ENSF 409 – Principles of Software Development

Winter 2018

Term Project

Due Dates	
Milestone I **Design**	Submit electronically on D2L **before 11:59 PM on Friday March 30.**
Milestone II: **Development** **(Iteration a)**	**Demo:** During the lab section on **Friday April 6**. ***All group members must be present for the demo.*** Submit electronically on D2L **before 2:00 PM on Friday April 6.**
Milestone III: **Development** **(Iteration b)**	**Demo:** During the lab section on **Friday April 13**. ***All group members must be present for the demo.*** Submit electronically on D2L **before 2:00 PM on Friday April 13.**

This is a Group assignment. Students can work in groups of up to three people. You can pair up regardless of your TA groups. Please note, students have the option of working on their own.

Project objectives

1. To gain experience with the design and development of a relatively larger software project
2. To gain practical experience in programing concepts introduced in ENSF 409 such as object-oriented design and development, client-server architectures, multi-threading, and JDBC
3. To gain experience with iterative development process
4. To develop team work and presentation skills as a programmer

Instructions groups

Students who work in groups, must follow the following guidelines and regulations:

1. The workload **must be divided equally** between group members. During the demos, the TAs will specifically ask about what each student has done in order to ensure all students have contributed.
2. All students must be present at each demo. If a student is absent, that student will not be awarded the grades for the demo.
3. Only one of you needs to submit the final project on D2L by the due date. However, please make sure all of your names are included in your submission.
4. Teams must use Git to manage versions of their project

Important Advice: When working as a team, make sure to clearly define your goals and expectations, as well as internal deadlines. Ensure to communicate regularly and schedule times to meet and work together.

Software Design and Development – Learning Platform (100 Marks and up to 20 bonus marks)

This project aims to closely simulate the design and development of a real-life software project. For a given software system, you will be given a set of functional requirements, as well as a set of constraints that your design and developed code must adhere to. In the interest of time, certain functionalities have been omitted or simplified. (See notes on this throughout the requirement description). The other notable difference between what is asked in this project and real-life projects is the absence of automated testing. In the interest of time, you are not asked to write J-Unit tests for your application.

Important Note:

This project provides you with the opportunity to use the skills and knowledge you have developed in this course and develop a solution for a problem.
As such you have been presented with a problem statement and no other directions. It is on you to research, design, develop and present your project, as you will have to do throughout your careers as software engineers.

To ensure you are on the right track, the class diagram of the system will be discussed in lecture.

Requirement Description

You have been asked to develop a client-server application for a learning platform (like D2L). This platform will have several courses which are saved in the system's database.

Each Course contains the following information:

- Course number (which is unique)
- Course name
- Professor name
- Whether it is *active* (visible to students) or not
- Course assignments

There are two types of users in this platform: students and professors. This means you should have two different GUIs: one for students, and one for professors (More about the functionalities of the two GUIs later).

Each user will login to the system by entering their ID and password in a window. This login information must be stored in the database and checked by the server. If successful, the program should start either the professor or student GUI for its user.

Functional Requirements:

Through the Professor GUI, the user will be able to:

1. Create new courses, and browse these courses (but not other professor's courses)
2. Set a course as *active* (which makes visible to students) or vice versa
3. Select a course to:
    a. Search for students by id number or last name, and browse the search results
    b. Select a student to enroll or unenroll from the course
    c. Upload a new assignment file to the course
    d. Send an E-Mail to all students enrolled in the course (these email addresses are stored in the systems database)
    e. View and grade student submissions in a dropbox (there should be one dropbox per assignment)
    f. Set an assignment as *active* (which makes it visible to students) or vice versa

Through the Student GUI, the user will be able to:

1. Browse through courses they are enrolled in
2. Select a course to view a course page.
3. From each course page students can:
    a. Select an *active* assignment, which allows students to view the assignment file inside the GUI window
    b. Submit a file to an *active* assignment's dropbox (PDF or Text format)
    c. View grades received on assignments
    d. Send an E-Mail to the course's professor

Note 1: I have not asked for any registration functionalities, so you can just load a set of students and professors (few of each) into your data base.

Note 2: Carefully consult the Project Help Document posted on D2L in designing and developing your project.

Constraints

Your design must adhere to the following constraints

1. This system must be developed as a client-server application
2. The client side of this application must have GUIs
    a. The GUI of the system must be organized, and user-friendly. You are strongly urged to do some research and look at similar online systems to come up with ideas for the flow of your GUIs
3. The server must be able to support multiple clients using it at one time

4. The system must include a database which is populated with the records as explained above
5. Your design must closely follow the principles of low coupling and high cohesion

Possible Bonus Marks (Up to 20):

NOTE: The bonus features MUST ONLY be attempted if all the requirements of the project has already been satisfied. NO marks will be given to bonus features if the must-have requirements are not completed.

1. Add a chatroom to each course for students and professors to post messages **(10 marks)**
2. Deploy the project; run the server and client on separate machines **(5 marks)**
3. Writing JUNIT tests for at least 2 classes **(5 marks)**

If you would like to add another feature instead of the ones listed above, let us know! You can get bonus marks for your creativity! Think of a feature to add and you can get some bonus marks depending on the complexity of the future. **Note: it is not possible to get more than 20 bonus marks.**

Scope of Implementation for Milestone II (Iteration a)

- Development of login GUI
- Development of professor GUI **without requirements d and e** of the professor GUI (i.e. These two requirements do not need to be implemented in Milestone II:
    d) Send an E-Mail to all students enrolled in the course (these email addresses are stored in the systems database)
    e) View and grade student submissions in a dropbox (there should be one dropbox per assignment)

All other requirements is to be completed by the end of milestone III (Iteration b).

What to submit:

Milestone 1 – Design (20 marks):

1. Submit your complete "Design Class Diagrams" in a pdf file on D2L by the due date mentioned above
2. You must clearly demonstrate which classes belong to which packages in your class diagram.

Important Hint: Depending on your design, it may be more appropriate to have several class diagrams:

- One high-level class diagram that shows all packages and the classes within those packages as well as their interactions with other packages. This diagram will give the full design of your system at one glance.
- Several low-level class diagrams for each package which show more details about the classes inside each package.

NOTE: Your lower level class diagrams must include all attributes and important operations (i.e. not getters and setters) as well as cardinality

Milestone 2 – Implementation (35 marks):

1. Export your project into a directory (by right clicking on the project, selecting 'Export'), then make a zipped file from this saved directory and submit it on D2L.

   Note: Make sure the directory contains all files and directory in your zipped file (test it). If files are missing or corrupted, your project may be returned unmarked.

2. If you are working with one or two partners, create a cover page (PDF format) with your name and your partner name and submit it the D2L. Also, make sure to have all names included as comments in your source files.

3. **Demo your project on the date and time mentioned above**

Milestone 3 – Implementation Cont'd (45 marks and up to 20 bonus marks):

1. Export your project into a directory (by right clicking on the project, selecting 'Export'), then make a zipped file from this saved directory and submit it on D2L.

   Note: Make sure the directory contains all files and directory in your zipped file (test it). If files are missing or corrupted, your project may be returned unmarked.

2. If you are working with one or two partners, create a cover page (PDF format) with your name and your partner name and submit it the D2L. Also, make sure to have all names included as comments in your source files.

3. Submit all HTML files for Javadoc (NOTE: 10 marks are allocated to having proper Javadoc)

4. **Demo your project on the date and time mentioned above**