

## Java Grading System

The challenge at hand was to software that can help a professor organize the grades of the students enrolled in her course at the end of the semester. The focus of this project was to understand the end users pain points and provide a robust solution. The professor's existing system was Microsoft Excel which she used primarily because of the flexibility it provides. Our motivation was to develop the software that addresses the users pain points and acknowledges the benefits of the current system she uses and to recreate those in our software. The entire development phase was done using the Agile principles.

Our approach began by a thorough requirement analysis of the end users needs. We interviewed the end users and gathered key insight about the users pain points with her current system and her expectations with the software. We also acknowledged the services her current system provides due to which she has been using her current system. The end user was particularly fond about the flexibility that her current system provided. She mentioned that the biggest complaint about her current system is the lack of reusability. We moved onto the Planning stage by determining a comprehensive list of functionalities our software would provide to the user which addressed her pain points and recreated the benefits of her current system. We labelled each of our proposed functionality as “Non Negotiables”, “Good to Have” and “Additional features”. We also prepared for the fact that the users requirement could change going forward. As part of our architectural designing phase, we discussed our plan with the end user to get her feedback and suggestions.

While developing, we decided to work on three different parts of the problem. We used a modular approach and focussed on integration later on. We started developing the MySQL database and the APIs to fetch values from the database, The controller logic that integrates the backend with the front end, and the User Interface. For the local MySQL server, we used MAMP. For our User Interface, we used Java Swing. Our entire software was based on Object oriented principles of Abstraction and Encapsulation. Our aim was to hide the complexity of the software and make it a smooth user experience. We also focused on encapsulating data and methods into classes which could be reused for another project.

To address the main pain point of reusability, we decided to create a “Template” which can be reused by the user again for future courses. She has the option to reuse the template completely, change components in it or create a new template altogether. Our software provided the functionality of adding multiple courses at the same time on the same user panel, the option to switch between courses at will and have multiple batches within courses. The user could add and remove students to cater for the situation where a student drops the course mid semester. We also provide the flexibility to define the rubrics of the templates and flexibility to enter grades as numbers or alphabets. We added built in statistical tools for determining the class average and curving grades. As an additional feature that we thought the system should have, we gave the user the option to add bonus points for a student on an assignment and leave comments.

Our entire project was developed using an AGILE methodology. We iteratively visited the requirements and the created prototypes to implement features iteratively. The key takeaway from this project was to create software/ programs that are focused on the end user. A technical challenge that we faced quite often was to integrate the modules we were creating. We worked around this by creating proper design documentations that helps the team

members understand the modules we were working on. We observed that the primary challenge while creating the software is to clearly understand the users needs and requirements and to be able to address those pain points in an efficient and robust way.