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

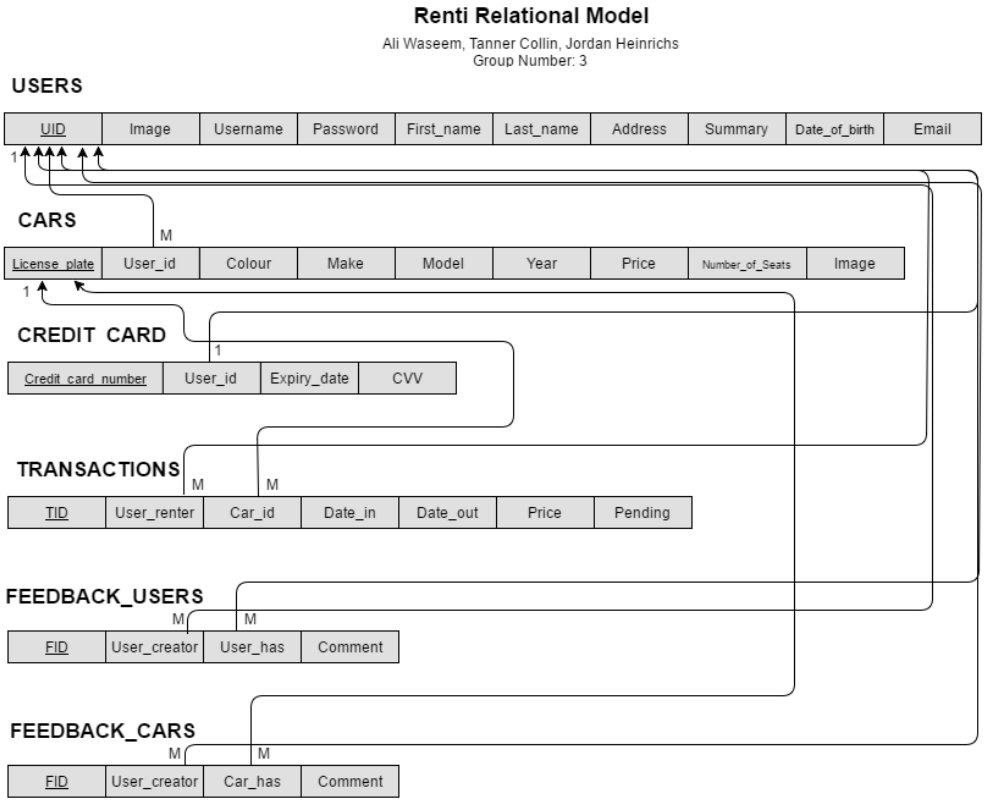

Figure 1: Relational Model

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:

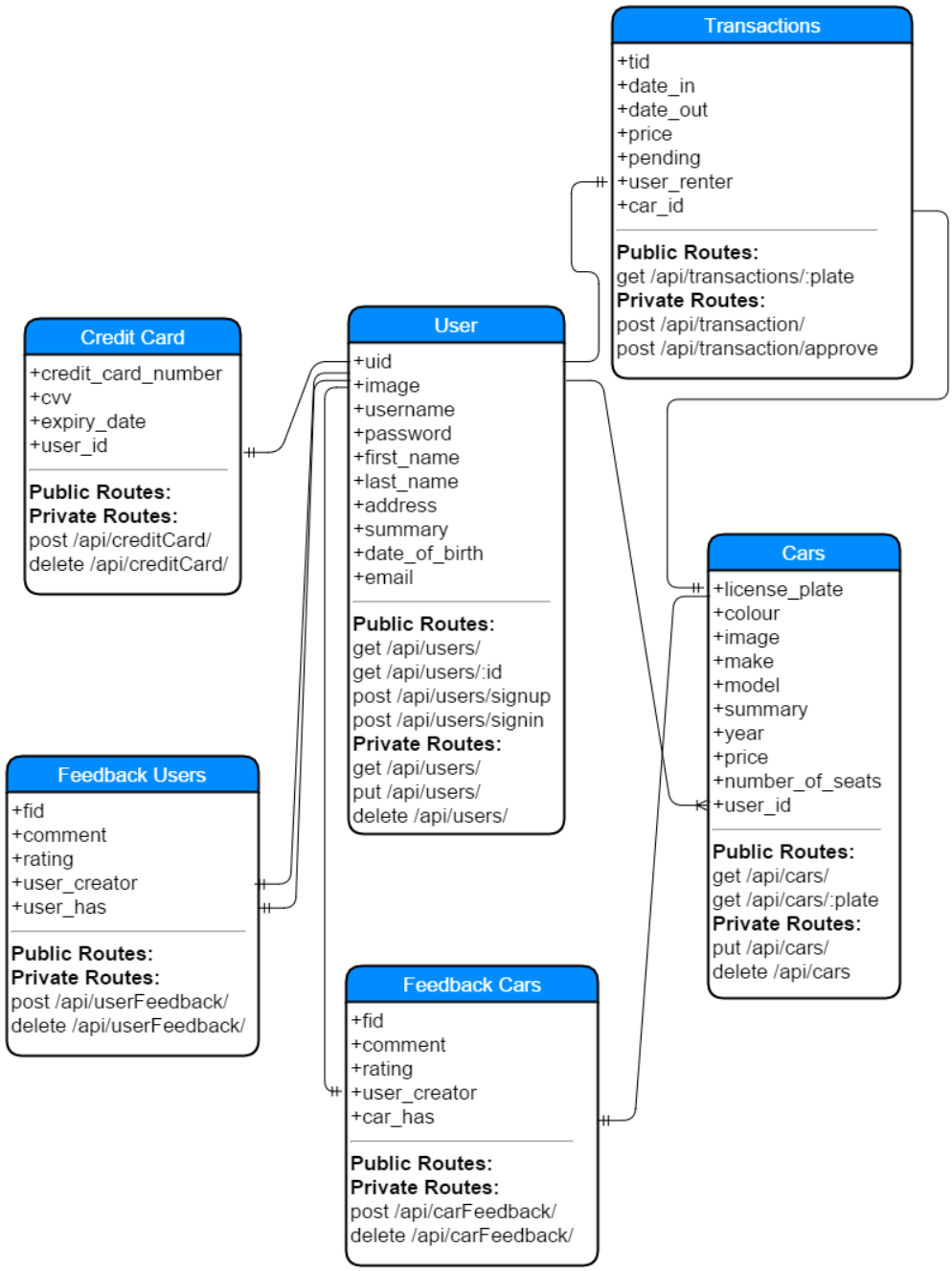

Figure 2: Database Object Orientated Model

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.* FROM USER AS u, FEEDBACK_USERS as f
    WHERE u.uid=f.user_has;
```

get /api/users/:id

```
SELECT uid, first_name, last_name, address, username, email, image,
    summary FROM USER WHERE uid="id";
```

post /api/users/signup

```
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

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

5.1.2 The private routes:

get /api/users/

```
SELECT * FROM USER where uid='1', username='tcollin', password='test';
```

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.*, f.* FROM CARS AS c, FEEDBACK_CARS as f WHERE c.
    license_plate=f.car_has;
```

get /api/cars/:plate

```
SELECT c.*, f.*  FROM CARS AS c, FEEDBACK_CARS as f WHERE
    license_plate='plate' AND c.license_plate=f.car_has;
```

5.3.2 The private routes:

post /api/cars/

```
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/

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

delete /api/cars/

```
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/

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

5.4.2 The private routes:

post /api/transactions/

```
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

```
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/

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

delete /api/carfeedback/

```
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/

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

delete /api/userfeedback/

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

6 General Hierarchical Structure

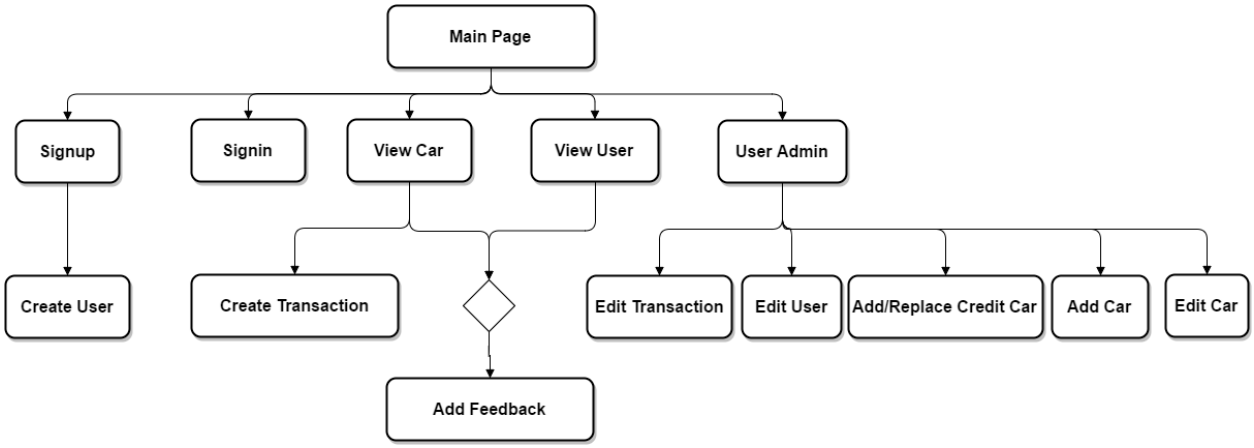

Figure 3: Hierachical design of the application

7 User Manual

8 Conclusion

9 Appendix A: Filler Data