**Waterfall:**

Create a scheduling system like leopardWeb. The system should utilize a database of users, and should implement different user roles including instructors, admins, and students.

After the system is tested bugs will appear over time. These bugs will need to be patched. Additionally, user feedback can be integrated into the system in order to make it meet additional needs.

Final testing will need to be done when the entire system is built. This testing can be handled by unittests to check basic functionality like updating users and classes.

The system should be designed in separate pieces. Focusing on the databases early is important as it will prevent the need to go back and update code later. The software will be designed in Python.

Unit testing can be handled by pythons unittest framework. Tests can be written for each individual component as the system is built out to ensure functionality.

**Requirements Definition:**

Estimated timeline: Week 1

* Create a scheduling system like leopard web that includes functionality such as adding and removing courses, adding and removing users, displaying a user’s schedule, and removing or adding classes to a student’s schedule
* Utilize databases to store user information
* Create an easy-to-use interface for users
* Provide different levels of user access depending on the classification of the user. For example, students should not be able to add or remove classes from the system

**System and Software Design:**

Estimated timeline: An initial system design should be decided upon by week 4 with adjustments made as needed.

* The overall system will need to be made up of individual components. This will include
  + Multiple relational databases for user information and class information
  + A Database control node to handle all database transactions
  + A set of classes defining users and user-related functions
  + A user interface

**Implementation and Unit Testing:**

Estimated timeline: a preliminary implementation with unit tests should be completed by week 11

* Breaking up the project into sections including the databases, user class, and others, write the individual components
* Utilize python unittest framework to create tests for every component

**Integration and System Testing:**

Estimated timeline: The system should be entirely ready for testing by week 13

* Combine all components including the databases, and UI into one application
* Have human users test the program and attempt to find bugs

**Operation and Maintenance:**

Estimated timeline: The remainder of the semester should be dedicated to operation and maintenance.

* As user feedback is returned from end-users make improvements as necessary
* Continue to verify that packages are up to date and patch any security vulnerabilities