|  |
| --- |
| SDOOP Mini Project |
| Patient Record Management System |
| III Semester ICAS |

Neil Bhutada 181627135

Kirat Alreja 181627132

Manav Bahl 181627136

Arya Sharma 181627134

Table of Contents

[1. Do a survey and develop the case study. 2](#_Toc23088086)

[2. Derive the use cases from the requirements and write the use case diagram 3](#_Toc23088087)

[3. Write the class diagram (with specification classes) 4](#_Toc23088088)

[4. Give interaction diagrams for at least three use cases 5](#_Toc23088089)

[5. Give activity diagram for the whole system 7](#_Toc23088090)

[6. Show the package diagram of the entire system with clear separation of 3 layers of software architecture 8](#_Toc23088091)

[7. Generate the source code in Star UML with the language of your choice 9](#_Toc23088092)

## 1. Do a survey and develop the case study.

A hospital must use a patient RMS (Record Management System) to store all the necessary details of a patient. The patient record system should be a part of a hospital system, which stores details of the hospital such as location or branch identification number. A doctor record system should also be used to assign doctors as required by patient illness/condition.

The patient record system stores basic information of the patient such as the Name, Phone Number, email and address. To identify the patient from previous visits or for future visits, a patient identification number should be used for every patient in the record. When a patient comes to the hospital & checks in at the reception, a ward should be allotted.

Wards can be allocated in different categories. There are normal, executive & luxury wards to choose from. Each ward category has some extra features like increased bed size & better air conditioning but the basic details like Ward ID are common to all. The food charges are not included in the ward & must be calculated separately. The cost of the ward is calculated on a daily basis, with extra charges for late payment.

A billing system must keep track of all patient’s status of dues. At the end of treatment, the billing system must calculate the total bill including ward, food & miscellaneous charges. A copy of the bill’s ID can be kept for future references.

When doctor is allotted to a patient, doctor system must assign a relevant secondary doctor system for the same. For example - Neurology, Paediatrics or Gastroenterology. A copy of patient’s details will be sent to the subsystem of the secondary doctor department.

Patient record should be the same for OPD patients & Non-OPD Patients. In the case of OPD patients, Ward ID is not applicable & a separate OPD department must be used for calculating bills. The OPD department records should also have a Patient ID for future purposes.

The lab testing department of the hospital is common to all doctor departments. Lab testing is called when a doctor department orders the patient. In that case, the patient details will be shared with the lab test department & a pdf file of the test results will be returned to the doctor department. A copy of lab pdf file is also stored in patient record & is accessible by other doctor departments with prior permission from patient.

The catering section of the hospital must customize food as per instructions sent by the doctor department. When a patient checks in at the reception, food preference such as veg/non veg/ vegan must be taken. When doctor departments wants to set a food instruction for patient as per diagnosis, the instruction & the food preference should be sent to catering.

## 2. Derive the use cases from the requirements and write the use case diagram

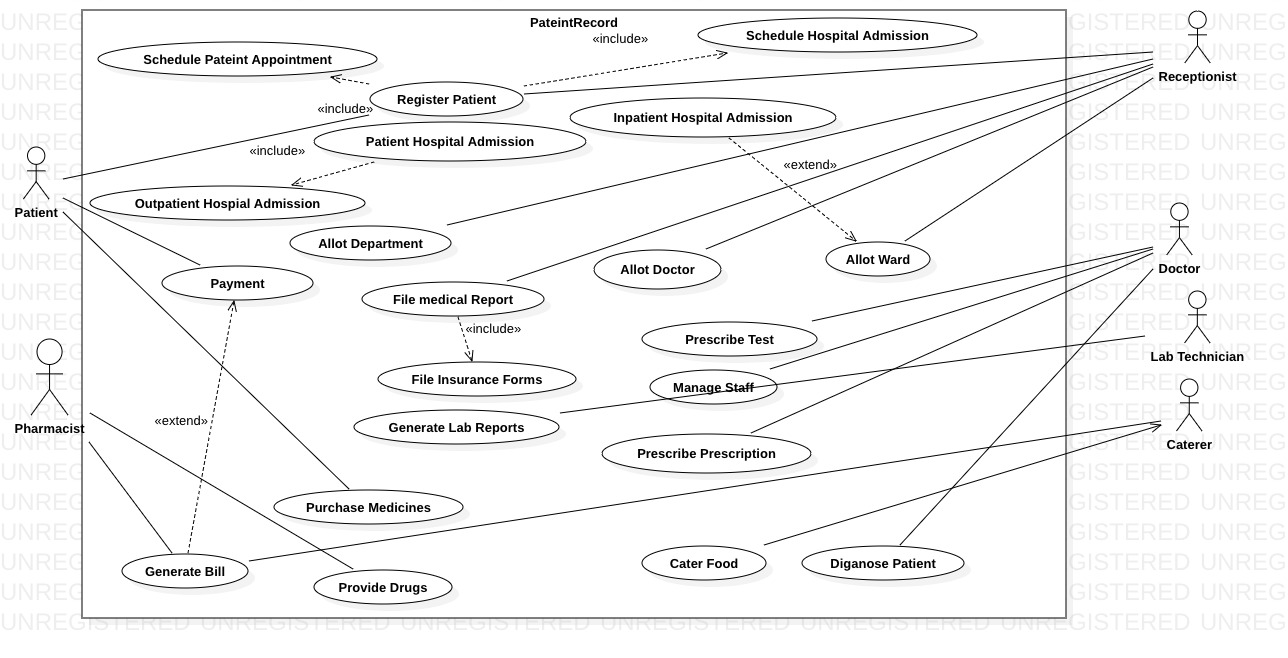


Figure 1

## 3. Write the class diagram (with specification classes)

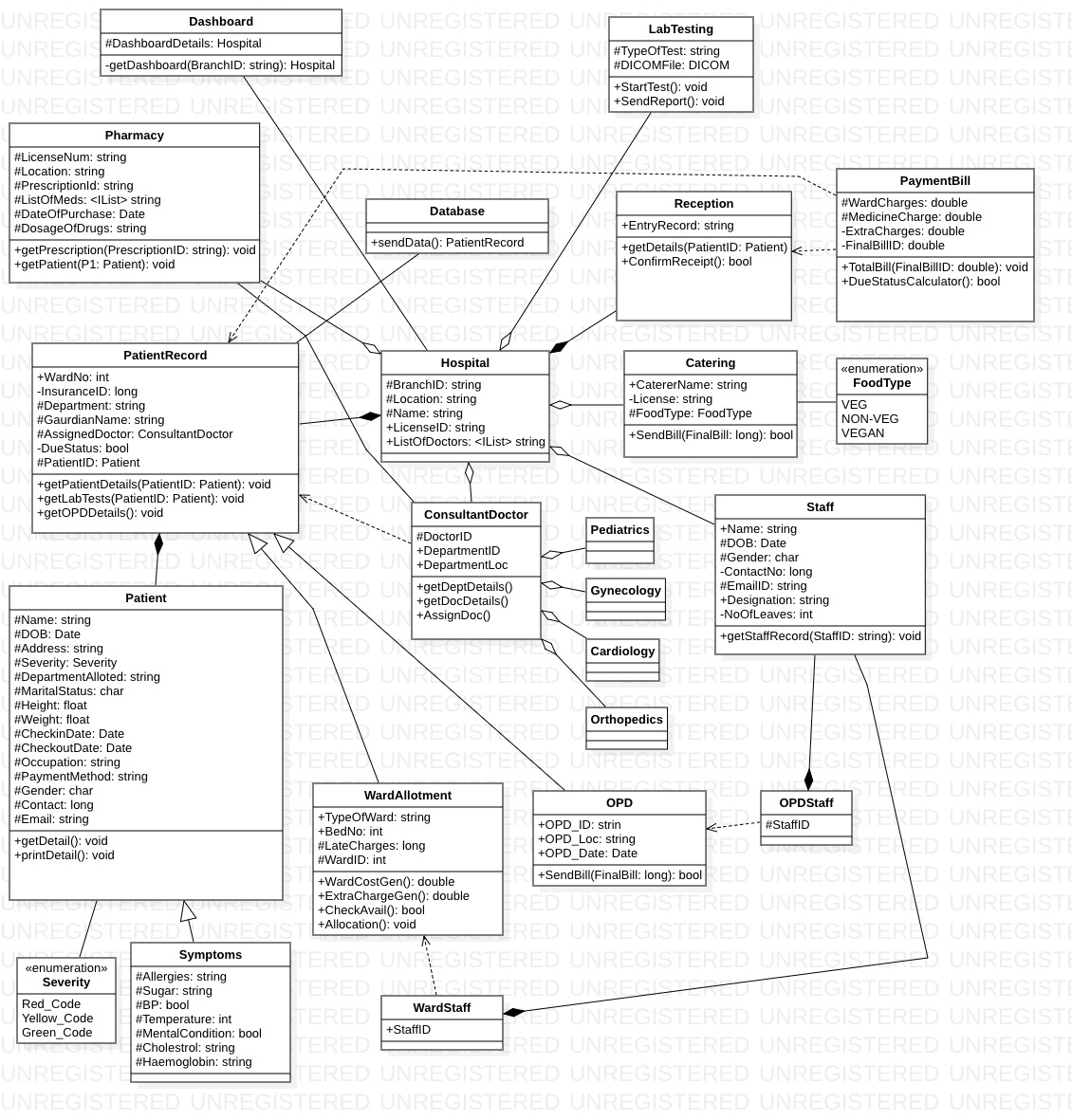
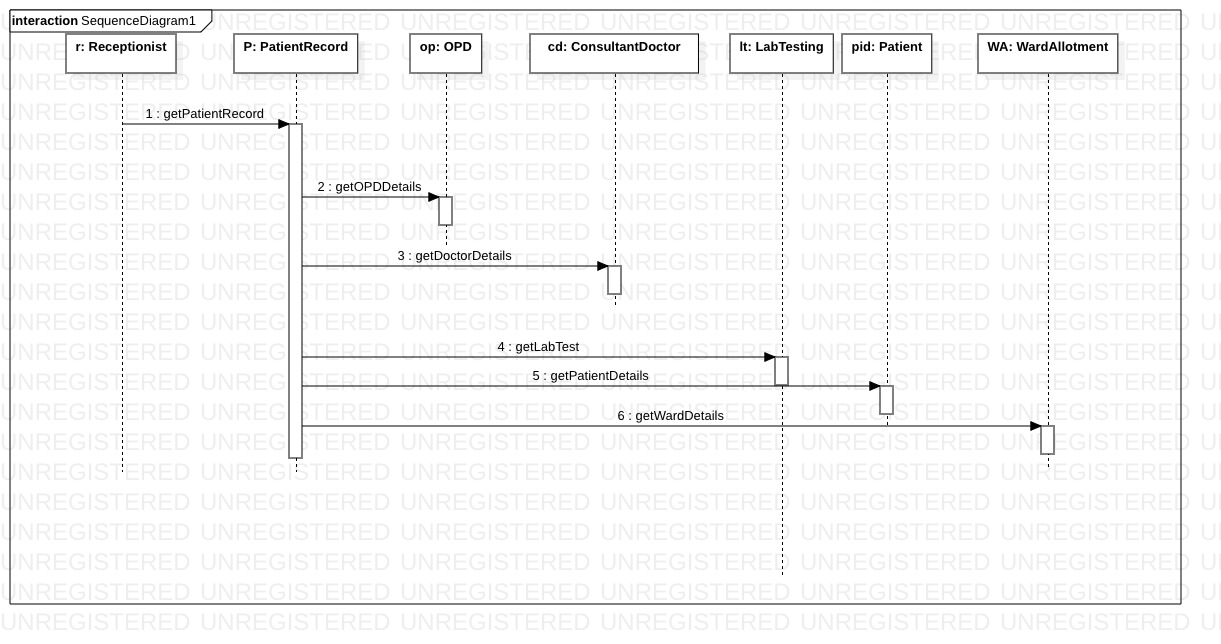


Figure 2

## 4. Give interaction diagrams for at least three use cases



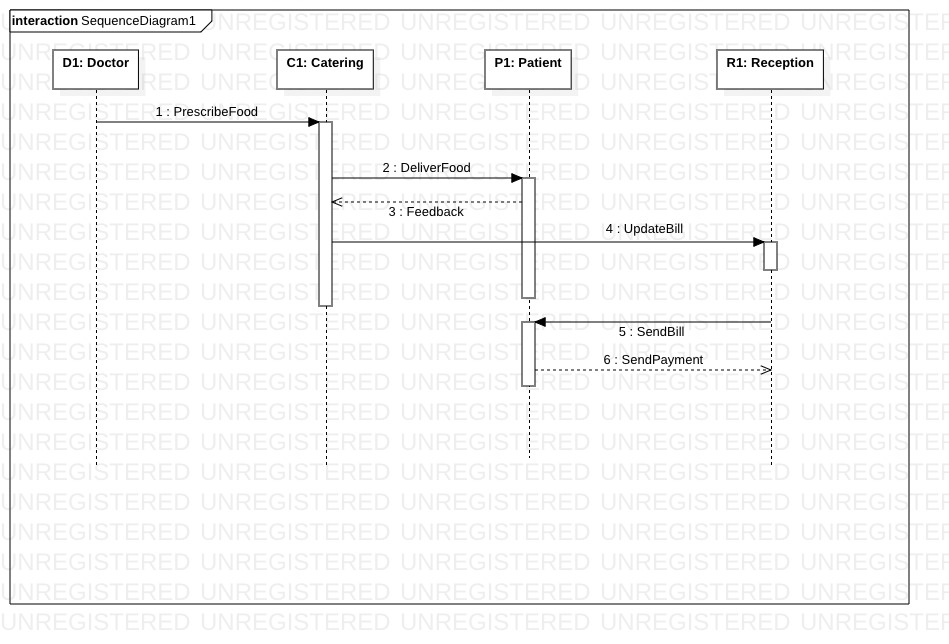


Figure 4 (Catering Food)

Figure 3 (Register Patient)

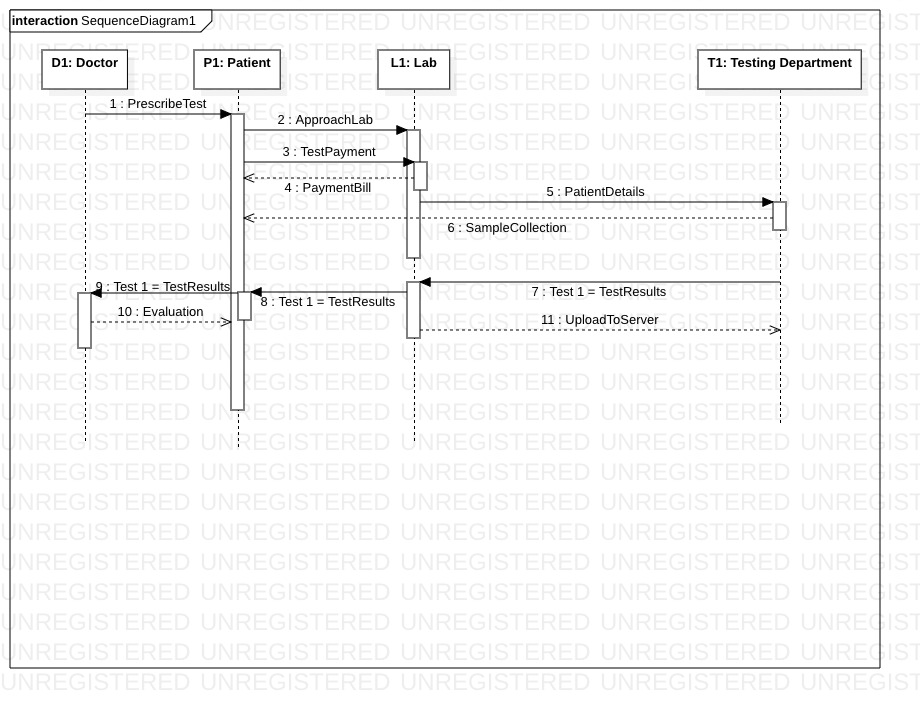


Figure 5(Generate Lab Reports)

## 5. Give activity diagram for the whole system

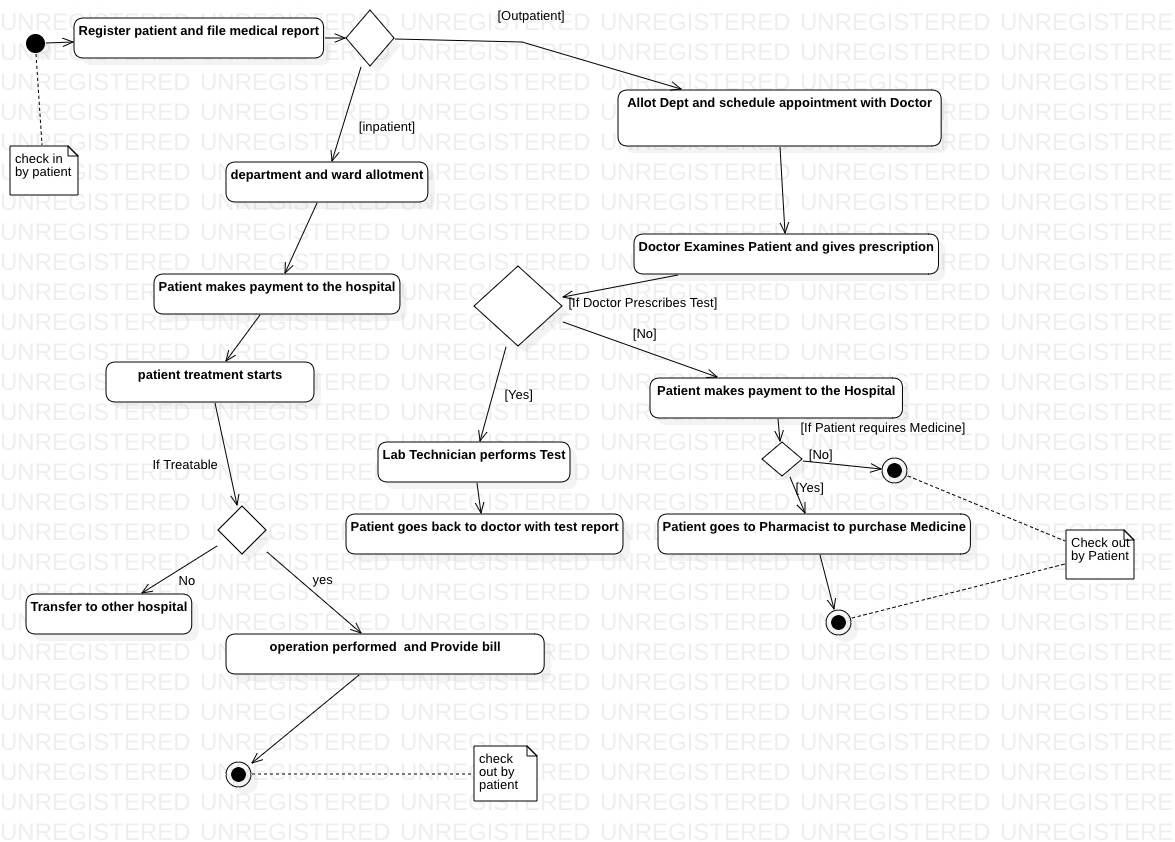


Figure 6

## 6. Show the package diagram of the entire system with clear separation of 3 layers of software architecture

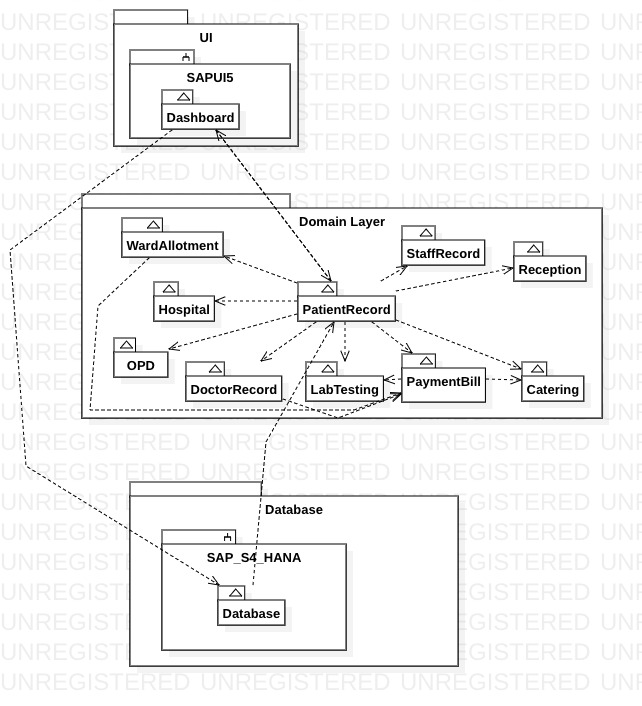


Figure 7

## 7. Generate the source code in Star UML with the language of your choice

import java.util.\*;

/\*\*

\*

\*/

public class WardStaff {

/\*\*

\* Default constructor

\*/

public WardStaff() {

}

/\*\*

\*

\*/

public void StaffID;

}

import java.util.\*;

/\*\*

\*

\*/

public class WardAllotment extends WardAllotment {

/\*\*

\* Default constructor

\*/

public WardAllotment() {

}

/\*\*

\*

\*/

public string TypeOfWard;

/\*\*

\*

\*/

public int BedNo;

/\*\*

\*

\*/

protected long LateCharges;

/\*\*

\*

\*/

protected int WardID;

/\*\*

\* @return

\*/

public double WardCostGen() {

// TODO implement here

return 0.0d;

}

/\*\*

\* @return

\*/

public double ExtraChargeGen() {

// TODO implement here

return 0.0d;

}

/\*\*

\* @return

\*/

public bool CheckAvail() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void Allocation() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Symptoms extends Patient {

/\*\*

\* Default constructor

\*/

public Symptoms() {

}

/\*\*

\*

\*/

protected string Allergies;

/\*\*

\*

\*/

protected string Sugar;

/\*\*

\*

\*/

protected bool BP;

/\*\*

\*

\*/

protected int Temperature;

/\*\*

\*

\*/

protected bool MentalCondition;

/\*\*

\*

\*/

protected string Cholestrol;

/\*\*

\*

\*/

protected string Haemoglobin;

}

import java.util.\*;

/\*\*

\*

\*/

public class Staff {

/\*\*

\* Default constructor

\*/

public Staff() {

}

/\*\*

\*

\*/

public string Name;

/\*\*

\*

\*/

protected Date DOB;

/\*\*

\*

\*/

protected char Gender;

/\*\*

\*

\*/

private long ContactNo;

/\*\*

\*

\*/

protected string EmailID;

/\*\*

\*

\*/

public string Designation;

/\*\*

\*

\*/

private int NoOfLeaves;

/\*\*

\* @param StaffID

\* @return

\*/

public void getStaffRecord(string StaffID) {

// TODO implement here

return null;

}

}

/\*\*

\*

\*/

public enum Severity {

Red\_Code,

Yellow\_Code,

Green\_Code

}

import java.util.\*;

/\*\*

\*

\*/

public class Reception {

/\*\*

\* Default constructor

\*/

public Reception() {

}

/\*\*

\*

\*/

public string EntryRecord;

/\*\*

\* @param PatientID

\*/

public void getDetails(Patient PatientID) {

// TODO implement here

}

/\*\*

\* @return

\*/

public bool ConfirmReceipt() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Pharmacy {

/\*\*

\* Default constructor

\*/

public Pharmacy() {

}

/\*\*

\*

\*/

protected string LicenseNum;

/\*\*

\*

\*/

protected string Location;

/\*\*

\*

\*/

protected string PrescriptionId;

/\*\*

\*

\*/

protected <IList> string ListOfMeds;

/\*\*

\*

\*/

protected Date DateOfPurchase;

/\*\*

\*

\*/

protected string DosageOfDrugs;

/\*\*

\* @param PrescriptionID

\* @return

\*/

public void getPrescription(string PrescriptionID) {

// TODO implement here

return null;

}

/\*\*

\* @param P1

\* @return

\*/

public void getPatient(Patient P1) {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Pediatrics {

/\*\*

\* Default constructor

\*/

public Pediatrics() {

}

}

import java.util.\*;

/\*\*

\*

\*/

public class PaymentBill {

/\*\*

\* Default constructor

\*/

public PaymentBill() {

}

/\*\*

\*

\*/

protected double WardCharges;

/\*\*

\*

\*/

protected double MedicineCharge;

/\*\*

\*

\*/

private double ExtraCharges;

/\*\*

\*

\*/

private double FinalBillID;

/\*\*

\* @param FinalBillID

\* @return

\*/

public void TotalBill(double FinalBillID) {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public bool DueStatusCalculator() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class PatientRecord {

/\*\*

\* Default constructor

\*/

public PatientRecord() {

}

/\*\*

\*

\*/

public int WardNo;

/\*\*

\*

\*/

private long InsuranceID;

/\*\*

\*

\*/

protected string Department;

/\*\*

\*

\*/

protected string GaurdianName;

/\*\*

\*

\*/

protected ConsultantDoctor AssignedDoctor;

/\*\*

\*

\*/

private bool DueStatus;

/\*\*

\*

\*/

protected Patient PatientID;

/\*\*

\* @param PatientID

\* @return

\*/

public void getPatientDetails(Patient PatientID) {

// TODO implement here

return null;

}

/\*\*

\* @param PatientID

\* @return

\*/

public void getLabTests(Patient PatientID) {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void getOPDDetails() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Patient {

/\*\*

\* Default constructor

\*/

public Patient() {

}

/\*\*

\*

\*/

protected string Name;

/\*\*

\*

\*/

protected Date DOB;

/\*\*

\*

\*/

protected string Address;

/\*\*

\*

\*/

protected Severity Severity;

/\*\*

\*

\*/

protected string DepartmentAlloted;

/\*\*

\*

\*/

protected char MaritalStatus;

/\*\*

\*

\*/

protected float Height;

/\*\*

\*

\*/

protected float Weight;

/\*\*

\*

\*/

protected Date CheckinDate;

/\*\*

\*

\*/

protected Date CheckoutDate;

/\*\*

\*

\*/

protected string Occupation;

/\*\*

\*

\*/

protected string PaymentMethod;

/\*\*

\*

\*/

protected char Gender;

/\*\*

\*

\*/

protected long Contact;

/\*\*

\*

\*/

protected string Email;

/\*\*

\* @return

\*/

public void getDetail() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void printDetail() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Orthopedics {

/\*\*

\* Default constructor

\*/

public Orthopedics() {

}

}

import java.util.\*;

/\*\*

\*

\*/

public class OPDStaff {

/\*\*

\* Default constructor

\*/

public OPDStaff() {

}

/\*\*

\*

\*/

protected void StaffID;

}

import java.util.\*;

/\*\*

\*

\*/

public class OPD extends OPD {

/\*\*

\* Default constructor

\*/

public OPD() {

}

/\*\*

\*

\*/

public strin OPD\_ID;

/\*\*

\*

\*/

public string OPD\_Loc;

/\*\*

\*

\*/

public Date OPD\_Date;

/\*\*

\* @param FinalBill

\* @return

\*/

public bool SendBill(long FinalBill) {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class LabTesting {

/\*\*

\* Default constructor

\*/

public LabTesting() {

}

/\*\*

\*

\*/

protected string TypeOfTest;

/\*\*

\*

\*/

protected DICOM DICOMFile;

/\*\*

\* @return

\*/

public void StartTest() {

// TODO implement here

return null;

}

/\*\*

\* @return

\*/

public void SendReport() {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Hospital {

/\*\*

\* Default constructor

\*/

public Hospital() {

}

/\*\*

\*

\*/

protected string BranchID;

/\*\*

\*

\*/

protected string Location;

/\*\*

\*

\*/

protected string Name;

/\*\*

\*

\*/

public string LicenseID;

/\*\*

\*

\*/

public <IList> string ListOfDoctors

}

import java.util.\*;

/\*\*

\*

\*/

public class Gynecology {

/\*\*

\* Default constructor

\*/

public Gynecology() {

}

}

/\*\*

\*

\*/

public enum FoodType {

VEG,

NON-VEG,

VEGAN

}

import java.util.\*;

/\*\*

\*

\*/

public class ENT {

/\*\*

\* Default constructor

\*/

public ENT() {

}

}

import java.util.\*;

/\*\*

\*

\*/

public class ConsultantDoctor {

/\*\*

\* Default constructor

\*/

public ConsultantDoctor() {

}

/\*\*

\*

\*/

protected void DoctorID;

/\*\*

\*

\*/

public void DepartmentID;

/\*\*

\*

\*/

public void DepartmentLoc;

/\*\*

\*

\*/

public void getDeptDetails() {

// TODO implement here

}

/\*\*

\*

\*/

public void getDocDetails() {

// TODO implement here

}

/\*\*

\*

\*/

public void AssignDoc() {

// TODO implement here

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Catering {

/\*\*

\* Default constructor

\*/

public Catering() {

}

/\*\*

\*

\*/

public string CatererName;

/\*\*

\*

\*/

private string License;

/\*\*

\*

\*/

protected FoodType FoodType;

/\*\*

\* @param FinalBill

\* @return

\*/

public bool SendBill(long FinalBill) {

// TODO implement here

return null;

}

}

import java.util.\*;

/\*\*

\*

\*/

public class Cardiology {

/\*\*

\* Default constructor

\*/

public Cardiology() {

}

}