Nick DosRemedios

Partner: Robbie Cullerton

ELEC3225-01A

Assignment 2

Part 1:

Waterfall (Phase per bullet point):

* Create top hierarchy of classes – users 🡪 (admin, teacher, students)
* Add attributes for parent class, then attributes for child classes (if any)
* Create methods for parent class (print info, change name), then child class specific methods (admin – add class, teacher – change grade, student – check class)
* Main function for simple text-based UI
* Polymorphism for child classes
* Modify main function for interfacing (user enters 1 🡪 system knows what to do)
* Develop database infrastructure for required size
* Create database link between itself and the program
* Develop working methods for each class and successfully call functions
* Maintenance for implementation
* Graphical UI

Incremental:

* Version 1 – Parent and child classes with attributes – no main function, just some simple class creation to establish the scope – user, admin, teacher, student. Include simple main function for testing.
* Version 2 – Implement methods for user and student classes with inheritance and test successful function calls using print out responses. Update main function for a simple text-based UI for coherent function testing.
* Version 3 - Implement methods for the remaining admin and teacher, still with simple print-out message instead of real functions.
* Version 4 – Database creation and implementation with working functions (Polymorphism) that parse data and can successfully read out to our text-based UI.
* Version 5 – Debug until confident enough to establish a GUI.
* Version 6 (Final) – Have a full setup of users and data in the system, where everyone can successfully call their required functions to edit/view data. The GUI should also enable a user-friendly environment while navigating the program.

Integration and Configuration:

* Starting out with the development of the class system is straightforward if we follow our design requirements. Personal creativity at this point isn’t really required as we are just trying to develop a functional system.
* Moving onto formatting and creating functions, previously developed can be used as a steppingstone for the direction and scope of the program. These are helpful but not necessarily required.
* While developing the database, this module would be among the most helpful to build off previous systems and ideas. It would accelerate our timetable as we would not have to create a database from scratch. This leaves more time for refining the functionality of the program and setting the database to work with our own data. (i.e. tables storing certain information).
* Developing a proper GUI will also be streamlined through the help of online resources and previously established programs. This area of the project is where we have very little to no experience in creating. We plan on leveraging a text-based UI during the bulk of our program creation and testing and then plan on moving forward with some form of graphical based UI if the time allows for it.

Part 2: