15

CSE 6th Semester

05 May 2023

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **SUMMER SEMESTER, 2021-2022**

DURATION: 3 HOURS **FULL MARKS: 150**

CSE 4635: Web Architecture

Programmable calculators are not allowed. Do not write anything on the question paper.

Answer **all 6 (six)** questions. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

The following scenarios will be used across multiple questions in this paper.

Scenario 1: ABXY is a hospital in Bangladesh and wants to create a web application to automate different processes. The application should have three separate portals: Doctor, Patient, and Finance. In the patient portal, patients can register and set appointments with different doctors. They can also keep track of any medication they need and can check if there is any due bill. In the doctor portal, doctors can check their daily appointments and see patients accordingly. They can also record details about each appointment like the patient's symptoms, diagnosis, suggested medications, etc. The finance portal can be used by the hospital management to keep track of bill payments, the hospital's regulatory fees, etc.

Scenario 2: A food delivery company wants to create a new web application for their customers, vendors, and administrators. The web application will have the following features:

1) There will be separate modules for customers, restaurants/vendors, riders, and administrators.
2) The customer module will consist of the following sub-modules: account management, order history, address book, payment method, restaurant browsing, menu browsing, shopping cart management, and reviews.
3) Restaurant/Vendor module will consist of the following sub-modules: restaurant and menu management, order management, customer queries, customer notification management, and reviews.
4) Admin module will consist of several sub-modules, one of which is a "Site Analytics" sub-module that keeps track of website traffic, number of orders, and number of user reviews.
5) Rider module will allow riders to see placed orders in restaurants nearby and perform activities like accept and complete orders,
6) The homepage of the web application will display a list of nearby restaurants based on user location. The list of restaurants can be filtered based on different cuisine categories.
7) Whenever a user clicks on a restaurant link, details about the restaurant like name, location, menu, delivery fees, etc. will be shown to the user. In addition, users will also be shown a list of similar restaurants nearby.
8) If the same menu item is offered by multiple restaurants, users will get the option to see which restaurant offers the best pricing options and delivery times.
9) If an item is not available on the menu, users will be able to place requests for that item and get notifications whenever the item is added to the menu. The restaurant/vendor module will handle this feature.
10) The application should be able to serve a growing number of customers and restaurants/vendors.

11) Multiple data sources may be used to design the database of the application.

12) The company plans to release an application for iOS and Android devices in the future.

1. a) Discuss the major differences between HTTP request message and HTTP response message. (CO (PO

   b) i. What is a thin client and a thick client? Explain with examples. 4+8

   ii. Demonstrate the use cases of get, post, put, and delete methods with examples. (CO1) (PO1)

   c) Describe the life cycle of a servlet with appropriate diagrams. Mention the methods that are invoked during its life cycle. 9 (CO1) (PO1)

2. a) i. Explain the role of dependency injection in improving your code with appropriate examples. 5+5 (CO1) (PO1)

   ii. Explain **ServletContext** and **ServletConfig** parameters and attributes with examples.

   b) For the scenario mentioned in **Scenario 1**, assume you have a **JavaBean** called **BillBean**, and it has the following properties: **billId**, **bdtAmount**, **patientId**, **patientName**, all with necessary setter and getter methods. Also assume that you have obtained an **ArrayList** object **billList** of 20 **BillBean** objects from your database, with **billId** values ranging from 101 to 120. Now, create a **BillRecords.jsp** page (with code) to dynamically load the information (**billId**, **bdtAmount**, **patientId**, **patientName**) about the bills whose amount is greater than 50000 BDT. 8 (CO3) (PO3)

   [Use JSP standard action tags to access the **JavaBean** objects and JSTL tags to show their information inside the JSP page.]

   c) 7 (CO3) (PO3)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Log In</title>
</head>
<body>
<form method="get" action="login-servlet">
    Access Level:
    <input type="radio" name="access" value="Doctor"/> Doctor
    <input type="radio" name="access" value="Patient"/> Patient
    <input type="radio" name="access" value="Finance"/> Finance
    <br>
    Username: <input type="text" name="username-field">
    <br>
    Password: <input type="password" name="pw-field">
    <br>
    <input type="submit" value="Sign In">
</form>
</body>
</html>
```

Code Snippet 1: Code Snippet for Question 2.c)

Create a corresponding **LoginServlet** (with code) that takes the information from the login form shown in Code Snippet 1, and delegates the request to **DoctorServlet**, **PatientServlet**, or **FinanceServlet**, based on the value of the radio button.

3+12
(CO3)
(PO3)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Username Generation</title>
</head>
<body>
<form method="get" action=" ">
    User Information:
    <br>
    Name: <input type="text" id="name-field">
    <br>
    Email: <input type="text" id="email-field">
    <br>
    Phone: <input type="text" id="phone-field">
    <br>
    Occupation: <input type="text" id="occupation-field">
    <br>
    Year of birth: <input type="text" id="year-field">
    <br>
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

Code Snippet 2: Code Snippet for Question 3.a)

i. What are the different ways of creating JavaScript functions? Write their syntax.

ii. Create a JavaScript function **generateUsername()** that will be invoked whenever a user clicks on the "Submit" button of the aforementioned HTML form in Code Snippet 2. The function should prevent the default form submission behavior. The function will create a JavaScript object called **userInfo** with the following attributes: name, email, phone, occupation, year of birth, age. Except for age and username, the attributes should be taken from the above HTML form. The age attribute should be calculated using the difference between the current year and the year of birth. The username attribute should be created as a concatenation of the user's name, phone, and age. Finally, the function should return the object in an **alertbox** as a JSON object. You can use the **JSON.stringify()** method to convert the JavaScript object into a JSON object.

b) Create a RESTful endpoint (with code) using a singleton session bean to implement **feature 4** of **Scenario 2**. Your session bean should contain methods for both updating and returning the value of the analytics like website traffic and number of orders. [You need to ensure proper concurrency management and locking mechanism for the session bean.] 10 (CO3) (PO3)

4. a) Discuss the steps involved in creating an RMI application. 5 (CO1) (PO1)

b) 4+4 (CO2) (PO2)

i. Determine a suitable design pattern to implement **feature 10** of **Scenario 2**. Explain.

ii. Determine the best suited web technology to implement **feature 11** of **Scenario 2**.

c) For the application mentioned in **Scenario 2**, suppose there are currently three registered restaurants. Assume that there is a **`static list`** of three **`ArrayList`** objects that store the information about the food products offered by each corresponding restaurant. The name of the static list is **`restaurantProductList`**. Information related to each individual product are as follows: (foodId, foodName, cateogory, price, amount). Now, create a RESTful endpoint (with code) that takes a restaurant ID as query parameter and returns the list of products for that restaurant as an array list of JSON objects. For simplicity, you can assume that the restaurant IDs are 0, 1, and 2, and correspond to the first three indices of the **`static list`**.
[You only need to create the RESTful endpoint. You can assume that any other required class is already created.]

5. a) The following steps are involved when a customer places an order in the application mentioned in **Scenario 2**. 15 (CO3) (PO3)
   - Customer will choose a restaurant to order food from.
   - Customer will choose items to order from the menu of the chosen restaurant.
   - Customer will add items to the shopping cart and check out.
   - Rider will accept the customer's order and place the order in the restaurant.
   - Restaurant will accept the order and prepare the food.
   - Rider will take the food and deliver it to the customer.
   - Customer will accept the order complete payment.
   - Customer will write review.
   - Order will be marked as completed and site analytics will be updated.

   Based on the aforementioned steps, use the Façade design pattern to create the necessary sub-modules, classes, and methods (with code) to implement the system.
   [You only have to create the java classes related to the sub-modules. The methods can have empty bodies, but you should comment what the purpose of the method is.]

   b) Explain the concept of Object Relational Mapping (ORM) with examples. 7 (CO1) (PO1)

   c) Create a sample entity class (with code.) 3 (CO3) (PO3)

6. a) For **features 2-12** of the application mentioned in **Scenario 2**, identify with proper reasoning what kind of session beans should be used to implement each feature. 15 (CO3) (PO3)
   [Note: One feature may require multiple session beans depending on the requirements.]

   b) Briefly describe the different identifier generation strategies with example code. 10 (CO1) (PO1)