CPSC 471: Final Report

Group Number 3
TA: Ibrahim Karakira

Tanner Collin, Jordan Heinrichs, Ali Waseem

Contents

1 Introduction

This report discusses the technical specifications and design of Renti, the peer to peer car rental service. The goal of Renti is to allow the users to easily and effortlessly rent out their vehicles while they are not using them. We have a similar business model to Airbnb except for cars. This report will cover the system requirements, deviations from proposed solution, entity-relationship design, object orientated model, and lastly provide a user manual.

2 System Requirements

3 Deviations from Original Requirements
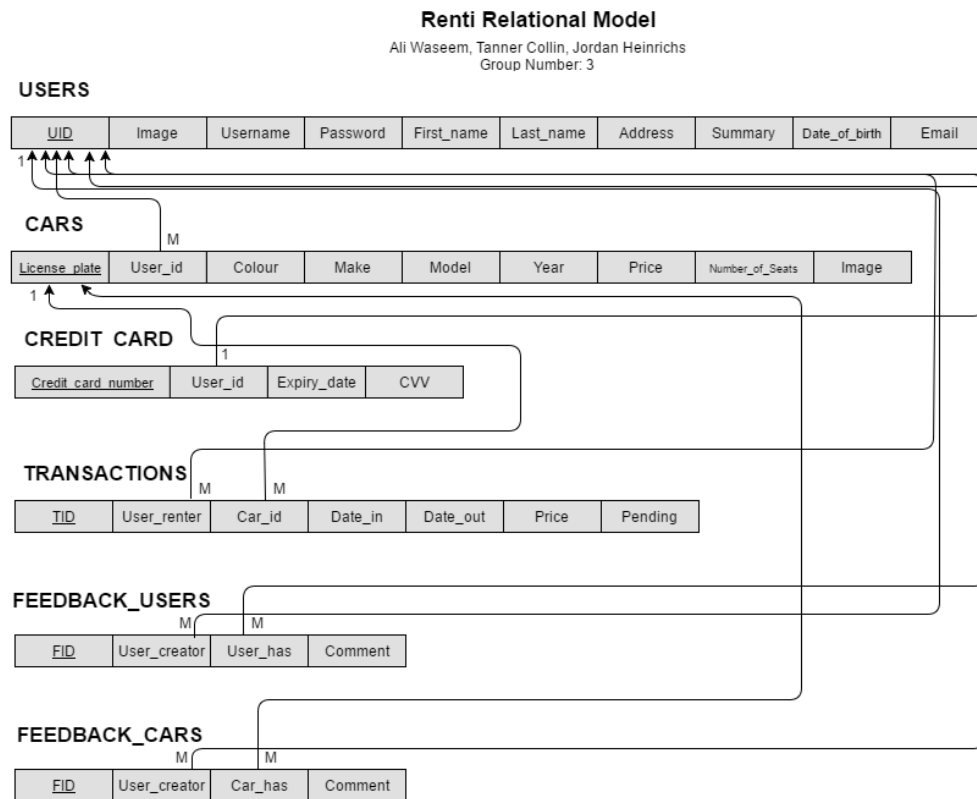
4 Entity-Relationship design

Figure 1: Relational Model

5 Object Orientated Model

The object orientated model is is shown in Figure 2. For each of the routes the SQL query is listed.

5.1 Users

The model of the user:
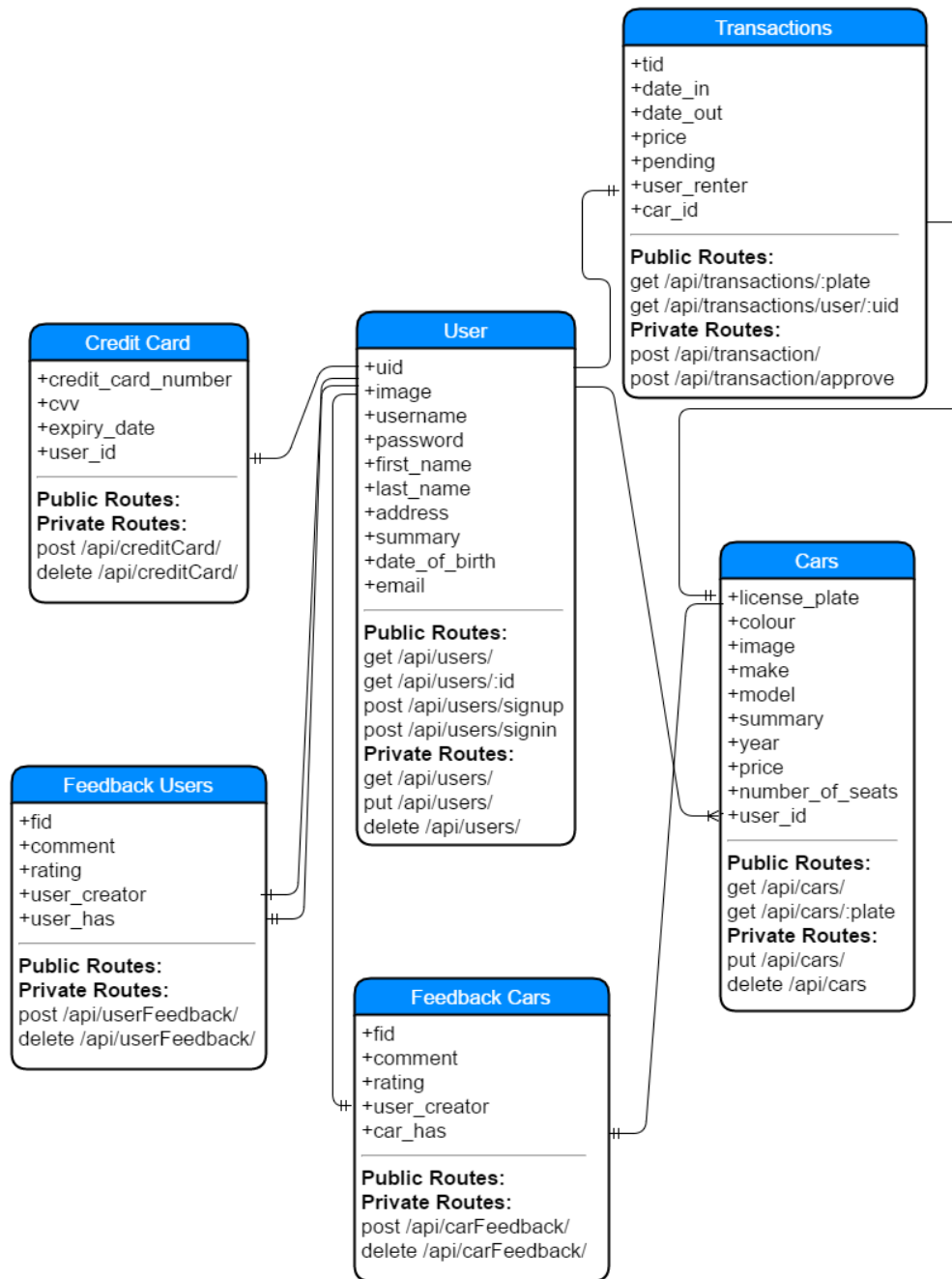
Figure 2: Database Object Orientated Model

5.1.1 The public routes:

get /api/users/

```
SELECT u.uid, u.first_name, u.last_name, u.address, u.username, u.
    email, u.image, u.summary, f.*, userF.uid, userF.first_name, userF.
    last_name, userF.address, userF.username, userF.email, userF.image,
     userF.summary, c.*, carF.*, carFUser.uid, carFUser.first_name,
    carFUser.last_name, carFUser.address, carFUser.username, carFUser.
```

```sql
    email, carFUser.image, carFUser.summary, t.*, userT.uid, userT.
    first_name, userT.last_name, userT.address, userT.username, userT.
    email, userT.image, userT.summary
FROM USER AS u, FEEDBACK_USERS AS f, USER AS userF, USER AS userT,
    USER AS carFUser, CARS AS c, FEEDBACK_CARS AS carF
WHERE u.uid=f.user_has AND c.user_id=u.uid AND c.license_plate=carF.
    car_has AND carFUser.uid=carF.user_creator AND c.license_plate=t.
    car_id AND t.user_renter=userT.uid;
```

get /api/users/:id

```sql
SELECT u.uid, u.first_name, u.last_name, u.address, u.username, u.
    email, u.image, u.summary, f.*, userF.uid, userF.first_name, userF.
    last_name, userF.address, userF.username, userF.email, userF.image,
     userF.summary, c.*, carF.*, carFUser.uid, carFUser.first_name,
    carFUser.last_name, carFUser.address, carFUser.username, carFUser.
    email, carFUser.image, carFUser.summary, t.*, userT.uid, userT.
    first_name, userT.last_name, userT.address, userT.username, userT.
    email, userT.image, userT.summary
FROM USER AS u, FEEDBACK_USERS AS f, USER AS userF, USER AS userT,
    USER AS carFUser, CARS AS c, FEEDBACK_CARS AS carF, TRANSACTIONS as
     t
WHERE u.uid="id" AND u.uid=f.user_has AND c.user_id=u.uid AND c.
    license_plate=carF.car_has AND carFUser.uid=carF.user_creator AND c
    .license_plate=t.car_id AND t.user_renter=userT.uid;
```

post /api/users/signup

```sql
INSERT INTO USER (image, username, password, first_name, last_name,
    address, summary, date_of_birth, email)
VALUES ('URL_OF_USER_PHOTO', 'tcollin', 'test', 'Tanner', 'Collin', '
    123 place rd NW', 'Hi. I love for you to rent one of my cars!',
    730869558, 'test@someemail.com');
```

post /api/users/signin

```sql
SELECT username, password FROM USER where username='tcollin', password
    ='test';
```

5.1.2 The private routes:

get /api/users/

```sql
SELECT u.uid, u.first_name, u.last_name, u.address, u.username, u.
    email, u.image, u.summary, f.*, userF.uid, userF.first_name, userF.
    last_name, userF.address, userF.username, userF.email, userF.image,
     userF.summary, c.*, carF.*, carFUser.uid, carFUser.first_name,
    carFUser.last_name, carFUser.address, carFUser.username, carFUser.
    email, carFUser.image, carFUser.summary, t.*, userT.uid, userT.
    first_name, userT.last_name, userT.address, userT.username, userT.
    email, userT.image, userT.summary
```

```
FROM USER AS u, FEEDBACK_USERS AS f, USER AS userF, USER AS userT,
    USER AS carFUser, CARS AS c, FEEDBACK_CARS AS carF, TRANSACTIONS as
    t, CREDIT_CARD AS cred
WHERE u.uid=f.user_has AND c.user_id=u.uid AND c.license_plate=carF.
    car_has AND carFUser.uid=carF.user_creator AND c.license_plate=t.
    car_id AND t.user_renter=userT.uid AND cred.user_id=u.uid;
```

put /api/users/

```
UPDATE USER SET summary='hi my name is tanner' WHERE uid='1';
```

delete /api/users/

```
DELETE FROM USER WHERE uid='1', username='tcollin', password='test';
```

5.2   Credit Card

The model of the credit card, there are no public routes for credit cards because of security reasons.

5.2.1   The private routes:

post /api/creditCard/

```
INSERT INTO CREDIT_CARD (credit_card_number, cvv, expiry_date, user_id
    )
VALUES (347249711260948, 433, 0617, 1);
```

delete /api/creditCard/

```
DELETE FROM CREDIT_CARD as c INNER JOIN USER as u ON c.user_id = a.uid
    WHERE
a.username='tcollin' AND a.password='test';
```

5.3   Cars

The model of the cars, contains both public and private routes.

5.3.1   The public routes:

get /api/cars/

```
SELECT c.*, carF.*, carOwner.uid, carOwner.first_name, carOwner.
    last_name, carOwner.address, carOwner.username, carOwner.email,
    carOwner.image, carOwner.summary, carFcreator.uid, carFcreator.
    first_name, carFcreator.last_name, carFcreator.address, carFcreator
    .username, carFcreator.email, carFcreator.image, carFcreator.
    summary, userF.*, userFcreator.uid, userFcreator.first_name,
    userFcreator.last_name, userFcreator.address, userFcreator.username
    , userFcreator.email, userFcreator.image, userFcreator.summary,
FROM CARS AS c, FEEDBACK_CARS AS carF, USER AS carOwner, USER AS
    carFcreator, FEEDBACK_USERS AS userF, USER AS userFcreator
WHERE c.license_plate=carF.car_has AND carFcreator.uid=carF.
    user_creator AND c.user_id=carOwner.uid AND userF.user_has=carOwner
    .uid AND userF.user_creator=userFcreator.uid;
```

get /api/cars/:plate

```sql
SELECT c.*, carF.*, carOwner.uid, carOwner.first_name, carOwner.
    last_name, carOwner.address, carOwner.username, carOwner.email,
    carOwner.image, carOwner.summary, carFcreator.uid, carFcreator.
    first_name, carFcreator.last_name, carFcreator.address, carFcreator
    .username, carFcreator.email, carFcreator.image, carFcreator.
    summary, userF.*, userFcreator.uid, userFcreator.first_name,
    userFcreator.last_name, userFcreator.address, userFcreator.username
    , userFcreator.email, userFcreator.image, userFcreator.summary,
FROM CARS AS c, FEEDBACK_CARS AS carF, USER AS carOwner, USER AS
    carFcreator, FEEDBACK_USERS AS userF, USER AS userFcreator
WHERE c.license_plate="plate" AND c.license_plate=carF.car_has AND
    carFcreator.uid=carF.user_creator AND c.user_id=carOwner.uid AND
    userF.user_has=carOwner.uid AND userF.user_creator=userFcreator.uid
    ;
```

5.3.2 The private routes:

post /api/cars/

```sql
INSERT INTO CARS (license_plate, model, make, year, number_of_seats,
    price, colour, image, summary, user_id)
VALUES ('892WSM', 'Focus RS', 'Ford', 2016, 4, 200, 'Blue', '
    URL_OF_CAR_PHOTO', 'This car is fast', 2);
```

put /api/cars/

```sql
UPDATE CARS SET model='VW', make='Golf R' WHERE license_plate='892WSM'
     AND user_id='2';
```

delete /api/cars/

```sql
DELETE FROM CARS AS c WHERE c.uid='2' AND c.license_plate='892WSM';
```

5.4 Transactions

5.4.1 The public routes:

get /api/transactions/

```sql
SELECT * FROM TRANSACTIONS AS t WHERE t.plate='123ABC';
```

get /api/transactions/user/:uid

```sql
SELECT t.*, c.*,  u.uid, u.first_name, u.last_name, u.address, u.
    username, u.email, u.image, u.summary
FROM TRANSACTIONS AS t, CAR AS c, USER AS u,
WHERE u.uid="uid" AND c.user_id=u.uid AND t.plate=c.license_plate;
```

5.4.2 The private routes:

post /api/transactions/

```sql
INSERT INTO TRANSACTIONS (date_in, date_out, price, pending,
    user_renter, car_id)
VALUES ('2016-3-27', '2016-3-21', 40, 1, '2','123ABC');
```

post /api/transactions/approve

```sql
UPDATE TRANSACTIONS as t SET pending='0' WHERE c.tid='10' AND EXISTS (
    Select * FROM CARS as c WHERE t.license_plate=c.license_plate AND c
    .user_id='2');
```

5.5 Car Feedback

There are no public routes with car feedback. The feedback is returned with a car get requests.

5.5.1 The private routes:

post /api/carfeedback/

```sql
INSERT INTO FEEDBACK_CARS (comment, rating, user_creator, car_has)
VALUES ('This car is really shiny.', '5', '2', '892WSM');
```

delete /api/carfeedback/

```sql
DELETE FROM FEEDBACK_CARS AS f WHERE f.fid='7' AND f.user_creator='2';
```

5.6 User Feedback

There are no public routes with user feedback. The feedback is returned with a user get request.

5.6.1 The private routes:

post /api/userfeedback/

```sql
INSERT INTO FEEDBACK_USERS (comment, rating, user_creator, user_has)
VALUES ('This user is not friendly.', '2', '2', '1');
```

delete /api/userfeedback/

```sql
DELETE FROM FEEDBACK_USERS WHERE fid='6' AND user_creator='2';
```

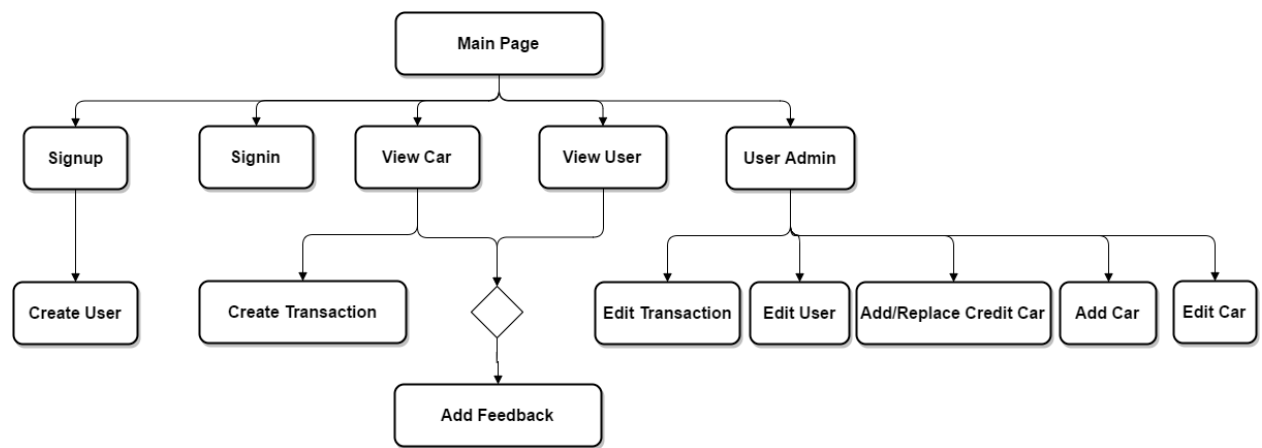# 6 General Hierarchical Structure

7 User Manual

8 Conclusion

9 Appendix A: Filler Data

Figure 3: Hierachical design of the application