**Group Size** – 6

**Submission** – Singl zip folder with naming pattern <lastname>\_<firstname>\_<usi>\_<group-name>.zip

1. Group Contribution Report.xlsx.
2. Complete source code project folder.
3. Design.pdf – containing all product documentation.

**1. Problem Statement**

When grading group assessments, lecturers must manually ensure that all group members receive a consistent comment and score. However, this is error prone especially since based on the group contribution report, not all members receive the same score. In other cases, the lecturer may fail to enter the grade for one of the group members. Yet, in other cases, one student may appear in multiple groups.

To help alleviate these and other problems related to grading group assessments, the lecturers will like an application to be built that does the following –

1. Loads a group contribution report in the format of an .xlsx file.
2. Load the assessment structure.
3. Allow a score to be assigned for each section of the assessment structure and have those automatically applied to all group members. However, for the total score, the value should be adjusted based on the “percent” column of the group contribution report.

1.1 Functional Requirements

1. The system shall allow the user to manage (search, create, read, update, and delete) courses.
2. The system shall allow the user to manage (search, create, read, update, and delete) assessment structure.
3. The system shall allow users to group assessment structures under courses.
4. The system shall allow the user to import a group contribution report.
5. The system shall allow the user to manage (create and edit) the scores for the group based on the assessment.
6. The system shall allow the user to manage (create and edit) a comment for each group based on the marking of the assessment.
7. The system shall allow the user to view all scores that were entered.
8. The system shall allow the user to export the entered scores based on the assessment structure template sample.
9. The system shall allow the user to import and previously exported score sheet and continue editing.

1.2 Non-Functional Requirements

1. Object-Oriented modelling must be applied to object at the data layer, that is, relational modelling must not be used.
2. The system must be implemented in JavaFX or PHP.
3. The MVC must be used as the overall architecture.

**2. Tasks**

1. Create a use case diagram and fully dresses use case narratives based on the requirements.
2. Produce a domain level class model.
3. Create a business level view of the operation of the system using an activity diagram. This diagram should be clear and easy to understand. Using notes in the diagram to highlight which requirements are met by which sections of the diagram.
4. Produce a fully detailed, refined design level class model.
5. **Create an application that addresses the problem statement including all requirements.**
   1. **The pseudocode may be used in the method bodies. If done this way, these should be entered as comments to avoid syntax errors.**

**3. Note**

1. Your OOA level artifacts need to be somewhat consistent with your OOD artifacts.
2. Your OOD level artifacts need to be identical with your OO Implementation. In cases where this does not occur, it should be highlighted in both the design and implementation using hashtags as codes to link both sides.