ELEC3225 Applied Programming Concepts

Assignment #2

Regis Lach

**Waterfall Timeline:**

With the waterfall development process, the ideal starting point would have to begin with the database- as all other features of the Leopard-Web system will rely on it. This will be composed of several tables that store information for each user type, as well as the courses available. This can be designed as follows:

Stage 1 – Create Database:

* Create a database that contains the necessary tables and column names for the required information.
  + Table 1: Student
    - Registered courses
    - Restrictions (if any)
    - School year and expected graduation date
    - Credit hours required for school year
  + Table 2: Faculty
    - Courses they are teaching
  + Table 3: Admin
    - Admin does not need special attributes as their main role is function related.
  + Table 4: Course
    - Course name
    - Course prerequisites
    - CRN
    - Schedule and time
    - Instructor
    - Availability
    - ID of registered students
* Additionally, each user would have a set of column names that are the same between user types, of which are:
  + First and last name
  + ID

Stage 2 – Functions:

* All users will be given the following:
  + A login interface will be designed that allows users to login to their accounts. This will be created with options to reset a password and require 2-factor authentication for security purposes.
* Student
  + Register for courses (if prerequisites are met and there are no restrictions)
  + View available courses
  + Print schedule
  + Add/drop courses from their schedule
  + Create multiple plans that they can select on registration day
* Faculty
  + View courses and course roster
  + Print their schedule
* Admin
  + Add/remove courses to/from the system
  + Add/remove users
  + Add/remove students from a course (regardless of met prerequisites/requirements)
  + Search roster/courses
  + Print roster/courses

Stage 3 – User Interface:

With the primary functionality created and tested, the next phase of the project would be to create a user interface that is easy to navigate and utilize. Examples of this would include displaying a depiction of a student’s schedule out on a per-week calendar, as opposed to just displaying their course names and times. Features like this give better visualization of how a course might fit into an existing schedule, or perhaps to get a better sense of their course load.

Stage 4 – Release and feedback:

After each stage is completed and tested for issues, the system is then released to the school for use. The designers will then wait for user feedback and address any potentially missing bugs/issues, but otherwise, the system’s primary functionality is complete.

**Incremental Development Timeline:**

If the software development cycle uses the incremental development system, the primary objective is to allow students to view and register for courses and allow faculty and admin to also view and modify this data. As a result, the basic functionality will be the main priority, and additional improvements to the user experience will be implemented in future releases.

Stage 1 – Simple system:

* A simple database to store class information and user details for students, faculty and admins, as well as course information
* A table for each user type (student, faculty, admin), with first and last names and ID as the main attributes. Additionally, there will be some unique attributes/functionality depending on the user type.
  + Student
    - Registered courses and schedule
    - View and search courses
    - Add/drop courses
  + Faculty
    - View course roster and available courses
    - Print their schedules
  + Admin
    - Add/remove users
    - Search and print rosters/courses.
* A table to store the course catalogue and additional information.
  + Course name
  + CRN
  + Course dates/times
  + Professors associated with the course
* A login interface that requires the correct credentials to login. In the initial release, there are no additional features such as 2-factor authentication or overriding student restrictions. There will be an option to reset a password in case the user forgets their password.

With this simple system developed and tested, it would then be released for utilization by the school, with continuous observation of user feedback as additional features are designed for future releases.

Stage 2 – Additional Features:

* Add functionality to create multiple schedules, make printouts for schedules, and create preferences for schedules.
* Add course restrictions and prerequisites, which will prevent a student from registering for a course if they do not meet these.
* Add student count for each course.
* Allow admin to add/remove students from courses regardless of requirements and prerequisites met.
* Design an aesthetically pleasing user interface for users to interact with.
* Add 2-factor authentication to the login.
* Any additional feedback received by the users of the system. This will not take precedence over the other planned features unless it is an issue that cannot be ignored

**Integration and Configuration Timeline:**

Under this development model, we could take a wide range of routes. One is to start by integrating a more generalized tool and building most of the infrastructure from that. The other would be to look into a more complete system to integrate- one that has a database, and user system already set up. While the latter would likely be the quickest to configure and deploy, it would be the most expensive option. So instead, we will choose a more bare-bones system to integrate and configure.

Stage 1 – Research:

Stage 1 of this development model is a bit different from the first two. Here we would invest time in looking for the database model that would best suit our needs in terms of feature set and pricing. One popular option is Amazon Web Services (AWS), which offers an entire suite of tools to create both our database and user interface. This would be my option to use, as I have experience using this service to create and manage a database.

Stage 2 - Integration and Configuration:

Database:

With AWS, the tools to set up are readily available for use. Here the database and tables are created, and column names are specified according to the outlined requirements. AWS has two databases to choose from- dynamoDB and Timestream. As timestream is designed for time-based applications, we will choose dynamoDB for the school database.

UI:

While we are already in the AWS ecosystem, a possible option to use is [AWS Amplify](https://aws.amazon.com/amplify/), which offers a streamlined way to design and configure our UI. With this application our interface would be designed, and our user functions developed. This would follow the same schema as the waterfall timeline.

Stage 3 - Deployment and feedback:

With the database and UI configured, the next step is to deploy the system for use by the school. As part of AWS’s system, the process for setting up users is as simple as giving the administrator access to the framework of the system, where they can provide account details and add users to the system. Once the system is set up, we will be continuously monitoring user feedback and making any necessary fixes if they are discovered after release.