

Zhuang Jianning's Project Portfolio Page

Project: ModMan

Short for Module Manager, ModMan is a desktop app designed to help teaching assistants manage their module(s). It is optimized for use via a Command Line Interface (CLI). ModMan helps to track module details as well as students' progress and data, all in one platform. It can also perform autograding for MCQ assignments.

Given below are my contributions to the project.

- **New Feature:** Added the ability to set deadlines for assignments and sort them by their deadlines.
 - What it does: Allows Teaching Assistants to set their own deadlines to grade assignments. Also allows them to sort the assignments in the module based on the deadline that they set.
 - Justification: Being able to set your own deadlines to grade assignments and sort them based on their urgency is crucial to help Teaching Assistants plan their time wisely and manage their workload.
 - Highlights: The Assignment Class implements the Comparable Interface to allow the natural ordering of assignments to change based on the compareTo() function. The sorting is also stable and accounts for assignments with null deadlines. If an assignment does not have a deadline, it would be sorted after other assignments with deadlines.
- **New Feature:** Added the ability to load and store answers for assignments.
 - What it does: Reads the answer key stored in a text file in the answers directory and stores it as an Answer object for the assignment.
 - Justification: Being able to load and store answers allow Teaching Assistants to easily retrieve and view the answer key to assignments. Answers can also be accessed offline.
- **New Feature:** Added the ability to autograde assignments.
 - What it does: All students' scripts found in the scripts directory will be graded against the answer key. The grade for each student will also be automatically saved and updated. Autograding also keeps track of which students have not submitted their assignment.

- Justification: Being able to autograde assignments would drastically help Teaching Assistants manage their workload. Autograding is also much faster and accurate than manual grading.
- Highlights: Autograding is a relatively complicated feature involving many components and classes.
- **Code contributed:** [RepoSense Link](#).
- **Project management:**
 - Set up GitHub team organisation and repo
 - Resolved some issues with Continuous Integration on GitHub
 - Managed releases v1.0 - v2.1 (3 releases) on GitHub
- **Enhancements to existing features:**
 - Caught Exceptions and Added JUnit Tests for the Parser Class: #52, #59
 - Implemented methods in the Storage Class to load and save students' scripts and answer keys from the respective directories: #120, #121
 - Improve Ui features: #6, #72
- **Documentation:**
 - User Guide:
 - Added documentation for the feature `list assignments` : #86
 - Added documentation for the feature `set assignment deadline` : #252
 - Added documentation for the feature `sort assignments by deadline` : #252
 - Added documentation for the feature `autograde assignment` : #257
 - Added documentation for the feature `view assignment answer` : #253
 - Added documentation for the feature `view student script` : #253
 - Developer Guide:
 - Added Class Diagrams for the Logic Component: #111
 - Added implementation details for the feature `set assignment deadline` : #142
 - Added implementation details for the feature `sort assignments by deadline` : #142
 - Added implementation details for the feature `autograde assignment` : #208
- **Community:**
 - PRs reviewed (with non-trivial review comments): #65
 - Reported bugs in other team's applications