CS 5200 - Database Management Systems
Project Final Report

Car Rental Management System

Group Name :
SujeethaBGunasekaranH

Group Members:
Bharath Sanjay Sujeetha
Haris Gunasekaran

1. README

Specifications

1.Softwares needed :

1. MySQL Workbench
https://dev.mysql.com/downloads/workbench/
2. Java JDK
https://www.oracle.com/java/technologies/downloads/
3. IntelliJ IDE
https://www.jetbrains.com/idea/download/#section=windows

2.Libaries/modules needed:

1.Java JDBC
https://dev.mysql.com/downloads/connector/j/(download the os independent version)

Steps for running the application:

1. Extract the project file.
2. Open up the dump file(Car_rental_Project.sql) that contains tables, database and procedures using mysql workbench.
3. Run the file and refresh the schema section on left to find if “rentaldb” Database is present.
4. Open IntelliJ IDE and create a new Project by going to File-> New -> New Project. Make sure you have JDK installed and select the JDK version you have installed.Then after creation, an empty src folder will be created which will be containing the source code.
5. Create a new directory named lib in the project directory.
6. Download JDBC from MySQL Connector/J drivers at dev.mysql.com(OS independent version).
7. Extract and copy paste the connector jar file into the lib directory.
8. Then include the downloaded jar file into the project by going to File->project structure -> project settings -> modules ->dependencies-> add module(+ sign)-> jar or directories-> and select the jar file in the lib folder.

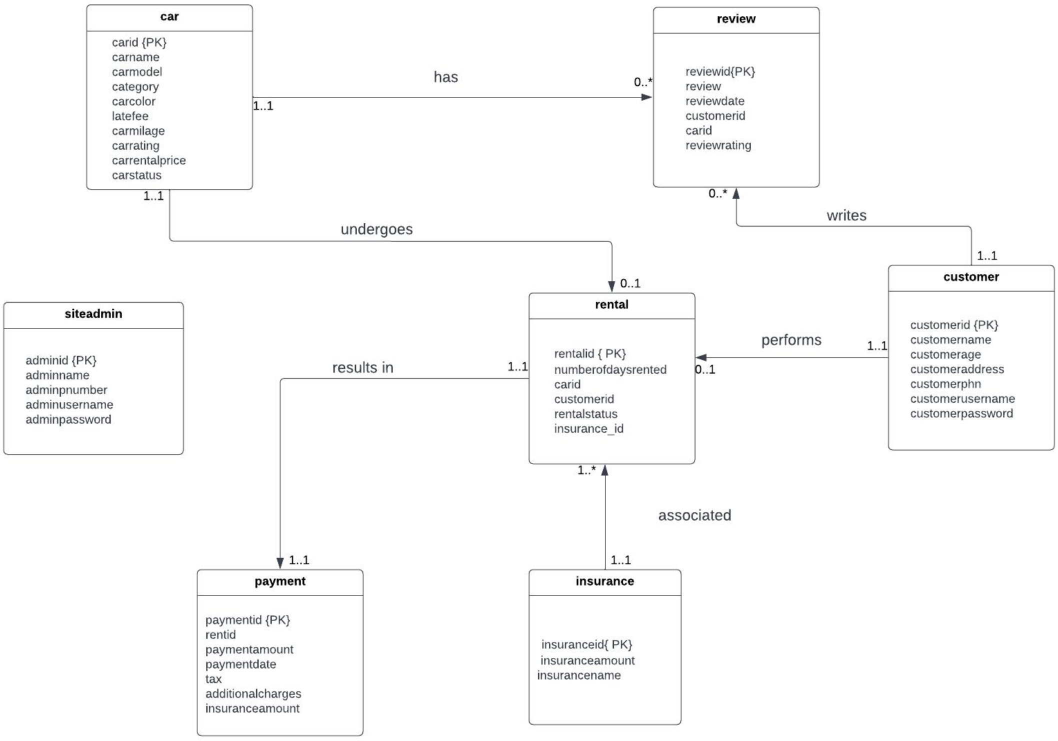
9. copy the Main.java from the project submission folder and paste it into the src folder.
10. See the connection link(In the constructor of the Main.java). The localhost that is connected to mysql is default (3306). Please edit and enter your username and password for the localhost connection as this is required to establish a connection between frontend and backend.

11. Setting up is done. Now run the Main program. The console should open.

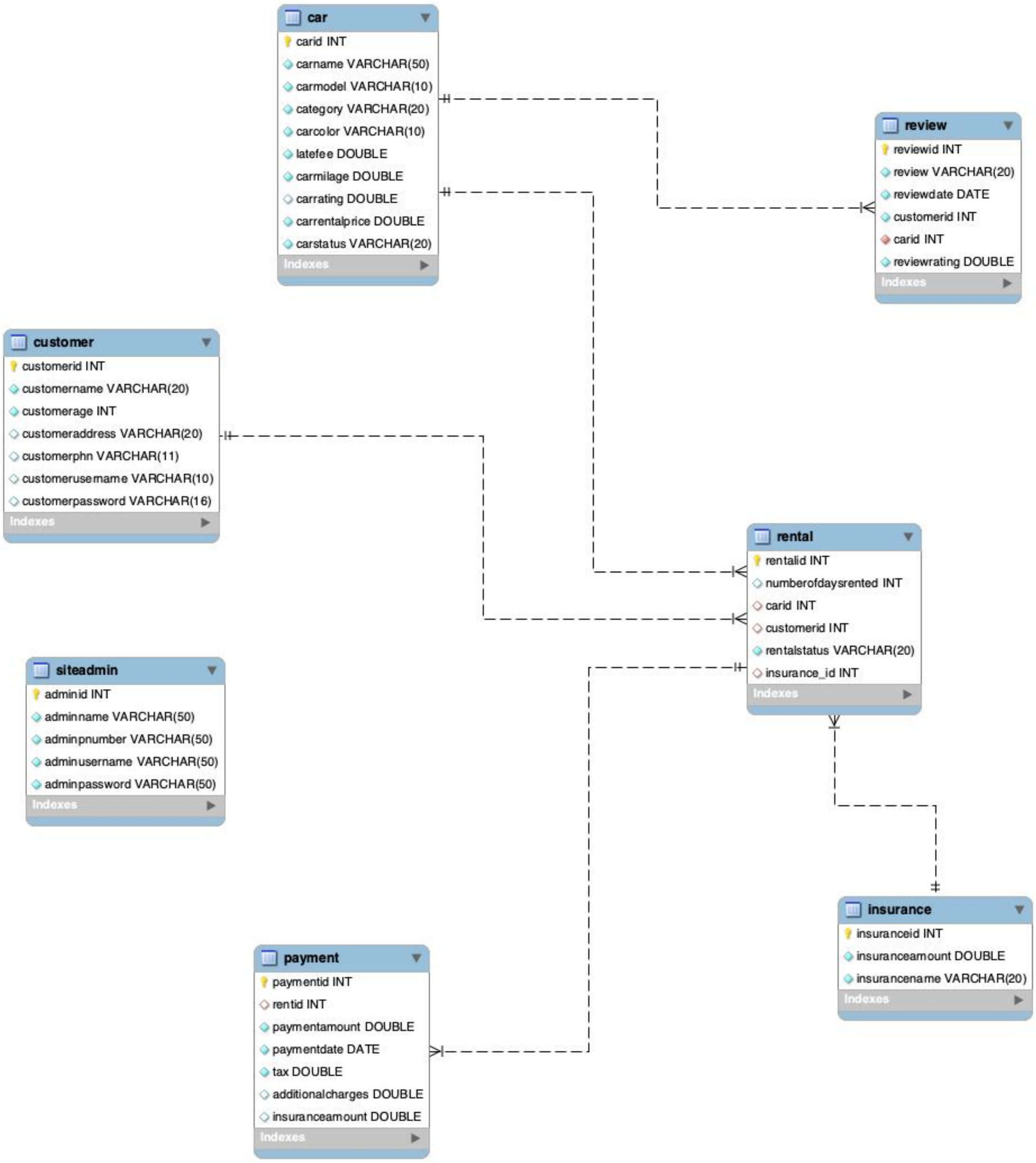
2. TECHNICAL SPECIFICATIONS

- We have created a command line user interface for our project.
- We used java to implement the front end application that is in connection with the database present in the backend.
- All the required storage and database programming objects are created and maintained through MYSQL.
- mySQLWorkbench to work on the data.

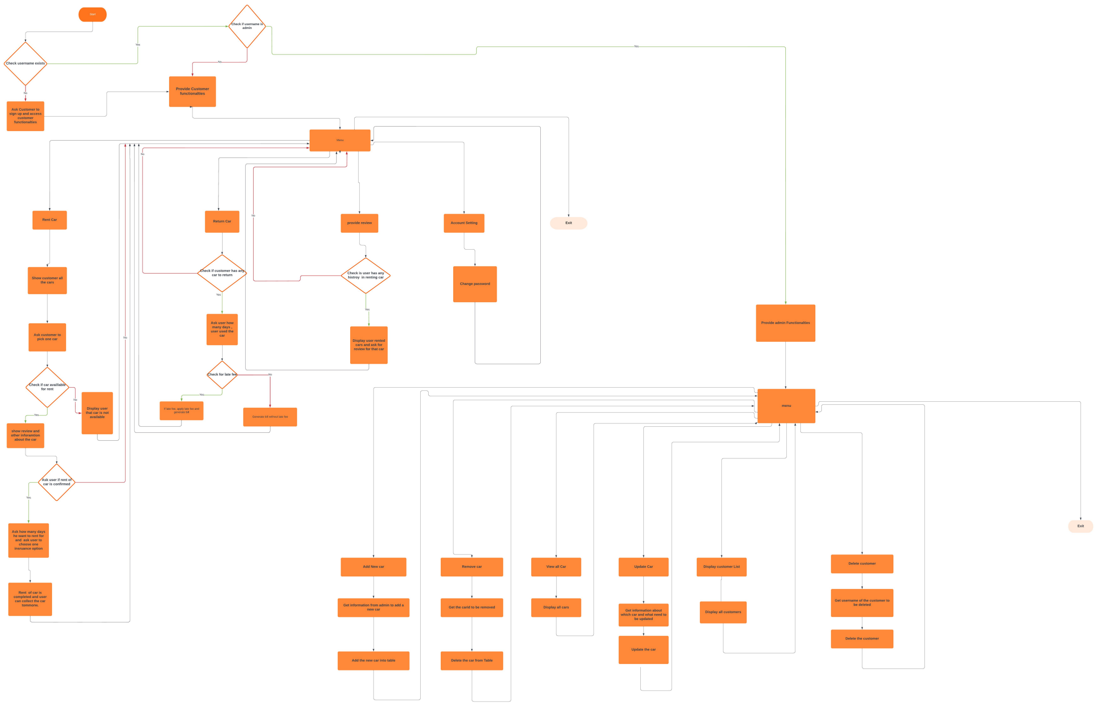
3. CONCEPTUAL DESIGN



4. LOGICAL DESIGN



5. USER FLOW


Start
Check username exists
Yes
No
Check if username is admin
Yes
No
Ask Customer to sign up and access customer functionalties
Provide Customer functionalties
Menu
Rent Car
Show customer all the cars
Ask customer to pick one car
Check if car availlable for rent
Yes
No
Display user that car is not available
show review and other inforamtion about the car
Ask user if rent of car is confirmed
Yes
No
Ask how many days he want to rent for and ask user to choose one insruance option
Rent of car is completed and user can collect the car tommorw.
Return Car
Check if customer has any car to return
Yes
No
Ask user how many days , user used the car
Check for late fee
Yes
No
If late fee, apply late fee and generate bill
Generate bill without late fee
provide review
Check is user has any histroy in renting car
Yes
No
Display user rented cars and ask for review for that car
Account Setting
Change password
Exit
Provide admin Functionalties
menu
Add New car
Get information from admin to add a new car
Add the new car into table
Remove car
Get the carid to be removed
Delete the car from Table
View all Car
Display all cars
Update Car
Get information about which car and what need to be updated
Update the car
Display customer List
Display all customers
Delete customer
Get username of the customer to be deleted
Delete the customer
Exit

1. The application supports operations for 2 users.
2. One user is the admin for the rentaldb database who has access over the entire database and another user is the customer who has access to certain member operations.
3. When a user first enters into the application, he/she can either sign up, login or quit from the application.
4. If a new user wants to register in the gym, they can sign up by providing all their details and will be registered as a member.

5. To login, the user needs to enter their username and password they gave while signing up and will be logged in.
6. If the user who logged in is a admin, their operations will be displayed.
7. If the user who logged in is a member, their operations will be displayed.
8. The operations provided for admin are :
    1. add new cars
    2. remove cars
    3. update car information
    4. view all cars
    5. delete customer
    6. display customers
    7. visualize data of cars
    8. Exit

9. If the admin chooses 1, the program will ask for the car details such as color, model and price/hr etc.
10. If the admin chooses 2, the program will ask for the car id in order to remove the car.
11. If the admin chooses 3, the program asks for valid car id. Then it asks the admin to update which detail about the car by listing choices.
12. If the admin chooses 4, the program will list all the cars in the database.
13. If the admin chooses 5, the program will list all the customers and asks for the customer id to delete.
14. If the admin chooses 6, the program will list all the customers present in the database.
15. If the admin chooses 7, then the program will list all cars asks car id to perform visualization on. It represents the number of times the car has been rented as “*” where each * represents 1 time rent.
16. If the admin chooses 8, then the program stops execution.

17. The operations provided for customer are:
    1. rent car
    2. return car
    3. write review
    4. account settings
    5. exit
18. If the customer chooses 1, the program lists all the cars and asks to choose the car id from the list of all cars. If the car is already booked, the customer will not be able to choose it.
19. If the customer chooses 2, the program will ask to confirm the return,the number of days used and finally the customer can return. If the number of days used is greater than the number of days requested by the customer, then latefee will be also added to the bill.
20. If the customer chooses 3, the customer will be able to write review as well as give rating out of 5 for the car he/she used. This will be stored and listed during the rent car option where other customers can read the reviews.
21. If the customer chooses 4, there will be 1 option. It is to change password.
22. If the customer chooses 5, then the program quits.

6.LESSONS LEARNED

1. Technical expertise gained:

1. As a result of implementing this project, we have realized how powerful and efficient databases are.

2. We managed to learn and utilize the features offered by SQL such as writing DDL and DML commands, Dynamic SQL, Stored Procedures, Functions, Transactions and Database connectivity using the existing programming languages .

3. We learnt how to write different procedures to perform CRUD operations instead of manually performing these operations many number of times.

4.We managed to learn and use the concept of dynamic SQL queries in certain operations which saved us time as well as code redundancy.

2. Insights

1. We ensured that all the tables are created and keys established with all required constraints.

2. While simultaneously writing procedures in the backend, we wrote front end code using java and also verified they are also created properly.

3. We utilized the usage of dynamic SQL in the case where admin has to update the car details. This usage of dynamic SQL saved us a lot of time.

4.Initially, we started off with one user for our database. Later we introduced the idea of a adminstrator. So we made our database accessible to two users but made sure they have access to only what is required and allowed for them as per the design.

3. Alternative design/Approaches

1. In the project proposal, we had only customer as the only user type and designed the system. Later during the implementation we realized the usage and imprortance of an admin in order to manage the data present in the database and hence added another user called admin.

2. We also came up with new entity called user review in order for the customers to write reviews for the cars they used which will be available for other users to read . This entity gets associated with the car entity.

3. We also added a new functionality where users can update their credentials(username and password) and login with their updated credentials.

7.FUTURE WORK

1. Potential areas for added functionality

As part of future work, we are planning to include a variety of features that enhances the application and its usages which includes :

1. upgrading the console based program to an UI based that is compatible for both web and mobile.

2. implementation of payment protocols such as Stripe to support card payments.

3. Incorporating maps to locate the car and for live tracking,and hosting the application on online servers.

4. Generating alerts to the user as message or email when it’s the due day to return car.

2. Planned use of the database

1. Since there can be many users of a car rental database, we have made sure that we could all customer, admin information, all bookings and tracking that the user requires is stored and updated properly in our database.

2. Each booking of the car by the customer is associated with only one car reservation at a time.

3. The customer may look up for availability for cars and book them anytime

4. The customers can also post their reviews and can update their Profile as well as passwords anytime they want from the console. Admin can Add/Manage car brands, manage cars, bookings, reviews, and many more.