CPSC 471: Initial Draft Design

Due on March 21
Group Number 3
TA: Ibrahim Karakira

Tanner Collin, Jordan Heinrichs, Ali Waseem

1 Introduction

This report contains the initial draft design and a progress report for our project Renti, a peer to peer vehicle renting service. The current progress will discuss the technologies chosen for the framework and much work is done implementing it. As our project uses an Object Relational Mapping (ORM) to access our database we have an object orientated model which will be outlined. Lastly, the draft of the user interface is discussed.

2 Current Progress and Technologies

For the back end of the project we are using Node.js with Express.js, Knex, and Bookshelf (this will be our ORM that handles our queries). The database is using SQLite3 with Bookshelf.js to provide the ORM interface. The back end is finished and the API is tested. As the front end is implemented there will likely need to be some changes to the API but it is in a state where work can be started on the front end for the web application and possibly the mobile application.

The front end will be implemented using React.js and Semantic-UI. The boiler plate code will be taken from Ali Waseem's blog including functions like user authentication. Each project member will be responsible for creating components to interact with the back end API. Work is currently started and the projected completion date for the front end is end of March.

3 Object Orientated Model

The object orientated model is is shown in Fig: 1. For each of the routes the SQL query is listed.

3.1 Users

The model of the user:

The public routes:

3.1.1 get /api/users/

```
SELECT uid, first_name, last_name, address, username, email, image,
    summary FROM USER;
```

3.1.2 get /api/users/:id

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

3.1.3 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');
```

3.1.4 post /api/users/signin

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

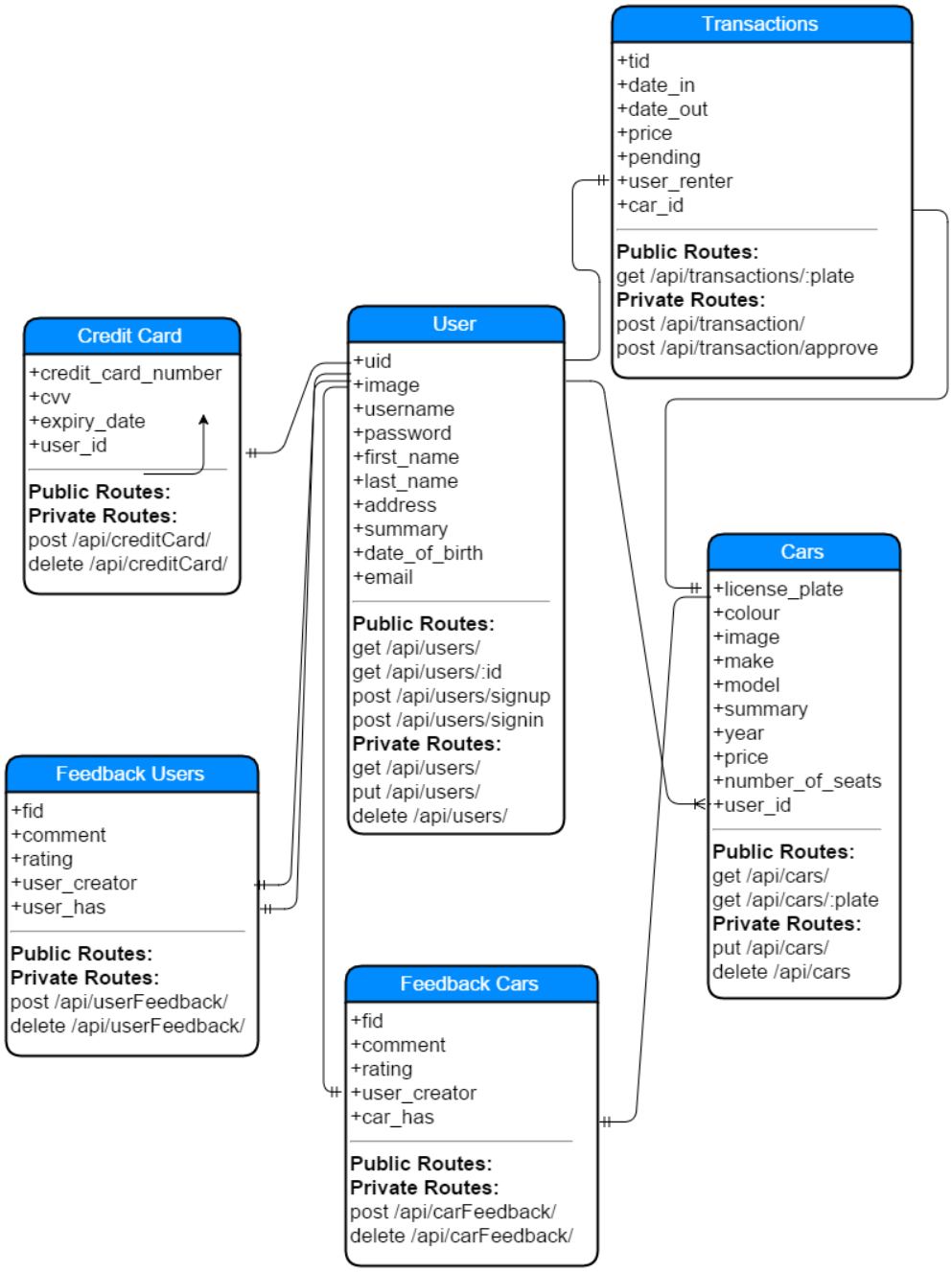

Figure 1: Database Object Orientated Model

The private routes:

3.1.5 get /api/users/

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

3.1.6 put /api/users/

```
UPDATE USER SET summary='hi my name is tanner' WHERE summary='Hi. I love
    for you to rent one of my cars!';
```

3.1.7 delete /api/users/

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

3.2 Credit Card

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

3.2.1 post /api/creditCard/

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

3.2.2 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';
```

4 User Interface Draft Design

5 Conclusion