Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Number	COE692
Course Title	Software Design and Architecture
Semester/Year	Winter Semester 2024
Instructor	Dr. Faezeh Ensan
TA Name	Bita

Lab/Tutorial Report No.	Lab 3

Report Title	Car Rental Management System

Section No.	07 & 02
Submission Date	
Due Date	

Student Name	Student ID	Signature*
Hamza Malik	501112545	HM
Kevin Bhatt	501093104	KM

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: http://www.ryerson.ca/content/dam/senate/policies/pol60.pdf*

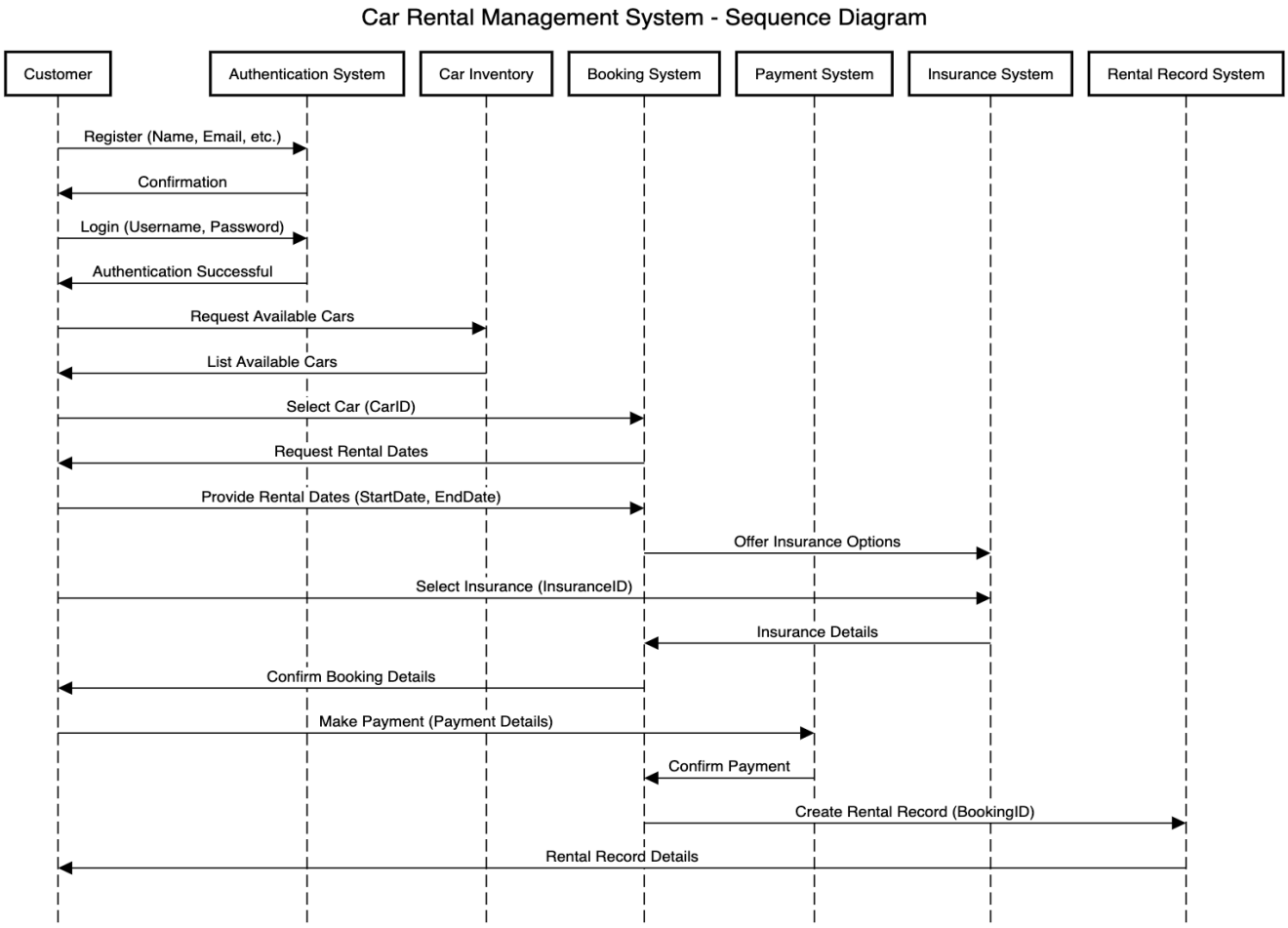
Exercise 2

You need to create sequence and component diagrams and also a description of the sequence and component diagrams:

- Create sequence diagrams for your use cases, you may need more than one sequence diagram for each use case, if the use case is a complex one. Zip all sequence diagrams and name them lab3_exc2_seq.zip. (10/20)
- Create a component diagram for your work including three components of GUI, Business, and Persistence. Create class diagrams inside each component. (Relationships between class does not need to be represented. Zip all diagrams and name them lab3_exc2_comp.zip. (5/20)
- Create a report using a word processor and describe your sequence diagram and your component diagram in at most two pages and store it as a single PDF and name it **Lab3_exc2_report.pdf**. (5/20)

Part 2: Design based on the N-Layer Architecture

Sequence Diagram



Sequence Diagram Process

Customer and Authentication System Interaction:

- The process begins with customers registering their details (name, email, etc.) with the authentication system.
- The system sends back a confirmation of the registration.
- The customer then proceeds to log in using their username and password.
- The authentication system verifies the details and sends an "Authentication Successful" message to the customer.

Customer and Car Inventory Interaction:

- Once authenticated, the customer requests a list of available cars from the car inventory.
- The car inventory system responds by listing the available cars.
- The customer selects a car by sending the chosen CarID to the booking system.

Customer and Booking System Interaction:

- The customer requests rental dates by specifying a start and end date for the rental period.
- The booking system acknowledges by providing the rental dates.

Insurance Interaction:

- The customer selects insurance by sending an InsuranceID to the booking system.
- The booking system forwards the InsuranceID to the insurance system, which in turn offers insurance options to the customer.
- The customer reviews and receives details about the insurance.

Booking and Payment Interaction:

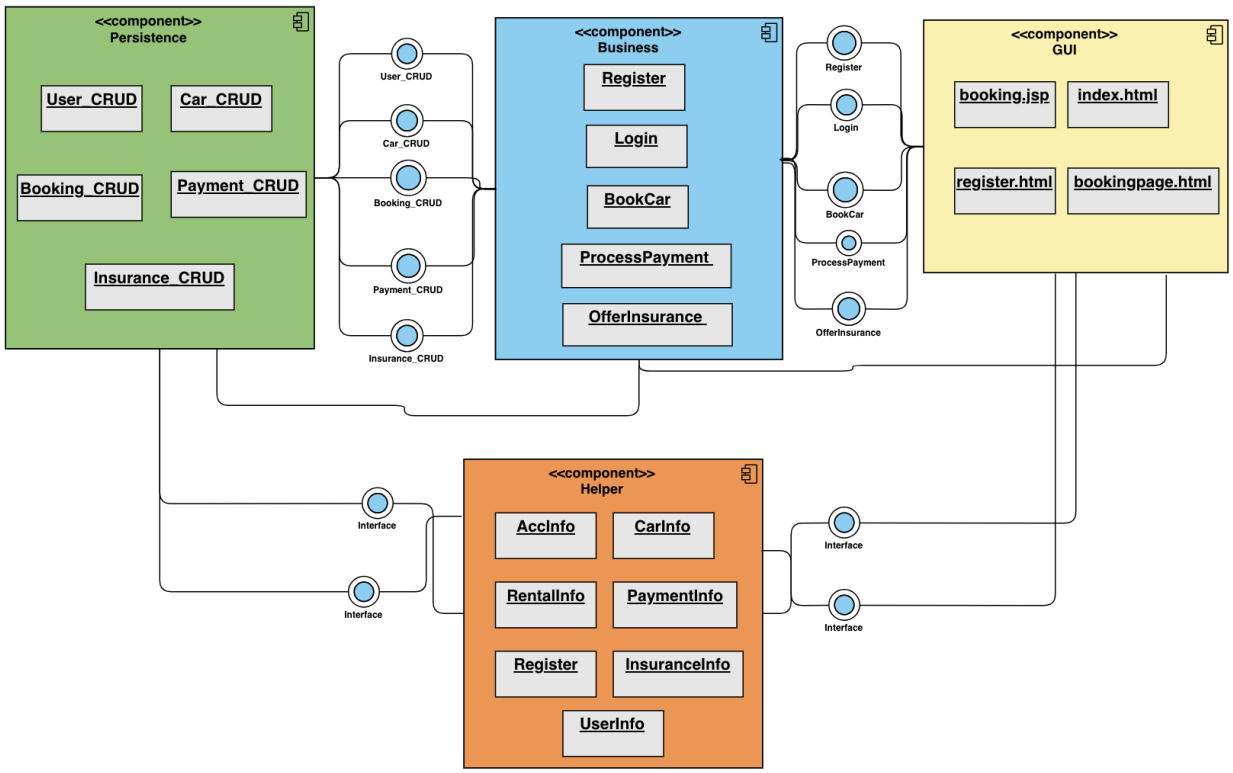
- The booking system sends the booking details back to the customer for confirmation.
- The customer confirms the details and proceeds to make a payment by sending payment details to the payment system.
- The payment system confirms the payment to the customer.

Rental Record Creation:

- Once the payment is confirmed, the booking system instructs the rental record system to create a rental record using the BookingID.
- The rental record system creates the record and sends the details back to the customer.

Customer: This is the actor who initiates the process. The lifeline begins when the customer starts interacting with the system, such as registering and logging in and ends after all transactions are completed and the rental record details are received.
Authentication System: This system handles the authentication of the customer. Its lifeline starts when the customer attempts to register and log in and ends after confirming successful authentication back to the customer.
Car Inventory: This subsystem is responsible for maintaining a list of cars that are available for rent. Its lifeline becomes active when the customer requests to view available cars and deactivates after providing the list of available vehicles to the customer.
Booking System: This is the central system that orchestrates the booking process. It interacts with the customer for various details such as car selection, rental dates, and insurance. Its lifeline starts with the car selection process and continues until it confirms the booking details with the customer.
Payment System: This system manages the financial transactions. It is activated when the customer is ready to make a payment and ends its activity once the payment is confirmed.
Insurance System: This subsystem offers and manages insurance options for the car rental. It becomes active when the customer selects insurance and ends after providing the insurance details to the customer.
Rental Record System: This system creates and maintains records of all rentals. It is activated towards the end of the process to create a rental record after payment is confirmed and deactivates after sending the rental record details to the customer.

Component Diagram



Component Diagram Process

Components

- Persistence: This component is responsible for saving and retrieving data from a database. It contains a Save User component and several interfaces, including:
    - RegisterServlet, LoginServlet.
- Business: This component contains the core logic of the car rental system. It extends the Helper component and contains several interfaces, including:
    - AccInfo, Carinfo, Loo (possibly a typo), RentalInfo, Paymentinfo, Insuranceinfo, BookingInfo.
- GUI: This component is responsible for rendering the user interface. It contains several interfaces, including:
    - Booking.jsp, register.html, BookingPage.html, Index.html.

Connections

- The Persistence component is connected to the Business component through the AccInfo interface.
- The Persistence component is connected to the RegisterServlet interface through the Save User component.
- The Business component is connected to the Helper component through the "Extend" relationship.

- The GUI component is connected to the BookingInfo interface through the BookingPage.html interface.
- The GUI component is connected to the RegisterServlet interface through the register.html interface.
- The GUI component is connected to the LoginServlet interface through the Index.html interface.

Relationships

- The Save User component in the Persistence component has a dependency relationship with the RegisterServlet interface.
- The BookingPage.html interface in the GUI component has a dependency relationship with the BookingInfo interface in the Business component.
- The Business component extends the Helper component, indicating that it inherits functionality from the Helper component.
- The RegisterServlet interface in the Persistence component has a dependency relationship with the Save User component in the Persistence component.
- The register.html interface in the GUI component has a dependency relationship with the RegisterServlet interface in the Persistence component.
- The Index.html interface in the GUI component has a dependency relationship with the LoginServlet interface in the Persistence component.