Data Structures Project

HOSPITAL MANAGEMENT SYSTEM

Type your Section Here

I Project Introduction

1.1 Overview

The Hospital Management System is a modern solution designed to revolutionize healthcare facility operations. It enhances patient care, optimizes the experience for both patients and staff members, and integrates data seamlessly across various departments. Core functionalities include patient registration, appointment scheduling, cancellation, rescheduling and display, locating a doctor's name by entering their specialty, checking and searching inventory using its ID, room assignment, staff management, and billing calculations. This comprehensive system streamlines hospital operations, ensuring better patient outcomes and a more efficient workflow for healthcare professionals.

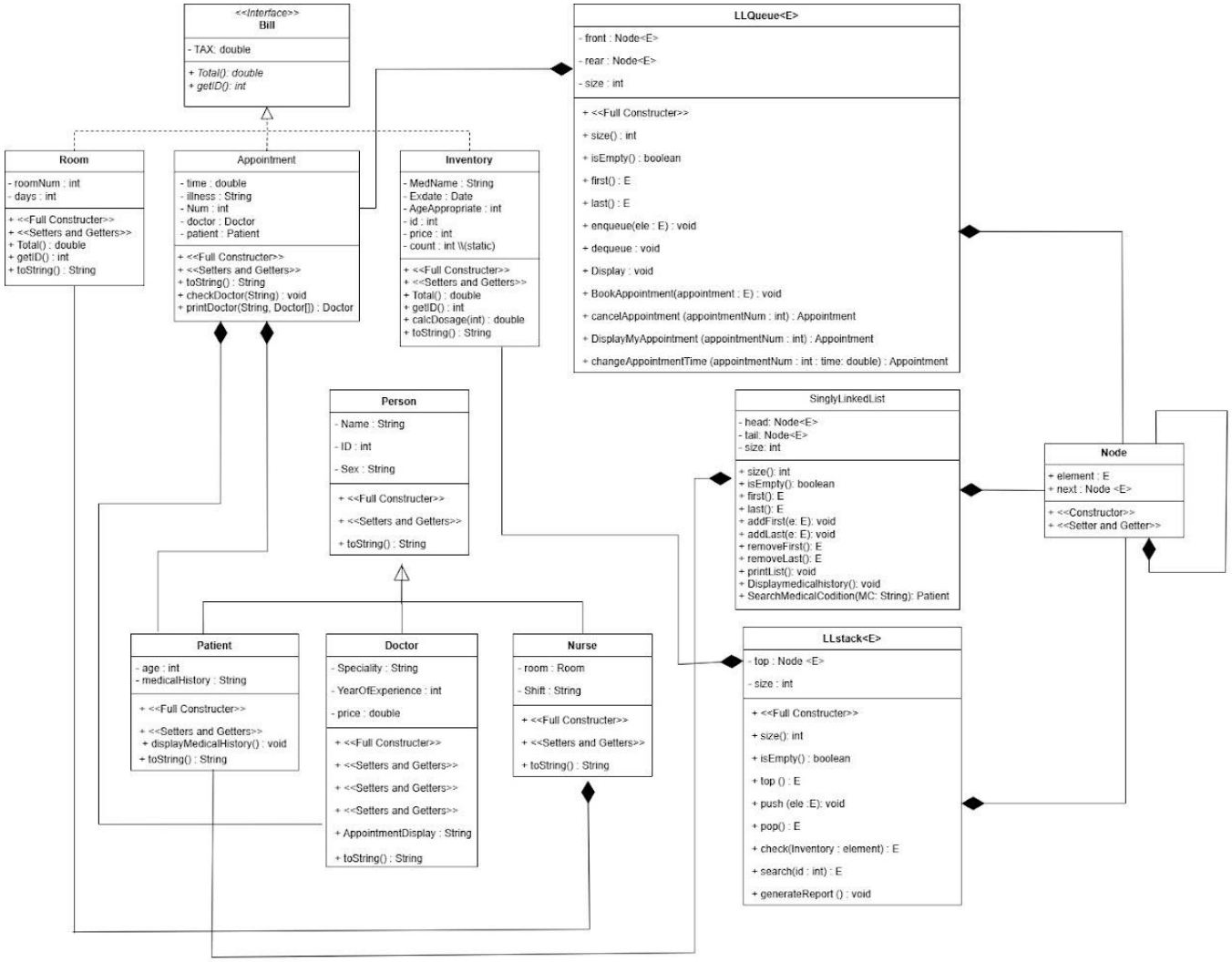
1.2 Problem and solution

Our system solves issues related to patient care by providing a seamless and integrated system for managing patient information, appointments, doctor lookups, inventory checks, room assignments, staff management, and billing processes. This comprehensive solution enhances the overall experience for both patients and healthcare professionals, leading to improved patient outcomes and streamlined workflows within the hospital.

1.3 Solution justification

We choose the following data structures to enhance the users experience with us .queue to manage patient appointments because it operates on a First-In-First-Out (FIFO) principle, ensuring that patients are served in the order they book their appointments. A linked list is ideal for storing patients' medical histories as it allows for efficient insertion and deletion of records, enabling easy updates over time with each node representing a medical record. For inventory management, a stack is suitable as it works on a Last-In-First-Out (LIFO) principle, making it perfect for scenarios where the last item added needs to be used first, such as items with expiration dates. This ensures efficient and orderly management of appointments, medical histories, and inventory within the hospital management system.

II Project class diagram



III Concepts covered

1. Queue

1.1 Book Appointment

```
public void BookAppointment(E appointment) {
    enqueue(elem: appointment);
    System.out.println("Appointment booked: " + appointment);
}
```

1.2 Cancel Appointment

```
public Appointment cancelAppointment(int appointmentNum) {

    if (isEmpty()) {
        System.out.println(x: "Queue is empty ,  There is no appointment to cancel");
        return null;
    }

    Appointment apppoooo = null;
    int size = size();
    for (int i = 0; i < size; i++) {
        Appointment aaa = (Appointment) dequeue();

        if (aaa.getNum() == appointmentNum) {
            apppoooo = aaa;
            System.out.println("Appointment with the number " + appointmentNum + " have been canceled");
        } else {
            enqueue((E) aaa);

        }
    }
    if (apppoooo == null) {
        System.out.println("Appointment with the number " + appointmentNum + " have been not found");
    }
    return apppoooo;

}
```

1.3 Change Appointment

```java
public Appointment changeAppointmentTime(int appointmentNum, double time) {

LLQueue<Appointment> temp = new LLQueue<>();

if (isEmpty()) {
    System.out.println(x: "Queue is empty. There are no appointments to change.");
    return null;
}

Appointment newapp = null;
int size = size();


for (int i = 0; i < size; i++) {
    Appointment v;
    v = (Appointment)dequeue();

    if (v.getNum() == appointmentNum) {

        v.setTime(time);
        newapp = v;
        System.out.println("Appointment with the number " + appointmentNum + " has been updated to new time: " + time);
    }


    temp.enqueue(elem: v);
}


while (!temp.isEmpty()) {
    enqueue((E)temp.dequeue());
}


if (newapp == null) {
    System.out.println("Appointment with the number " + appointmentNum + " was not found.");
}
```

```java
    while (!temp.isEmpty()) {
        enqueue((E)temp.dequeue());
    }


    if (newapp == null) {
        System.out.println("Appointment with the number " + appointmentNum + " was not found.");
    }


    return newapp;
}

}
```

DSProject (run) #8 running... × (7 more...) 143:65

1.4 Display My Appointment

```
public Appointment DisplayMyAppointment(int appointmentNum) {

    LLQueue<Appointment> temp = new LLQueue<>();
    if (isEmpty()) {
        System.out.println(x: "Queue is empty ,  There is no appointment to display");
        return null;
    }

    Appointment app = null;
    int size = size();
    for (int i = 0; i < size; i++) {
        Appointment a = (Appointment) dequeue();

        temp.enqueue(elem: a);
    }

    for (int i = 0; i < size; i++) {
        Appointment b = (Appointment) temp.dequeue();

        if (b.getNum() == appointmentNum) {
            app = b;
            System.out.println("Appointment with the number " + appointmentNum + " will now be shown:\n" + b.toString());
        }
        enqueue((E)b);
    }

    System.out.println(x: "This appointment doesnt exisit");
    return null;
}
```

2. Stack

2.1 Check

```
    public void check(Inventory element) {
    LLstack<Inventory> temp = new LLstack<>();
    boolean found = false;

    int x = element.getId(); // Get the ID to check

    // Check if the stack is empty
    while (!isEmpty()) {
        Inventory ele = (Inventory)pop(); // Pop the element from the stack

        // Compare the IDs of the popped element and the input element
        if (x == ele.getId()) {
            found = true;
            System.out.println("Element " + ele + " is found in the stack.");
        }

        // Push the element into the temporary stack
        temp.push(elem: ele);
    }

    // Restore the original stack from the temporary stack
    while (!temp.isEmpty()) {
        push((E)temp.pop()); // No cast needed here as temp holds Inventory objects
    }

    // If the element was not found, print a message
    if (!found) {
        System.out.println("Element with the ID " + x + " is not found in the stack.");
    }
}
```

2.2 Search

```
    // 2. Search for an element by its ID (assuming elements implement Bill interface)
    public E search(int id) {
        LLstack temp = new LLstack();
        E foundElement = null;

        while (!isEmpty()) {
            E ele = (E) pop();
            if (ele instanceof Bill) {
                Room bill = (Room) ele;
                if (bill.getID() == id) {
                    foundElement = ele;
//                      System.out.println("Element with ID " + id + " is found.");
                }
            }
            temp.push(elem: ele);
        }

        while (!temp.isEmpty()) {
            push((E) temp.pop());
        }

//        if (foundElement == null) {
//            System.out.println("Element with ID " + id + " is not found.");
//        }

        return foundElement;
    }
```

2.3 Generate Report

```
// 4. Generate a report of all items in the stack
public void generateReport() {
    double totalRoomCost = 0;
    double totalInventoryCost = 0;
    int roomCount = 0;
    int inventoryCount = 0;

    LLstack temp = new LLstack();

    while (!isEmpty()) {
        E element = (E) pop();

        if (element instanceof Room) {
            Room room = (Room) element;
            double roomTotal = room.Total();
            totalRoomCost += roomTotal;
            roomCount++;
            System.out.println("Room ID: " + room.getID() + " | Cost: " + roomTotal);
        } else if (element instanceof Inventory) {
            Inventory inventory = (Inventory) element;
            double inventoryTotal = inventory.Total();
            totalInventoryCost += inventoryTotal;
            inventoryCount++;
            System.out.println("Inventory Item: " + inventory.getMedName() + " | ID: " + inventory.getID() + " | Cost: " + inventoryTotal);
        }

        temp.push(elem: element);
    }

    while (!temp.isEmpty()) {
        push((E) temp.pop());
    }

    System.out.println(x: "\n--- Summary Report ---");
    System.out.println("Total Rooms: " + roomCount + " | Total Room Cost: " + totalRoomCost);
    System.out.println("Total Inventory Items: " + inventoryCount + " | Total Inventory Cost: " + totalInventoryCost);
    System.out.println(x: "-----------------------");
}
```

2.4 Print stack

```
    // 3. Display all elements in the stack
    public void printStack() {
        LLstack temp = new LLstack();

        System.out.println(x: "--- Stack Elements ---");

        while (!isEmpty()) {
            E element = (E) pop();
            System.out.println(x: element);
            temp.push(elem: element);
        }

        while (!temp.isEmpty()) {
            push((E) temp.pop());
        }

        System.out.println(x: "-----------------------");
    }
}
```

3. Singly Linked list

3.1 Search Medical Condition

```
    public Patient SearchMedicalCodition(String MC) {
        Node<E> current = head;

      Patient appointment = null;
        while (current != null) {

             Patient a = ( Patient) current.element;
            if (a.getMedicalhistory().equals(anObject:MC)) {
                appointment = a;
            }

            current = current.getNext();
        }
        return appointment;
    }
}
```

3.2 Display Medical History

```
public void Displaymedicalhistory() {
    Node<E> current = head;

    while (current != null) {

     ((Patient)current.getElement()).displayMedicalHistory();
        System.out.println(x: "");
        current = current.getNext();
    }

}
```

3.3 Print list

```
public void printList() {
    Node<E> current = head;

    while (current != null) {

        System.out.println(x: current.element);
        System.out.println(x: "");
        current = current.getNext();
    }

}
```

4 Main Class

```java
import java.util.Scanner;
import java.util.InputMismatchException;
import java.util.Date;

public class DSHP {

    public static void main(String[] args) {

        Scanner input = new Scanner( source:System.in);

        SinglyLinkedList<Person> s = new SinglyLinkedList<>();
        LLstack<Inventory> stack = new LLstack<>();
          LLstack<Room> roomstack = new LLstack<>();
        LLQueue q = new LLQueue();

        Doctor d1 = new Doctor( Speciality:"Cardiologist",  YearOfExperience:4,  price:700,  Name:"Dr.Yasmeen",  ID:1001,  Sex:"Female");
        Doctor d2 = new Doctor( Speciality:"Dermatologist",  YearOfExperience:4,  price:900,  Name:"Dr.Aroob",  ID:1002,  Sex:"Female");
        Doctor d3 = new Doctor( Speciality:"Psychaitrist",  YearOfExperience:4,  price:500,  Name:"Dr.Ahmed",  ID:1003,  Sex:"Male");
        Doctor d4 = new Doctor( Speciality:"Paeditrician",  YearOfExperience:4,  price:500,  Name:"Dr.Fahad",  ID:1004,  Sex:"Female");
        Doctor d5 = new Doctor( Speciality:"Radiologist",  YearOfExperience:4,  price:500,  Name:"Dr.Ghaliah",  ID:1005,  Sex:"Female");

        Doctor array[] = {d1, d2, d3, d4, d5};

        Nurse n1 = new Nurse( roomNum:110,  Shift:"night",  Name:"Ameena",  ID:2001,  Sex:"Female");
        Nurse n2 = new Nurse( roomNum:120,  Shift:"night",  Name:"sara",  ID:2002,  Sex:"Female");
        Nurse n3 = new Nurse( roomNum:210,  Shift:"night",  Name:"latifa",  ID:2003,  Sex:"Female");
        Nurse n4 = new Nurse( roomNum:220,  Shift:"night",  Name:"khalid",  ID:2004,  Sex:"Male");
        Nurse n5 = new Nurse( roomNum:310,  Shift:"Day",  Name:"omar",  ID:2005,  Sex:"Male");

        Room room1 = new Room( roomNum:110,  days:3);
        Room room2 = new Room( roomNum:120,  days:5);
        Room room3 = new Room( roomNum:210,  days:5);
        Room room4 = new Room( roomNum:220,  days:5);

        Inventory med1 = new Inventory( MedName:"Aspirin", new Date(),  AgeAppropriate:12,  id:1000,  price:15);
        Inventory med2 = new Inventory( MedName:"Tylenol", new Date(),  AgeAppropriate:16,  id:1001,  price:20);

        roomstack.push( elem:room1);
        roomstack.push( elem:room2);
        roomstack.push( elem:room3);
        roomstack.push( elem:room4);
        stack.push( elem:med1);
        stack.push( elem:med2);

        System.out.println( x:"\t\tWelcome to PNU Hospital!!");
```

```java
        System.out.println( x:"\t\tWelcome to PNU Hospital!!");

        int choice = 0;
        int num = 0;
        int PatientID = 0;

        do {
            try {
                System.out.println( x:"\nPlease choose what you want to do :");
                System.out.println("1-Find Doctor \n"
                        + "2-Book Appointment\n"
                        + "3-Cancel Appointment\n"
                        + "4-Change Appointment Time\n"
                        + "5-Display Appointment\n"
                        + "6-Register as a new patient\n"
                        + "7-Display medical history\n"
                        + "8-Search Medical History\n"
                        + "9-find rooom with using ID\n"
                        + "10-Check Medication \n "
                        + "11-Generate Report\n"
                        + "12-Print Rooms and medications\n"
                        + "13-Exit");
                System.out.print("-----------------------------------------\n" + "Your  select  >>>  ");

                choice = input.nextInt();

                switch (choice) {

                    case 1:
                        System.out.println( x:"please enter the specialty you need");
                        String specialty = input.next();
                        Appointment.checkDoctor( Speciality:specialty, array);
                        break;
                    case 2:
                        System.out.println( x:"""
                                please enter your information
                                the time :
                                your illness:
                                with the doctor:
                                Patient Name:
                                age:
                                Sex:""");
                        int t = input.nextInt();
                        String ill = input.next();
                        String doc = input.next();
```

```
                String doc = input.next();
                String patientN = input.next();
                int patientA = input.nextInt();
                String patientS = input.next();
                ++num;
                ++PatientID;

                Patient p = new Patient( age:patientA, medicalhistory:ill , Name:patientN,  ID:PatientID,  Sex:patientS);

                Doctor ddd = Appointment.printDoctor( n:doc, array);
                Appointment appo = new Appointment( time:t,  illness:ill,  Num:num,  doctor:ddd,  patient:p);

                q.BookAppointment( appointment:appo);
                s.addFirst( e:p);

                break;
            case 3:
                System.out.println( x:"please enter your appointment number :");
                int appnum = input.nextInt();

                q.cancelAppointment( appointmentNum:appnum);
                break;

            case 4:
                System.out.println( x:"please enter your appointment number to change your time:");
                int appcha = input.nextInt();
                System.out.println( x:"enter the time you want ");
                double newT = input.nextInt();
                q.changeAppointmentTime( appointmentNum:appcha,  time:newT);
                break;
            case 5:
                System.out.println( x:"please enter your appointment number :");
                int an = input.nextInt();
                q.DisplayMyAppointment( appointmentNum:an);
                break;
            case 6:
                System.out.println("please enter your information\n"
                        + "age :\n"
                        + "name :\n"
                        + "ID:\n"
                        + "sex:\n");
                int a1 = input.nextInt();
                String n = input.next();
                int i = input.nextInt();
                String ps = input.next();
```

```
                String ps = input.next();

                Patient newP = new Patient( age:a1,  Name:n,  ID:i,  Sex:ps);

                System.out.print( s:"you are now registered!");
                break;

            case 7:
                s.Displaymedicalhistory();
                System.out.println( x:"");
                break;
            case 8:
                System.out.println( x:"what is the condition you are looking for ");
                String MC = input.next();
                System.out.print(  obj:s.SearchMedicalCodition(MC));
                System.out.println( x:"");
                break;

            case 9:
                System.out.print( s:"Enter ID to search your room: ");
                int searchId = input.nextInt();
                Bill found = roomstack.search( id:searchId);
                if (found != null) {
                    System.out.println("Found: " + found);
                } else {
                    System.out.println("No element with ID " + searchId + " found.");
                }
                break;

            case 10:
                System.out.print( s:"Enter medication ID to check: ");
                int inventoryId = input.nextInt();
                Inventory k = new Inventory( id:inventoryId);

                stack.check( element:k);

                break;
            case 11:

                stack.generateReport();
                break;
            case 12:
                stack.printStack();
                roomstack.printStack();
                break;
```

```
                        stack.check( element:k);

                        break;
                    case 11:

                        stack.generateReport();
                        break;
                    case 12:
                        stack.printStack();
                        roomstack.printStack();
                        break;

                    case 13:
                        break;
                    default:
                        System.out.println( x:"Invalid option!");
                        System.out.print( s:"please enter one of the exiting options.");

                }
            } catch (InputMismatchException e) {
                System.err.println( x:"Invalid input");
                input.next();

            }

        } while (choice != 13);

    }
}
```

```
                Welcome to PNU Hospital!!

Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  1
please enter the specialty you need
Cardiologist

Doctor Name: Dr.Yasmeen ID:1001 Sex:Female
Speciality:Cardiologist Years Of Experience:4
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  2
please enter your information
the time :
your illness:
with the doctor:
Patient Name:
age:
Sex:
4
pain
Dr.Yasmeen
Maysam
19
female
Appointment booked:
WithDr.Yasmeen: ,Doctor ID:1001 Sex:Female
Speciality:Cardiologist, Years Of Experience:4
Patient{age=19, medicalhistory=null}, Appointment NUmber: 1, Appointment Time:4.0, Illness:pain
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  3
please enter your appointment number :
1
Appointment with the number 1 have been canceled
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  5
please enter your appointment number :
2
Appointment with the number 2 will now be shown:

WithDr.Aroob: ,Doctor ID:1002 Sex:Female
Speciality:Dermatologist, Years Of Experience:4
Patient{age=20, medicalhistory=null}, Appointment NUmber: 2, Appointment Time:8.0, Illness:headache
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-----------------------------------------
Your  select  >>>  6
please enter your information
age :
name :
ID:
sex:

19
maysam
100
female
you are now registered!
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-----------------------------------------
Your  select  >>>  7
Medical history:vomiting


Medical history:pain


Medical history:headache
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  8
what is the condition you are looking for
headache
Patient{age=19, medicalhistory=headache}
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  9
Enter ID to search your room: 110
Found: Room{roomNum=110, days=3}
```

```
please enter one of the exiting options.
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  10
Enter medication ID to check: 1000
Element Inventory{MedName=Aspirin, Exdate=Sun Nov 17 23:42:59 AST 2024,
AgeAppropriate=12, id=1000, price=15} is found in the stack.
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  11
Inventory Item: Tylenol | ID: 1001 | Cost: 22.0
Inventory Item: Aspirin | ID: 1000 | Cost: 16.5

--- Summary Report ---
Total Rooms: 0 | Total Room Cost: 0.0
Total Inventory Items: 2 | Total Inventory Cost: 38.5
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  12
--- Stack Elements ---
Inventory{MedName=Tylenol, Exdate=Sun Nov 17 21:19:54 AST 2024, AgeAppropriate=16, id=1001, price=20}
Inventory{MedName=Aspirin, Exdate=Sun Nov 17 21:19:54 AST 2024, AgeAppropriate=12, id=1000, price=15}
----------------------
--- Stack Elements ---
Room{roomNum=220, days=5}
Room{roomNum=210, days=5}
Room{roomNum=120, days=5}
Room{roomNum=110, days=3}
----------------------
```

```
Please choose what you want to do :
1-Find Doctor
2-Book Appointment
3-Cancel Appointment
4-Change Appointment Time
5-Display Appointment
6-Register as a new patient
7-Display medical history
8-Search Medical History
9-find rooom with using ID
10-Check Medication
 11-Generate Report
12-Print Rooms and medications
13-Exit
-------------------------------------------
Your  select  >>>  13
BUILD SUCCESSFUL (total time: 5 minutes 26 seconds)
```