ELEC3225 Applied Programming Concepts

Assignment #2

Micah Rieley

GroupMates: Regis Lach, Billy Hingston, Jared Hiller

Waterfall

1. **Requirements Definition**

Feasibility study:

As of right now, everything we need is available to us, this means we will not need any additional software or anything else. This means it will cost anything and we can do it cost effectively. We will be using c++ which is a language we are all highly familiar with and should be easy to do the majority of the work with, the only difficulty may be adding the functionality for the database and how each user type interacts with the database.

Requirement Elicitation and Analysis:

Looking at some previous code we wrote in previous coding classes can help us. This can give us general ideas on how we want to create the code and classes. We can also use online resources to give us an idea how we want to setup the database and how we want each user's functionality to be restricted/set. This will allow us to have an idea on how we want each user types functionality defined/created.

2. **System and Software Design**

Design for the system:

- **User:**
    - Attributes:
    - First Name
    - Last Name
    - WIT ID
    - Functions:
    - Set First Name
    - Set Last Name
    - Print Full Name/ID

- **Student:** Inherits from User
  - Functions:
  - Search Courses
  - Add/Drop
  - Print Schedule
- **Teacher:** Inherits from User
  - Print Schedule
  - Print Class List
  - Searc for Courses
- **Admin:** Inherits from User
  - Add Courses to System
  - Remove Courses from System
  - Add / Remove Users
  - Add / Remove Students from Courses
  - Search and Print Rosters / Courses

3. **Implementation and Unit Testing**

Use the previously created outline/objectives to create a table for each user type

Create the different classes and subclasses that will be needed, students, teacher, and admin. While adding each one it will be important to test different functionality along the way such as how they interact with the database and if they can edit or interact with it,

4. **Integration and System Testing**

Testing the overall functionality and how each user type interacts with each other. For example, how the admin is able to interact with the classes or students and how it will affect the student's schedules

5. **Operation and Maintenance**

Test as many functions as possible, and even try to break it, and let other people try and use it. This will help get as many different ways to test as possible as people will try and break the code in unique ways.