Assignment 1

Topic: Ride & Pickup Database Management System

CPS 510 - Database Systems 1
Start of Logical Database Design/ Phase I

Shayaan Kirubakaran - 501242746 - Section 5

Kirusanth Palakanthan - 501232510- Section 10

Ahmed Hasan - 501238263 - Section 5

Assignment 1: Application Description

Application: Ride and Pickup DBMS - A database system that allows for ordering rides and or picking up items, similar to Uber.

Table of Contents:

Introduction:

The Ride & Pickup Database Management System combines ride-sharing and delivery services into a single cohesive platform. Customers can ask for rides or deliveries, and the system promptly connects them with nearby drivers who are available. All transactions, such as fares, delivery charges, and merchant payments, are securely handled, guaranteeing precision and dependability when compared to manual or disjointed methods.

The system oversees critical elements like customers, drivers, vehicles, merchants, orders, payments, and reviews at the conceptual level. Every entity possesses key attributes (e.g., Customer ID, Driver ID, Order ID) to ensure consistency and data integrity. Connections mirror real-life applications: clients can make several purchases, drivers might perform different jobs with one registered car, and every order creates a payment history.

Ride & Pickup caters to various users through the database's external level. Customers monitor orders, examine previous purchases, and give input. Drivers can view their assigned tasks, check schedules, and track their earnings. Merchants log products, verify delivery orders, and monitor payments. Administrators manage operations, create reports, and guarantee seamless operation of the platform.

The DBMS provides essential data, including ride histories, expense details, delivery logs, and earnings overviews all stored at the internal level. Integrity regulations prohibit conflicting driver assignments, confirm vehicle registrations, and associate reviews with authenticated orders. Ride & Pickup merges transportation and delivery into one system, providing convenience, efficiency, and safety for everyone involved.

Functions:

1. **Customer Management:**
    - Register new customers into the system by storing all of the customers information including login information. (Name, Age, Address, etc)
    - Store a balance associated with the customer in which the customer can add money at any time.
    - Stores a customer rating given by the driver. (1-5 stars)
2. **Driver and Vehicle Management:**
    - Registers new drivers to the system by storing the drivers information and creating a user and password for the driver to sign in through.
    - Store all the information related to the car. (Make, Model, Colour, License, etc.)
    - Stores the drivers balance from trips made. The driver is able to withdraw the amount at any point.
3. **Order Management:**
    - Link a driver to an order upon a user request
    - Ask the customer for the address of the ride/the order at the restaurant

4. **Assignment to Driver:**
    - Assign the driver to a ride order / food order based on location, availability, and number of seats
    - Send the address of the passenger/restaurant and show the route that is the most efficient.
5. **Location Tracker:**
    - Have real time GPS tracking on both the customer and the driver.
    - Stores pick up and drop off points.
6. **Payment Processing:**
    - Processes payments using credit/debit/preloaded balance.
    - Generate receipts for customers and update the balance of both the customer and the driver.

Descriptions:

Entities:

1. **Customer** – Stores customer ID, name, contact details, signup date, and rating; customers can place multiple orders.

2. **Driver** – Contains driver ID, license information, contact details, and rating; drivers fulfill ride and delivery requests.

3. **Vehicle** – Linked to drivers, with attributes such as vehicle ID, license plate, make, model, and insurance expiry.

4. **Service Order** – Represents ride or delivery requests, including order ID, customer, driver, status, timestamps, and fare.

5. **Location** – Defines pickup and drop-off points with address and GPS coordinates.

6. **Payment** – Records transactions linked to orders, including payment ID, method, amount, and status.

7. **Merchant** – Represents restaurants or stores involved in delivery orders, identified by merchant ID and location.

8. **Rating System** – Feedback from customers and drivers, including stars, comments, and order reference.

Relationships:

1. Customer–Service Order: A customer can place many service orders (rides or deliveries), but each service order belongs to one customer (one to many relationship). This ensures that all requests can be traced back to the correct customer.

    - Entities involved: Customer, Service Order

2. Driver–Service Order: A driver may complete many service orders, but each service order is assigned to one driver at a time (one to many relationship). This avoids conflicts and maintains accountability.

    - Entities involved: Driver, Service Order

3. Driver–Vehicle: A driver may register multiple vehicles, but only one vehicle can be used per active trip (one to many relationship). This ensures compliance with insurance and licensing requirements.

    - Entities involved: Driver, Vehicle

4. Service Order–Payment: Each service order must be linked to at least one payment record, which guarantees financial transactions are tracked accurately for billing and auditing.

    - Entities involved: Service Order, Payment

5. Service Order–Location: A service order is connected to one or more locations, typically a pickup and a drop-off. This allows both rides and deliveries to store routing information.

    - Entities involved: Service Order, Location

6. Merchant–Service Order: A merchant can serve many delivery orders, but each delivery order is tied to one merchant (one to many relationship). This keeps the source of goods clear and simplifies reporting on merchant activity.

    - Entities involved: Merchant, Service Order

7. Customer–Rating: A customer may give many ratings, but each rating is linked to a specific service order and its driver (one to many relationship). This creates a history of customer feedback for quality control.

    - Entities involved: Customer, Rating, Service Order, Driver

8. Driver–Rating: Drivers may also rate customers after an order, creating a two-way feedback system. This helps encourage respectful behavior from both sides.

    - Entities involved: Driver, Rating, Service Order, Customer

Conclusion:

As stated before, the intentions of this database is to provide the ability to easily request for food delivery and rideshare services. Various key attributes are going to be present for all entities involved in the form of several IDs and licenses in order to easily identify all entities present for the ER model. Customers will be able to make multiple requests for delivery, with all orders made tied to that one person for tracking all service requests in an organized fashion. Conversely, drivers will be able to register vehicles and choose which vehicle to keep active during their rideshare experience. Both parties will be able to rate one another to ensure a smooth experience on the platform. The payment system is critical for this system to work for actual profit. That being said, payment processing can sometimes have misconceptions or issues in regards to users. The balances pre-loaded finances due to debit and credit accounts users have will allow for an easy to understand way for customers to see if enough currency is available for trips. Tracking deliveries through a single merchant will allow for the same efficiency by being able to track orders through the merchants. With these features, this will be a polished experience for all users interacting with the database.