Employee Management Application

21:198:335:02 Data Structures & Algorithms
Data Structures Final Project

Zeyad Rashed
Rutgers ID: 199009651
Professor Weiping (Veronica) Zhang

December 6, 2024

Abstract

This document describes the design, implementation, and analysis of the Employee Management Application. The project demonstrates key data structures and algorithms concepts, including sorting algorithms, method overloading, file I/O, and GUI design.

1 Introduction

The Employee Management Application was developed as part of the Data Structures & Algorithms course final project. This project incorporates fundamental principles discussed in *Data Structures and Algorithm Analysis in Java, Third Edition* by Clifford A. Shaffer. The application meets the following requirements:

- Management of employee records, including attributes such as ID, name, salary, and department.
- Multiple sorting options, including Bubble Sort, Heap Sort, and Comparator-based sorting.
- A JavaFX-based graphical user interface for seamless interaction.
- Performance measurement for sorting operations.

1.1 Graphical User Interface

The GUI was developed using JavaFX, enabling user-friendly interaction. Features such as toggling between different algorithms and different sorting views are included.

2 Application Views

2.1 Application Overview

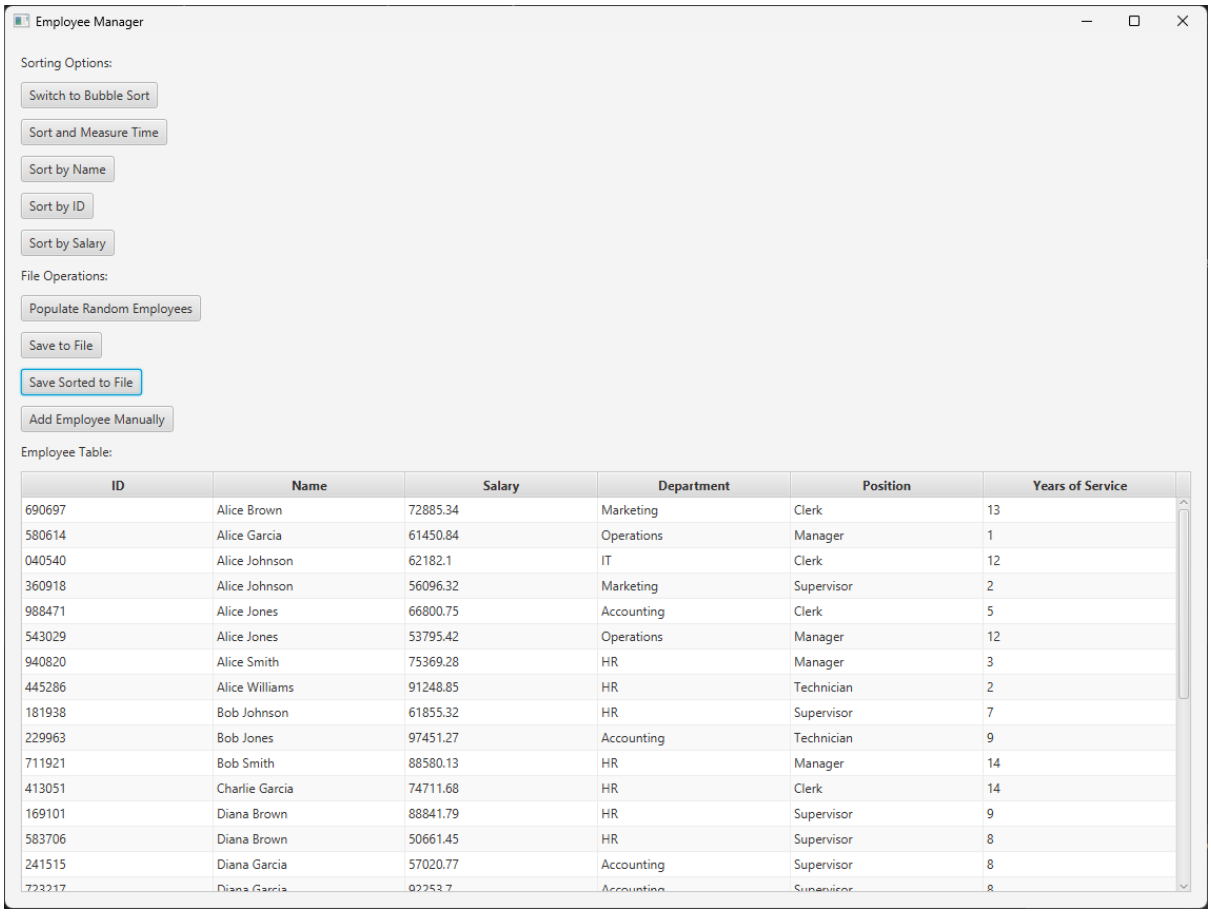

Figure 1: Main view of the Employee Management Application.

2.2 Generating Random Employees

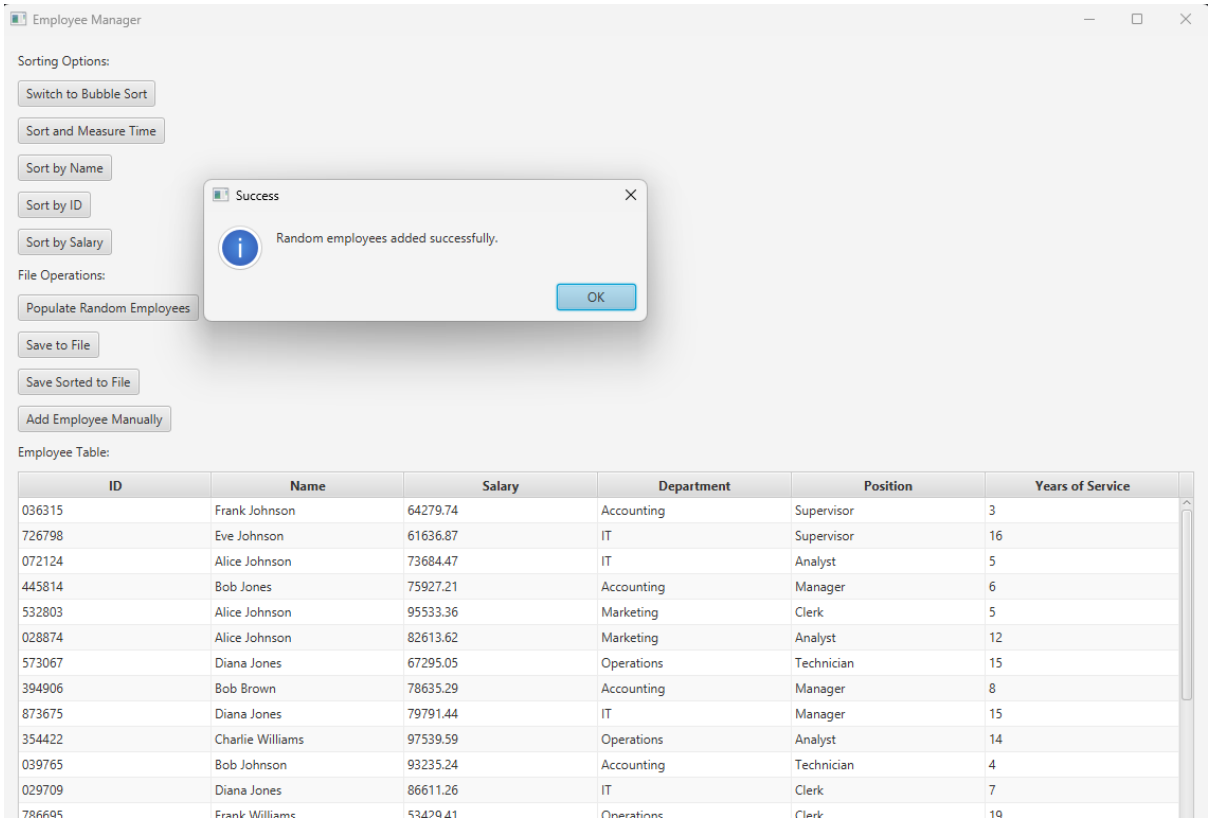

Figure 2: Populating the table with random employee records.

2.3 Sorting by Name and Performance Readout

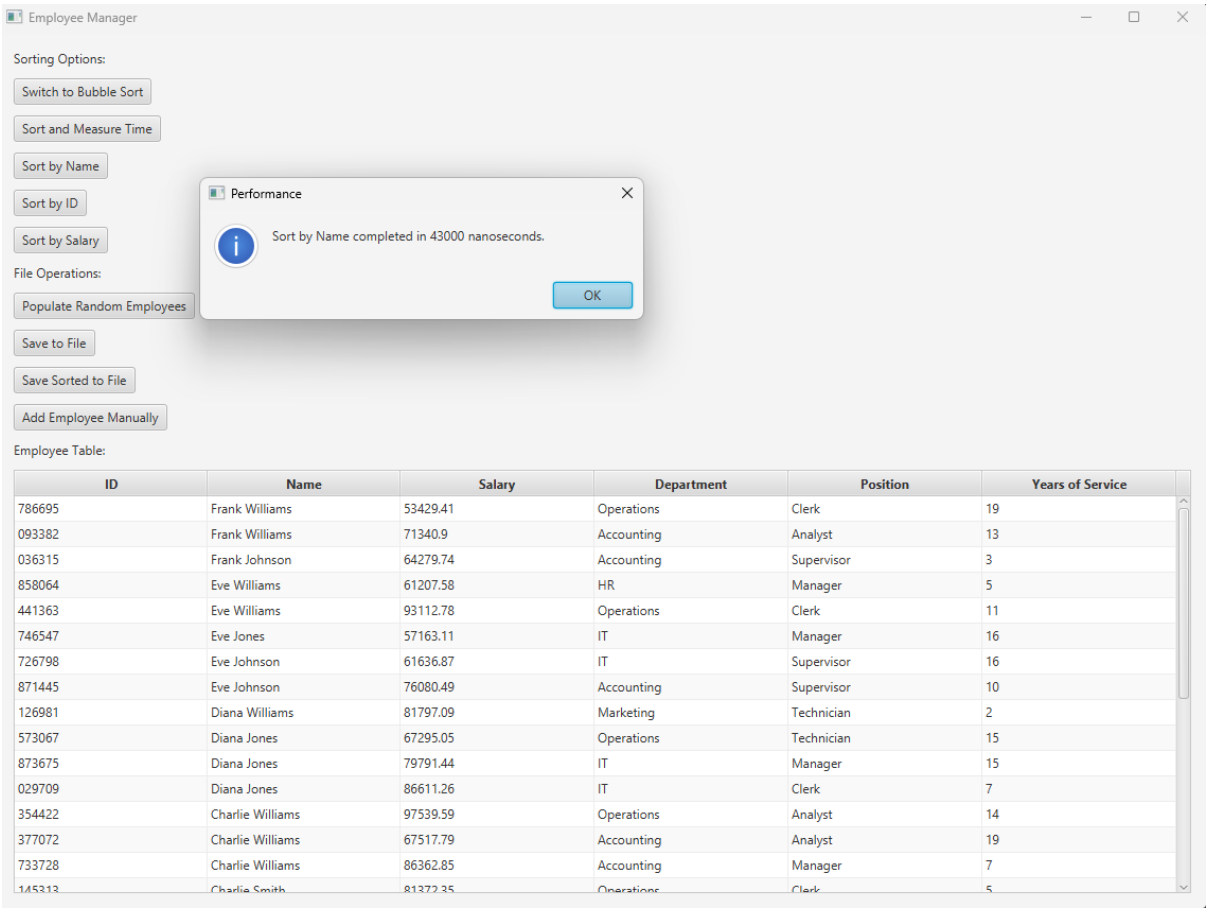

Figure 3: Sorting employees by name and displaying the performance of the sort. (Bubble sort)

2.4 Saving Sorted Data to File

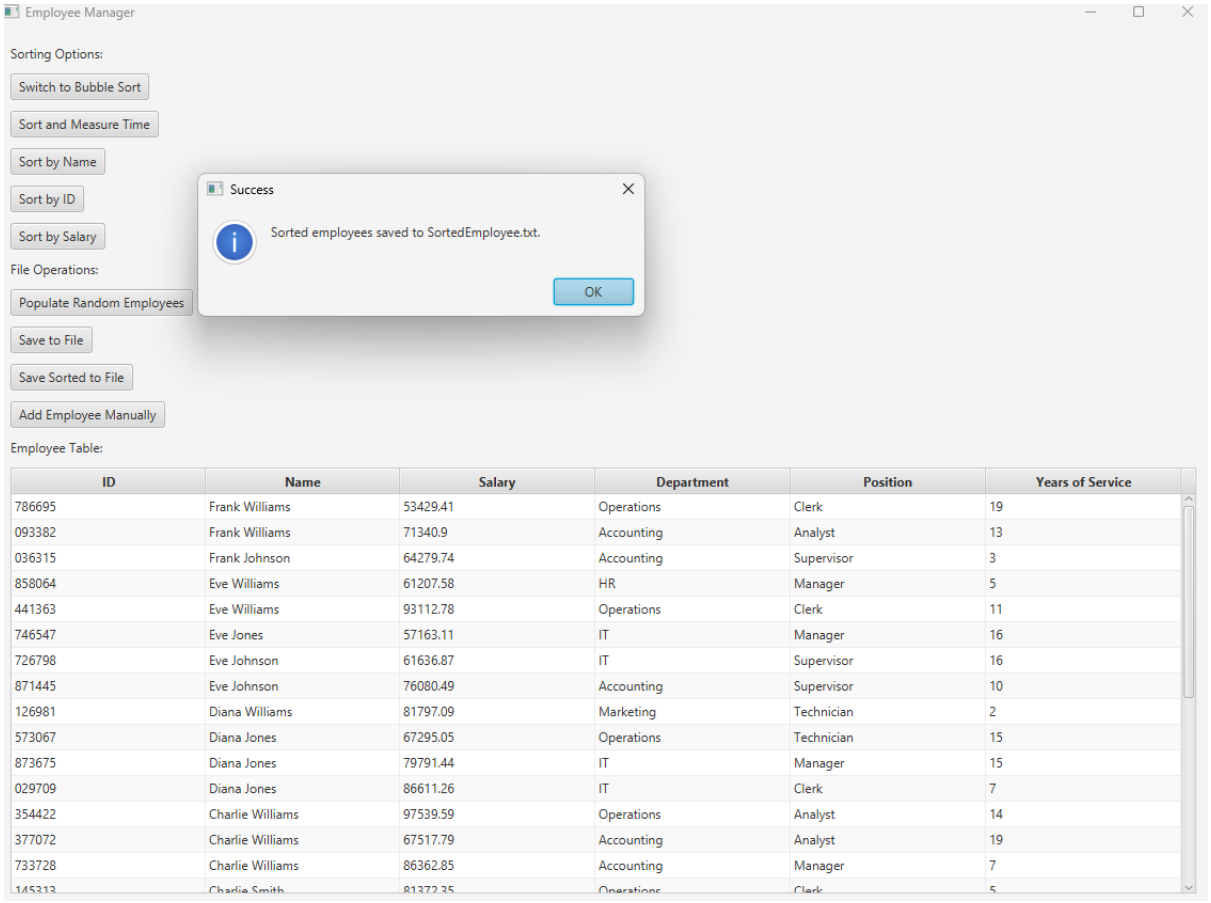

Figure 4: Saving the sorted employee list to a file.

2.5 Switching Sorting Algorithms

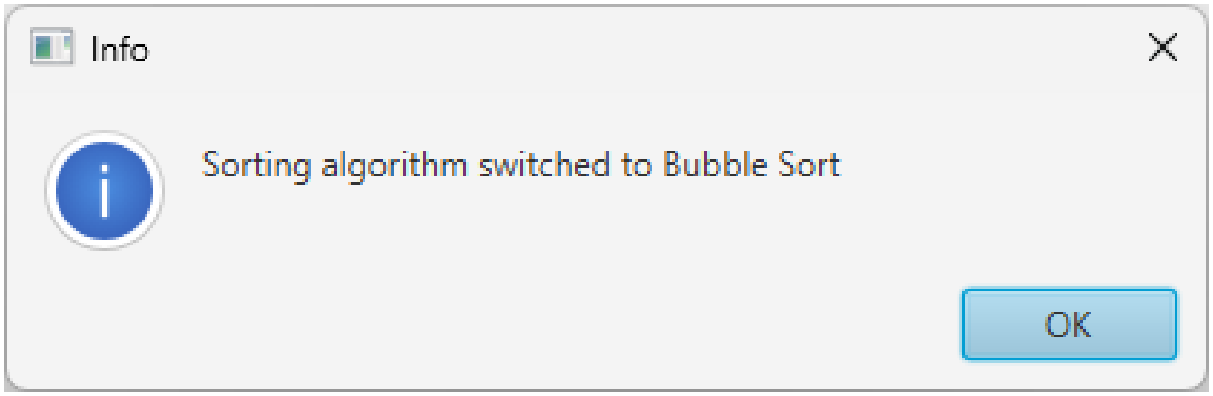

Figure 5: Switching the sorting algorithm to Bubble Sort.

3 Design Philosophy

The project adheres to Shaffer's principle of balancing trade-offs between space and time complexity. Emphasis was placed on modularity, scalability, and maintainability. By encapsulating sorting logic in a dedicated class and delegating file I/O operations, the project follows the abstraction principles outlined in Shaffer's text.

4 Key Features and Implementation

4.1 Sorting Algorithms

Sorting is a fundamental operation in computer science. The application employs both $O(n^2)$ (Bubble Sort) and $O(n \log n)$ (Heap Sort) algorithms, allowing users to toggle between them dynamically.

```
1        /** Bubble Sort implementation for sorting employee lists
             */
2        public static <T> void bubbleSort(List<T> items,
             Comparator<T> comparator) {
3            for (int i = 0; i < items.size() - 1; i++) {
4                for (int j = 0; j < items.size() - i - 1; j++) {
5                    if (comparator.compare(items.get(j), items.
                         get(j + 1)) > 0) {
6                        Collections.swap(items, j, j + 1);
7                    }
8                }
9            }
10       }
```

4.2 Performance Measurement

The project tracks sorting times to demonstrate efficiency differences. Results are displayed in the GUI to provide feedback.

```
1         /** Measures performance of the current sorting algorithm
              */
2         private void sortAndMeasurePerformance(Comparator<
             Employee> comparator) {
3             long startTime = System.nanoTime();
4             if (useHeapSort) {
5                 StackSorter.heapSort(employees, comparator);
6             } else {
7                 StackSorter.bubbleSort(employees, comparator);
8             }
9             long duration = System.nanoTime() - startTime;
10            displayArea.setText(formatEmployeeList());
11            showAlert("Sorting Performance", "Time taken: " +
                 duration + " ns", Alert.AlertType.INFORMATION);
12        }
```

5 Diagrams

5.1 UML Diagram

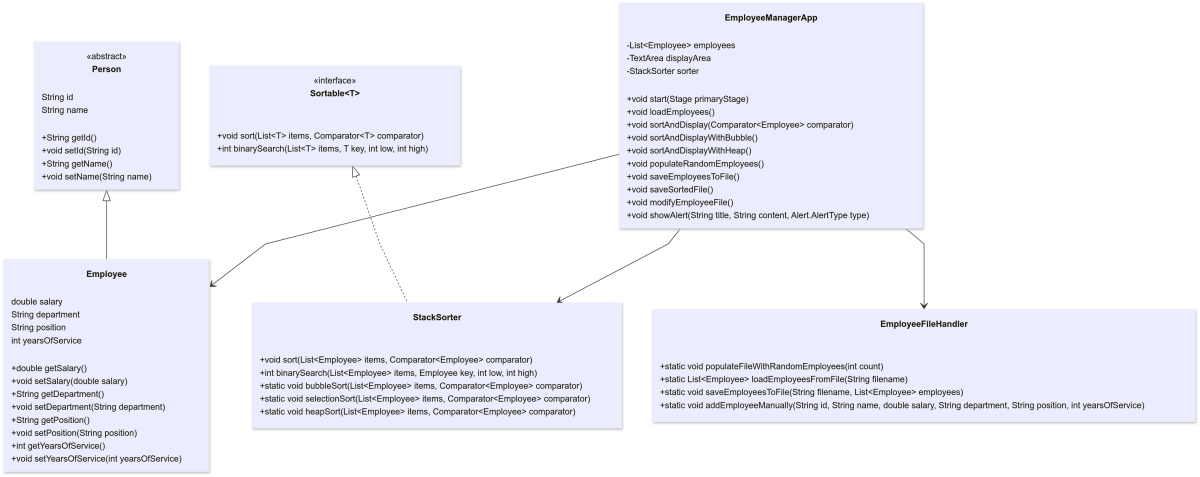

Figure 6: UML Diagram of the Employee Management Application

5.2 Flowchart

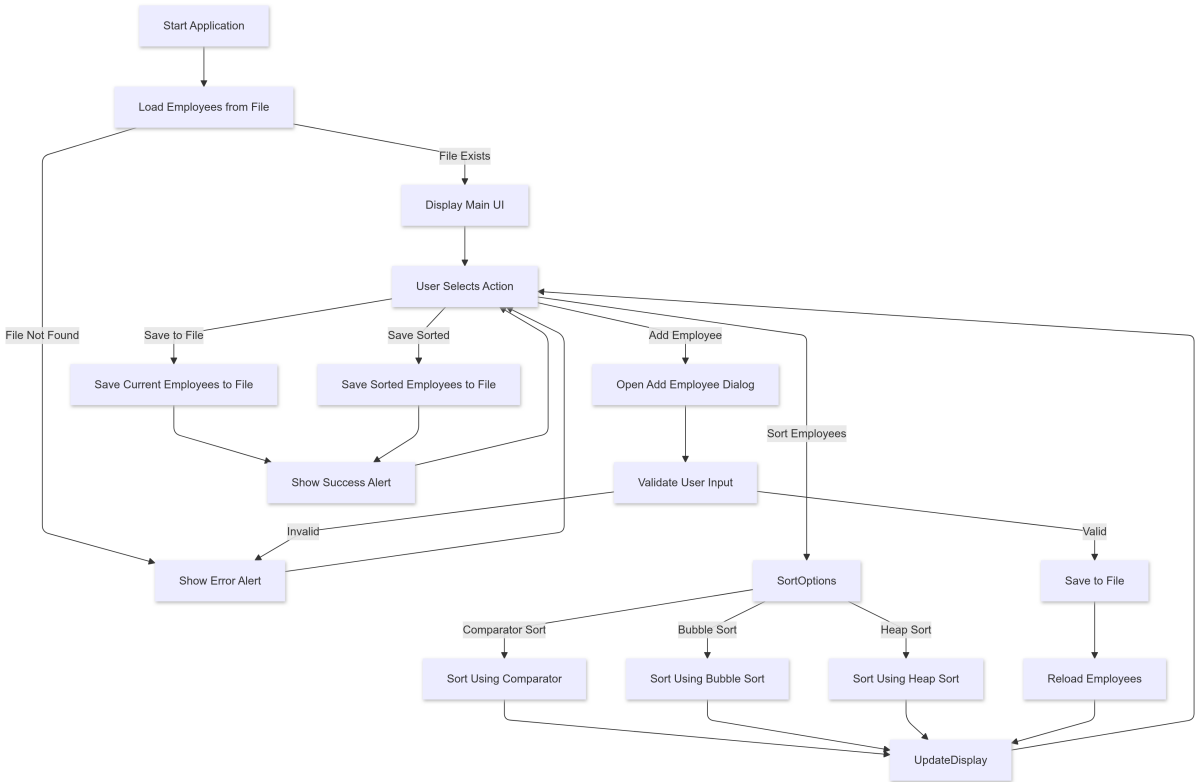

Figure 7: Flowchart of Application Workflow

6 Conclusion

The Employee Management Application showcases the practical application of data structures and algorithmic principles. By integrating theoretical knowledge with real-world implementation, the project exemplifies efficient, maintainable, and user-focused software design.

7 References

1. Shaffer, C. A. (2013). *Data Structures and Algorithm Analysis in Java, Third Edition.*
2. JavaFX Documentation: `https://openjfx.io/javadoc/23/`
3. Java Collections Framework: `https://docs.oracle.com/javase/8/docs/technotes/guides/collections/overview.html`