**DELHI TECHNOLOGICAL UNIVERSITY**



**PROJECT**

**Car Rental Management System**

**IT-359a Java Programming**

*By:*

*Ahmad Habib Khan 2K19/SE/003*

*Amritpal Singh 2K19/SE/008*

## INDEX

[**ABSTRACT** **4**](#_Toc44366)

[**PROBLEM STATEMENT** **5**](#_Toc44367)

[**INTRODUCTION** **6**](#_Toc44368)

**FLOWCHARTS AND DIAGRAMS 8**

**WORK DONE 9**

**SCREENSHOTS 10**

**ADVANTAGES AND SHORTCOMINGS 11**

**FUTURE SCOPE 12**

**CONCLUSION 14**

**REFERENCES 15**

# 

# ABSTRACT

Car Rental System is based on a concept to rent cars and generate a rental invoice of a rental company. Before stepping into the main system a user has to pass through a login system to get access, then only the user can select cars with a different model and rent for certain days. This mini project contains limited features, but the essential one.

Talking about the features of the Car Rental System, after logging in as a user he/she has to provide a name then the user can select available cars. After selecting a car, the system displays selected car details which contain maximum power, mileage, and many more. Then the user has to provide information such as Car number and number of days to rent the car. After all these procedures, the system calculates rent and displays Customer Invoice presenting invoice number, customer’s name, car model, number, number of days, and total rental amount.

Car Rental system is developed using JAVA Programming Language and different variables, strings have been used for the development of it.

# PROBLEM STATEMENT

The aim of our project is to create a Java Application which enables the administrator of a movie theater to:

* Add/Delete/Edit Cars and its details
* Add/Delete/Edit Customers and their details
* Add/Delete/Edit Car Bookings and its details

# INTRODUCTION

Car Rental System is based on a concept to rent cars and generate a rental invoice of a rental company. Before stepping into the main system a user has to pass through a login system to get access, then only the user can select cars with a different model and rent for certain days. This mini project contains limited features, but the essential one.

Talking about the features of the Car Rental System, after logging in as a user he/she has to provide a name then the user can select available cars. After selecting a car, the system displays selected car details which contain maximum power, mileage, and many more. Then the user has to provide information such as Car number and number of days to rent the car. After all these procedures, the system calculates rent and displays Customer Invoice presenting invoice number, customer’s name, car model, number, number of days, and total rental amount.

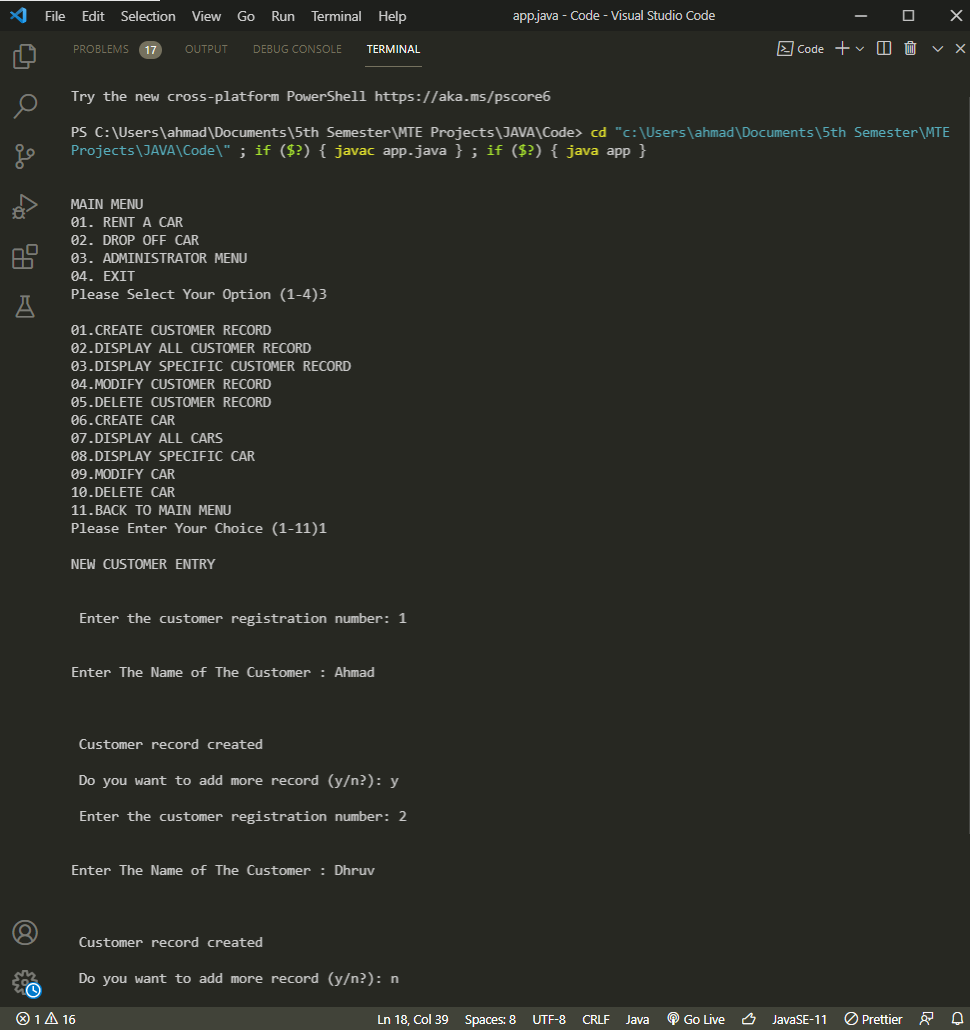
Car Rental system is developed using JAVA Programming Language and different variables, strings have been used for the development of it.

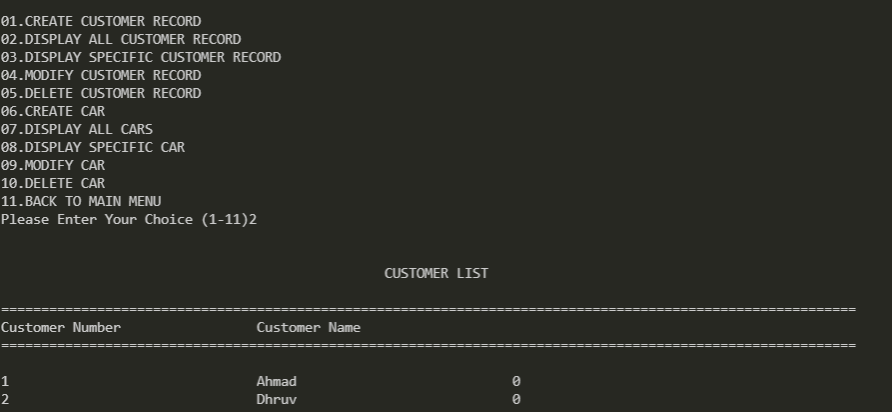
The main objective of the JAVA Project on the Car Rental System is to manage the details of Cars, Passengar Informations, Drivers, Car Bookings, and Invoices. It manages all the information about Cars, Car Routes and the time of rental. The project is totally built at the administrative end and thus only the administrator is guaranteed access. The purpose of the project is to build an application program to reduce the manual work for managing the Cars, Passengar Informations, Car Routes, Drivers, etc.

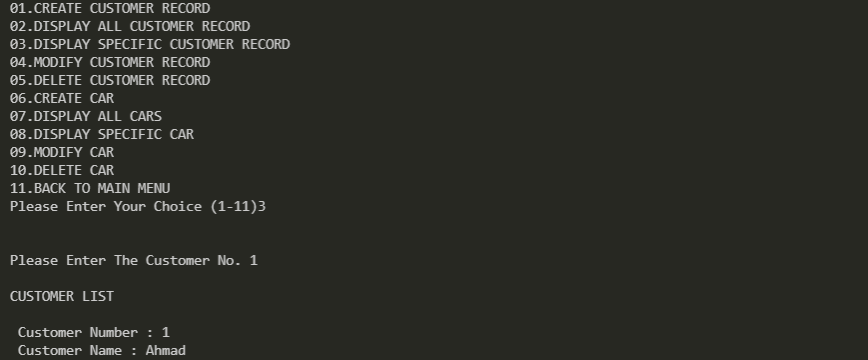
## USE CASE DIAGRAM

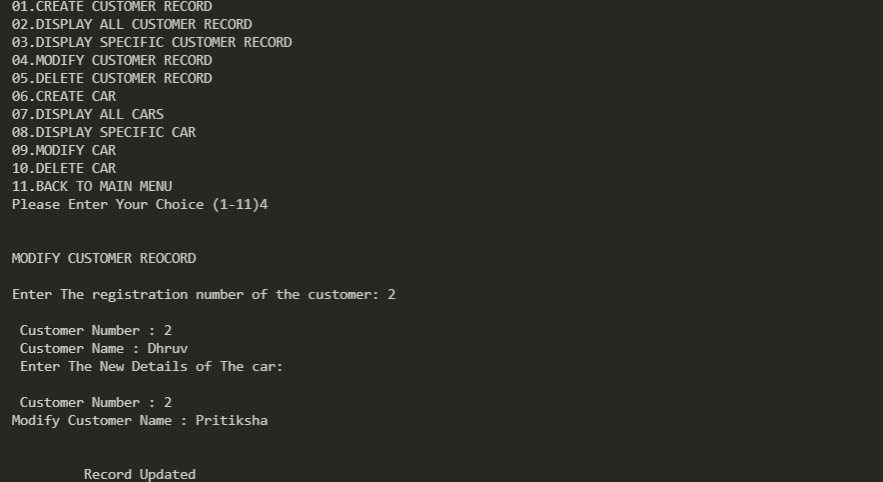
## https://lh3.googleusercontent.com/QqbeVeEB52qt5Ea8KcJNZc2aRZYwQDqGccR5oa-JEZx5rI6iUhnOQvuD47as3b7WmdrBSVMMj0v-C_9WAgeJEqA0cpoD_Ltywh9uBz403eCw-v1F7Qn0nB3N7uwxtEC3TuIgLUzl

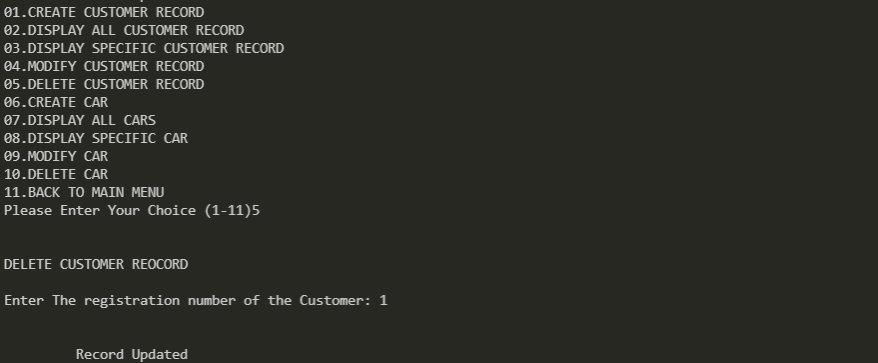
## SCREENSHOTS

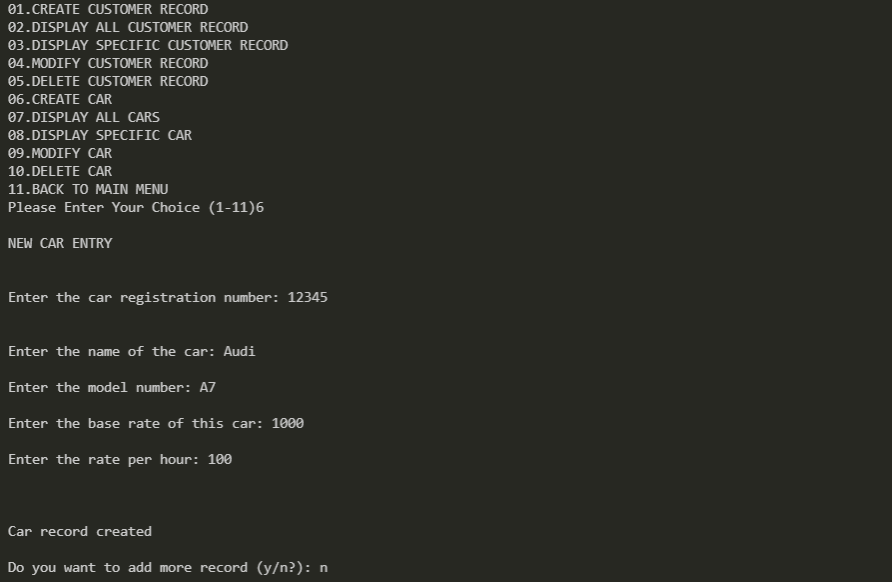


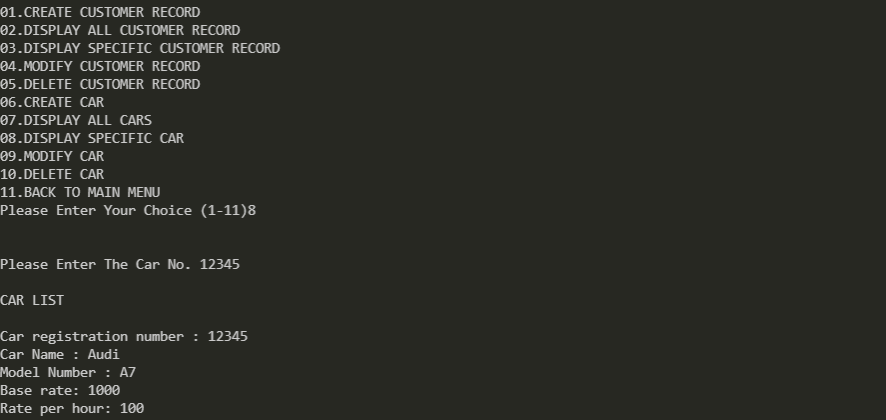
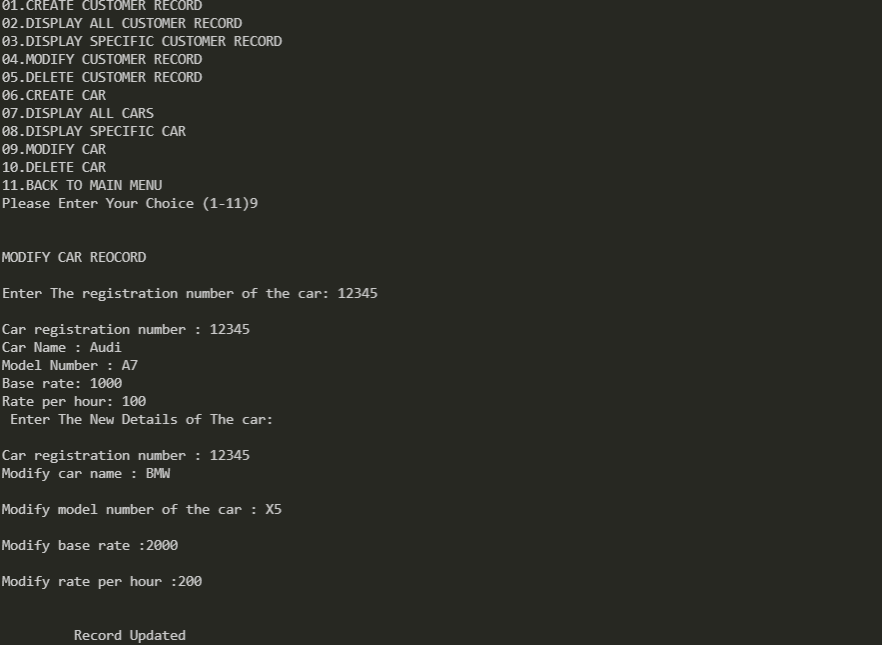
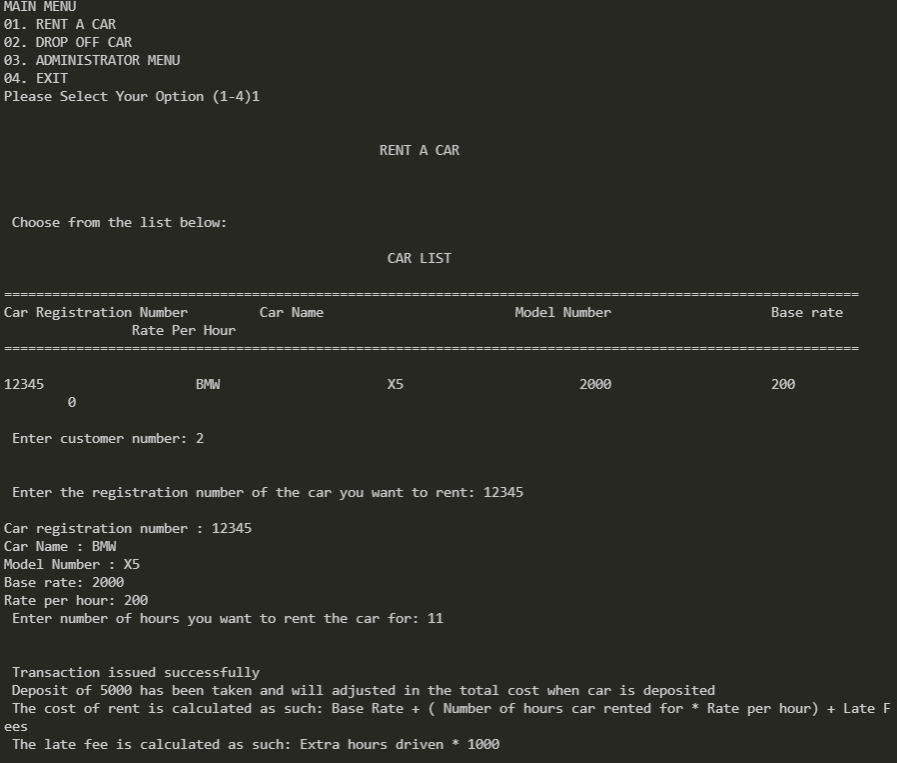


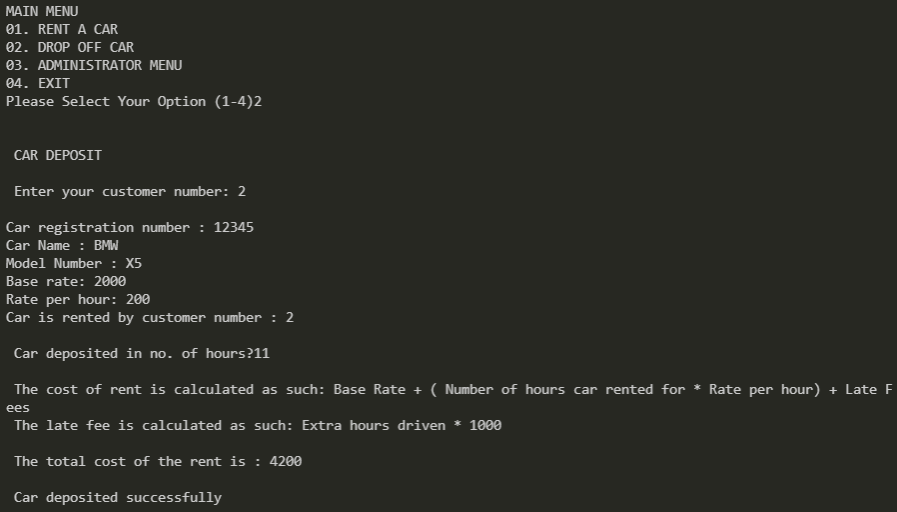










## WORK DONE

Two classes are used:

1. Car class: This class contains the following data members and member functions.
   1. String cno- Car registration number
   2. String cname - Car Name
   3. String mno - Model Number
   4. int baserate - Base rate for rent
   5. int ratephr - Rate per hour for rent
   6. String cuscno - Gets Customer Number of the customer renting the car
   7. int token - To check if the car is currently rented or not
   8. float hr - Gets the number of hours the car is rented
2. Customer class
   1. String cusno - Customer number
   2. String name - Customer name
   3. String cuscno - Gets car number of the car rented by the customer
   4. int token - To check if a car is currently rented by the customer

Modules used:

1. void write\_car() - To write a car record :
   1. This modules will allow admin to record the car details.
   2. The details of the car are as follows:
      1. String cno- Car registration number
      2. String cname - Car Name
      3. String mno - Model Number
      4. int baserate - Base rate for rent
      5. int ratephr - Rate per hour for rent
      6. String cuscno - Gets Customer Number of the customer renting the car
      7. int token - To check if the car is currently rented or not
      8. float hr - Gets the number of hours the car is rented
2. void write\_customer() - To write a customer record:
   1. This modules will allow admin to record the car details.
   2. The details of the car are as follows:
      1. String cusno - Customer number
      2. String name - Customer name
      3. String cuscno - Gets car number of the car rented by the customer
      4. int token - To check if a car is currently rented by the customer
3. void display\_spc(char n[]) - To display specific car record
   1. This module will display specific a car record and will show if car is rented out or not
4. void display\_spcus(char n[]) - To display specific customer record
   1. This module will display specific a customer record and will show if customer has rented a car out or not
5. void modify\_car() - To modify a specific car record
   1. Will modify all details excpet “cusno”
6. void modify\_customer() - To modify a specific customer record
   1. Will modify all details excpet “cno”
7. void delete\_customer() - To delete a specific customer record
   1. Will delete a record when propmted with “cusno”
8. void delete\_car() - To delete a specific car record from file
   1. Will delete a record when prompted with “cno”
9. void display\_allcus() - To display all customer records
   1. Will display all relavent customer records
10. void display\_allc() - To display all the car records
    1. Will display all relavent car records
11. void car\_issue() - To issue a car to a customer
    1. Will rent out a car to customer only if they are not currently renting another car.
12. void car\_deposit() - To deposit a car
    1. Will allow customer to return a car
    2. Deposit of 5000 has been taken and will adjusted in the total cost when car is deposited
    3. The cost of rent is calculated as such: Base Rate + ( Number of hours car rented for \* Rate per hour) + Late Fees
    4. The late fee is calculated as such: Extra hours driven \* 1000"
13. void admin\_menu() - Admin menu
    1. Contains all the above mentiontioned modules from 1-10

## ADVANTAGES

This program has various features and advantages that come by it, which include:

* The simple menu with clear instructions at each step would help the user navigate through the program with ease.
* Any detail that may have an error while being inputted can be edited out easily at any point.
* The program has been thoroughly tested so as to ensure that all possible inputs are accounted for, thereby never allowing the computer to reach an error state.
* All the details of any car, or any customer is accessible quickly and easily through the ‘View all customers and ‘View all cars sections.
* The data is stored in a compact nature, by the use of vectors.

## SHORTCOMINGS

The program has the following shortcomings:

* Doesn’t use file management, so the data that is input by the user isn’t stored after the application has been stopped, i.e. the data that was input in a previous runtime will not be stored for a later runtime.
* The lack of a GUI results in a dull experience using the program.
* Due to the use of vectors and matrices, the memory used isn’t optimal.
* As the applicati0on isn’t password locked, there is no distinction between the users that run the program.
* Movies have to be manually deleted after the screening is done, it isn’t time gated.

## FUTURE SCOPE

This project has quite some scope for improvement, and thus, in the future we plan to implement the following:

* An interactive GUI that is user-friendly and improves the redundancy that occurs in the UI of the program.
* Implementation of file handling, so as to store the data that has been inputted by the user in a more permanent manner.
* A better seating map, that is more interactive than the current one.
* Inclusion of start and end dates for movies. Movies automatically are deleted from the system once it’s expired past its end date.
* Returning Clients to be able to create accounts, so as to not create a new client each time and to store information of previous movies watched.
* Inclusion of password protection.

## CONCLUSION

Movie theatres deal with a lot of data on an everyday basis, data that is usually logged into several logbooks. This is a very inefficient way of running a movie theatre, as this leads to loss of valuable data for the movie theatre due to wearing out of logbooks and paper, inaccuracies in data collection are hard to recheck and rectify, and it is hard to find specific information in these often dusty books, and several other problems.

Our project seeks to rectify this, and creates an all encompassing platform for the administrator of a movie theatre to log in all the relevant details of each movie being shown in the theatre, client’s names, ids and their movie and seating choices, in an easy to access and view manner. This allows the administrator to plan out his business in a much more efficient manner.

Thus, we believe our project code has achieved the required outcome. However, our project still requires much fine-tuning, as has been elaborated in the shortcomings portion of this report and needs to be worked on further, alongside the implementation of future work.