CCT College Dublin

Assessment Cover Page

Module Title:	Algorithms & Constructs
Assessment Title:	System Modelling & Build
Lecturer Name:	Muhammad Iqbal
Student Full Name:	Vadym Tymeichuk
Student Number:	2024388
Assessment Due Date:	May 10, 2025
Date of Submission:	May 10, 2025

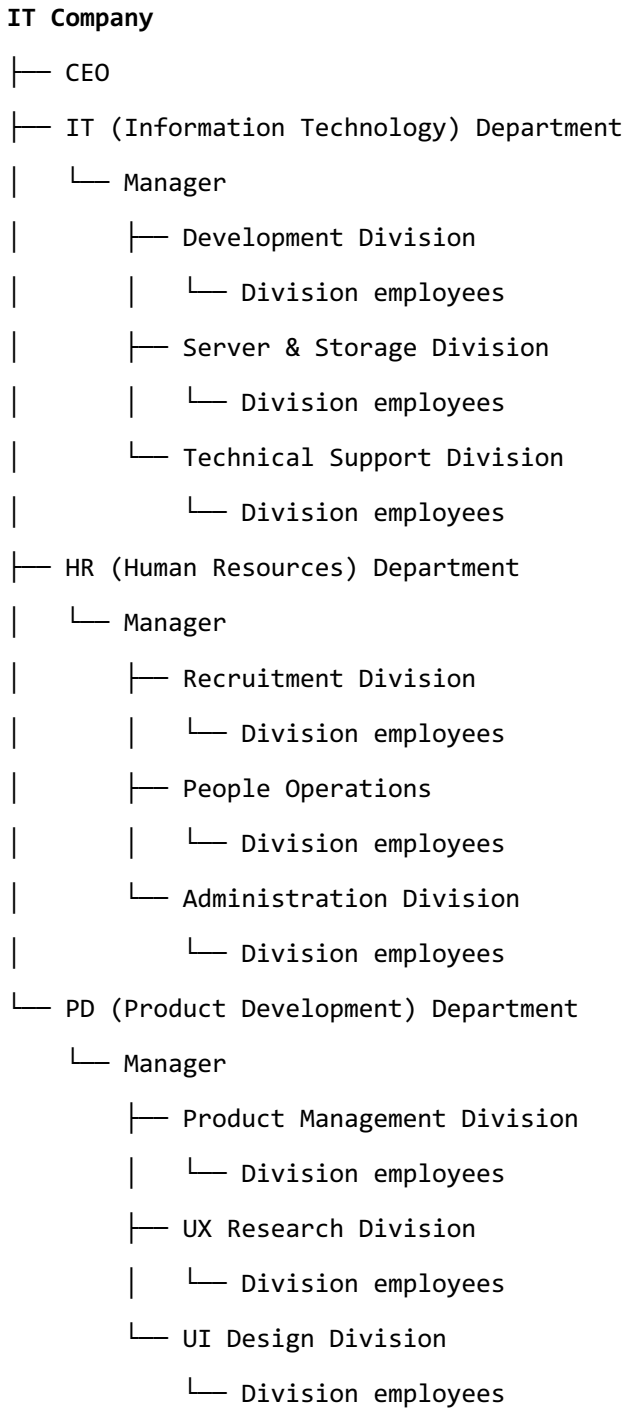
Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Table of Contents

INTRODUCTION

The IT Company app represents structural design of the IT company that consists of departments and divisions as shown below:



SORTING

The IT Company app automatically randomly generate data to populate company departments and divisions with managers and employees. The SORT menu option gives the user an opportunity to sort all company employees by name. Additionally, the user is provided with an option to read and sort the content of an external file, the Applicants_Form.txt specifically.

In both cases the Merge sorting algorithm has been chosen, which is well-suited for handling large or arbitrarily ordered datasets. The reason for choosing the Merge sorting algorithm is because, unlike simpler algorithms such as Bubble Sort or Insertion Sort that are appropriate for small and nearly sorted lists, Merge sorting keeps a time complexity of O(n log n), meaning that even if the amount of entries grows (like number of employees in our case), the sorting time and efficiency remains at the same level. This means, that even if the company records will grow over time, the sorting time and efficiency will not degrade.

SEARCH

Since the sorting algorithm is implemented, the Binary search is the obvious choice when it comes for search queries. In other words, since we have sorting in place, the Binary search would be much more efficient and faster than other searching algorithms. Unlike iterative nature of Bubble or Insertion sort algorithms, the Binary search reduces number of comparisons with each step, which makes it highly efficient, especially as the size of the dataset increases with time.

Similarly to SORT option, the search could be performed either on the company employees, or the Applicants_Form.txt file. Additionally, user can select the search option: first or last name.

It is worth noting that the search implemented in the way that it not only finds the first matching records, but all the matching records. This means that if there are several employees with the same first or last name (either company employees or Applicants_Form.txt file), information about all such employees will be returned by the search.

NEW RECORD INPUT

One of the App functions is adding a new employee. According to the design, this could be achieved in two ways: 1) auto generation; 2) manual input.

If manual input selected, the user can enter employee first name, last name, select a department and choose a position type: regular employee or manager.

In addition, when a new employee is added, all of the existing employees' information is written to a file, and the new employee info is appended to it. Also, the new employee added to the company employee list. This, in turn, will trigger resorting of the existing employee list next time user try to perform sort or search.

RANDOM GENERATION

Random generation of employees happens in two ways.

1) When the company object is initialized, a random number of employees is generated for each department. The number is different depending on the department and varies from 20 to 35. Also, one manager is generated for each department.
2) A single employee could be auto generated and added to the company employee list when auto generation option is chosen when adding an employee (ADD RECORD option).

ENUMS

In the App design Enums are used several types. First, the menu options are represented as an Enum with menu option names and respective numbers. If, when trying to get the menu name by the number entered by the user a null value is returned, it means that user entered incorrect option for the menu and the user is prompted to reenter the number. Additionally, Enum is used to represent department and division names.

Repository link:

https://github.com/Vadym11/AlgorithmsAndConstructs-CA2