

San Francisco State University
CSC675
Database System
Product Description and Database Requirements
Milestone One

Project Name: Car Rental Database System

Fahim Amiri

921205685

famiri1@mail.sfsu.edu

Github repository link→

<https://github.com/sfsu-joseo/csc675-775-database-systems-sp24-fahimamiri>

Date Started: Feb/07/2023

Due Feb/20/2023

Table of contents

I. Product Summary	3
II. Motivations	3
III. Use Cases of The Car Rental Database	3
1. Use case : Reservation and booking management	3
Functional Requirements	4
Vehicle Management	4
2. Booking management	4
3. Customer management.	5
4. Billing and Invoice	5
5. Vehicles maintenance and repair	5
6. Inspection and reporting	6
7. Insurance	6
Non Functional Requirements	7
IV. Entities And Attributes	7
1. Vehicle Entity:	7
Attributes:	7
2. Customer Entity:	8
Attributes	8
3. Reservations Entity:	8
Attributes	8
4. Booking Entity:	8
Attributes	8
5. Locations Entity:	9
Attributes	9
6. Maintenance Entity:	9
Attributes	9
7. Insurance Entity:	9
Attributes	9
8. Feedback Entity:	10
Attributes	10
Entity Relationship Diagram (ERD)	11

I. Product Summary

In today's business landscape, database systems serve as essential tools for the retention and storage of data across various industries. Similarly, within the realm of car rental services, databases play a pivotal role in enhancing operational efficiency and functionality.

The car rental database system empowers rental companies to effectively manage their vehicle inventory. This includes comprehensive tracking of the fleet, encompassing vehicle details, roadworthiness, and

legal compliance prior to customer rental. Customers value a seamless reservation and booking process when utilizing car rental services. Therefore, the integration of a robust database system proves instrumental in efficiently storing essential customer information, including rental dates, vehicle preferences, pickup locations, and payment details. The utilization of a rental database system offers rental companies numerous advantages, including the ability to maintain comprehensive records of customer information. Moreover, it enables efficient tracking of vehicle maintenance and repair activities, encompassing maintenance schedules, service records, repair histories, and alerts for ensuring vehicle reliability and safety. Additionally, through the analytical and reporting capabilities of the database system, companies can stay abreast of rental trends, customer preferences, and operational performance, facilitating informed decision-making and fostering potential growth opportunities.

II. Motivations

As a software engineer, my passion lies in problem-solving and enhancing the human experience. Engaging with this project presents an opportunity to craft a seamless system that benefits both car rental companies and their valued customers. While existing car rental database systems may fall short of user expectations, I am committed to delivering a solution that prioritizes ease of use, safety, and reliability. Customers value simplicity and accessibility in service delivery, and I am dedicated to fulfilling these expectations through innovative database system design.

In this project, my aim is to introduce a rental car database system that prioritizes efficiency and user-friendliness. It is imperative that car rental companies can navigate the system with ease. Addressing the limitations of existing car rental databases, my objective is to streamline processes and mitigate any concerns that may arise for rental companies. By developing a solution that is intuitive and robust, I aim to enhance operational efficiency and provide a seamless experience for both rental companies and their customers.

III. Use Cases of The Car Rental Database

1. Use case : Reservation and booking management

Actor: Renal company employee (Sara)

Description : A customer, seeking to rent a vehicle from X Rental Company, selects and books a car for a specific date. The customer arrives at the designated location of X Rental Company, situated at 123 H&H Street, to pick up the reserved car. Sara, the rental company employee, assists the customer upon arrival.

Upon verifying the booking confirmation in the system, Sara locates the car reserved under the customer's name. However, upon reaching the parking area, Sara discovers that the selected car is unavailable as it is undergoing servicing. Recognizing the issue, Sara promptly conducts further investigation and identifies the unavailability of the car due to maintenance reasons.

In response, Sara offers alternative options to the customer, ensuring that their rental needs are adequately met despite the unforeseen circumstance. This incident underscores the importance of effective vehicle availability management within the car rental service.

The integration of a robust car rental database system empowers X Rental

Company to promptly update vehicle statuses, such as maintenance schedules, ensuring accurate availability information for customers. By streamlining this process, both customers and X Rental Company benefit from enhanced efficiency, minimized conflicts, and a smoother rental experience.

Functional Requirements

Vehicle Management

- ☒ 1.1 — ~~Users shall be able to log in in the DB system.~~ Ty
- ☒ 1.2 — ~~Users shall be able to see many of the reservations.~~
- ☒ 1.3 — ~~Users shall be able to see all vehicle inventory.~~
- ☒ 1.4 — ~~Authenticate users with unique login credentials,~~
- ☒ 1.5 — ~~Users shall be able to add new cars to the database including make model , year and registration info.~~
- ☒ 1.6 — ~~Users shall be able to Update the status of existing vehicles.~~
- ☒ 1.7 — ~~Users shall be able to remove sold , or damaged vehicles~~
- ☒ 1.8 — ~~Users shall be able to track the vehicle by their VAN numbers~~
- ☒ 1.9 — ~~Users shall be able to categorize vehicles based on their type sedan , SUV, truck.~~
- ☒ 1.10 — ~~The system shall allow Users to check reservations for customers , including selecting the desired vehicle , specifying the rental dates , and record customer details.~~

2. Booking management

- ☒ 2.1 — ~~The booking confirmation shall be stored in the database~~
- ☒ 2.2 — ~~Confirmation should be visible for the user when a customer reserved.~~
- ☒ 2.3 — ~~Users shall be able to record customer information, name ,phone , email.~~
- ☒ 2.4 — ~~Users shall be able to search customer information.~~
- ☒ 2.5 — ~~Users shall be able to send booking confirmation through email.~~
- ☒ 2.6 — ~~Users shall be able to handle customer deposit and payment securely.~~

- ☒ ~~2.7 — User shall be able to rent only one vehicle to a customer~~

3. Customer management.

- ☒ ~~3.1 — users shall be able to create profile for new customer~~
- ☒ ~~3.2 — user shall be able to maintain the rental history of the customer including previous reservation information.~~
- ☒ ~~3.3 — user shall be able to recognize loyal customers, to provide discounts and points.~~
- ☒ ~~3.4 — System shall allow customers to add or update their information on their profile.~~
 - ☒ ~~3.5 — System shall be able to generate customer feedback for analytics and improvement purposes.~~

4. Billing and Invoice

- ☒ ~~4.1 — Users shall be able Calculate rental fees based on factors such as vehicle type, rental duration, and additional services.~~
- ☒ ~~4.2 — Users shall support a real time zone based on customer or employee location.~~
- ☒ ~~4.3 — User shall to support different billing methods~~
- ☒ ~~4.4 — User shall be able to calculate taxes and other misalliance charges on each locations~~
- ☒ ~~4.5 — Users shall be able to download customer invoices form their account.~~
- ☒ ~~4.6 — User shall be able to converts other foreign curreneies~~
- ☒ ~~4.7 — Users shall be able to generate revenue and lost reports on a monthly , yearly basis.~~
- ☒ ~~4.8 — Users shall be able to secure customers payment information such as bank info.~~
- ☒

5. Vehicles maintenance and repair

- ☒ ~~5.1 — User shall not be able to rent vehicle that are listed for maintenance~~
- ☒ ~~5.2 — users shall be able set alerts which vehicle needs to schedule for maintenance.~~
- ☒ ~~5.3 — users shall track service history and maintenance records for each vehicle.~~

- ☒ ~~5.4 — users shall be able to set maintenance for each vehicle.~~
- ☐ 5.5

6. Inspection and reporting

- ☒ ~~6.1 — users shall be able to check the record each vehicle inspections~~
- ☒ ~~6.2 — users shall be able to record and add new damages to each vehicle.~~
- ☒ ~~6.3 — users shall be able to provide access to vehicle inspection reports to customers.~~
- ☒ ~~6.4 — user shall be able to flag down new repair for each vehicle after receiving the car from the customers~~
- ☒ ~~6.5 — User should be able to analyze inspection of each vehicle and schedule repairs~~

7. Insurance

- ☒ ~~7.1 — user shall be able to add at least one insurance for each vehicles~~
- ☒ ~~7.2 — sara should be able to give preference insurance for customer.(optional or spacial insurance)~~
- ☒ ~~7.3 — user should be to verify converges to customers at time of booking~~
- ☒ ~~7.4 — users should be able to handle insurance claims for vehicle accidents or damages during the rentals.~~
- ☒ ~~7.5 — user shall not be able to rent vehicle without insurance~~
- ☒ ~~7.6 — User shall accept insurance from different companies~~
- ☒ ~~7.8 — user shall not accept expired insurance~~
- ☒ ~~7.9 — user shall be able to verify insurance~~

8. Vehicle Inventory

- ☒ ~~8.1 — user should be able to check the see vehicle ready for rent~~
- ☒ ~~8.2 — user should be remove vehicle~~
- ☒ ~~8.3 — user should be able add vehicle~~

- ☒ ~~8.4 — users shall update the vehicle inventory automatically as each car rent out.~~
- ☒ ~~8.5 — user shall be to obtain vehicle information by its VIN~~
- ☒ ~~8.6 — user shall be able to access the value of each var in to days\$ value~~
- ☒ ~~8.7 — user shall check categorization of each type of vehicles.~~

Non Functional Requirements

- 1- The system shall be able to respond to user requests , such obtaining transections.
- 2- system shall be able to increase the number of users , vehicles , and transactions without significantly impacting performance.
- 3- system should have a back up data automatically and avoid data loss .
- 4- system shall be able to protect the internal data of both the customer and the rental company.
- 5- system shall be compatible with a wide range of devices, browsers and operating systems.
- 6- system shall be to maintain and update with well organized code.

IV. Entities And Attributes

1. Vehicle Entity:

Attributes:

- 1. Vehicle ID
- 2. Make
- 3. Model
- 4. Year
- 5. Color
- 6. Mileage
- 7. Registration
- 8. Number VIN (Vehicle Identification Number)

9. Transmission
10. Type
11. Fuel
12. Type
13. Rental Rate

2. Customer Entity:

Attributes

1. Customer ID
2. First Name
3. Last Name
4. Email Address
5. Phone number
6. Address
7. DL number
8. Loyalty points

3. Reservations Entity:

Attributes

1. Reservation ID
2. Customer ID
3. Vehicle ID
4. Rental Start Date
5. Rental End Date
6. Rental Locations
7. Drop off Location
8. Total Rental price
9. Reservation status (confirmed, pending , canceled, hold)

4. Booking Entity:

Attributes

1. Booking ID

2. reservation ID
3. Booking date
4. Payment Type
5. Payment status(confirmed, pending , canceled, hold)

5. Locations Entity:

Attributes

1. Locations ID
2. Location name
3. Address
4. Contact Info
5. POC Name

6. Maintenance Entity:

Attributes

1. Maintenance ID
2. Vehicle ID
3. Maintenance date
4. Maintenance type (regular or repair
5. Descriptions
6. Total cost

7. Insurance Entity:

Attributes

1. Insurance ID
2. Vehicle ID
3. Insurance company name
4. Policy number
5. Policy Start date
6. Policy end date
7. Premium amount

8. Feedback Entity:

Attributes

1. Feedback ID
2. Reservation ID
3. Customer ID
4. Rating
5. Comments
6. Date

9 DB User(employee)

Attributes

1. Login
2. USer ID
3. Password
4. Email
5. Name

10 Inspections

Attributes

- 10.1 Inspection ID
- 10.2 Vehicle ID
- 10.3 Ins Date
- 10.4 User ID

11 Vehicle Inventory

Attributes

- 11.1 Vehicle ID
- 11.2 Make
- 11.3 Model
- 11.4 Year
- 11.5 VIN
- 11.6 Color
- 11.7 Price

Entity Relationship Diagram (ER)