Data Management and

Database Design

**CAR RENTAL DATABASE MANAGEMENT SYSTEM**

Contents

[Team Members 3](#_TOC_250004)

[Project Topic 3](#_TOC_250003)

[Problem Statement 3](#_TOC_250002)

[Objectives 3](#_TOC_250001)

Proposed Solution for identified problem 4

[Entity Relationship Diagram 5](#_TOC_250000)

Database Dictionary 6

Business Rules………………………………………………………………………………………………………………………….12

Access and permissions 13

Views 15

# Team Members

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name** | **Email Id** |
| 1 | Bhawna Singh | singh.bhaw[@northeastern.edu](mailto:qiu.min@northeastern.edu) |
| 2 | Yashwanth Kolla | [kolla.y@northeastern.edu](mailto:kolla.y@northeastern.edu) |
| 3 | Shereen John | [john.sh@northeastern.edu](mailto:john.sh@northeastern.edu) |
| 4 | Uday Kumar Reddy Somireddy | [somireddy.u@northeastern.edu](mailto:somireddy.u@northeastern.edu) |
| 5 | Sanket Khadke | [khadke.s@northeastern.edu](mailto:khadke.s@northeastern.edu) |
| 6 | Madhava Potturu | [potturu.m@northeastern.edu](mailto:potturu.m@northeastern.edu) |

# Project Topic



# Problem Statement

* For many people, buying a car and bearing its insurance cost would be expensive

## For a traditional car rental service, a customer must physically visit the rental store which is inefficient

* In our car rental system, a user can book a car ride from any place conveniently, and after the payment is made, s/he will be notified of the nearest location from where s/he can pick up the car
* Due to high availability of these locations, the user can get to these locations in a short duration of time
* User can then use the car for the specified session and then return it to one of the closest locations

# Objectives

To design a database for the car rental platform which should support the following functionalities:

* Users must be able to register into the system with details such as username, email, address, etc., and then login using those credentials
* Users must be able to view the current availability of cars for rent in his/her nearby location
* Users must be able to book a car followed by the payment of the session
* Ride details must be stored, and users must be able to view the booking details
* Maintain details of car location such as parking spot, area name, zip code, etc.

Proposed Solution for the identified problems

* 1. **Difficulty in tracking daily transactions and increased chance of error and complexity in maintaining them**

To overcome the above problem, a database management system is created for the car rental system, where instead of maintaining the data manually, the data will be logged in its respective entity tables. This will facilitate efficient managing of the structured data. Objects such as indexes, triggers, views, etc. would be used which would fine tune the system as well as maintain the ETL process of the database.

* 1. **Same data is being entered daily, which is a redundant process (car, location data)**

Redundancy is being eliminated by:

* + 1. Passenger data need not be entered manually every time the passenger books a car for rent. All the required customer data will be stored in the "passenger" table.
    2. All the required details about cars available in the rental system will be stored in the tables "car" and "car category".
    3. Database will ensure that at a time, no two reservations on the same car are being made.
    4. Tables will be normalized to ensure data consistency and uniqueness.
  1. **No database available for the passenger, car, and the location of car spots**

Three different tables for passenger, car and location have been created. Each table, **passenger**, **car,** and **location** have their unique ids, as **passenger\_id**, **license\_no** and **location\_id,** respectively. These unique ids will help determine each of the passengers, car, and location uniquely.

* 1. **There is no system of booking a car for a certain time online**

“Reservation” entity is created in the database to book a car online, data in this table has **reservation\_id** attribute as primary key to maintain uniqueness. The reservation table has the attributes **passenger\_id**, **car\_id** and **parking\_lotno** as foreign keys that specify what car a passenger has booked and in which parking lot would he/she find the required car.

* 1. **Audit of transactions done by customer bookings is difficult**

With the creation of the database, auditing of past transactions is easy to track.

The table ‘transaction’ keeps the records of all the transactions done. The attribute **transaction\_id** identifies a specific transaction, the foreign key **reservation\_id** stores the ID of the reservation which was done by a particular passenger. It also stores information in attributes like **trip\_amount**, **trip\_invoice**, and **transaction\_date** which stores the total trip amount, the invoice number generated, and the date of transaction, respectively.

# Entity – Relationship Diagram

# 

Database Dictionaries

**Database details for Car Rental System**

***Passenger***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| passengerID | Unique ID assigned to each  passenger | NUMBER |  | Primary Key |
| fname | Passenger’s first  name | VARCHAR | 45 | Not Null |
| lname | Passenger’s last  name | VARCHAR | 45 | Not Null |
| email | Passenger’s email  id | VARCHAR | 45 | Not Null |
| drivers\_license | Passenger’s  driver’s license  number | VARCHAR | 20 | Not Null |
| date\_of\_birth | Passenger’s date  of birth | DATE |  | Not Null |
| location | Passenger’s  location | VARCHAR | 45 | Not Null |

***Reservation***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| reservationID | Unique ID assigned to each  reservation made | NUMBER |  | Primary Key |
| pickup\_date | Time when the  ride starts | DATETIME |  | Not Null |
| return\_date | Time when the  ride ends | DATETIME |  | Not Null |
| carID | Foreign key |  |  | Not Null |
| passengerID | Foreign key |  |  | Not Null |
| lotNO | Foreign key |  |  | Not Null |

***Car***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| carID | Primary Key |  |  |  |
| licNO | License number | VARCHAR | 15 | Primary Key |
| brand | Make of car | VARCHAR | 50 | Not Null |
| mfgyear | Manufactured  year | VARCHAR | 50 | Not Null |
| color | Car color | VARCHAR | 50 | Not Null |
| car\_category | Car category | VARCHAR | 9 | Not Null |
| car\_model | Model of the car | VARCHAR | 50 | Not Null |

***Transaction***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| transactionID | Transaction ID | INT |  | Primary Key |
| trip\_amount | Trip amount | FLOAT |  | Not Null |
| trip\_invoice | Trip invoice | VARCHAR | 15 | Not Null |
| transaction\_date | Transaction date | DATE |  | Not Null |
| status | Status of  transaction | VARCHAR | 15 | Not Null |
| payment\_type | Payment type | VARCHAR | 35 | Not Null |
| reservationID | Foreign Key |  |  |  |

***Customer\_support***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| ticketNO | Ticket ID identifying ticket  uniquely | INT |  | Primary Key |
| tkt\_toc | The time, the ticket was created | TIMESTAMP |  |  |
| ticketStatus | Whether the ticket  is open or closed | VARCHAR | 5 | Not Null |
| empID | Foreign key |  |  | Not Null |
| passengerID | Foreign key |  |  | Not Null |
| categoryID | Foreign key |  |  |  |

***Parking lot***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| lotNO | Parking lot  number | INT |  | Primary Key |
| location | Parking location | VARCHAR | 45 | Not Null |
| lotname | Parking lot name | VARCHAR | 45 | Not Null |
| slotnumber | Parking slot  number | INT |  | Not Null |
| carID | Foreign Key |  | 15 | Not Null |

***User Review***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| reviewID | Unique ID assigned to each  review | INT |  | Primary Key |
| rating\_in\_stars | Passenger’s rating | VARCHAR | 1 | Not Null |
| car\_model\_rating\_in\_stars | Rating in stars for a certain model | VARCHAR | 1 |  |
| passengerID | Foreign key |  |  | Not Null |
| carID | Foreign key |  |  |  |

***Employee***

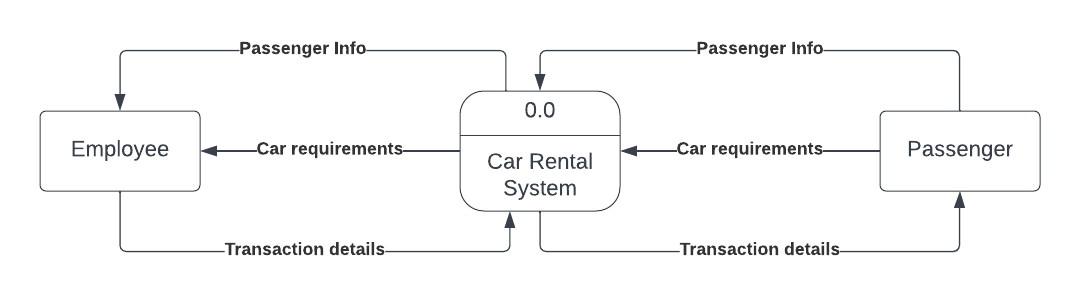
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| empID | The ID of  employee. | INT |  | Primary Key |
| emp\_name | The name of the  employee | VARCHAR | 20 | Not Null |
| emp\_dob | Employee’s date  of birth | DATE |  | Not Null |
| deptID | Foreign key |  |  | Not Null |

***Emp\_department***

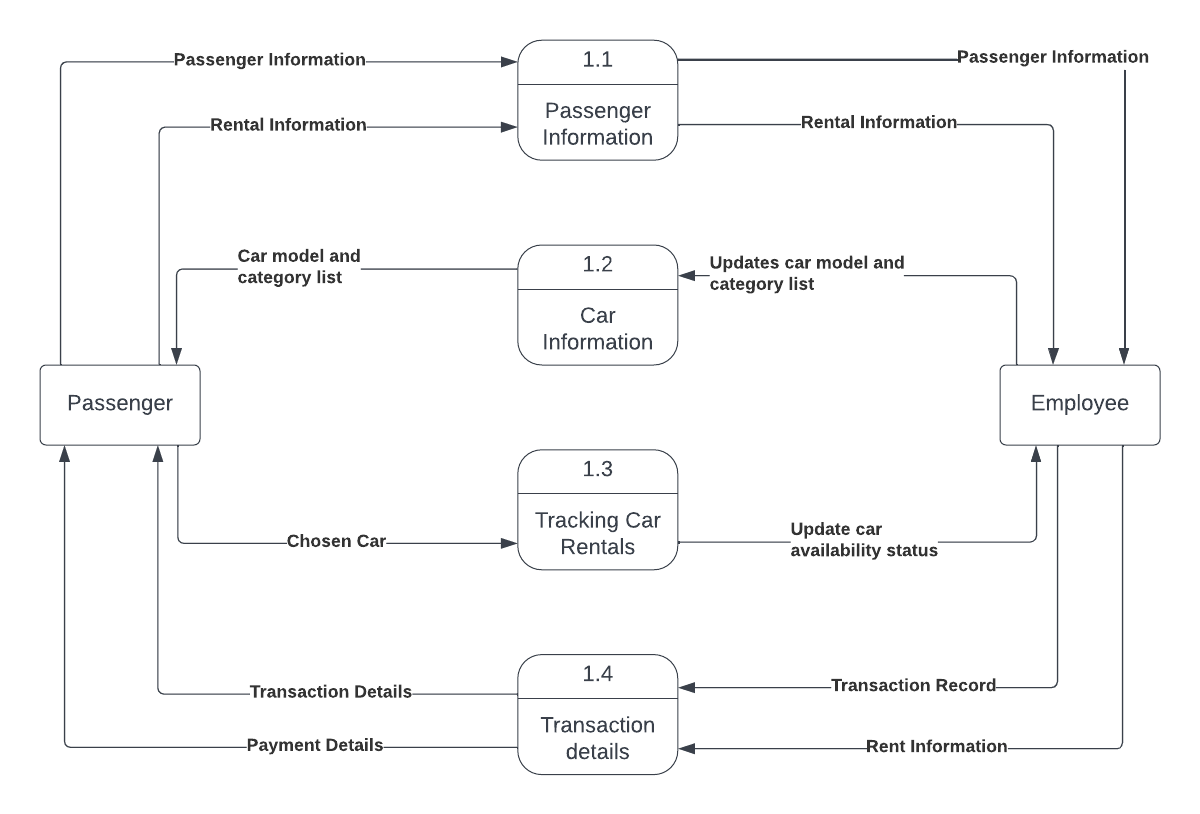
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Description** | **Type** | **Length** | **Comments** |
| deptID | The department ID  of employee | INT |  | Primary Key |
| dept\_name | The department  name of the employee | VARCHAR | 50 | Not Null |

Data Flow Diagram

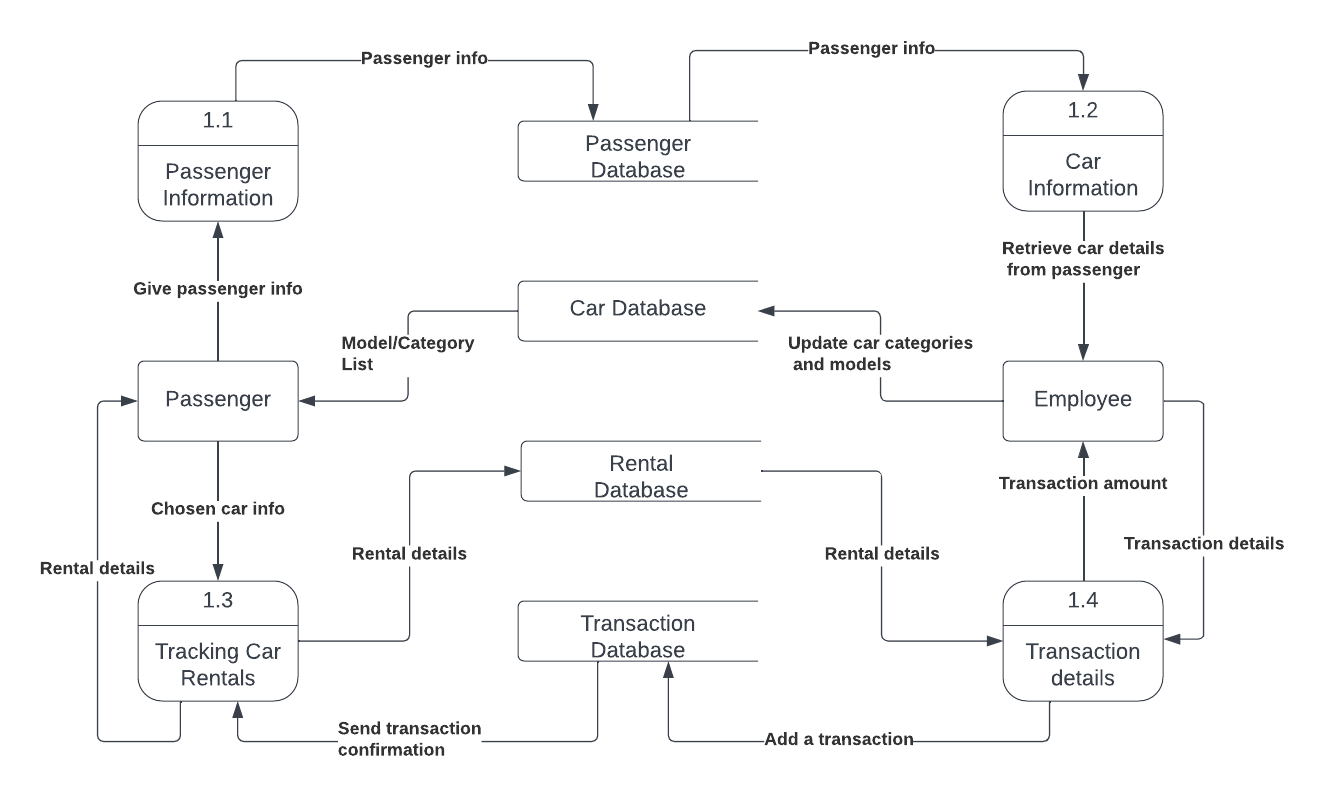
DFD Level 0



DFD Level 1



DFD Level 2:



BUSINESS RULES

The focus of the business is renting cars and the database is to manage the booking system.

1) A **passenger** cannot hire a **car** for longer than a week.

2) If a **car** is available, the passenger's details are stored and a new **reservation** under that car is made.

3) **Passenger**s must pay for the vehicle at the time of hire.

4) An invoice is generated after the passenger books a car.

5) **Car**s are categorized into the following four **categories**: sedan, hatchback, SUV, and minivan.

6) Record stored for each **reservation** includes details of passenger, details of car, date of hire and

the date on which the vehicle is to be returned.

7) Each **passenger** can be associated with only one account.

8) Each **employee** can be associated with only one **department**.

9) Each **car** can be associated with only one **category**.

10) Each **category** can have one or more car **model**s.

11) Each **car** can be associated with only one pickup slot (slot number).

12) Each **parking lot** can have many pickup slots.

13) One or many parking lots can be present in one **location** (zip code).

14) Each **passenger** can give only one **review** after a ride.

15) One **department** can have one or many **employees**.

16) Each **customer support ticket** is assigned to only one **employee**.

17) One **employee** can solve one or many **customers support ticket**s.

**Requirements:**

1) The car rental agency should have a collection of cars.

2) Each **car** belongs to a particular **car category**.

3) Each **car** belongs to a pickup slot in a **location**.

4) Based on the **location** and **car category** preferences selected, the **passenger** would be shown a list of **car**s available to rent, from which he/she must choose from.

5) After the **passenger** decides on a car model, he/she rents the car for a certain period.

6) The **passenger** must pay the trip amount after he/she books the car.

7) A **transaction** log is generated every time a **passenger** books a car showing the status of **transaction** (successful or unsuccessful).

8) The **passenger** must then return the car on the return date specified.

9) Once a **car** is returned, it becomes available for booking.

**Assumptions:**

1) Each **passenger** can only make once **reservation** at a time.

2) Each **car** can only be reserved once at a given time.

3) Each **car** can only be reserved for a max. of 7 days.

3) Every **reservation** is associated with a **transaction** log.

4) Each **car** will be associated to only one pickup slot in a certain **parking lot**.

5) User has to mention only the zip code as address.

SECURITY CONSTRAINTS: USER LEVEL ACCESS AND PERMISSIONS

For a **passenger**:

1. Does not have any access to transaction, employee, employee department tables.
2. Has only READ access on car, parking lot tables.
3. Has READ/WRITE/UPDATE access to passenger, reservation, user review, customer support tables.

For an **employee**:

1. Has only READ access to reservation, transaction, parking lot, user review, employee department tables.
2. Has READ/UPDATE access to customer support tables.
3. Has READ/WRITE/UPDATE access to employee table.

For the **admin**:

The database admin has full access on all the tables.

VIEWS

View 1: cars\_by\_category

- This view will display the **number of cars available** in a certain car category.

Following users will have READ user access and permission over this view:

**passenger, employee, admin**

View 2: highest\_rated\_model\_per\_category

- This view will display the **highest rated model** in every category.

Following users will have READ user access and permission over this view:

**passenger, employee, admin**

View 3: tickets\_per\_issue\_category

- This view will display the **number of tickets** that were raised for a specific issue.

Following users will have READ user access and permission over this view:

**employee, admin**

View 4: top\_3\_employees

- This view will display the **top three employees** who resolved the maximum number of tickets.

Following users will have READ user access and permission over this view:

**employee, admin**

View 5: avg\_rating\_per\_model

- This view will display the **average of star ratings** given by previous passengers to a certain

car model.

Following users will have READ user access and permission over this view:

**passenger, employee, admin**

View 6: rentals\_in\_a\_zipcode

- This view will display the **number of cars that were rented** from a specific location.

Following users will have READ user access and permission over this view:

**employee, admin**

View 7: avg\_rating\_per\_category

- This view will display the **average of star ratings** for every car category.

Following users will have READ user access and permission over this view:

**employee, admin.**