Uva Wellassa University of Sri Lanka
Faculty of Science and Technology
Department of Computer Science and Technology
200 Level 2nd Semester Examination – Jan. / Feb. 2016
CST 224-3 Rapid Application Development

Uva Wellassa University

Instructions to candidates

Part B

Duration: 02 hours
Number of questions: Three (03)
Answer all the questions
Mark allocation: 70
Create a **separate project/solution** for each question.
CMS will have **three separate links** for the questions and you need to upload the project/solution folder as a **ZIP file** to the relevant link.
When you are uploading, rename the **ZIP files** with your **Examination Number** and the **Question Number** as shown in the example (i.e. UWU_EX_13_0001_Q1).
You are allowed to refer your own notes but **sharing notes** is **strictly prohibited.**

---

1. Administrator of Virtual Learning Environment (VLE) needs to generate **random passwords** for every Computer Science and Technology (CST) student for 2013/2014 batch. Assume, there are only **fifty (50)** students in this batch.

   You are required to create a **Java Server Page (JSP)** to show a table with **two (02)** columns and **fifty one (51)** rows (including header row) in order to show student **registration number** and the generated **password**. Consider the following conditions when you are creating the page.

   a. The password should consists of **five (05) letters and/or numbers**.
   b. The web page should **regenerate** passwords when user refresh the page.

   Your final web page should be viewed on the browser as shown in the following format.

Registration Number	Password
UWU/CST/13/0001	JmviW
UWU/CST/13/0002	W4OBJ
UWU/CST/13/0003	w01VE
UWU/CST/13/0004	pnwhJ

   Note:

   - Use the following code segment to generate random text.

```
String text = "";
String possible =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";
for( int i = 0; i < 5; i++ )
    text += possible.charAt( (int)Math.floor( Math.random() * possible.length() ) );
```

(20 mark)

08

2. There is a need for a system to register students for a seminar at Uva Wellassa University. **Only CST and IIT students** can register for this seminar. The available number of seats for the seminar is **fifteen (15)**.

   Write a **Windows Forms** application using **Visual C#** to satisfy above requirement. Your application should have the followings.

   a. When you run the application, first it should ask **student registration number** and the **name** of the student.
   b. Then system should display a message to the student indicating whether the registration is successful. The message should be one of the following formats.
      - For a successful registration:

        Thank you <student's name>. Your seat number is <seat_no>
      - For an unsuccessful registration:

        Sorry <student's name>, all the seats are reserved.
   c. Create a **Service-based** database item called **Seminar2016** and store all registered student details in a table called **StudentDetails** inside the database.

Note:

- The first student who registers for the seminar will get seat number 1; the second student should get the seat number 2 and so on until all the seats are occupied.
- Use a **Windows MessageBox** to show the relevant messages.
- Use **<string>.Split('<separator>')** method to split a string and **Int32.Parse(<string>)** to convert string value into an integer value.
- The following query can be used to create the **StudentDetails** table.

```
CREATE TABLE StudentDetails
(
        RegistrationNo VARCHAR(255) NOT NULL,
        Name VARCHAR(255) NOT NULL,
        SeatNo INT NOT NULL,
        PRIMARY KEY (RegistrationNo)

);
```

(25 mark)

3. 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 these requirements. The team is following the Scrum methodology and they agreed to release **Account Holder Registration** functionality during their first sprint.

   The **Account Holder 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.

The team decides to use Object Relational Mapping (ORM) method when developing the application and all agreed to go with **Hibernate framework** during the planning meeting. Suppose you are assigned to implement above functionality.

a. **Design the interface** accourding to Figure 01.
b. **Implement an action listener** to **save** these **field values** to the given database **using Hibernate framework**. (You should check required fields before saving these values)

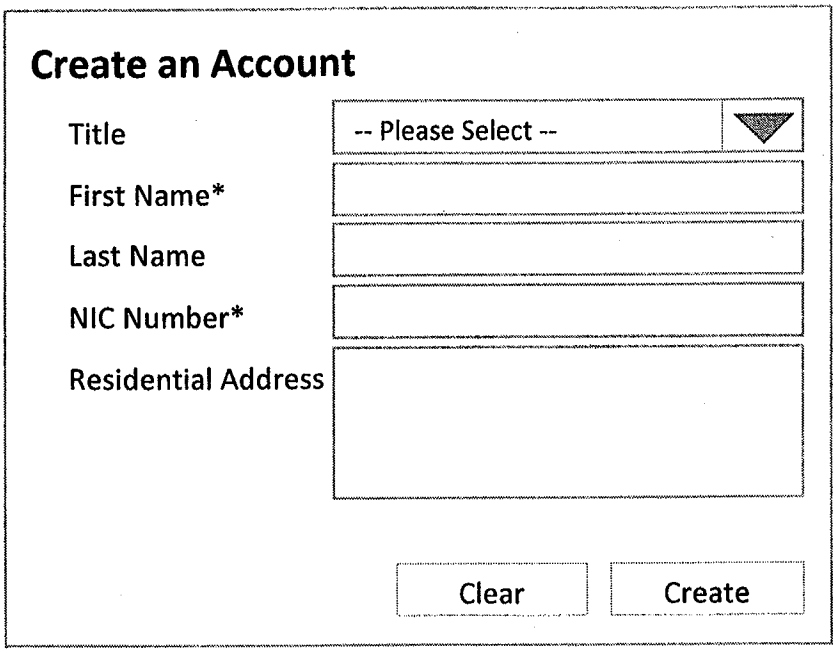

Figure 01: Account Holder Registration Interface

Note:

- Use **showMessageDialog** method of **JOptionPane** class to show required field validation messages.
- Create a database schema as **ruralbank** and use 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)

);
```

(25 mark)

08