

Waterfall

Complete each step completely before moving on to the next one in order.

1. Requirements Definition

Feasibility study – This project will not require any budget and the current technology available to us will allow for the project to be completed technology as it will require databases, coding, and a user interface.

Requirements elicitation – Ask the professor about any requirements that are unclear to us. Research can be done to look at systems that appear to be similar to the one we have in mind.

2. System and Software Design

There will be a base class: User.

- All users have a first name, last name, ID, and email. (attributes).
- All users can edit their attributes (individual functions for each attribute) and print all their attributes.

Student class will inherit User.

- Students will also have an expected graduation year and major.
- Search for courses that will allow them to check their availability.
- Add/drop courses from their schedule. The student will be notified of any time conflicts that may result from adding a course to their schedule.
- Print their schedule. (Each created Student has their own schedule).

Instructor class will inherit User.

- Instructors will also have title, year of hire, and department.
- Print their schedule. (Each created Instructor has their own schedule).
- Print their class list.
- Search courses.

Admin class will inherit User.

- Admins will also have title and office.

- Add courses to the system.
- Remove courses from the system.
- Add/Remove users.
- Add/Remove students from a course.
- Search and print rosters and courses.

It was decided to use a text-based user interface. Functions will be used to create the methods for each class and every User, Student, Instructor, and Admin will be created as an object. A database will be utilized with four tables.

Student – ID, first name, last name, expected graduation year, major, email.

Instructor – ID, first name, last name, title, year of hire, department, email.

Admin - ID, first name, last name, title, office, email.

Classes – CRN, Title, department, time, day(s) of the week, semester, year, credits.

3. Implementation and Unit Testing

Write code, create a database, and create tables all while testing the components individually.

4. Integration and System Testing

Integrate the components and test them working together.

5. Operation and Maintenance

Let others use the system. Make note of any bugs and fix them.