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.

Requirement Elicitation and Analysis:

Looking at some previous code we wrote in previous coding classes can help us.

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

4. **Integration and System Testing**

testing each ones functionality in relation to other user types

5. **Operation and Maintenance**

Test as many functions as possible, and even try to break it, and let other people try and use it

Incremental Development

Incremental Development

If the software development cycle uses the incremental development system, the primary objective is to allow students to view and register for courses and allow faculty and admin to also view and modify this data. As a result, the following would be prioritized for the initial release:

- A single database to store class information and user details for students, faculty and admins, as well as course information
- A table to store each user type (student, faculty, admin), where each table includes unique attributes depending on the user type.
    - Student: Name, ID, registered courses
    - Faculty:

    - Admin:
- A table to store the course catalogue and additional information.
    - Course name
    - Course dates/times
    - Professors associated with the course
    - Course capacity

In future releases: (WIP)

- Add functionality to create multiple schedules, make print-outs for schedules, and created preferences for schedules.

Integration and Configuration

Using Integration and Configuration, start by looking online to get a base idea of what you are going to try and do. From there, you can alter or change some code to get it to help you attain you desired results.

https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/

After finding a source to follow, using this source gives you a few important pieces that can be altered/used to benefit the project, this includes helping with creating a database, helping with testing if the database is open, creating and deleting items from the database, and selecting items.

I would start by implementing how they created the database, although while doing it, it is important to change the inputs/values in the table, so they align with what we want to use it for. For example, for a student table, we would want their name, their grad year, their ID, first name, last name, email, and major, but for instructor we want a different set, so for each table, it's important we have separate tables for each so the data doesn't get mixed or any other issues.

For selecting, deleting, and adding, it is a similar case, we need to alter the code to make it so it would be usable with the data we would be adding or searching for. This will allow us to be able to use the functions for our needs such as searching for students by last name or WIT ID number, but also adding students with certain data, such as only first or last name but not having any classes yet.