**Waterfall:**

**Requirement Specification:**

* System will allow all students, faculty, and administrators to add courses, search for courses, print schedules, etc...
  + Display the student’s schedules to Admin and Faculty
  + Display to Faculty their class list of students
  + Display all courses offered
* Faculty and Admins can view and update course information
  + Admins can add and remove courses offered
  + Admins can add and remove CRNs
  + Faculty/Admin can access the database of student schedules
  + Admin can view rosters
* Admin can change Student enrollment status
  + Admin can add additional students
  + Admin can remove students
* The system will include multiple semesters
* Display courses that are offered for each semester

**System and Software Design:**

Architecture Design:

* Student
  + Functions to search course, add course to schedule, drop course from schedule, withdraw course and display schedule
* Faculty
  + Functions to print schedule, display their class list, update class time and search classes
* Admin
  + Functions to add available courses, remove courses, add and remove students, view student schedule, adjust student schedule, search a roster, and display all courses

Interface Design:

Students will only be able to control what courses they are registered for and display their schedule. Faculty can view roster schedules and their own. Admin can add and remove courses as well as update students’ enrollment status. Each class will be derived from the parent class User which will have all the common attributes and functions.

Component Design:

Each class has similar methods that we can reuse such as search courses and display schedules.

Database Design

* Student table:
  + First name, last name, ID, courses registered, CRN, schedule
* Faculty table:
  + First name, last name, ID, roster, schedule, courses taught
* Course table:
* Course name, CRN, times, and instructor.

Next Step:

**Implement** – write your code, create your database, etc.

**Unit testing** – test the components individually; usually automated

Next Step:

**Integrate all components both front and back end into a functional website and test the website as a user.**

Next Step:

**Let an actual user use the website and if any bugs occur, retest and make any changes.**