

Final Project Documentation

CIS152 Data Structures

Caden Black

11/22/2024

## **Proposal**

Currently efficient management of workers is important for companies, from scheduling to task management, many organizations struggle with optimizing their workforce. This project aims to create a Scheduling System that uses queues & linked lists. Queues will be used to give priority to the task & a necessary skill level to certain tasks along with how many people are needed to complete it. Linked Lists will be used to store information about a coworker's skill level. By implementing a Selection Sort algorithm, the system will efficiently organize the coworkers based on their skill level and time worked so far.

- Sprint 1 - Classes for Tasks & coworker's along with tests to make sure queues & linked lists are implemented properly to make sure a task is filled with coworkers based on their skill level.
- Sprint 2 - (If I am allowed to) I'd like to use Python Tkinter to create/design a GUI system for this.
- Sprint 3 - Add functionality within the GUI to implement queues/linked lists for tasks/coworkers and have the GUI display a finalized schedule for the coworkers.
- Sprint 4 - Make sure everything is functional and working as intended.

## **Time/Change Log**

- 10/15/2024: Final Project program is created in Python and coding begins.
- 10/25/2024: Main classes are defined and coded into the program. Queue system is implemented. Linked Lists are implemented. Tasks class is implemented with Skill Level, priority, and workers needed. Workers class is implemented with Worker Id, Skill Level, and Time Worked. The selection sort is also implemented. Preparations are made to begin coding the main GUI.
- 11/09/2024: Main GUI application is completed. The GUI contains the ability to create and append workers. The GUI also allows the user to create tasks with a given priority and needed workers/skill level. Finally the selection sort is used within the GUI to assign workers to a selected created class.
- 11/15/2024: Bug fixes are made to the overall program. Some minor fixes like GUI buttons not working depending on some invalid inputs. Overall input validation is fixed. Some minor queue problems were fixed with workers accidentally being selected if they had worked over 40 hours.
- 11/22/2024: Color is added to the GUI application to make it look nicer. GUI size and framing is adjusted to look nicer. Color is primarily added to the backgrounds and buttons to make the application not look as bland.

## **Lessons Learned**

The scope of my project was minimal. My goal was to make a management system for workers and overall I was successful in this implementation. Throughout the project I had no major blockers and was able to work at a steady pace to meet all the requirements and finish the application on time.

## **Conclusion**

Overall, my final project was a success. MERUSE was used by having comments included throughout the program to explain functionality of the code. The structure of the code was also nice and neat allowing users to easily access certain segments of it and understand what is going on.

My project aimed to create a task management system and overall that was a success. The user is able to create and append worker information into the program and can create tasks as well with a given priority. Users are also able to assign the workers to a given task if the given skill level & time worked requirements are met.

Future versions may consist of persistent data. As of now the program does not store data in a database and every time you boot up the application you would need to recreate workers and tasks. Future versions would include a database file so when you create users and tasks they save in between sessions. Along with this future versions may include a “task completed” button for a given task assignment so that when a task is finished the user can click this button and remove the task fully from the list.