Michael Sobchuk

ELEC3225

Professor Decker

June 15, 2021

Waterfall

The first step to waterfall design is to identify the requirements. For the CURSE program, the requirements are as follows. It must contain a database of users, and a database of course. The users will consist of 3 types, students, instructors, and admin. The student will be able to see available courses and their schedule. Instructors will be able to see available courses and their course rosters. The admin will be able to see everything and edit everything. The program should be capable of keeping track of multiple semesters and printing out schedules. With the requirements now defined, the next step in waterfall is to design the software. This software will be composed of two data bases, one for the courses and one for the users. These will both be used in the program. Data from the course database will be needed in the user database and vice versa. This will be programed in sqlite3 using c++. Sqlite3 is a library that can be included into a c++ program to execute sql code. Each of the subclasses of user will contain functions that will interact with the data base. Their variables will also be initialized by a method that reads the user database file and creates the correct user objects. The student class will contain get methods for their attributes as well as methods to access their classes in the database. The instructor will contain the same getter methods but have methods to access their courses and rosters. The admin will contain the getter methods, in addition to the methods for editing database entries and user objects. With a design plan laid out, the next step in the waterfall process is implementation. This entails turning the planned code into real code that functions. Once all the code is written, the next step is verification. In this step, the code will be tested thoroughly to ensure there are no bugs and that it functions as intended. Once this step is finished, the code is essentially done. The only remaining step is maintenance which is fixing any issues with the program that may come up that didn’t from previous testing.

Note from Editor:

In this model not much was said on the pseudocode on how the user will be able to access the databases (through user interface). Also, on the verification step what should be mentioned is how are you going to be able to verify the program through testing, for this example the verification should come from the professor and any students that come by and use your CURSE program.

Incremental

The essential idea behind incremental programing is creating a small portion of the code that works as intended, and then adding additional functionality once that works. You repeat these processes over and over until all functionality has been implemented. For the CURSE program, the most logical place to being is creating the class structure. This means creating a working code that you can create a student, instructor, and admin, all derived from the user super class. These will also contain the empty methods for the database accesses. While there will be no functionality in these, they will be created. Once this is done, the user database will be implemented into the code. This will require some additional methods to be added. The program will be required to created user objects from this database. The admin class should be able to edit the entries in this database using methods contained in the class. Once this is functioning properly, the course database will be implemented. To properly implement this, functionality for the methods in the student class will be added to access the database and see their courses. In the instructor’s class, implementation of methods to view the database and view the class roster of their courses. The admin class will need to have functionality to edit the database entries. After the course database is added, the code will need to be modified to support multiple semesters. This should be relatively simple, but some method of ensuring the correct term is selected when accessing the class database. In between each step, the code must be tested thoroughly. Once the last iteration is complete, the code is fully finished and has all the desired functionality.

Note from Editor:

Preferably my first iteration would be creating a simple user interface/ GUI that links up to a simple base class called User and possibly add more derived classes and methods onto the next iteration and then after the database in SQL.

Integrate and configure

When using the integrate and configure, you utilize already existing code and implement it into your design. By doing so, you are reducing the amount of work that you must complete in order to achieve the same outcome. For the CURSE program, implementation for the databases is readily available and can be used for this application. An open-source object-relational mapping system called ODB allow for you to persist, or link, objects to a database. This does so without the programmer having to deal with and sql code. While we would still need to use sqlite3 for the course database, this could be applied to the user database. This is because these users will be classes as well as stored in a database. With this, it drastically saves the amount of code that would be required to write for the user database while achieving the desired functionality.

Note from Editor:

Add the screenshot or a link of the preexisting code then go a little more into detail with the actual syntax of the code from the ODB and how it will be implemented and adjusted to fit what you want to do.