Project: ModuLight

ModuLight is a lightweight application for professors to manage and perform statistical analysis on the educational records of students. Given below are my contributions to the project.

- **New Features**
  - `addComp` - allows users to add a graded component to the app.
    - **Justification**: Professors will need to add graded components into the system to track associated student scores.
  - `editComp` : allows users to edit a graded component in the app.
    - **Justification**: Professors may want to edit the name, maximum mark, or weightage of a component as the semester progresses.
    - **Highlights**: Implementing this feature to automatically update all calculations and display it dynamically in the UI was challenging.
  - `deleteComp` : allows users to delete a graded component in the app.
    - **Justification**: Professors will need to add graded components into the system to track associated student scores.
- **Code contributed**
  My work was mostly foundational and/or related to the UI display.
  - Major highlights:
    - Created and modified Model-related methods to fit our own entities (`StudentBook` , `GradedComponentBook` , `StudentScoreBook` and `ModelManager`)
    - Modified the UI and JavaFX code (`StudentCard` , `GradedComponentCard` , `StudentScoreCard` and other FXML files) to accommodate three entities and dynamically update the UI whenever changes are made.
    - Modified the class infrastructure to link `Student` , `GradedComponent` , `StudentScore` and allow calculation of overall/mean scores.
  - Testing related to add/edit/delete gradedComponents commands.
  - For more details, refer to the RepoSense Link

- **Project management**:
  - As group leader, delegated tasks weekly to members.
  - Largely managed the issue tracker, creating/closing issues and milestones from v.1.2 to v.1.4.
  - Created demo version of Modulight for CS2101, with significant additional features including unused graphing feature.

- **Contributions to User Guide**:
  - Documented `addComp` , `editComp` , and `deleteComp`
  - Added parameter information for graded components.
  - Added notes about score calculation and some FAQs.
- **Contributions to Developer Guide**:
  - Documented implementation of the UI component
  - Documented sequence and activity diagrams of `addComp` , `editComp` , `deleteComp` commands.
  - Added a use case for deleting a graded component.