Waterfall Method:

Requirement analysis and definition: The Leopard web project must include a database of users, a database for courses that contain CRN, course name, time, and instructor. We need classes for users, students, instructors, and admins with their own specific methods. The class users are the base class and the other classes are inherited from the base class.

System and software design: The overall system of the project is to have a graphical user interface that will have menus and calendars that will allow users of specific levels to either add courses, create new ones, basically follow the methods that each class has, and implement those into the GUI. There are three classes and the main menu will be the host to other menus to access all the different methods. The GUI will be specific so that any front-end user will be able to work the GUI and not have any issues when adding new classes, students, etc. This is good for system and software design.

Implementation and unit testing: This section will be making sure that the code works together and that we are able to create classes and have these classes accessed through a basic user interface. Checking for basic flaws in the code before implementing a database. This section will also create the basic database but will not integrate into the code during this step.

Integration and system testing: For this section, we will create a database, implement it into the GUI, and initialize testing. The database will be a lot smaller for testing purposes, but the main idea of this section is to make sure that the database and GUI are working seamlessly together where users can be added and accessed through the GUI. The basic database will have data of all the students, instructors, admin and the courses an more.

Operation and maintenance: This section will be trying to break the code that we implement and find any bugs that may come from the GUI or from the database. From here based on our issues we will reformulate what is needed and what may be unrealistic to implement.

Incremental development:

Specification: An initial build that will have a basic GUI and a basic database to test and implement. This will have the basic requirements and will be put together so a very basic initial version of leopard web. This will be very basic in order to have room for improvement and suggestions. For example, this will have one menu selection and will have one user input just to test and check with the customer to make sure the layout, functionality, and implementation are up to par.

This is the most basic part hit all spots.

Development: this will be where the heart of the code is created. We take our basic initial version and develop more advanced code and GUI. Here we will have new versions for each basic change and a new version for larger changes. An example of larger changes would be new menu sections, or creating schedules for students, anything major change would be kept track of in a new version of the GUI.

Validation: This step is to confirm our changes and get direct feedback from the customer and go back to the Development or specification section depending on feedback. The importance of this section is to take suggestions from the user to make the project as ideal as possible for the customer. Also in this section, we discuss deadlines and what is possible to add/change from a software perspective.

Integrate and configure:

Component analysis: At a quick Google for School software in Python It came up with a bunch of pay-for-software classes or just for the software itself but <https://www.sourcecodester.com/python/14520/school-management-system-project-python.html>

has a YouTube video that shows some steps for this.

This section will help you get started for the assignment and give you the tools you need for starting to code.

Requirement Modifications: Taking this we can implement a different gui but have similar ways of collecting information such as through user input textboxes so using the video for the method behind doing so might help us formulate a similar output.

Our requirements still need to meet the project standards.

System design: Take the textbox example and make it our own in a way by having menus and a calendar option for the schedule and possibly having drop-down menus for courses that can be added and removed for students and new courses can be added through either the admin or teacher view. The example software doesn’t seem to have different levels to the software that we will be adding such as the teacher view and admin privileges. This is where you design the system and even make a user interface.