**Applied Programming Concepts**

**Assignment #5 – Report**

**A close up of a logo

Description automatically generated**

**Team Names: Abdiel Danastor, Abdullahi Ibrahim, Joseph**

**Gbedema**

**Date: 07/08/2022**

ELEC3225 Applied Programming Concepts

Assignment #5

**What we implemented:**

* Add/remove course from semester schedule (based on course ID number).
* Assemble and print course roster (instructor).
* Add/remove courses from the system (admin).
* Log-in, log-out (all users).
* Search all courses (all users).
* Search courses based on parameters (all users) – you should be able to enter the parameters such as course code, day/time, etc.
* A menu to implement the use-cases.
* Edit classes as necessary to reflect the class diagrams.

We used Tkinter library to implement these functionalities to our leopard web application.

**Description**

Our Leopard web has a login/logout interface that allows users to login using their username (**email** in database from Assignment 3) and a **password** (Admin: **pass123** – Student: **pass1235** – Instructor**: pass1234**). The system can recognize whether that username belongs to an administrator, student or instructor and match it with one the previously mentioned password. You will get an error if username is not in database or password doesn’t match user type. Below are some screenshots of the login run-through.

Graphical user interface, application

Description automatically generated

Figure 1: Welcome/Login Page of LeopardWeb GUI

Graphical user interface, application

Description automatically generated

Figure 2: Admin Vera Successfully Login

Graphical user interface, application

Description automatically generated

Figure 3: Prof Joseph Successfully Login

Graphical user interface, application

Description automatically generated

Figure 4: Student Isaac Successfully Login

Graphical user interface, application

Description automatically generated

Figure 5: Login Failed (Wrong Password Entered)

After Login in, depends on the user, you get a new screen with a menu of the use-cases according to our use-case diagram. Let’s go through the menu for each user:

1. **Admin:**

Graphical user interface, application

Description automatically generated

Figure - Result after searching for course using CRN by an Admin

Graphical user interface, application

Description automatically generated

Figure - Screenshot after successfully adding a course to database by Admin

A picture containing text, scoreboard, monitor, screen

Description automatically generated

Figure - Updated database after adding course with CRN=4000

Graphical user interface, application

Description automatically generated

Figure - Removed course with CRN =3225 from the system

A screenshot of a computer

Description automatically generated with medium confidence

Figure - Updated database after removing course with CRN = 3225

**2 – Student:**

Graphical user interface, text, application, email

Description automatically generated

Figure - Student searching for Courses using CRN

Graphical user interface, application

Description automatically generated

Figure - Student add course with CRN=4000 to their schedule (attribute from student class)

Graphical user interface, text, application

Description automatically generated

Figure - Student tried to remove course that wasn't previously in schedule

Graphical user interface, text, application

Description automatically generated

Figure - Successfully removed course from schedule

**3 – Instructor:**

Graphical user interface, text, application

Description automatically generated

Figure - Instructor searching for course