Aryan Kumawat
Sohum Dharamsi

RamShare

Real Word Problem

As college students, we have a multitude of responsibilities to juggle. From managing schoolwork and attending classes, to preparing meals, making new friends, and working jobs, our plates are undoubtedly full. Amidst all these demands, heading home for breaks becomes an added responsibility and an additional source of stress. Coordinating transportation, finding affordable travel options, and aligning schedules with others is extra busy work piled onto an already demanding routine. Moreover, 86% of students who attend VCU are in-state. Meaning that oftentimes, people are going to the same places, whether it be Northern Virginia, the Hampton Roads metropolitan area, or even RVA itself. Despite the common destinations, many students face similar challenges when planning their journeys home. It's a scenario where the need for efficient, cost-effective travel solutions becomes abundantly clear. Carpooling presents an efficient and eco-friendly solution for a huge percentage of VCU students, fostering a sense of community while reducing individual travel costs (gas money) and time. On the other hand, carpooling does not need to be strictly limited for rides to and from home. Students might want to travel to other colleges for a weekend to attend concerts, watch sporting events, or even to simply visit a friend. Taking advantage of the cost effective travel solutions near them will not only make their trip worth it but also stress-free. Therefore we need an application that is able to assist all students in whatever ways they need, and by offering a seamless carpool system, we aim to simplify the process of planning trips for VCU students.

Aryan Kumawat
Sohum Dharamsi

Expected information and functionality

Users

Users will need to create an account using their VCU email to utilize the application. No guest profiles should be allowed as potential security concerns arise when carpooling. There should be some sort of user verification (involving the VCU email). Each user should be able to select whether they wish to be a driver or a rider, and based upon their selection, they will be further prompted to make choices.

RamShare Post/Request

A driver may create ride share posts whenever they are traveling home or to another college for the weekend. Each post will have a unique ride id and the caption section will include the driver name, trip status, distance, date and time posted, date and time of ride, cost (how much the driver was charging for the ride), payment method (the driver's preferred method of receiving payment), departure address, destination address, luggage information, as well as a post content/additional information section which could include things like trunk space, music preferences, and other miscellaneous things. Riders will have a feed which they can scroll through to see if a post made by a driver matches their travel plans. If travel plans match, a rider can respond to the driver by requesting a ride from them. Each request will have the rider's name, status of the request, as well as the time the request was made.

Ride

Once a driver creates a rideshare post and accepts a riders request, a ride id is created. This will show up in the user's “Trip History” and will add to the number of trips provided if they are a driver or add to the number of trips taken if they are a rider. Every ride must be associated with one vehicle while a vehicle may do many rides.

Vehicle Registration

Users should have the capability to include vehicles in their account, which is a prerequisite for providing rides. Without vehicle registration, a driver will not be able to host rides. They need to provide details such as the vehicle's model, year, color, state, and license plate number. Since a user might possess multiple cars, they should be able to add numerous vehicles to their account and remove them as needed.

Rating & Review

Allow both riders and drivers to rate each other after a ride has been completed. This way, there is increased transparency and accountability for all parties and it would allow for increased trust. Allow riders to leave any reviews. Such comments would be publicly visible, which would be helpful for other riders to make choices in regards to picking a driver.

Trip History

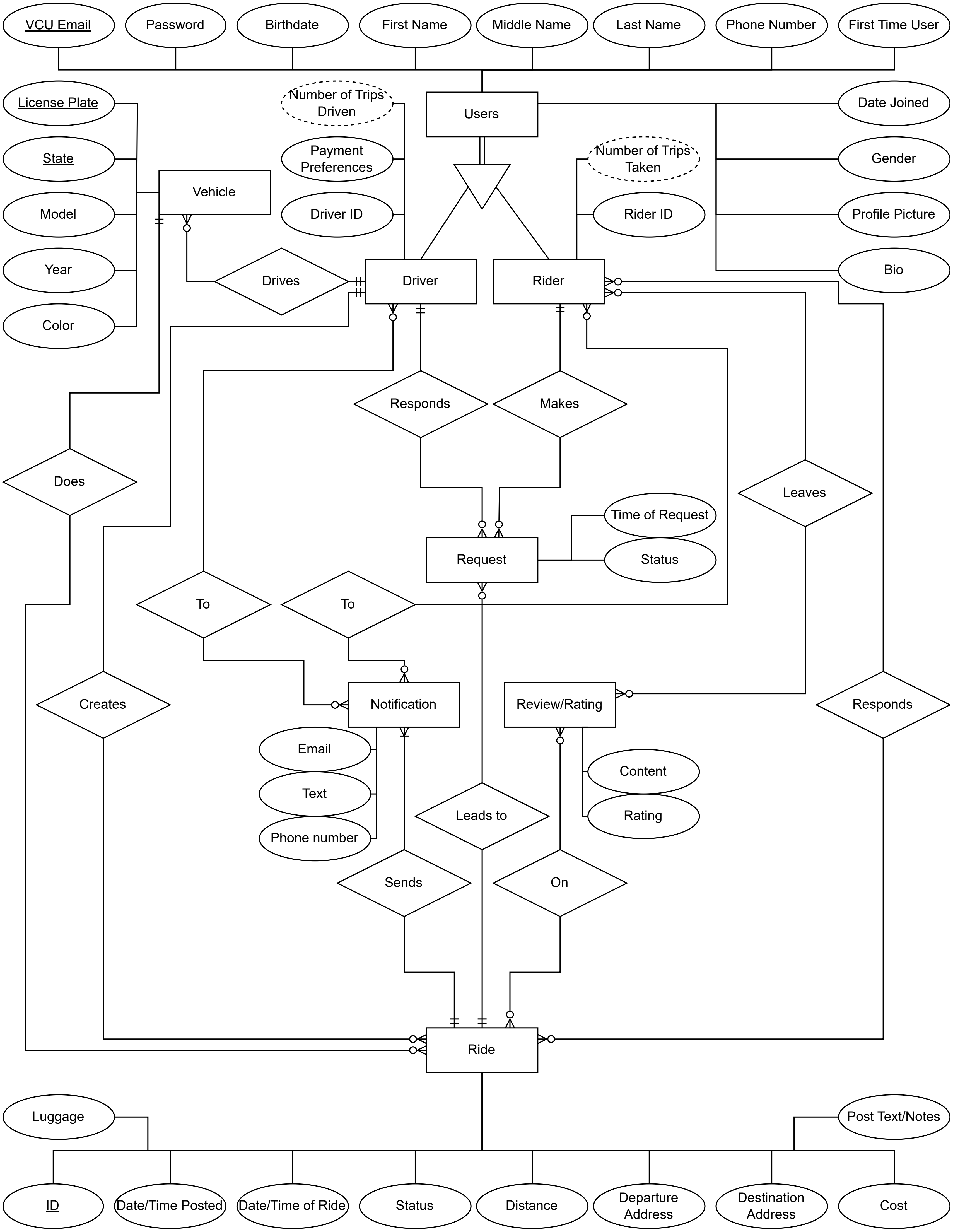
There should be a page where the user should be able to see all the trips they have provided/taken utilizing the app, indicating who the driver was if they were a rider, how long the trip took, how much it cost them, and distance. If they were a driver for the given trip, who their passengers were, how long the trip took them, and the distance of the trip.

User Profile

There should be additional rider and driver information in order to assist in matchmaking. Each person should have a user profile available for each user to see. User Profiles will consist of VCU email & password, date of birth, full name (first, middle, last), phone number, age, start date, gender, profile pic, biography, and what type of user they are. If they are a Driver then they will have information regarding the number of trips they’ve provided as well as their payment preferences. If they are a Rider then they will have information regarding the number of trips they have been part of.

Notifications

Set up a notification system that alerts riders/drivers when matched. For example, if a rider requests a ride, send a text/email notification notifying the driver to review the request. If a driver accepts a ride, send a text/email notification notifying the rider. If any cancellations occur, a cancellation notification will be sent as an alert 30 minutes/1 hour before the anticipated ride. A notification prompting users to rate the trip. Would also need a notifications setting page to assist with the users preferred way of notifications.


VCU Email
Password
Birthdate
First Name
Middle Name
Last Name
Phone Number
First Time User
License Plate
Number of Trips Driven
Users
Date Joined
State
Payment Preferences
Number of Trips Taken
Gender
Vehicle
Model
Driver ID
Rider ID
Profile Picture
Year
Drives
Driver
Rider
Bio
Color
Responds
Makes
Does
Leaves
Time of Request
Request
Status
To
To
Creates
Notification
Review/Rating
Responds
Email
Content
Text
Leads to
Rating
Phone number
Sends
On
Ride
Luggage
Post Text/Notes
ID
Date/Time Posted
Date/Time of Ride
Status
Distance
Departure Address
Destination Address
Cost

Tables

UserProfile (*VCU_Email*: [PK], *Birthdate*: [not null], *UserPassword*: [not null], *First_Name*: [not null], *Middle_Name*, *Last_Name*: [not null], *Phone_Number*: [not null], *Date_Joined*: [not null], *Gender*: [not null], *Profile_Picture*, *Bio*, *Type_Of_User*: [not null], *First_Time_User*)

Driver (*VCU_Email*: [PK], *Driver_ID*: [not null, unique], *Payment_Preferences, Driver_Rating, Number_Of_Rides)*
FK1 {*VCU_Email*} references {UserProfile.*VCU_Email*}

Rider (*VCU_Email*: [PK], *Rider_ID*: [not null, unique], *Rider_Rating, Number_Of_Rides*)
FK1 {*VCU_Email*} references {UserProfile.*VCU_Email*}

Vehicle (*License_Plate*: [PK], *Vehicle_State*: [PK], *Model*: [not null], *Vehicle_Year*: [not null], *Color*: [not null], *Driver_ID*: [not null, unique])
FK1 {*Driver_ID*} references {Driver.*Driver_ID*}

Ride (*Ride_ID*: [PK], *Driver_ID*: [not null, unique], *Trip_Status*: [not null], *Distance*: [not null], *Date_Time_Ride_Posted*: [not null], *Date_Time_Of_Ride*: [not null], *Cost*: [not null], *Post_Content, Departure_Address*: [not null], *Destination_Address*: [not null], *Luggage*)
FK1 {*Driver_ID*} references {Driver.*Driver_ID*}

Reviews_Ratings (*Rider*: [PK], *Driver*: [PK], *Ride_ID*: [PK], *Content, Rating)*
FK1 {*Rider*} references {Rider.*Rider_ID*}
FK2 {*Driver*} references {Driver.*Driver_ID*}
FK3 {*Ride_ID*} references {Ride.*Ride_ID*}

RamShare_Request (*Ride_ID*: [PK], *Rider*: [PK], *Request_Status*: [not null], *Time_Request_Was_Made*: [not null]*)*
FK1 {*Rider*} references {Rider.*Rider_ID*}
FK2 {*Ride_ID*} references {Ride.*Ride_ID*}

Notification (*ID*: [PK], *Ride_ID*: [PK], *Type_Of_Notification*: [PK], *Phone Number*: [not null], *VCU_Email*: [not null, unique], *Notification_Text*: [not null]*)*
FK1 {*Ride_ID*} references {Ride.*Ride_ID*}
FK2 {*VCU_Email*} references {UserProfile.*VCU_Email*}
FK3 {*Phone_Number*} references {UserProfile.*Phone_Number*}

Functional Dependencies

UserProfile (*VCU_Email*, *Birthdate*, *UserPassword*, *First_Name*, *Middle_Name*, *Last_Name*, *Phone_Number*, *Date_Joined*, *Gender*, *Profile_Picture*, *Bio*, *Type_Of_User*, *First_Time_User*)

- F = {*VCU_Email -> Birthdate*, *UserPassword*, *First_Name*, *Middle_Name*, *Last_Name*, *Phone_Number*, *Date_Joined*, *Gender*, *Profile_Picture*, *Bio*, *Type_Of_User*, *First_Time_User*}
- CK = {{*VCU_Email*}}
- In BCNF/4NF

Driver (*VCU_Email, Driver_ID, Payment_Preferences, Driver_Rating, Number_Of_Rides)*

- F = {*VCU_Email -> Driver_ID, Payment_Preferences, Driver_Rating, Number_Of_Rides,*
  Driver_ID -> VCU_Email, Payment_Preferences, Driver_Rating, Number_Of_Rides}
- CK = {{*VCU_Email*}, {*Driver_ID*}}
- In BCNF/4NF

Rider (*VCU_Email, Rider_ID, Rider_Rating, Number_Of_Rides*)

- F = {*VCU_Email -> Rider_ID, Rider_Rating, Number_Of_Rides,*
  Rider_ID -> VCU_Email, Rider_Rating, Number_Of_Rides}
- CK = {{*VCU_Email*}, {*Rider_ID*}}
- In BCNF/4NF

Vehicle (*License_Plate, Vehicle_State, Model, Vehicle_Year, Color, Driver_ID*)

- F = {*License_Plate -> Vehicle_State, Model, Vehicle_Year, Color, Driver_ID,*
  Driver_ID -> License_Plate, Vehicle_State, Model, Vehicle_Year, Color}
- CK = {{*License_Plate*}, {*Driver_ID*}}
- In BCNF/4NF

Ride (*Ride_ID, Driver_ID, Trip_Status, Distance, Date_Time_Ride_Posted, Date_Time_Of_Ride, Cost, Post_Content, Departure_Address, Destination_Address, Luggage*)

- F = {*Ride_ID -> Driver_ID, Trip_Status, Distance, Date_Time_Ride_Posted, Date_Time_Of_Ride, Cost, Post_Content, Departure_Address, Destination_Address, Luggage*
  Driver_ID -> *Ride_ID, Trip_Status, Distance, Date_Time_Ride_Posted, Date_Time_Of_Ride, Cost, Post_Content, Departure_Address, Destination_Address, Luggage*}
- CK = {{*Ride_ID*}, {*Driver_ID*}}
- In BCNF/4NF

Reviews_Ratings (*Rider, Driver, Ride_ID, Content, Rating)*

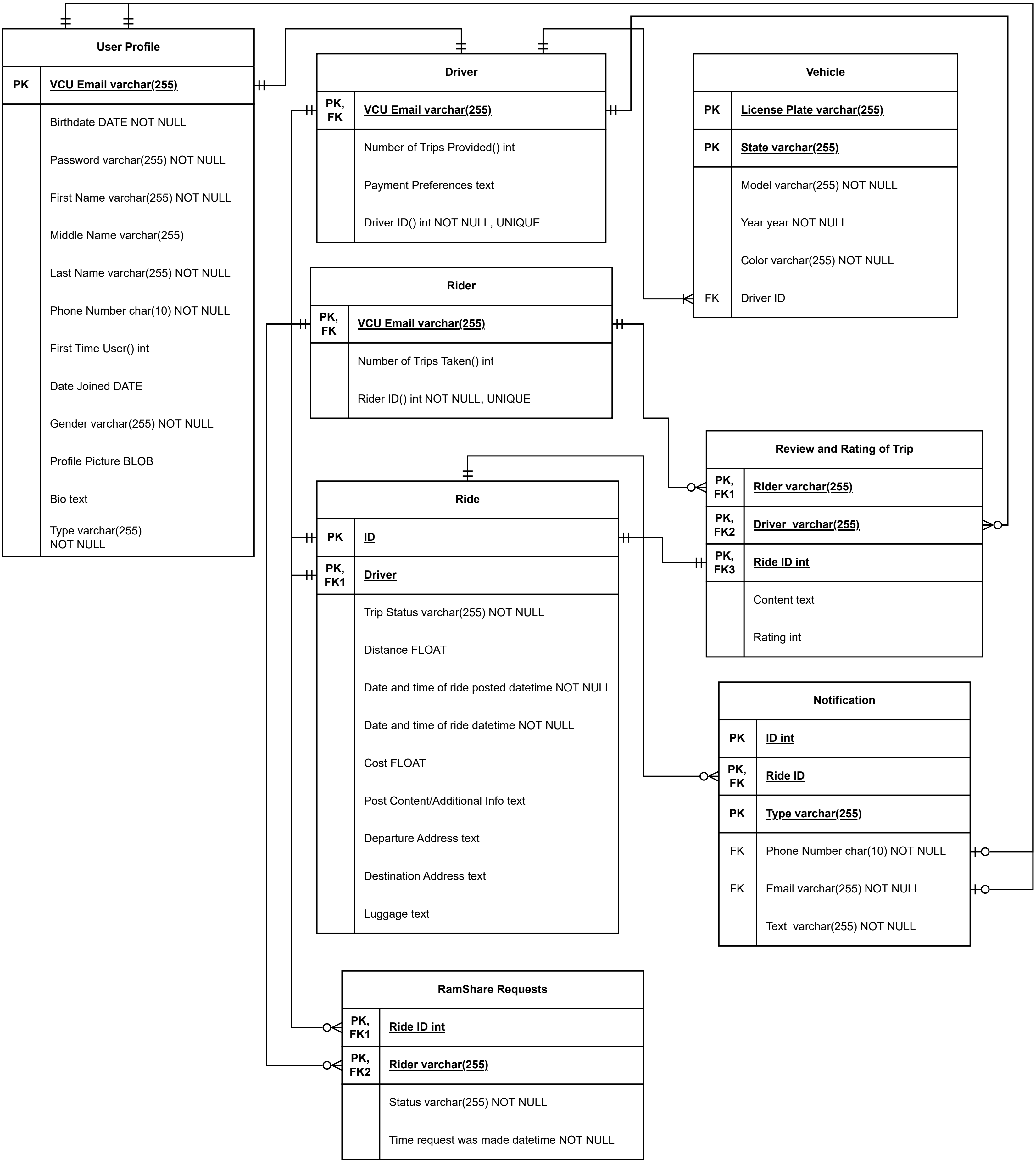
- F = {*Rider* -> *Driver, Ride_ID, Content, Rating,*
  Driver -> *Rider, Ride_ID, Content, Rating,*
  Ride_ID -> *Rider, Driver, Content, Rating*}
- CK = {{*Rider*}, {*Driver*}, {*Ride_ID* }}
- In BCNF/4NF

RamShare_Request (*Ride_ID, Rider, Request_Status, Time_Request_Was_Made*)

- F = {*Ride_ID* -> *Rider, Request_Status, Time_Request_Was_Made,*
  Rider -> *Ride_ID, Request_Status, Time_Request_Was_Made*}
- CK = {{*Ride_ID*}, {*Rider* }}
- In BCNF/4NF

Notification (*ID, Ride_ID, Type_Of_Notification, Phone Number, VCU_Email, Notification_Text)*

- F = {*ID* -> *Ride_ID, Type_Of_Notification, Phone Number, VCU_Email, Notification_Text,*
  Ride_ID -> *ID, Type_Of_Notification, Phone Number, VCU_Email, Notification_Text,*
  VCU_Email -> *ID, Ride_ID, Type_Of_Notification, Phone Number, Notification_Text*}
- CK = {{*ID*}, {*Ride_ID* }, {*VCU_Email* }}
- In BCNF/4NF



User Profile

PK	VCU Email varchar(255)
	Birthdate DATE NOT NULL
	Password varchar(255) NOT NULL
	First Name varchar(255) NOT NULL
	Middle Name varchar(255)
	Last Name varchar(255) NOT NULL
	Phone Number char(10) NOT NULL
	First Time User() int
	Date Joined DATE
	Gender varchar(255) NOT NULL
	Profile Picture BLOB
	Bio text
	Type varchar(255) NOT NULL

Driver

PK, FK	VCU Email varchar(255)
	Number of Trips Provided() int
	Payment Preferences text
	Driver ID() int NOT NULL, UNIQUE

Vehicle

PK	License Plate varchar(255)
PK	State varchar(255)
	Model varchar(255) NOT NULL
	Year year NOT NULL
	Color varchar(255) NOT NULL
FK	Driver ID

Rider

PK, FK	VCU Email varchar(255)
	Number of Trips Taken() int
	Rider ID() int NOT NULL, UNIQUE

Review and Rating of Trip

PK, FK1	Rider varchar(255)
PK, FK2	Driver varchar(255)
PK, FK3	Ride ID int
	Content text
	Rating int

Ride

PK	ID
PK, FK1	Driver
	Trip Status varchar(255) NOT NULL
	Distance FLOAT
	Date and time of ride posted datetime NOT NULL
	Date and time of ride datetime NOT NULL
	Cost FLOAT
	Post Content/Additional Info text
	Departure Address text
	Destination Address text
	Luggage text

Notification

PK	ID int
PK, FK	Ride ID
PK	Type varchar(255)
FK	Phone Number char(10) NOT NULL
FK	Email varchar(255) NOT NULL
	Text varchar(255) NOT NULL

RamShare Requests

PK, FK1	Ride ID int
PK, FK2	Rider varchar(255)
	Status varchar(255) NOT NULL
	Time request was made datetime NOT NULL