GUI Programming
Module Code:

Module Title:	GUI Programming
Assignment Type:	Assignment
Project Title:	Student Management System
Project Date:	3/07/17
Assignment Compiler:	Kyle Goslin
Weighting:	100
Due Date:	**31st July 2017**
Method of Submission:	Moodle uploader link, on your course page, just below assignment.

Assignment Introduction

A college has come to you as a developer and outlined that currently they do not have any software to track the different students that are enrolling in their college. They have asked you to develop a system which will allow them to do this.

The system will have two different account types:

> **Admin** – The admin is responsible for enrolling students into courses and for creating new courses on the system, which students can be enrolled into.

> **Student** – A student can view the classes that they are currently enrolled in and their student profile.

Specific Requirements

- The system should have a login page where both admin staff and students can log into the system. Depending on the login type, the user should be directed to the correct dashboard:

  - Admin staff should go to **AdminDashboard.java**

  - Students should go to **StudentDashboard.java**

- The **admin** dashboard should have the following features:

  - View all the current courses in the system

- Add a new course into the system

- View all the students in the system (regardless of the course they are in)

- View all the students who are in a specific course

- Delete a student from the system

- Add a new student in the system – Each student must have their own unique student ID.

- Update a student profile in the system.

- Each student should have a profile in the system containing all the basic student information. An example of a student profile can be seen in the Appendix of this document.

- When students log into the system, they should be able to see their current profile information and make changes to the profile. Any changes that they do make should be saved in the database.

- A student should be able to see the name of the course they are currently enrolled in.

- The admin should have the ability to type in a student name and **search** through the database for the student they want to find. When they find the student they want, a new window should open with the student's profile information.

- Both admin staff and students should have the ability to log in and out of the system.

Notes

- All information which is in the system should be stored inside of a local MySQL database. Any changes which are made to courses or to a students' profile should be updated in the database also.

- When the system loads, a connection should be made to the database to allow for the Read/ Write/Updating of information.

- Remember **KEEP IT SIMPLE** – Draw a wireframe diagram to help yourself understand the layout of the program before you write it. If you start coding and then try to design the application you will get confused very quickly!

- Comment your code!!

- All code must be your own and not cut and paste from somewhere on the internet.

- This is an individual assignment, if your assignment is similar to another student's assignment you will both score zero.

Deliverables

- Java Source Code

- Database Table Dump

Marking Scheme Summary

Description	Weighting
1 – Admin and student dashboard working correctly and clear layout	10
2 – Admin can add a new student into the system and a unique student ID is generated for each student	10
3 – Admin can see all students; see collections of students in specific courses; assign students into courses	10
4 – Admin can make new courses and delete courses	10
5 – Admin can update a student profile that already exists	10
6 Student can update their own profile and changes are pushed to database.	10
7 Student can view the courses they are currently enrolled in	10
8 – Admin can search for a student by name and a new window will open showing that student's profile.	10
9 – Login and logout of both profiles working correctly	10
10 – Overall code quality, layout and design	10
TOTAL	**100**