Uva Wellassa University of Sri Lanka
Faculty of Science and Technology
Department of Computer Science and Technology
200 level 2nd Semester Examination – Dec./Jan. 2017
CST226-3 Rapid Application Development

Uva Wellassa University

Instructions to candidates
Duration: Two (02) hours
Number of questions: Three (03)
Mark allocation: 100
Answer all the questions
Create a separate project/solution for each question.
You need to upload the project/solution folder as a ZIP file to the CMS.
When you are uploading, rename the ZIP file with your "Index Number" as shown in the example (i.e. UWU_EX_14_XXXX).

---

1. The Co-operative Rural Bank has requested to build an application to automate their daily banking processes. The development team decided to build a Standalone Java Application to cater this requirement. The team is following the Scrum methodology and they agreed to release "Account Holder's Registration" functionality during their first sprint.

   The "Account Holder's Registration" interface has five (05) fields, namely, Title, First Name, Last Name, NIC Number and Residential Address. To register for an account, applicant has to provide at least First Name and NIC Number as required details into the interface. The interface should be as Figure 01.

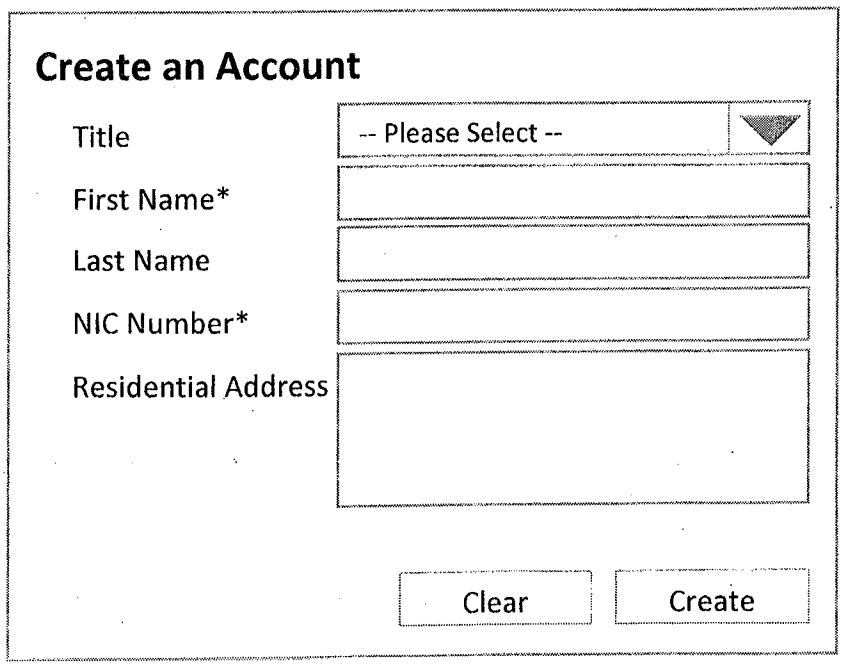

Figure 01: Account Holder's Registration Interface

   a. Design the interface according to Figure 01.

b. Implement the functionality of "Create" button to save these field values to the given database. (You should check required fields before saving these values)

Note:

- Use "com.mysql.jdbc.Driver" as the driver name for database connectivity.
- Use "getSelectedItem()" method to get the selected value from a combo box and "showMessageDialog()" method of "JOptionPane" class to show required field validation messages.
- Create a database schema as "ruralbank" and use the following query to create the "accounts" table inside the database to store these relevant information.

```
CREATE TABLE accounts
(
        Id INT NOT NULL AUTO_INCREMENT,
        Title VARCHAR(10),
        FirstName VARCHAR(255) NOT NULL,
        LastName VARCHAR(255),
        NIC VARCHAR(255) NOT NULL,
        Address VARCHAR(255),
        PRIMARY KEY (Id)

);
```

(35 mark)

2. The Uva Wellassa University is requested to build a Standalone Application to facilitate their distance learners to access university resources. The development team decided to build a Windows Forms application using Visual C# to cater this requirement.

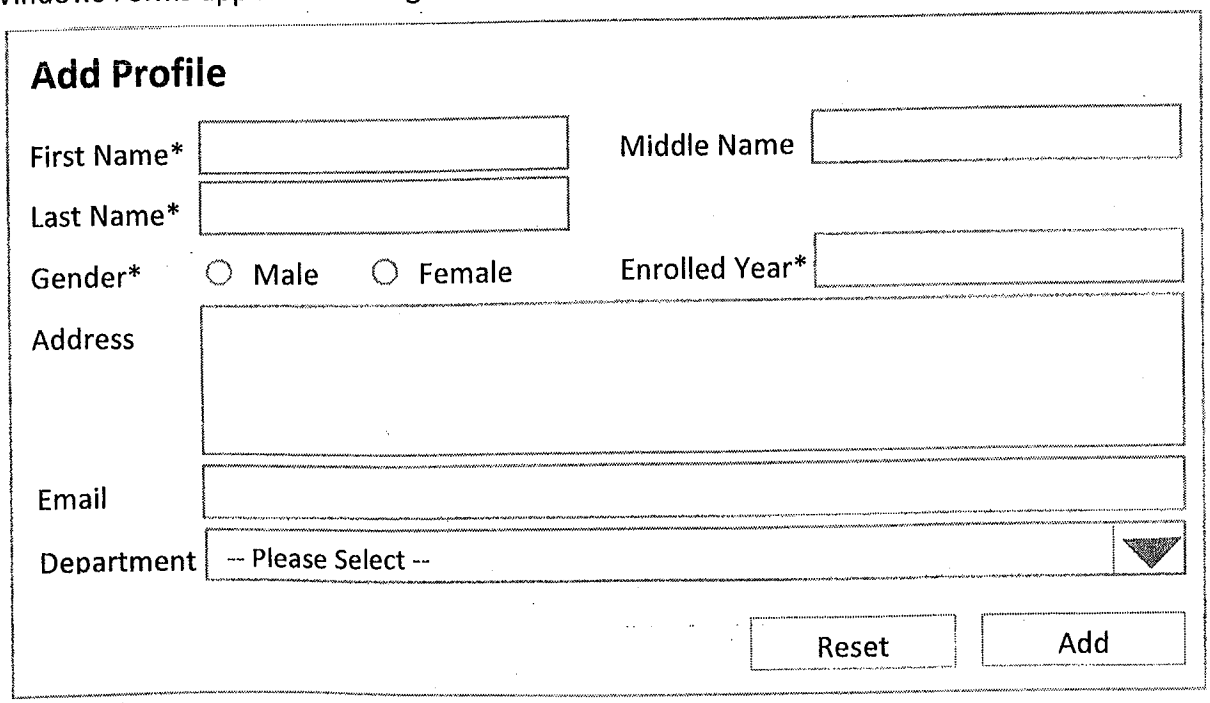

Figure 02: Add Profile Interface

The team is following the Scrum methodology and agreed to release several backlog items during the first sprint. Suppose you are in the development team and responsible to develop "Add Profile" functionality.

In the interface, First Name, Middle Name, Last Name, Enrolled Year and Email are textboxes, Male and Female are radio buttons and Department is a combo box. The Department combo box should have "Computer Science and Technology" and "Science and Technology" as values for the selection options other than the "--Please Select--" option.

a. Design the interface according to Figure 02.
b. Implement the functionality of "Add" button to save these field values to the given database.

Note:

- Use a Windows "MessageBox" to show the success and error messages.
- To get the selected value from a combo box, you can use "SelectedItem" property (i.e. comboBox1.SelectedItem.ToString();).
- Create a database schema as "external" in SQL Server and use following query to create the "profile" table inside the database to store these relevant information.

```
CREATE TABLE profile
(
        Id INT IDENTITY(1,1) PRIMARY KEY,
        FirstName VARCHAR(255) NOT NULL,
        MiddleName VARCHAR(255),
        LastName VARCHAR(255) NOT NULL,
        Gender VARCHAR(6) NOT NULL,
        EnrolledYear VARCHAR(4) NOT NULL,
        Address VARCHAR(255),
        Email VARCHAR(255),
        Department VARCHAR(255)
);
```

(40 mark)

3. Following is a calculator which can be used to calculate a person's Body Mass Index (BMI).

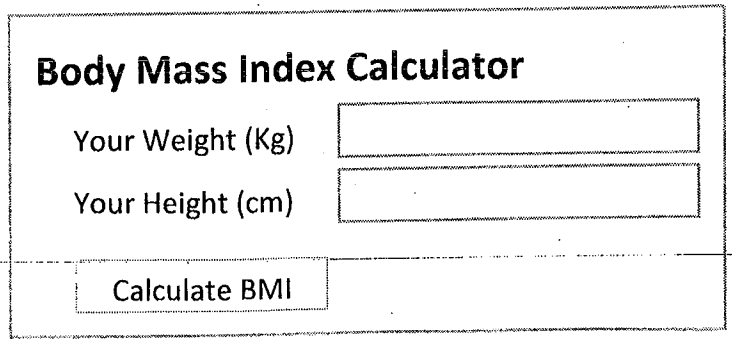

Figure 03: index.html Page

Implement simple HTML form (index.html) in Figure 03, where user can input weight (in Kilograms) and height (in centimeters). After user clicks on the "Calculate BMI" button, Java Servlet page (bmi.java) is called from the form's action, which will execute the calculation process and generate the final results. Your Servlet output should be viewed on the browser as shown in the Figure 04.

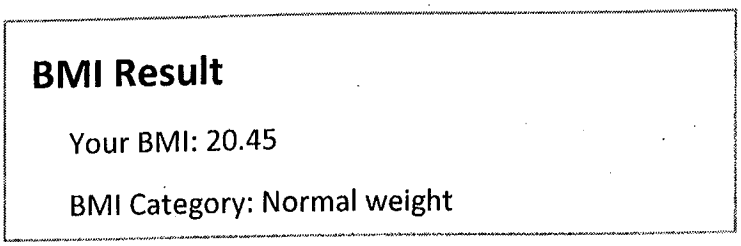

Figure 04: Servlet Page Output

Note:

- When you are displaying results in "bmi.java" page, results should be based on the user entered values.
- Use "Double.parseDouble (<string>)" to convert string value into a floating point value.
- Following conditions can be used to display the correct BMI category.
    - Underweight → <18.5
    - Normal weight → 18.5–24.9
    - Overweight → 25–29.9
    - Obesity → BMI of 30 or greater
- Use the following formula to calculate the BMI with the precision of two (02) decimal points.

    Math.round((weight / ((height/100) * (height/100))) * 100.0)/100.0;

(25 mark)