Waterfall Model

Requirements Definition:

Database of users: the system should work for 100 students, 10 instructors, and 1 admin, however, we will test with fewer. Database of courses: this will contain information such as the CRN, course name, times, and instructor. Three types of users: student – can register, can see available courses and their own schedule. instructor – can see available courses and their own course roster. admin – can see everything, can edit courses/users/schedules. The system should include multiple semesters, print-out of schedule, scheduling preferences. The system as a whole and all components must be tested thoroughly.

//the system should have a GUI that allows users to interact

System and Software Design

You need a base class that is the User and have more classes such as, student instructor and admin inheriting from the base class. Each class has there own functions for example admin has access to everything compared to the student.

We need a database to hold all the information and a query to be able to find the data. We also need a user interface that resembles leopard web for the user to find data.

Implementation and Unit Testing

This is when we put the code together and run tests on our code. We will have to test everything for each class and make sure all functions work. Also connect our database and create the query.

//the database should allow the different items to work together and have unique keys to class the different information

Integration and System Testing

//we will use python to intergrade it as well as visual studio code and mysql

This is where we will integrate the database with our code so we can fully test the system and work on long term testing to get to the next step.

Operation and Maintenance

This is where the program is almost finished and we are fixing small bugs and implying updates to make sure all the data is up to date.

//we will fix the bugs that we find in testing

Specification:

This is where we have to meet the specification for our code. And make sure you have everything you will need to make the code. This is the beginning version.

Development:

This will be where we develop the code and make sure we meet our requirements. This is the intermediate version.

Validation: This is when someone will check over the work and tell you what needs to be fixed or if It meets all the requirements. This is for the final version.

Integration and Configuration

Requirements Specification

//add what the requirements are for the GUI

Database of users: the system should work for 100 students, 10 instructors, and 1 admin, however, we will test with fewer. Database of courses: this will contain information such as the CRN, course name, times, and instructor. Three types of users: student – can register, can see available courses and their own schedule. instructor – can see available courses and their own course roster. admin – can see everything, can edit courses/users/schedules. The system should include multiple semesters, print-out of schedule, scheduling preferences. The system as a whole and all components must be tested thoroughly.

Component Analysis

This is where we will research and find any existing code that could help us with our code. This step will keep us moving more.

Requirements Modification:

This is when you will revide original specs with components and make sure all your speifications are correct and what they should be.

System Design with Reuse: //maybe add some spefic compnents we would use

This is where you will use the components and the code and will modify anything that is needed to keep on going to the next step.

Development and Integration:

This is where you will integrate everything you have developed