Database Design, Conceptual ERD – DRIVESHARE

Introduction :

DriveShare Database System aims to expand its services across different customer bases promoting quick and efficient connections between drivers and riders. It provides affordable transportation options and helps cab booking service providers optimize operations through a well-organized database. Additionally, it creates income opportunities for drivers and car owners who are willing to rent out their vehicles.

Ultimately, DriveShare seeks to enhance accessibility and affordability in transportation while offering a smoother experience for all users.

BUSINESS RULES:

Administrative Rules:

- **Admin Approvals:** Administrative users can approve Users for creating profiles such as Rider, Car Owner, or Driver. The approval process may involve multiple steps, including a confirmation email to the User before granting access.
- **User Permissions:** Admins can assign specific roles (Rider, Car Owner, Driver) and manage their account details securely.
- **Generation of Reports and Resolving disputes :** Admin users can access invoices for trips and audits to generate reports to observe the overall working of in the system.

User Profiles:

- **Profile Assignment:** Each User selects a profile (Rider, Car Owner, Driver) based on their registered account type. Car Owners gain access to vehicle information, while Drivers are responsible for managing assigned vehicles.

Car Management:

- **Car Ownership:** A Car Owner can manage multiple cars, each with unique registration details. Cars are displayed separately under the User's account for easy access and management.
- **Car Assignment:** Drivers can have his own car or be assigned a car allowing them to provide service for the generated trip request.

.

Request Management:

- **Driver Requests:** Drivers can select ride requests based on availability, pickup/drop-off locations, and drivers preferences.

Payment Gateways:

- **Payment Methods:** Riders select their preferred payment gateway from options like credit/debit cards, Online wallet and cash.

Request Details:

- **Request Information:** Each request clearly specifies source (pickup location) and destination (drop-off location), aiding in trip planning and tracking.

Invoice Generation:

- **Trip to Invoice:** Invoices are generated for each trip request from, logging all the detailing for the request timestamps, payment methods,trip details and rider can access his trip information for his invoices and admin can audit invoices to observe the working for the system.

ENTITIES:

Admin :

- **Description:** The administrator's main responsibility is to ensure the database is updated promptly. Admin approval is required for tasks such as adding new Car Owner or Driver profiles.
- **Attributes:**AdminID,UniqueID,ApprovalStatus,DisputeManagement,ReportsGenerated, InvoiceID

RegisteredUser :

- **Description:** The system monitors all users of DriveShare services. Each user must register and select one of the following profiles: Rider, Driver, or Renter.
- **Attributes:** UniqueID,ContactNumber,UserName,Password, FirstName, LastName, Email, Type, UserName

Renter:

- **Description:** Renter entity is designed to monitor individuals offering their vehicles for taxi services, while also allowing them to view their total earnings
- **Attributes:** RenterID, UniqueID, VehicleID(FK),Vehicle Type, RentTime, TotalEarnings

Driver :

- **Description:** Drivers subusers on the service app provide trips for the riders. Driver can select the trip request that he wants to serve from the pool.
- **Attributes:** DriverID, UniqueID, LicenseNo , AvailabilityStatus, CompletedRides, Rating , TotalEarnings

Rider:

- **Description:** Users who register as a rider are categorized under the Rider entity. When a customer signs up, they are assigned a unique RiderID using the same they can request for trip rides
- **Attributes:** RiderID, UniqueID, RequestedRides , CompletedRides, AccountCredits.

Car:

- **Description:** The system maintains records of all car details, including model, capacity, and distance traveled, within the Cars entity. This entity also tracks the Owner and Driver assigned to each vehicle.
- **Attributes:**CarID,RenterID,DriverID,VehicleID,Model,Capacity,TraveledDistance.

TripRequest:

- **Description:** This entity tracks ride requests placed by riders, who may submit multiple requests. It also monitors the status of each booking to determine if a driver has accepted the request, and records details such as trip estimated distance, pickup location, cost, and trip duration.
- **Attributes:** RequestID, RiderID, StartingLocation, Destination, TripDistance, Status, EstimatedCost, EstimatedEndTime.

DisputedTrip:

- **Description:** A dedicated sub-entity under TripRequest to record details about an uncompleted trip and its outcome, indicating reasons for issues or disputes that arose. It includes information such as resolution status, resolution details, and resolution date.
- **Attributes:** DisputeID, AdminID,ResolutionStatus, ResolutionDetails, ResolutionDate

Payment:

- **Description:** The Payment entity is essential for offering riders multiple options to settle their fare. It tracks whether the customer has completed the payment and records the payment method chosen by the rider, such as Cash, OnlineWallet, or Card.

  Attributes: PaymentID, RequestID, Amount , PaymentMethod, Status , Timestamp

Invoice:

- **Description:** An invoice is generated for each trip to collect information about the trip details to inform the rider and admin of the trip details.Which can be further used in report analytics and auditing.
  Attributes: InvoiceID, PaymentID, Status , RiderID, Timestamp.

Card:

- **Description:** A sub entity under payment entity , if a rider chooses card as the payment method it will store the details such as Card number , card holder name , CVV and other card details

  Attributes: CardNumber(PK) , Cardholder , ExpiryDate , CVV , Type.

Online Wallet:

- **Description:** A sub entity under payment entity, if a rider chooses Wallet as the payment method it will store the details such as WalletProvider , ID and the email which wallet is registered to.
- **Attributes:** WalletProvider , WalletID (PK), AccountEmail

Cash:

- **Description:** A sub entity under payment entity, if a rider chooses Cash as the payment method there are not much details stored in the database apart from the confirmation from the receiver.
- **Attributes:** ReceiverConfirmation(PK)

5. Entity Relationships

5.1 RegisteredUser

- **Admin:** One-to-Many (An admin approves multiple registered users, but each user is approved by one admin).
- **Renter:** One-to-One (A registered user can be a renter, but a renter must be a registered user).
- **Driver:** One-to-One (A registered user can be a driver, but a driver must be a registered user).
- **Rider:** One-to-One (A registered user can be a rider, but a rider must be a registered user).

5.2 Admin

- **RegisteredUser:** One-to-Many (An admin approves multiple registered users).
- **Audit:** One-to-Many (An admin can perform multiple audits).

5.3 Renter

- **Car:** One-to-Many (A renter can rent multiple cars, but each car is rented by one renter).

5.4 Driver

- **TripRequest:** One-to-Many (A driver can fulfill multiple trip requests, but each trip request is handled by one driver).
- **Car:** One-to-One (each car belongs to one driver).

5.5 Rider

- **TripRequest:** One-to-Many (A rider can make multiple trip requests, but each trip request belongs to one rider).

- **Invoice:** One-to-Many (A rider can have multiple invoices, but each invoice is linked to one rider).

5.6 Car

- **Renter:** Many-to-One (A car is rented by one renter, but a renter can rent multiple cars).
- **Driver:** One-to-One (A car is assigned to one driver).

5.7 TripRequest

- **Rider:** Many-to-One (A trip request is made by one rider, but a rider can make multiple requests).
- **Driver:** Many-to-One (A trip request is assigned to one driver, but a driver can handle multiple requests).
- **Payment:** One-to-One (Each trip request has one corresponding payment).

5.8 Payment

- **TripRequest:** One-to-One (Each trip request has one corresponding payment).
- **Invoice:** One-to-One (A payment is linked to one invoice).

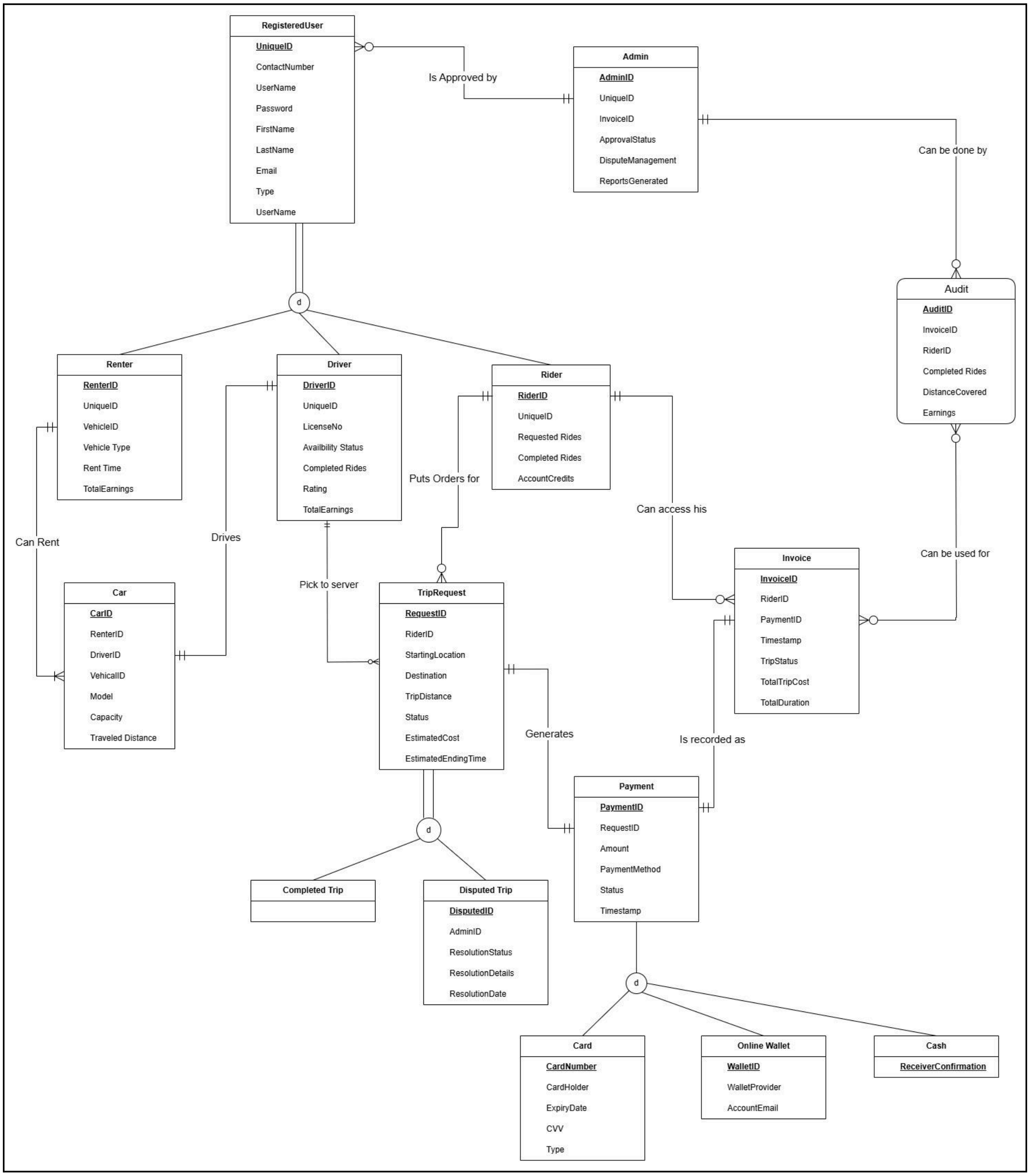
5.9 Invoice

- **Rider:** Many-to-One (A rider can have multiple invoices, but each invoice is linked to one rider).
- **Payment:** One-to-One (Each invoice is linked to one payment).

5.10 Audit

- **Admin:** Many-to-One (An admin can perform multiple audits, but each audit is performed by one admin).
- **Rider:** Many-to-One (A rider can have multiple audits, but each audit is linked to one rider).

6. ER Diagram for DriveShare


RegisteredUser
UniqueID
ContactNumber
UserName
Password
FirstName
LastName
Email
Type
UserName
Is Approved by
Admin
AdminID
UniqueID
InvoiceID
ApprovalStatus
DisputeManagement
ReportsGenerated
Can be done by
Audit
AuditID
InvoiceID
RiderID
Completed Rides
DistanceCovered
Earnings
d
Renter
RenterID
UniqueID
VehicleID
Vehicle Type
Rent Time
TotalEarnings
Driver
DriverID
UniqueID
LicenseNo
Availbility Status
Completed Rides
Rating
TotalEarnings
Rider
RiderID
UniqueID
Requested Rides
Completed Rides
AccountCredits
Puts Orders for
Can access his
Can be used for
Can Rent
Drives
Pick to server
Car
CarID
RenterID
DriverID
VehicalID
Model
Capacity
Traveled Distance
TripRequest
RequestID
RiderID
StartingLocation
Destination
TripDistance
Status
EstimatedCost
EstimatedEndingTime
Invoice
InvoiceID
RiderID
PaymentID
Timestamp
TripStatus
TotalTripCost
TotalDuration
Generates
Is recorded as
Payment
PaymentID
RequestID
Amount
PaymentMethod
Status
Timestamp
d
Completed Trip
Disputed Trip
DisputedID
AdminID
ResolutionStatus
ResolutionDetails
ResolutionDate
d
Card
CardNumber
CardHolder
ExpiryDate
CVV
Type
Online Wallet
WalletID
WalletProvider
AccountEmail
Cash
ReceiverConfirmation