**GUI –STUDENT ASSIGNMENT DUE – MONDAY 14 OCTOBER**

**Question**

Create an application that will help VUT to register their Students for the course and also be able to enrol for all subjects available for a specific course. Student can enrol for more than 1 subjects of the same course. The application must allow them to also search for a Student, it must also allow a student to cancel the subject as well as retrieving all students’ records per subject enrolled for or retrieve all students they have in the system.

**Note the following is provided in the folder for you:**

* Package: vut.gui – template classes for GUI Screens,
* Package: vut.data- Student.java Class, StudentDA.java and various exception classes,
* Files: courses.txt and subjects.txt

Implement a **three-tier** design with Problem Domain (PD), Data Access (DA) and GUI classes. Initial classes are provided as specified above. You will need to modify classes as required.

**DA - StudentDA class: - Methods must raise necessary exceptions**

* **Initialise** the data storage – read Student records from file and stored it an array list - called **arStudent** and **terminate** the data storage - must save arraylist to file. The files must be read and saved using object streams in a file called – student.dat
* **AddNewStudent ()** – it must add a new athlete record to arraylist, make sure there are no duplicate records, use the student no and subject to check for duplicates.
* **FindStudRecord** method, must find and return student based on the student no.
* **DeleteSubject** method, must cancel the student subject, use student no to find the record of the student who needs to cancel the subject and make sure the cancelled subject is not cancelled for all students who are enrolled for the same subject.
* **getAllStudents** method, must return all students stored in an arraylist.
* **getAllStudents** method, it must return all students enrolled for a specific subject.

**PD –** Student **class:** -

* modify class where necessary
* invoke (call) all DA methods accordingly

**GUI classes:**

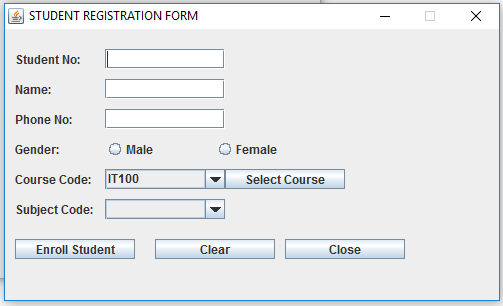
**StudentRegistration class**:

Your Screen (Frame) must be created as shown in the screen shot below:

It must have a fixed size and not resizable.

All components must be added to panel.

All components must be aligned correctly- no floating components



**Create the following method:**

* **getCourses(),** void method that read all courses from a given text file : courses.txt and all the data read must be populated to a course combo box.

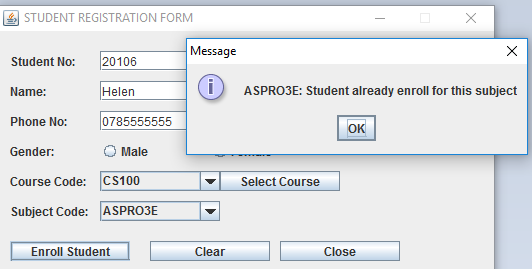
**Default constructor of this class** – call **getCourses()** method.

**On Button Select Course**

* Read all subjects per course code from a given text file called **subjects.txt** and all the subjects read must be populated to a subject combo box.

**On Button Enroll Student**

* **handle necessary excep**tion
* It must get all inputs from the user and call Student class and call **AddNewStudent** method to add Student to arraylist, if student already enrolled for the subject, it must show error message as in figure below.



**On Button Clear**

* It must restore form to its original state and set focus to student no textbox.

**On Button Close**

* It must close only this screen, not the whole application

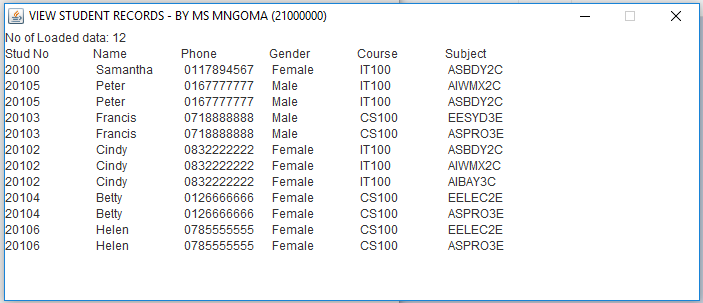
**ViewStudentRecords class**:

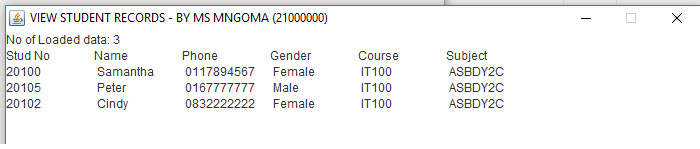
Your Screen (Frame) must be created as shown in the screen shot below:

**It must have textarea component, placed at the center of the border.**

**TextArea must not be editable.**

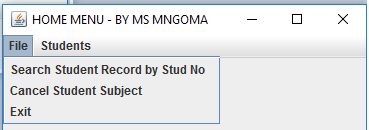
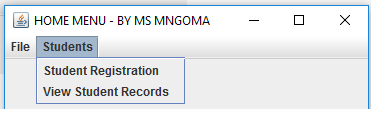
* Get subject from the user
* If subject input is empty – it must call getAllStudents () method to display all students from arraylist, otherwise it must call getAllStudents () to display all students enrol for specific subject. See figures below.
* It must also display the number of all records stored in the system as well as number of all records stored for a specific subject.
* Your display must have correct heading, as in figures below.





**HomeMenu class**:

* Your Screen (Frame) must be created as shown in the screen shot below:

On the **default constructor** of this class:

* Handle exception accordingly
* Call initialise method of PD class

On **Student Registration** menu item

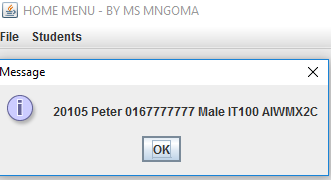
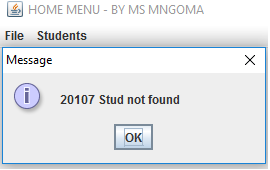
* Open the **Student Registration** screen, with screen name, your name and your student no as title of the screen
* It must have a fixed size and not resizable.

On **View Student Records** menu item

* Open the **View Student Records** screen, with screen name, your name and your student no as title of the screen
* It must have a fixed size and not resizable.

On **Search Student Record By Stud No** menu item

* Handle exception accordingly
* It must use input dialog to get the Athlete ID to be searched.
* It must call the **findStudRecord** method **of PD class** accordingly.
* It must display returned data in the message dialog box.

On **Cancel Student Subject** menu item

* Handle exception accordingly
* It must use input dialog to get the **Student No** to be searched.
* It must call the **findStudRecord** method **of PD class** accordingly.
* It must use input dialog to get subject.f
* It must call the **deleteSubject** method **of PD class** accordingly for the found student.

On **Exit** menu item

* Handle exception accordingly
* Call terminate method of PD class
* must exit application