Persia Pena Cruz

ELEC3225 Applied Programming Concepts

May 13, 2024

**Waterfall**

A University needs a scheduling system similar to LeopardWeb. The system will allow students, faculty, and an admin to add courses, search for courses, print schedules, etc depending on the role they choose.

**Requirements Definition:**

|  |
| --- |
| Software Specification   * Feasibility study – The project can be built with current software and hardware. * Requirements Elicitation – Interview the user (Admin, Instructor, and students) to find out what the system needs most. Look at Leopard Web (Scheduling System) to see what would most benefit the system and see similar functionality.   Requirements Specification   * All users can view a list of all the courses available.   + Display a table of all courses listing course number, course name, CRN, location, meeting time, instructor, and number of credits.   + Display a table of all subjects offered at University: For example, list all math courses or all electronics courses. * Admin can view and update all information/data in the scheduling system.   + Can view and update rosters and class lists (add/remove students)   + Can view and update the users in the system (add/remove users)   + Can view and update the courses in the system (add/remove courses) * Instructors can view all information pertaining to their schedule   + Can view and print their schedule (courses they teach)   + Can view and print their class list   + Can view and search for courses * Students can edit all information pertaining to their schedule   + Can view and search for a list of courses   + Can View and update registered classes (add/remove courses from schedule)   + Can view and print their schedule (courses their registered for)   Requirements Validation: Check to see if the requirements match what is required for the system and matches what the customer wants. |

**THEN after this has been completed**

**System and Software Design -**

|  |
| --- |
| Requirements Specification -   * Architectural Design - * Interface Design – Text Based User Interface * Component Design – User information such as name, and ID number will be reused through the code for class lists * Database Design – For now the code will have 2 tables for the courses and class lists   Courses Table   * Table Name: Available Courses * Columns: Course Name, Course Number, CRN, Meeting Times, Instructor, Location   Classlist/Roster Table:   * Table Name: ClassList for (Course Name) * Instructor Information: First name, Last name, ID number, Role * Student Information: First name, Last name, ID number, Role   Does everything need to be stored?   * Any action made by a student, instructor, and admin needs to be stored. For example, every course a student registers for must be stored so it will show their name under the class list. When a student is removed from a course by an Admin, that information must be stored. |

**THEN**

**Implementation and Unit Testing** –

|  |
| --- |
| Implementation- Write code and create database and  Unit Testing - test each component unless the code is successful. |

**THEN**

**System Testing**

|  |
| --- |
| Integrate the components of the system and test it again. |

**THEN**

**Operation and Maintenance**

|  |
| --- |
| Let the user use the system and test for any errors. If the user needs any changes, start at the beginning of the waterfall process model again. |