B.Sc. Engg. CSE 6th Semester 12 April 2022

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

SEMESTER FINAL EXAMINATION **SUMMER SEMESTER, 2020-2021**

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**. Marks of each question and corresponding CO and PO are written in the right margin with brackets.

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

Scenario 1: XYZ is a new bank in Bangladesh and its target customers are villagers in rural areas of the country. It now wants to develop a web application to handle its operations and functions. To make things easier for the rural people, they decided to allow opening of bank accounts if a customer has a phone number. XYZ has decided that it will allow multiple customers to share the same account, and at the same time it will allow a customer to open multiple accounts. XYZ also wants to introduce a feature through which customers will be able to access different information regarding their accounts (i.e., balance, transaction history, and fund transfer) from their phones. Aside from these features, the bank wants the application to be developed quickly and without too much monetary investment. They want to ensure that there is minimum latency during transactions, and they want fast development and deployment based on the requirements they gather after proper consultation. For now, XYZ wants to focus on building the application for handling their business needs in a homogenous environment.

Scenario 2: An e-commerce company has laid out the following features for their web application:

1) There will be separate modules for users, vendors, admin, and products.
2) User module will consist of the following sub-modules: user account management, items wish list, past purchases, address book, payment method, products/category browsing, and shopping cart management.
3) Vendor module will consist of the following sub-modules: Product and inventory management, customer queries, customer notification management etc.
4) Admin module will consist of several sub-modules, one of which is a "Visitor Count" sub-module that keeps track of the number of visitors visiting the website.
5) The homepage of the web application will display a list of products. The list of products can be filtered based on different item categories.
6) Whenever a user clicks on a product link, details about the product like name, price, color, vendor etc. will be shown to the users. In addition, users will also be shown a list of other items similar to the item currently being displayed.
7) If the same product is offered by multiple vendors, users will get the option to see which vendor offers the best pricing options.
8) If an item is out of stock, users will be able to place requests for that item and get notifications whenever the item is back in stock. The vendor module will handle this feature.
9) The application should be able to serve a growing number of customers and vendors.
10) Multiple data sources may be used to design the database of the application.
11) The company plans to release an application for android devices in the future.

1. a) Explain with suitable examples the role of Model, View, and Controller in an MVC pattern. 6 (CO1) (PO1)

   b) Given the applications mentioned in **Scenario 1** and **Scenario 2**, explain with proper justification which application architecture should be used to design each of these two applications. 6 (CO2) (PO2)

   c) "HTTP is a stateless protocol, but it allows stateful sessions" – Evaluate this statement. 7 (CO1) (PO1)

   d) Create a diagram to show the several stages of the lifecycle of JSP. In this diagram, show in which steps the three lifecycle methods of JSP are invoked. 6 (CO3) (PO3)

2. a) What are the major differences between `<%@include%>` and `<jsp:include>`? 3 (CO1) (PO1)

   b) Consider the following figure and answer the subsequent question: 15 (CO3) (PO3)

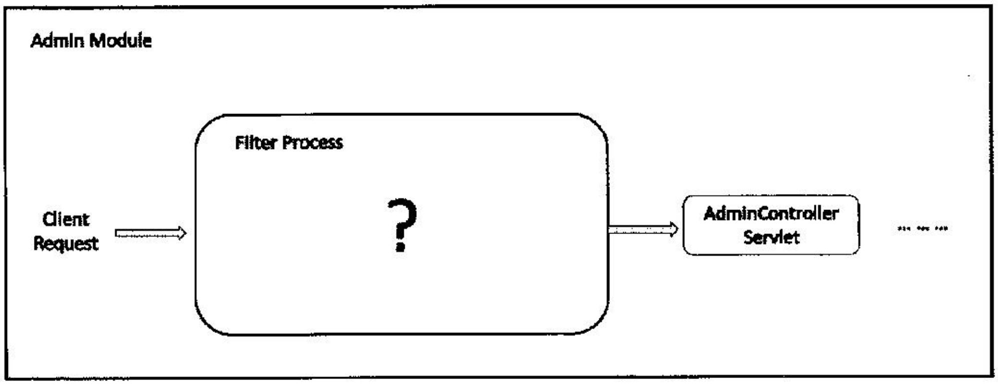

Figure 1: Figure for Question 2 (b)

Figure 1 shows part of the admin module mentioned in **Scenario 2**.
Create the necessary classes, pages etc. (all with code) required to complete the "Filter Process" shown in the figure. If the client is authenticated with the username and password values as "admin" and "adminpw" respectively, the client request should be forwarded to a servlet called **`AdminControllerServlet`**. Otherwise, the client request will be forwarded to a page called **`AdminLogin.jsp`**.

   c) For the application mentioned in **Scenario 1**, assume you have a **`JavaBean`** called **`TransactionBean`**, and it has the following variables: *transId*, *transDate*, *customerId*, *transAmount*, all with necessary setter and getter methods. Also assume that you have obtained a list of 15 such transactions from your database, with *transId* values ranging from 2101 to 2115. Now, create a **`CustomerTransactions.jsp`** page (with code) to dynamically load the information (*transId*, *transDate*, *customerId*, *transAmount*) about the first 10 transactions. 7 (CO3) (PO3)
Note that You cannot hard-code the information about the transactions. Avoid using scriptlets inside the **`CustomerTransactions.jsp`** page.

3. a) Explain the dynamic typing nature of JavaScript with example. 4+3 (CO1) (PO1)
What will be the boolean values of the following conditional expressions in JavvaScript?
   i. `(10=="10")`
   ii. `(20!=="20")`
   iii. `(100!="100")`

b) Assume you are using “Front Controller” design pattern to handle client requests for the user module (feature number 2) mentioned in **Scenario 2**. Using the Servlet Front strategy, create a class (with code) that implements this pattern. 8 (CO3) (PO3)

For each of the sub-modules, you can assume that there are separate JSP pages to which you can dispatch the client requests. You do not have to create the JSP pages.

c) Consider the following code snippet for an HTML form and answer the subsequent question: 10 (CO3) (PO3)

```
<form id=”form1” onsubmit=”getFormValue()”>
    First Name: <input type=”text” id=”fname-input” placeholder=”First Name”>
    <br>
    Last Name: <input type=”text” id=”lname-input” placeholder=”Last Name”>
    <br>
    <input type=”submit” class=”submit-btn” value=”Submit”>
</form>
```

Create a JavaScript function **`getFormValue()`** that will be invoked whenever a user clicks on the “Submit” button of the aforementioned HTML form. The function will use a *prompt* to take the year of birth as an additional input from the user. It will then calculate the age of the user based on the entered year of birth and print the following message in an *alert box*:

“Welcome, {First Name} {Last Name}! Your age is {Age}.”

The values of {First Name} and {Last Name} should be obtained from the HTML form, and the value of {Age} should be calculated using the year of birth of the user.

4. a) Define marshaling and unmarshaling; Which classes are responsible for performing the marshaling and unmarshaling processes in RMI? 5 (CO1) (PO1)

b) “The concept of RMI focuses on the fact that local processes can operate on objects stored in remote nodes as if they were operating on objects stored in local nodes.” – Evaluate this statement. 5 (CO1) (PO1)

c) Read the following information about the application mentioned in **Scenario 1**: 3 (CO2) (PO2)

“XYZ also wants to introduce a feature through which customers will be able to access different information regarding their accounts (i.e., balance, transaction history, and fund transfer) from their phones.”

What technology should be used to fulfil this requirement? Justify your answer.

d) For the application mentioned in **Scenario 2**, suppose there are currently three registered vendors. Assume that there is a **`static list`** of three **`ArrayList`** objects that store the information about the products in their inventory for each corresponding vendor. The name of the **`static list`** is **`VendorProductList`**. Information related to each individual product are as follows: prodId, prodName, quantity, price, manufacturerName. 12 (CO3) (PO3)

Now, create a RESTful endpoint (with code) that takes a vendor ID as input and returns the list of products for that vendor as an array list of JSON objects. For simplicity, you can assume that the vendor 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) For the features marked 2-11 for 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 that one feature may require multiple session beans depending on the requirements.

b) Create a session bean (with code) that will handle the “Visitor Count” sub-module mentioned in feature number 4 of the application mentioned in **Scenario 2**. Your session bean should contain two methods: One for updating the visitor count, and the other for showing the current visitor count. 10 (CO3) (PO3)

6. a) Assume you are currently working with the following two entities for the database of the application mentioned in **Scenario 1**: 10 (CO3) (PO3)

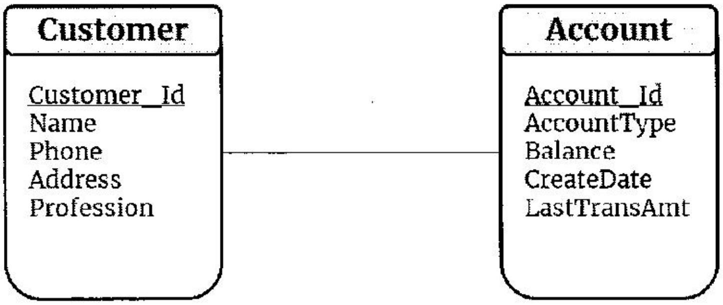

Figure 2: Entities mentioned in Question 6 (a)

Here, *Customer_ID* and *Account_ID* are primary keys that need to be generated using the "Identity" generation scheme.

Assuming bidirectional association, create entity classes (with code) for these two tables, maintaining the proper relationship between them.

b) Consider the content of the **persistence.xml** file shown below and answer the subsequent question: 5 (CO3) (PO3)

```
<persistence xmlns="http://xmlns.jcp.org/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence
            http://xmlns.jcp.org/xml/ns/persistence/persistence_2_2.xsd"
             version="2.2">
<persistence-unit name="my_pu" transaction-type="JTA">
<provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
<properties>
<property name="javax.persistence.jdbc.url"
value="jdbc:h2:tcp://localhost/~/test"/>
<property name="javax.persistence.jdbc.driver" value="org.h2.Driver"/>
<property name="javax.persistence.jdbc.user" value="sa"/>
<property name="javax.persistence.jdbc.password" value=""/>
</properties>
</persistence-unit>
</persistence>
```

Create an **EntityManager** using the persistence unit defined in the aforementioned **persistence.xml** file.

c) Assume that the database module of the application mentioned in **Scenario 1** has three entities: *Customers*, *Accounts*, *Transactions*. Now, create this module using the Data Access Object (DAO) design pattern. 10 (CO3) (PO3)

You do not have to write the code for the classes. However, you must briefly describe the purpose of each class that you use to develop this module using Data Access Object design pattern.