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

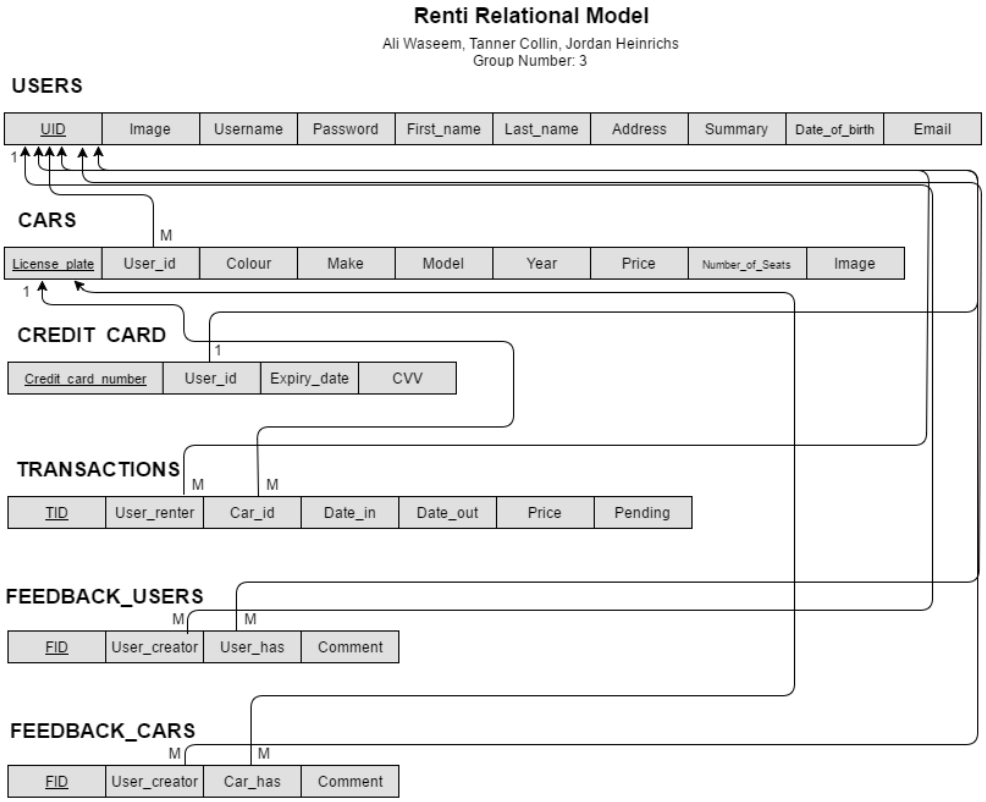

Figure 1: Relational Model

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:

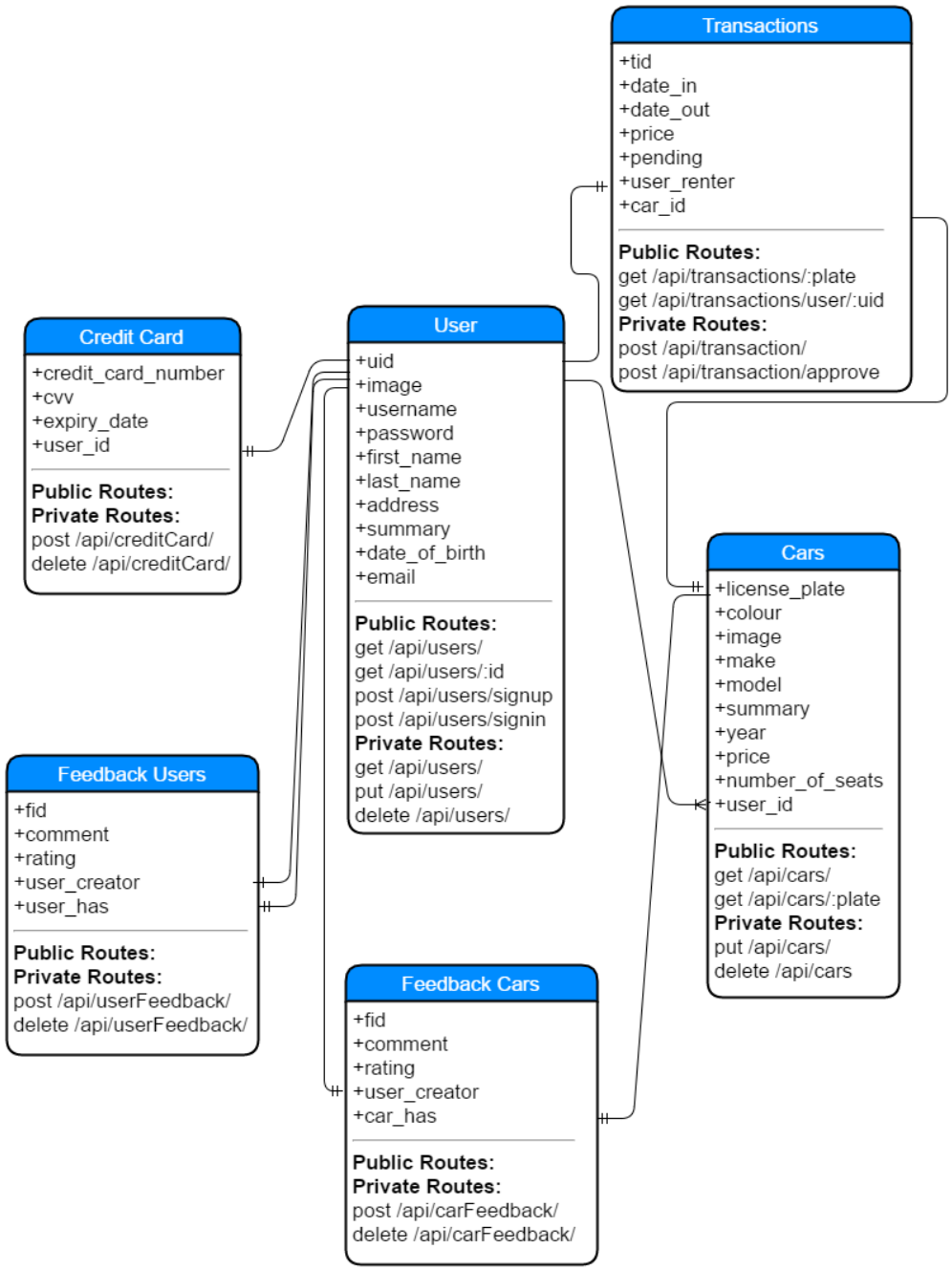

Figure 2: Database Object Orientated Model

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.
```

```
    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

```
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

```
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 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

```
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/

```
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';
```

get /api/transactions/user/:uid

```
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/

```
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

This is the hierarchical design of Renti. Most of the database options are available within the user admin page. Otherwise the user is limited to creating transactions and adding feedback.

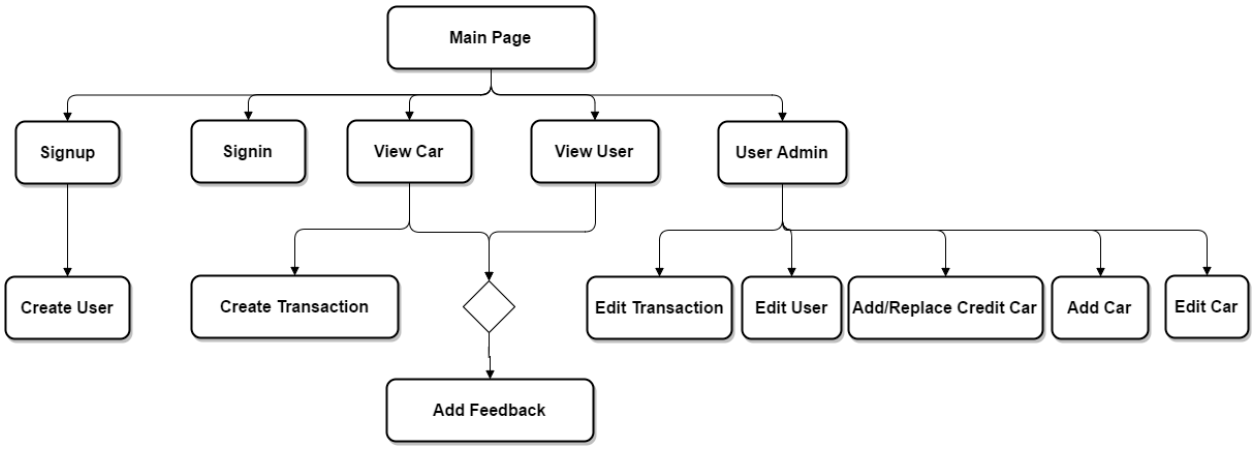

Figure 3: Hierarchical design of the application

7 User Manual

8 Conclusion

9 Appendix A: Filler Data

The following is the filler data that is used to set up the database. It will be automatically added during an 'npm install'.

```
var bcrypt = require("bcrypt"); // eslint-disable-line

exports.seed = function(knex, Promise) {
    return Promise.join(

        // Inserts entries for the user table
        knex("users").insert({
            username: "tcollin",
            first_name: "Tanner",
            last_name: "Collin",
            date_of_birth: "730869558",
            address: "123 place rd NW",
            email: "test@someemail.com",
            summary: "Hi. I love for you to rent one of my cars!",
            image: "https://cdn4.iconfinder.com/data/icons/SIGMA/
               project_managment/png/400/tester.png",
            password: bcrypt.hashSync("test", 10)
        }),

        knex("users").insert({
            username: "awaseem",
            first_name: "Ali",
            last_name: "Waseem",
```

```
    date_of_birth: "749186517",
    address: "123 newway rd NW",
    email: "hello@someemail.com",
    summary: "Hi. Please rent one of my cars!",
    image: "https://cdn4.iconfinder.com/data/icons/SIGMA/
       project_managment/png/400/tester.png",
    password: bcrypt.hashSync("test", 10)
}),

knex("users").insert({
    username: "jheinrichs",
    first_name: "Jordan",
    last_name: "Heinrichs",
    date_of_birth: "738559317",
    address: "123 cotunk rd NW",
    email: "hi@someemail.com",
    summary: "Hi. check out my cars for rent!",
    image: "https://cdn4.iconfinder.com/data/icons/SIGMA/
       project_managment/png/400/tester.png",
    password: bcrypt.hashSync("test", 10)
}),

// Credit card for Tanner
knex("credit_card").insert({
    credit_card_number: "371131951383474",
    cvv: 123,
    expiry_date: "0918",
    user_id: 1
}),

// Credit card for Ali
knex("credit_card").insert({
    credit_card_number: "347126747971237",
    cvv: 123,
    expiry_date: "0318",
    user_id: 2
}),

// Credit card for Jordan
knex("credit_card").insert({
    credit_card_number: "372552024488355",
    cvv: 123,
    expiry_date: "0118",
    user_id: 3
}),

// Car for Jordan
knex("cars").insert({
    license_plate: "bmz-123",
```

```
    colour: "gray",
    image: "https://upload.wikimedia.org/wikipedia/commons/e/ef
       /06-07_Honda_Civic_LX_Sedan.jpg",
    make: "honda",
    model: "civic",
    summary: "A clean honda civic up for rent!",
    year: 2009,
    price: 40.00,
    number_of_seats: 4,
    user_id: 2
}),

// Car for Jordan
knex("cars").insert({
    license_plate: "agh-234",
    colour: "red",
    make: "mazda",
    image: "http://www.gunaxin.com/wp-content/uploads
       /2013/08/2014-Mazda6-01.jpg",
    model: "6",
    summary: "A nice red car",
    year: 2006,
    price: 30.00,
    number_of_seats: 4,
    user_id: 3
}),

// Car for Jordan
knex("cars").insert({
    license_plate: "sdf-2344",
    colour: "gray",
    make: "BMW",
    image: "https://upload.wikimedia.org/wikipedia/commons
       /1/18/2012_BMW_320d_(F30_MY13)_Luxury_Line_sedan_
       (2015-07-24)_01.jpg",
    model: "3 series",
    summary: "A nice BMW for rent!",
    year: 2008,
    price: 60.00,
    number_of_seats: 4,
    user_id: 3
}),

// Car for Tanner
knex("cars").insert({
    license_plate: "ign-2344",
    colour: "gray",
    make: "Ford",
```

```
    image: "http://www.torquenews.com/sites/default/files/image
       -106/13f150-lariat_02_hr.jpg",
    model: "F-150",
    summary: "A big truck for rent!",
    year: 2010,
    price: 20.00,
    number_of_seats: 4,
    user_id: 1
}),

// Feedback for Tanner created by Ali
knex("feedback_users").insert({
    comment: "Tanner was keeps his car very clean!",
    rating: 5,
    user_creator: 2,
    user_has: 1
}),

// Feedback for Tanner created by Jordan
knex("feedback_users").insert({
    comment: "Tanner was very rude!",
    rating: 2,
    user_creator: 3,
    user_has: 1
}),

// Feedback for Ali created by Tanner
knex("feedback_users").insert({
    comment: "Ali was very nice!",
    rating: 5,
    user_creator: 1,
    user_has: 2
}),

// Feedback for Ali created by Jordan
knex("feedback_users").insert({
    comment: "Ali was very humble!",
    rating: 3,
    user_creator: 3,
    user_has: 2
}),

// Feedback for Jordan created by Tanner
knex("feedback_users").insert({
    comment: "Jordan was very mean!",
    rating: 1,
    user_creator: 1,
    user_has: 3
}),
```

```
        // Feedback for Jordan created by Ali
        knex("feedback_users").insert({
            comment: "Jordan was very nice with his car!",
            rating: 5,
            user_creator: 2,
            user_has: 3
        }),

        // Feedback for Tanner's ford f150 by Ali
        knex("feedback_cars").insert({
            comment: "This truck was great for driving!",
            rating: 4,
            user_creator: 2,
            car_has: "ign-2344"
        }),

        // Feedback for Ali's honda civic by Tanner
        knex("feedback_cars").insert({
            comment: "This civic was great for driving!",
            rating: 5,
            user_creator: 1,
            car_has: "bmz-123"
        }),

        // Feedback for Jordans's bmw 3 series by Ali
        knex("feedback_cars").insert({
            comment: "This bmw was fun!",
            rating: 5,
            user_creator: 2,
            car_has: "sdf-2344"
        })
    );
};
```