UTM UNIVERSITI TEKNOLOGI MALAYSIA

Department of Computer Science
Faculty of Computing

Assignment 1

(*Hospital Management System*)

Programme	:	Bachelor of Computer Science (Data Engineering)
Subject Code	:	SECJ2013
Subject Name	:	Data Structures and Algorithms
Session-Sem	:	2023/2024-1

Prepared by : 1) Neo Zheng Weng (A22EC0093)
2) Joseph Lau Yeo Kai (A22EC0055)
3) Lee Yik Hong (A21BE0376)

Section : 02

Group : Codera

Lecturer : Ms.Lizawati binti Mi Yusuf

Objectives

- Develop a hospital management system.
- Uses the sorting and searching data structure and algorithm in this hospital management system for management.
- Allow healthcare workers to get patient's information, contact information, and their medical history.

Synopsis

This hospital management system has been developed to manage patients' records. The system is designed to get patient information, contacts, and medical history. Healthcare workers can get and manage patients' records in a better way.

Healthcare workers choose to sort or search for this system. Besides, the system uses the Merge sort sorting method in ascending order followed by the attributes as it has a stable and predictable performance. This is because Merge sort has an O(n log n) time complexity, where "n" is the number of elements to be sorted. Attributes examples are name, IC, age, gender, contact number, disease, and diagnosis date. Moreover, this system uses Binary search algorithms as it gives a quick and more accurate search performance for the healthcare workers to search specific patient records or data they want.

In a nutshell, the Hospital Management System helps minimize the workload for healthcare workers in replacing finding patient records manually. The system contributes to efficient healthcare services.

Class Diagram

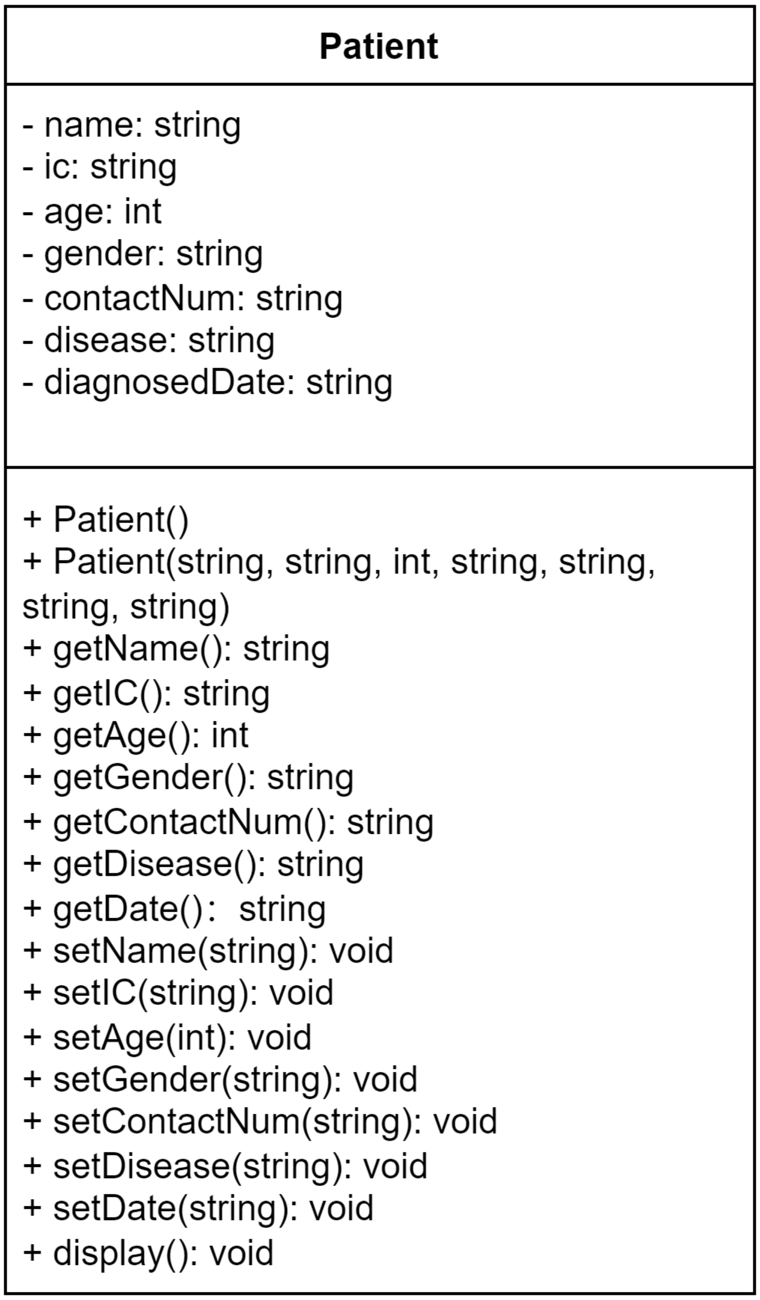

Figure 1.1

Pseudocode

```
1. Start
2. Define a class "Patient" with attributes: name, ic, age, gender, contactNum, disease, date.
3. Initialize an array of Patient objects.
4. Open the file "record.txt" for reading.
   4.1 If the file cannot be opened, display an error message and exit the program.
5. While not at the end of the file:
   5.1 Read a line from the file, storing the data in the attributes of a new Patient object.
   5.2 Add the new Patient object to the array.
6. Close the file.
7. Display a main menu with options for Sorting, Searching, and Logout.
8. Get the user's choice(opt).
9. If opt is 1 (Sorting):
   9.1 Display a submenu with options to sort by name, IC, age, gender, contact number, disease, and diagnosed date.
   9.2 Get the user's choice, choice.
       9.2.1 If choice is 1, call mergeSortByName.
       9.2.2 If choice is 2, call mergeSortByIC.
       9.2.3 If choice is 3, call mergeSortByAge.
       9.2.4 If choice is 4, call mergeSortByGender.
       9.2.5 If choice is 5, call mergeSortByContactNum.
       9.2.6 If choice is 6, call mergeSortByDisease.
       9.2.7 If choice is 7, call mergeSortByDate.
   9.3 Display the sorted list of patients.
10. If opt is 2 (Searching):
   10.1 Display a submenu with options to search by name, IC, and disease.
   10.2 Get the user's choice, select.
   10.2.1 If select is 1, call binarySearchByName.
   10.2.2 If select is 2, call binarySearchByIC.
   10.2.3 If select is 3, call binarySearchByDisease.
   10.3 Display the search results.
11. If opt is 3 (Logout), exit the program.
12. If opt is not 1, 2, or 3, display an error message.
13. Repeat steps 6-11 until the user selects Logout.
14. End
```

Description of How to Implement Data Structure Operations

The system allows the healthcare professionals to manage patient records by providing the option to sort based on different attributes such as name, IC, age, gender, contact number, disease, and diagnosed date. Our team has decided to use the Merge Sort algorithm due to its small complexity time and faster sorting process. However, we only apply ascending order sorting in our system.

1. **Sorting:**

Divide and Conquer algorithm(Merge Sort) is used to divide the input array into two parts, sort the two halves independently, and finally merge them together. Next, the MergeSort function is a recursive function that continuously divides an array until it reaches the individual elements (base case: if first < last). Once all the elements are created successfully into their own, they are recombined into a sorted order using the Merge function.

For example, mergeSortByName() is used to sort the patient record by name in ascending order. Three parameters will be accepted in the program: the patient array, and the first and last indices of the section of the list to be generated. The middle index will be calculated, then the program sorts the two halves of the array iteratively, and finally is merged into two ordered halves. A similar concept is used in the other Merge Sort functions in the system to sort the patient list in ascending order based on the attributes.

2. **Searching:**

In addition to sorting, the function of searching for patients by name, IC, or disease is offered. It is performed by using a binary search. For example, if the user wants to search by name, the binarySearchByName function is called. Then perform a binary search on the list of Patient objects based on the name attribute. If the user wants to search by IC, the binarySearchByIC function is used. To search for disease, the system uses the binarySearchByDisease function to find all patients of the specified disease.

Binary search is done by dividing the list of objects by two times what the object must do. For example, in binarySearchByName, a list of patients is searched by a name. The program accepts four parameters: the patient name array, the lowest and highest indices of the part of the array being searched, the name to search for. The program calculates the middle index, then compares the names in the middle index with names. If the names match, it returns an index. If the target name is less than the name in the middle list, the left side of the array is searched. If there is more than one target name, the right side is searched. This process is repeated until the target name is found or no target name is found in the whole list. However, in binarySearchByDisease, the vector concept is used to store elements with the same data types and to get the range of data. It is to search for all patients with the same disease.