Toufic Shoukeir

Computer Networks for Engineers

Section 01

Assignment 2

Professor Rawlins

19 May 2025

Problem Statement

An Aquarium needs a software system to manage their marine mammals and birds, and to manage their staff (trainers, veterinarian technicians, and veterinarians).

Requirements Engineering

1. Feasibility Study

• Does the technology needed to create the system exist?

Yes, the technology to create our own version of Leopard Web exists. We need objects, classes, a database and a user interface.

• Does it fit in the budget?

We always assume yes in this course.

Result of Feasibility Study:

The project is feasible and affordable to create.

2. Requirements Elicitation

I have examined past systems such as the current version (and have used other versions over my time at WIT) of Leopard Web. From this, the requirements are:

- A database to store and manage user informaion
  - Schedules
  - Courses

  - Names, IDs, Job Titles, User Types
- Support for 4 types of users
  - Admin
  - Instructor
  - Student
  - User

Specifications from Users/Customers:

- The system should work for 100 students, 10 instructors and 1 admin
- Database of courses should include CRN, course name, times and instructors.
- System should include multiple semesters
- The system will allow students, faculty and admin to add and search for courses, print schedules, and more.

<u>3. Requirements specification</u>

- Base users should be able to:
  - Set their first and last name
  - Set their IDs
  - Print their information from the database
- Students, derived from the User class should be able to:
  - Search courses
  - Add/drop courses
  - Print their schedule
- Instructors, also derived from the User class should be able to:
  - Print their schedule
  - Print their class lists

  - Search for courses

- Admin, again derived from the User class should be able to:
  - Add courses to the system
  - Add/remove users from the system
  - Add/remove users from a course
  - Search/print all rosters
  - Search/print all courses

4.Requirements Validation

{This is where the group mates validate that the requirements specified match this document.}

Design and Implementation

1. Architectural Design: This system will consist of multiple high-level components such as classes, objects, functions, database and a user interface.
2. Interface Design: These will all be connected by importing our own programming files into each other, allowing class inheritance, database connection and a smooth UI experience for the user.
3. Component Design: The classes will all be derived from the base class, “User”. This will allow for functions to work between all classes (such as print_schedule and print_classlist). For now, everything is text based. In the upcoming weeks we will learn how to incorporate a UI into the program to allow a realistic experience on the front end of the system.
4. Database Design: We will have a table for each class, which will be the users, students, instructors and admin. The system will be able to pull data from these tables at any time. The number of columns in the table is determined by the class. For users, there will be three columns, one with first name, one with last name and one with ID number. For

Students, they will have more columns, adding the category of courses. For instructors, we will add the category of which department.

- Users: first and last names, ID number
- Students: first and last names, ID numbers, majors, courses
- Instructors: first and last names, ID numbers, department/school, classes being taught
- Admin: first and last names, ID numbers, department/school

Software Validation

1. Component Testing: we will test the individual elements as they are being built, such as each class, each database table, class lists and schedules.
2. System Testing: Once we integrate the components into our system, we will do a more extensive round of testing now that our system is more put together.
3. Acceptance Testing: We will test the system with real data

Software Evolution

1. We will continue to modify the system over the course of the semester, adding more functionality and fixing any bugs that come up.