Hospital Management System

By

Parshwa Shah 1641068

Nishant Patel 1641041

Yash H Patel 1641044

1. Introduction

Our project is Hospital Management System. Hospital Management System used now are way too bloated and involve different programs for each sector of the Hospital (like doctor, nurse etc).They are also expensive and involve yearly contracts that cannot be unsubscribed anytime.

The approach that we had was to make the program as streamlined as possible and make the user interface appealing.We plan on making it a web based application in the future.

2. Detailed Features

Head Doctor

- Give Prescriptions
- View Patient History
- Surgery view Time/Date

Assistant Doctor

- Give Prescriptions
- View Patient History
- View head Doc timings/surgeries

Nurse

- View Room Assigned to
- View Patient Med Requirements

Accountant

- View Accounts
- Add Accounts

Contact Manager

Receptionist

Book Appointment
Bill Generate
View Appointments
View Doctor Availability

Management

Add new Staff(Doctor,Receptionist,Nurse,Accountant)
View Accounts
Assign Staff to diff Rooms(Nurse)
View Staff Data

Pharmacist

Generate Medical Bill
View Available Medicines
Order new Medicines

Lab Incharge

Make Reports
Generate Lab Bills
View Reports
View Lab Appointments
Book Lab Appointments

3. Advantages

This application will help the hospital management to solve problems of security as it uses strict programming approach.

It will also be much more cheaper than what they are using currently.

It will make all the sector/staff (doc, accountant, etc) easy to use as it uses simple flow.

4. Limitations

We have considered three types of doctor available - Head doctor, Assistant doctor, Lab incharge.

Prescription will be atmost of 5 medicines and their corresponding dosage and to prescribe more medicines it will generate new prescription.

Pharmacy Bill can contain at most 3 Medicines.

If any medicine is not available inside a pharmacy than the IPD patient has to take medicines from outside.

Morgue is not considered here as a special type of room.

If Head doctor is not present than assistant doctor should be taken as an incharge this point is not taken into consideration yet.

Every IPD patient undergoes surgery.
Ambulance in case of emergency is not considered here.

Also the patient priority in case of emergency is not considered.

5. Assumptions

We have assumed the following things:

Head doctor will do surgery and also treat OPD patients i.e. head doctor is also a surgeon.

All the required doctors will be available when needed.

Nurse is managed by mangement.

Each Nurse works as per allocated bed by the Management.

Only Management can create, update, or delete a doctor.

For IPD patients it is compulsory to buy medicines from the pharmacy of the hospital while OPD patients have a choice that they can take medicines from the pharmacy of the hospital or from somewhere outside.

Total cost of IPD patient is included in the final bill and is paid to the receptionist (where total cost includes all the services provided by the hospital eg charge of room, cost of medicines, doctors visiting cost and other expenses).

Only Receptionist, Lab Incharge and pharmacist can take money in form of cash or cheque.

Every day all the money taken by receptionist and pharmacist are given off to the accountant.

Every Assistant Doctor works under a Head Doctor and it is the responsibility of the head doctor to manage the assistant doctor.

IPD patient does not have a choice he must give his tests to the lab of hospital while OPD patient can do his test inside the lab of hospital or outside the lab of hospital.

All the main instruments are allocated as per operation theatre and it's the responsibility of nurse to maintain that instruments and for lab lab Incharge has the responsibility to maintain its instruments.

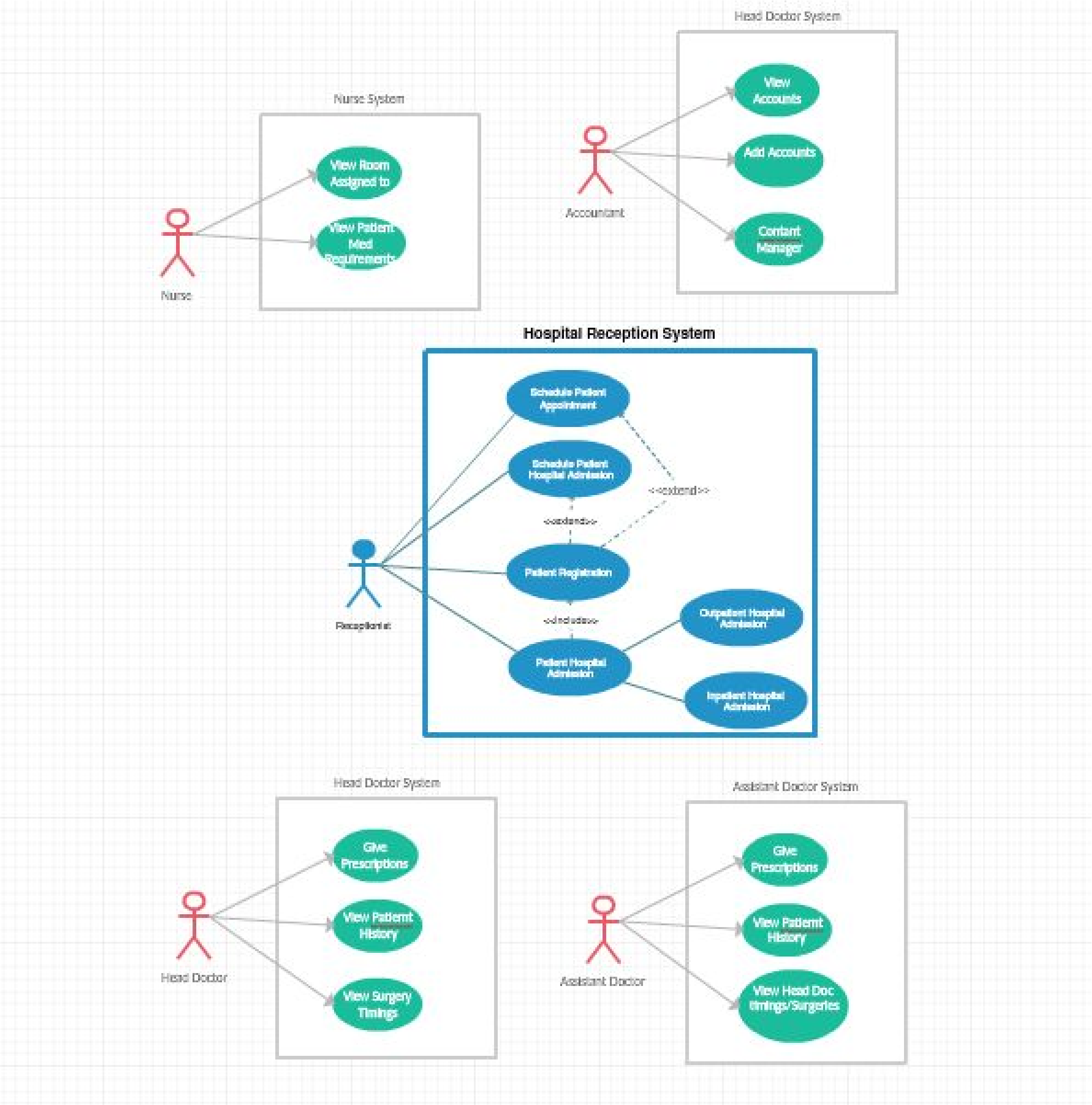
Pathologist is a special type of assistant doctor but here he is considered as lab incharge.

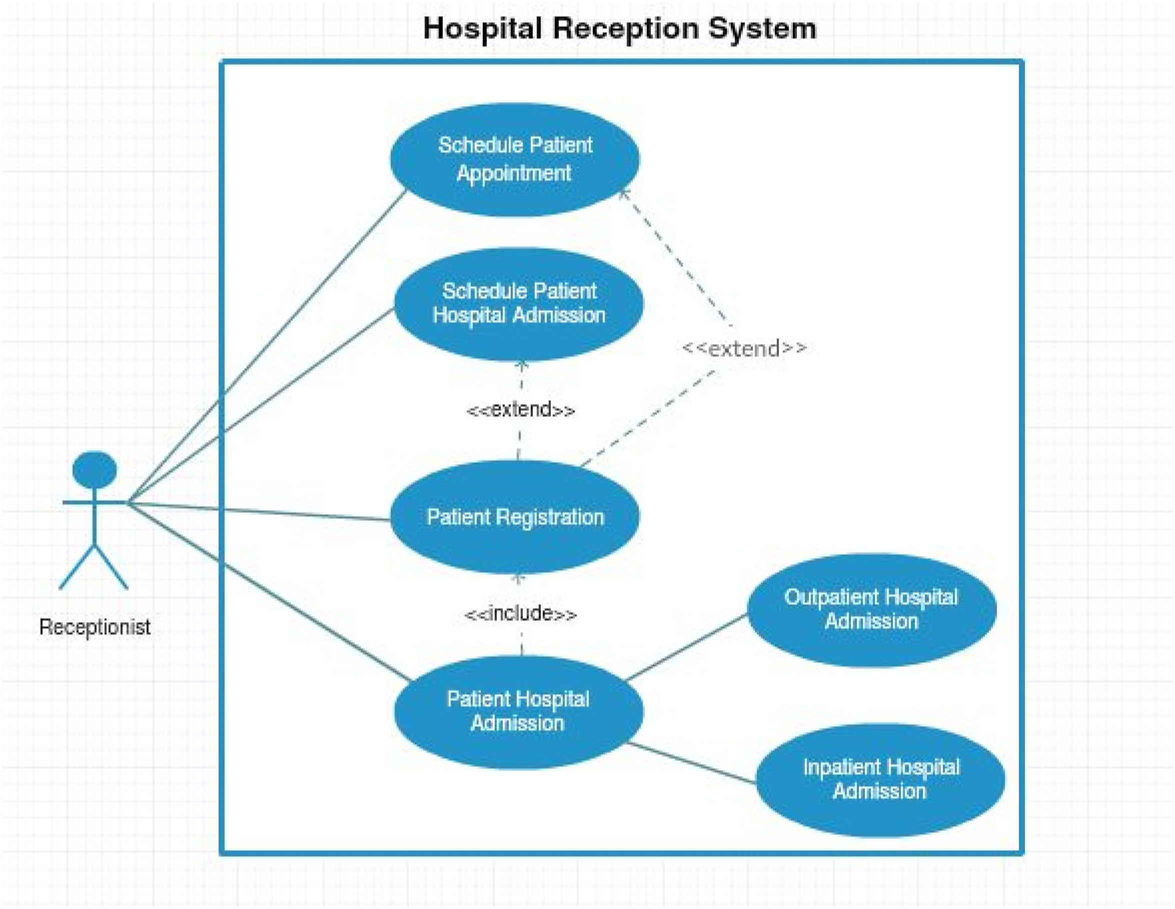
An OPD patient can visit to the hospital only after taking appointment.

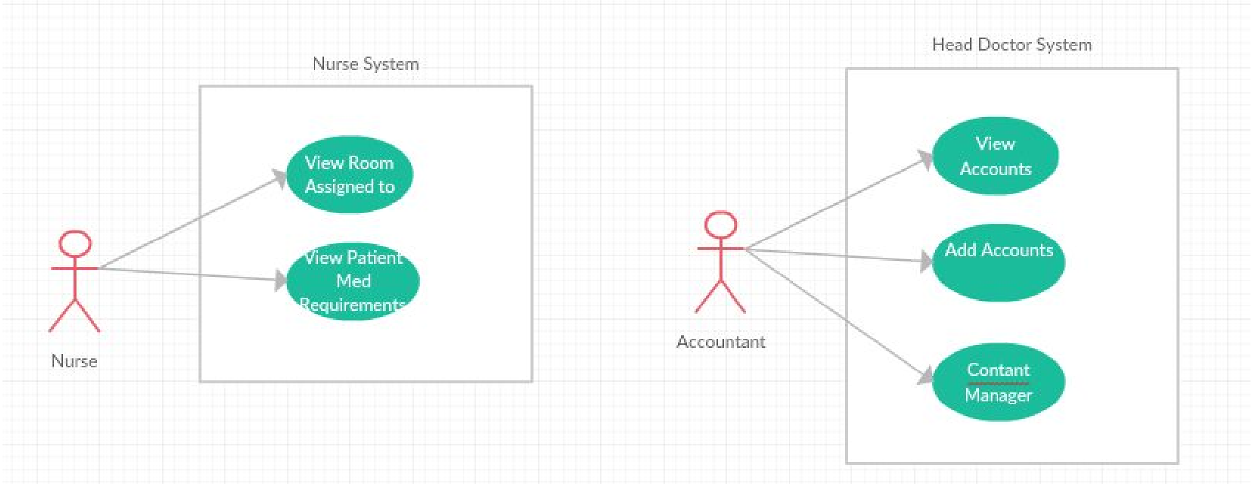
On visiting the hospital on appointment time OPD patient pays the visiting charge to receptionist and receptionist provides bill to the patient in which token no is given to the patients and per token no the patient can visit the doctor.

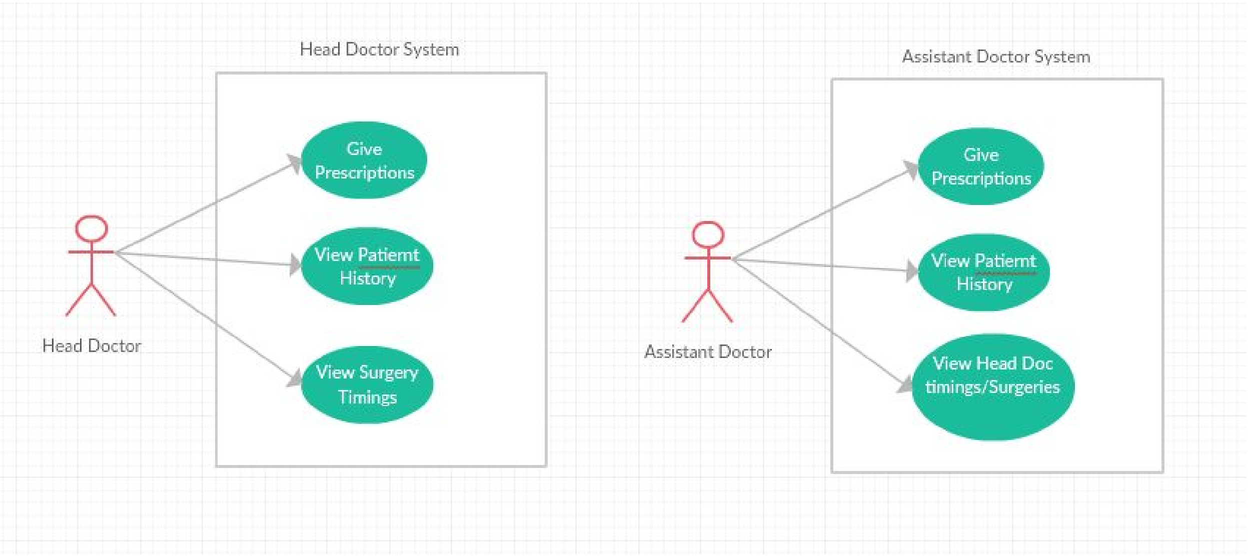
For ICU nurse there is a ratio of nurse to patient which is 1:1 and for Nurse serving normal rooms the ratio is 1:4.

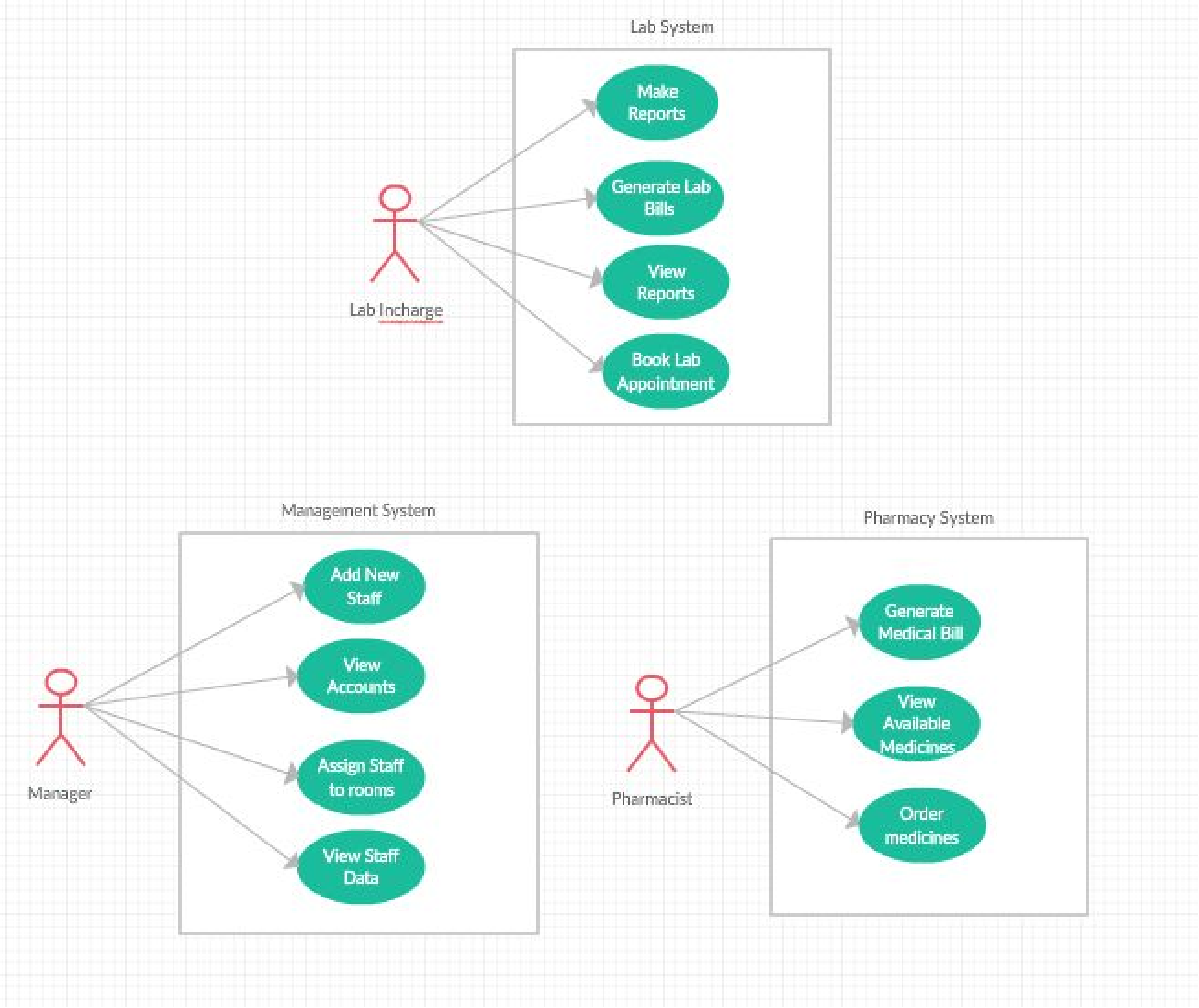
6. Use Case Diagram




Hospital Reception System
Schedule Patient Appointment
Schedule Patient Hospital Admission
<<extend>>
<<extend>>
Patient Registration
<<include>>
Receptionist
Outpatient Hospital Admission
Patient Hospital Admission
Inpatient Hospital Admission

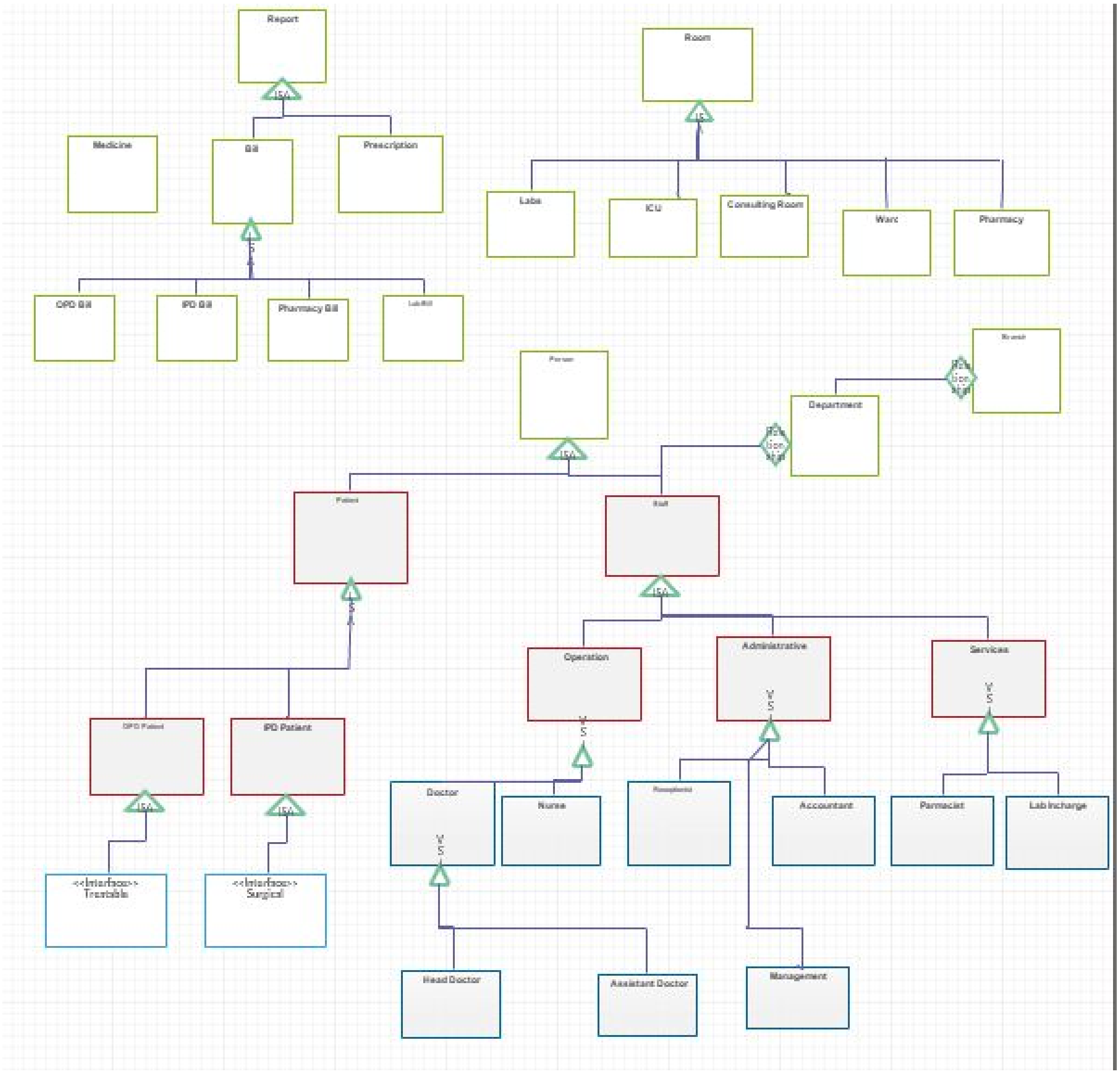

Nurse System
View Room Assigned to
View Patient Med Requirements
Nurse
Head Doctor System
View Accounts
Add Accounts
Contant Manager
Accountant

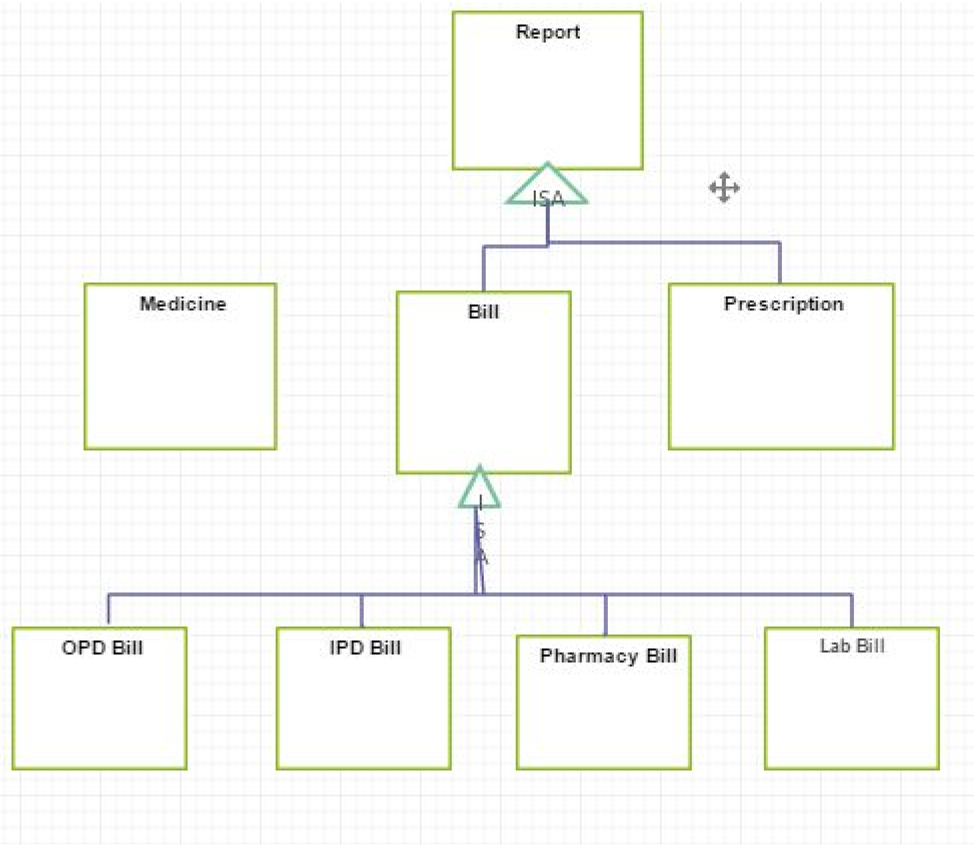

Head Doctor System
Give Prescriptions
View Patiemt History
View Surgery Timings
Head Doctor
Assistant Doctor System
Give Prescriptions
View Patiemt History
View Head Doc timings/Surgeries
Assistant Doctor

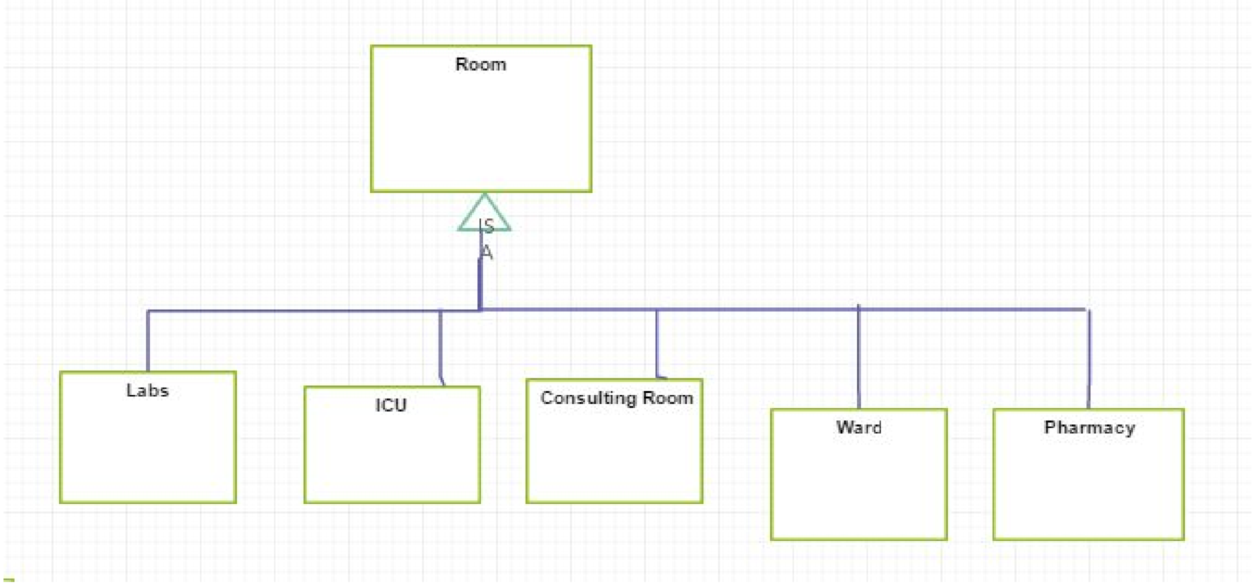

Lab System
Make Reports
Generate Lab Bills
View Reports
Book Lab Appointment
Lab Incharge
Management System
Add New Staff
View Accounts
Assign Staff to rooms
View Staff Data
Manager
Pharmacy System
Generate Medical Bill
View Available Medicines
Order medicines
Pharmacist

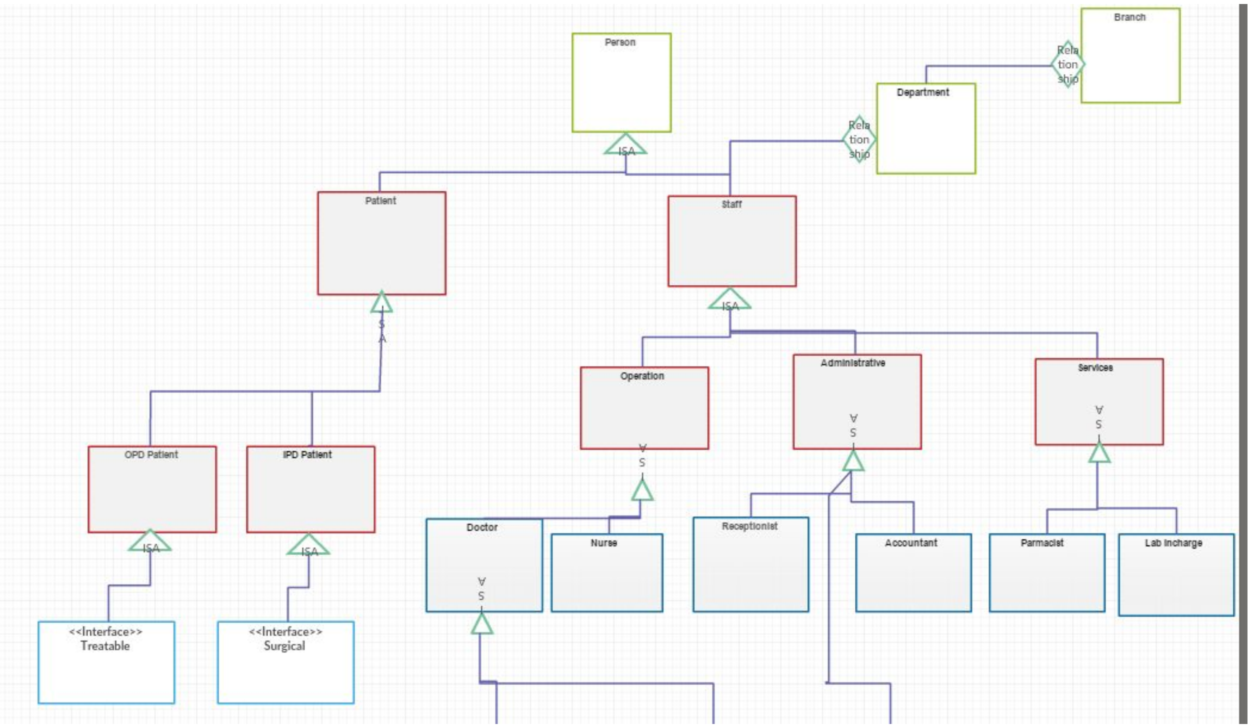
7. Class Diagram

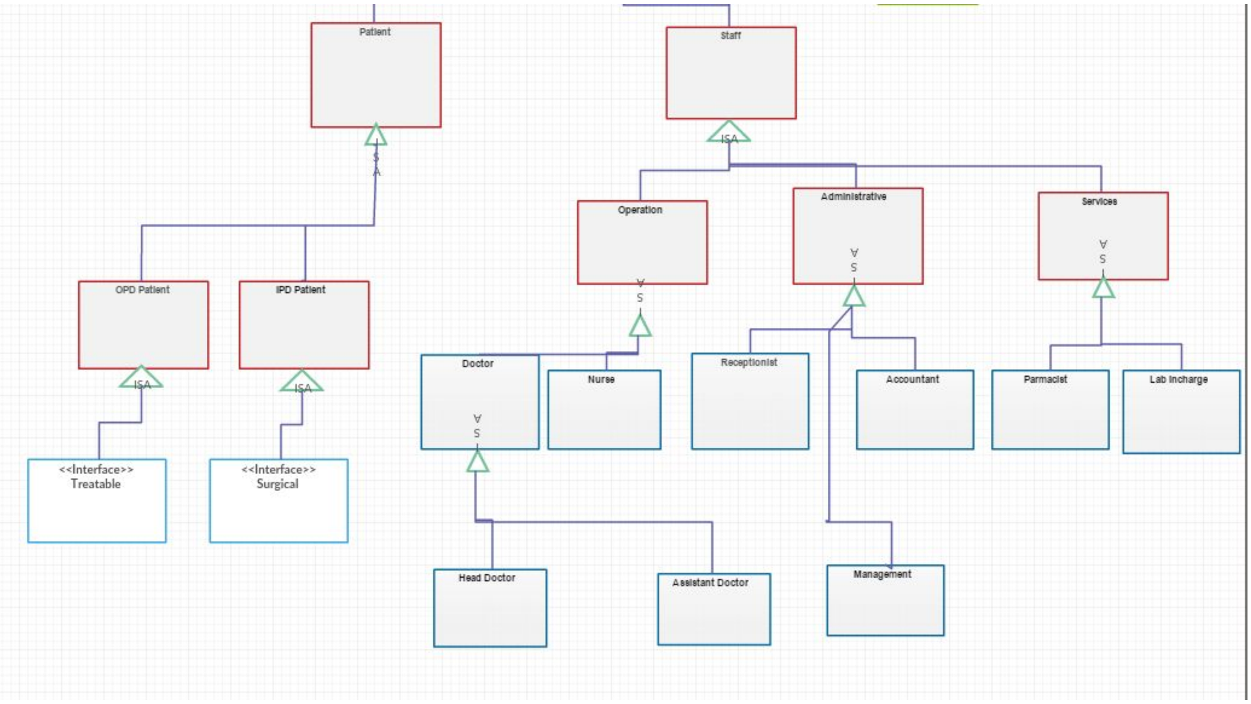
Class Hierarchy:

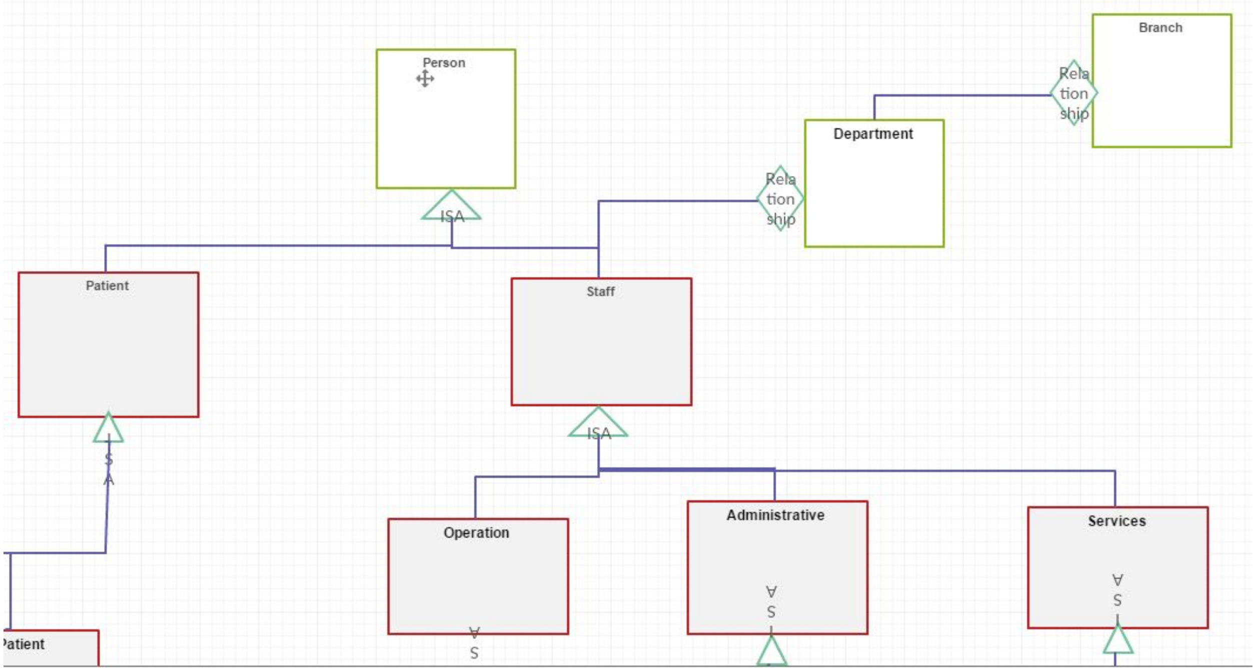


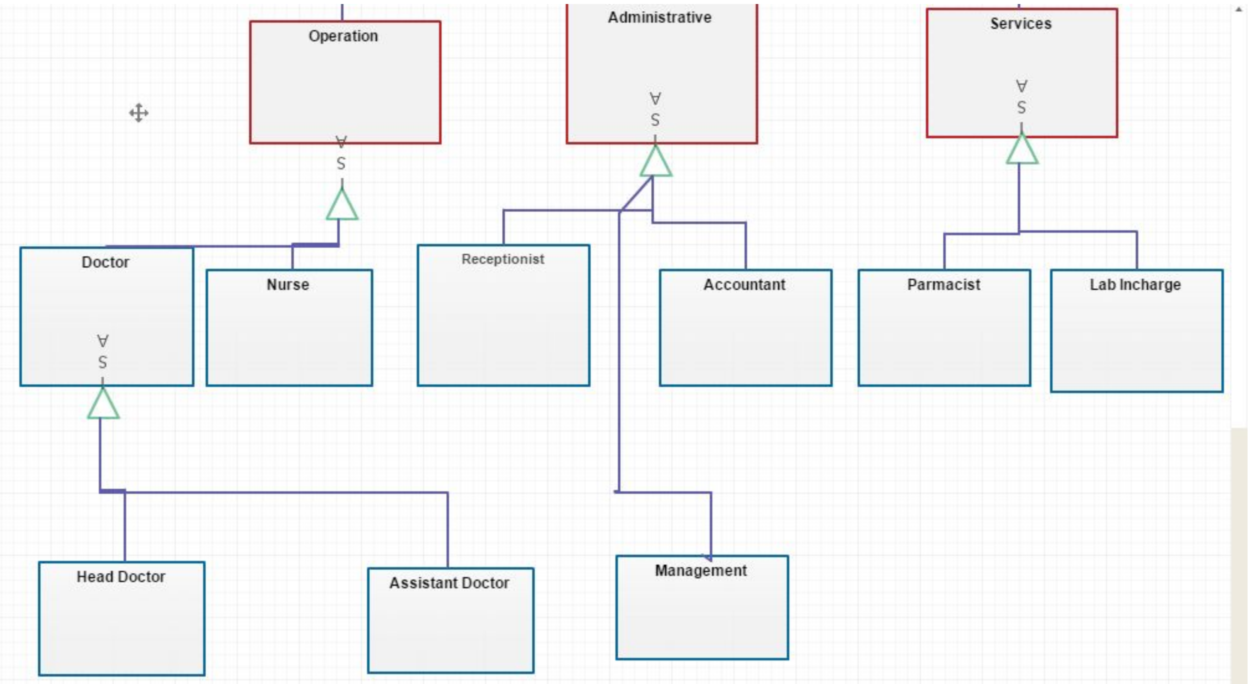

Report
ISA
Medicine
Bill
Prescription
I
S
A
OPD Bill
IPD Bill
Pharmacy Bill
Lab Bill

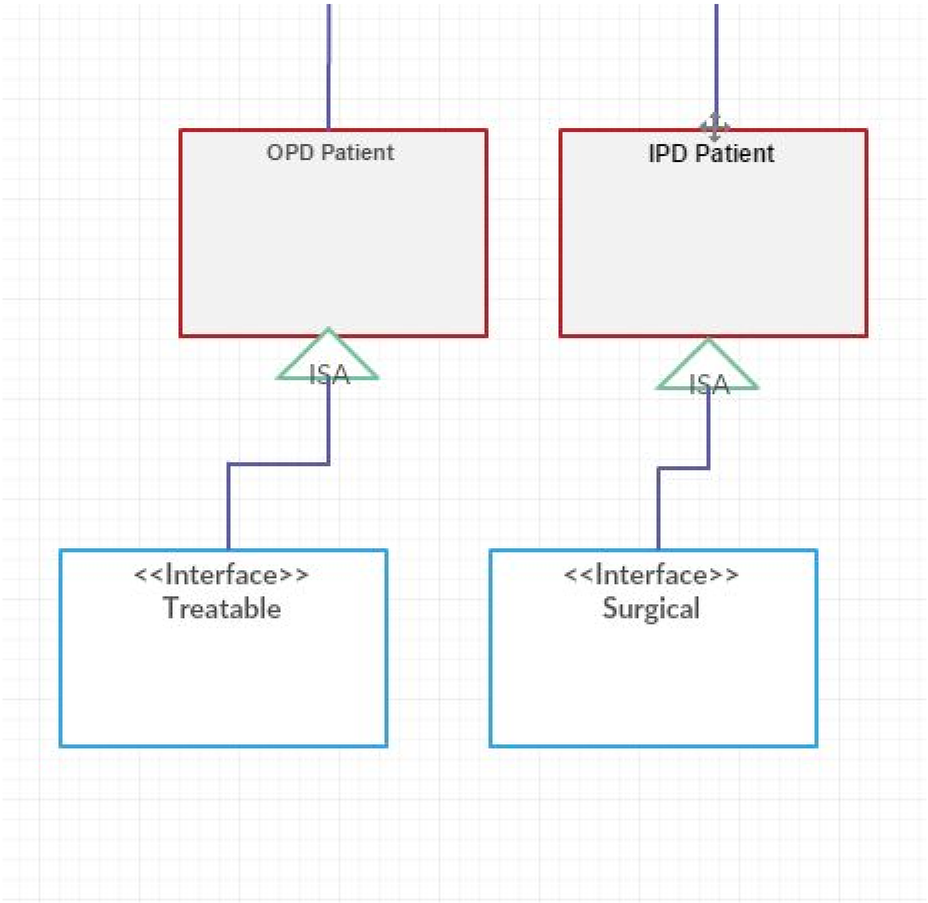

Room
IS
A
Labs
ICU
Consulting Room
Ward
Pharmacy


Person
Branch
Rela
tion
ship
Department
Rela
tion
ship
ISA
Patient
Staff
I
S
A
ISA
Operation
Administrative
Services
A
S
OPD Patient
IPD Patient
Doctor
Nurse
Receptionist
Accountant
Parmacist
Lab Incharge
ISA
<<Interface>>
Treatable
<<Interface>>
Surgical


Patient
Staff
I
S
A
ISA
Operation
Administrative
Services
A
S
OPD Patient
IPD Patient
Doctor
Nurse
Receptionist
Accountant
Parmacist
Lab Incharge
ISA
<<Interface>>
Treatable
<<Interface>>
Surgical
Head Doctor
Assistant Doctor
Management

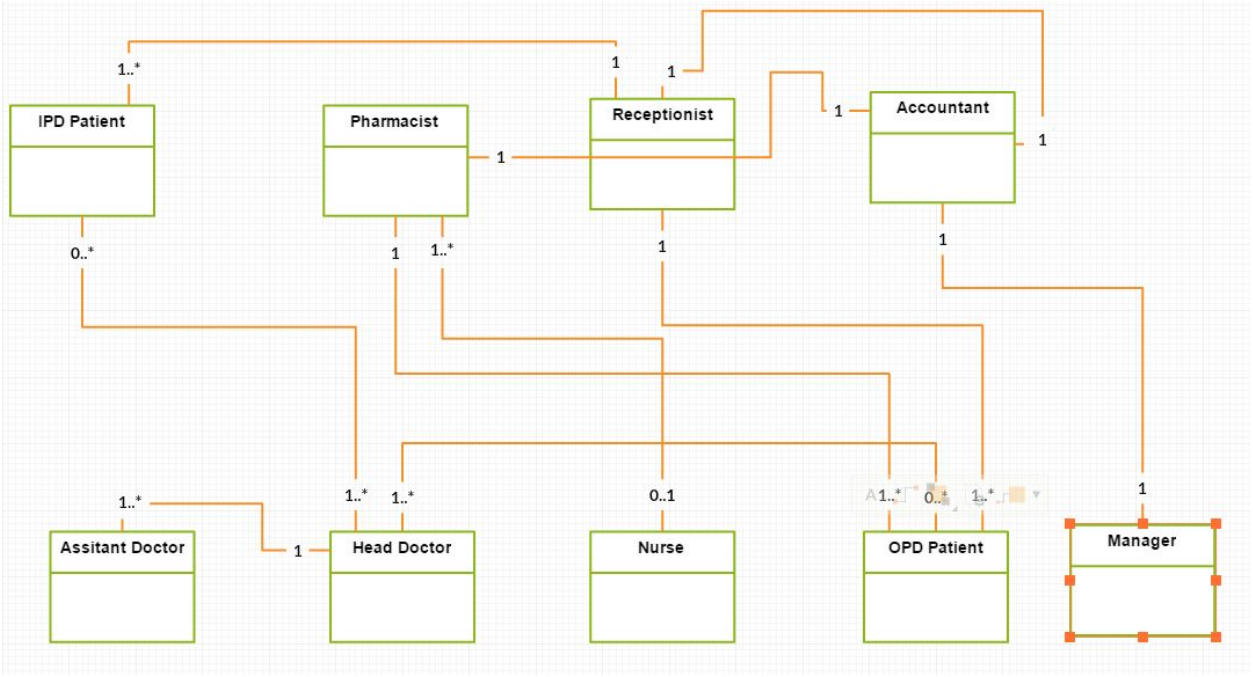

Person
Branch
Rela
tion
ship
Department
Rela
tion
ship
ISA
Patient
Staff
S
A
ISA
Operation
Administrative
Services
∀
S
∀
S
∀
S
atient


Operation
Administrative
Services
∀
S
∀
S
∀
S
Doctor
Nurse
Receptionist
Accountant
Parmacist
Lab Incharge
∀
S
Head Doctor
Assistant Doctor
Management

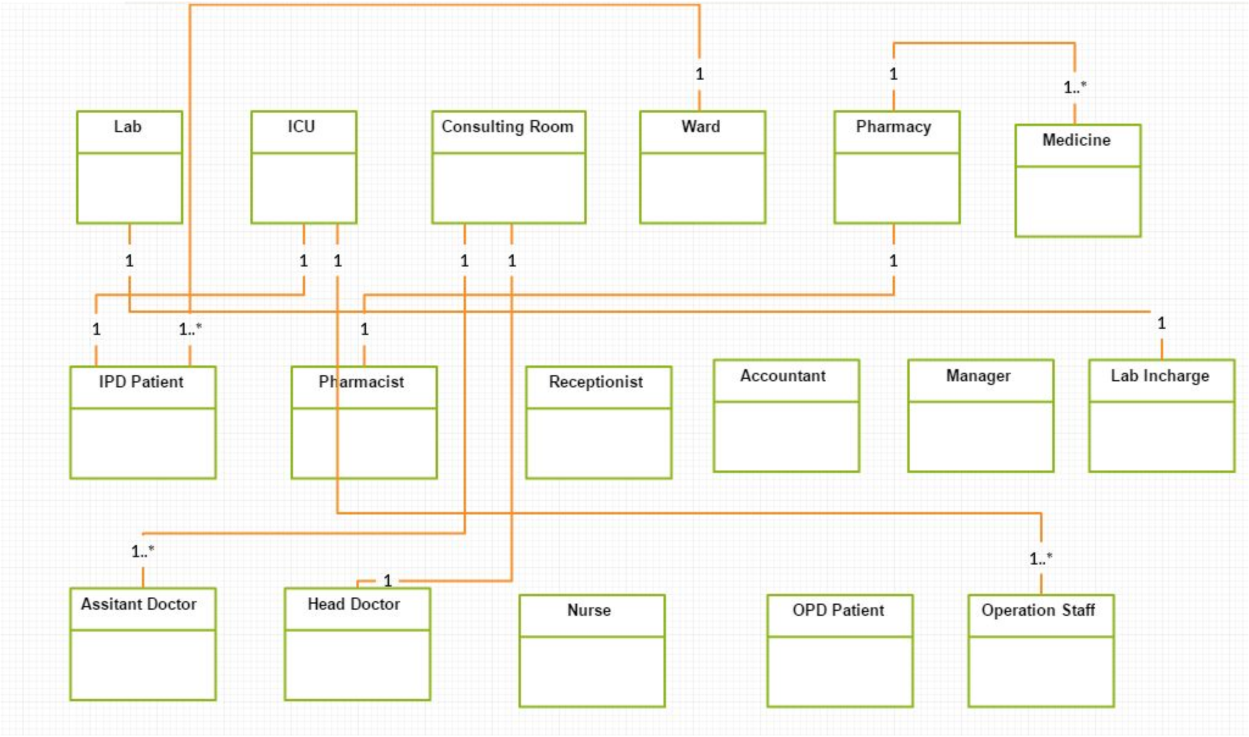


Class Relationships:

Living - Living Relationships

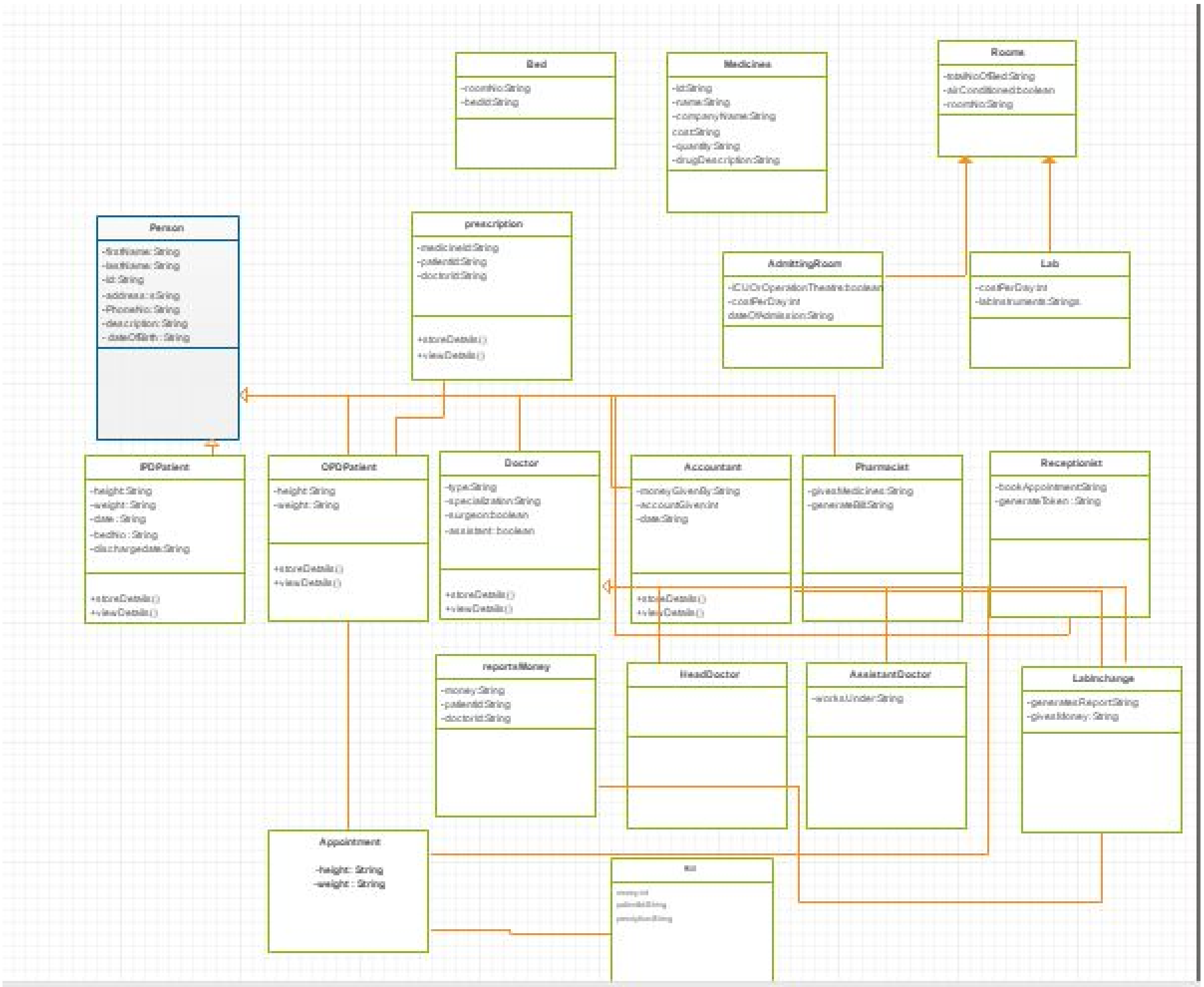


Living - Non Living relationships

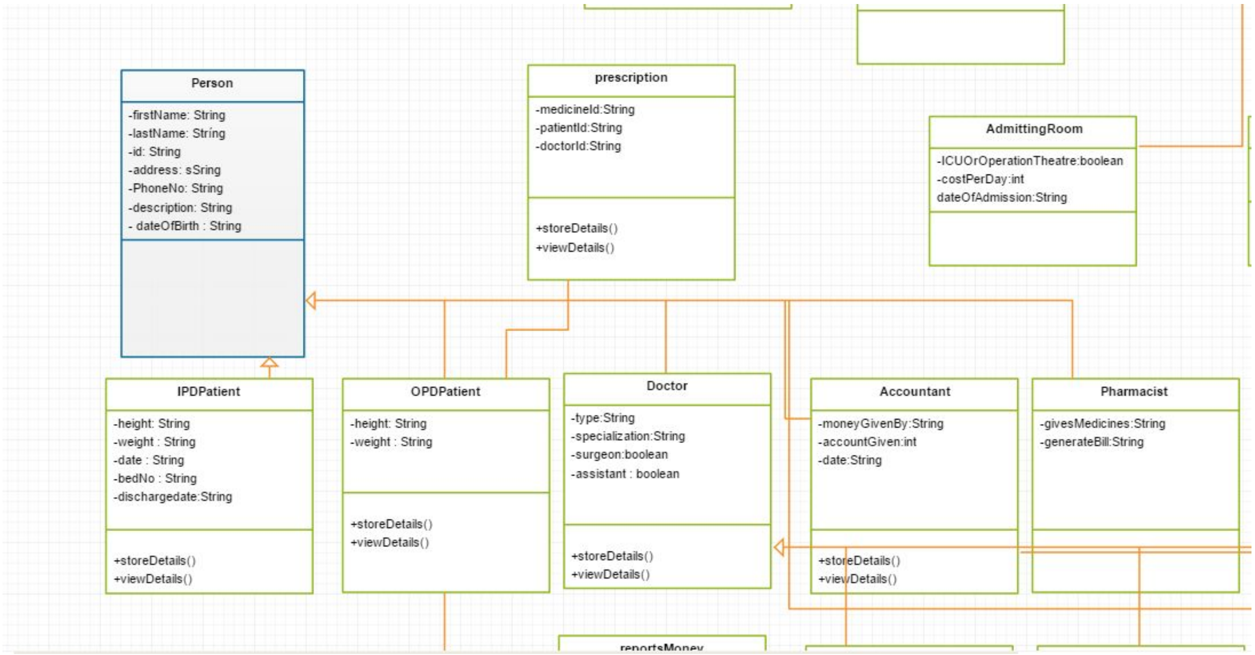


Association Class

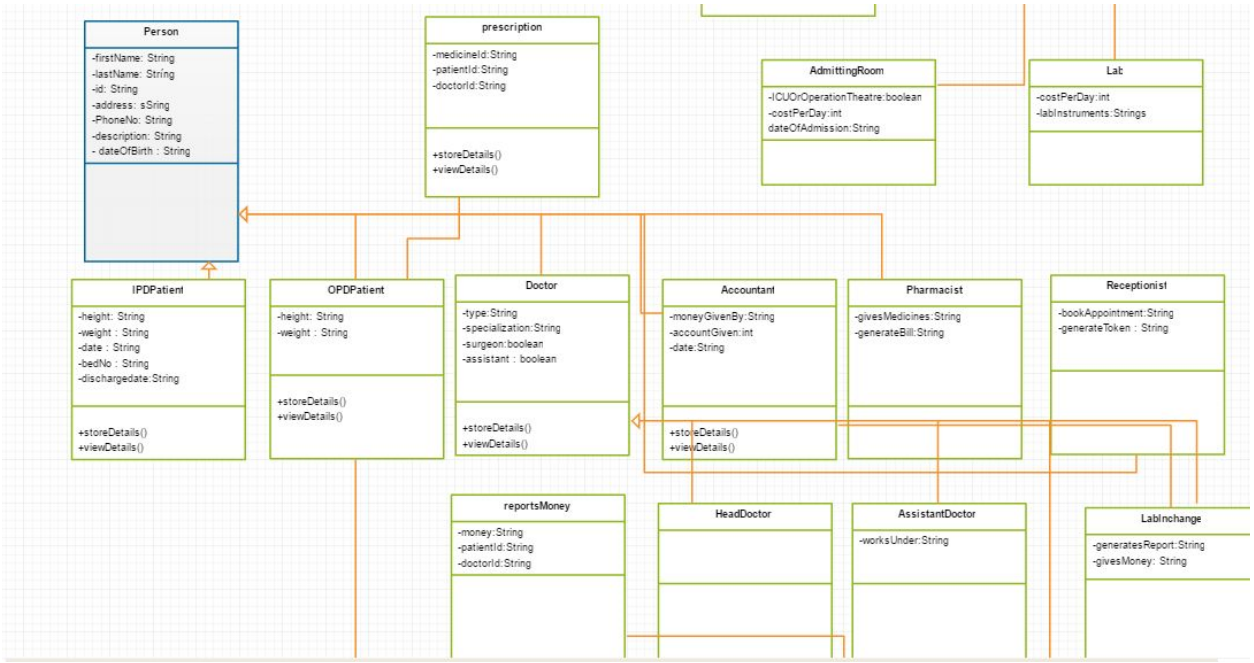
ClassName : reportsMoney, Prescription, Appointment, Bill



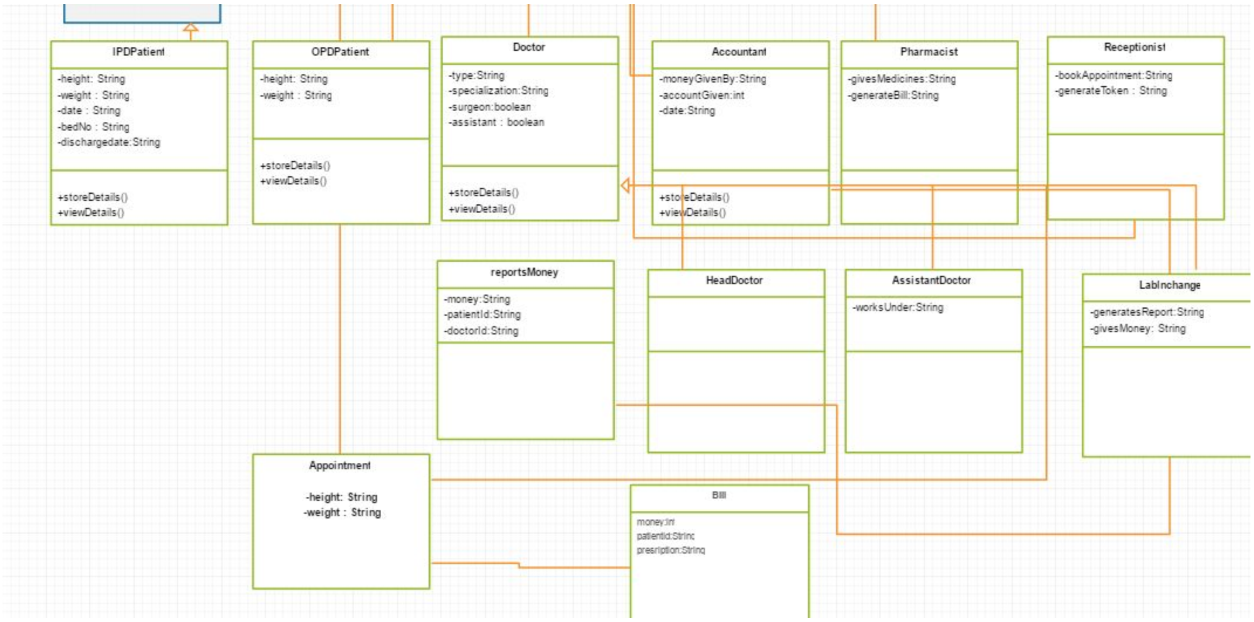
Prescription



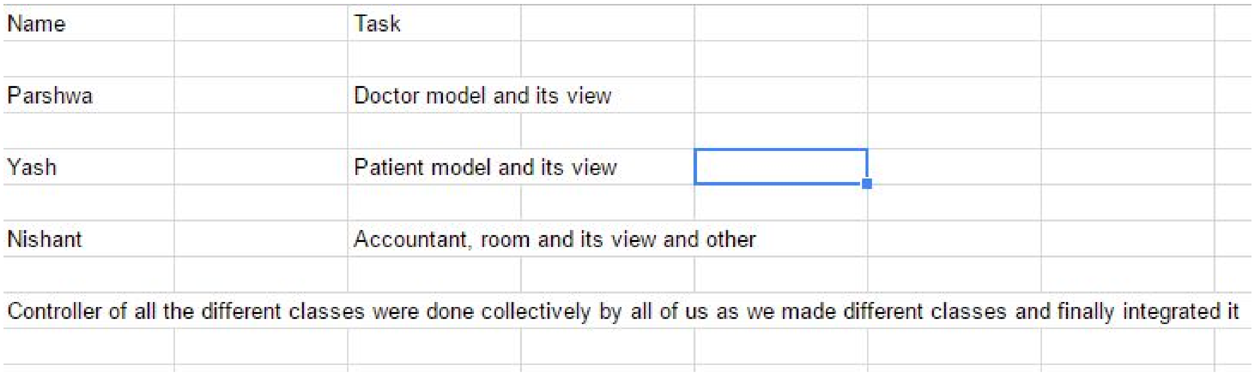
reportsMoney



Appointment and Bill



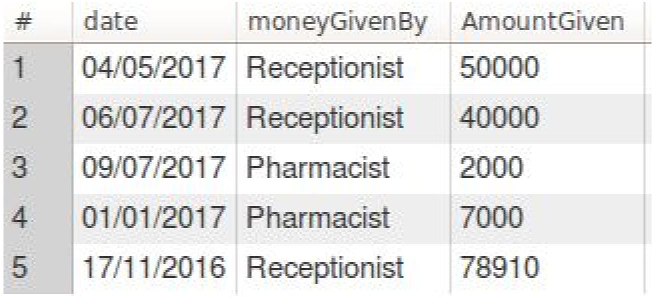
8.Division of Task among Team Members



Name	Task
Parshwa	Doctor model and its view
Yash	Patient model and its view
Nishant	Accountant, room and its view and other
Controller of all the different classes were done collectively by all of us as we made different classes and finally integrated it	

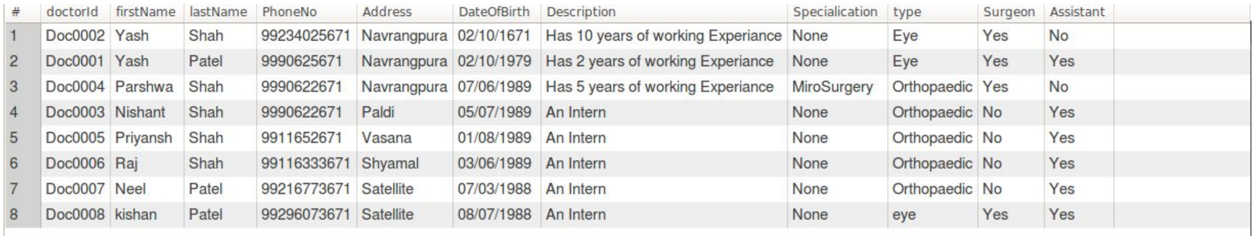
9. Data Storage Structure

Account Data:



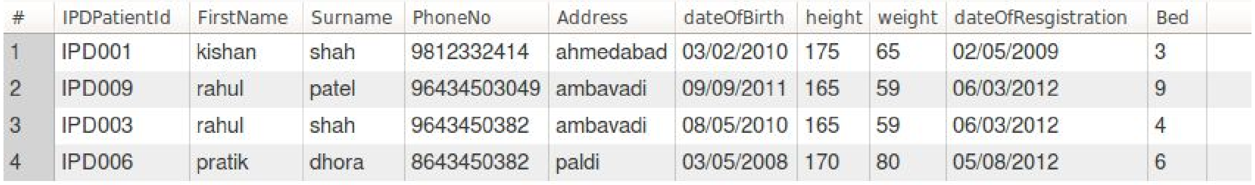
#	date	moneyGivenBy	AmountGiven
1	04/05/2017	Receptionist	50000
2	06/07/2017	Receptionist	40000
3	09/07/2017	Pharmacist	2000
4	01/01/2017	Pharmacist	7000
5	17/11/2016	Receptionist	78910

Doctor Data:



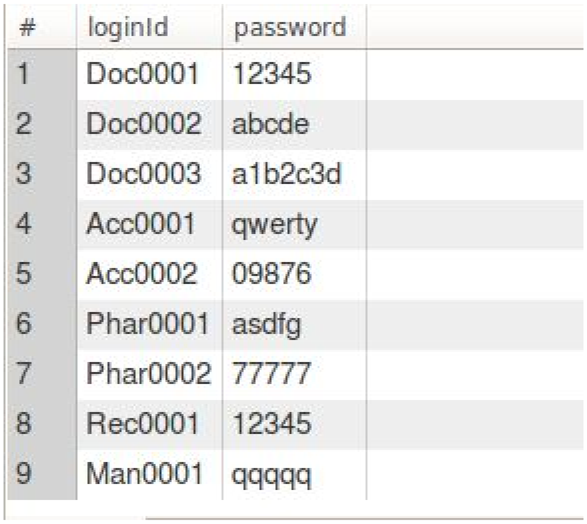
#	doctorId	firstName	lastName	PhoneNo	Address	DateOfBirth	Description	Specialication	type	Surgeon	Assistant
1	Doc0002	Yash	Shah	99234025671	Navrangpura	02/10/1671	Has 10 years of working Experiance	None	Eye	Yes	No
2	Doc0001	Yash	Patel	9990625671	Navrangpura	02/10/1979	Has 2 years of working Experiance	None	Eye	Yes	Yes
3	Doc0004	Parshwa	Shah	9990622671	Navrangpura	07/06/1989	Has 5 years of working Experiance	MiroSurgery	Orthopaedic	Yes	No
4	Doc0003	Nishant	Shah	9990622671	Paldi	05/07/1989	An Intern	None	Orthopaedic	No	Yes
5	Doc0005	Priyansh	Shah	9911652671	Vasana	01/08/1989	An Intern	None	Orthopaedic	No	Yes
6	Doc0006	Raj	Shah	99116333671	Shyamal	03/06/1989	An Intern	None	Orthopaedic	No	Yes
7	Doc0007	Neel	Patel	99216773671	Satellite	07/03/1988	An Intern	None	Orthopaedic	No	Yes
8	Doc0008	kishan	Patel	99296073671	Satellite	08/07/1988	An Intern	None	eye	Yes	Yes

IPD Patient Data:



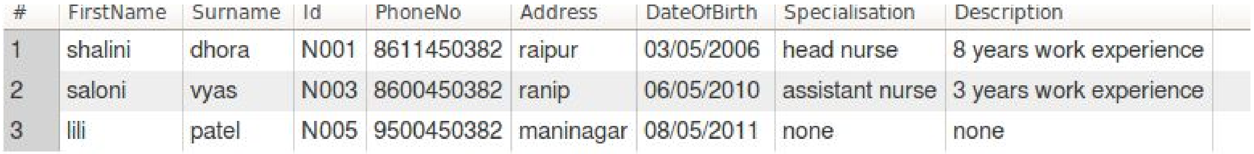
#	IPDPatientId	FirstName	Surname	PhoneNo	Address	dateOfBirth	height	weight	dateOfResgistration	Bed
1	IPD001	kishan	shah	9812332414	ahmedabad	03/02/2010	175	65	02/05/2009	3
2	IPD009	rahul	patel	96434503049	ambavadi	09/09/2011	165	59	06/03/2012	9
3	IPD003	rahul	shah	9643450382	ambavadi	08/05/2010	165	59	06/03/2012	4
4	IPD006	pratik	dhora	8643450382	paldi	03/05/2008	170	80	05/08/2012	6

Login Data :



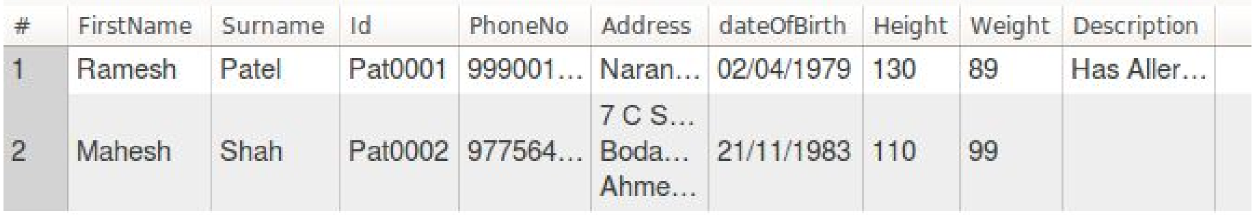
#	loginId	password
1	Doc0001	12345
2	Doc0002	abcde
3	Doc0003	a1b2c3d
4	Acc0001	qwerty
5	Acc0002	09876
6	Phar0001	asdfg
7	Phar0002	77777
8	Rec0001	12345
9	Man0001	qqqqq

Nurse Data:



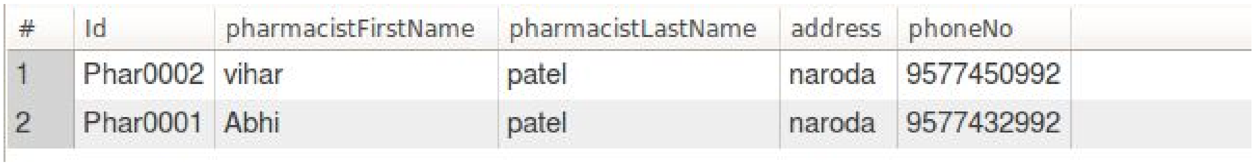
#	FirstName	Surname	Id	PhoneNo	Address	DateOfBirth	Specialisation	Description
1	shalini	dhora	N001	8611450382	raipur	03/05/2006	head nurse	8 years work experience
2	saloni	vyas	N003	8600450382	ranip	06/05/2010	assistant nurse	3 years work experience
3	lili	patel	N005	9500450382	maninagar	08/05/2011	none	none

OPD Patient data:



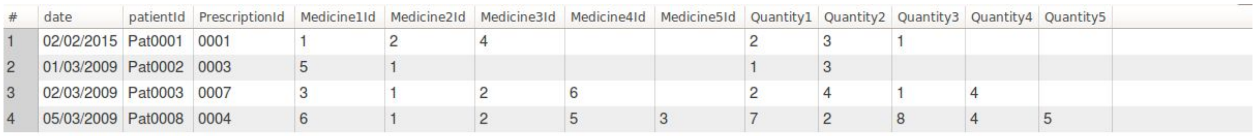
#	FirstName	Surname	Id	PhoneNo	Address	dateOfBirth	Height	Weight	Description
1	Ramesh	Patel	Pat0001	999001…	Naran…	02/04/1979	130	89	Has Aller…
2	Mahesh	Shah	Pat0002	977564…	7 C S… Boda… Ahme…	21/11/1983	110	99	

Pharmacist data



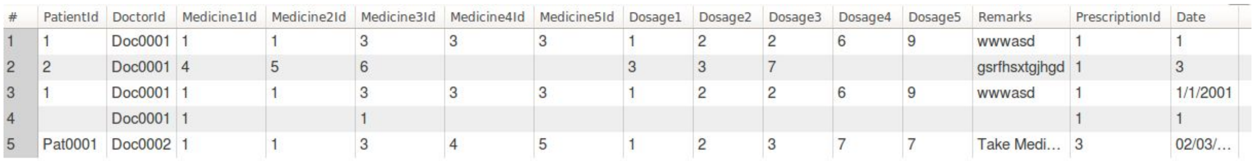
#	Id	pharmacistFirstName	pharmacistLastName	address	phoneNo
1	Phar0002	vihar	patel	naroda	9577450992
2	Phar0001	Abhi	patel	naroda	9577432992

Pharmacy Bill Data



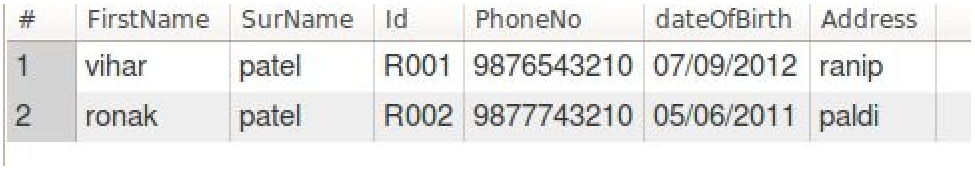
#	date	patientId	PrescriptionId	Medicine1Id	Medicine2Id	Medicine3Id	Medicine4Id	Medicine5Id	Quantity1	Quantity2	Quantity3	Quantity4	Quantity5
1	02/02/2015	Pat0001	0001	1	2	4			2	3	1		
2	01/03/2009	Pat0002	0003	5	1				1	3			
3	02/03/2009	Pat0003	0007	3	1	2	6		2	4	1	4	
4	05/03/2009	Pat0008	0004	6	1	2	5	3	7	2	8	4	5

Prescription Data:



#	PatientId	DoctorId	Medicine1Id	Medicine2Id	Medicine3Id	Medicine4Id	Medicine5Id	Dosage1	Dosage2	Dosage3	Dosage4	Dosage5	Remarks	PrescriptionId	Date
1	1	Doc0001	1	1	3	3	3	1	2	2	6	9	wwwasd	1	1
2	2	Doc0001	4	5	6			3	3	7			gsrfhsxtgjhgd	1	3
3	1	Doc0001	1	1	3	3	3	1	2	2	6	9	wwwasd	1	1/1/2001
4		Doc0001	1		1									1	1
5	Pat0001	Doc0002	1	1	3	4	5	1	2	3	7	7	Take Medi...	3	02/03/...

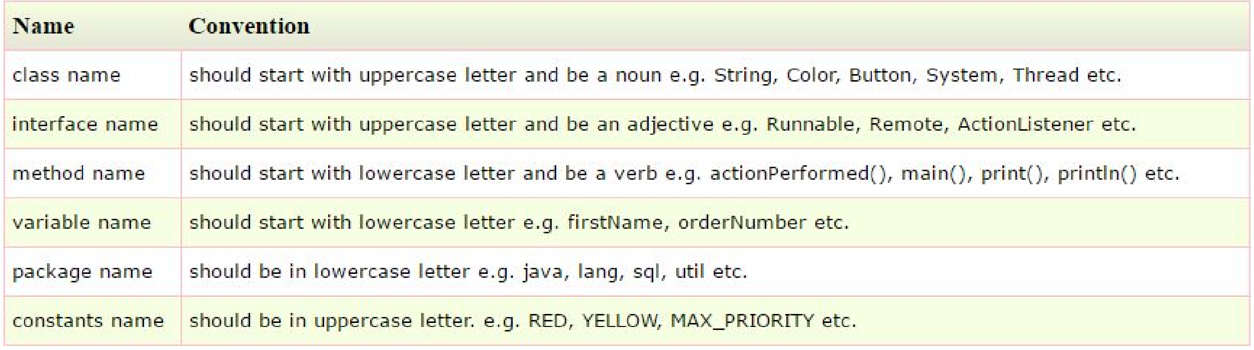
Receptionist Data:



#	FirstName	SurName	Id	PhoneNo	dateOfBirth	Address
1	vihar	patel	R001	9876543210	07/09/2012	ranip
2	ronak	patel	R002	9877743210	05/06/2011	paldi

Other tables are made as concrete classes are made and some tables made are of association classes.

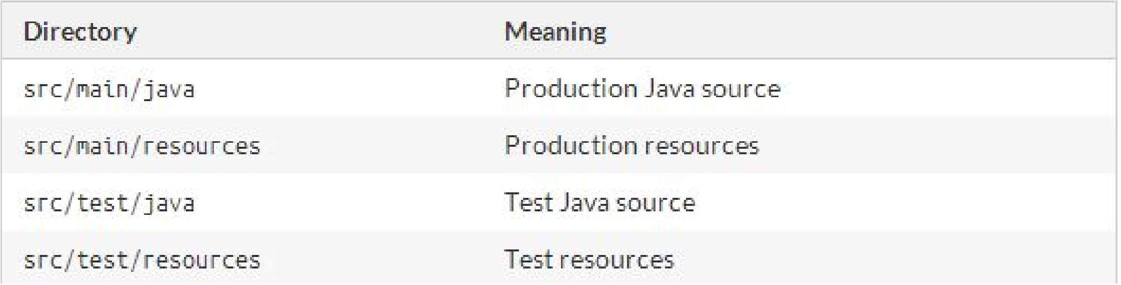
9. Naming Conventions



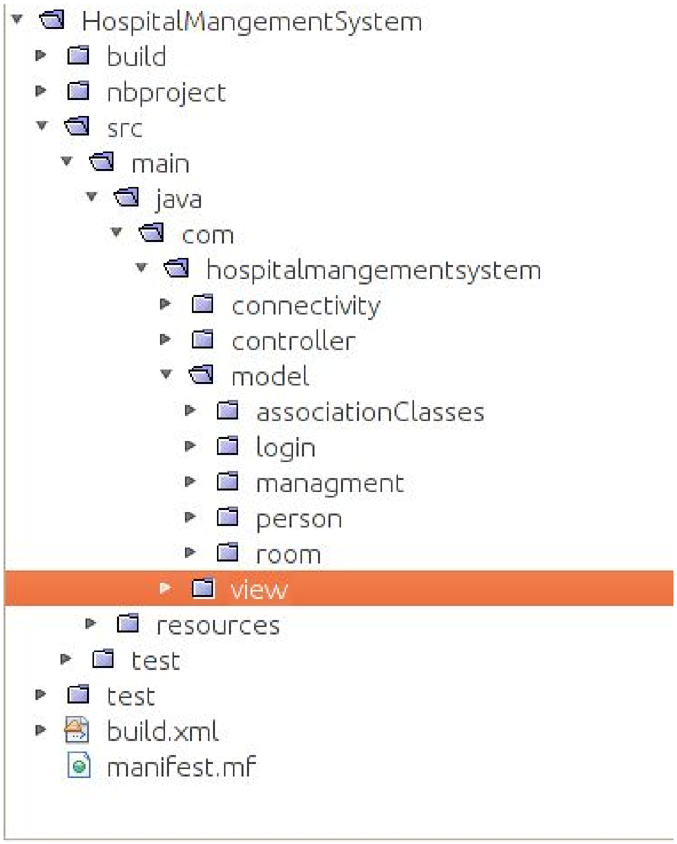
Name	Convention
class name	should start with uppercase letter and be a noun e.g. String, Color, Button, System, Thread etc.
interface name	should start with uppercase letter and be an adjective e.g. Runnable, Remote, ActionListener etc.
method name	should start with lowercase letter and be a verb e.g. actionPerformed(), main(), print(), println() etc.
variable name	should start with lowercase letter e.g. firstName, orderNumber etc.
package name	should be in lowercase letter e.g. java, lang, sql, util etc.
constants name	should be in uppercase letter. e.g. RED, YELLOW, MAX_PRIORITY etc.

Package Structure

Gradle/Maven Standard directory structure is used.



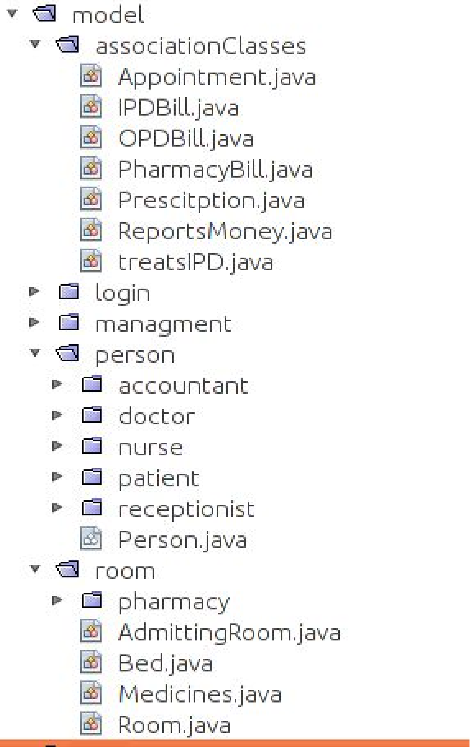
Directory	Meaning
src/main/java	Production Java source
src/main/resources	Production resources
src/test/java	Test Java source
src/test/resources	Test resources



```
HospitalMangementSystem
  build
  nbproject
  src
    main
      java
        com
          hospitalmangementsystem
            connectivity
            controller
            model
              associationClasses
              login
              managment
              person
              room
            view
      resources
    test
  test
  build.xml
  manifest.mf
```

` Model

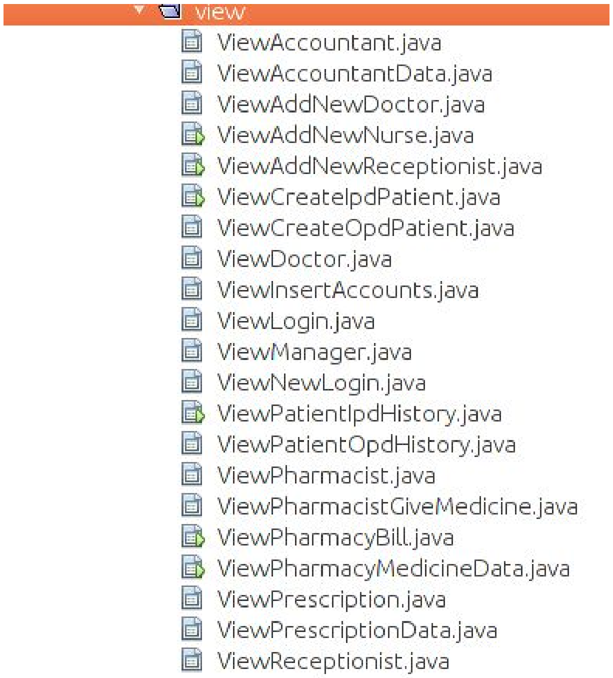
Standard naming conventions were used.



```
model
  associationClasses
    Appointment.java
    IPDBill.java
    OPDBill.java
    PharmacyBill.java
    Prescitption.java
    ReportsMoney.java
    treatsIPD.java
  login
  managment
  person
    accountant
    doctor
    nurse
    patient
    receptionist
    Person.java
  room
    pharmacy
    AdmittingRoom.java
    Bed.java
    Medicines.java
    Room.java
```

View

Naming Convention: View(ClassName)



view

- ViewAccountant.java
- ViewAccountantData.java
- ViewAddNewDoctor.java
- ViewAddNewNurse.java
- ViewAddNewReceptionist.java
- ViewCreateIpdPatient.java
- ViewCreateOpdPatient.java
- ViewDoctor.java
- ViewInsertAccounts.java
- ViewLogin.java
- ViewManager.java
- ViewNewLogin.java
- ViewPatientIpdHistory.java
- ViewPatientOpdHistory.java
- ViewPharmacist.java
- ViewPharmacistGiveMedicine.java
- ViewPharmacyBill.java
- ViewPharmacyMedicineData.java
- ViewPrescription.java
- ViewPrescriptionData.java
- ViewReceptionist.java

View GUI Variables

Naming Convention:

Button: btnVariableName

TextField: txtVariableName

RadioBtn: radioBtnVariableName

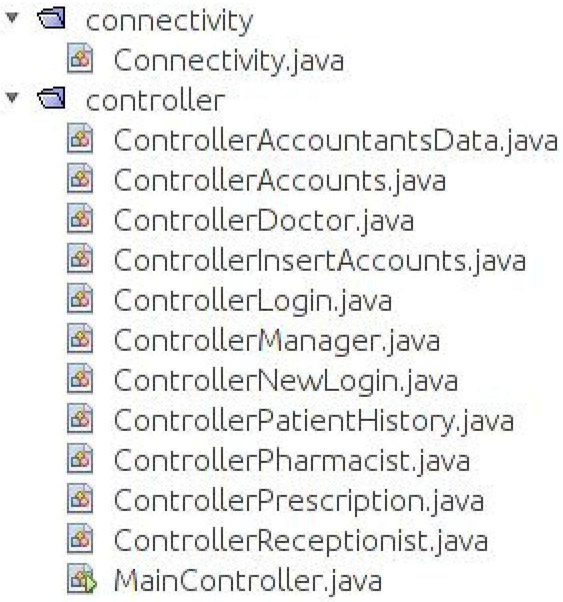
CheckBox: checkBtnVariableName

Label: lblVariableName

Controller

Naming Convention:

Class Name : Controller(ViewClassName)

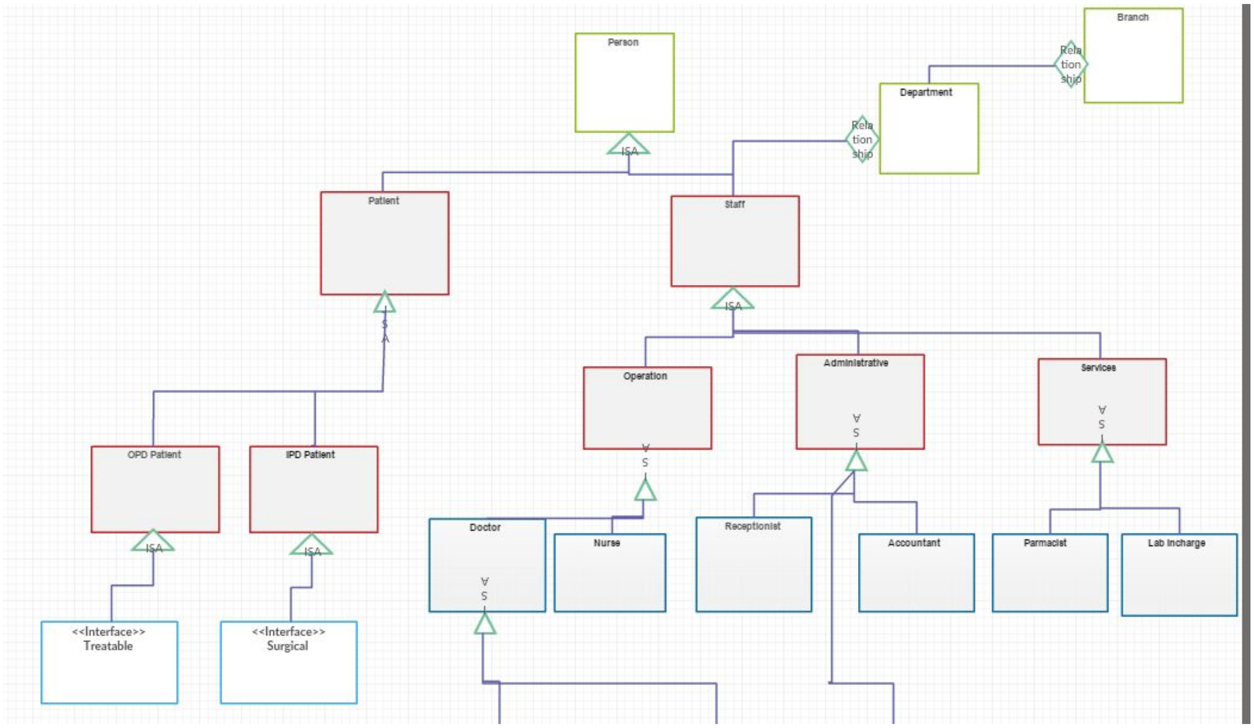


```
connectivity
    Connectivity.java
controller
    ControllerAccountantsData.java
    ControllerAccounts.java
    ControllerDoctor.java
    ControllerInsertAccounts.java
    ControllerLogin.java
    ControllerManager.java
    ControllerNewLogin.java
    ControllerPatientHistory.java
    ControllerPharmacist.java
    ControllerPrescription.java
    ControllerReceptionist.java
    MainController.java
```

10. Object Oriented Approach in Design and Coding

1. **Inheritance**

   Inheritance results in less duplicate code.



Person:

Use:

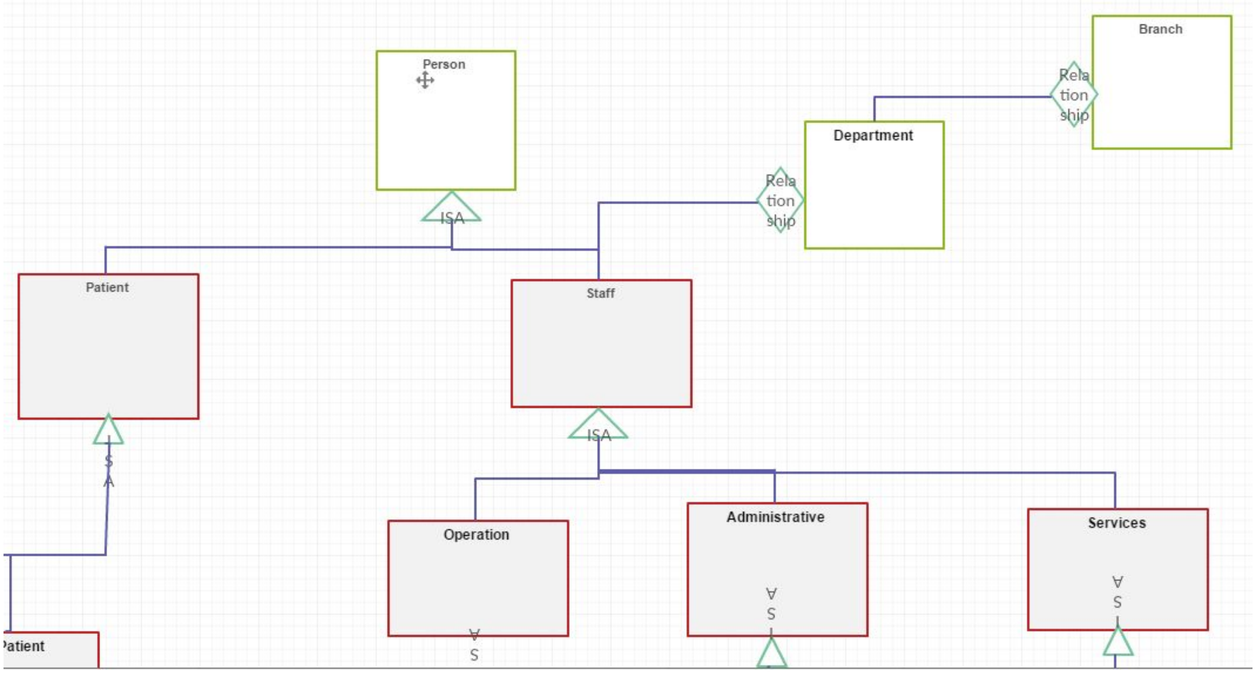
Every Living entity involved with the hospital will inherit the person class.

Advantage

This reduces duplicate code.

Alternatives:

To Have the person code in every class.



Staff:

Use:

Operation,Administrative and Services inherit the Staff Class.

Advantage

This separates out the the different sectors. If there is a problem in one Sector then it can be resolved quickly.

Alternatives:

All sectors in the same class.

Department and branch (Has A):

Use:

Department contains Staff and Branch Contains Department .

Advantage

This makes it easier to create new Departments and Branch. This will also Make resolving a problem for a particular branch easier.

Alternatives:

Not having Department and Branch class.

Patient

Use:

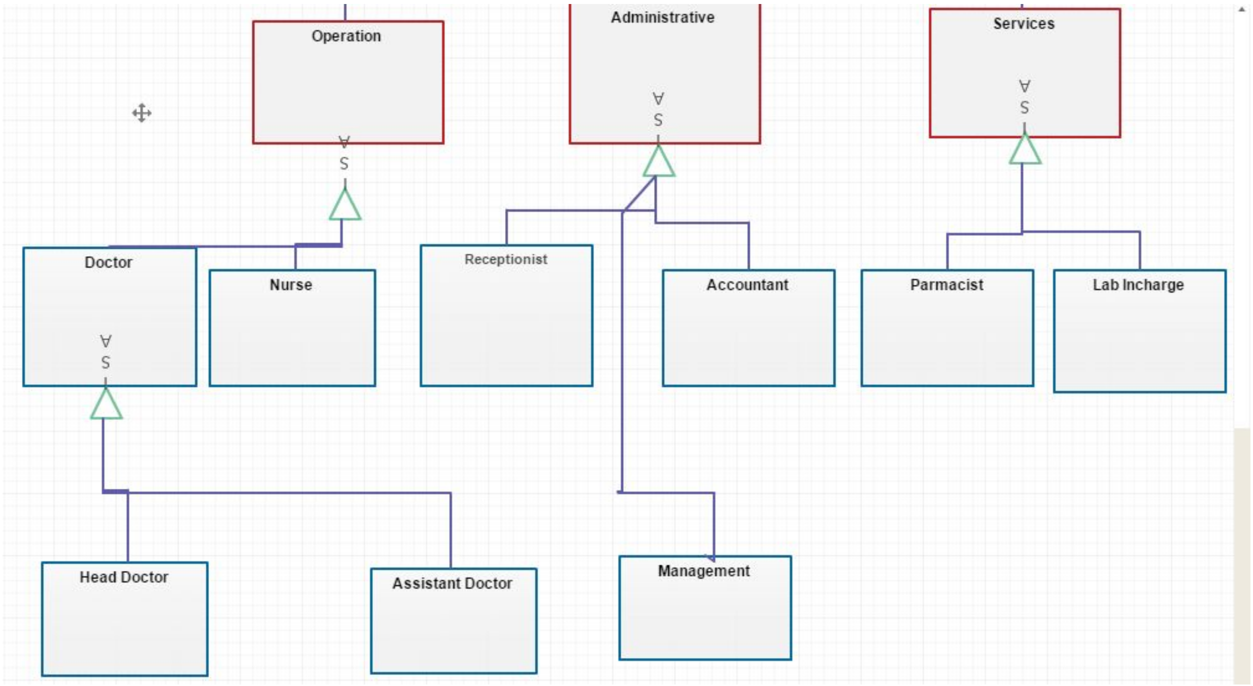
Patient has 2 child classes: OPD Patient and IPD Patient.

Advantage

Both type of patients have different types of ID. And involve very different roles. Hospital have a system wherein every patient has a Opd ID and the ones which need to be admitted are transferred from OPD to IPD and given IPD ID.

Alternatives:

Have only a patient class with no child classes**.**



Operation Staff

Use:

Operation Staff has 2 child classes: Doctor and Nurse.

Advantage

This makes it much easier to link ICU room and the operation Staff.

Alternatives:

Have no operation staff parent class.

Doctor:

Use:

Doctor has 2 sub classes: Head Doctor and Assistant Doctor.

Advantage

Head Doctor and assistant Doctor have very different relationships with Other entities. Head Doctor has a boolean field for surgeon.

Alternatives:

No head Doctor and Assistant Doctor , only a single Doctor class.

Services Staff:

Use:

Services staff has 2 child classes : Lab Incharge and pharmacist.

Advantage

Making the Services Staff separate will help in adding different services In the future and also in using external services in hospital.

Alternatives:

Having Services Staff in the Administrative staff.

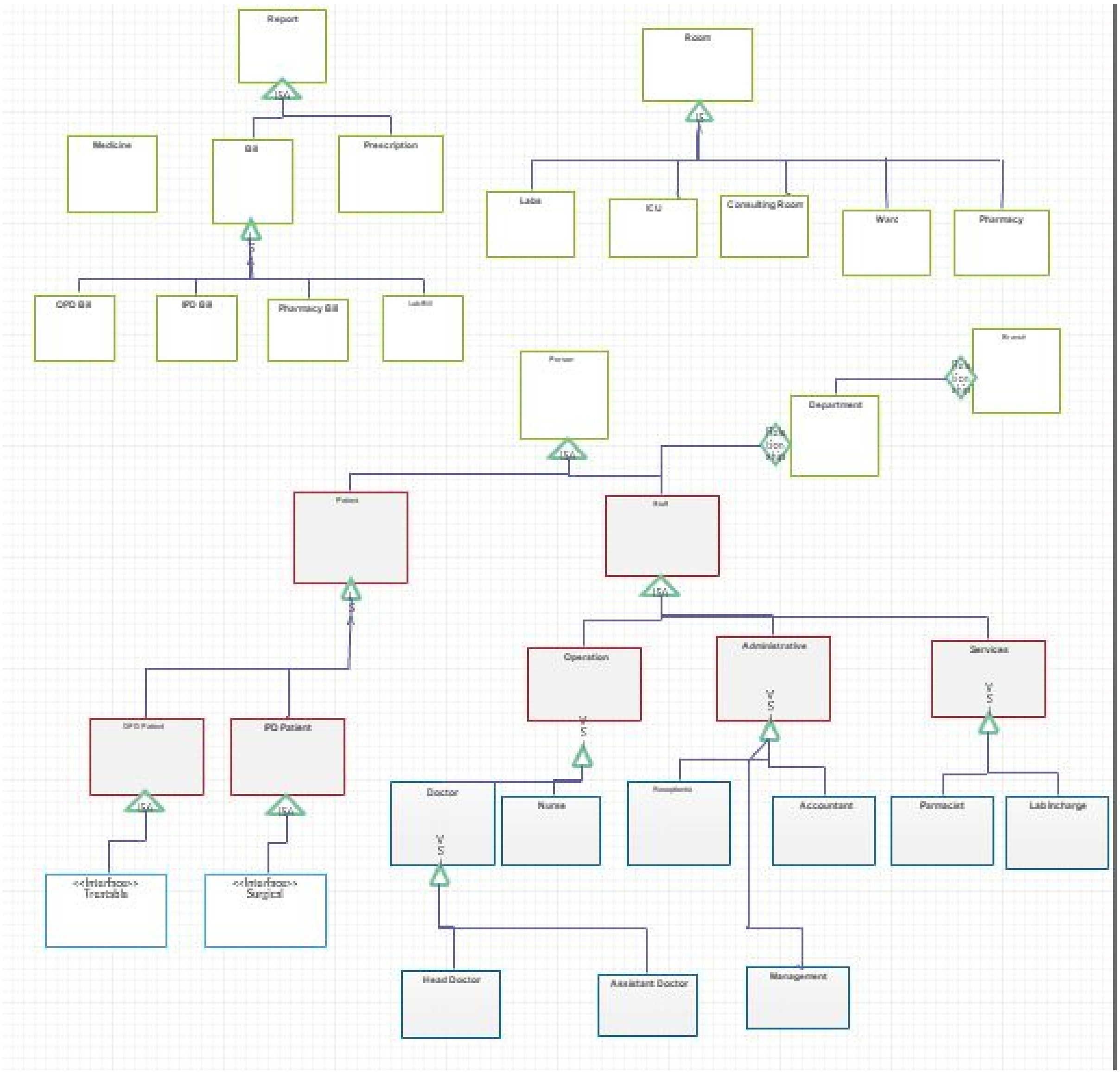
Having them directly inherit from the Staff class.

2. Abstraction

Means “you know what it does but not how it does it”

Can be done in 2 ways : Abstract Class and Interfaces.

Abstract Class:



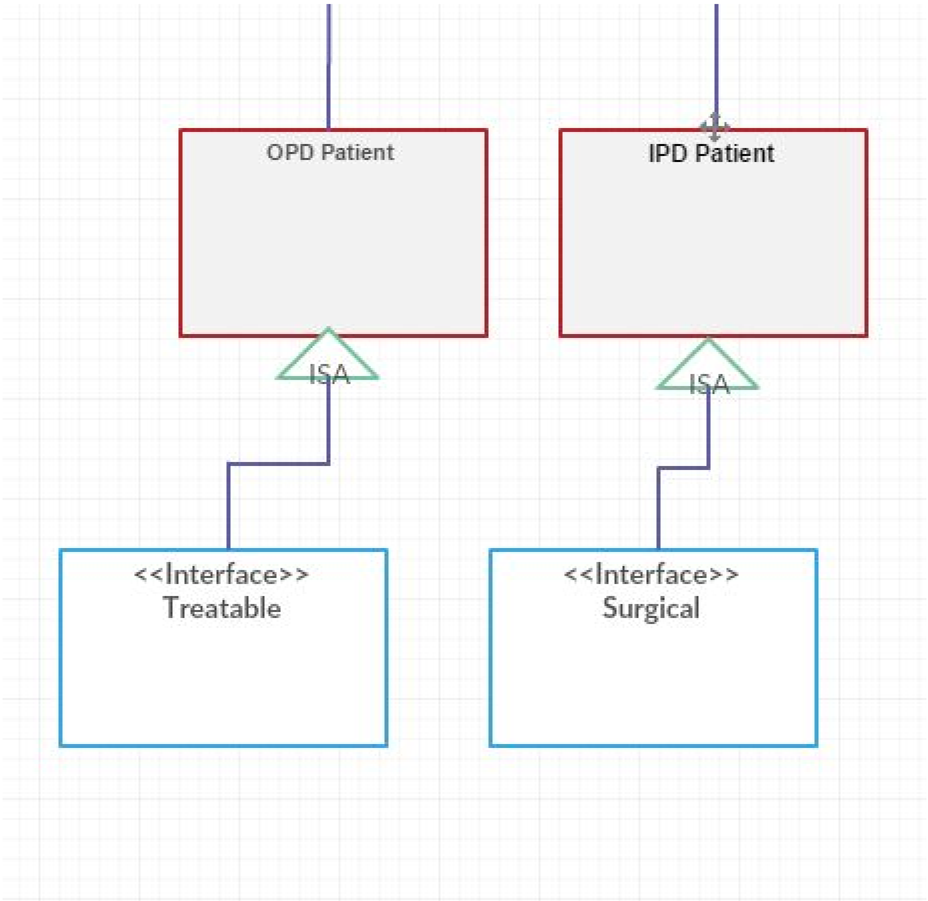
Abstract classes: Person, Patient, Staff, Administrative Staff, Operation Staff, Services Staff, Doctor, Room, Report, Bill.

Advantages:

This mainly helps in simplifying the code. This will also help in enforcing base properties/base functions. All the classes which need to be instantiated are declared abstract.

Interfaces:

There are 2 interfaces being used : Treatable and Surgical
We have assumed that every IPD patient will be undergoing surgery.



Advantage:

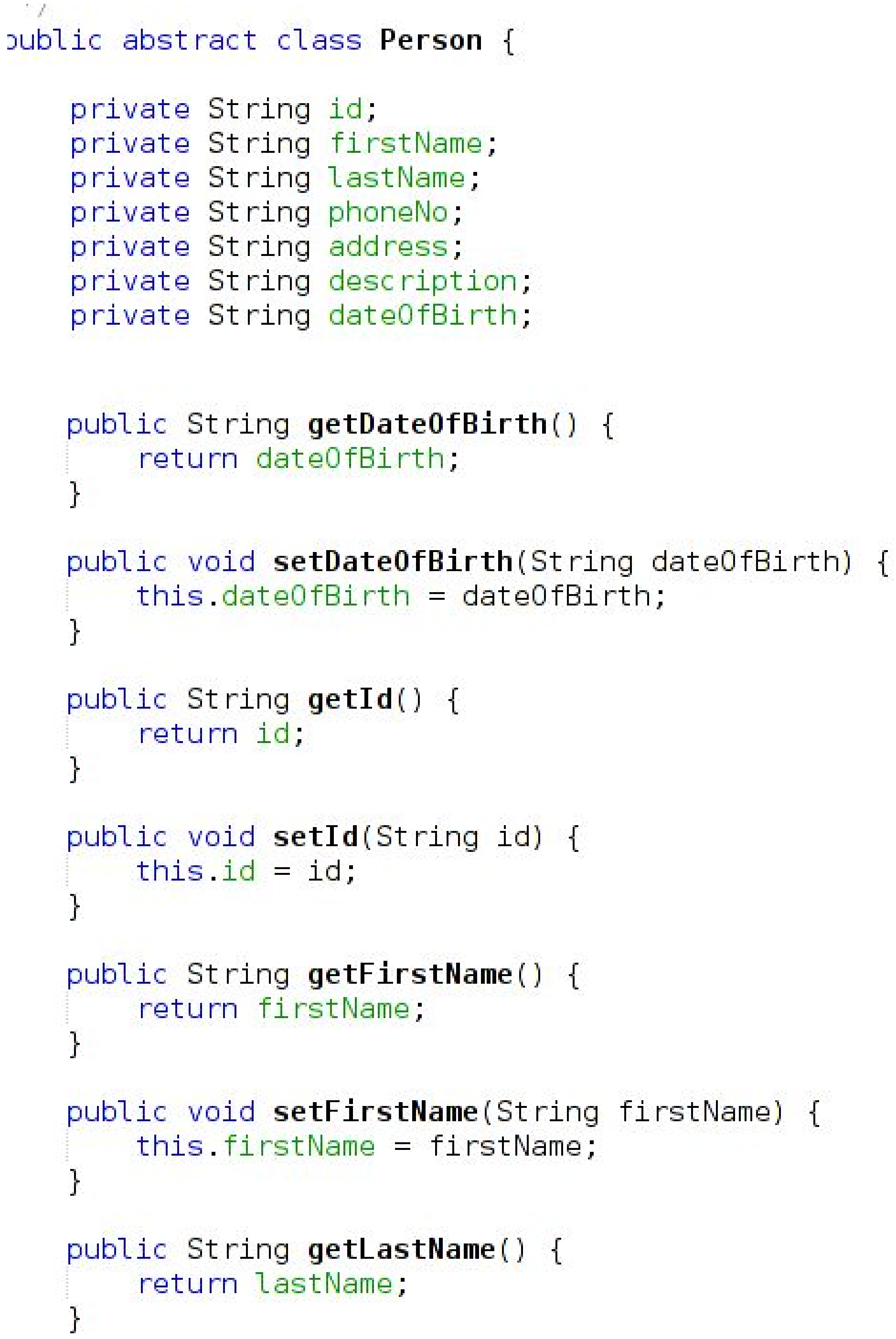
Having different interfaces for OPD and IPD patient mainly helps in simplifying the program.Both have clearly different behaviour/ roles in the hospital and hence two different interfaces.

Alternatives:

Having no interface.
Having Only one interface for both the class.

3. Encapsulation :-’

Private Data Members and their getters-setters.



```
 */
public abstract class Person {

    private String id;
    private String firstName;
    private String lastName;
    private String phoneNo;
    private String address;
    private String description;
    private String dateOfBirth;


    public String getDateOfBirth() {
        return dateOfBirth;
    }

    public void setDateOfBirth(String dateOfBirth) {
        this.dateOfBirth = dateOfBirth;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }
```

Usage:-

Data members of every class are made private so they can be encapsulated from each other.No other class can directly access the data members of the class they can access them through the getters and setters provided by every classes.

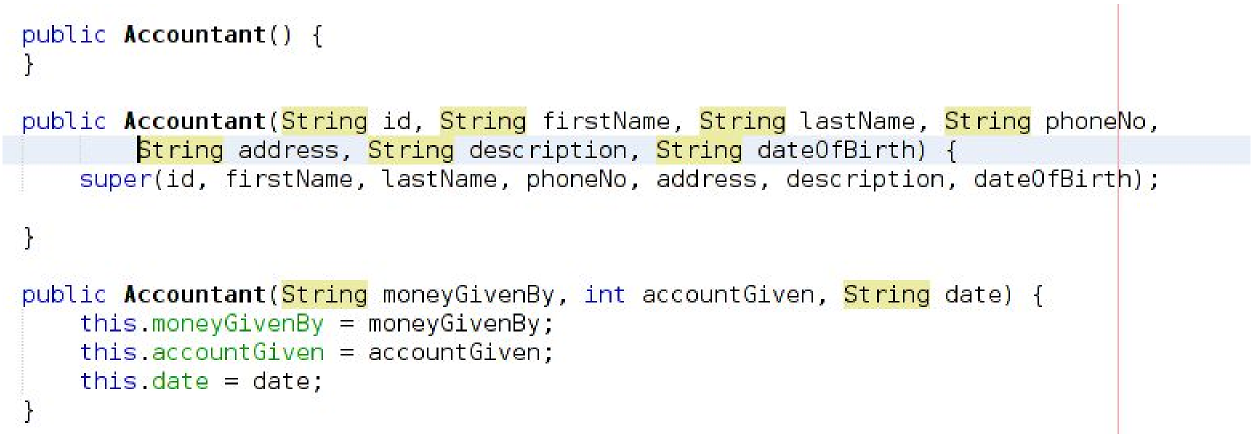
Advantage:-

Makes code less vulnerable

Alternatives:-

Make data members public this alternative makes the code more vulnerable.

4. Polymorphism:-



```
public Accountant() {
}

public Accountant(String id, String firstName, String lastName, String phoneNo,
        String address, String description, String dateOfBirth) {
    super(id, firstName, lastName, phoneNo, address, description, dateOfBirth);

}

public Accountant(String moneyGivenBy, int accountGiven, String date) {
    this.moneyGivenBy = moneyGivenBy;
    this.accountGiven = accountGiven;
    this.date = date;
}
```

Static Polymorphism:-

Function Overloading

Usage:-

Copy constructors are made at various places.When we try to store data in a file parameterised constructors are used while creation of object and when we try to view from database them we do not require constructors to set them

Advantage :-

Makes it easy to read the code.

Alternatives :-

Only non-parameterised constructor is made and setters are used to set the value of data-members.

Dynamic Polymorphism:-

Usage:-

Person class's reference variable is used and Object of Receptionist class or Manager Class is passed over.

Advantage:-

Creation of different reference variables is reduced to much extent.

Alternatives:-

Different reference variables must be made for each instance.

11.References

www.bankersheart.com/

https://www.youtube.com/watch?v=ibl5T-hxT7Y&list=PLGDO4Z63KR2ZU6ERsUTF0lrKb9rVNBoPo

https://www.google.co.in/search?q=hospital+management+classes&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiK6uS6juXTAhXFMY8KHWYlC_cQ_AUICygC&biw=1366&bih=662

https://www.udemy.com/learn-java-by-building-projects/