Submitted By: Sabayna Ali (213062) Immad Amir (210841)

|

Ride Hailing Service

PROJECT REPORT

# **PROJECT IDEA:**

The aim of our project is to design and implement a database system for an internet-based easy to use ride-hailing service application.

Ride-hailing is a term to describe booking rides and paying for car service through a smartphone app with a transportation network company (TNC) such as Uber or Careem. A ride hailing service is, in most cases, a comfortable method for door-to-door transport. Ride hailing services use online-enabled platforms to connect between passengers and local drivers using their personal non-commercial vehicles. This service turns out to be cheaper and safer and comfortable than the usual means of transport services such as taxis and buses. It is worth considering that at after the ride is done the passengers can use the app to rate the driver. Unlike the traditional Taxicab services where the drivers have no interest in being nice to their passengers, most of the ride hailing service drivers tend to be as nice as they can to their customers so that at the end of the rides the customers will give them a good rating.

The aim of our project is to develop a database system that is meant to completely computerize the work performed in the ride hailing service like booking a ride, record routes taken for each ride, fare charges of each ride, method of payment, store record of the customer and driver alike and the rating given to each driver by the customers that will further help customers decide whether to ride with a driver or not. The customer can also cancel the ride at any moment with a small amount deducted from their virtual wallet which is stored in the account of each customer.

To implement this idea we need to cater for the three basic entities that are involved in this system which are:

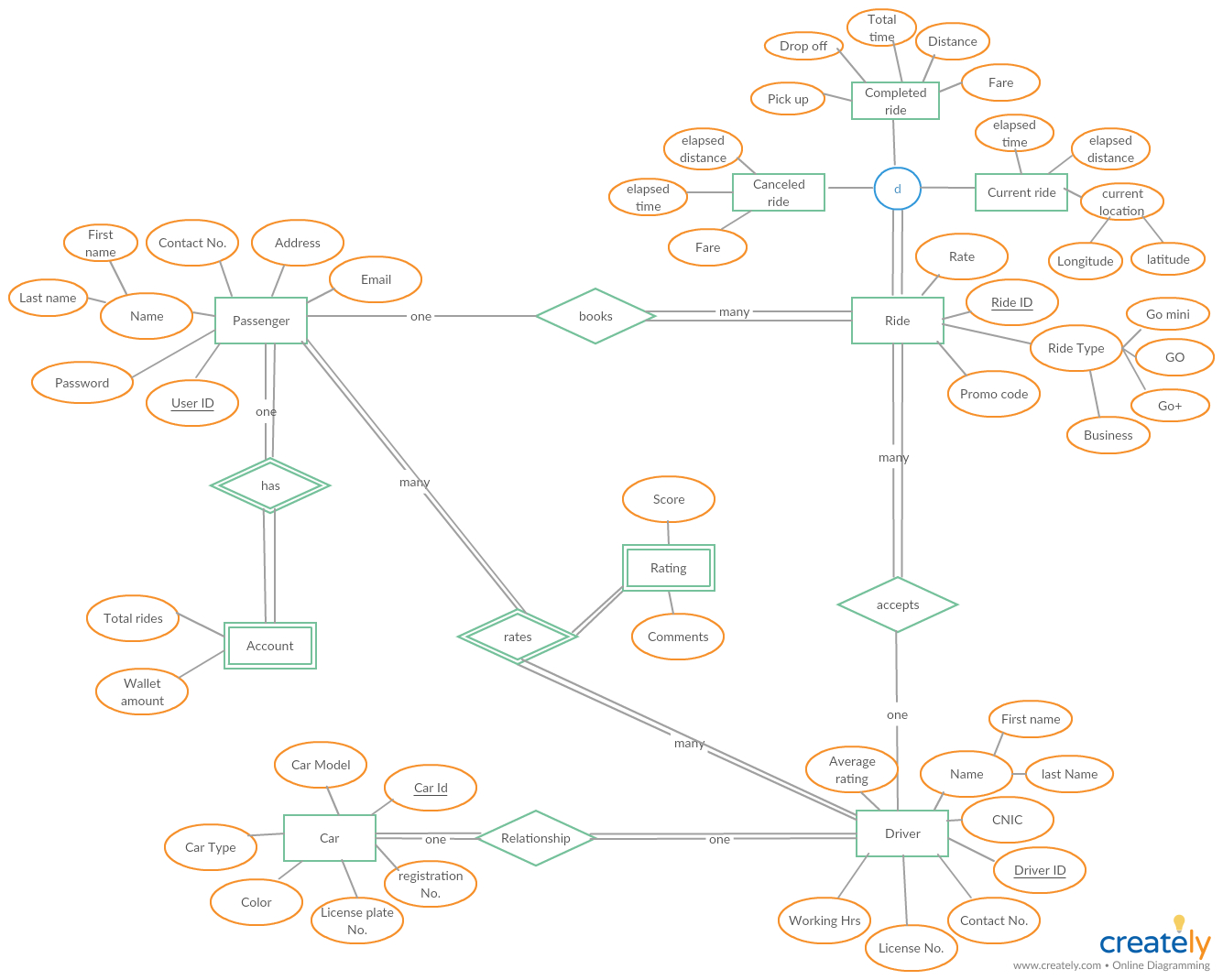
* Passengers
* Drivers
* Rides

To show all the tables required to store all the relevant information and to understand the relationship between them we have created an Entity Relationship Diagram and also a logical schema that are as follows:

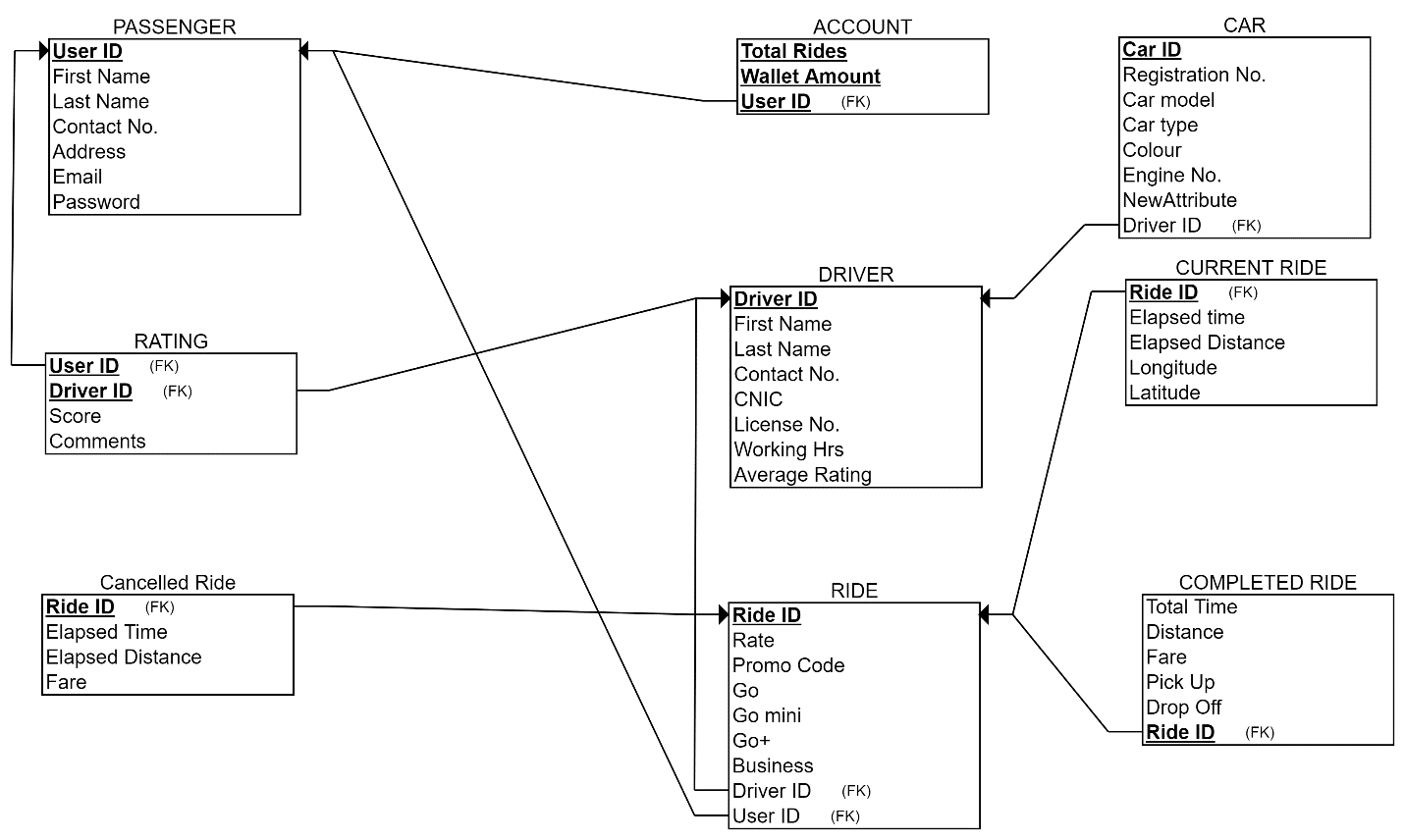
# Mini World:

A ride hailing service similar to Careem is being designed that keeps track of passengers and their accounts, drivers and their rating, cars and rides. The passenger’s details include a unique user ID, first and last name, contact information, email address and home address and password. Each user has an account that contains information about the total rides of a user and current balance in their account. A user books a ride which has different types such as Go mini, Go, Business etc. Each booked ride has a unique ride ID. It has a rate depending on the time of day and promo discount if promo code is applied for the ride. Each ride is also divided into three sub categories: Current ride stores time, distance and current location (longitude, latitude). Completed ride has total time and distance, pickup and drop off location and fare for the ride which is derived from distance multiplied by rate and divided by promo discount. A driver can accept many rides. Each driver has a unique driver ID, name, CNIC, contact number, license number, hours worked and their average rating. Each driver is associated with one car which has a unique car ID, its registration number, model, type and color.

# **ER DIAGRAM:**



# **SCHEMA:**



# **Functional Dependencies:**

**PASSENGER:**

User ID 🡪 First Name, Last Name, Contact No., Address, Email, Password

Email 🡪 User ID, First Name, Last Name, Contact No., Address, Password

Contact No. 🡪 User Id, First Name, Last Name, Address, Email, Password

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.
* All the attributes on the left hand side of the functional dependencies are super keys, the relation is in BCNF.

**ACCOUNT:**

User ID 🡪 Total Rides, Wallet Amount

Total Rides 🡪 User ID, Wallet Amount

Wallet Amount 🡪 User ID, Total Rides

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.
* All the attributes on the left hand side of the functional dependencies are super keys, the relation is in BCNF.

**RATING:**

User ID 🡪 Driver ID, Score, Comments

Driver ID 🡪 User ID, Score, Comments

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.
* All the attributes on the left hand side of the functional dependencies are super keys, the relation is in BCNF.

**DRIVER:**

Driver ID 🡪 First Name, Last Name, Contact No., License No., CNIC, Working Hrs, Average rating,

CNIC 🡪 First name, Last Name

License No. 🡪 CNIC, First Name, Last name

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.
* All the attributes on the left hand side of the functional dependencies are super keys, the relation is in BCNF.

**CAR:**

Car ID 🡪 Registration No., Engine No., Car model, Car Type, Color, Driver ID

Registration No. 🡪 Engine No., Car Model, Car type, Color

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.
* All the attributes on the left hand side of the functional dependencies are super keys, the relation is in BCNF.

**RIDE:**

Ride ID 🡪 Rate, Promo Code, Go, Go mini, Go+, Business, User ID, Driver ID

Go 🡪 Rate

Go+ 🡪 Rate

Go mini 🡪 Rate

Business 🡪 Rate

* The relation is in 1NF since all the values are atomic values. There is no multivalued attributes in the relation.
* There are no partial functional dependencies so the relation is in 2NF.
* It is also 3NF because the transitivity property does not hold.