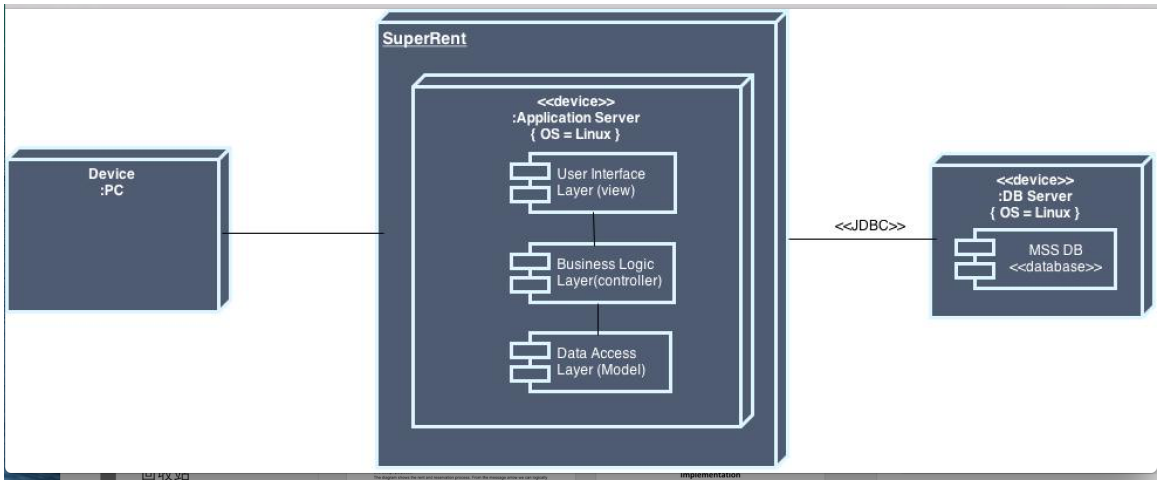


Figure 17 Deployment diagram

Activity diagram:

Activity diagrams in Figure 18 show the workflow from a start point to the finish point detailing the many decision paths that exist in the progression of events contained in the activity. They may be used to detail situations where parallel processing may occur in the execution of some activities. In our design, this diagram shows the rent process activities. When a customer rent a car, the

clerk will first check the customer's reservation and make his first choice. Then the clerk will make other choices about the storage capacity issues as well as verifying issues. In the whole work flow, the customer also should make some choices to determine his renting process.

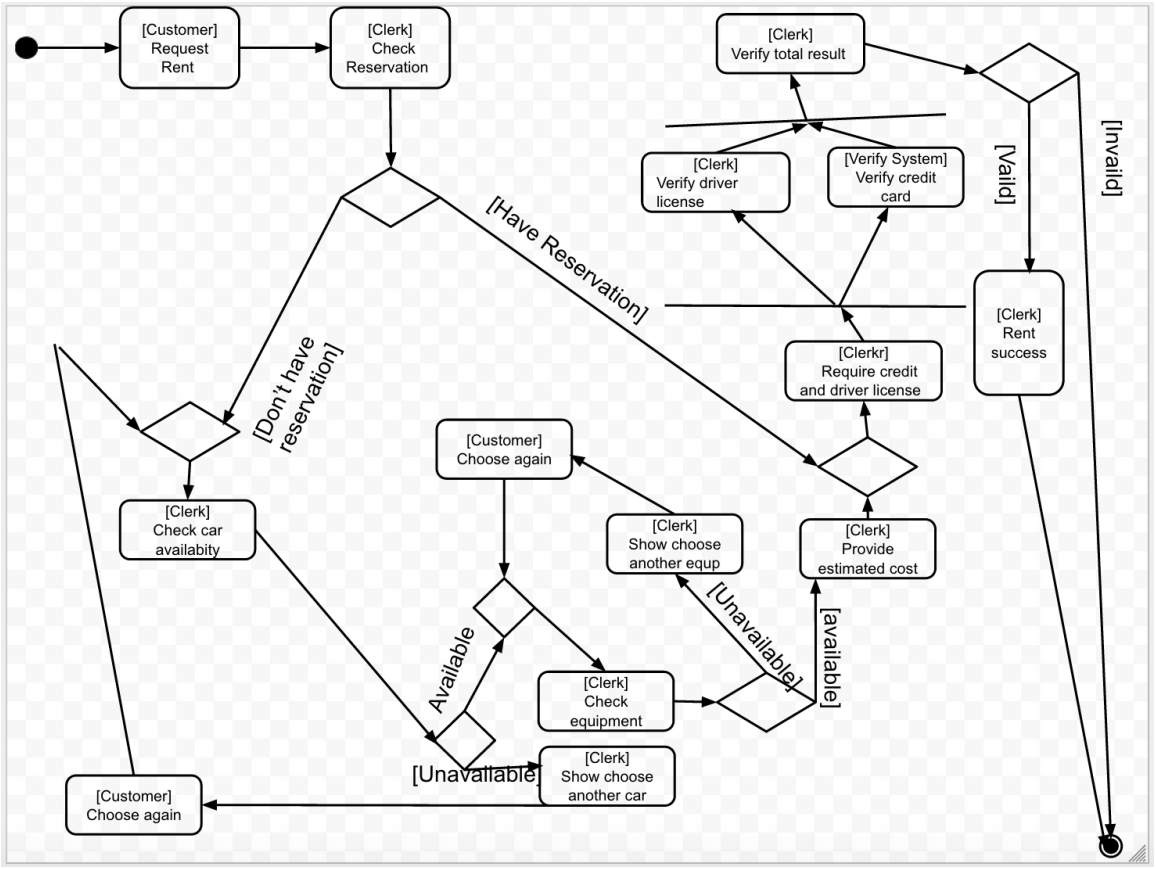


Figure 18 Activity diagram

Chapter 5 Test and Screenshots Results

5.1 Black-Box Test

Black Box Test method is used in this project which is shown in Figure 19. A package named Test Packages is provided to serve as automatic test scripts. For the sake of time limit, only major methods of Admin model and User Model

are tested, while the others are tested manually by taking screenshots of the database updates caused by the application. JavaNG test approach is adopted by giving some test cases, and the updates caused by the test cases will be printed out from the database to compare with the expected outputs. Four major functions tested are Reserve, Rent, Add and Remove clerk, and all the tests passed. The following figure is the screenshot of the JavaNG test of the Reserve and Rent functions, and the test results are printed out to the console. Also, since frequent Scrum meetings are adopted, team members have the chance to do code reviews with each other, which reduces the bugs in our project.

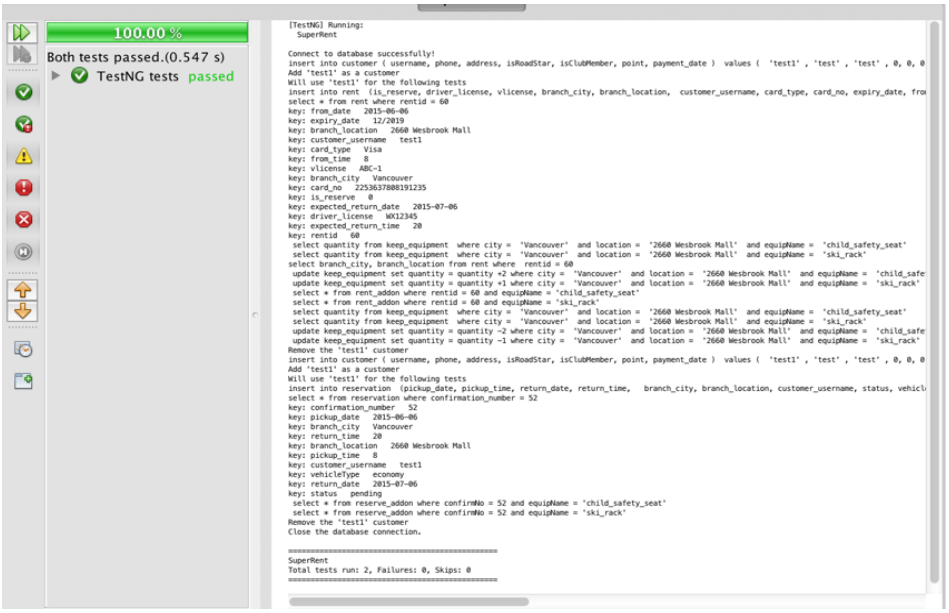


Figure 19 JavaNG test for Reserve&Rent functions

5.2 Manual Test Screenshots

Screenshots are used to test the functionalities of the application. Most of the Test Cases are included in the Appendix A: Test Cases, while some typical ones serve as examples to show the test process. For example, the test case screenshots for Generating Reports is shown below. The entire process is from Figure 20 to Figure 25. The figures are also the guideline to show the operations of generating different types of reports. This function does not do any update to the database, so only data queries are shown under different conditions. Figure 2

to Figure 5 show different types of report contents, and Figure 6 shows how to export the report. The results are compared with the database manually, and the results match the database records. Based on this, the test passes.

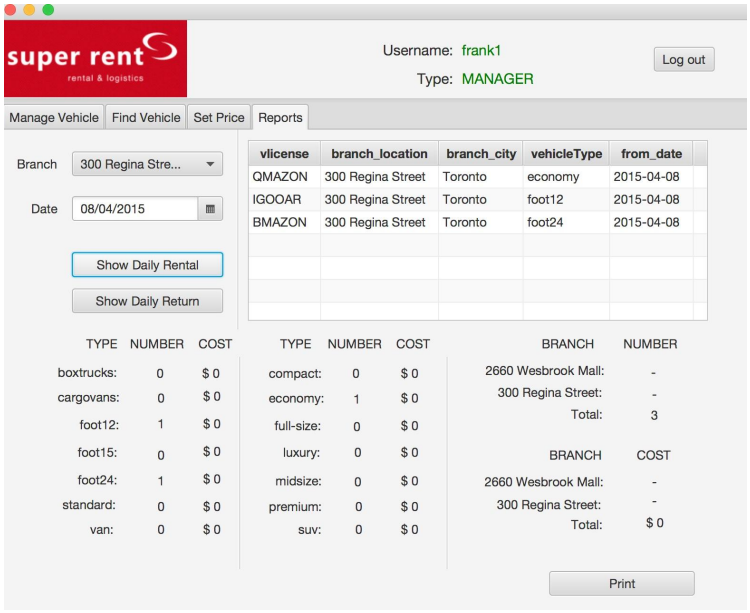


Figure 20 Getting the reports for specific branch for the Daily Rental

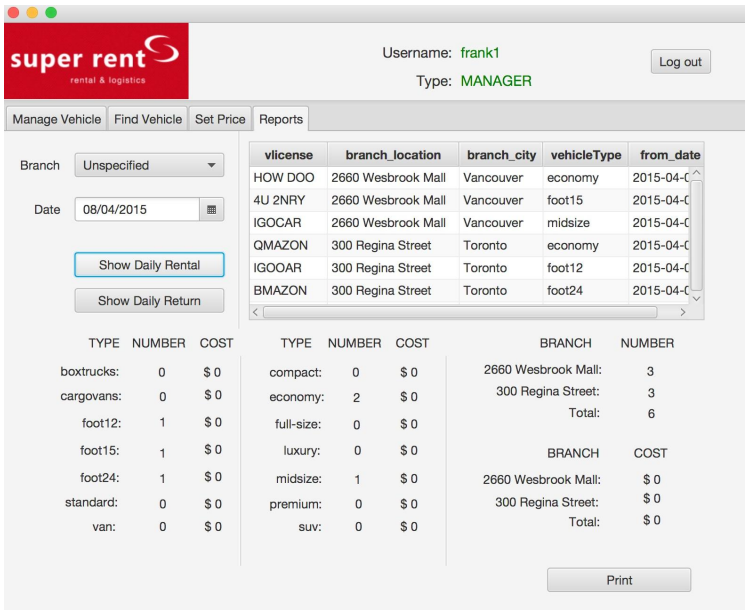


Figure 21 Getting the reports for Daily Rental for all the branches

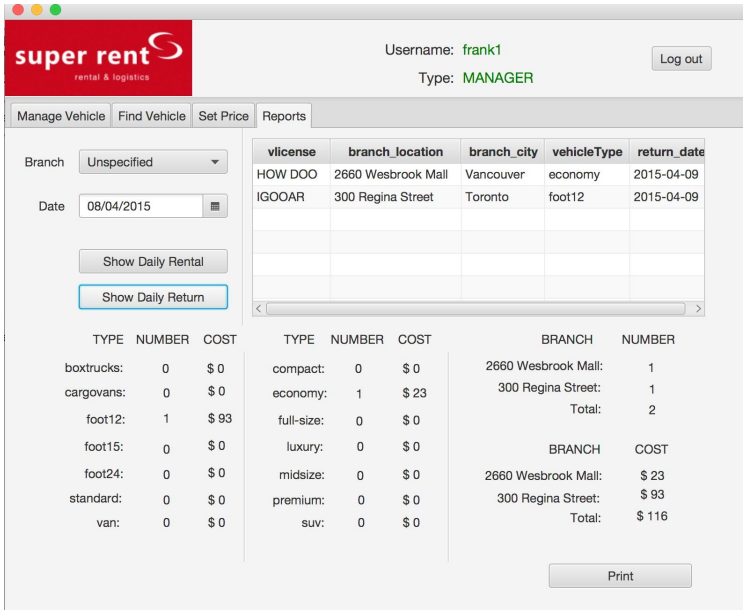


Figure 22 Getting the reports for Daily Return for all the branches

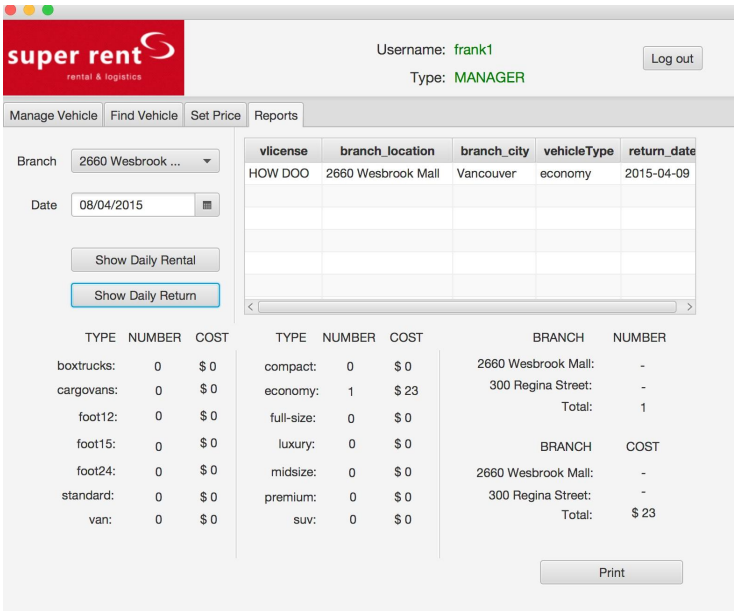


Figure 23 Getting the reports for Daily Return for specific branch

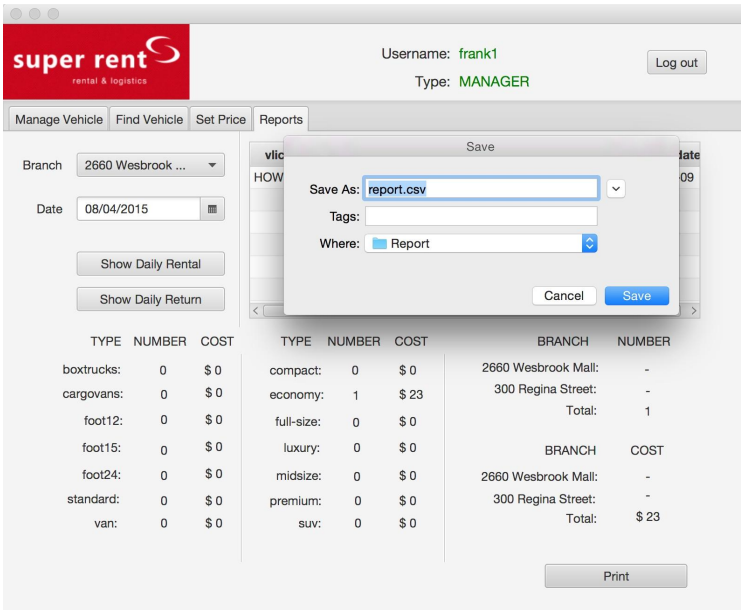


Figure 24 Save the Report as a CSV file for printing

License	Location	City	Type	Date	
HOW DOO	2660 Wesbrook Mall	Vancouv	economy	2015-04-09	
IGONAR	2660 Wesbrook Mall	Vancouv	van	2015-04-09	
IGOOAR	300 Regina Street	Toronto	foot12	2015-04-09	
Type	Number	Cost			
boxtrucks	0	$0			
cargovans	0	$0			
foot12	1	$93			
foot15	0	$0			
foot24	0	$0			
standard	0	$0			
van	1	$23			
compact	0	$0			
economy	1	$23			
full-size	0	$0			
luxury	0	$0			
midsize	0	$0			
premium	0	$0			
suv	0	$0			
Subtotal for Westbrook Branch	2	$46			
Subtotal for 300 Regina Street Branch	1	$93			
Total Numbers	3	$139			

Figure 25 A sample of CSV file for the daily print

Another example is setting the price used by manager. Figure 26 to Figure 29 shows the process for manager to set the new price for different types of vehicles. For here, the foot15 truck is selected, and the new rates are shown in Figure 9. The unit used in database is cent, so the new values match the input. The test passes.

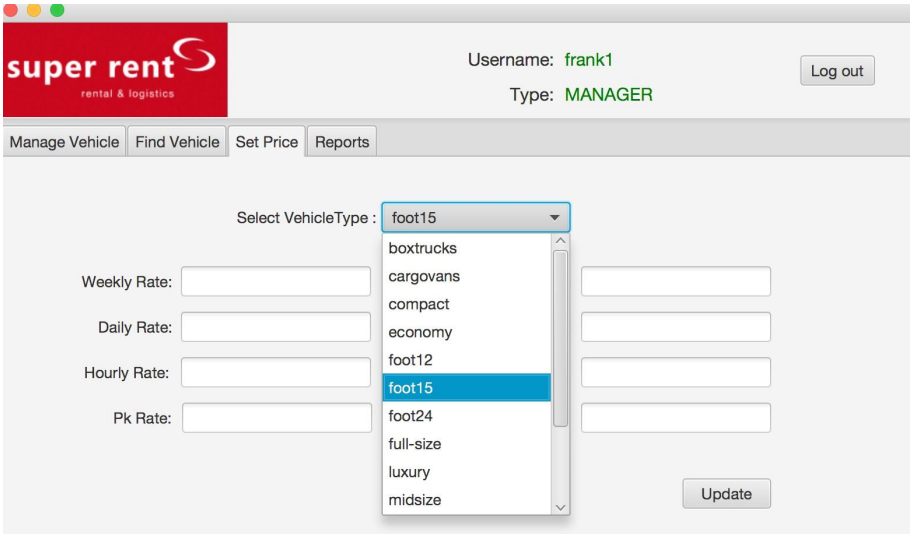


Figure 26 Manager selects different car types and set different rates

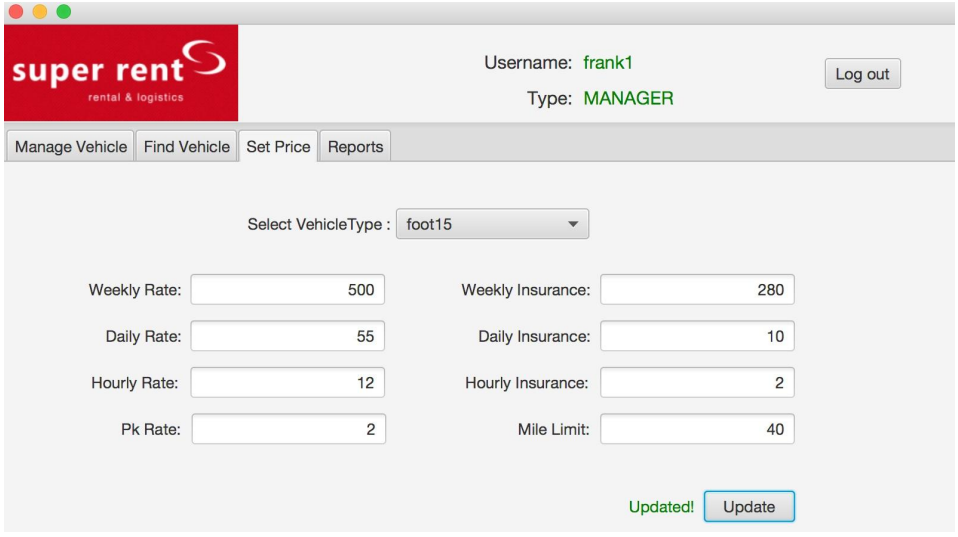


Figure 27 Manager has selected vehicle type and set the new rate

foot12	55000	8000	1500	200	28000	1000	200	40
foot15	70000	95000	17000	300	29000	200	1100	5000
foot24	60000	8000	1700	200	28000	1000	200	40

Figure 28 Database records before updating

compact	40000	5500	1100	200	28000	1000	200	40
economy	35000	5000	1000	200	28000	1000	200	40
foot12	55000	8000	1500	200	28000	1000	200	40
foot15	50000	5500	1200	200	28000	200	1000	40
foot24	60000	8000	1700	200	28000	1000	200	40
full-size	49000	7000	1400	200	28000	1000	200	40

Figure 29 Database records after setting the new prices

Chapter 6 Conclusion

The objective of this project is to design and implement a car rental application named SuperRent to a vehicle renting company. The key functions for the application are Reserve, Rent or Return vehicles and generate reports. Other important functions include some database management functions such as add/remove users, add/remove vehicles, membership and so on. These major functions are completed; moreover, system tests are also done to test the performance of them. Currently, the version of the application can work well but more tests and refinements of the functions are needed. Bugs cannot be avoided, and the interface is not quite user-friendly. Other limitations are that some non-functional requirements are not considered yet. Some constraints of the database are not added such as the valid time of the membership and validation of the reservation status.

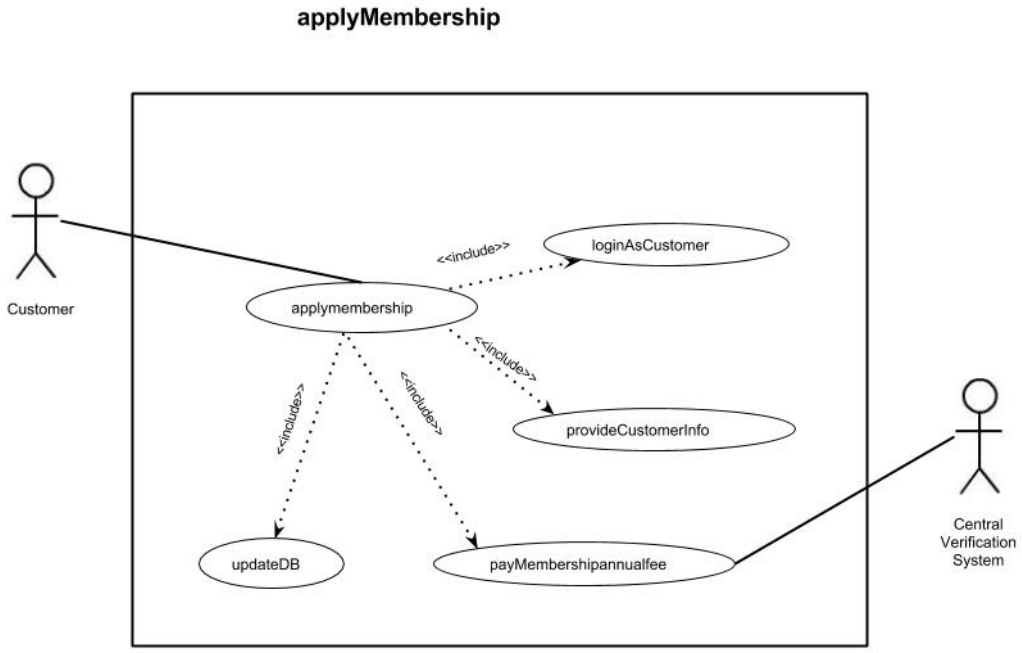
Acknowledgement

We would like to give our deepest appreciation to Dr. Liu and Dr. Wu to help us complete the project.

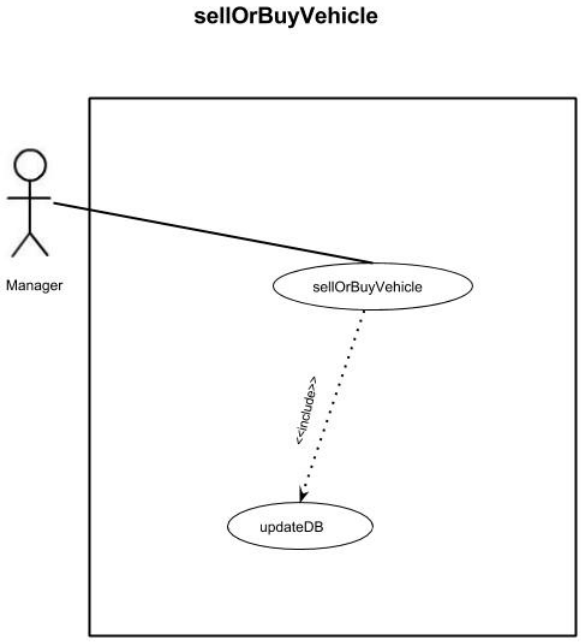
Appendix A: Use Case Diagrams

Appendix A-1 Use Case Diagram (Apply membership)

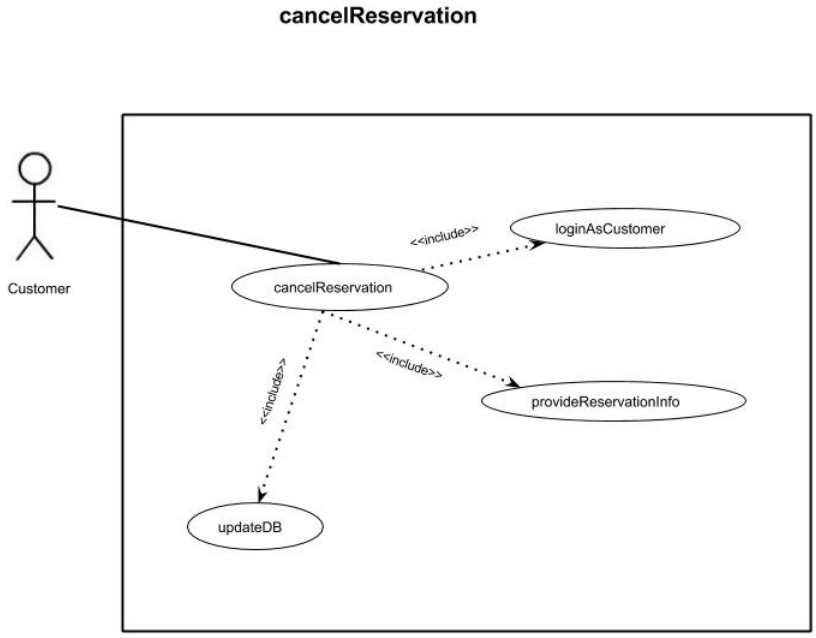
Description: Customer can apply for membership, login in as customer, query his information, and pay annual fees.



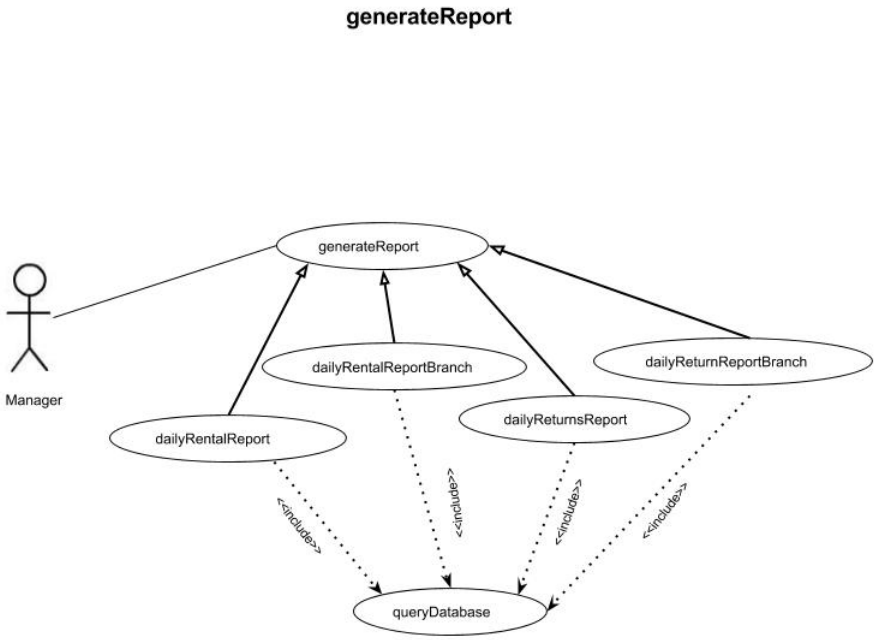
Appendix A-2 Use Case Diagram (Buy or sell vehicle)



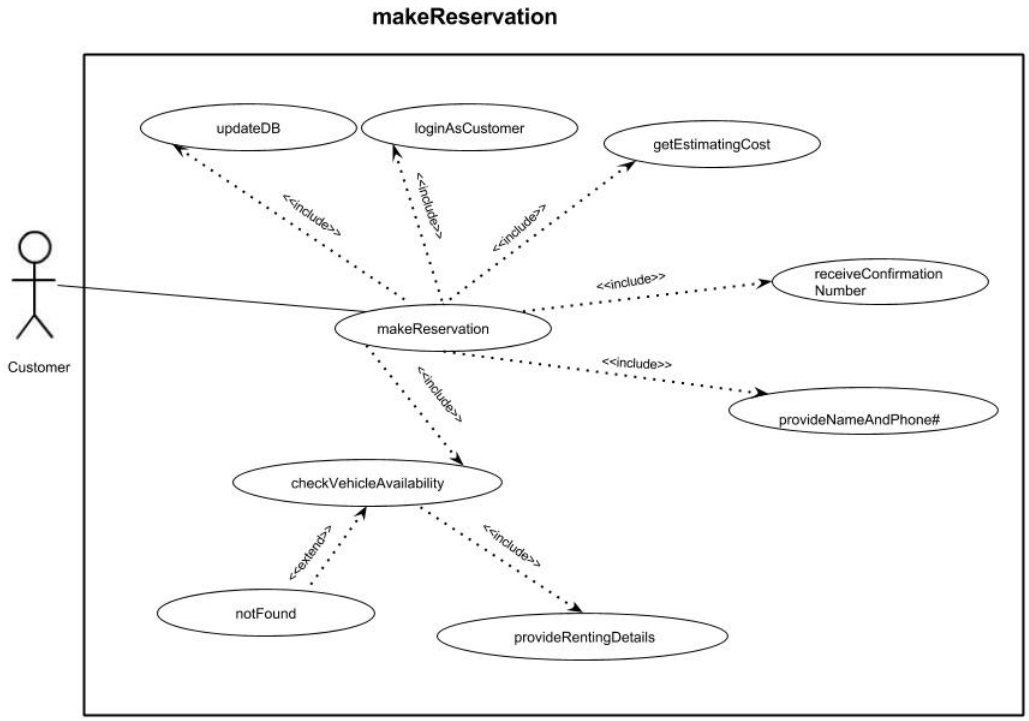
Appendix A-3 Use Case Diagram (Cancel Reservation)



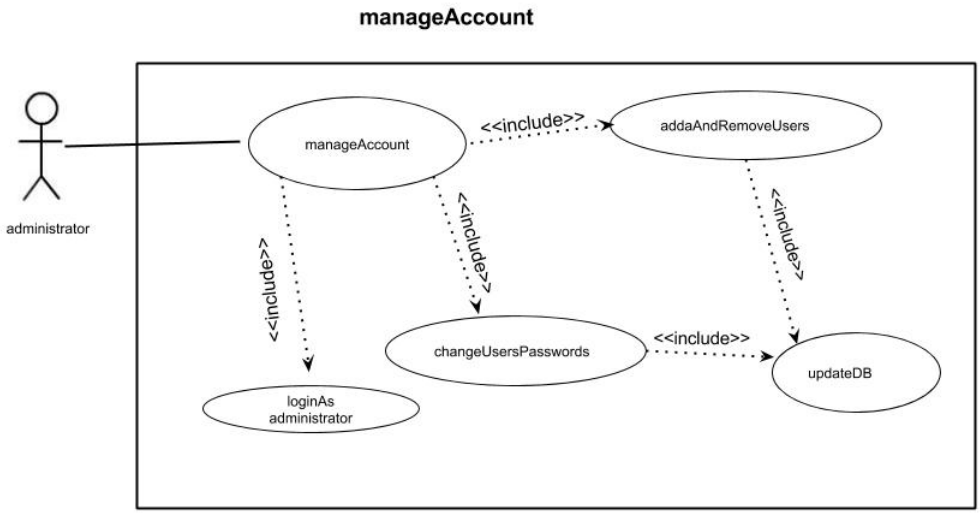
Appendix A-4 Use Case Diagram (Generate Report)



Appendix A-5 Use Case Diagram (Make Reservation)

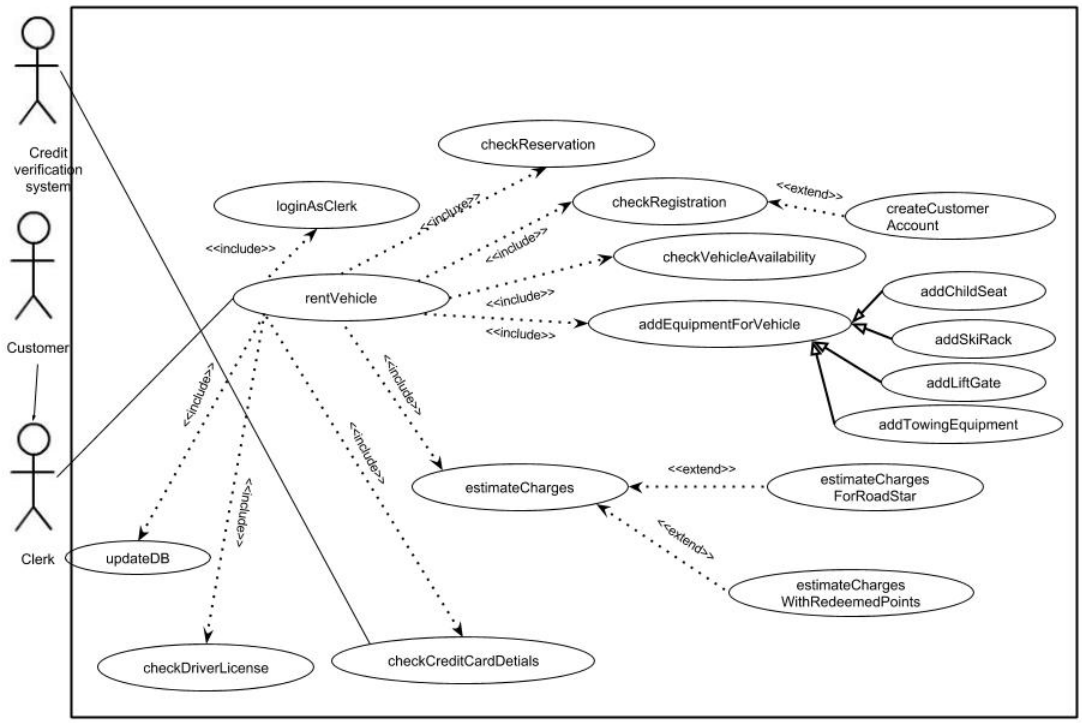


Appendix A-6 Use Case Diagram (Manage Account)

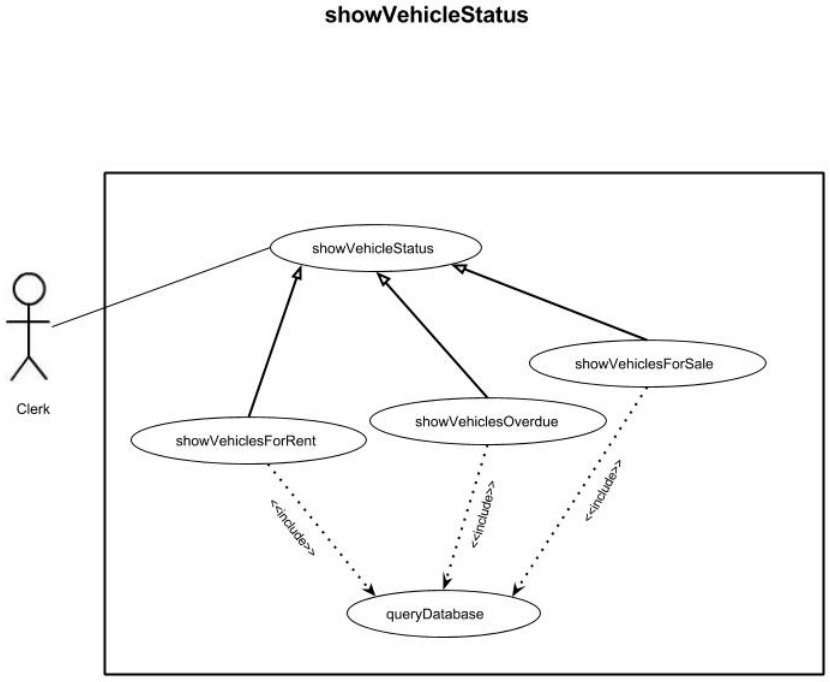


Appendix A-7 Use Case Diagram (Rent Vehicle)

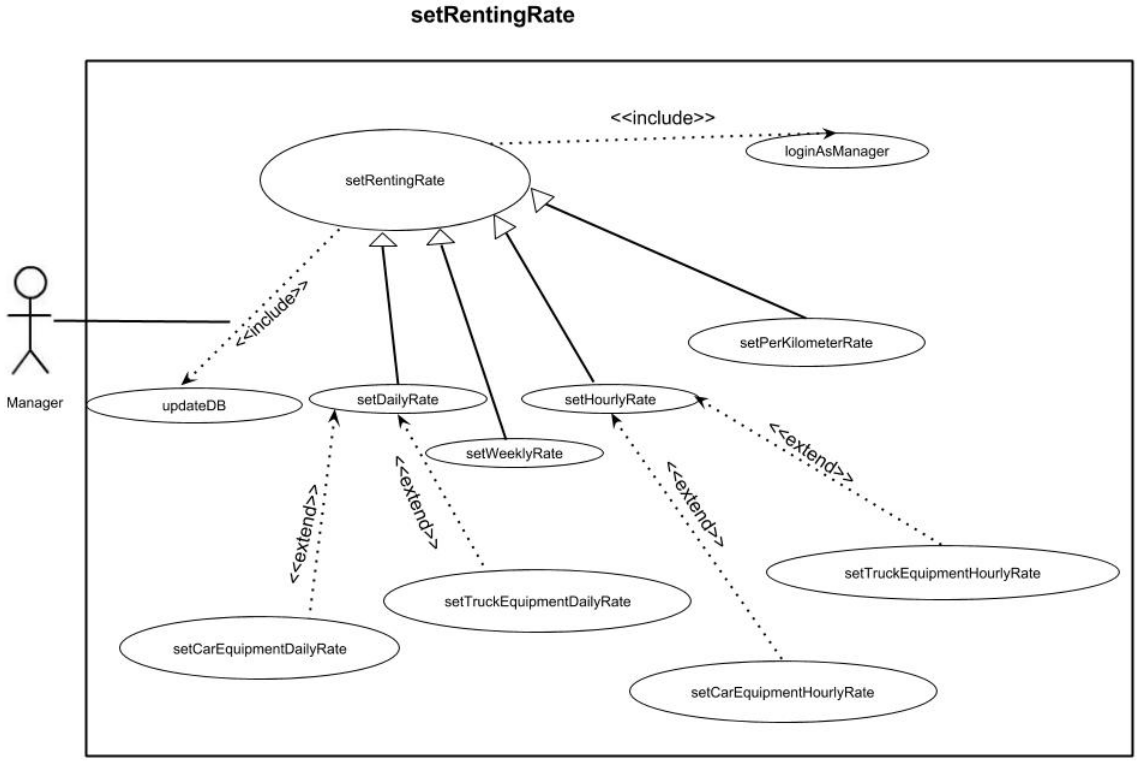
rentVehicle



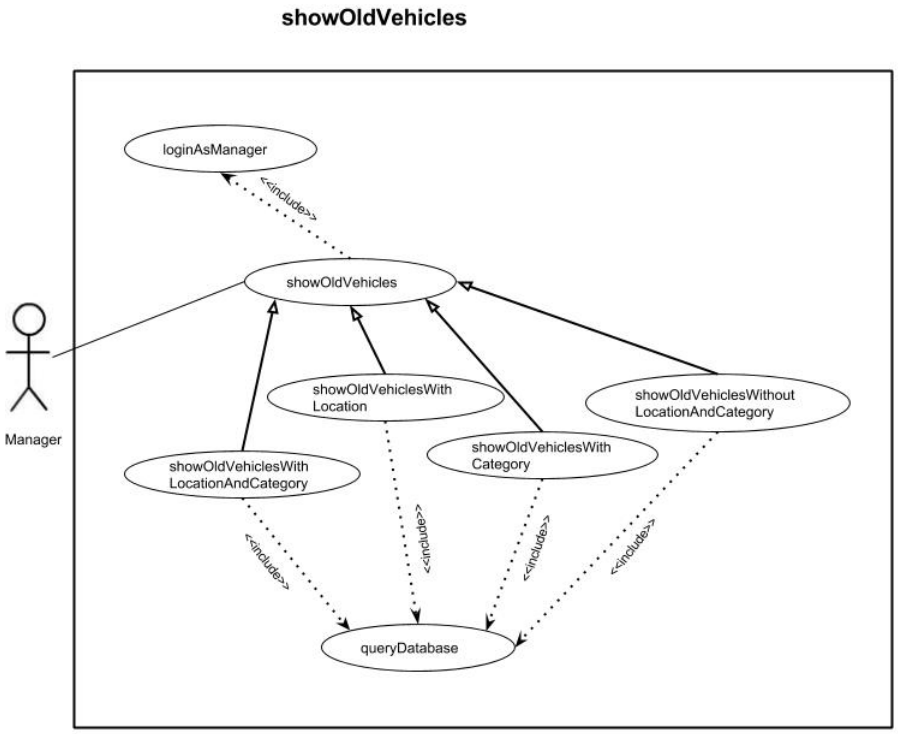
Appendix A-8 Use Case Diagram (Show Vehicle Status)



Appendix A-9 Use Case Diagram (Set Renting Rate)



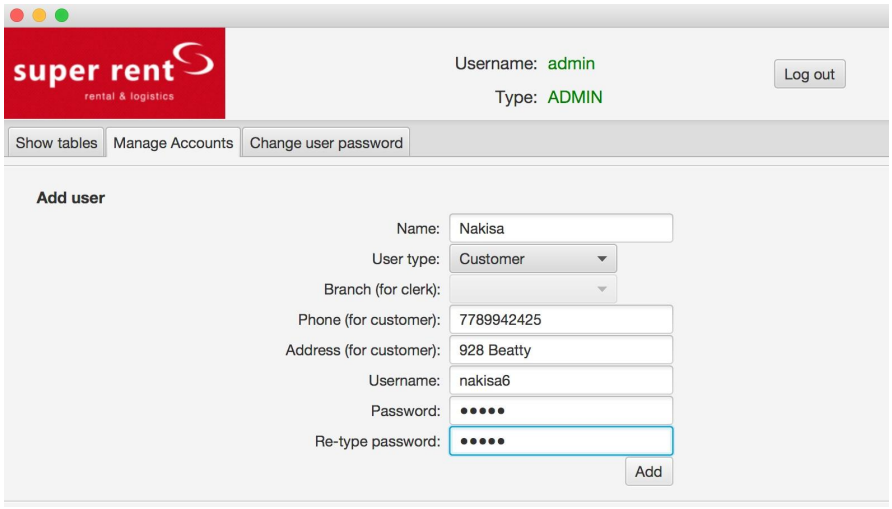
Appendix A-10 Use Case Diagram (Show Old Vehicles)



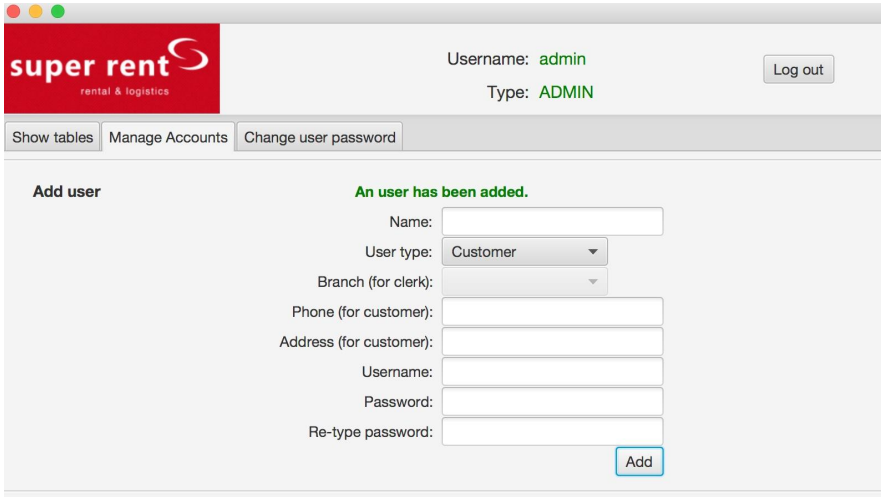
Appendix B: Test Cases

Appendix B-1: Test Case (Admin: Manage Accounts: Add user)

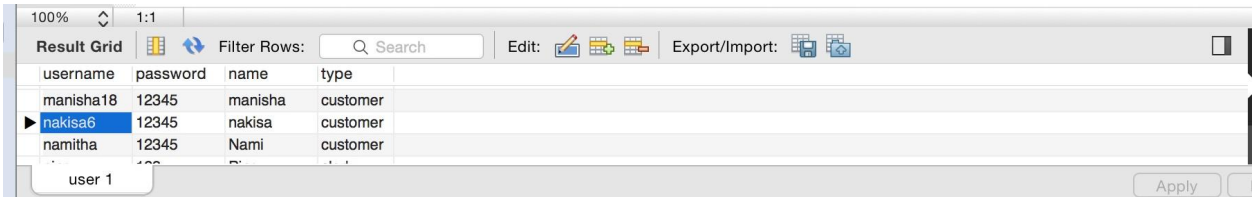
Before clicking add button, the user information is shown as follows:



After clicking add button, the user information should be added to the database, and an indicator is shown to tell the user that the operation is completed.

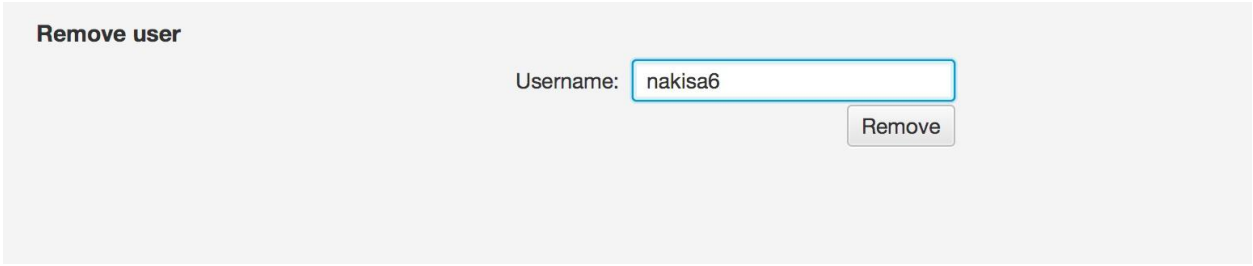


Test passed: user has been added to the customer table in the database.

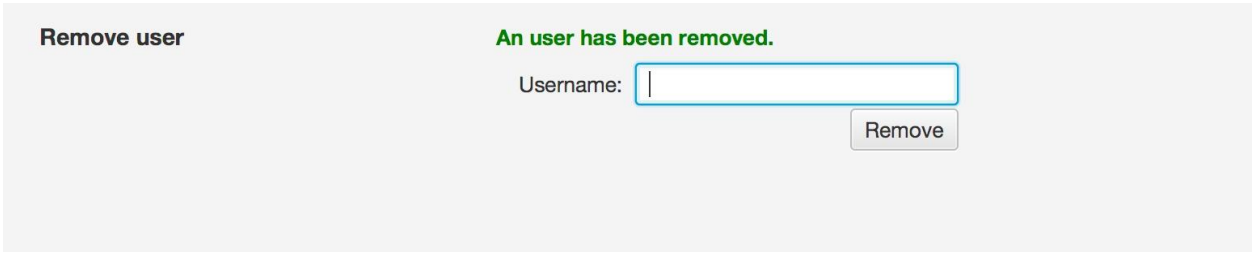


Appendix B-2: Test Case (Admin: Manage Accounts: Remove User)

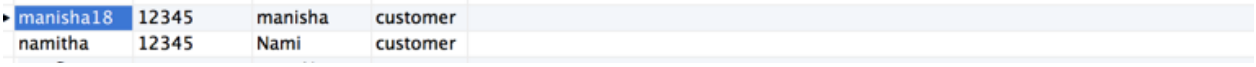
Before clicking remove button, the username to be removed is shown as follows:



After clicking remove button, an indicator is used to shown the user has been successfully removed.



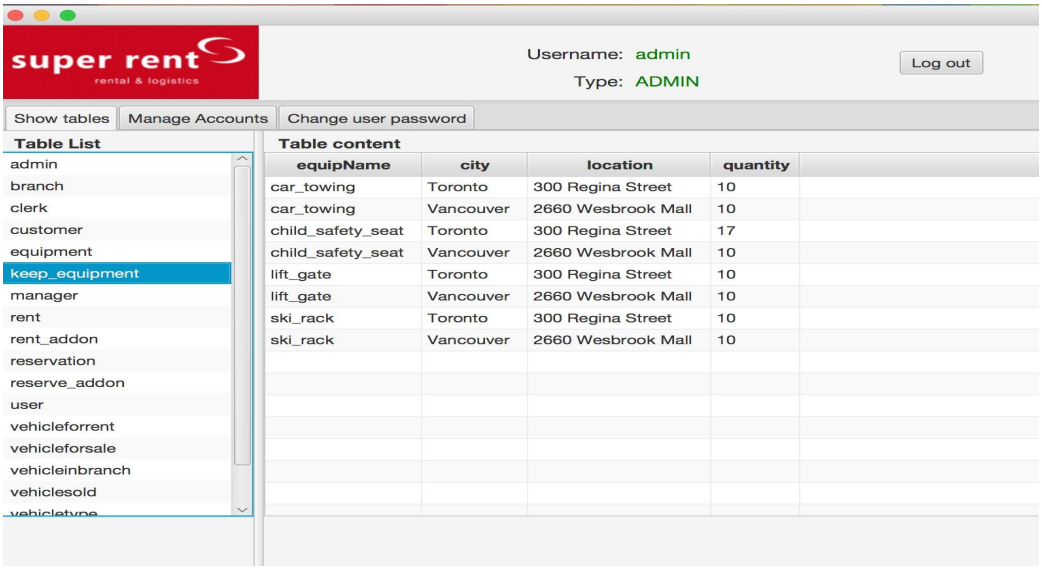
Test passed: user has been removed from the database.



manisha18	12345	manisha	customer
namitha	12345	Nami	customer

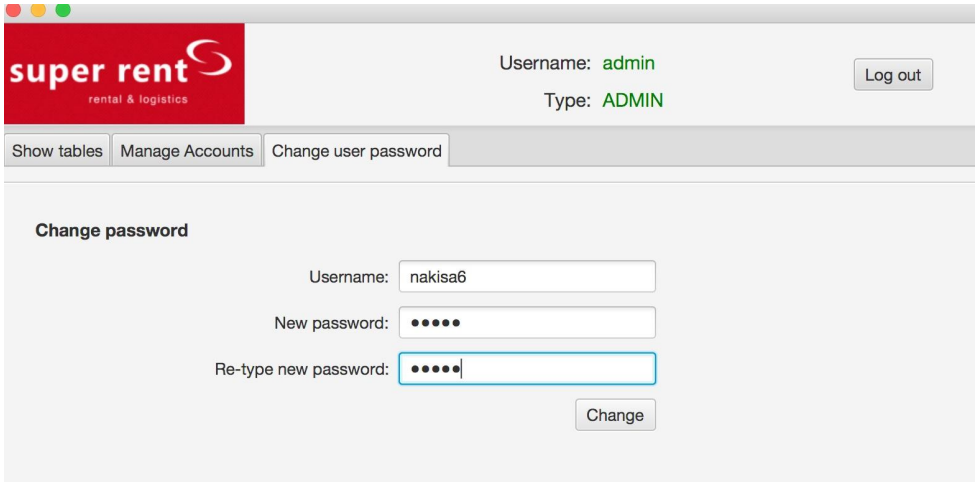
Appendix B-3: Test Case (Admin: Show table)

Test passed: when user clicks the Show Tables tab, all tables in the database are shown.

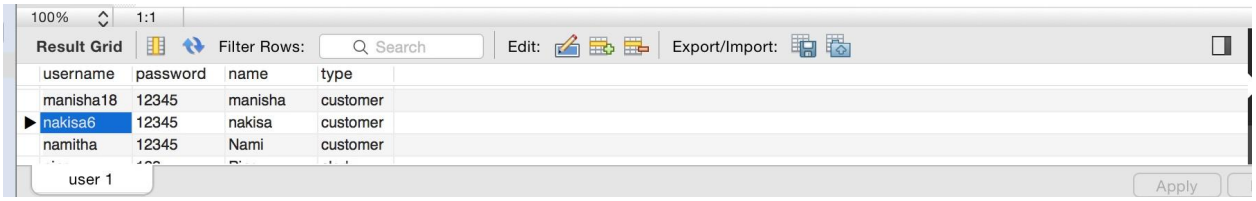


Appendix B-4: Test Case (Admin: Change User Password)

Before clicking change button, the information is shown as follows:



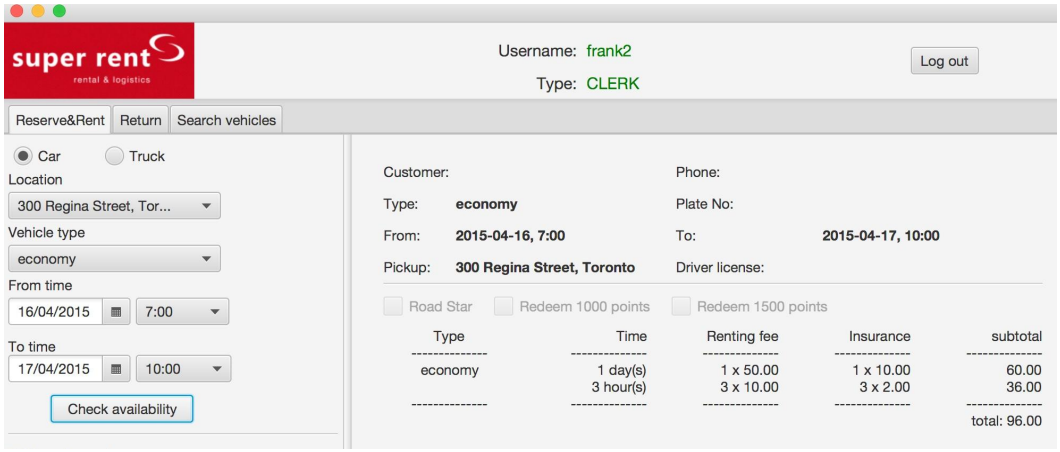
After clicking change button, database has been updated.



Test Passed (Password has been changed)

Appendix B-5: Test Case (Clerk: Reservation and Rent: Check Availability)

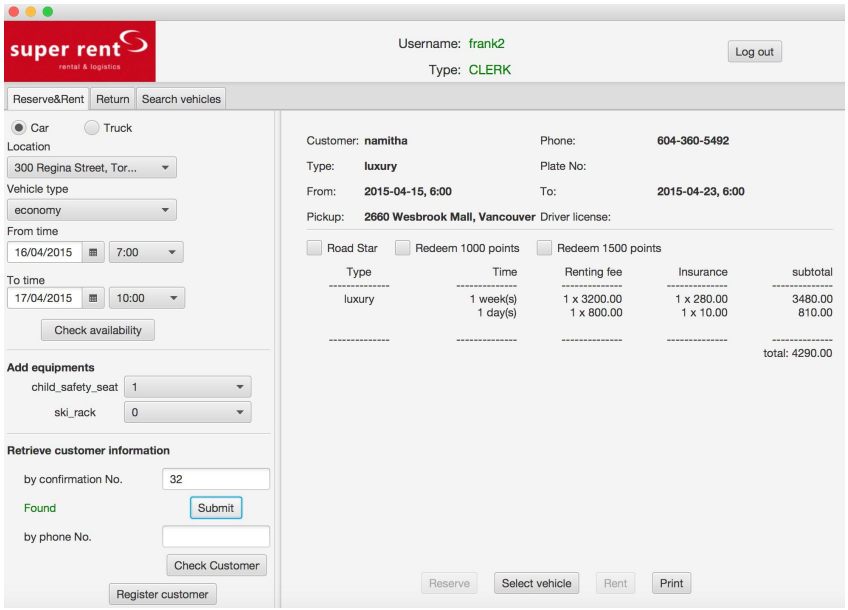
User first inputs the following information and selects the Location, Type and Date. After clicking Check availability button, the input information will be shown on the right, and the estimation cost is also displayed.



Test Passed

Appendix B-6: Test Case (Clerk: Reservation and Rent: Retrieve Reservation status by confirmation number)

Before clicking submit button, the confirmation no is as follows:

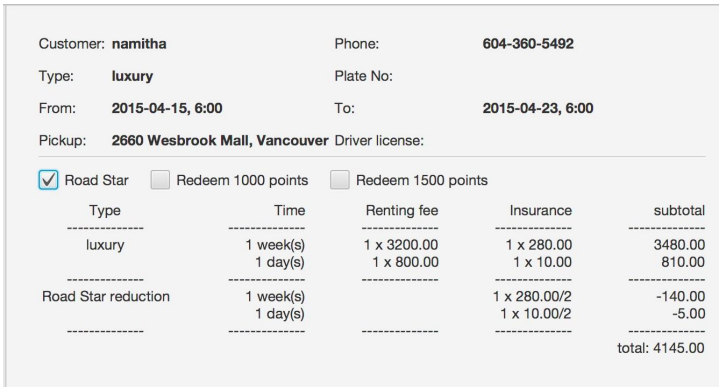


After clicking submit button, the information of the customer is also shown on the right.
After the customer reserve the vehicle, the status of the vehicle will be changed to pending.

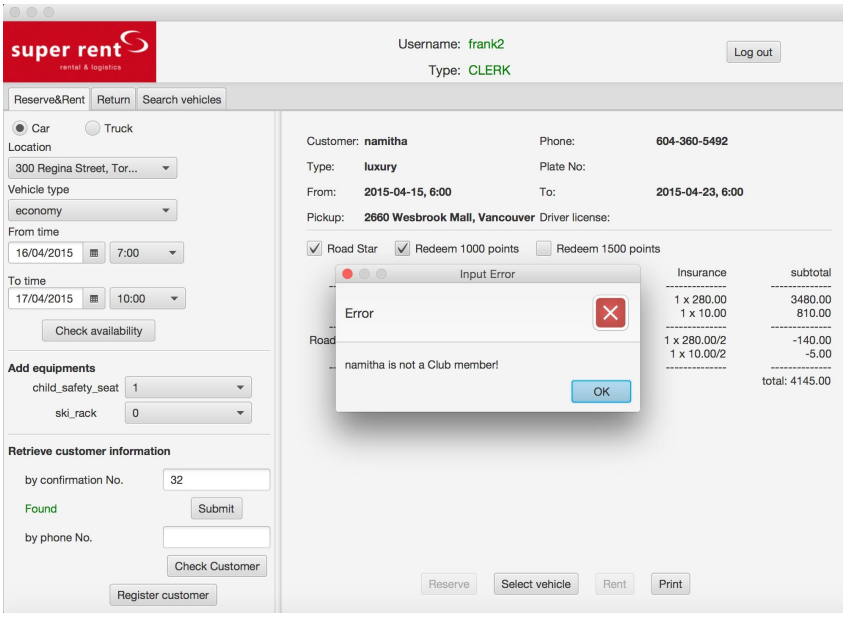
Test Passed.

Appendix B-7: Test Case (Clerk: Reservation and Rent: Show estimation cost)

Check whether the Customer is a Road star:



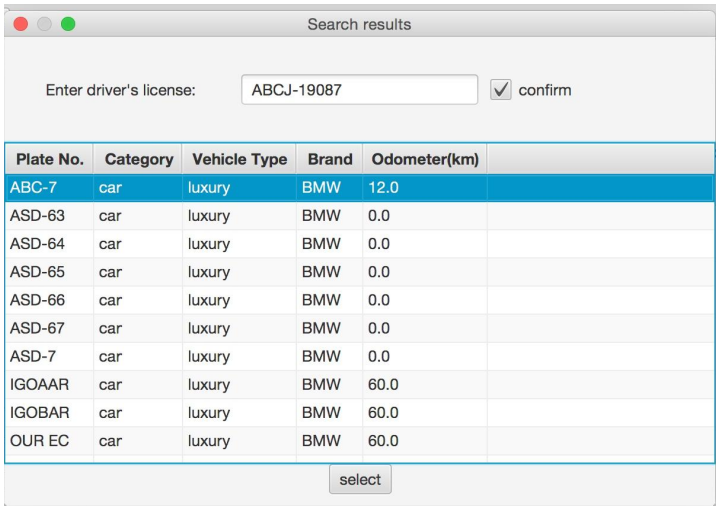
Check whether the customer is a club member:



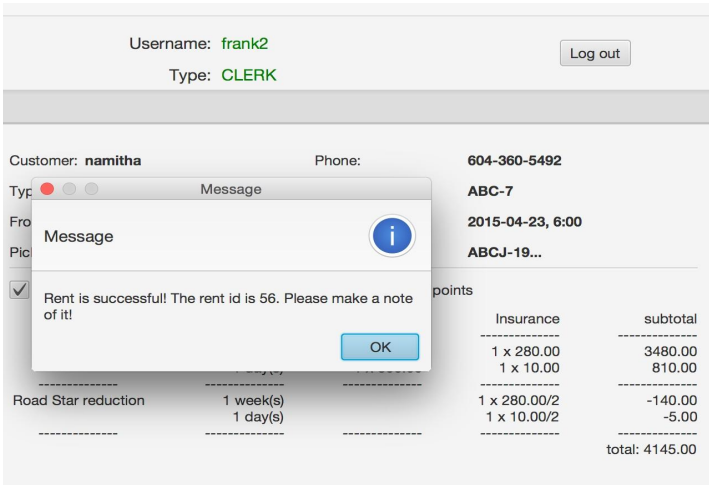
Test Passed (All displayed information agree with those in database).

Appendix B-8: Test Case (Clerk: Select and Rent)

User input the driver's license for selecting the vehicle:



User can click rent button after selection:

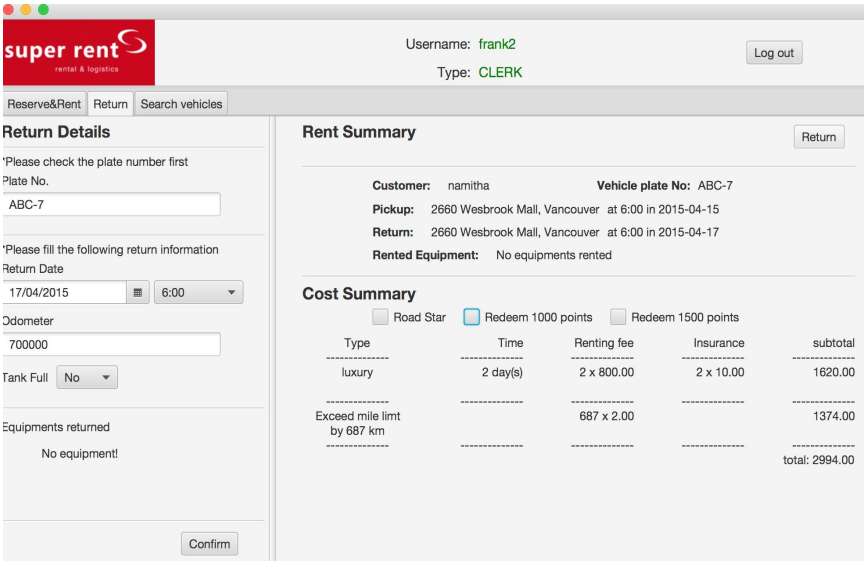


Database has been updated: After the vehicle has been selected, it is added to the renting table.

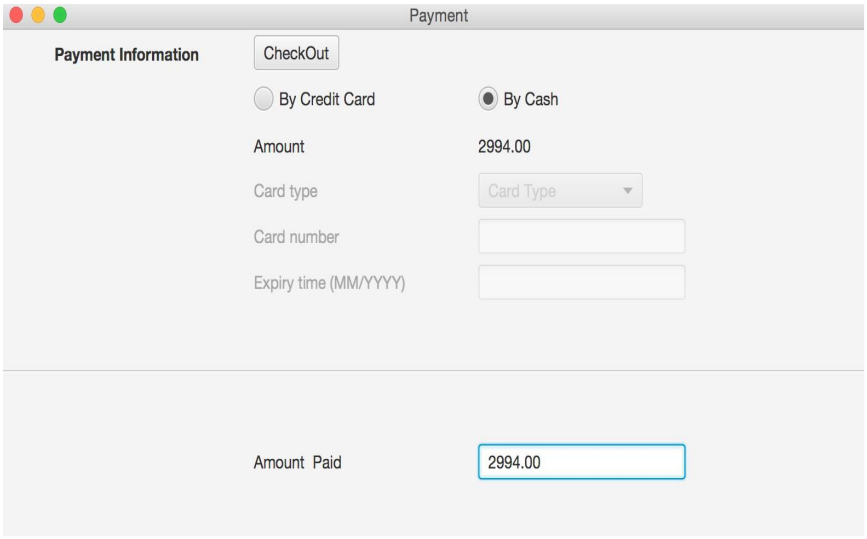
Test Passed.

Appendix B-9: Test Case (Clerk: Return: Getting the Rent summary by entering the Plate No)

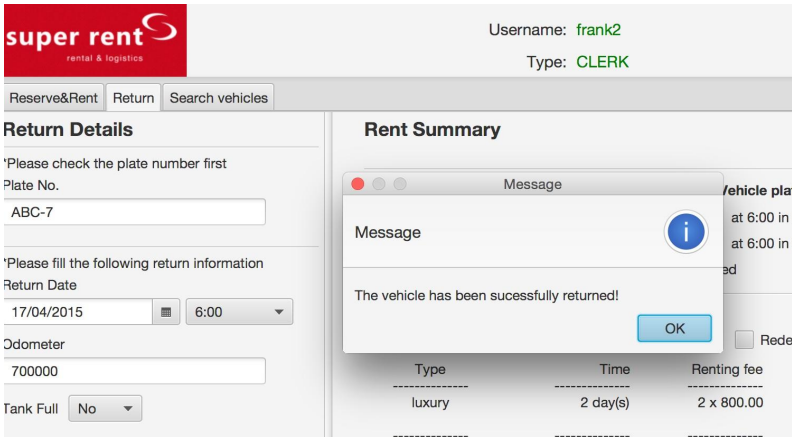
Information about the vehicle to be returned is shown as follows, and the rent cost is displayed on the right when confirm button is pressed.



After the return button is clicked, the payment interface is shown as follows:



If the payment is successful, then an indicator will be shown.



The rent table before update:

47	1	C1234	ASD-6	Vancouver	2660 Wesbrook Mall	namitha	MasterC...	5415935026170659	09
▸ 56	1	ABCJ-19087	ABC-7	Vancouver	2660 Wesbrook Mall	namitha	Visa	4507556798979872	03

The return table after update:

41	46	2015-04-30	6	Toronto	300 Regina Street	1	10000	2025	Cash
▸ 54	56	2015-04-17	6	Vancouver	2660 Wesbrook Mall	0	700000	2994	Cash

The status of the vehicle in the vehicleforrent table is also updated (1 means available now).

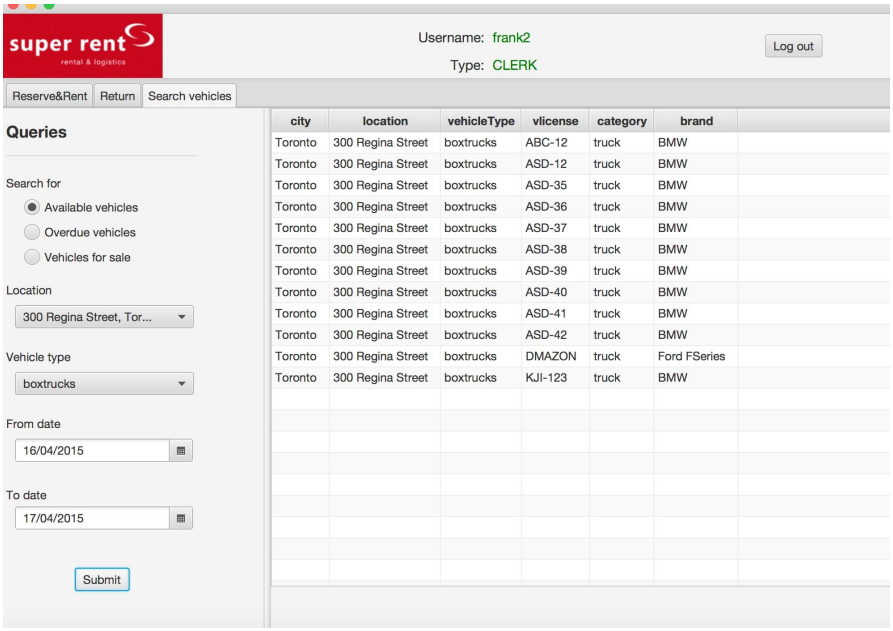
ABC-6	0	2015-04-10	car	full-size	BMW	frank1	0
▸ ABC-7	1	2015-04-10	car	luxury	BMW	frank1	700000
ABC-8	1	2015-04-10	car	midsize	BMW	frank1	0
ABC-9	1	2015-04-10	car	premium	BMW	frank1	0
AMAZON	1	2006-05-01	truck	foot15	Ford...	frank1	30000
ASD-1	1	2015-04-11	car	economy	BMW	frank1	0
ASD-10	1	2015-04-11	car	standard	BMW	frank1	12000

vehicleforrent 1

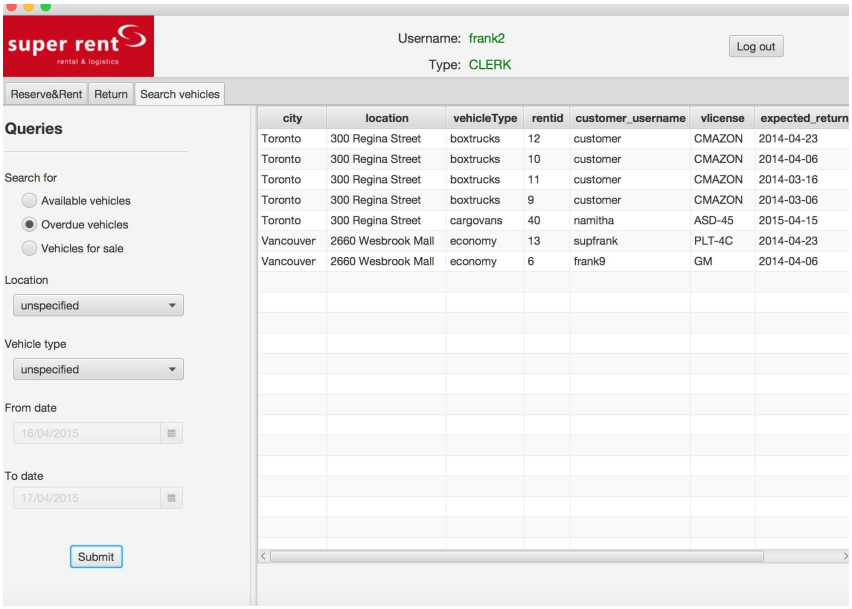
Test Passed (All related data are updated).

Appendix B-10: Test Case (Clerk: Search Vehicle)

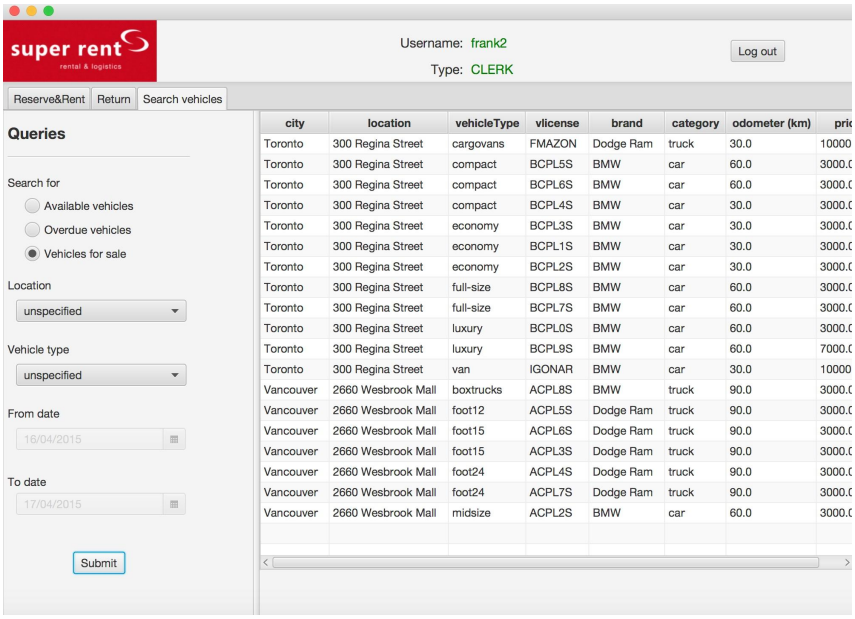
User input all required vehicle information, and all available vehicles will be shown if available vehicles is checked.



Check Overdue Vehicles: Unspecified location and type



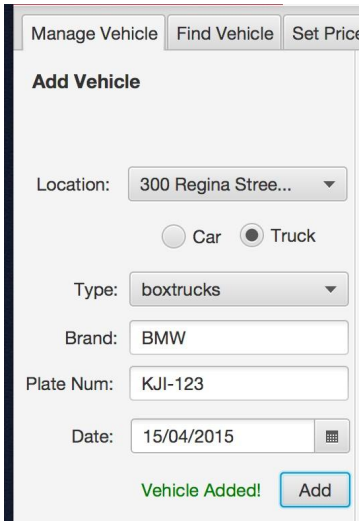
Check Vehicle for sale: By location, type and date



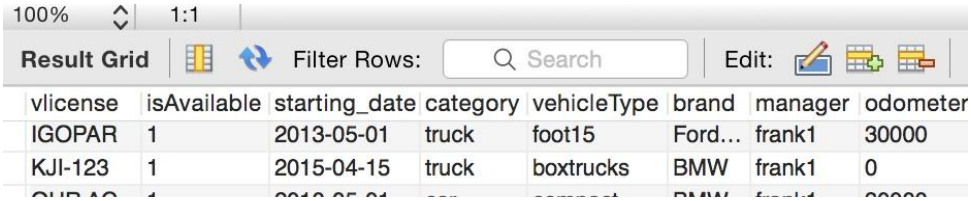
Test Passed (All queries returned correct information).

Appendix B-11: Test Case (Manager: Manage Vehicle: Add Vehicle)

Manager gives the vehicle information and click add button.



Database has been updated: Vehicle has been added to the database

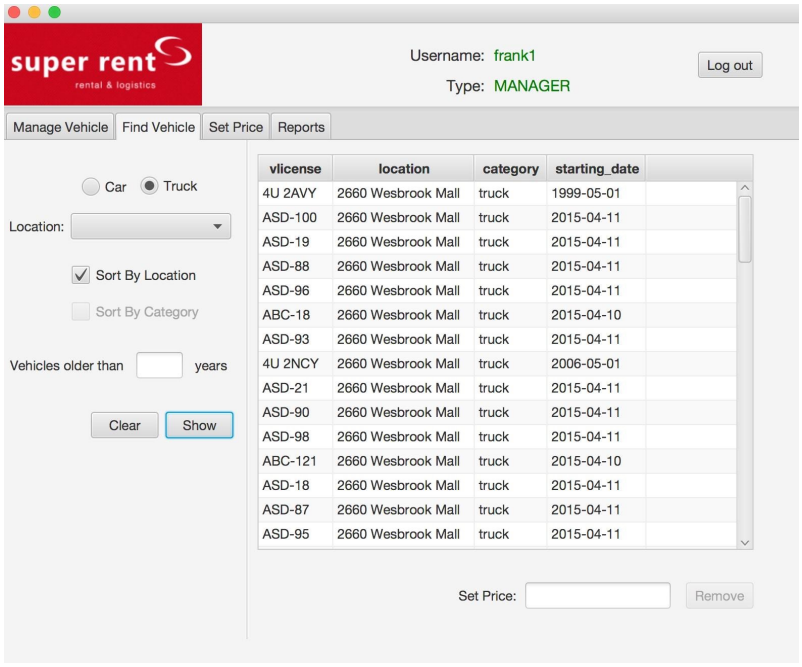


vlicense	isAvailable	starting_date	category	vehicleType	brand	manager	odometer
IGOPAR	1	2013-05-01	truck	foot15	Ford...	frank1	30000
KJI-123	1	2015-04-15	truck	boxtrucks	BMW	frank1	0

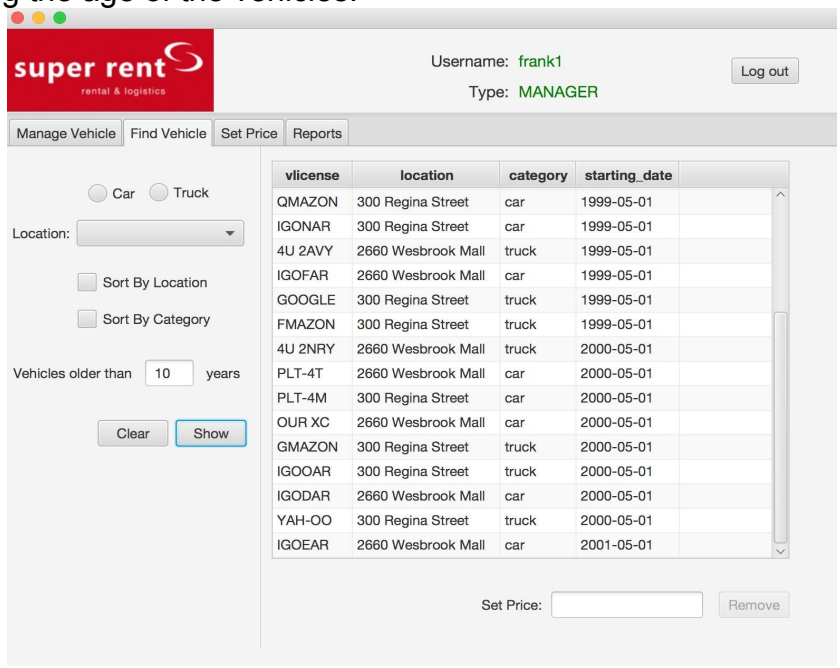
Test Passed.

Test Case A-12 (Manager: Find Vehicle)

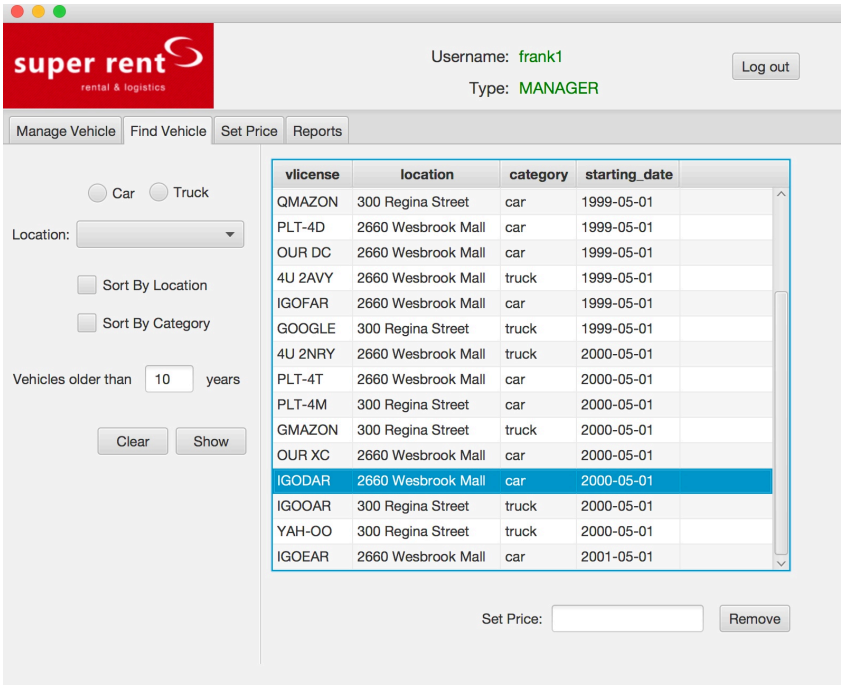
By choosing the type and sorting by location



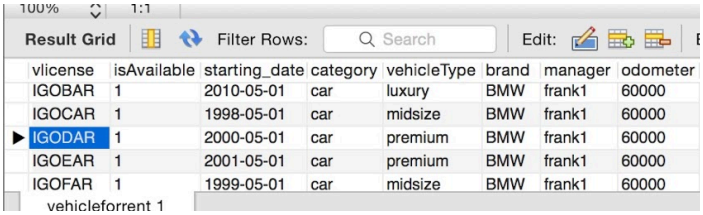
By choosing the age of the vehicles:



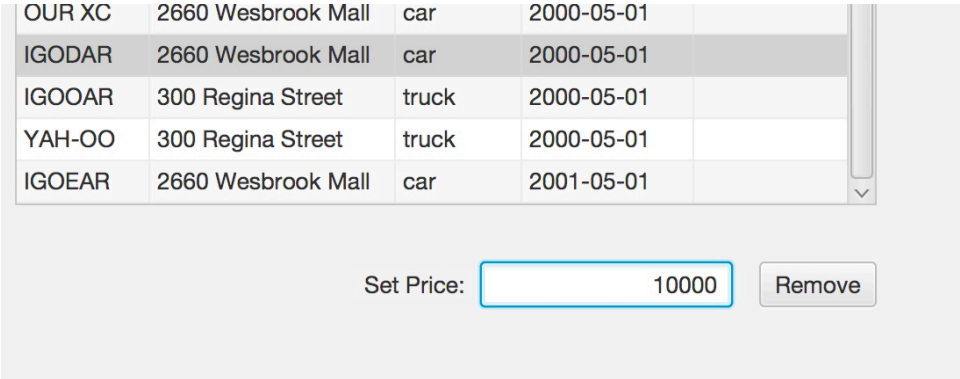
Selecting a vehicle and set the price:



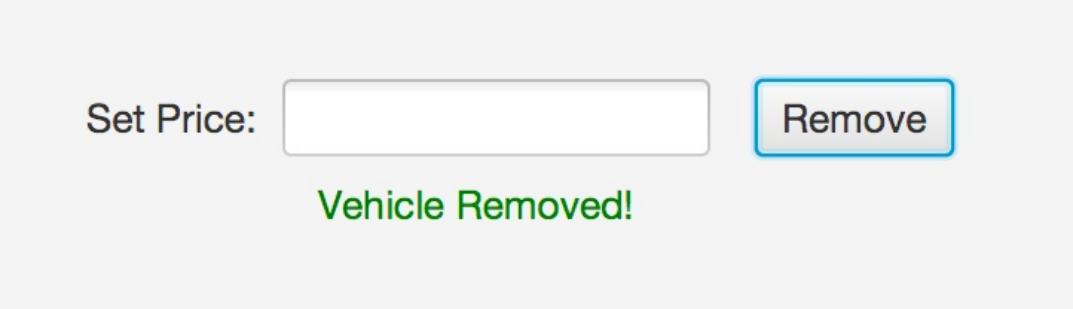
Database before removing the vehicle from Rent to for Sale table:



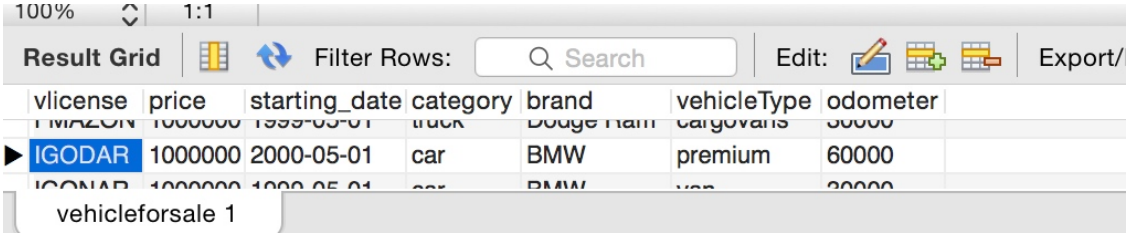
Setting the price and removing the vehicle from rent:



Vehicle is removed:



Database: The vehicle has been removed from Vehicle for Rent Table and added to vehicle for sale:



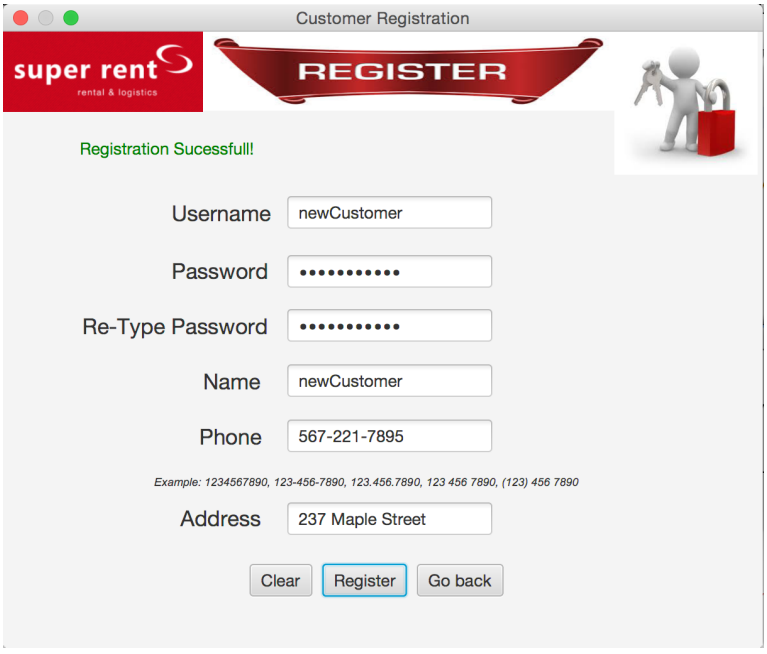
Test Passed.

Appendix B-13: Test Case (Customer: registration)

Before the registration, ‘newCustomer’ doesn’t exist

username	password	name	type
customer5	customer5	customer5	customer
customer6	customer6	customer6	customer
customer7	customer7	customer7	customer
customer8	customer8	customer8	customer
customer9	customer9	customer9	customer
frank10	12345	frank10	customer
frank7	12345	frank7	customer
frank8	12345	frank8	customer
frank9	12345	frank9	customer
manisha1	123456	manisha	customer
manisha18	12345	manisha	customer
namitha	12345	Nami	customer
supfrank	12345	frank	customer
NULL	NULL	NULL	NULL

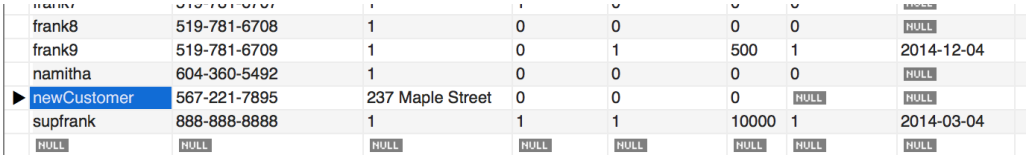
A customer registered as ‘newCustomer’



After this operation, ‘newCustomer’ is in the User table.

namitha	12345	Nami	customer
▶ newCustomer	newCustomer	newCustomer	customer
supfrank	12345	frank	customer
NULL	NULL	NULL	NULL

Also, the customer table is updated.

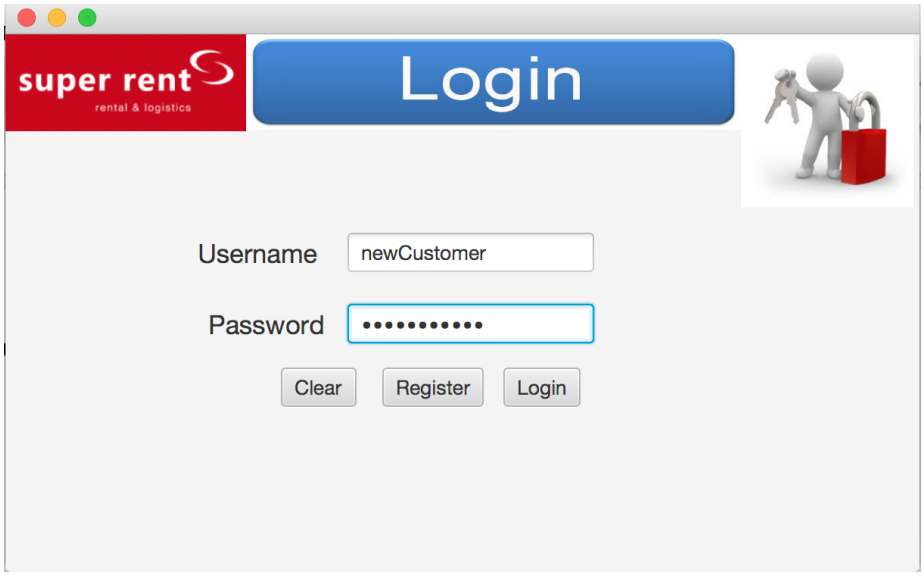


frank8	519-781-6708	1	0	0	0	0	NULL
frank9	519-781-6709	1	0	1	500	1	2014-12-04
namitha	604-360-5492	1	0	0	0	0	NULL
▶ newCustomer	567-221-7895	237 Maple Street	0	0	0	NULL	NULL
supfrank	888-888-8888	1	1	1	10000	1	2014-03-04
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

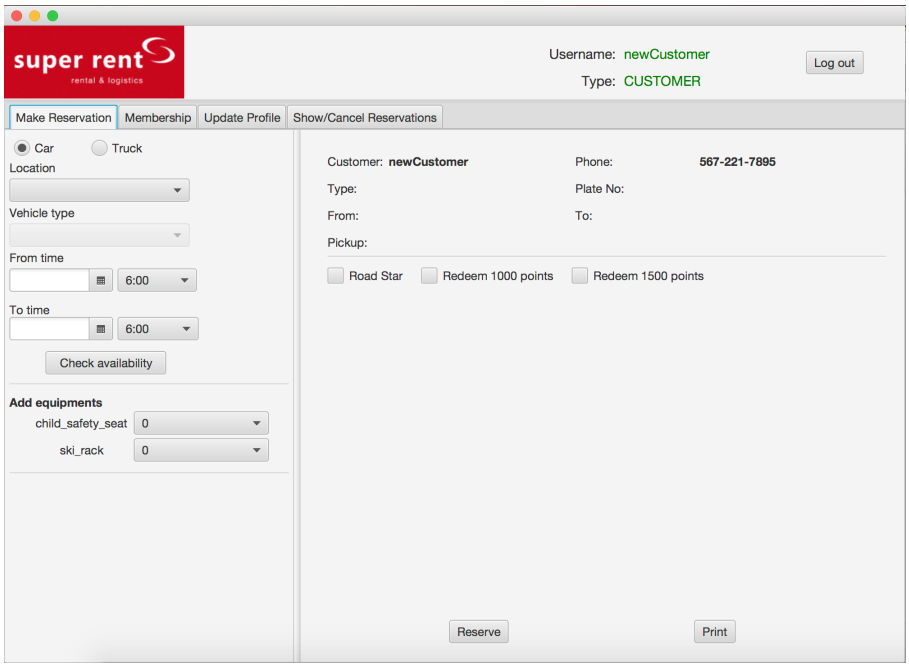
Test Passed.

Appendix B-14: Test Case (Customer: customer login)

User input username and password:



User login successfully:



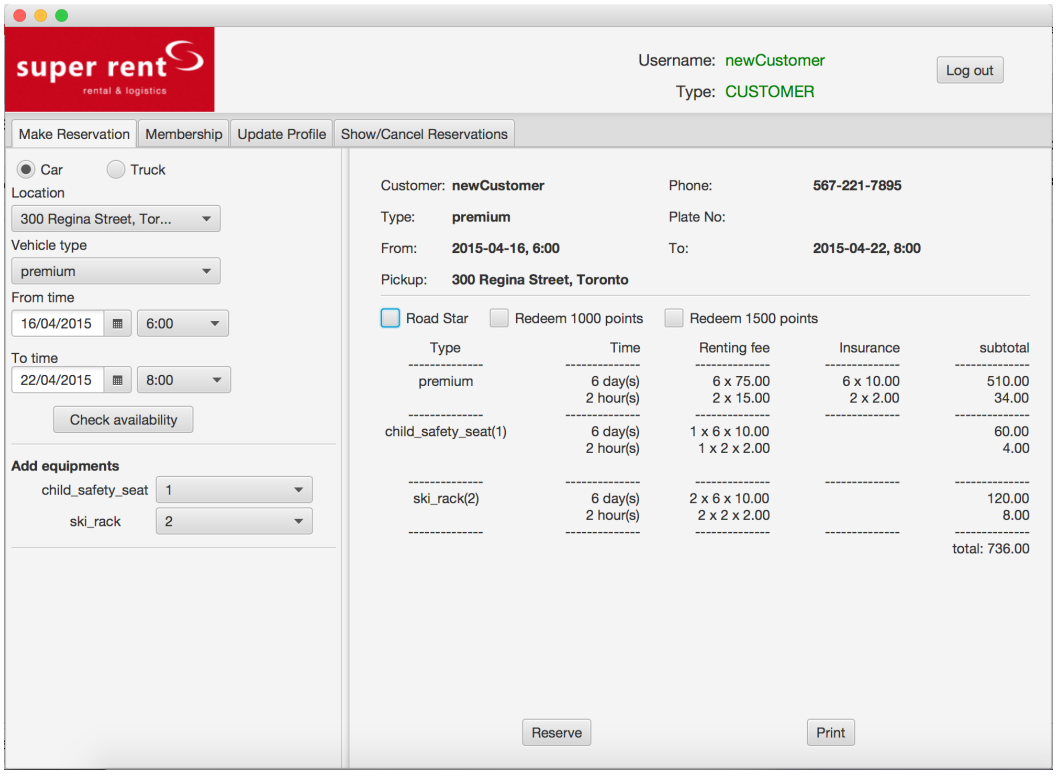
Test Passed.

Appendix B-15: Test Case (Customer: Make reservation)

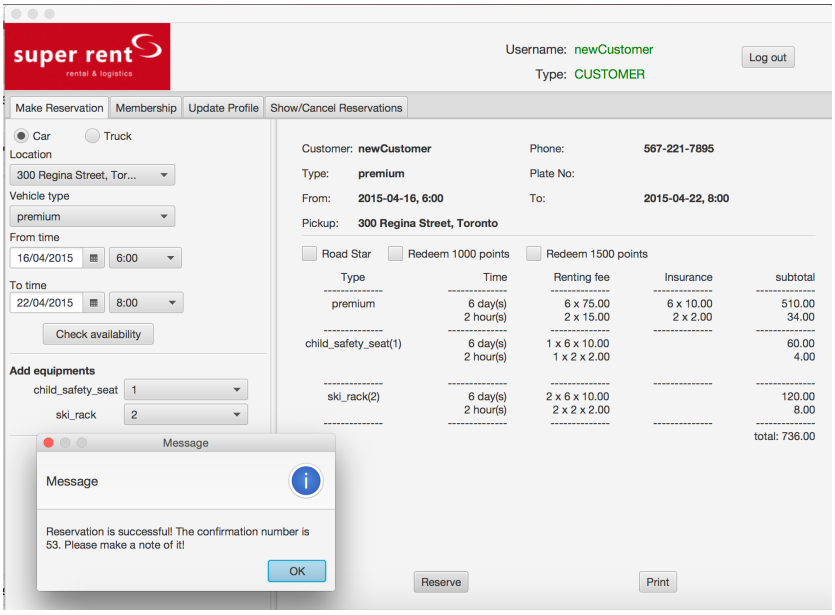
Before the reservation:

confirmation_number	pickup_date	pickup_time	return_date	return_time	branch_city	branch_location	customer_username	status	vehicleT
13	2015-04-09	6	2015-04-30	9	Vancouver	2660 Wesbrook Mall	supfrank	rented	economy
15	2015-04-08	6	2015-04-22	7	Vancouver	2660 Wesbrook Mall	frank8	rented	economy
16	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer26	rented	economy
17	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer27	rented	boxtruck
18	2015-04-10	6	2015-04-23	11	Vancouver	2660 Wesbrook Mall	customer14	pending	midsize
19	2015-04-11	6	2015-04-14	6	Toronto	300 Regina Street	customer	pending	suv
22	2015-04-13	6	2015-04-14	6	Toronto	300 Regina Street	namitha	cancelled	economy
23	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	cancelled	economy
24	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	rented	cargova
25	2015-04-20	6	2015-04-21	6	Toronto	300 Regina Street	namitha	rented	van
26	2015-04-14	6	2015-04-22	9	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
27	2015-04-15	6	2015-04-30	9	Toronto	300 Regina Street	namitha	rented	economy
28	2015-04-17	6	2015-04-30	10	Toronto	300 Regina Street	namitha	rented	suv
29	2015-04-17	6	2015-04-30	10	Vancouver	2660 Wesbrook Mall	namitha	rented	premium
31	2015-04-23	6	2015-04-29	11	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
32	2015-04-15	6	2015-04-23	6	Vancouver	2660 Wesbrook Mall	namitha	rented	luxury
▶ 48	2015-04-16	6	2015-04-21	12	Vancouver	2660 Wesbrook Mall	customer	pending	standard
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Making reservation:



After reservation has been made:



Check database for the reservation:

confirmation_number	pickup_date	pickup_time	return_date	return_time	branch_city	branch_location	customer_username	status	vehicleT
15	2015-04-08	6	2015-04-22	7	Vancouver	2660 Wesbrook Mall	frank8	rented	economy
16	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer26	rented	economy
17	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer27	rented	boxtruck
18	2015-04-10	6	2015-04-23	11	Vancouver	2660 Wesbrook Mall	customer14	pending	midsize
19	2015-04-11	6	2015-04-14	6	Toronto	300 Regina Street	customer	pending	suv
22	2015-04-13	6	2015-04-14	6	Toronto	300 Regina Street	namitha	cancelled	economy
23	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	cancelled	economy
24	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	rented	cargovan
25	2015-04-20	6	2015-04-21	6	Toronto	300 Regina Street	namitha	rented	van
26	2015-04-14	6	2015-04-22	9	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
27	2015-04-15	6	2015-04-30	9	Toronto	300 Regina Street	namitha	rented	economy
28	2015-04-17	6	2015-04-30	10	Toronto	300 Regina Street	namitha	rented	suv
29	2015-04-17	6	2015-04-30	10	Vancouver	2660 Wesbrook Mall	namitha	rented	premium
31	2015-04-23	6	2015-04-29	11	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
32	2015-04-15	6	2015-04-23	6	Vancouver	2660 Wesbrook Mall	namitha	rented	luxury
48	2015-04-16	6	2015-04-21	12	Vancouver	2660 Wesbrook Mall	customer	pending	standard
▶ 53	2015-04-16	6	2015-04-22	8	Toronto	300 Regina Street	newCustomer	pending	premium
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Also, the reserve_addon table is also updated to indicate the reserved equipment.

confirmNo	quantity	equipName
▶ 53	1	child_safety_seat
53	2	ski_rack
NULL	NULL	NULL

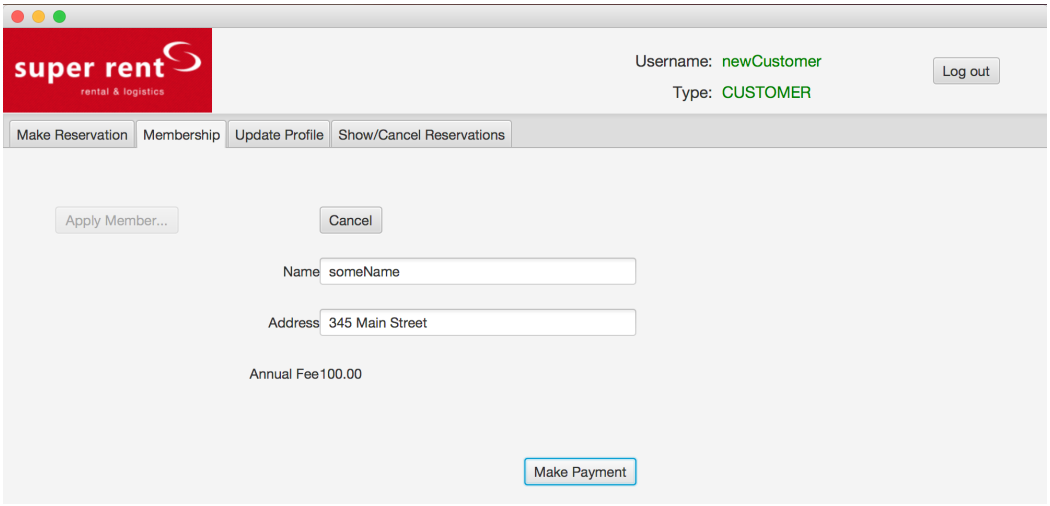
Test Passed.

Appendix B-16: Test Case (Customer: apply membership)

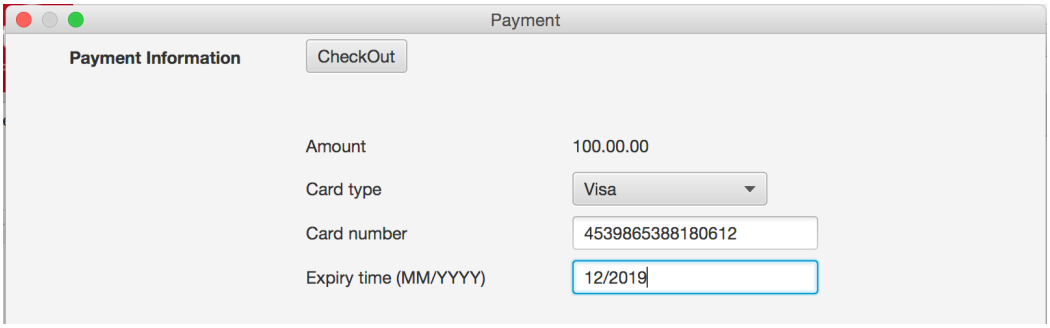
Before the operation, ‘newCustomer’ is not a member

username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
newCustomer	567-221-7895	237 Maple Street	0	0	0	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

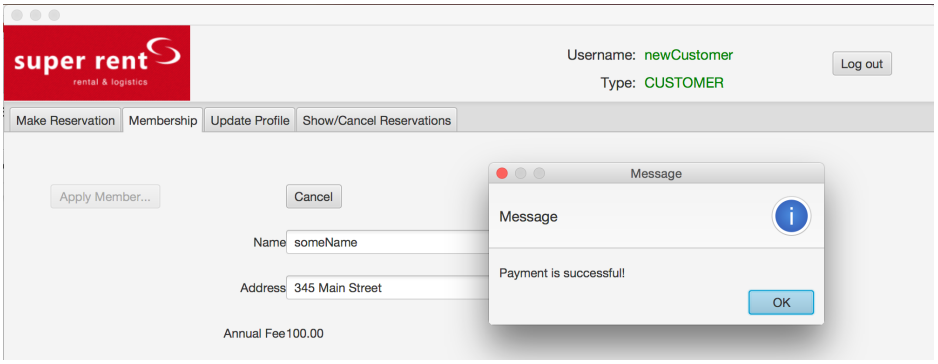
Apply for membership:



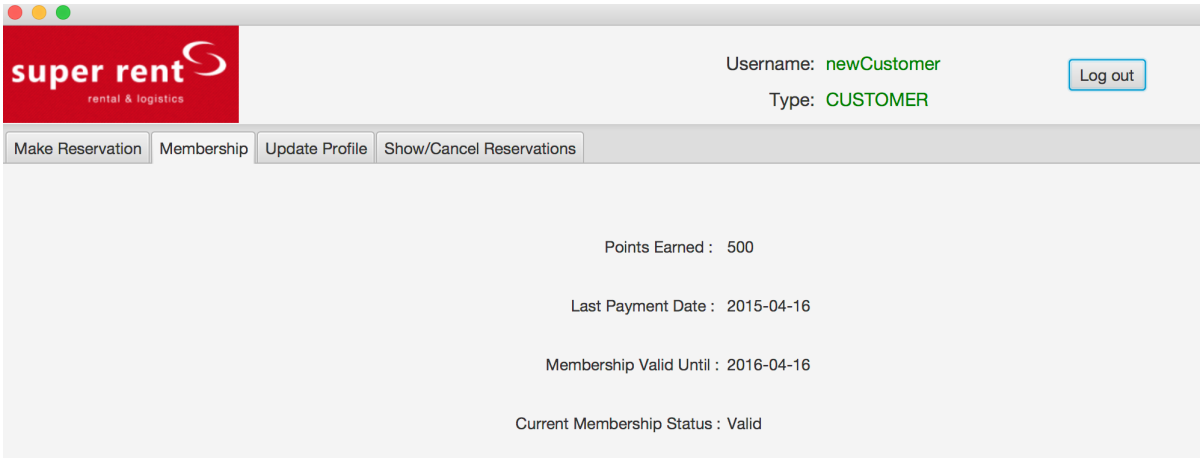
Pay annual fees:



Successful payment:



Show membership information:



After the operation, 'newCustomer' becomes a member

username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
▶ newCustomer	567-221-7895	345 Main Street	0	1	500	NULL	2015-04-16
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

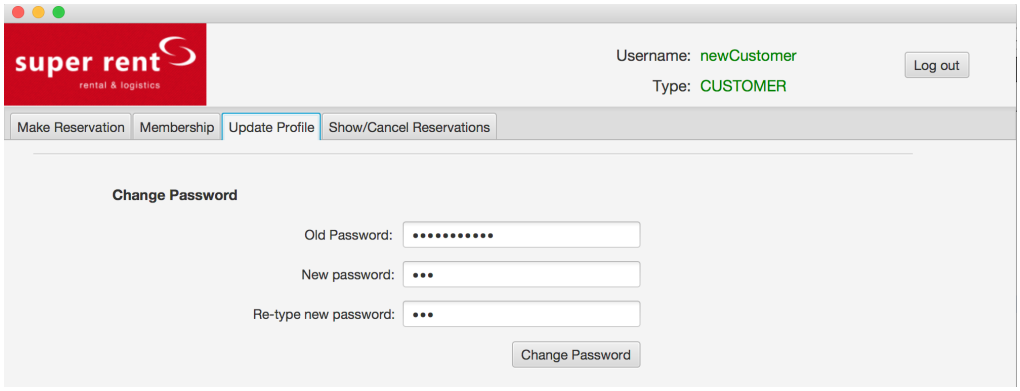
Test Passed.

Appendix B-17: Test Case (Customer: Change password)

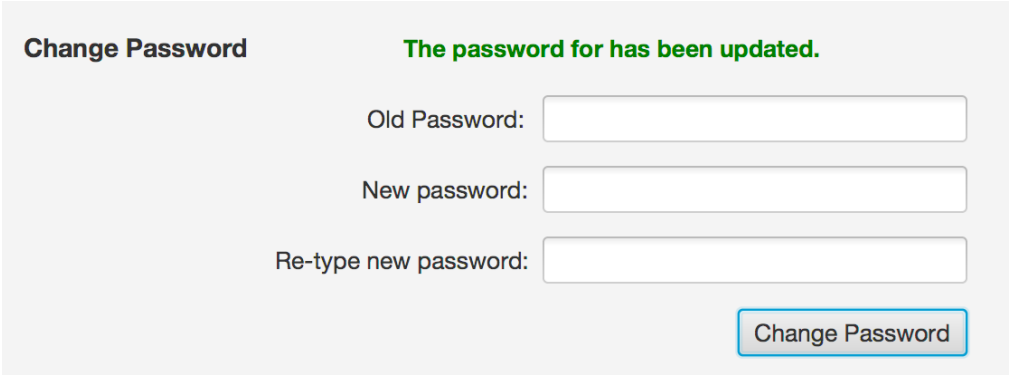
Before the operation, the password is 'newCustomer'

username	password	name	type
newCustomer	newCustomer	someName	customer
NULL	NULL	NULL	NULL

Change password to ‘new’



Successful change:



The database has been updated:

username	password	name	type
newCustomer	new	someName	customer
NULL	NULL	NULL	NULL

Test Passed.

Appendix B-18: Test Case (Customer: Change phone number)

Before the operation, the phone number is 567-221-7895:

	username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
▶	newCustomer	567-221-7895	345 Main Street	0	1	500	NULL	2015-04-16
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Change the phone number to 123-456-1111:

After the operation, the phone number becomes 123-456-1111

	username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
▶	newCustomer	123-456-1111	345 Main Street	0	1	500	NULL	2015-04-16
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

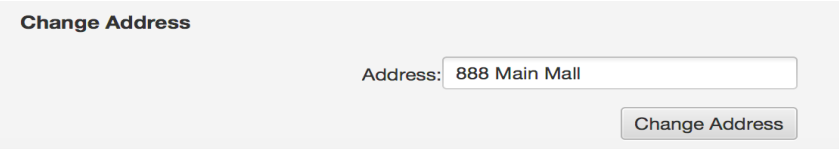
Test Passed.

Appendix B-19: Test Case (Customer: change address)

Before the operation:

	username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
▶	newCustomer	123-456-1111	345 Main Street	0	1	500	NULL	2015-04-16
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Change address:



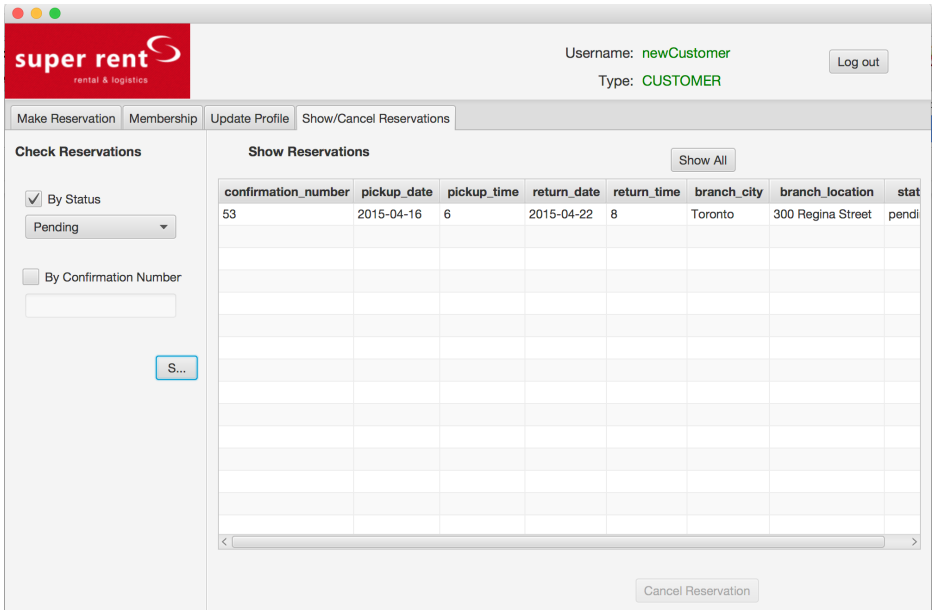
After the operation:

	username	phone	address	isRoadStar	isClubMember	point	isAnnualPaid	payment_date
▶	newCustomer	123-456-1111	888 Main Mall	0	1	500	NULL	2015-04-16
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

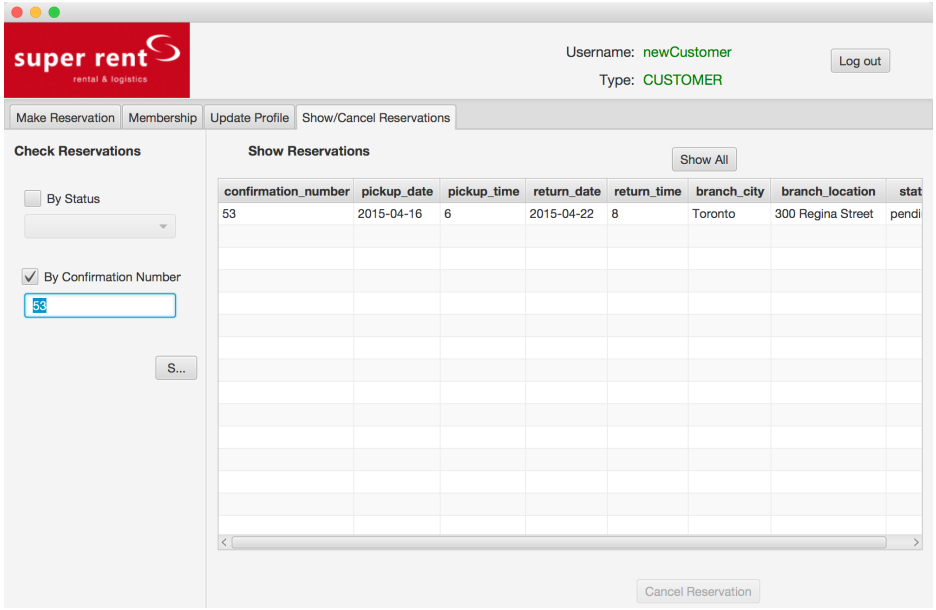
Test Passed.

Appendix B-20: Test Case (Customer: check reservation)

Customer can see his own reservation by status:



Customer can see his own reservation by confirmation number:



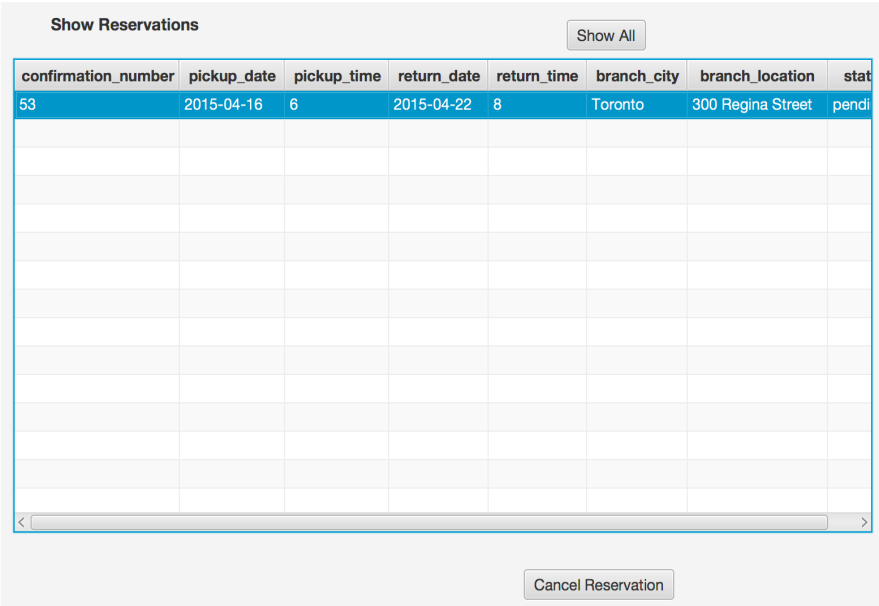
Test Passed.

Appendix B-21: Test Case (Customer: cancel reservation)

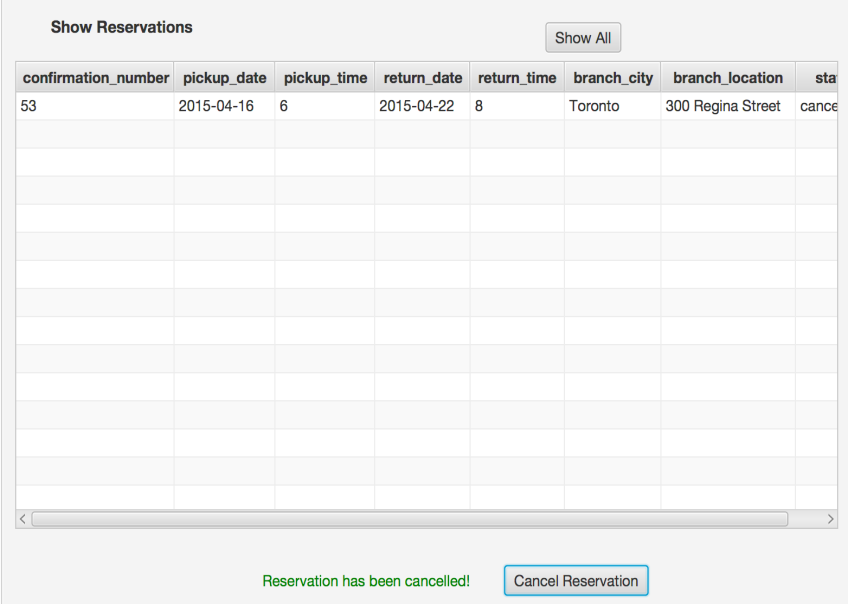
Before the operation:

confirmation_number	pickup_date	pickup_time	return_date	return_time	branch_city	branch_location	customer_username	status	vehicleTyp
15	2015-04-08	6	2015-04-22	7	Vancouver	2660 Wesbrook Mall	frank8	rented	economy
16	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer26	rented	economy
17	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer27	rented	boxtrucks
18	2015-04-10	6	2015-04-23	11	Vancouver	2660 Wesbrook Mall	customer14	pending	midsize
19	2015-04-11	6	2015-04-14	6	Toronto	300 Regina Street	customer	pending	suv
22	2015-04-13	6	2015-04-14	6	Toronto	300 Regina Street	namitha	cancelled	economy
23	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	cancelled	economy
24	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	rented	cargovans
25	2015-04-20	6	2015-04-21	6	Toronto	300 Regina Street	namitha	rented	van
26	2015-04-14	6	2015-04-22	9	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
27	2015-04-15	6	2015-04-30	9	Toronto	300 Regina Street	namitha	rented	economy
28	2015-04-17	6	2015-04-30	10	Toronto	300 Regina Street	namitha	rented	suv
29	2015-04-17	6	2015-04-30	10	Vancouver	2660 Wesbrook Mall	namitha	rented	premium
31	2015-04-23	6	2015-04-29	11	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
32	2015-04-15	6	2015-04-23	6	Vancouver	2660 Wesbrook Mall	namitha	rented	luxury
48	2015-04-16	6	2015-04-21	12	Vancouver	2660 Wesbrook Mall	customer	pending	standard
53	2015-04-16	6	2015-04-22	8	Toronto	300 Regina Street	newCustomer	pending	premium
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Cancel reservation:



Successful cancellation:



After the operation:

confirmation_number	pickup_date	pickup_time	return_date	return_time	branch_city	branch_location	customer_username	status	vehicleTy
15	2015-04-08	6	2015-04-22	7	Vancouver	2660 Wesbrook Mall	frank8	rented	economy
16	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer26	rented	economy
17	2015-04-10	8	2015-04-23	14	Vancouver	2660 Wesbrook Mall	customer27	rented	boxtrucks
18	2015-04-10	6	2015-04-23	11	Vancouver	2660 Wesbrook Mall	customer14	pending	midsize
19	2015-04-11	6	2015-04-14	6	Toronto	300 Regina Street	customer	pending	suv
22	2015-04-13	6	2015-04-14	6	Toronto	300 Regina Street	namitha	cancelled	economy
23	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	cancelled	economy
24	2015-04-14	6	2015-04-15	6	Toronto	300 Regina Street	namitha	rented	cargovans
25	2015-04-20	6	2015-04-21	6	Toronto	300 Regina Street	namitha	rented	van
26	2015-04-14	6	2015-04-22	9	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
27	2015-04-15	6	2015-04-30	9	Toronto	300 Regina Street	namitha	rented	economy
28	2015-04-17	6	2015-04-30	10	Toronto	300 Regina Street	namitha	rented	suv
29	2015-04-17	6	2015-04-30	10	Vancouver	2660 Wesbrook Mall	namitha	rented	premium
31	2015-04-23	6	2015-04-29	11	Vancouver	2660 Wesbrook Mall	namitha	rented	full-size
32	2015-04-15	6	2015-04-23	6	Vancouver	2660 Wesbrook Mall	namitha	rented	luxury
48	2015-04-16	6	2015-04-21	12	Vancouver	2660 Wesbrook Mall	customer	pending	standard
53	2015-04-16	6	2015-04-22	8	Toronto	300 Regina Street	newCustomer	cancelled	premium

Test Passed.