UMS
UNIVERSITI MALAYSIA SABAH

INDIVIDUAL PROJECT 2

OBJECT ORIENTED PROGRAMMING

SEMESTER 2 2019/2020

DUEDATE:	**06/08/2020**

TITLE : **FKI REGISTRATION SYSTEM**

SECTION : **1**

SUBJECT CODE : **KK14203**

SUBJECT : **OBJECT ORIENTED PROGRAMMING**

LECTURER : **DR. SAMRY @ MOHD SHAMRIE SAININ**

NO.	STUDENT NAME	MATRIC NO.
1.	NURUL ASYIKIN BINTI AMAT	BI19110094

1. INTRODUCTION

The main idea of this project is to create software that helps FKI students to register their course, residential, and club. The software is called FKI Registration System. In real life application, FKI Registration System can be very handy as it can register in 3 types which I had mention above. This make things easier for FKI students, cause it will be less to encounter website error. In addition, students for every batch are not that many. This system can store the student's name, matric no, year, semester-session, no phone, club, and programme for future reference to FKI lecturers. It is a simple user interface which means user will not face any complexities whenever they are using it.

2. OBJECT ORIENTED PROGRAMMING CONCEPTS

a. Class and Object

Class is a set of instructions that is used to build a specific type of object while object is an instance of class. For example:

```
class LoginPanel extends JFrame { // this is a class
. . . . . .
        public static void main(String[] args){
                LoginPanel frame = new LoginPanel(); // this is an object
        }
}
```

b. Inheritance

Inheritance refers to the ability of a class to inherit from another class. In this coding, there is an "extends" keyword which means the class will inherit the properties of the java swing. For example:

```
class LoginPanel extends JFrame {
} or
class MainMenu extends JFrame{
} or
class CourseRegistration extends JFrame{
} or
```

```
class ResidentialRegistration extends JFrame{
} or
class ClubRegistration extends JFrame{
}
```

c. Polymorphism

Polymorphism allows us to perform a single action in different ways. There is a Jbutton Save that is used to save the details into a text file. In addition, it will be implemented in action listener for it to work. When we click the Jbutton, the showMessage will pop out. For example:

```
JButton btnSave = new JButton("Save");
        btnSave.addActionListener(new ActionListener(){
        . . . .
        BufferedWriter bfw = new BufferedWriter(new FileWriter("course
        data.txt"));
        . . . .
        JOptionPane.showMessageDialog(null, "Success");
)}
```

d. Composition

Composition is a design technique to implement has-a relationship in classes. In this case, we can say that JFrame has a relationship with JButton because JButton cannot exist without JFrame. For example:

```
public class MainMenu extends Jframe{
. . . . .
        public MainMenu(){
        JButton btnCourse = new JButton("Course");
        add(btnCourse);
        }
}
```

e. Encapsulation

Encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. Class' variables will be hidden from other classes, and it can be accessed through the methods of the current class. For example:

```
public class ResidentialRegistration extends JFrame {

        private JPanel contentPane;
        private JTextField textField_name;
        private JTextField textField_matric;
        private JTextField textField_date;
        private JTable table;
        private String output = "";
    public ResidentialRegistration() {
        . . . .
        setContentPane(contentPane);
        . . . .
        textField_name.setBounds(162, 42, 207, 22);
        textField_matric.setBounds(162, 72, 207, 22);
        textField_date.setBounds(162, 159, 207, 22);
        table.setModel(new DefaultTableModel(
        );
    }
}
```

3. READ AND WRITE IMPLEMENTATION

a. Read

This code will read the data first before going to save it at file name residential data.txt.

```
String filepath = "residential data.txt";
File file = new File(filepath);
BufferedReader reader = null;
  try{
      FileInputStream fis = new FileInputStream(file);
      reader = new BufferedReader(new FileReader(file));
      String line;
```

```
        DefaultTableModel model = (DefaultTableModel)
    table.getModel();

     while((line = reader.readLine()) != null) {
              model.addRow(line.split(","));
     }
         reader.close();
    } catch (Exception ex){
           ex.printStackTrace();
    }
```

b. Write

This code will read a file named course data.txt and the data will be shown on the table so that user can see the data

```
try{
BufferedWriter bfw = new BufferedWriter(new FileWriter("course
data.txt"));
                    for (int i = 0 ; i < table.getRowCount(); i++){
          for(int j = 0 ; j < table.getColumnCount();j++){
              bfw.write((String)(table.getValueAt(i,j)));
              bfw.write(",");;
          }
     bfw.newLine();
   }
     bfw.close();
          JOptionPane.showMessageDialog(null, "Success");
catch(Exception e){
   e.printStackTrace();
                JOptionPane.showMessageDialog(null,"Error");
}
```

4. SOURCE CODE (Please open this code using eclipse for it to work)

https://github.com/nurulasyikinamat/KK14203-/tree/master/Project%202

5. USER MANUAL GUIDE

1. Login Form

i. Firstly, user has to login before going into the system.

ii. At the login form, user can enter the name and password.

iii. Make sure to fill up both of it, if not error message will pop out.

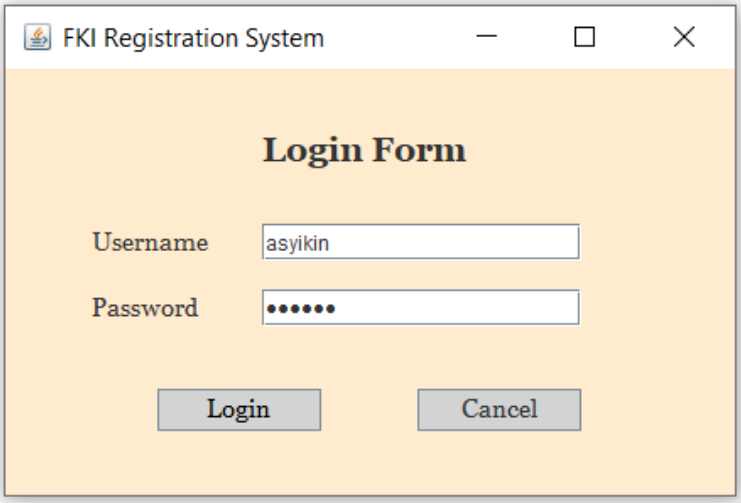

Figure 1: Login Form

2. Main Menu

i. At the main menu, user can see 3 option which are Course, Residential and Club.

ii. User can only choose one option at a time to continue.

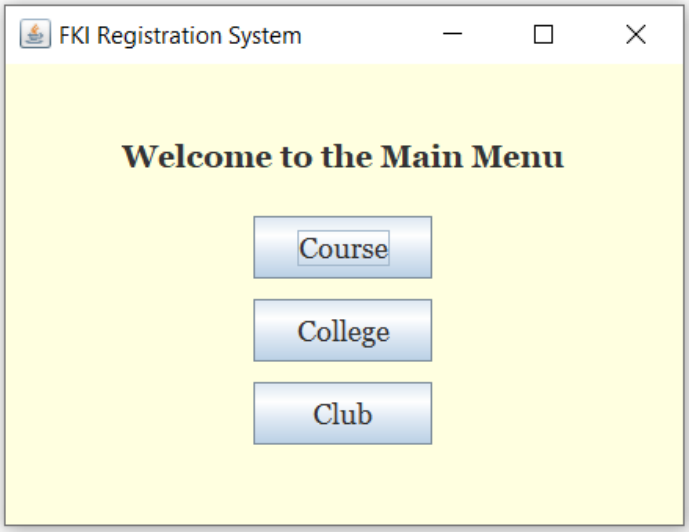

Figure 2: Main Menu

3. Course Registration Form

i. User can enter full name, matric no, choose the programme, semester-session, year, and course.

ii. Make sure to fill up everything, if not error message will pop out.

iii. User can insert the whole detail into the table by clicking the add button or clear the whole detail by clicking the clear button.

iv. User also can save the detail into text file by clicking the save button or delete the detail in the table by selecting the row to be deleted.

v. The detail also can be printed by clicking the print button and if user already fill everything, they can either click the back button to register residential and club or click the exit button to exit the system.

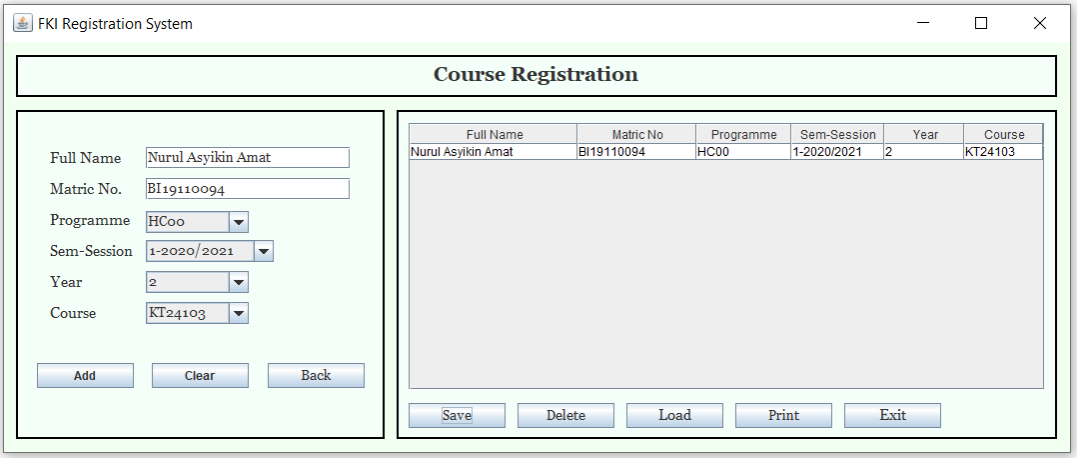

Figure 3: Course Registration Form

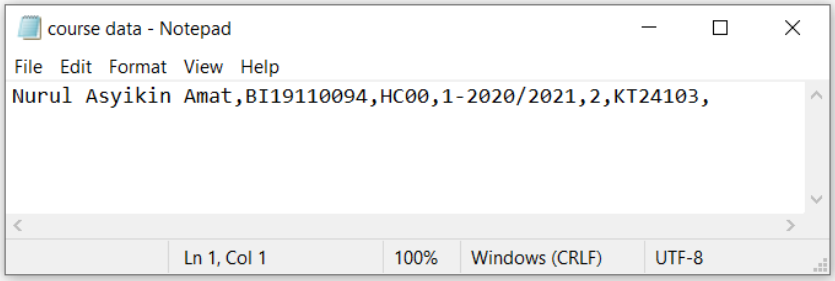

Figure 4: Course data is saved in text file.

4. **Residential Registration Form**

i. User can enter full name, matric no, choose semester-session, year, and enter the date.

ii. Make sure to fill up everything, if not error message will pop out.

iii. User can insert the whole detail into the table by clicking the add button or clear the whole detail by clicking the clear button.

iv. User also can save the detail into text file by clicking the save button or delete the detail in the table by selecting the row to be deleted.

v. The detail also can be printed by clicking the print button and if user already fill everything, they can either click the back button to register course and club or click the exit button to exit the system.

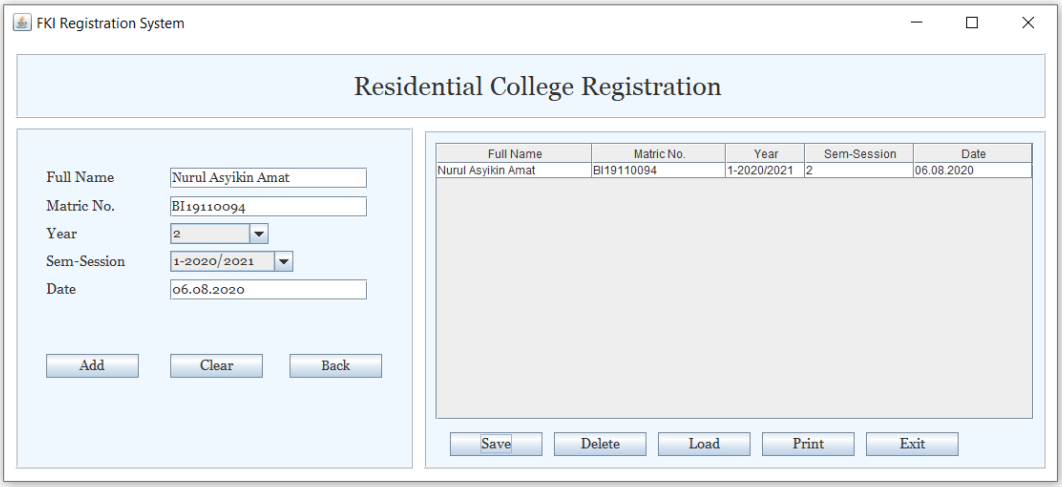

Figure 6: Residential College Registration

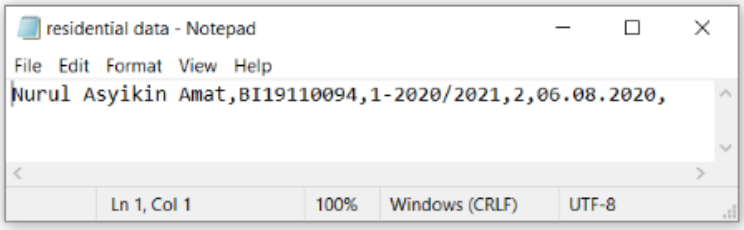

Figure 7: Residential data is saved at text file.

5. **Residential Registration Form**

i. User can enter full name, matric no, phone no, choose faculty, semester-session, year, and club

ii. Make sure to fill up everything, if not error message will pop out.

iii. User can insert the whole detail into the table by clicking the add button or clear the whole detail by clicking the clear button.

iv. User also can save the detail into text file by clicking the save button or delete the detail in the table by selecting the row to be deleted.

v. The detail also can be printed by clicking the print button and if user already fill everything, they can either click the back button to register course and residential or click the exit button to exit the system.

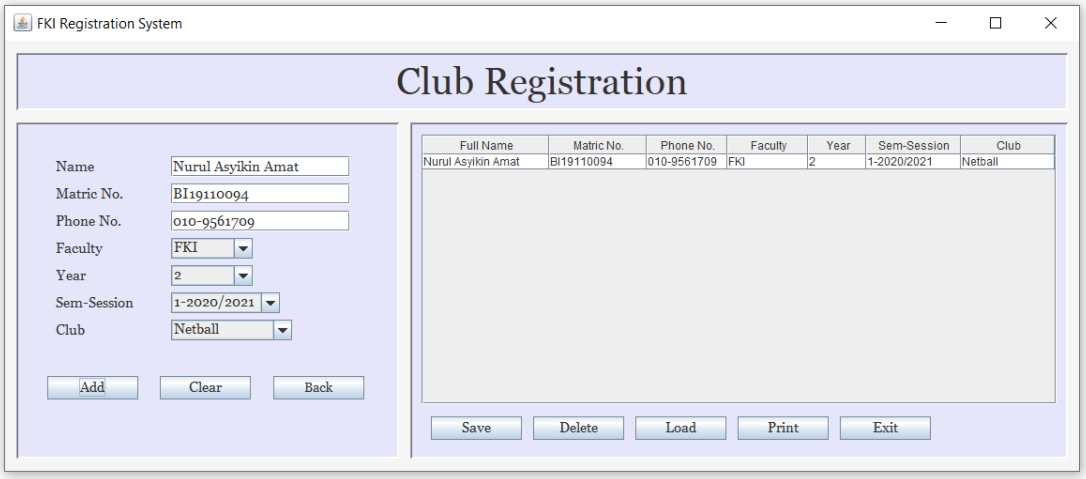

Figure 8: Club Registration Form

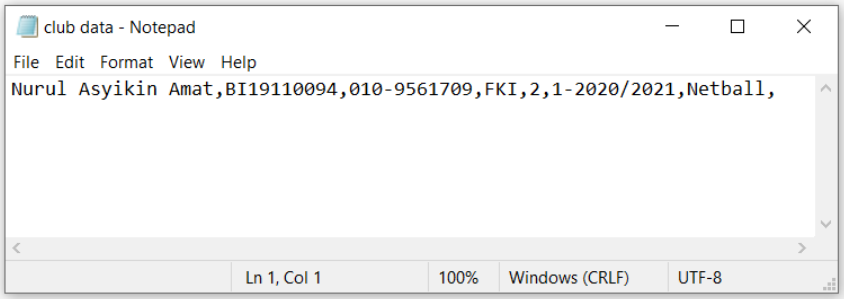

Figure 9: Club data is saved in text file.