**Incremental:**

An initial version would not integrate pieces like databases and a UI.

The final version should include all the features necessary to have a working leopard web program. Databases should be functioning, and a user-friendly UI should be present.

Intermediate versions will include additionally functionality that will be needed in the final version. Databases could be added in a version, and a UI could be added in another.

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.

**Outline Description:**

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

**Initial Version:**

Estimated timeline: Week 5

* Specification
  + The initial can include core functionality like the basic class structure and different user types.
  + This release does not need to include databases or a robust UI and can instead focus on core functions
* Development
  + Add core functionality necessary in an initial release
* Validation
  + Create unittests and have developers check all initial functionality

**Intermediate Versions:**

Estimated timeline: Rolling releases as they are ready up until week 12. A version including databases might be release as early as week 7, while a UI might take until week 10.

* Specification
  + Break the project into pieces
  + Separate features like the addition of databases into separate releases
  + Each release can test its respective additions and be tagged on GitHub
  + Each release should retain all previously added functionality and serve as a functioning standalone release.
* Development
  + Add components depending on the version
  + Early versions might include the addition of databases, and a web interface
  + Later versions might include bug patches
* Validation
  + Create unittests and have developers check all added functionality
  + By testing functionality periodically in every version, the chance of finding a core bug after release is decreased.

**Final Version**

Estimated timeline: Week 14

* Specification
  + The final version should be as robust as possible integrating all the components needed to get the leopard web program to work.
  + Everything including databases, and a UI, should be included in this release
* Development
  + Ensure that all necessary components are available within this version.
* Validation
  + Using feedback from end-users, changes can be made to validate all functionality, additional functionality can also be added.