Republic of the Philippines
**CEBU TECHNOLOGICAL UNIVERSITY**
DAANBANTAYAN CAMPUS
Agujo, Daanbantayan, Cebu, Philippines
Website: http://www.ctu.edu.ph E-mail: info-daanbantayan@ctu.edu.ph
Phone: +6332 437 8526 loc. 102

**COLLEGE OF TECHNOLOGY AND ENGINEERING**

# PROJECT DOCUMENTATION

## 1. Project Title:

- **"Hierarchical Agent Management System with Commission Tracking"**

## 2. Project Description:

- **Overview:** Hierarchical Agent Management System with Commission Tracking provides functionalities for adding, removing, and managing employees under a hierarchical structure. Each employee, including the manager, has specific roles, subordinates, and commission calculations based on the number of direct subordinates.

- **Objective:** The primary objective of this project is to streamline the management of employees in a call center by allowing the user to:
  - Add subordinates to employees in the hierarchy.
  - Remove employees from the hierarchy.
  - Search for employees and view their commission details.
  - Save and display the employee hierarchy.

- **Scope:**
  - Represent employees and their relationships in a tree structure.
  - Enable dynamic management of employees (hiring and removal) while maintaining the integrity of the hierarchy.
  - Compute and display commissions based on direct subordinates.
  - Save the current hierarchy to a file.

- **Limitations**:
  - Each employee can have a maximum of two direct subordinates.
  - Employees with existing subordinates cannot be removed until their subordinates are removed.
  - The hierarchy is saved in a simple text format without support for reloading or advanced data serialization.
  - The system assumes single-level commissions and does not support complex commission structures or inheritance of commissions across levels.

# 3. Features:

### A.) AGENT HIERARCHY MANAGEMENT

- Allows user to create the root agent, typically the manager, who acts as the top-level node in the hierarchy. It also supports hiring employees for any agent in the hierarchy the dynamically expanding the tree structure. Removal of employees is permitted, but only those without any subordinates can be removed.

### B.) DYNAMIC AGENT RELATIONSHIPS

- Each agent can hire one or two subordinates based on availability.
- Validates whether hiring slots are available and handles full capacity properly.

### C.) COMMISSION CALCULATION

- Automatically computes the commission of each employee, especially when an employee is hired by the specific employee.

### D.) EMPLOYEE SEARCHING

- In hiring and removing an employee, the search algorithm is applied for user-efficiency.

# 4. Technologies Used:

- Programming Languages: C Programming Language
- Frameworks/Libraries: stdio.h, stdlib.h, string.h
- Tools: VSCode, CodeBlocks
- Databases: None

# 5. Project Structure:

```
├── main.exe                    = executable file of the program
   ├── main.c                   = main program
├── README.md                   = project documentation
└── HierarchicalStructure.txt   = data storage file
```

# 6. Usage:

First, you are going to enter a name and an ID number for the manager.

```
Enter name for the Manager: Zea
Enter ID for the Manager: #143
```

After entering the manager's name and ID number, the program would display the menu and let the user choose according to the menu provided.

```
+-------------------------------------------------+
|                                                 |
|              CALL CENTER AGENCY                 |
|                                                 |
+-------------------------------------------------+
|                                                 |
|     1. Display Hierarchy                        |
|     2. Hire Employee                            |
|     3. Remove Employee                          |
|     4. Search Employee                          |
|     5. Exit                                     |
|                                                 |
+-------------------------------------------------+

        Enter your choice:
```

When the user chooses 2, it would let the user hire an employee under a specific employee.

```
Who's Hiring the Employee/s? (Enter Name): Zea
Enter ID: #143

How many Employees would Zea Hire? (1 or 2): 2

        Enter Name of Employee 1 (Left): Jerome
        Enter ID for Employee 1: #147
        Jerome with the ID of #147 is OFFICIALLY HIRED by Zea!

        Enter name of Employee 2 (Right): Lourence
        Enter ID for Employee 2: #789
        Lourence with the ID of #789 is OFFICIALLY HIRED by Zea!

        Hierarchy Saved to 'HierarchyStructure.txt'.
```

But if the employee entered is not found, the program would display an error message and would let the user choose from the menu again.

```
Who's Hiring the Employee/s? (Enter Name): Jerome
Enter ID: #236

ERROR! Jerome NOT FOUND in the List of Employees.
```

When user chooses 1, the program would display the hierarchy that is saved in the file.

```
CALL CENTER AGENCY EMPLOYEE HIERARCHY

Zea (ID: #143, Commission: 200)
        Jerome (ID: #147, Commission: 200)
                Christine (ID: #456, Commission: 0)
                Bliza (ID: #654, Commission: 0)
        Lourence (ID: #789, Commission: 100)
                Kier (ID: #852, Commission: 0)
```

When user chooses 3, it would let the user remove a specific employee. The user can only remove an employee without subordinates.

```
Enter the Name of the Employee to Remove: Bliza
Enter ID of the Employee: #654

Employee 'Bliza' with the ID #654 REMOVED SUCCESSFULLY!
```

If the employee entered is the manager or has subordinates, the employee cannot be removed.

```
Enter the Name of the Employee to Remove: Zea
Enter ID of the Employee: #143

        ERROR! CANNOT REMOVE MANAGER.
```

```
Enter the Name of the Employee to Remove: Jerome
Enter ID of the Employee: #147

ERROR! CANNOT REMOVE EMPLOYEE Jerome with ID #147 because he/she has Subordinate/s.
```

When user chooses 4, the program would let the user enter a name. When the name is in the file, the program would display the name together with its commission and ID number.

```
Enter the Name of the Employee to Search: Zea
Enter ID of the Employee: #143

Employee 'Zea' with the ID #143 is the MANAGER with a COMMISSION of 200.
        Employees Hired: Jerome (ID: #147) and Lourence (ID: #789)
```

Otherwise, the program would display an error message.

```
Enter the Name of the Employee to Search: Marae
Enter ID of the Employee: #143

ERROR! Employee 'Marae' with ID #143 NOT FOUND in the List of Employees.
```

When user chooses 5, the program would terminate

```
      Enter your choice: 5

   Thank you for participating.
     Exiting the program...
```

# 7. Acknowledgements:

a. PROJECT TEAM:

- Zea Marae Ramil
- Jerome Montellano
- Lourence Zambas
- Alyza Pitogo
- Christine Rosalejos
- Ma. Bliza Tayo
- Kier Louie Arriesgado

b. RESOURCES

- https://www.geeksforgeeks.org/preorder-traversal-of-binary-tree/
- https://www.w3schools.com/c/c_ref_stdio.php
- https://www.w3schools.com/c/c_ref_stdlib.php
- https://www.w3schools.com/c/c_ref_string.php
- https://youtu.be/MQIF-WMUOL8?si=DJyJmU04phXIoXLj
- https://youtu.be/UqB4EgUxapM?si=Tju0I48gcUQOZuYp
- https://youtu.be/Hzg3kCHJcxI?si=W86hN0KTatcINF1B

c. SPECIAL THANKS

Special thanks to everyone who have contributed to the success of this Data Structures and Algorithms final project – to our class, BSIT-2A, to the leaders of each group, and especially to our professor, Mr. Kenneth Roi Novabos, who guided us throughout this journey.

# 8. Contact Information:

- **Author:** Zea Marae Ramil
- **E-mail:** zearamil9@gmail.com
- **GitHub:** https://github.com/zeamarae