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
CST224-3 Rapid Application Development

Instructions to candidates
Duration: Three (03) 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 folders 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_13_XXXX).
You are allowed to refer your own notes but sharing notes is strictly prohibited.

---

1. The ABC Company is planning to implement a **Visual C# application** for the following functionalities. The application allows user to calculate salary of an employee and store data to a database.

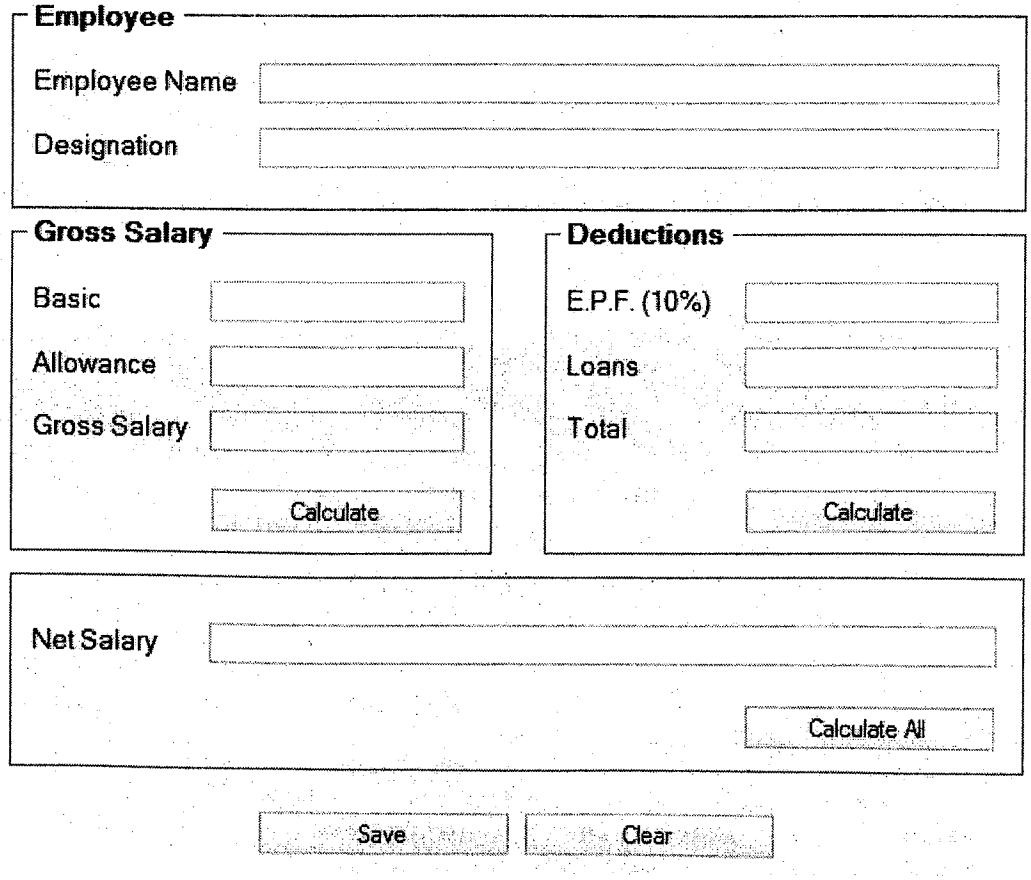

Figure 01: Calculate Salary Interface

The functionalities of the GUI are as follows,

- "Calculate" button under "Gross Salary" section should calculate the gross salary and display the amount in front of "Gross Salary" label.

- Use the following function for the gross salary calculation.

  Gross Salary = Basic Salary + Allowance
- "Calculate" button under "Deductions" section should calculate the total deduction and display the amount in front of "Total" label.
- Use the following functions for the total deduction calculation.

  EPF = Basic Salary * 10 / 100

  Total Deductions = EPF + Loans
- "Calculate All" button should calculate the net salary and show the amount in front of "Net Salary" label.
- Once user clicks on the "Calculate All" button, it should perform both gross salary calculation and deductions calculation and show the values in front of the relevant label.
- Use following functions for the net salary calculation.

  Net Salary = Gross Salary – Total Deductions
- "Clear" button should clear all the text fields on the GUI with calculated values.

You are required to,

a. Create a **Visual C# application** according to above specifications.
b. Create a Service-based database item called "EmpData" and store all relevant salary details in a table called "SalaryInfo" using the following query.

```
CREATE TABLE SalaryInfo
(
salaryID INT IDENTITY(1,1) PRIMARY KEY,
empName varchar(255) NOT NULL,
basicSalary decimal(10,2) NOT NULL,
allowance decimal(10,2) NOT NULL,
loan decimal(10,2) NOT NULL
);
```

c. Write relevant codes for the "Save" button to get the relevant data from the GUI and store to the database.

Note:

- Use a Windows "MessageBox" to show the success and error messages and "Double.Parse(<string>);" to convert string value into a floating point value.

(30 mark)

2. Administrator of the Course Management System (CMS) 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 Figure 02.

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

Figure 02: Final Output

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)

3. Implement the following **Standalone Java Application** which can be used as a Point of Sales system in E & Y Company.

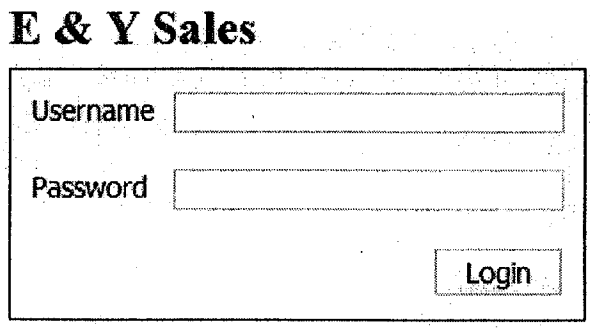

Figure 03: Login Interface

In the "Login" interface (Figure 03), "username" should be a text field and "password" should be a password field. When user clicks on "Login" button in the interface, system should check whether the entered username and password is matched with the database. If it is matched, then the application should direct user to the "Main Menu" interface (Figure 04). If not, application should show an error message to the user (i.e. Invalid login, try with correct username and password).

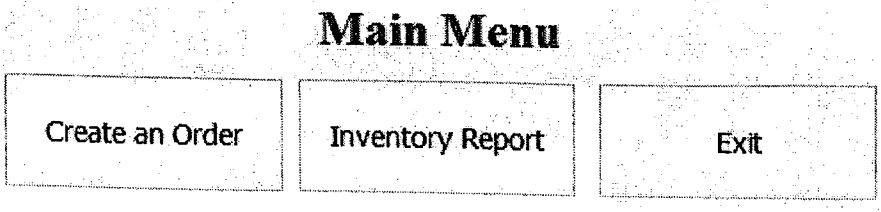

Figure 04: Main Menu Interface

In the "Main Menu" interface, "Create an Order", "Inventory Report" and "Exit" are buttons. When user clicks on "Create an Order" button, "Main Menu" interface should be disappeared and "New Order" interface (Figure 05) should be visible to the user.

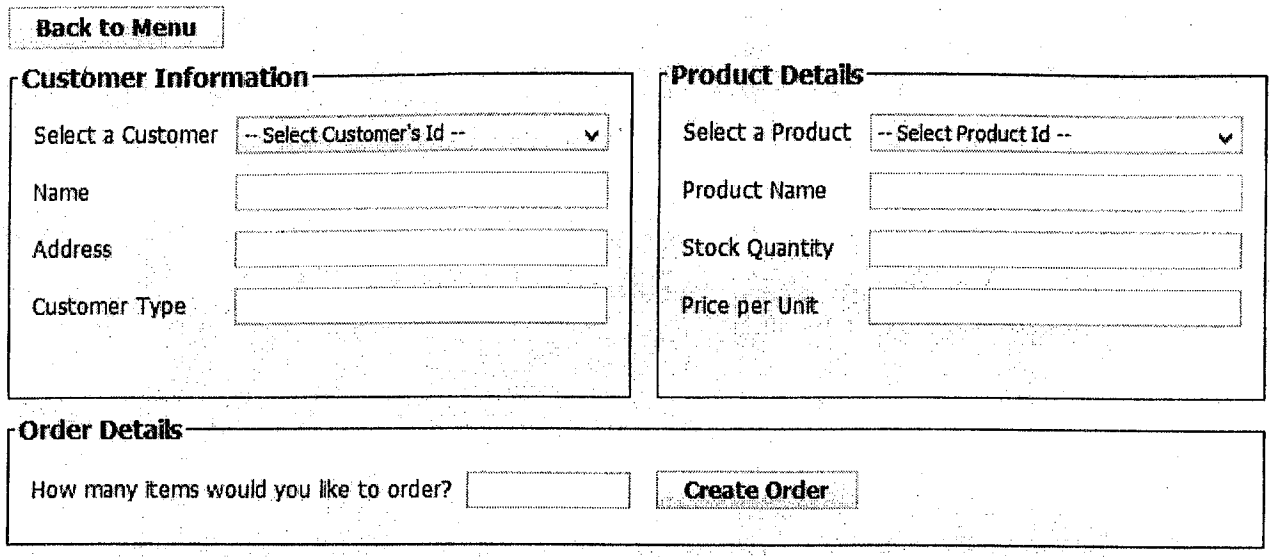

Figure 05: New Order Interface

In the "New Order" interface, "Back to Menu" is a button and by clicking it, user can disappear current window and open the "Main Menu" interface again and "Customer Information", "Product Details" and "Order Details" are different sections which can be implemented using Panel Swing Container.

The functionalities of those sections are as follows,

- "Customer Information" panel:
    - In this panel, "Select a Customer" is a combo box and values for that combo box should be loaded from the given database. Here, it will be the "CustomerId" in "Customer" table.
    - "Name", "Address" and "Customer Type" are read-only text fields.
    - After user selects a customer id, the text field values should be loaded automatically from the database.
- "Product Details" panel:
    - In this panel, "Select a Product" is a combo box and values for that combo box should be loaded from the given database. Here, it will be the "ProductId" in "Product" table.
    - "Product Name", "Stock Quantity" and "Price per Unit" are read-only text fields and after user selects a product id, the text field values should be loaded automatically from the database.
- "Order Details" panel:
    - User can enter the quantity in to the text field available in this panel and click on the "Create Order" button.

- If the user entered quantity exceeds the quantity in the stock, the application should show a message like "Sorry, the quantity ordered exceeds our inventory" (*see* Figure 06).
- Else, application should show a message like, "Your order is processed successfully" to the user (*see* Figure 07) and store order details to the "order" table in the given database (You can download the database table creation query file from the CMS).

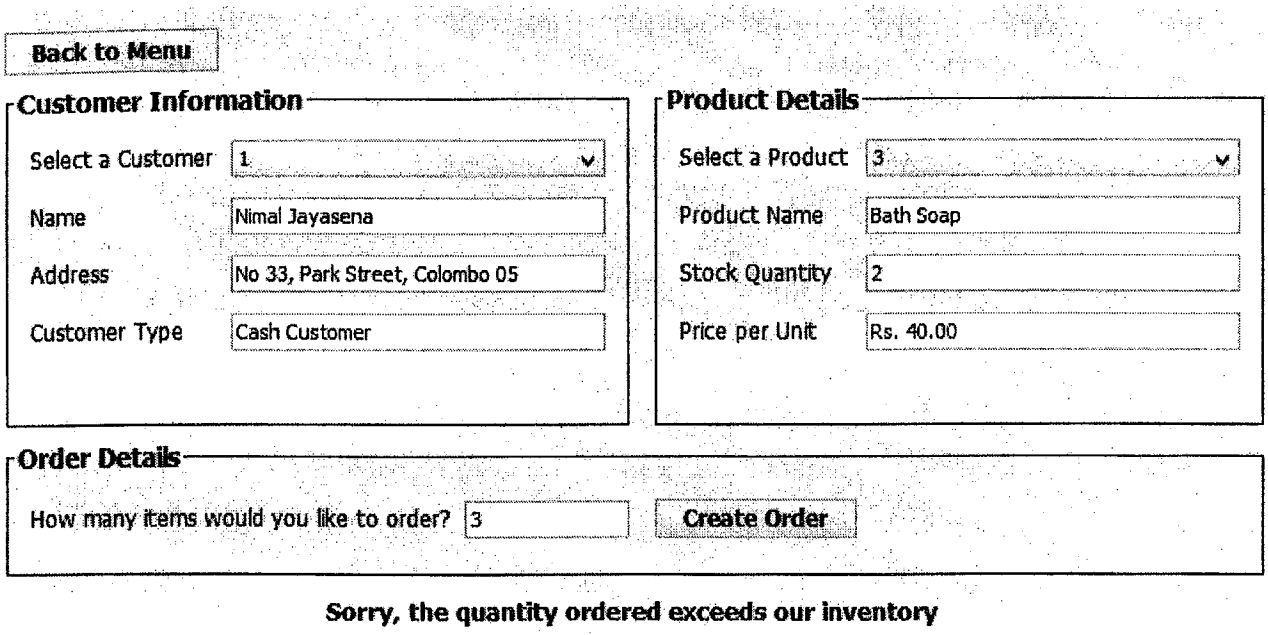

Figure 06: New Order Interface with an Error Message

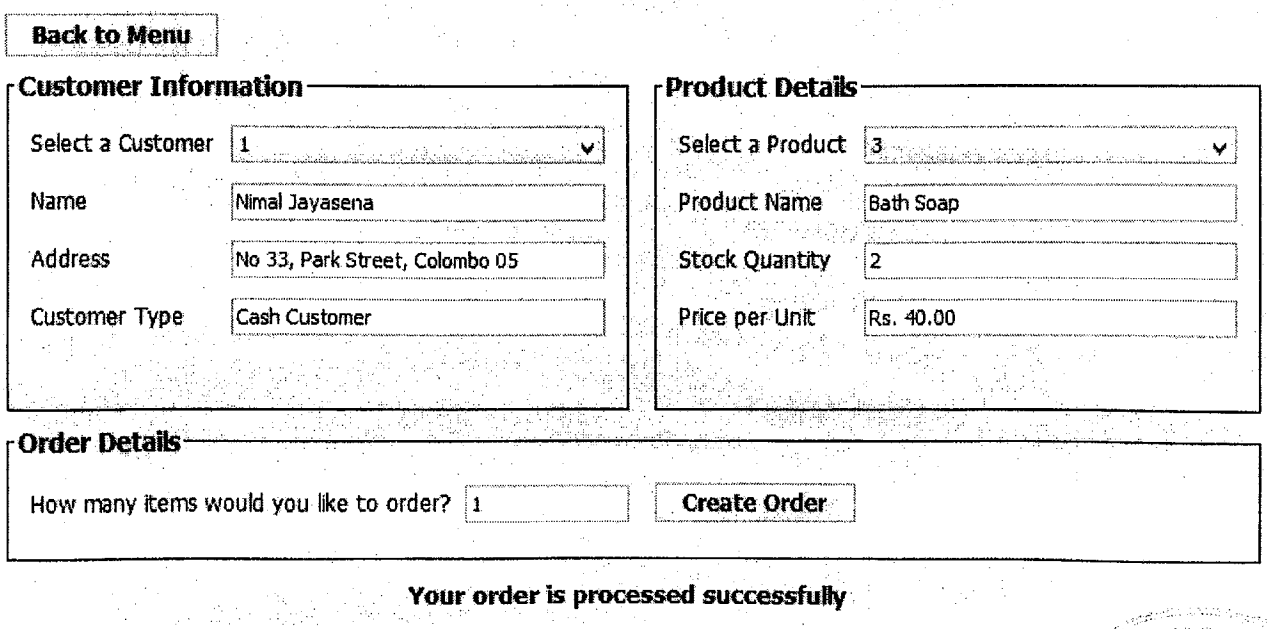

Figure 07: New Order Interface with a Success Message

You are required to,

a. Create a **Java application** according to above specifications.
b. Create a database schema as "EYSales" and **use the file available in the CMS** to create the necessary tables inside the database.
c. Write relevant codes for the "Create Order" button to get the relevant data from the GUI and store to the "order" table in the "EYSales" database.

Note:

- Use "addItem()"method to add database values as items to the relevant combo box (i.e. jCombobox1.addItem();).
- Use "getSelectedItem()" method to get the selected value from a combo box (i.e. jCombobox1.getSelectedItem().toSting();).
- Use "showMessageDialog()" method of "JOptionPane" class to show messages.
- Use "Double. parseDouble(<string>);" to convert string value into a floating point value and "Integer. parseInt(<string>);" to convert string value into a integer value.

(50 mark)