Curriculum Change Management System (Prototype)

COMP 4513 Assignment 1

Joey Lai – 201547499

Table of Contents

Functionality (Use Cases)

I chose the workflow of submitting a program request for the proof of concept. The workflow (approval chain) is persisted to the database, and can be easily changed through a SQL script. I have only chosen a select number of fields for a new project request as part of the proof of concept. More fields can be easily added to the system with minimal effort.

The current approval chain I have set up currently for a new program request is:
1. Faculty Curriculum Member
2. Faculty Council Member
3. APPC Member
4. GFC Member

Once the last responsible party has approved the request, the request will be updated to a status of "Completed". The responsible party can also reject a request, updating the status to "Rejected". Once a request has been rejected, it will not longer advance to the next approval chain step.

The following are the functional use cases I chose to implement for the prototype:
1. A user can sign up for an account for the system with a specific role.
2. A user can login to the system.
3. A user can submit request to create a new program.
4. A user can view the status of their request.
5. A user can approve any awaiting requests.
6. A user can reject any request.
7. A user can search for any requests in the system.
8. A user can add comments to a request.
9. A user can add files to a request.
10. A user can download any files uploaded for a request.

The visual designs/user experience will be improved in the second iteration of this assignment, using client side code.

Application Layers

The design of the system is broken down into four primary layers: View, Controller, Model, and Persistence.

View

The view layers are the HTML web pages presented to the user. These pages consist of data transfer objects (DTO), which are responsible to presenting data to the user as well as taking in user input.

Controller

The controller layer takes DTO's from the View layer and is responsible to passing them to the Model and Persistence layer. The controller layer uses the Façade class to make the appropriate requests. The controller is also responsible for sending the response back to the user and redirect them to the appropriate page.

Model

The model layer is the actual business domain object model layer. This layer is responsible for any assertions and validations. Any business logic is also found here. The business domain classes mirror the tables on the database.

Persistence

The persistence layer is responsible for persisting business domain objects to the database. This layer calls the appropriate Repository classes to persist/read from the database.

ERD Diagram

View attached document, "curriculum_change_management_system.png", for ERD diagram.
The business domain classes mirror the tables on the database.

Steps for loading test data

The mysqldump file, named "mysqldump.sql", contains the backup of the MySQL database with all tables and stored procedures.

The stored procedure, "recreate", cleans up the database and loads test data into the following tables:
- Faculty
  - Bissett School of Business
  - Science and Technology
- Disciplines
  - Management
  - Marketing
  - Computer Science
- Roles
  - Faculty Member
  - Faculty Curriculum Member
  - Faculty Council Member
  - APPC Member
  - GFC Member
- ApprovalChain
  - Program
- ApprovalChainSteps
  - The approval workflow for program
- Status
  - PENDING_APPROVAL
  - APPROVED
  - COMPLETED
  - REJECTED

Appendix

Code revisions can be viewed/downloaded on Github:
https://github.com/jlai403/COMP4511-Curriculum_CMS