CAR RENTAL SYSTEM

**INTRODUCTION**

We have chosen to produce a Car Rental system. In our system, Customer can rent a car based on make and a model. A rental service is a service which customers arrive to request the hire of a rental unit. It is more convenient than carrying the cost of owning and maintain the unit. A car rental is a company that rent automobiles for short period of time for a fee for few hours or a few days or a week.

It helps to book the cars or vehicles online rather than using the traditional manual system of vehicle reservation. This eliminates the risk of erroneous booking and reduce overall lead time and ensures growth in customer satisfaction. They can book any car according to their brands and price.

**OBJECTIVE**

* To transform the manual process of hiring car to a computerize system
* To validate the Rental car system using user satisfaction test
* To produce the documentation such as Software Requirement Specification (SRS), Software Design Description as system development reference

**Tools Used:**

* **Xampp:**
* **Apache:**
* (Application Server) Apache , often referred to as Server, is an open-source Java Servlet Container developed by the Apache Software Foundation.
* **MySqlServer:**
* It handles larege databases much faster than existing solutions.
* It consists of multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and application programming interfaces (APIs)
* Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.
* **Sublime Text 3.1.1-** Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.
* **Web browsers**: Google Chrome,Mozilla Firefox, Opera and Internet Explorer.
* **Git Hub**: GitHub Inc. is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features.

**ENTITIES**

1. Customer:

Customer will be the one who is using car rental system for reserving a car. He can be a member of the system or a non-member of the system. Member of the system will have membership id. Customer entity will store details like customer driving license number, email, address, name, and phone number.

1. Car:

Car entity will have list of cars available in the system. Each car will be associated with a car category and car will have attributes like make, model, mileage and registration number. Car will also have separate flag to check the availability of the car.

1. Car Category:

Every car has a car category. Price is calculated based on the car category. Car category will have attributes like no of person, no of luggage’s, name, and cost per day and late fee per hour.

1. Location:

Location entity here denotes the pickup and drop off location of the car. Customer can pick up the car from the particular location and can have same or different drop off location. Location will have attributes like Location id, name and address.

1. Booking:

Each car reservation will be monitored in the entity called booking. Booking will have attributes like booking id, from date and time of booking and due return date and time and actual return date and time of the booking, and booking status. This booking amount might also include rental insurance and discount code.

1. Billing:

When a customer returns a car, a bill will be generated on the particular booking. Billing have attributes like Bill ID, bill date, bill status, total late fee, tax amount, and total amount.

g) Discount:

Customer can apply discount code while the bill is generated. Each discount code has different discount percentage. Discount will have attributes like discount code, name, expiry date and discount percentage.

h) Car Rental Insurance:

Customer may already have car rental insurance or can buy one while booking the car. Car rental insurance will have attributes like insurance code, coverage type, name and cost per day.

**RELATIONS:**

1. Car to Car Category:

Every car is associated with a car category. Once customer selects a car, the cost per day is obtained from the car category that the selected car belongs to. The relation name is ‘Belongs to’.

1. Car to Location:

Customer will be picking up or dropping the car in a particular location. Customer can pick up or drop-off the car at the particular location. So, cars will be present at a location. The relation name is ‘Current location’

1. Booking to Billing:

Once customer returns a car bill will be generated for each booking. There can be case like booking is cancelled in that case no bill will be associated with the booking. The relation name is ‘Gives’.

1. Booking to Discount:

Customer may apply a discount code when he/she books a car. This discount will be applied to the total amount after tax and late fee while the bill is generated. Based on the discount code total amount will be reduced by some percentage. The relation name is ‘Has’.

1. Booking to Car Rental Insurance:

Customer can select rental insurance while booking a car so that rental insurance will cover damages based on the coverage type. The relation name is ‘Includes’.

1. Booking to Location:

Customer can pick a car for rent from a particular location. The relation name is ‘Pick up location’.

1. Booking to Location:

Customer can drop off rental car in a particular location. The relation name is ‘Drop off location’.

1. Customer to Car to Booking:

Customer will select car for rent. So the customer will be related to the both car and the booking. The relation between these 3 entities is a ternary relation and the relation name is ‘Rents’.

**ER DIAGRAM:**

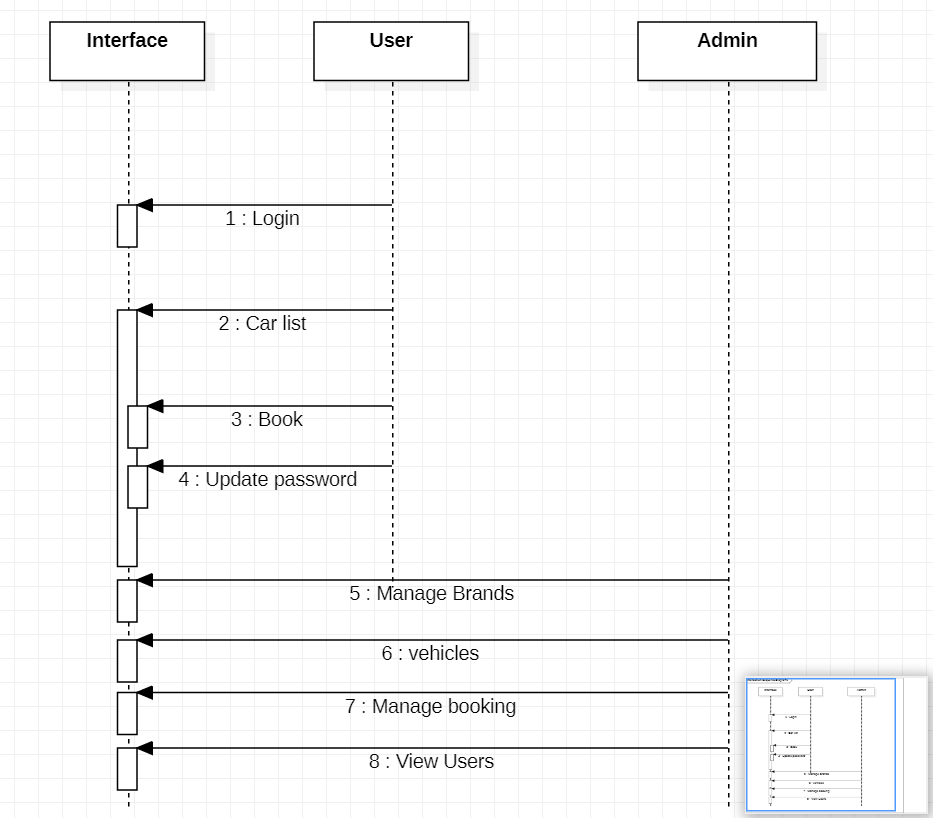
ER diagram show all the relationships between entity sets stored in the database. It illustrates the logical structure of the database. It helps to visualize how data is connected in general ways.

Diagram

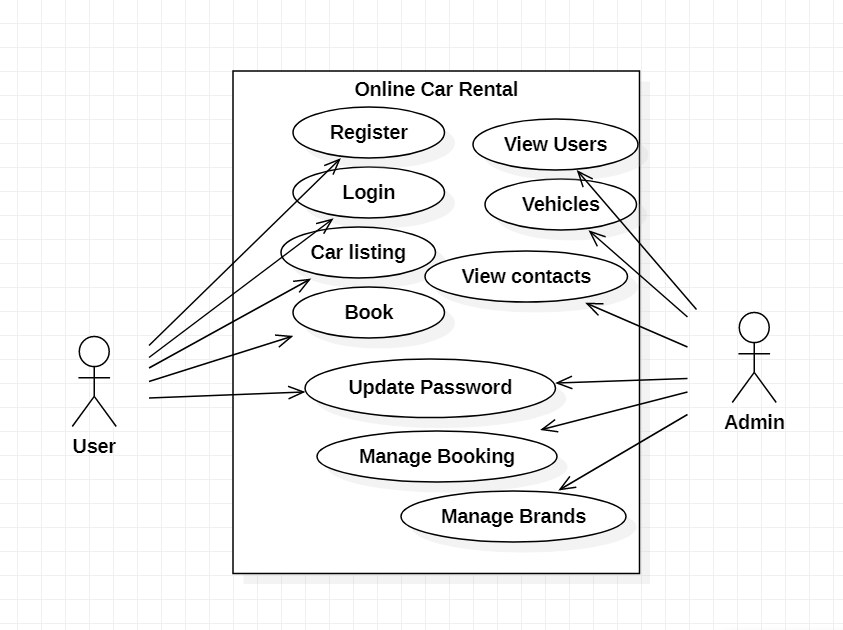
Description automatically generated

**SEQUENCE:**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. A sequence diagram specifically focuses on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.



**USE CASE:**



**SQL STATEMENTS:**

1. Customer\_details:

CREATE TABLE CUSTOMER\_DETAILS

( DL\_NUMBER CHAR(8) NOT NULL,

FNAME VARCHAR(25) NOT NULL,

MNAME VARCHAR(15),

LNAME VARCHAR(25) NOT NULL,

PHONE\_NUMBER NUMBER(10) NOT NULL,

EMAIL\_ID VARCHAR(30) NOT NULL,

STREET VARCHAR(30) NOT NULL,

CITY VARCHAR(20) NOT NULL,

STATE\_NAME VARCHAR(20) NOT NULL,

ZIPCODE NUMBER(5) NOT NULL,

MEMBERSHIP\_TYPE CHAR(1) DEFAULT 'N' NOT NULL,

MEMBERSHIP\_ID CHAR(5),

CONSTRAINT CUSTOMERPK

PRIMARY KEY (DL\_NUMBER)

);

1. CAR\_CATEGORY

CREATE TABLE CAR\_CATEGORY

( CATEGORY\_NAME VARCHAR(25) NOT NULL,

NO\_OF\_LUGGAGE INTEGER NOT NULL,

NO\_OF\_PERSON INTEGER NOT NULL,

COST\_PER\_DAY NUMBER(5,2) NOT NULL,

LATE\_FEE\_PER\_HOUR NUMBER(5,2) NOT NULL,

CONSTRAINT CARCATEGORYPK

PRIMARY KEY (CATEGORY\_NAME)

);

1. LOCATION\_DETAILS

CREATE TABLE LOCATION\_DETAILS

( LOCATION\_ID CHAR(4) NOT NULL,

LOCATION\_NAME VARCHAR(50) NOT NULL,

12

STREET VARCHAR(30) NOT NULL,

CITY VARCHAR(20) NOT NULL,

STATE\_NAME VARCHAR(20) NOT NULL,

ZIPCODE NUMBER(5) NOT NULL,

CONSTRAINT LOCATIONPK

PRIMARY KEY (LOCATION\_ID)

);

1. CAR

CREATE TABLE CAR

( REGISTRATION\_NUMBER CHAR(7) NOT NULL,

MODEL\_NAME VARCHAR(25) NOT NULL,

MAKE VARCHAR(25) NOT NULL,

MODEL\_YEAR NUMBER(4) NOT NULL,

MILEAGE INTEGER NOT NULL,

CAR\_CATEGORY\_NAME VARCHAR(25) NOT NULL,

LOC\_ID CHAR(4) NOT NULL,

AVAILABILITY\_FLAG CHAR(1) NOT NULL,

CONSTRAINT CARPK

PRIMARY KEY (REGISTRATION\_NUMBER),

CONSTRAINT CARFK1

FOREIGN KEY (CAR\_CATEGORY\_NAME) REFERENCES

CAR\_CATEGORY(CATEGORY\_NAME),

CONSTRAINT CARFK2

FOREIGN KEY (LOC\_ID) REFERENCES LOCATION\_DETAILS(LOCATION\_ID)

);

1. Booking\_details

CREATE TABLE BOOKING\_DETAILS

( BOOKING\_ID CHAR(5) NOT NULL,

FROM\_DT\_TIME TIMESTAMP NOT NULL,

RET\_DT\_TIME TIMESTAMP NOT NULL,

AMOUNT NUMBER(10,2) NOT NULL,

BOOKING\_STATUS CHAR(1) NOT NULL,

PICKUP\_LOC CHAR(4) NOT NULL,

DROP\_LOC CHAR(4) NOT NULL,

REG\_NUM CHAR(7) NOT NULL,

DL\_NUM CHAR(8) NOT NULL,

INS\_CODE CHAR(4),

ACT\_RET\_DT\_TIME TIMESTAMP,

DISCOUNT\_CODE CHAR(4),

CONSTRAINT BOOKINGPK

PRIMARY KEY (BOOKING\_ID),

CONSTRAINT BOOKINGFK1

FOREIGN KEY (PICKUP\_LOC) REFERENCES LOCATION\_DETAILS(LOCATION\_ID),

CONSTRAINT BOOKINGFK2

FOREIGN KEY (DROP\_LOC) REFERENCES LOCATION\_DETAILS(LOCATION\_ID),

CONSTRAINT BOOKINGFK3

FOREIGN KEY (REG\_NUM) REFERENCES CAR(REGISTRATION\_NUMBER),

CONSTRAINT BOOKINGFK4

FOREIGN KEY (DL\_NUM) REFERENCES CUSTOMER\_DETAILS(DL\_NUMBER),

CONSTRAINT BOOKINGFK5

FOREIGN KEY (INS\_CODE) REFERENCES RENTAL\_CAR\_INSURANCE(INSURANCE\_CODE),

1. Billing\_details

CREATE TABLE BILLING\_DETAILS

( BILL\_ID CHAR(6) NOT NULL,

BILL\_DATE DATE NOT NULL,

BILL\_STATUS CHAR(1) NOT NULL,

DISCOUNT\_AMOUNT NUMBER(10,2) NOT NULL,

TOTAL\_AMOUNT NUMBER(10,2) NOT NULL,

TAX\_AMOUNT NUMBER(10,2) NOT NULL,

BOOKING\_ID CHAR(5) NOT NULL,

TOTAL\_LATE\_FEE NUMBER(10,2) NOT NULL,

CONSTRAINT BILLINGPK

PRIMARY KEY (BILL\_ID),

CONSTRAINT BILLINGFK1

FOREIGN KEY (BOOKING\_ID) REFERENCES BOOKING\_DETAILS(BOOKING\_ID)

);

**FLOWCHART:**

