Mahidol University
International College

EGCI211 Advanced Computer Programming

Final Project Report

(Employee Management System)

Group 10

Achira Pornputtichai 6680996
Chaiyapat Triyanam 6680730
Phurilap Kitlertpaisan 6680251

Mahidol University International College
28 March 2025

I. Project Background

The aim of this employee management system is to improve to sdg8 (Decent Work and Economic Growth). The program manages key details of the employee such as name, salary and experience in order to have a clear record and get fair treatment.

II. Program Usage

```
@Phurilap ➜/workspaces/Final-Project (main) $ ./final "John" 5 50000 "Alice" 3 45000 "Bob" 7 60000
Employee John created
Employee Alice created
Employee Bob created

Employees BST order:
Name: Alice
Salary: 45000 Baht
Experience 3 Years
Name: Bob
Salary: 60000 Baht
Experience 7 Years
Name: John
Salary: 50000 Baht
Experience 5 Years

Employees original order:
Name: John
Salary: 50000 Baht
Experience 5 Years
Name: Alice
Salary: 45000 Baht
Experience 3 Years
Name: Bob
Salary: 60000 Baht
Experience 7 Years

Employees sorted by salary:
Name: Alice
Salary: 45000 Baht
Experience 3 Years
Name: John
Salary: 50000 Baht
Experience 5 Years
Name: Bob
Salary: 60000 Baht
Experience 7 Years

Updating experience and salary for John
Name: John
Salary: 54500 Baht
Experience 6 Years

Found via BST search:
Name: John
Salary: 54500 Baht
Experience 6 Years
```

```
Processing candidate queue:
Processed: John
Processed: Alice
Processed: Bob
Employee Alice removed
Employee John removed
Employee Bob removed
```

- The program uses command-line arguments through argc/argv. It will store each employee data as triplets(name, exp, salary). It will then be process through 3 data structures(array, BST, queue)
- When we run **./final "John" 5 50000 "Alice" 3 45000 "Bob" 7 60000,** each employee has 3 data, now this will create 3 employees.
- It will now display:
    - Display in BST order(InOrder)
    - Original Array(insertion order)
    - Sorted by salary: Uses **bubbleSort** to sort the employees array first before calling **displayEmployeesArray** to display the sorted array.
    - Updating Employee Data: getEmployee gets the second employee and increase salary by 4500
    - BST search: searches for argv[1](name) in the BST.
    - Queue Processing: **hasCandidates** function is called which calls the dequeue function and removes each employee from the array in FIFO order. It continues until there are no more candidates left.

III. Requirements

A. Array & Pointers:

The program uses an array of pointers to manage employee data, with a capacity set to 10. The pointers are pointing to the employees which means they have direct access to the data inside each employee. The Company constructor uses the new **Employee*[capacity]**. And for the destructor, it will first delete all the employees in the array and then the array itself afterwards to ensure complete deletion and prevent memory leaking.

B. Overloading operator:

There are 2 operator overloading in this program, operator++ and operator+=. The operator++ is used when we want to increase an employee's experience by 1 year. And for the operator+=, it is called when we want to increase an employee's salary by a specific amount. For example, if we want to increase Employee salary by **5678**, we do *emp+=**5678.**

C. Binary search tree:

The Binary Search Tree (BST) structure sorts the Employee objects based on their names.This ordering is crucial and is maintained during insertion.Which the insertNode method places new employees in their correct positions based on name comparisons. Efficient searching by using **findEmployee** function, when the BST's sorted nature to navigate the tree, branching left or right depending on name comparisons. To display the employees in alphabetical order, the **displayEmployees** method employs an in-order traversal. Finally, to prevent memory leaks, the **freeTree** method systematically deallocates all dynamically allocated nodes when the **CompanyBST** object is destroyed, ensuring proper memory management.

D. Sorting:

The employee array is sorted in ascending order of salary. To achieve this, the getSalary() function is used to compare the salaries of each employee. When a higher salary is encountered before a lower one, a temporary variable (temp), facilitates the swapping of their positions within the original array. This process ensures that employees with lower salaries are placed before those with higher salaries, directly modifying the array in place to reflect the sorted order.

E. Queue:

The Queue class manages job candidates using a First-In, First-Out (FIFO) approach. The enqueue() function adds a new candidate to the rear of the queue if there is available space, while the dequeue() function removes and returns the candidate at the front. The queue uses a circular structure, ensuring efficient use of memory by reusing freed-up spaces. The isEmpty() function checks if the queue has any candidates left. This system ensures a fair and organized hiring process by processing applications in the order they were received, preventing bias and maintaining transparency in recruitment.

IV. Limitation

Bubble sorting algorithm: As the data increases it may take a long time to sort as bubble sort compares only two adjacent elements at a time.

Display: Our programming displays everything all at once instead of having a menu for the user to used.This may create a confusion for the user.

Queue Implementation: The queue has a **fixed capacity**, meaning it cannot handle unlimited candidates unless dynamically resized. Since it follows **FIFO (First-In, First-Out)**, priority-based hiring cannot be implemented.

No Error Handling for Invalid Inputs: The program does not check for **negative salaries, incorrect data types, or missing information**, which could cause unexpected behavior.

V. Work distribution:

First we do a group discussion, searching for a suitable program for our sdg goal(sdg8). Once that is established, each member helps come up with features to add to the program. After the outline of the program is done, we divided the work equally throughout the group for efficiency. In the presentation part, we divided the presentation into sections. And each one of us will be responsible for a specific part of the presentation.

VI. Q&A Questions

This is a dynamic array of pointers to Employee objects, used to store candidates waiting in the queue.
It's used with variables like **front, rear**, and **count** to **manage the queue in a circular way** (so when it reaches the end, it loops back to the beginning).

The array in **company.h** is used to store and manage employees who are already hired, allowing updates, display, and sorting by salary. The queue in **queue.h** holds job candidates waiting to be hired

VI. References

GeeksforGeeks. (2025a, January 21). *Bubble Sort Algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/bubble-sort-algorithm

GeeksforGeeks. (2025b, March 11). *Tree traversal techniques*. GeeksforGeeks. https://www.geeksforgeeks.org/tree-traversals-inorder-preorder-and-postorder

Team, S. M. (2023, October 3). *Goal 8: Decent work and economic growth*. SDG Move. https://www.sdgmove.com/2016/10/06/goal-8-decent-work-and-economic-growth