1. Overview

   Learn Enterprise OO development key concepts by implementing a portion of well understood application such as CampusConnect, D2L, CDM Intranet, or anything that you are well versed with on requirements.

2. Requirements
   a. For the number of people in your group, select 1 major functionality.
      i. Refer to rubric for technical functionality requirements
3. Documentation section areas (README.md)
   a. Functionality overview
   b. Discussion of how your design met the requirements
   c. Sequence of major functionality
   d. Lessons learned
   e. Decision log and consequence of the decision
4. What to turn in each milestone
   a. Group Submission – link to github
   b. Response as Issue
5. See sample from DePaul-KenYu/se452-project-demo

6. Recommended milestones

   *** Upload project document with reference to github for EACH milestone

   i. Milestone 1:  Focus area: Project scope and Prototype

      Deliverable:

      - Document:
        - Project scope area
        - Initial key features for each of the members of the team
        - Github location
        - How conflicts will be resolved
        - Communication mechanism during the project
        - Meeting journal
        - Decision made (eg editor, project scope)
        - Screen shot of working code (clean compile) in Appendix.
      - Code: Skeleton code from Spring uploaded to github, each person commit and update readme.md with their own information

   ii. Milestone 2 Focus area: Persistence Related

      Deliverable:

      - Document:
        - Update prior content as necessary (eg scope, tables..), lessons learned, decisions made
        - Add SQL based persistence
        - Add screen shot of working code in Appendix.
      - Code: Updated with persistence related code

   iii. Milestone 3 Focus area: Non Persistence Related

      Deliverable:

      - Document:
        - Update prior content as necessary (eg scope, controller..), lessons learned, decisions made
        - Add Service or UI Layer
      - Code:  Updated with services related code

   iv. Final Submission

      Deliverable:

      - Document:
        - Update prior content as necessary (eg scope, tables..), lessons learned, decisions made
      - Code: Updated to include restful services with persistence code

## Grading Rubrics

| Section pt range | 5 | 3-4 | 2 | 0-1 |
|---|---|---|---|---|
| **Group Milestone**<br>**(3 of these so total of 15 points)** | Code and document with feedback incorporated | Code and document without feedback incorporated | Document or code within code of milestone | No submission minimal submission or repeat of prior milestone |
| Section pt range | 11-15 | 5-10 | 3-4 | 0-2 |
| **Group Code** | Single code branch for all the project code, CI-CD to remote environment with shared configuration such as sql and nosql repository | Single code repository, build, deployment process on a CI-CD build that can be deployed to remote environment… | Some cohesion including single code repository, build, deployment process on a local build but differences in code standard adherence…<br>Definition of who is working on what in the main HELP.md | Frankenstein of concept, presentation, documentation, coding standards and version control in main branch |
| Section pt range | 31-35 | 21-30 | 10-20 | 0-9 |
| **Individual - Persistence** | Add feature not discussed during class demos, eg additional finders, combination of SQL and NoSQL<br>NEED to be documented in the package-info.java to get credit. If I need to guess, you will NOT get those points. | CRUD and unit tests with 2 repository | Mainly CRUD functionality with 1 repository | Limited functionality or compilation error |
| Section pt range | 31-35 | 21-30 | 10-20 | 0-9 |
| **Individual – Non Persistence related** | Additional feature not discussed in class demo, eg beyond logging.<br>NEED to be documented in the package-info.java to get credit. If I need to guess, you will NOT get those points | Additional logic between repository and service eg logging… | One for one between repository and service definition | Limited functionality or compilation error |