

CSC/ECE 517 (OO Design and Development)

Program 2: Ruby on Rails

Due Date - September 25, 2017 (Monday) 11:59 PM

Car-rental app

You are tasked with creating a car-rental app.

- There can be multiple cars with the same model and manufacturer.
- The status of a car in the system may have one of the three values :
 - “reserved”
 - “checked out”
 - “available”
- A user can reserve a car or rent a car without a previous reservation. A user is not allowed to have more than one car rented at a time. Only after canceling a reservation or returning a rented car, can a user can reserve or rent another car.
- The reservation process is automatic, and does not require an admin to approve requests. If the status of a car is “available” and a user rents it, the status of that car immediately transitions to “reserved”. But an admin can edit the reservation later.
- When a reservation is made, it must specify the time that the car is to be picked up (“checked out”) and returned. In order to simulate checkout and returns actions in the real world, there should be buttons that a user can manually click to indicate that a car has been checked out or returned.
- The minimum rental period is 1 hour and the maximum is 10 hours. The system should cancel a reservation if a user doesn’t check out a car in time. The time elasticity is 30 minutes after the pickup time. For example, if a user reserved a car to pick up at 3:00pm, if the car was not checked out by 3:31pm, then the car should become “available” again.
- If a user doesn’t return a car on the date promised when it was checked out, the system will sent a notification (e.g. system message) to the user and automatically change the car status to “available”.
- A user can only reserve a car 7 days in the future. (for example, if today is 01/20/2017 1pm, a user can only reserve a car starting between 01/20/2017 1pm and 01/27/2017 1pm; he can’t reserve a car that will be checked out at 01/28/2017)
- A user can return a car before the specified return time by clicking the “return” button. If a user does this, then the car should become “available”.

- There's no elasticity in the return time. If the user doesn't return the car in time (at the specified return time), the car will automatically become "available" and other users can check out this car. For example, user A reserves a car from 2pm to 4pm, user B reserves the same car from 4pm to 6pm. If A doesn't return the car at 4pm, the system should automatically change the car status to "reserved" and user B can click "check out" button to pick up the car. If no user has reserved this car at 4pm, then the car should automatically become "available".
- There's no penalty when a user doesn't return a car. This app just has a simple payment system. A penalty would only make the calculation more complex.
- The rental charge in user profile is not a deposit. It's the total fee for the previous successful rental. The system doesn't need to check if a user has enough on deposit to check out a car because there's no deposit in this app.

There will be three types of users in the system:

- Superadmin
- Admin
- User

Superadmin

The system should have a preconfigured admin (superadmin). A superadmin should be capable of performing all operations performed by an Admin (listed below) and should also be able to manage admins (e.g. to delete other admins).

Superadmins should be able to

- Create new superadmins
- View the list of superadmins and their profile details (except password)
- Should not be able to delete other superadmins.

Admin

An admin user will have the following attributes: email (unique for each admin), name and password.

All admins can perform the following tasks:

- Log in with an email and password
- Edit their own profile

- Manage admins
 - Create new admins
 - View the list of all the admins and their profile details (except password)
 - Delete admins (except themselves and the superadmins)
- Manage Cars
 - Add a Car to the system. A Car will have the following attributes:
 - License-plate number (a 7-digit string unique for each Car)
 - Manufacturer
 - Model (String ... this app doesn't need to validate that the model is actually produced by the indicated manufacturer.)
 - Hourly Rental rate
 - Style (Coupe, Sedan or SUV)
 - Location (office where the car can be picked up)
 - Status to indicate availability:
 - Checked out - when a customer checks out the car or an admin checks out the car on behalf of a customer
 - Available - otherwise
 - View the list of all Cars
 - View the attributes of a Car. Also, show the user who has the car if its status is "Checked out" or "Reserved".
 - Edit the attributes of a Car
 - An admin can also change the status of a car i.e., she/he can reserve, check out or return a car on behalf of a customer.
 - When an admin reserves, checks out or returns a car on behalf of a customer, the information captured in your database should be the same as if the car was reserved or checked out by a customer..
 - An admin can edit a reservation on behalf of a customer.
 - View the checkout history of a car.
 - Delete a car from the system
- Manage customers
 - List all customers and profile attributes (except password), with an option to edit a particular customer
 - View the checkout history of a customer
 - Delete a customer

Customers

A customer will have the following attributes: email (unique for each member), name and password and rental charge. The rental charge is computed from (rental rate \times the number of hours the customer held the car)

Anyone can sign up as a customer using their email, name and password. After signup, they can perform the following tasks:

- Log in with email and password
- Edit their own profile
- Search cars using location or model or manufacturer or style or status
 - search results should be a list of cars matching the search criteria
- View the attributes of a car
- Reserve or check out a car if its status is “Available”, and, as part of both operations, specify a return time car.
- Return a checked-out car
- View their own checkout history

Miscellaneous

Testing

- Thoroughly test one Model and one Controller. Feel free to use any testing framework.

Deployment

Please ensure that your deployment is always accessible for grading. You can deploy your app to any of the following.

- Heroku or any similar PaaS (OpenShift etc) with free plans
- Amazon AWS

Bonus (Extra Credit)

You can do any or all of the below for extra credit (each item below is worth 5 points).

- If a car has status “Checked out”, then a customer can register to receive a notification email when the car becomes available.

- A customer can suggest a new car to be added to the system. An admin can view and approve the suggestions. An admin should be able to edit car attributes during the approval process. After approval, the car becomes available in the system.

Submission

Your submission in Expertiza should consist of the following-

- A link to your deployed application
- A link to your repository (use [github.ncsu.edu](https://github.com/ncsu.edu) to host your repository)
 - Credentials for the preconfigured admin and any other information that reviewers would find useful should be in the README.md file in this repository
 - Some information that reviewers may find useful is:
 - how to test various features (e.g., how to access certain pages, what details to enter in the form etc.)
 - use cases for some of the edge-case scenarios (e.g., what will happen if admin tries to delete a user who currently has a car checked out.).