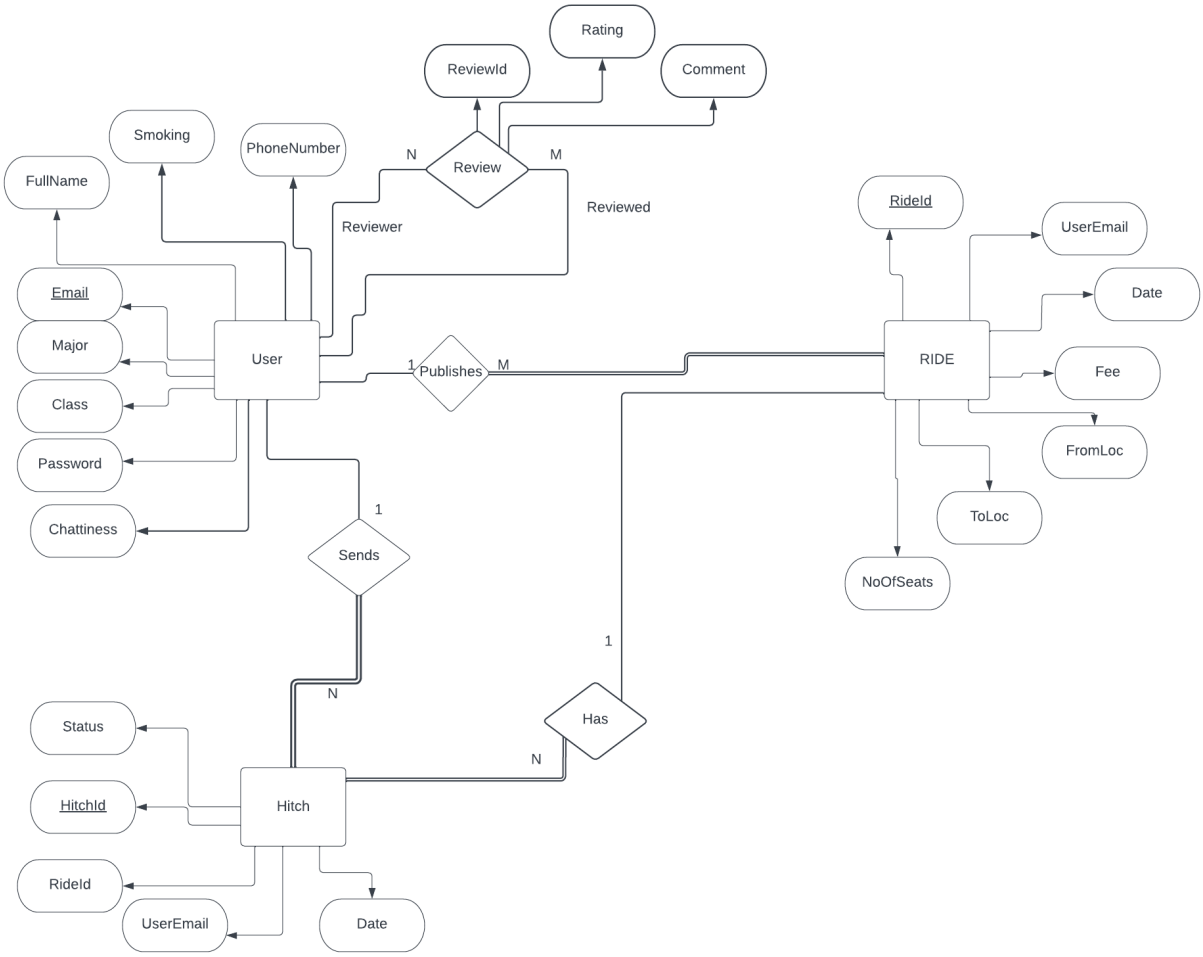
KUTRAVEL

BERAT KARAYILAN, KUTLUHAN PALALIOĞLU,
BEGÜM ŞEN, UTKU ALTINTAŞ

Project Description

We have developed a hitchhiking app that will make it easier for students to hitchhike. Students who have a car will post rides. These rides will include data about date, starting and endpoints of the journey. Hitchhikers will be able to search for riders who will travel between the certain starting point and certain destination. Users will also be able to rate and review each other. In this way, hitchhikers will be able to find the most suitable and high quality ride for them.

Entity-Relationship Diagram



Relational Database Design

CREATE TABLE Ride (rideId INT, userEmail VARCHAR(30), date DATE, fee INT, fromLoc VARCHAR(255), toLoc VARCHAR(255), noOfSeats INT, PRIMARY KEY(rideId), FOREIGN KEY (userEmail) REFERENCES User(userEmail));

CREATE TABLE User (email VARCHAR(30), fullName VARCHAR(30), smoking INT, phoneNumber VARCHAR(30), password VARCHAR(50), major VARCHAR(100), class VARCHAR(30), chattiness INT, PRIMARY KEY(email));

CREATE TABLE Hitch (hitchId INT, rideId INT, userEmail VARCHAR(30), status INT, PRIMARY KEY(hitchId), FOREIGN KEY(userEmail) REFERENCES User(email), FOREIGN KEY(rideId) REFERENCES Ride(rideId));

CREATE TABLE Review (reviewerMail VARCHAR(30), reviewedMail VARCHAR(30), reviewId INT, rating INT, comment VARCHAR(255), FOREIGN KEY(reviewerMail) REFERENCES User(userEmail), FOREIGN KEY(reviewedMail) REFERENCES User(userEmail), PRIMARY KEY(reviewId));

Complex SQL Queries

1.

```
SELECT
  Ride.RideId,
  Ride.UserEmail,
  User.FullName,
  User.Major,
  Ride.Date,
  Ride.Fee,
  Ride.FromLoc,
  Ride.ToLoc,
  Ride.NoOfSeats,
  User.Smoking,
  User.Chattiness,
  AVG(Review.Rating) AS AveragePoints
FROM
  Ride
JOIN
  User ON Ride.UserEmail = User.Email
LEFT JOIN
  Review ON User.Email = Review.ReviewedMail
WHERE
  Ride.Date > CURDATE()
GROUP BY
```

```
    Ride.RideId
ORDER BY
    AveragePoints DESC;
```

This query is used for searching rides to hitchhike. With this query, information about each ride and their riders are collected. With using Group by and order by, we sort the rides according to their rider's average points. So user will see rides on the screen in a sorted order according to rider's average points.

2.

```
SELECT
    Ride.ToLoc AS destination,
    MIN(Ride.Fee) AS min_fee,
    User.FullName AS rider_name
FROM
    Ride
JOIN
    User ON Ride.UserEmail = User.email
GROUP BY
    Ride.ToLoc,
    User.FullName
HAVING
    MIN(Ride.Fee) = (
        SELECT
            MIN(Fee)
        FROM
            Ride AS R
        WHERE
            R.ToLoc = Ride.ToLoc
    );
```

This query is used on statistics page. It is used to present the cheapest ride going to each location. It demonstrates all the destinations of the rides that are currently available. It also demonstrates the cheapest ride going to that destination and the rider of that ride.

3.

```
SELECT
   email,
   COUNT(email) AS count
FROM
   User u2
WHERE
   (
      SELECT
         COUNT(*) AS HitchCount
      FROM
         Hitch h
      INNER JOIN
         Ride r ON h.RideId = r.RideId
      WHERE
         h.UserEmail = '\(User.sharedInstance.email)'
         AND r.UserEmail = u2.Email
         AND h.Status = 1
   )
   -
   (
      SELECT
         COUNT(*) AS ReviewCount
      FROM
         Review
      WHERE
         ReviewerMail = '\(User.sharedInstance.email)'
         AND ReviewedMail = u2.Email
   ) > 0
GROUP BY
   email;
```

The query finds which user our logged in user can make a review. In the inner query, to make a review first we check how many times our logged in user sent a hitchhike request to the user that is accepted (hitchCount) and then we subtract it with how many times our logged in user already made a review to that user. If the result is positive then we understand that our logged in user can make a review to that user.The in the outer query we take the emails of that users that match with the condition. To make it clearer, let's assume Begüm made a hitch request to Utku 4 times and they are all accepted. This means that Begüm shared a ride with Utku 4 times. Thus Begüm has the chance to make 4 separate reviews to Utku. And let's assume Begüm already made 3 reviews. Thus we subtract 3 from 4 and leave with 1 which is greater than 0 thus Begüm can make a review to Utku.

4.

```
SELECT
   User.major,
   AVG(Ride.Fee) AS average_fee
FROM
   Ride
JOIN
   User ON Ride.UserEmail = User.email
GROUP BY
   User.major;
```

This query is used to see average fee's requested by students grouped by their majors. This is on the statistics page. This information can be used gain information about how students from different majors set fees of their rides.

5.

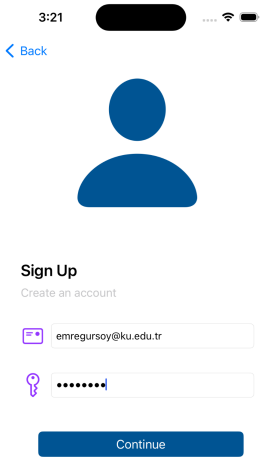
```
SELECT
   User.FullName
FROM
   User
JOIN
   Review ON User.Email = Review.ReviewedMail
WHERE
   Review.Comment LIKE '%Great%'
GROUP BY
   User.FullName
HAVING
   AVG(Review.Rating) >= 4;
```

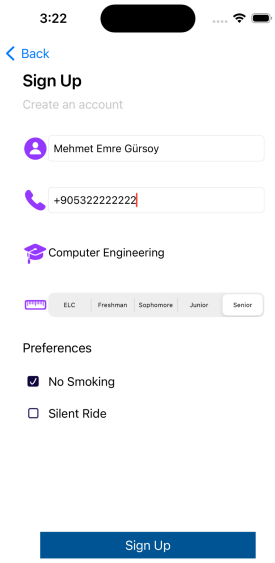
This query is used to identify gold users. Gold users are users that have average rating larger than 4 and has a comment which has word great in it. This query can be used to identify the users that has high average ratings and has at least one comment which has the word great. Since comments can be more descriptive than average points. We included comments while deciding on which users are gold users.

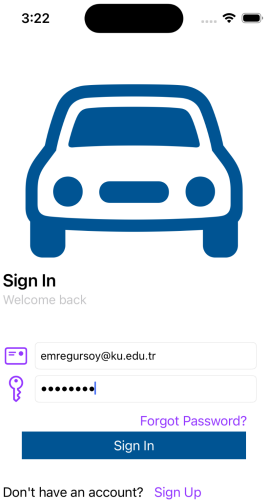
Working App with ScreenShots

1. SignUp & LogIn

The user first need to SignUp to use the app. If the user already signedUp to the app, they can directly SignIn to use the app.

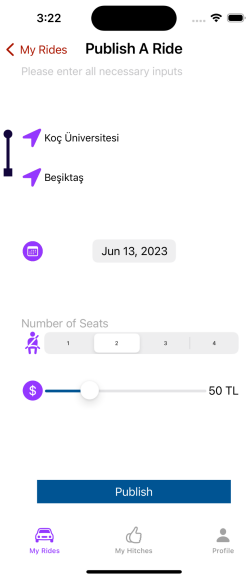


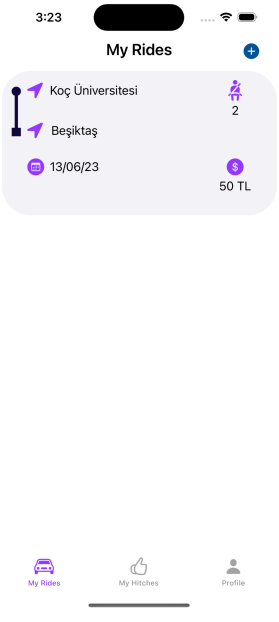


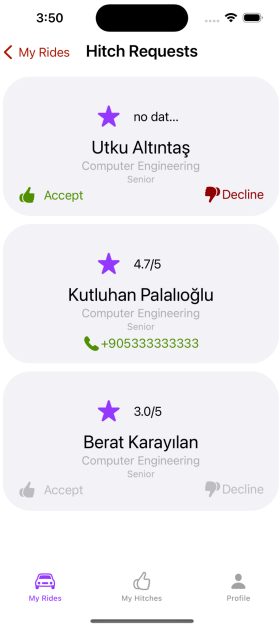


2. Rides Screen (Publish & MyRides)

The user can publish a new Ride, see their previously published Rides and their hitchhike Requests by clicking to the cells.

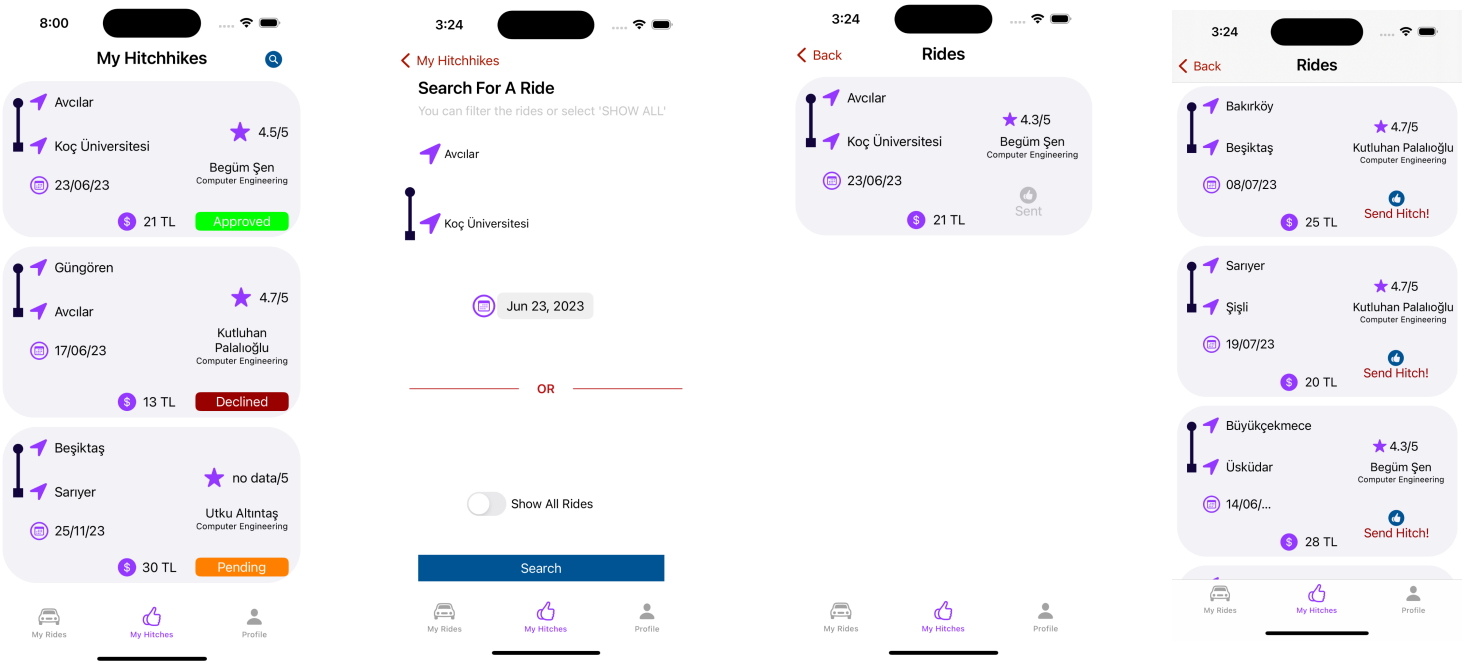






3. Hitchhike Screens (MyHitches & Search For A Ride & Rides)

The user can see their previous hitchhike request that they sent to a Ride shared by another user. To search for another ride, they have two options. If they do not choose to see all rides, they can specify their from and to locations and the date. Then only the rides that has the same

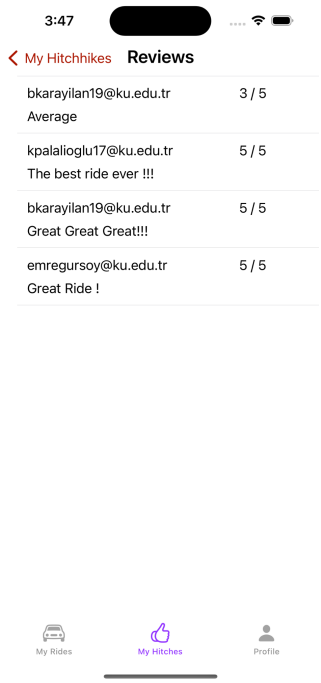


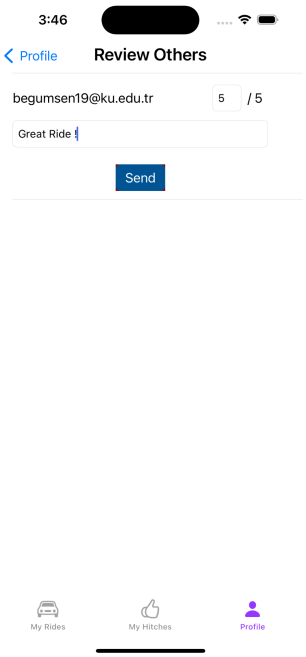
from & to destination and date will be shown. Otherwise by tuning the Show All Rides option they can see all of the rides published by other users no matter the destination or date. In both of the options the rides will be sorted using the average review score of the users.

4. Profile & Review & Stats Screens

The profile screen show the information related to the logged in user. From this screen the user can make review to other users or see their own review. An important reminder that if you click any of the cell in other screen that has a review point present you can see the review of that user. In the statistics screen, we have 4 statistics. In the first statistics we list average fees by major. User can see average of fees set by people from different majors. In the second statistic we list minimum fee for destinations. This shows all the destinations that has a ride in the future along with the fee that is lowest among other rides with same destination. In the third one we list gold users. Gold user are users with average points more than 4 and has a comment to their name that includes word great. In the last statistic we have most popular destinations. It lists the destinations by the number of rides which includes that destination.


3:22
Profile
Sign Out
Mehmet Emre Gürsoy
Computer Engineering
Senior
+905322222222
emregursoy@ku.edu.tr
Preferences
NoSmoking
SilentRide
Review Others
My Reviews
Statistics
My Rides
My Hitches
Profile


3:47
My Hitchhikes
Reviews
bkarayilan19@ku.edu.tr
3 / 5
Average
kpalalioglu17@ku.edu.tr
5 / 5
The best ride ever !!!
bkarayilan19@ku.edu.tr
5 / 5
Great Great Great!!!
emregursoy@ku.edu.tr
5 / 5
Great Ride !
My Rides
My Hitches
Profile


3:46
Profile
Review Others
begumsen19@ku.edu.tr
5
/ 5
Great Ride !
Send
My Rides
My Hitches
Profile


3:47
Profile
Statistics
AVERAGE FEE BY MAJORS:
MAJOR || AVERAGE FEE
Physics || 51.2
Chemistry || 29.5
Business Administration || 22.75
Minimum FEE FOR DESTINATIONS:
DESTINATION || MIN FEE || RIDER
Çekmeköy || 15 || Arda Aliz
Esenler || 23 || Arda Aliz
Sultanbeyli || 26 || Arda Aliz...
GOLD USERS:
Kutluhan Palalıoğlu
Begüm Şen
MOST POPULAR DESTINATIONS:
DESTINATION || RIDE COUNT
Beşiktaş || 7
Sarıyer || 5
Fatih || 4...
My Rides
My Hitches
Profile