**Waterfall**

**Requirement analysis and definition**

|  |
| --- |
| **Software Specification**  Feasibility study: The system can be built with current hardware and software, and it is affordable.  Requirement elicitation and analysis: Look at LeopardWeb for similar functionality.  Requirement specification:   * All users will have first name, last name and ID as their attributes and print all info for the object. * Students can register, can see available courses and their own schedule. * Display course table – CRN, course name, times, and instructor. * Can search for course. * Can add/drop course. * Can print their schedule. * Instructors can see available courses and their own course roster. * Display course table – CRN, course name, times, and instructor. * Can print their schedule. * Print their class list. * Search for courses. * Admin can see everything, can edit courses/users/schedule. * Display user table – first name, last name, ID, occupation. * Display course table – CRN, course name, times, and instructor. * Can add/remove course to the system. * Can add/remove users. * Can add/remove students from a course. * Can search and print rosters and course.   Requirement validation: The requirements are realistic and match what the customer wants. |

After completing this step, move on to the next step.

**System and software design**

|  |
| --- |
| Design and Implementation  Architectural design – database, classes and objects, functions, and user interface.  Interface design – Design graphic user interface.  Component design – Use user as base class to derive to 3 classes (student, instructor, and admin) and use function for anything that will be used multiply time.  Database design – There will be 2 tables (users and courses)  Users table  Column: first name, last name, ID, occupation.  Courses table  Column: CRN, course name, times, and instructor. |

After completing this step, move on to the next step.

**Implementation and unit testing (component testing)**

|  |
| --- |
| Implement – Create database (users and courses), write code.  Unit testing – test the components individually. |

After completing this step, move on to the next step.

**System testing**

|  |
| --- |
| Integrate the components into a system and test again. |

After completing this step, move on to the next step.

**Operation and Maintenance**

|  |
| --- |
| Let the user use the LeopardWeb clone and get feedback. If they need any changes, start again. |