Concordia University

Software Engineering (BEng) – Co-op

Development Journal of C++ Project: HR Management System

Step-by-Step Thought Process

Djazy Faradj

Email: djazy.faradj@outlook.com GitHub: https://github.com/Djazy-Faradj

Undergraduate Student (Year 1, Term 1)

Project Start Date: September 23rd, 2024

Target End Date: October 7th, 2024

Project End Date: October 6th, 2024

Abstract

This journal will document the development process for my second individual project in the Software Engineering program, where I design and implement a console-based Human Resources (HR) management system for a small college. This application models departments, teachers, and staff and demonstrates object-oriented programming (OOP) principles, data handling, and customized error handling. The goal is to improve my OOP skills while applying them in a real-world context.

Introduction

This project involves the development of a Human Resources (HR) management system for a small college. The system is designed to model various departments, their staff, and teaching personnel, while incorporating key programming concepts such as object-oriented design and file handling. It will include department objects containing teachers (both full-time and part-time) and staff, with each department having a dean, who must be a teacher. The system will allow users to input and manage data through a console-based interface, storing information in text files for persistence between sessions. A primary feature of the project is a payroll computation system that calculates salaries for staff and teachers based on their roles, degree qualifications, and work hours. The project also focuses on robust error handling, with custom exceptions to prevent invalid or duplicate data entries. By leveraging inheritance, aggregation, polymorphism and interface implementation, this system serves as a comprehensive application of programming techniques learned throughout my C++ course. Ultimately, this project will provide practical experience in designing and implementing a functional software model for real-world HR management tasks. The primary objective of this project is to apply object-oriented programming (OOP) concepts such as inheritance, aggregation, and polymorphism to build a functional HR management system. A clear console-based interface will be developed for easy user interaction, and file input/output operations will be implemented to manage employee data effectively. Additionally, custom exception handling will be used to ensure robust error management. I will also write the program's "pseudocode" on Notepad++ and complete UML class diagrams for all classes and interfaces using Microsoft Visio. Finally, thorough documentation will be created to guide future users in navigating and using the system. This project will train all the fundamental aspects required in a software engineer's skill set.

In this project, I aim to:

1. Apply OOP concepts, such as inheritance, aggregation, and polymorphism.
2. Build a clear console-based interface for easy interaction with the HR system.
3. Implement file I/O to manage employee data.
4. Ensure error handling through custom exceptions.
5. Create documentation to assist future users of the system.

Initial Application Concept

The HR Management System will:

- Model department objects containing lists of teachers (both full-time and part-time) and staff members.
- Handle payroll calculations for each type of employee (full-time, part-time, or staff)
- Implement a user interface using a console-based interaction model to allow for simple input and output of data.
- Include error handling for attempts to add employees to nonexistent departments or duplicate entries.

For implementation details, feel free to read the README file!

Challenges and Reflections

Difficulty Encountered: User Interface Design

One of the main challenges I encountered during the development of this HR management system was designing a user-friendly console-based interface. While a GUI might have been more intuitive, I decided to stay with the console due to the scope of the project. However, organizing the menus and ensuring smooth interaction without overwhelming the user required a lot of trial and error.

1. Error Management:

Error management became tedious as the project grew. Adding error handling for every possible user input or scenario felt repetitive. If I were to redo the project, I would aim to create a more modular approach to handle these exceptions.

2. Code Structure:

I realized that splitting the long code into multiple files (e.g., separating class definitions, payroll, and department functions) would have made the project easier to maintain and debug. The current implementation had all code in one large file, which made it harder to navigate as I progressed. If I were to revisit this project, I would refactor the code to increase readability and scalability.

3. Pseudocode and Development Adjustments

When I started the project, I wrote down pseudocode to outline the key functions and logic I intended to implement. However, as I delved deeper into the development, I realized

that certain functions and features were missing from my initial pseudocode. For example, I had not planned for the detailed error handling routines or certain helper functions that emerged as necessary later on.

As I progressed, I added these functions directly into the code without updating the pseudocode, as it felt more efficient to continue coding rather than going back to rewrite the design. In retrospect, keeping the pseudocode updated would have provided a clearer blueprint, but at the time, I prioritized getting the project functional.

4. Lack of Comments

I started with diligent commenting of the code but stopped after a few days, which made it harder to return to certain parts of the project. Next time, I will try to maintain consistent documentation throughout, especially for complex sections.

Learning Outcomes

1. Object-Oriented Programming (OOP):

This project significantly improved my understanding of OOP principles. I used inheritance, polymorphism, and aggregation extensively. The class hierarchy for Person, Teacher, and Staff allowed for clear structure and implementation of shared behaviors. I am much more confident in creating and organizing class structures to reflect real-world relationships.

2. Pointers and Memory Management:

Another major learning curve was pointers. I had limited exposure to pointers before this project, but their application in dynamic memory management, especially for objects like teachers and departments, deepened my understanding. I also learned the importance of ensuring proper cleanup and avoiding memory leaks.

3. UML Diagrams:

Practicing with UML diagrams helped visualize the architecture before diving into the code. While I initially found it challenging to model everything correctly, this exercise ultimately contributed to a more structured design process.

Personal Reflections

1. Confidence in Task Completion:

Completing this project made me more confident in my ability to set goals and meet them. I managed to finish the project a day ahead of schedule due to disciplined time management, which involved setting clear milestones and sticking to them.

2. Time Management:

Effective time management played a huge role in completing the project early. By breaking down tasks and setting weekly objectives, I was able to avoid last-minute rushes and unexpected bottlenecks.

3. Deeper Knowledge of C++:

Beyond OOP and pointers, this project also broadened my understanding of C++'s more advanced features like file handling, exception management, and STL containers. I feel I have a stronger grasp of how to use C++ effectively in real-world applications.

Final Thoughts

This HR management system project was a great learning experience. I was able to apply several theoretical concepts from class to a real-world-like scenario, and this helped deepen my understanding. There is still room for improvement, particularly in code structure and error management, but overall, I am proud of the final product and the learning journey I went through.