

# HOSPITAL RESERVATION SYSTEM

# DATA STRUCTURE

AIDAN ANDREW DANIEL (A22EC0036)  
MUHAMMAD NABIL IMAN BIN MOHD FAZLI (A22EC0046)  
MUHAMMAD HAZIQ BIN SAAMSOL (A22EC0214)

# Problem Statement

Effective patient record management is vital for the seamless functioning of a hospital.

In response to this need, a comprehensive Hospital Reservation System (HRS) is to be developed as a software application. The system aims to manage patient records efficiently by incorporating essential data structures, ensuring a dynamic and organized workflow within the hospital.

## Problems:

- Efficiently assigning waiting numbers for appointments.
- Maintaining order in the check-in process based on waiting numbers.
- Implementing a memory-clearing mechanism during system shutdown.

# PROJECT OVERVIEW

The Hospital Reservation System (HRS) is designed to overcome challenges in patient record keeping by implementing essential data structure concepts: stack, queue, and tree. The primary objective is to ensure a dynamic, well-organized workflow within the hospital.

## Patient Records Management

- Efficient monitoring of prescriptions, check-ins, and appointments.
- Precise implementation of selected data structures.

## Data Structure Implementation

- Stack, queue, and tree principles applied with precision.
- Unison with unique hospital administration system requirements.

## Doctor Integration

- Seamless addition of doctors to support consultations.
- Contributes to a comprehensive healthcare management approach.

## Queue-Based Handling

- Requests for prescriptions, check-ins, and appointments managed using a queue.
- Systematic and organized procedures implemented with arrays or linked lists.

## Stack-Based Record-Keeping

- Stack-based data structure maintains an updated record of daily hospital visits.
- Last In First Out (LIFO) concept ensures proper record-keeping.

## Tree

- Inserts new nodes by recursively finding the right position based on the system-set value (e.g., name).
- Traverses the tree in an in-order manner and displays nodes, sorted by name, during traversal.
- Nodes can be deleted based on specified criteria (eg..name)

# Objectives

01.

Patients should be able to request appointments, check in, and receive prescriptions using a queue-based data structure to manage appointment requests, patient check-ins, or prescription requests. This will maintain order in the check-in process based on waiting numbers.

02.

The system aims to manage patient records efficiently by incorporating a stack-based data structure for maintaining an updated record of daily hospital appointments. This will help in memory-clearing during system shutdown.

03.

Patient record organization is a critical aspect, and we achieve this through the implementation of a Tree structure by utilizing the Binary Tree concept. This is to ensure efficient sorting, retrieval, and deletion of patient records, allows for systematic arrangement of patient record.

# Data Structure Implementation



# Queue : Appointment system

## 01.

This Queue will provide function such as :  
Add appointments.  
Check in for appoinments  
To Get medication prescription

## 02.

Queue operation :  
The patient that been add into the queue will added from the back.  
  
And when the patient is pop out of the queue it will out from the front.

## 03.

This Queue like first come first serve.  
  
Our system not implimented priority so the first patient that do appointment and check in will be the first in queue. So make sure to check in first.

# Stack : Appointment History

## 01.

Last in First out Principle

In this system the stack we use linked list to give an unlimited storage to hold many patient.

## 02.

This Stack will auto add patient that has done the appointment into the history stack.

And will pop out before the system turn off to release memory.

## 03.

Also this stack is used to create a txt file for appointment history.

All the element in the stack which is patient will looping to export data into txt file.

# Tree : Appointment History (Alphabetical Order)

01.

In This project we are using binary tree so it has 2 children only for each node which is left and right.

02.

This tree provide function to :  
Add node (recursive function)  
delete node (remove by name)  
In order transversing (sorted by name)

03.

Tree concept is that node that has value less than root will be add at the left.  
If the value large than root it will add at the right.

# User Guideline

# Make Appointment

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 1
```

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 1  
Enter your name: haziq  
Enter your contact information: 1234567889  
Enter reason for the appointment: demam
```

1. The system will display a menu.
2. Choose (key in 1) to add a new appointment.

3. Insert Patient name, contact information, and reason for appointment.

```
Appointment request submitted successfully. Your waiting number is: 1  
Press any key to continue . . .
```

4. The system will display a success notification that means that the appointment has been added.

# Check-in for an appointment.

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 2
```

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 4  
  
Not checked-in List:  
Empty  
  
Checked-in List:  
Waiting Number: 1      Name: haziq      Contact: 1234567889      Reason: demam  
  
Prescription waiting List:  
Empty  
Press any key to continue . . .
```

1. The system will display a menu.
2. Choose (key in 2) option 2 to check in for an appointment (Check means that the patient gets ready to make an appointment).

You can check if the user is in the queue by keying in option 4 in the menu. (you do this with other operations to check if the user is added to the queue or not)

```
Check-in successful. You are now in the waiting list.  
Press any key to continue . . .
```

1. The system will display a success message to notify that the patient has a successful check-in.
2. The patient will be in a waiting queue for the next check-up.

# Call the patient to get a Medication prescription

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 3
```

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 3  
Prescription requested for patient haziq.  
Medication prescribed for patient haziq.  
Press any key to continue . . .
```

1. The system will display a menu.
2. Choose (key in 3) option 3 to move the patient in the queue to get medication after the doctor's discussion.
3. The system will automatically remove patients that have got their medication.

4. The system will display a success message of the patient getting a turn to get medication and the medication that the patient is given.

# Check the waiting list queue

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 4
```

```
Not checked-in List:  
Empty  
  
Checked-in List:  
Empty  
  
Prescription waiting List:  
Empty  
Press any key to continue . . .
```

- 1.The system will display a menu.
- 2.Choose (key in 4) option 4 to look at all the waiting queues.

The system will display all the waiting queues for patients that have not checked in yet, have checked in, and are waiting to get medication.

# To view the history of the Hospital appointment

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 5
```

```
Prescription History:  
Patient Name: haziq      Contact: 1234567889      Reason: demam  
Press any key to continue . . .
```

1. The system will display a menu.
2. Choose (key in 5) option 5 to view the list of all patients that do a discussion with the doctor.

3. The system will display all the Patients that have made the appointment only.

# To export the history of the appointment hospital.

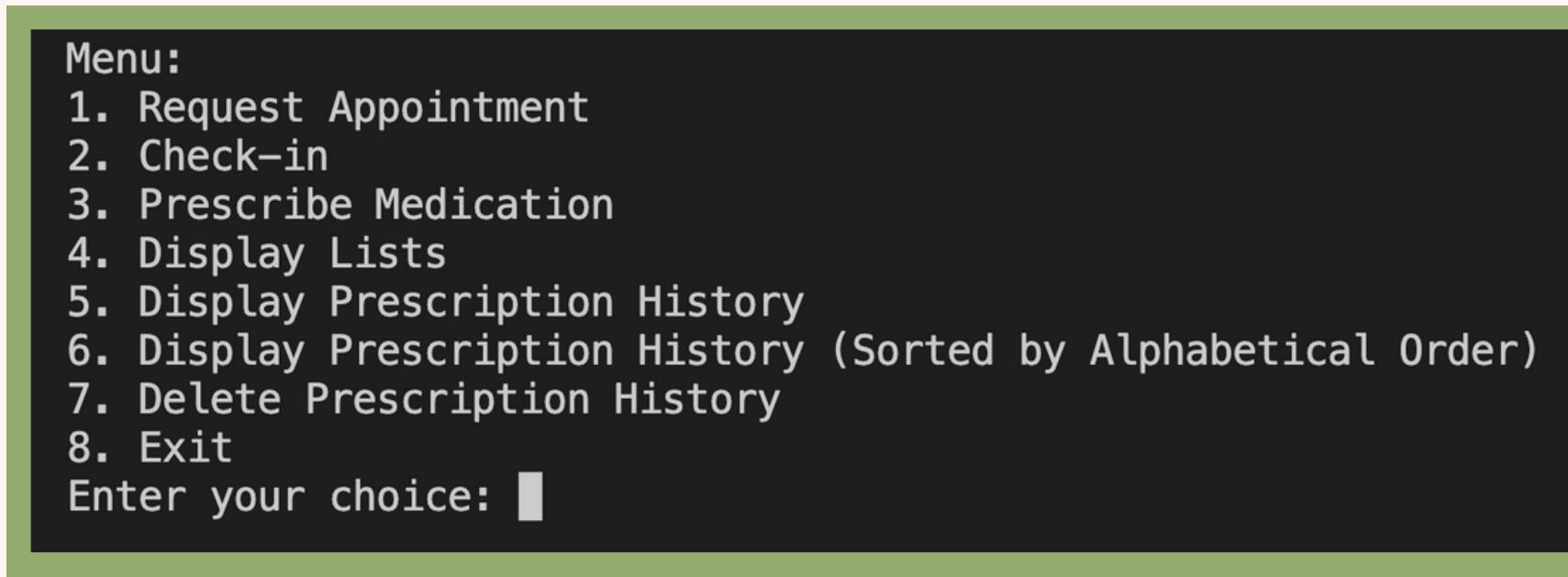
```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 5
```



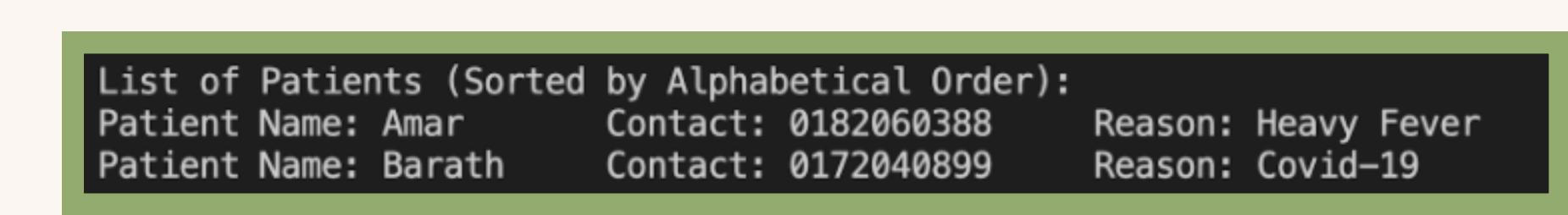
1. The system will display a menu.
2. Choose the (key in 5) option which will display all patients that have done an appointment.

3. The system will automatically export data into a text file.
4. You can find the txt file in the C directory in folder History.

# Display Prescription History (Alphabetical Order)



1. The system will display menu.
2. Choose (Key in 6) option 6 to view all patients history based on Alphabetical Order



3. The system will display all Patients that have made appointments.

# To delete history entry of a Patient

Menu:

- 1. Request Appointment
- 2. Check-in
- 3. Prescribe Medication
- 4. Display Lists
- 5. Display Prescription History
- 6. Display Prescription History (Sorted by Alphabetical Order)
- 7. Delete Prescription History
- 8. Exit

Enter your choice:

Enter your choice: /

Enter the name of the patient to delete: Amar  
Patient Amar's history deleted successfully.

1. The system will display menu.

2. Choose (Key in 7) option 7 to delete history entry of a Patient

3. Input intended patient name in the space.

4. If patient name is found, success message will appear  
If the patient name is not found, the program will notify the user that the patient name is not found in the system.

Enter your choice: 7

Enter the name of the patient to delete: NoName

Patient NoName not found in the system.

If the patient name is not found, the program will notify the user that the patient name is not found in the system.

# End the program

```
Menu:  
1. Request Appointment  
2. Check-in  
3. Prescribe Medication  
4. Display Lists  
5. Display Prescription History  
6. Exit  
Enter your choice: 6
```

```
Exiting program.  
Press any key to continue . . .
```

1. The system will display a menu.
2. Choose (key in 6) option 6 to exit the system.
3. The system will remove all the patient records that are still in the queue and the history.

4. The system will display a message before the end of the program

# SYSTEM DEMO