Project Specification

Computing and Control Engineering 2023/2024

Project Task

It is necessary to develop a web application for a system that supports car rental.

Data Model

The application is used by 4 user groups (roles): Customer, Manager, Administrator. The application manages the following entities:

User

- Username (unique)
- Password
- First Name
- Last Name
- Gender
- Date of Birth
- Role (Customer, Manager, Administrator)
- All Rentals (if the user is a Customer)
- Shopping Cart (if the user is a Customer)
- Rent a Car Object (if the user is a Manager)
- Number of Accumulated Points (if the user is a Customer)
- Customer Type

Customer Type

- Type Name (e.g., Gold, Silver, Bronze)
- Discount (percentage used for rental price calculation during purchase)
- Required Points (the necessary points for a user to become, for example, a Gold customer)

Location

- Geographic Longitude
- Geographic Latitude
- Address in the format: street and number, city/town, postal code

Rent-a-car Object

- Name
- Vehicles available for rent
- Business hours of the object
- Status (Open or Closed)
- Location
- Logo (image)
- Rating

Vehicle

- Brand
- Model
- Price
- Type (car, van, mobile home, ...)
- Associated Object
- Transmission Type (manual, automatic)
- Fuel Type (diesel, gasoline, hybrid, electric)
- Fuel Consumption
- Number of Doors
- Passenger Capacity
- Description
- Image
- Status (Available, Rented)

Shopping Cart

- Vehicles in the cart
- User who owns the cart
- Price

Order

- Unique Order Identifier (10 characters)
- Rented Vehicles
- Rent-a-car object from which it was rented
- Date and time of rental
- Rental duration
- Price
- Customer (first name and last name)
- Status (Processing, Approved, Picked Up, Returned, Rejected, Canceled)

Comment

- Customer who left the comment
- Rent-a-car object the comment refers to
- Comment text
- Rating (on a scale from 1 to 5)

Functionalities

Registration: Unregistered users can register on the application by filling in fields including username, password (entered twice for confirmation), first name, last name, gender, and date of birth. After registration, they become Customers. Administrators are loaded programmatically from a text file and cannot be added later. Only administrators can create Managers. A Customer cannot become a Manager.

System Login: Unregistered users can log in to the system by entering their username and password for the account they registered. After successful login, the user is authenticated and can perform activities based on their role.

Rent-a-car Object

Administrators can create a new object. When creating one, they must input information such as name, location, business hours, and select a logo. Additionally, they need to choose a manager responsible for the object. When selecting a manager, only managers not responsible for any objects should be available for selection. If none are available, allow the administrator to register a new manager, who will be automatically linked to the newly added object.

The homepage should display all rent-a-car objects. This page is visible to all user types, including unregistered users. Objects should be displayed with open ones at the top. Display the following information for each object on this page: name, location, logo, and average rating.

Enable search for all objects based on the following criteria:

- Object name
- Vehicle type
- Location (user inputs the city or country name)
- Average rating When displaying search results, include the following information:
- Object name
- Location
- Object logo
- Average object rating

Provide sorting and filtering options for search results. Implement sorting in ascending or descending order based on the following parameters:

- Object name
- Location
- Average rating Enable filtering based on the following criteria:
- Vehicle type
- Vehicle fuel type
- Display only open objects.

Selecting a displayed object redirects the user to a page showing detailed information about that specific object, including:

- Name
- Business hours
- Status (Open or Closed)
- Location
- Logo (image)
- Object rating (if available)
- Comments about the object (if available)
- List of vehicles the object offers, along with all vehicle details. Display the address in the format defined within the Location entity. If an additional task 1 is implemented, show a map.

Managers can add new vehicles to the object they are responsible for. When creating a vehicle, they must input the brand, model, price, type, transmission type, fuel type, consumption, number of doors, passenger capacity, and an optional description. They should also upload an image.

In addition to adding new items, allow managers to edit the data of existing vehicles and delete vehicles.

Rental

Customers have the option to view all their rentals within their user profile.

Managers have the option to view all rentals related to the object they are responsible for.

Rentals can be searched and sorted. Searching can be done based on the following criteria:

- Object from which it was rented (only for Customers)
- Price (within a range)
- Rental date (from-to) Sorting can be done (ascending and descending) based on:
- Name of the object from which it was rented (only for Customers)
- Price
- Date

A Customer can rent a car from an object. The steps for renting are as follows:

1) The Customer enters the start and end date for which they need a vehicle. All available vehicles for that time period are displayed (vehicles with a status of "Available" and not reserved for the entered period) from all objects, along with their details (brand, model, price, type, transmission type, fuel type, consumption, number of doors, passenger capacity, description, image).
2) The Customer selects their desired vehicle from the list and adds it to the cart.
3) After finishing the selection, the Customer goes to the cart overview, which displays information about the vehicle (brand, model, price, image), as well as the total price. The Customer can remove a vehicle from the cart or change the quantity of vehicles in the cart (quantity cannot be less than 1). The total cart price updates with each change.
4) The Customer clicks on the "Rent" button, which creates an order. The created order is in the "Processing" status and is added to the list of orders related to the object from which it was rented.

When a user rents vehicles, they receive a certain number of points. This number of points is calculated using the following formula:

points = total_order_price / 1000 * 133

Customers have the option to cancel an order while it is in the "Processing" status. If they cancel the order, they lose points according to the following formula:

lost_points = total_order_price / 1000 * 133 * 4

The Manager responsible for the object can change the status to "Approved" or "Rejected." If the Manager rejects the order, they must provide a reason.

If the Manager approves the order, they can change the order status to "Picked Up." The order can transition to this status no earlier than the initial day of vehicle rental. All vehicles related to that order change their status to "Rented."

If the order is in the "Picked Up" status, the Manager can change the status to "Returned." After the order status changes to "Returned," all vehicles related to that order change their status to "Available."

Comments

After vehicles are returned (order status is "Returned"), the user who created the order can leave a comment and rate the object from which they rented.

Once a comment is created, it does not appear on the object's page until it is approved by the Manager (or it can be rejected).

Customers can only view comments that the Manager has approved.

Managers and Administrators can view all comments (both approved and rejected).

Users

All logged-in users can view their profiles and update their personal information.

Administrators have an overview of all users registered in the system.

Managers have an overview of their object, orders related to that object, and customers who have placed orders from that object.

Administrators have the ability to display all registered system users. They can search, filter, and sort users. Searching can be done based on:

- First Name
- Last Name
- Username Sorting should be implemented in ascending and descending order, and it can be done based on the following parameters:
- First Name
- Last Name
- Username
- Number of accumulated points

Filtering can be done based on the following parameters:

- Roles
- User Type

Implement a combined search for objects based on the previously mentioned criteria in the object search. A combined or multiple search means that the user can select multiple options for conducting the search.

Additional Notes

Deletion of all entities in the system is logical. All entities that can be added can also be deleted.
Depending on the specific implementation, students can freely extend the given entities or add others.
Students should independently define the points scale for User Types (e.g., to become Silver, a user needs to accumulate 3000 points and get a 3% discount on each rental, while to become Gold, they need to accumulate 4000 points and get a 5% discount on each rental).

Additional Tasks for Higher Grades

1. When selecting a location, use OpenLayers maps (https://openlayers.org/) or an alternative map service to search for the location of the object based on the location or input location when creating the object.
2. Allow administrators to view all "suspicious" users who frequently cancel their orders. If a user (Customer) cancels more than 5 orders within a month, they are considered suspicious and are listed separately. Allow the Administrator to block such users.
3. Administrators have the ability to block registered users (if they are not administrators). Blocked users cannot log in to their accounts and cannot perform any actions based on their role.

Evaluating the Project Task

For versioning the project, you should use Git. The repository should be accessible on https://gitlab.com. You need to add the web-e2-ftn account as a Maintainer.
After forming your team and project, provide the following information to your assistant:

- List of team members
- Link to the public repository

The Git history must clearly show the development process, with all team members involved. Projects with a small number of large commits will not be accepted.
The user interface design and styles (CSS) are left to the discretion of the student and are mandatory. You are allowed to use CSS libraries such as Bootstrap, Foundation, and Materialize.
The application should persistently store data in text files, either in CSV or JSON format.
The use of any database or Java object serialization is not allowed.
Permitted technologies include:

- Vue.js 1.x or 2.x for the frontend
- JAX-RS REST (Jersey) or Node.js for the backend

If you want to use additional libraries for the specified technologies, you should contact your assistant for permission.
The use of technologies not listed (e.g., Spring, programming languages other than Java and JavaScript) is not allowed.

For a grade of 6, the application does not need to support vehicle rental functionality. You should provide "hardcoded" values for orders and load them when the application starts.

For a grade of 8, additional tasks do not need to be implemented.

For a grade of 9 or 10, the application must support all the specified functionalities as well as the additional tasks.