A
PROJECT REPORT

ON

“HEALTH CARE ANDROID APPLICATION”

Designed By

Mr. Drustant Ganpat Metar.

Submitted To

University Of Mumbai

In the partial fulfillment of

B.Sc. Computer Science (Third year)

Under Guidance of

Prof. P. N. Talankar

Through

THE HEAD,

DEPARTMENT OF COMPUTER SCIENCE

S.R.M. COLLEGE, KUDAL.

2018-2019

DECLARATION

To,

The Head,

Department of Computer Science,

S. R. M. College, Kudal.

Respected Sir,

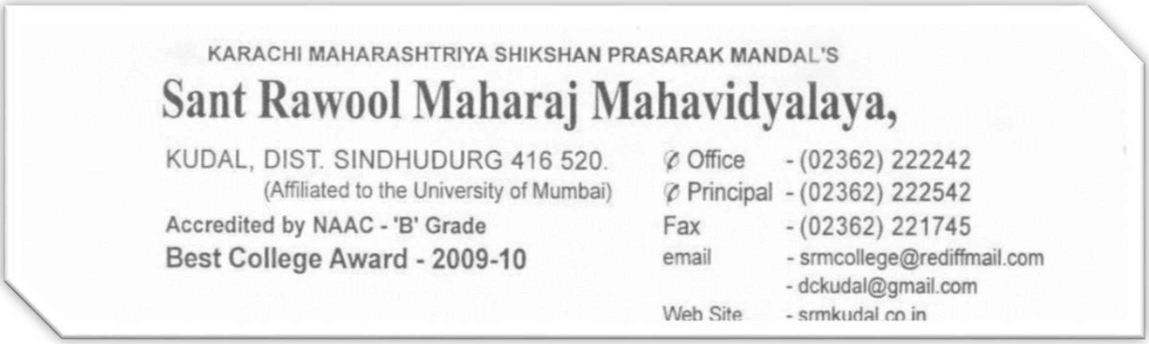
I undersigned, hereby declare that the project on "**Health Care Android Application"** is developed under the guidance of our **HOD Prof. P.N.Talankar.**

The conclusion in this report is based on the data, which is collected by me. I am declaring that this is my original work. I have not copied any materials, which are useful to my work, or other reports that are submitted to the S. R. M. COLLEGE, KUDAL this year.

I do undersign that if my work is found to be copied, then I am liable to punishment as per the university rule.

DATE:

PLACE: KUDAL. **(Mr.Drustant Ganpat Metar.)**



KARACHI MAHARASHTRIYA SHIKSHAN PRASARAK MANDAL'S

Sant Rawool Maharaj Mahavidyalaya,

KUDAL, DIST. SINDHUDURG 416 520.
(Affiliated to the University of Mumbai)
Accredited by NAAC - 'B' Grade
Best College Award - 2009-10

✆ Office	- (02362) 222242
✆ Principal	- (02362) 222542
Fax	- (02362) 221745
email	- srmcollege@rediffmail.com - dckudal@gmail.com
Web Site	- srmkudal.co.in

CERTIFICATE

This is certify that **Mr. Drustant Ganpat Metar** as completed his work on Project Report Titled,

“Health Care Android Application”

As per the syllabus prescribed for T. Y. BSc (Computer Science) of **Mumbai University, Mumbai**. It is also certified that this is his own work completed during academic year 2018 – 2019. The work done is satisfactory and is presented as per the specifications.

PROJECT GUIDE EXTERNAL EXAMINIER HEAD OF DEPARTMENT

DATE :

PLACE : KUDAL.

ACKNOWLEDGEMENT

I would like to thanks to various Doctors. Who give me a information of the entire workout of their present system. I would like to also thanks mostly **Dr.R.R. karambalekar**, who gives me opportunity to develop a project for him.

I would especially like to thanks,our HOD. **Prof. P.N. Talanakar** sir and All teaching and non- teaching staff of computer science faculty for inspiring me in completion of project.I am thankful for my project guide **Prof. P.N.Talankar sir** for his timely help and guide in completing this project successfully.Lastly , I would like to all thank all those who directly and indirectly helped in completion of this project.

PREFACE

This report contains the basic logic used for software development along with the diagrams so that the logic may be apprehended without difficulty. For detail information, screen layouts provided with the report can be viewed.

Although this report is prepared with at most care, there may be some errors for the project is subjected to further enhancement as per the requirements of organization.

INDEX

CHAPTER – I

PRELIMINARY INVESTIGATION

ORGANISATIONAL OVERVIEW

Project Name :	Health Care Android Application
Objective:	To help doctors to treat patients better.
People interacting in the system	Doctor, patient and assistance**.**
Location	All over Maharashtra.

DESCRIPTION OF CURRENT SYSTEM

HEALTH CARE ANDROID APPLICATION helps the doctor to get patient details i. e, old as well as new (latest).Which will help the doctor to understand what kind of diseases the patient was going through and can understand what is going on with his/ her patient? The patient will be provided with his/her smart card which is easy to portable and can store a lot of information rather than an old file. Any doctor can use **HEALTH CARE ANDROID APPLICATION** web application anywhere around Maharashtra. The doctor can either use mobile or desktop/pc which is affordable to him/her.

The doctor is the main entity for this application. The doctor can add assistance/receptionist as well as the patient. The doctor can only fill details and prescription of patient but cannot edit or delete old patient details, to keep patient details more useful to doctor. The doctor can search patient which has already registered.

The patient is the second main entity in this application. The patient can register in **HEALTH CARE ANDROID APPLICATION** after registering he/she will get a smart card, by which doctor can find patient details easily. The patient can only view his/her record but cannot change it. He/she can change editable records such as address, contact number, etc.

The last but not the least is assistance/receptionist. After doctor assistance/receptionist handles things such as a registering patient, filling patient details etc. Receptionist/assistance will accept money and give a report to the doctor.

LIMITATIONS OF CURRENT SYSTEM

Limitation of HEALTH CARE ANDROID APPLICATIONare:

1. If there is no internet connectivity then there is a problem of entering data.
2. Each time patient has to carry a smart card.

To overcome all these limitations there is a need of better
&
Efficient Computerized System.

PROPOSED SYSTEM

HEALTH CARE ANDROID APPLICATION

The need for system arises because there was no such system existing.

Reasons for these of new system are-

- With the help of **HEALTH CARE ANDROID APPLICATION**, the doctor can treat the patient properly and correctly.
- The doctor can know old disease of the patient and can treat him/her accordingly if the patient has registered.
- The proposed system will work on web application, so that any customer (doctor) can access this web application from anywhere, at any time.
- The patient will be given a unique smart card which will work only on **HEALTH CARE ANDROID APPLICATION.**
- The patient will need not to go clinic for taking an appointment, instead, he/she can register for **HEALTH CARE ANDROID APPLICATION** web application from home.
- The old history of the patient will be available to the doctor so that he can treat the patient more correctly.

ADVANTAGES OF PROPOSED SYSTEM

1. The new and attractive website will fulfils the requirement.
2. The system will have a user-friendly approach.
3. Speeding up the process.
4. Efficient data retrieval & updation.
5. Efficient search facility.
6. This system will be a secure system, which will provide corporate level security.
7. The doctor can know patient old record of disease for addition checkup.
8. The doctor can check which medicines were provided to the patient by his as well as other doctors.

FEASIBILITY STUDY

Project feasibility analysis is an activity that verifies whether a project can be started & successfully completed.

Activities to confirm project Feasibility are as following.

1. **Economic Feasibility**
2. **Organizational & cultural & cultural feasibility**
3. **Technological feasibility**
4. **Resource feasibility**
5. **Schedule feasibility**

1. Economic Feasibility:-

It consists of-

1) Is the anticipated value of the benefits greater than the projected cost of development?
2) Does the organization have the adequate cash flow to fund the project during the development period?
   - Reduced paperwork & save the money needed for printing forms.
   - The project will give intangible benefits like,
   - The doctor can know what that patient has gone through.
   - The doctor can know predict what disease his/she might get.

Hence, project passes economic feasibility.

2. Organizational & Cultural Feasibility:

Each organization has its own culture

1) The committee is working with the well-know organization.
2) No member of committee will come across any loss.

Hence, project passes organization & cultural feasibility.

3. **Technological Feasibility** :-

   The developing project may produce challenges to a current employee due to lack of knowledge. If the project needs expertise from outside location to maintain software the problems may occur regarding the money.

   a) The project technologically feasibility as the organization have a well-skilled employee to maintain software.

   Hence, project passes technological feasibility.

4. **Resource Feasibility :-**

   a) The resources like a computer, physical facilities are available.

   b) Employee with enough programming skills are available to maintain software.

   Hence, project passes resource feasibility.

5. **Schedule Feasibility** :

   a) As the project has passes all the feasibility tests, then the project will be successful.

   b) The schedule feasibility means the project will be completed in given time. But during completion of the project there can occur minor difficulties.

   c) We can complete this project on time by proper planning and work. But we can’t guarantee that our project will be completed with its all requirements within the less amount of time.

STAKEHOLDER

1. **Doctor** :

The doctor is the main entity in this project, treats the patients. The doctor is the main entity in HEALTH CARE ANDROID APPLICATION.

2. **Patient** :

The patient is also main entity but it comes after doctor. Patient comes with his/her query near a doctor.

3. **Receptionist** :

Who manages patient after doctor. And collect payment from the patient.

4. **Project Developer** :

The project developer is responsible for developing the project, coding, designing, modifying etc. The project developer handles all project development activities from initial site surveys, feasibility studies, financial models, contracts, permits, installation, construction management, and ongoing maintenance and operation contracts.

5. **Project Tester** :

It is the person who test whether the project user-friendly and fulfils all the requirements of the user or not. Execute all the test case and report defects, defines security and priority for each defect. Carry out regression testing every time changes are made to the cod to fix defects. A software tester is responsible for designing testing scenarios for usability testing. He is responsible for conducting testing, thereafter analyzed the results and then submits his observations to the development team.

TECHNOLOGIES USED

Software :

- Front End : Android Studio 3.2.1,Java.
- Back End : MySql
- Operating System : Windows 10

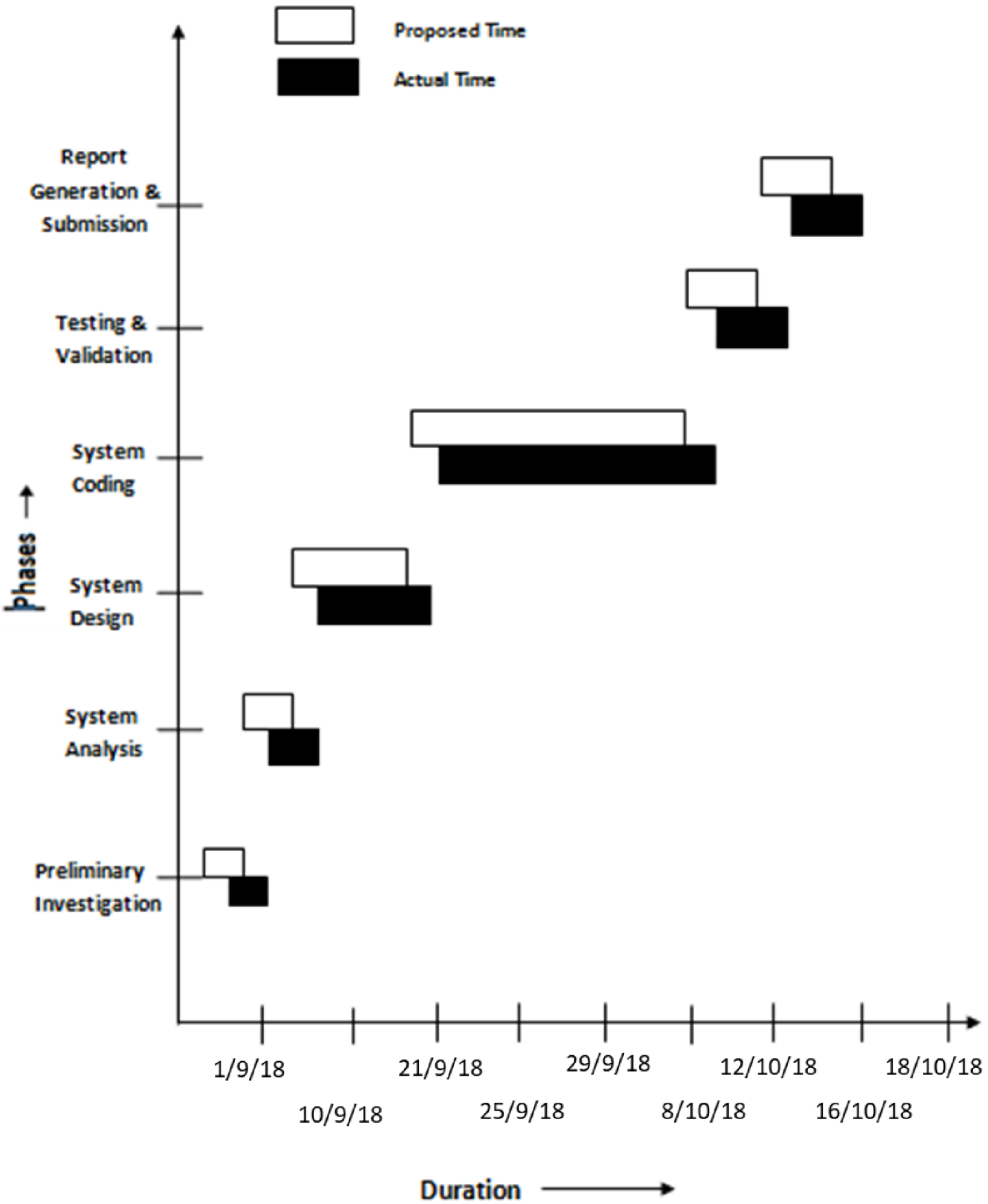
Hardware :

- Processor : Intel Core i5
- Memory : 8 GB RAM
- Storage : 1 TB Hard Disk Space

Tools :

- Star UML
- Microsoft Office Word
- Microsoft Office Powerpoint

GANTT CHART


Proposed Time
Actual Time
Report Generation & Submission
Testing & Validation
System Coding
System Design
System Analysis
Preliminary Investigation
Phases
1/9/18
10/9/18
21/9/18
25/9/18
29/9/18
8/10/18
12/10/18
16/10/18
18/10/18
Duration

CHAPTER – III

SYSTEM ANALYSIS

FACT FINDING TECHNIQUE

The fact-finding technique is one of the parts of the system analysis. At the time of analysis of the system or before starting actual work, system analyst collects the information . for gathering information prefers any fact-finding technique such as,

1. Interview
2. Record Review
3. Questionary
4. Observation

While developing this system I have done that part by using interview & questionary.

1.Interview

It is used to collects information from groups or from the individual. The interview must plane in advanced and should know the problem using the consideration. There are 2 types of interview.

- **Structure Interview**

  In the specific question are asked for the covered specific area. In this type of technique interview period may be short.

- **Unstructured interview**

  In this different types, question are to collects extra information . in this type of technique different type of question ate asked and there is the specific area.

Questionary :-

Questionary contains a sequence of question are asked to collect extra information from a large number of pearsons.

- **Open ended question :**

  Open-ended question are used to learn the opinion , feeling general experience about the problem.

- **close-ended question :**

  Close-ended question specific question and response from which respondent has to choose best one.

USE CASE DIAGRAM

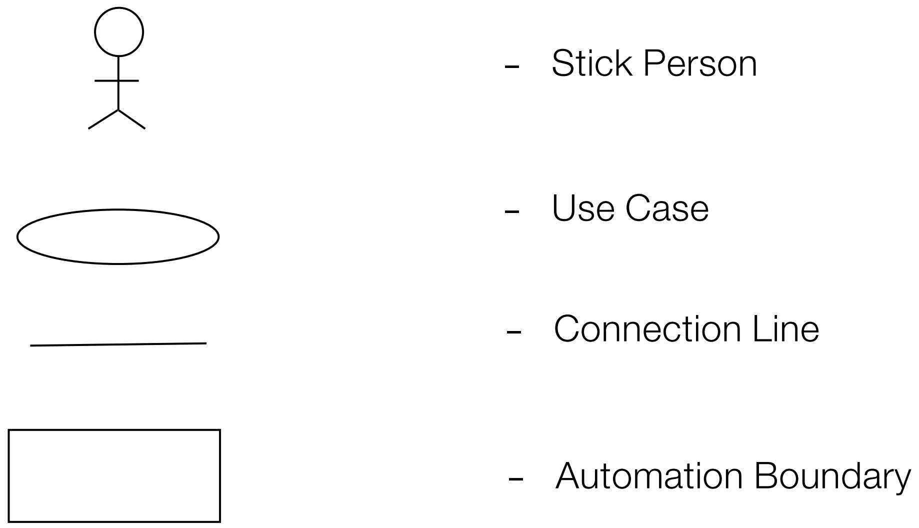
Use case describes the behavior of a system from a user's standpoint by using actions & reactions. They allow the definition of the system's boundary & the relationships between the system & the environment.

Use cases associated with object-oriented technique provide a complete approach for the whole project lifecycle, from specification to implementation. A use case corresponds to a specific king of system use. It is an image of a system's functionally, which is triggered in response to the simulation of an external actor.

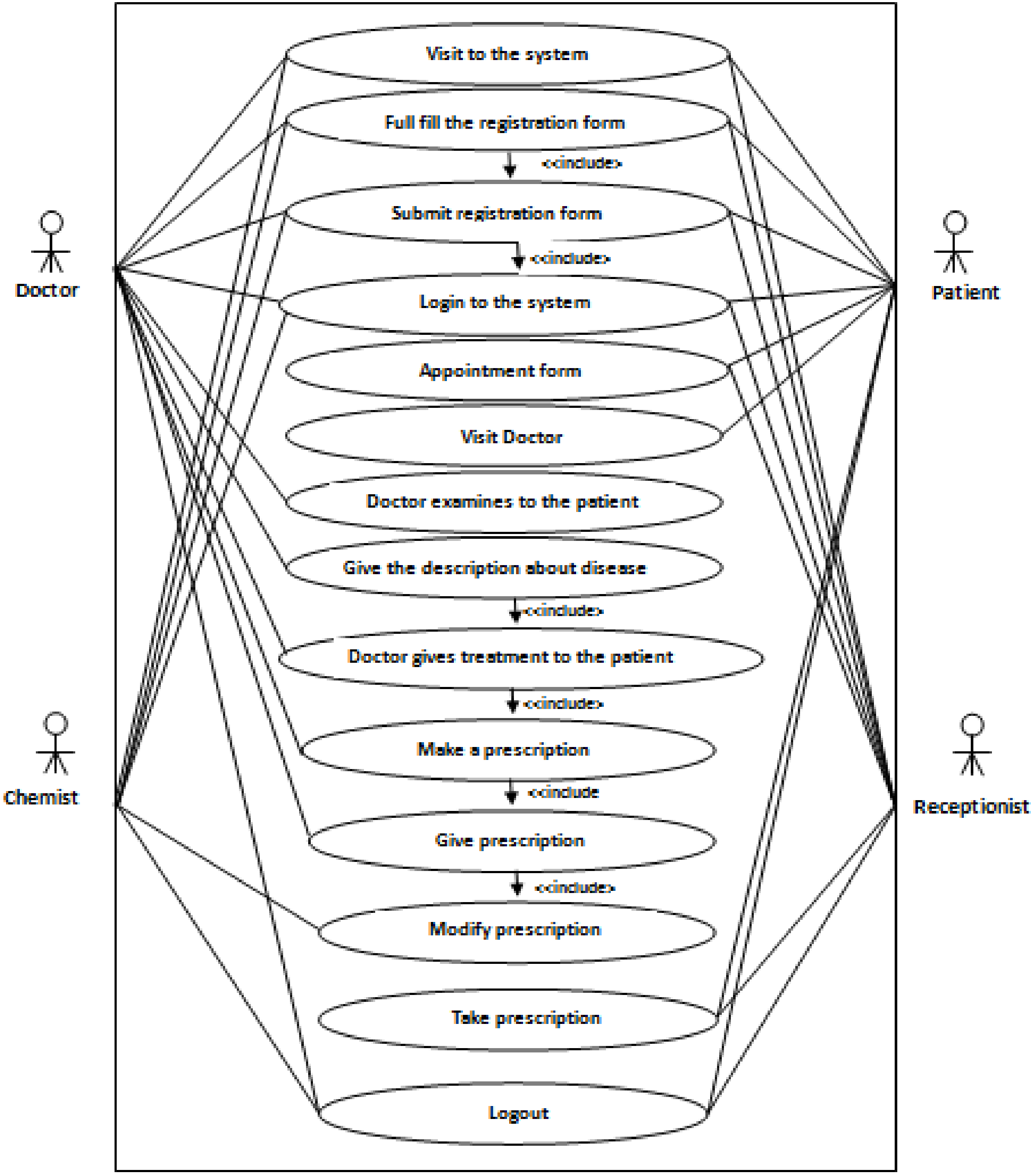
How to draw use case diagram-

1) Identify actors of the system.
2) After identifying the role of the actors next developed the list of flow of activities as the starting point for identifying various scenarios.

Symbols used for use case diagram-



USECASE DIAGRAM



ENTITY RELATIONSHIP DIAGRAM

Entity – Entities are person, places, events, objects or concepts.
Relationship – a relationship is a naturally occuring association among entities.

Types of Relationships –

1) Mandatory Relationship.
2) Optional Relationship.
3) Binary Relationship.
4) Ternary Relationship.
5) Unary Relationship.

The traditional approach places a great deal of emphasis on data storage requirements for the new system. The model used to define the data storage requirements is called the Entity Relationship Diagram.

Rectangles represents the data entities.

Lines connecting the rectangles show relationship among the data entities.

ER Diagram is the high level conceptual diagram, which is based on the perception of real world that consist of set of basic objects called entities.

So objects of ER Model are,

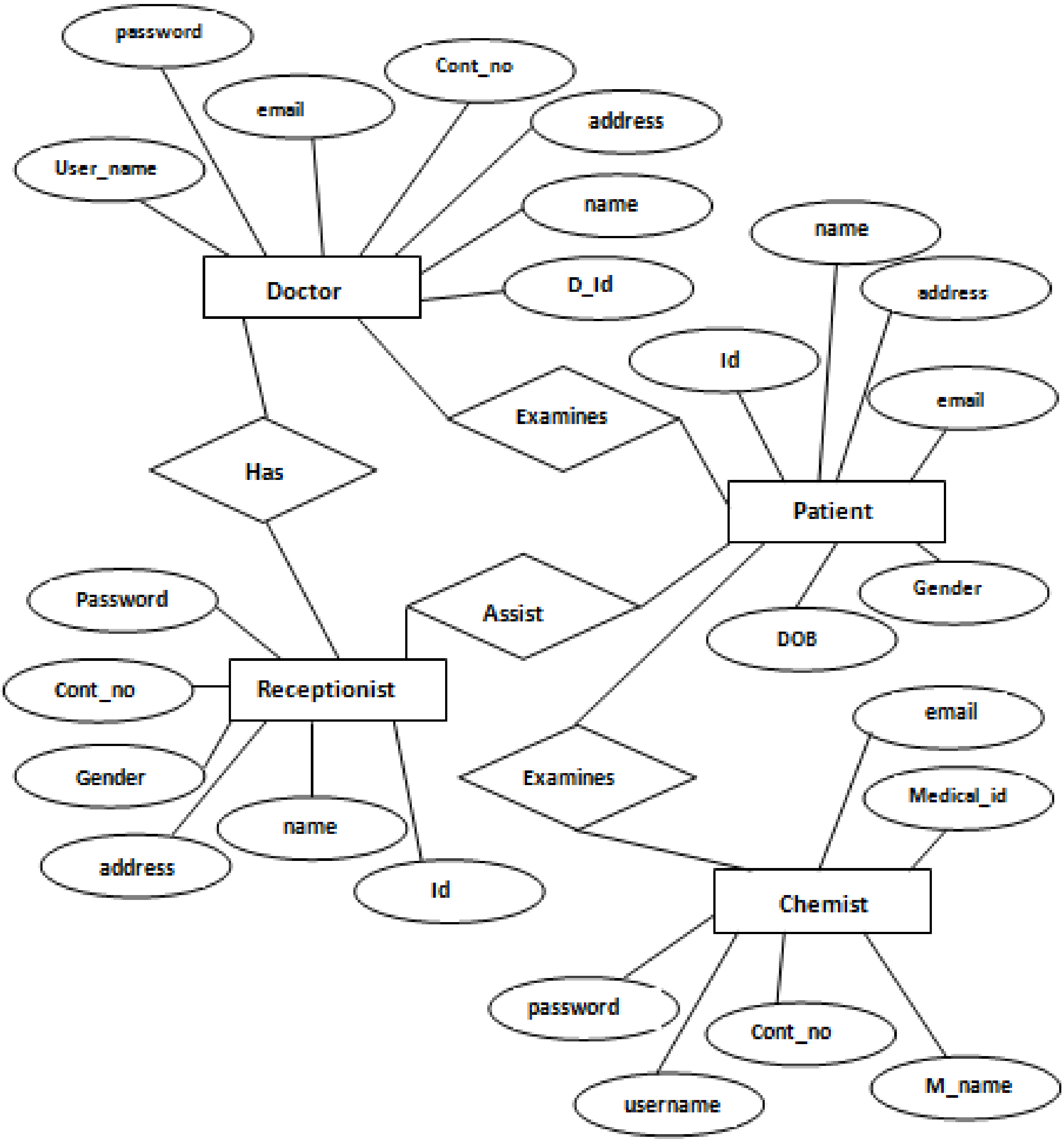
a) Entity Set
b) Relationships
c) Attributes

Entity is basic object of ERD, which is thing in real world with an interdependent existence. i.e. it is distinguishable from other objects.

Among entities relationship is shown. Relationship is of type one-one, one-many, many-one, many-many. It is also called Cardinality.

Each entity is described by set of properties called Attributes.

ENTITY RELATIONSHIP DIAGRAM



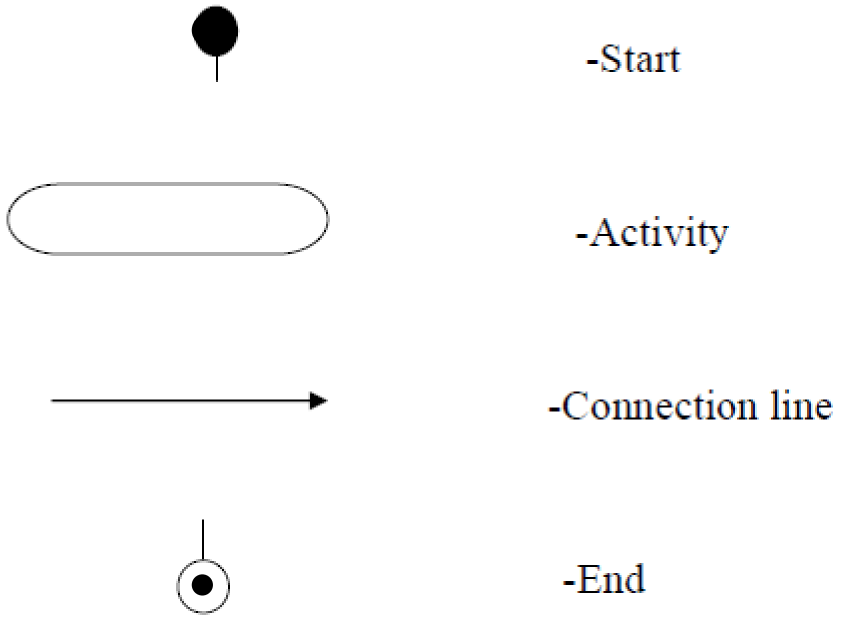
ACTIVITY DIAGRAM

An activity diagram is a variant of state chart diagram organized according to actions, and mainly targeted towards representing the internal behavior of a method or a use case. An activity is represented by a rounded rectangle.

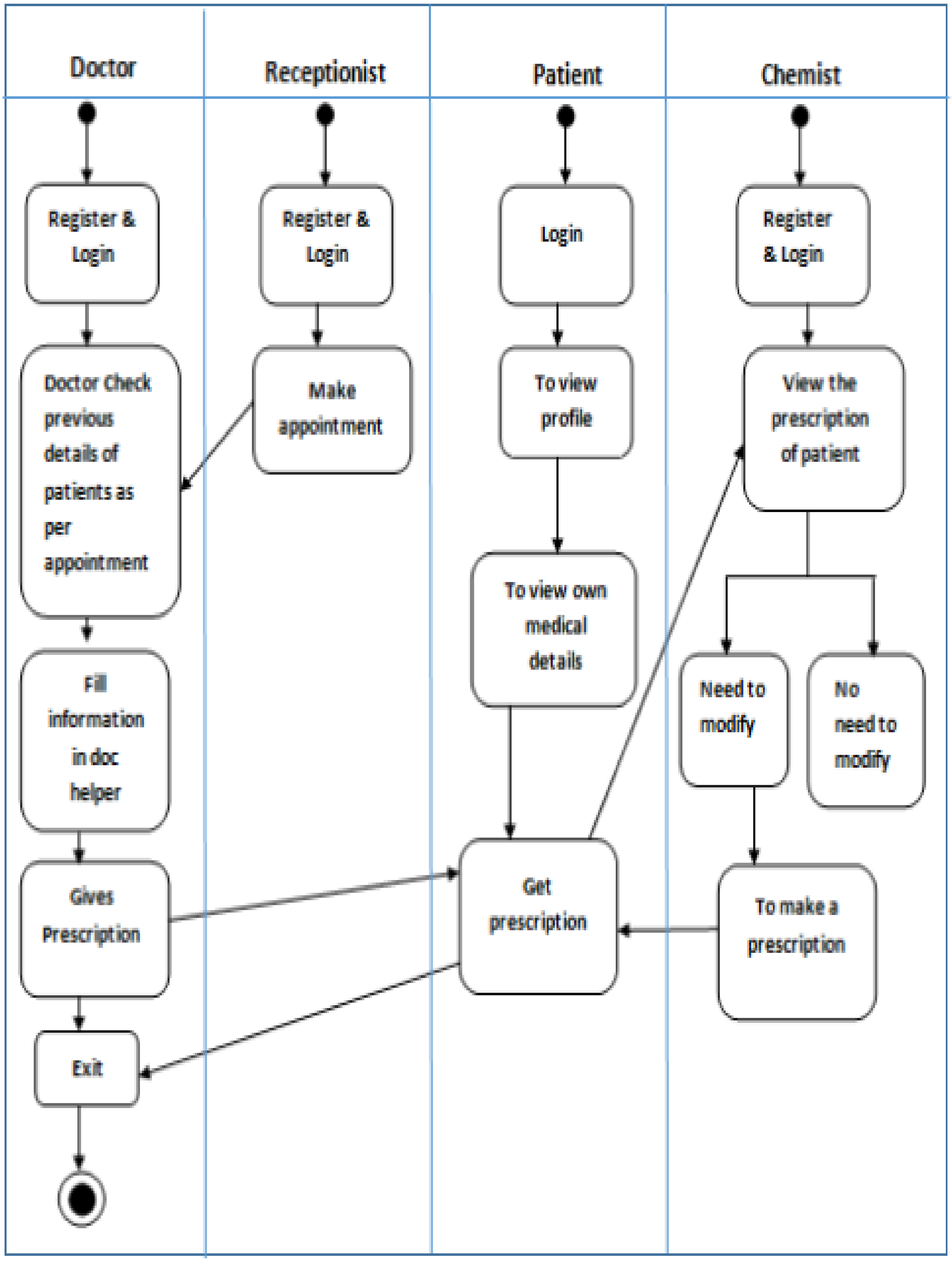
How to develop activity diagram:

1] Identify swim length.
2] Identify input message.
3] Describe message from external actor to system using message notation.
4] Identify and add any special condition on the input messages including iteration
and true or false condition.
5] Identify and add the output return messages.

Symbols used for Activity Diagram :



ACTIVITY DIAGRAM



CLASS DIAGRAM

Class diagram express, in a general way, the static way, the static structure of a system, in terms of classes and relationships between those classes. Just as a class describes a set of objects, an association describes a set of links; objects are class instances, and links are association instances.

The rectangles that act as a symbol for the class may also contain a stereotype and properties. UML defines the following class stereotype.

StereotypeDefinition

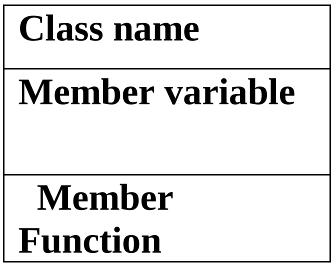
<<Signal>>	A notable event that trigger.
	A transaction within a state machine.
<<Interface>>	A description of visible operations.
<<Metaclass>>	The class of a class, as in small talk.
<<Utility>>	A class reduced to the concept of the module and which cannot be Instantiated.

How to draw a class diagram:

1) Class diagram are more popular UML diagrams used for the construction of software application. So it is very important to learn the drawing procedure of class diagram.

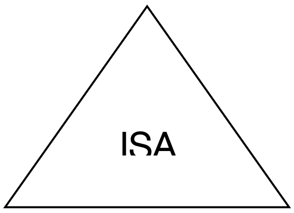
2) Class diagram have a lot of properties to consider while drawing but here the diagram will be considered from a top level view.

SYMBOLS USED FOR CLASS DIAGRAM:

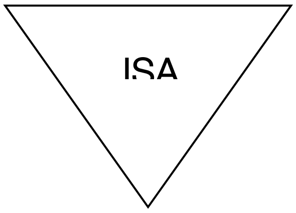


-Rectangle box(used to represent class)

- Connection line (Used to represent the association between two classes).

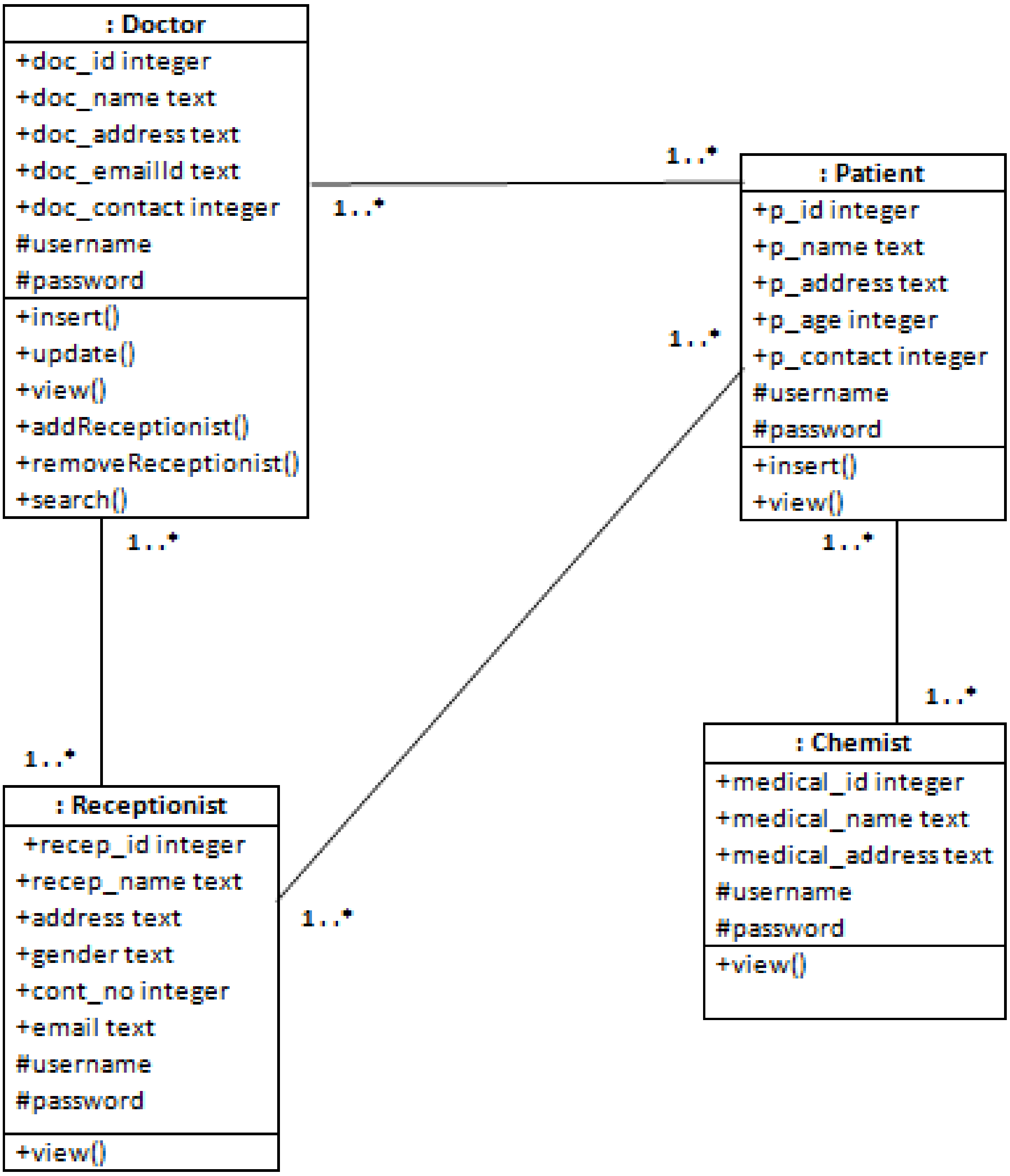


-Triangle (Used to specialization of class)



-Triangle (Used to Generalization of class)

CLASS DIAGRAM



OBJECT DIAGRAM

Object diagrams, or instance diagram, illustrate objects and links. As in the case of class diagrams, object diagrams represent the static structure .The notation used for object diagrams is derived from that of class diagrams; elements that are instances are underlined.

Object diagrams are primarily used to show a context –before or after an interaction, for example. However, they are also used to aid the understanding of complex data structures, such as recursive structure.

How to draw object diagram:

1] Analyze the system and decide which instance are having important data and association.

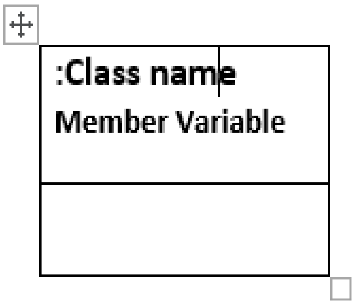
2] Consider only those instances which will cover those functionality.

3] Object diagram should have meaningful name indicate its purpose.

4] Association among object should be clarified.

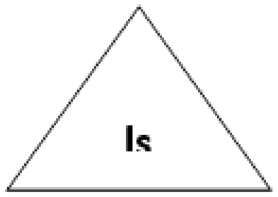
5] Add proper notes at points where more clarity is required.

Symbols used for object Diagram:

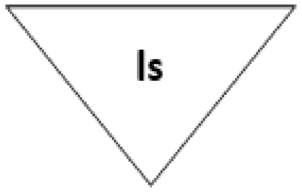


-Rectangle box (used to represent Object)

-Connection Line (Used to represent the association between two object)

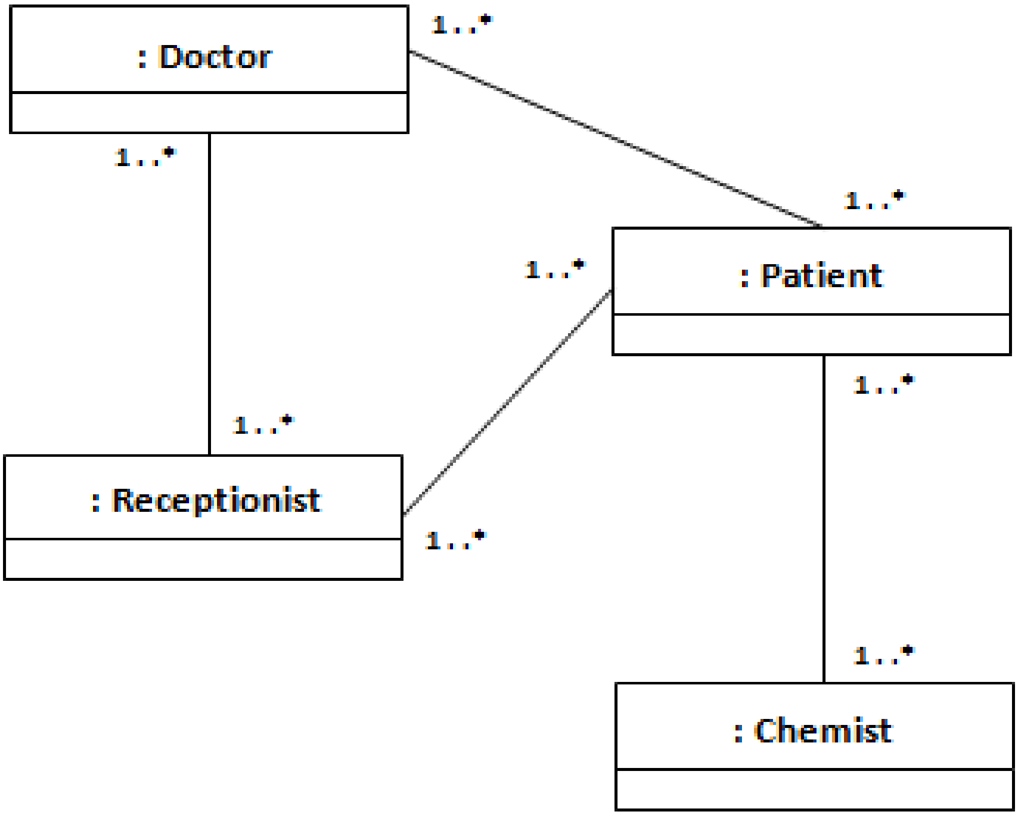


-Triangle (Used to Specialization of object)



-Triangle (Used to Generalization of object)

OBJECT DIAGRAM



SEQUENCE DIAGRAM

Sequence diagram documents the information flows within a single use case or a single scenario. Sequence diagram shows the sequence of the interaction between object that occurs during the flow of event of single scenario or use case.

How to draw sequence diagram:

1] Identify all object and actors that involved in scenario

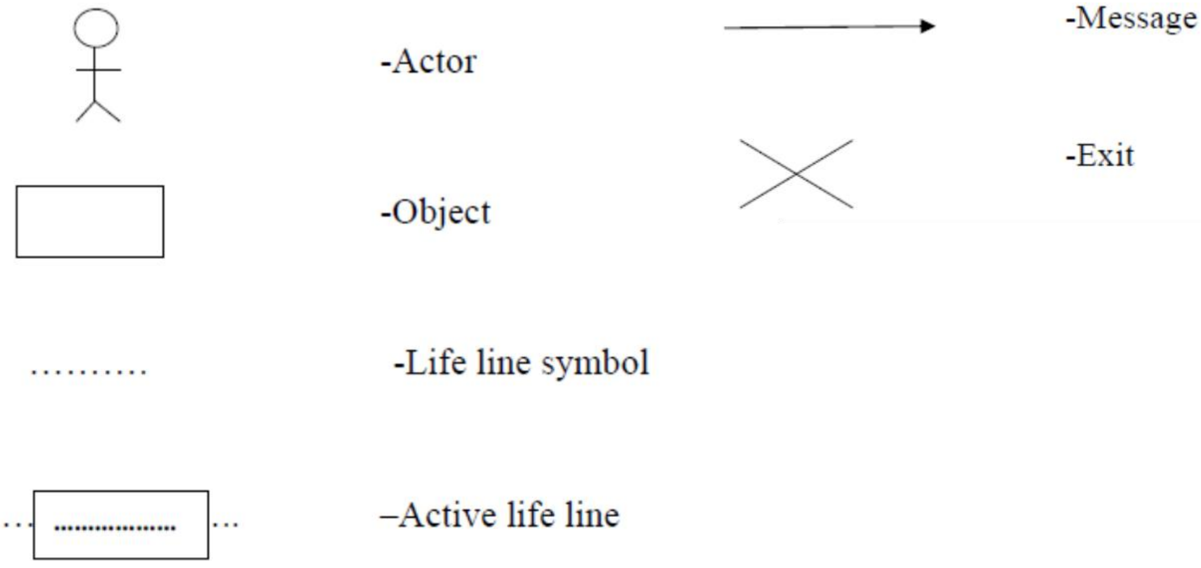
2] Based on the flow activities, identify each message that will be required to carry out the scenario .identify both the source object and the actor for message and destination object or actor.

3] Determine whether each message is always sent or whether it is sent only under certain condition.

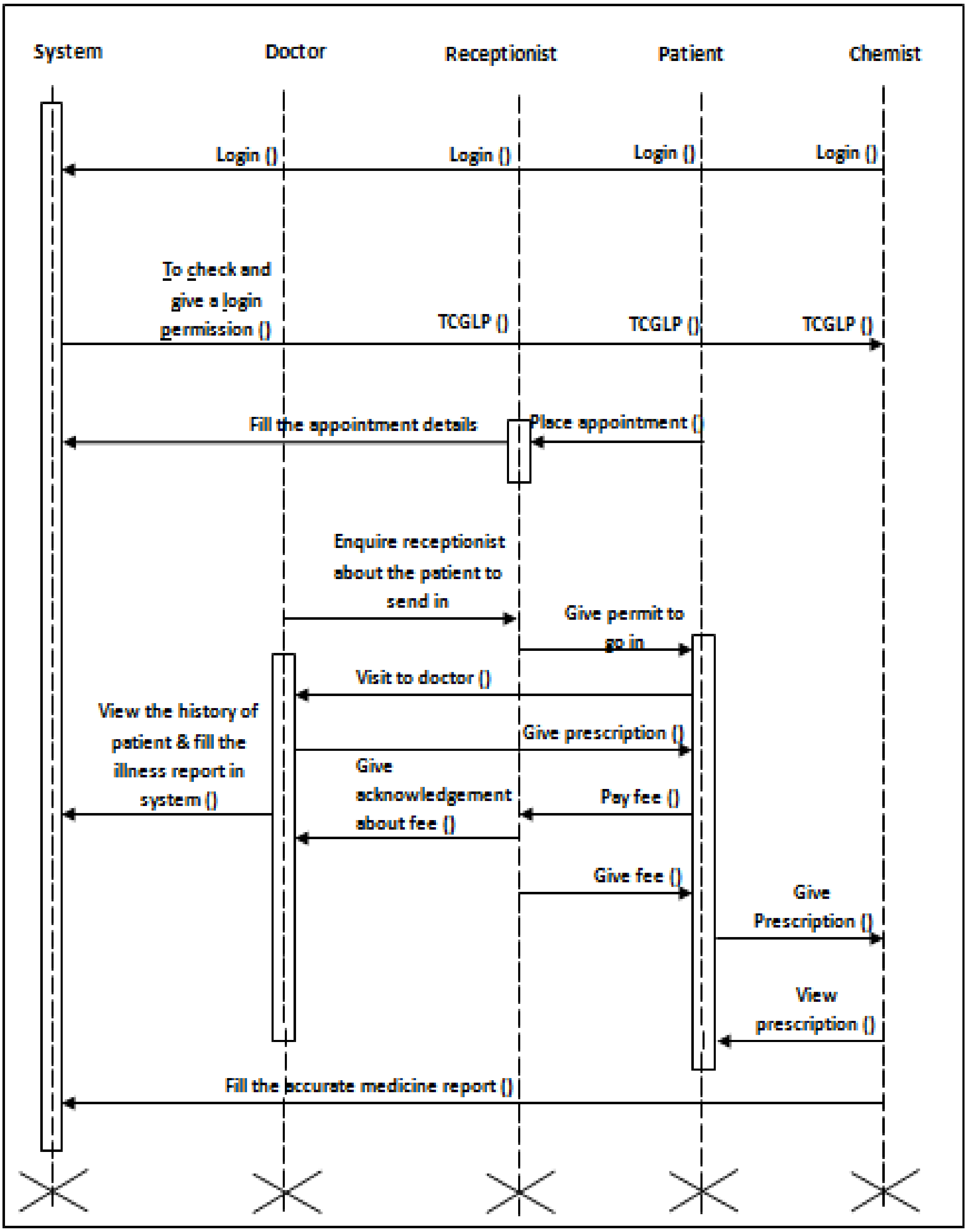
4] Sequence the message correctly and attach them to the appropriate life lines of the actors and objects.

5] Add formal syntax on the message to describe condition, message name, and passed parameters

Symbols used for sequence Diagram:



SEQUENCE DIAGRAM


System
Doctor
Receptionist
Patient
Chemist
Login ()
Login ()
Login ()
Login ()
To check and give a login permission ()
TCGLP ()
TCGLP ()
TCGLP ()
Fill the appointment details
Place appointment ()
Enquire receptionist about the patient to send in
Give permit to go in
Visit to doctor ()
View the history of patient & fill the illness report in system ()
Give prescription ()
Give acknowledgement about fee ()
Pay fee ()
Give fee ()
Give Prescription ()
View prescription ()
Fill the accurate medicine report ()

STATE DIAGRAM

A state diagram is a diagram used in computer science to describe the behavior of a system considering all the possible states of an object when an event occurs. This behavior is represented and analyzed in a series of events that occur in one or more possible states. Following are the widely used components of state diagram.

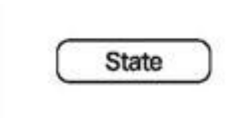
- **Initial State**

The initial state represents the source of all objects:

It is not a normal state, because objects in this state do not yet exist.

- **State**

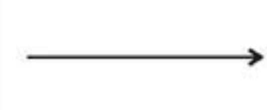
The state of an object is always determined by its attributes and associations. States in statechart diagrams represent a *set* of those value combinations, in which an object *behaves the same* in response to events:



Therefore, not every modification of an attribute leads to a new state.

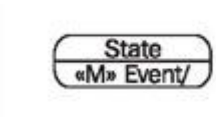
- **Transition**

A transition represents the change from one state to another:



- **Internal Transition**

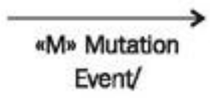
An internal transition is a transition from one state to itself. This means that the object handles the event without changing its state:



The events that initiate the internal transition are listed in the lower part of the state symbol. For instance, a frequent flyer card object in the state normal remains in the state normal when the event «M» add miles occurs.

- **Mutation Event**

A mutation event is the initiator of a transition from one state to another, or for an internal transition, where the state remains the same:



- **Action**

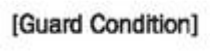
An action is the activity of an object that is initiated by an event:

«M» Event/Action

An action describes what the object does in response to the event. This description can be textual or formalized.

- **Guard Condition**

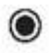
A guard condition is a condition that has to be met in order to enable the transition to which it belongs:



Guard conditions can be used to document that a certain event, depending on the condition, can lead to different transitions.

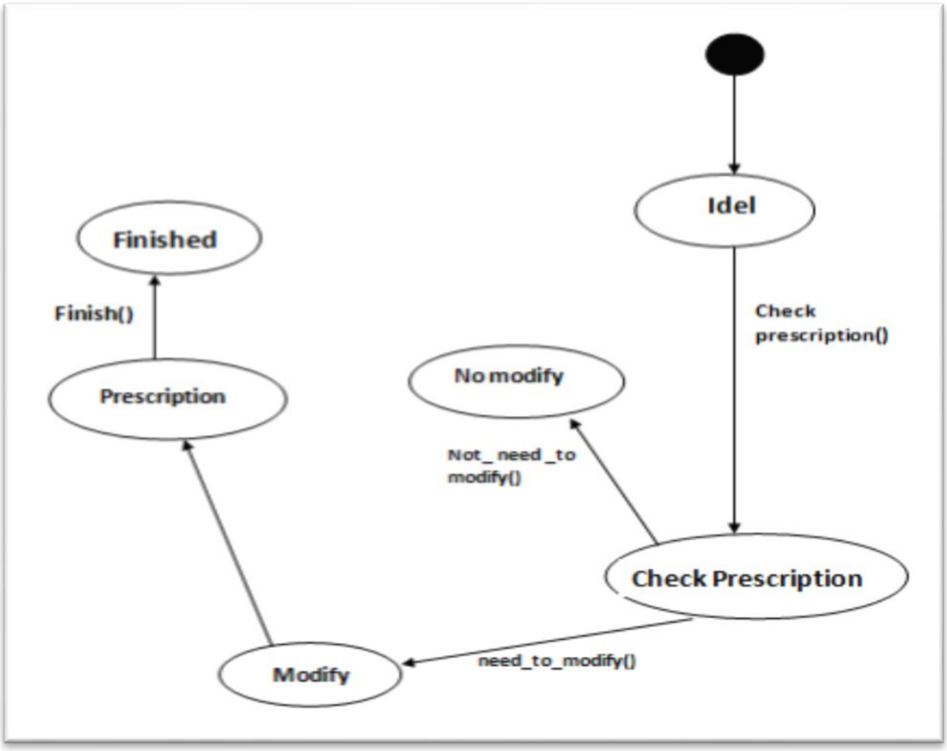
- **Final State**

The final state represents the end of an object's existence:

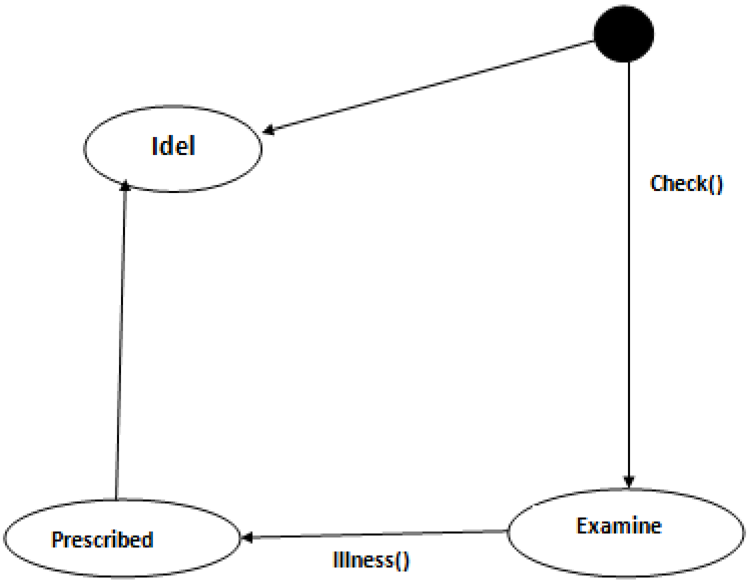


State Diagram

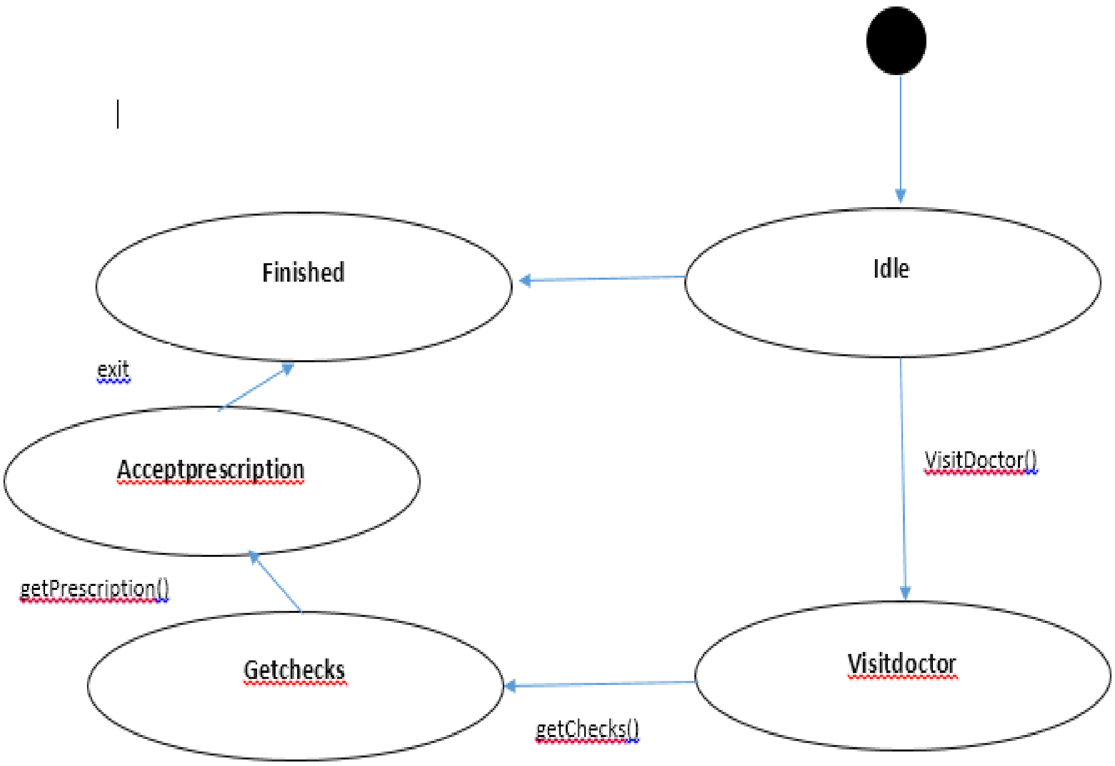
chemist diagram



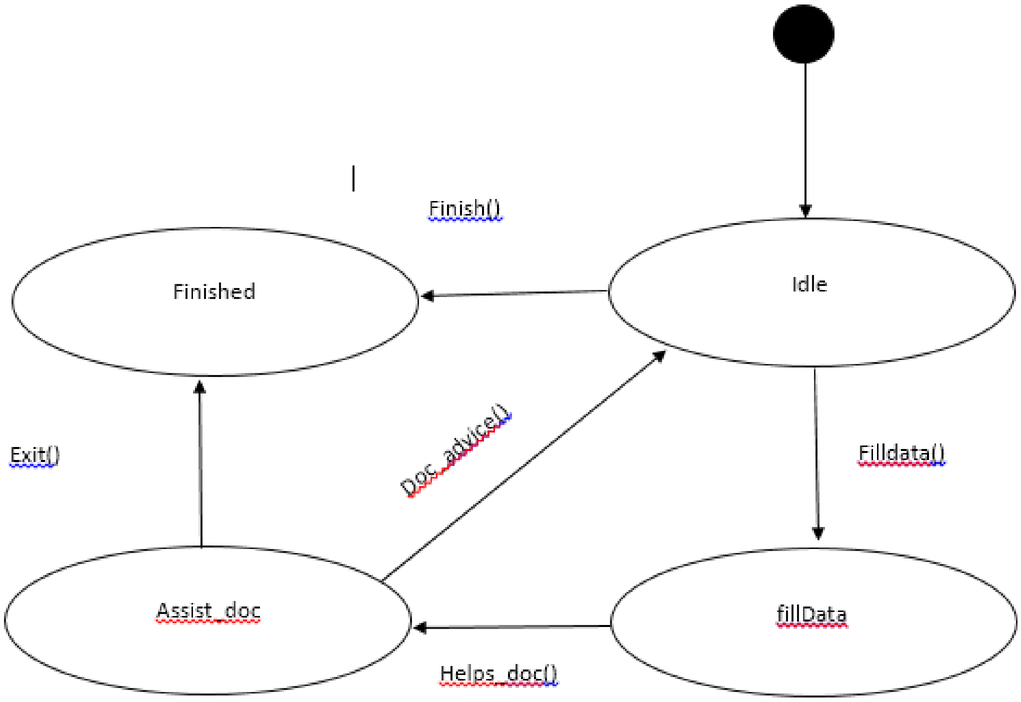
Doctor



Patient


Finished
Idle
exit
Acceptprescription
VisitDoctor()
getPrescription()
Getchecks
Visitdoctor
getChecks()

Receptionist



CHAPTER – III

SYSTEM DESIGN

CONVERTING ERD TO TABLES

TABLE :DOCTOR REGISTRATION TABLE

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Doctor id
NAME	VARCHAR	50	-		Doctor name
ADDRESS	VARCHAR	150	-		Doctor address
CONTACT	INT	12	-		Doctor contact
EMAIL	VARCHAR	50	-		Doctor email
QUALIFICATION	VARCHAR	20	-		Doctor qualification
GENDER	VARCHAR	6	-		Doctor gender
TYPE_DOCTOR	VARCHAR	20	-		Doctor type-doctor
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Doctor username
PASSWORD	VARCHAR	50	-		Doctor password

Table description : This table keeps information of Doctor.

TABLE :PATIENT REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Patient id
NAME	VARCHAR	50	-		Patient name
ADDRESS	VARCHAR	150	-		Patient address
CONTACT	INT	12	-		Patient contact
EMAIL	VARCHAR	50	-		Patient email
GENDER	VARCHAR	20	-		Patient gender
BLOOD GROUP	VARCHAR	6	-		Patient blood group
STATUS	VARCHAR	20	-		Patient status
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Patient username
PASSWORD	VARCHAR	50	-		Patient password

Table description : This table keeps information of patient registration.

TABLE : Appointment Registration

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Patient id
NAME	VARCHAR	50	-		Patient name
ADDRESS	VARCHAR	150	-		Patient address
CONTACT	INT	12	-		Patient contact
EMAIL	VARCHAR	50	-		Patient email
GENDER	VARCHAR	20	-		Patient gender
DOCTOR NAME	VARCHAR	20	-		Doctor name
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Patient username
PASSWORD	VARCHAR	50	-		Patient password

Table description: This table keeps information Appointment of the patient

TABLE : RECEIPTNIST REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Receiptnist id
NAME	VARCHAR	50	-		Receiptnist name
ADDRESS	VARCHAR	150	-		Receiptnist address
CONTACT	INT	12	-		Receiptnist contact
EMAIL	VARCHAR	50	-		Receiptnist email
QUALIFICATION	VARCHAR	20	-		Receiptnist qualification
GENDER	VARCHAR	6	-		Receiptnist gender
STATUS	VARCHAR	20	-		Receiptnist status
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Receiptnist username
PASSWORD	VARCHAR	50	-		Receiptnist password

Table description :This table keeps information of Receiptnist Registration.

TABLE:MEDICAL REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Medical id
NAME	VARCHAR	50	-		Medical name
ADDRESS	VARCHAR	150	-		Medical address
CONTACT	INT	12	-		Medical contact
EMAIL	VARCHAR	50	-		Medical email
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Medical username
PASSWORD	VARCHAR	50	-		Medical password

Table description: This table keeps information medical registration.

TABLE:MASTER PRESCRITION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Prescription id
DATE	DATE	10	-	-	Date
DOCTOR_ID	INT	10	-	-	Doctor id
PATIENT_ID	INT	10	-	-	Patient id

Table description: This table keeps information of master prescription.

TABLE :-PRESCRIPTION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Incremented id
P_ID	INT	10	-	1	Prescription id
NAME	VARCHAR	50	-		Medical name
QUANTITY	VARCHAR	50	-		Medicine quantity
MORNING	VARCHAR	50	-		Dose1
AFTERNOON	VARCHAR	50	-		Dose2
EVENING	VARCHAR	50	-		Dose3

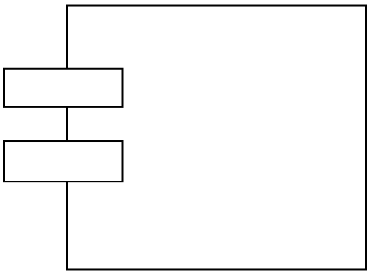
Table description: This table keeps information of prescription.

COMPONENT DIAGRAM

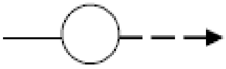
1] Presence of association and role information shows that the diagram represents classes of nodes .the diagram shows a system composed of a server with surrounding PC's that drive the opening and closing gates. The number of PC's has not been established conversely it appears that every PC may control at most 10 gates.3 x terminal play the role of console to access the system. A printer is connected to the server.

2] Component diagram describes software component and their relationship within the implementation environment.

Symbols used for component Diagram:

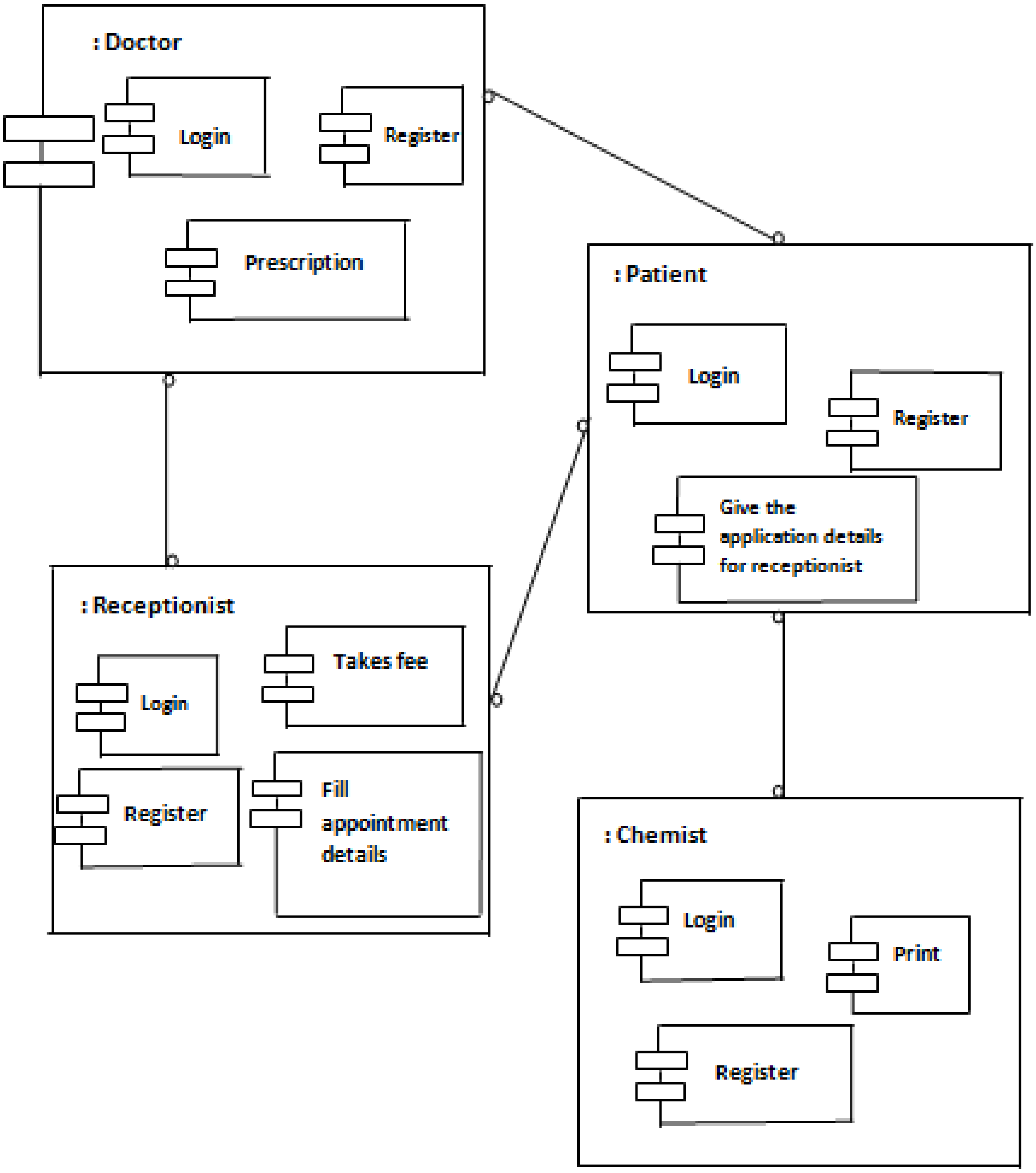


-Component



-lollipop (use to show the specification)

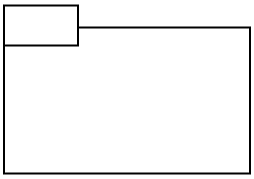
COMPONENT DIAGRAM



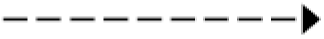
PACKAGE DIAGRAM

It is used to identify major components of system .To simplify complex class diagram we can group classes into package .package is a collection of logically related UML elements.

Symbols used for package Diagram:

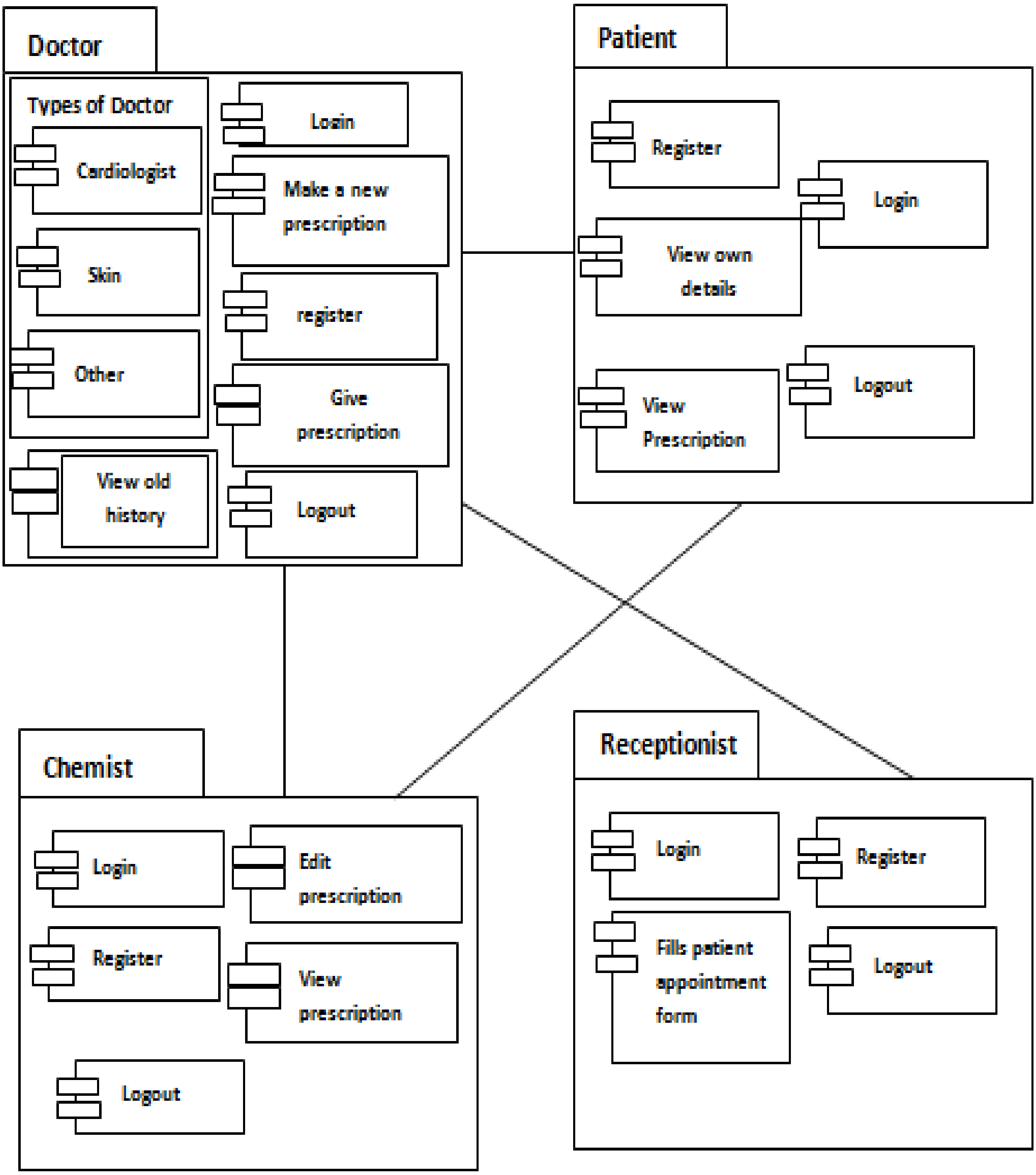


-Tabbed rectangle/folder (show subsystem)



-Dashed line (show dependency).

Package Diagram

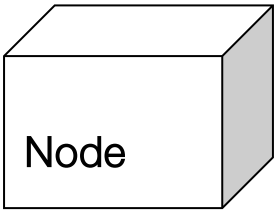


DEPLOYMENT DIAGRAM

The diagram shows physical layout various hardware component that compose a system as well as the substitution of executable program on this hardware.

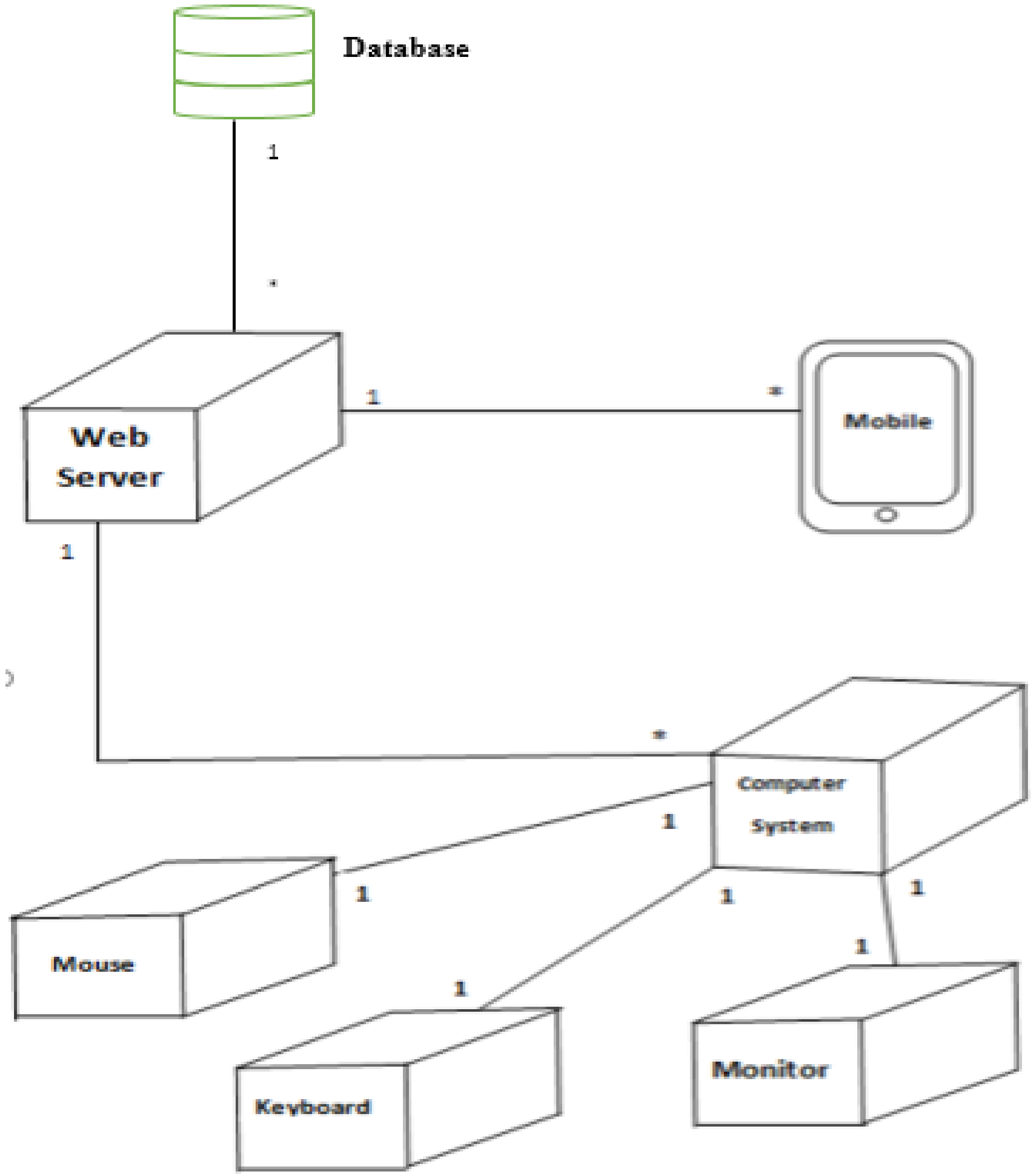
Representation of node:

Each hardware resource is represented by a queue .Indicating the physical presence of the equipment in the system.



The nature of the equipment may be specified using a stereotype

Deployment Diagram



CHAPTER – IV

SYSTEMCODING

LIST OF TABLES WITHATTRIBUTE AND CONSTRAINTS

TABLE :DOCTOR REGISTRATION TABLE

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Doctor id
NAME	VARCHAR	50	-		Doctor name
ADDRESS	VARCHAR	150	-		Doctor address
CONTACT	INT	12	-		Doctor contact
EMAIL	VARCHAR	50	-		Doctor email
QUALIFICATION	VARCHAR	20	-		Doctor qualification
GENDER	VARCHAR	6	-		Doctor gender
TYPE_DOCTOR	VARCHAR	20	-		Doctor type-doctor
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Doctor username
PASSWORD	VARCHAR	50	-		Doctor password

Table description : This table keeps information of Doctor.

TABLE :PATIENT REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Patient id
NAME	VARCHAR	50	-		Patient name
ADDRESS	VARCHAR	150	-		Patient address
CONTACT	INT	12	-		Patient contact
EMAIL	VARCHAR	50	-		Patient email
GENDER	VARCHAR	20	-		Patient gender
BLOOD GROUP	VARCHAR	6	-		Patient blood group
STATUS	VARCHAR	20	-		Patient status
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Patient username
PASSWORD	VARCHAR	50	-		Patient password

Table description : This table keeps information of patient registration.

TABLE : Appointment Registration

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Patient id
NAME	VARCHAR	50	-		Patient name
ADDRESS	VARCHAR	150	-		Patient address
CONTACT	INT	12	-		Patient contact
EMAIL	VARCHAR	50	-		Patient email
GENDER	VARCHAR	20	-		Patient gender
DOCTOR NAME	VARCHAR	20	-		Doctor name
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Patient username
PASSWORD	VARCHAR	50	-		Patient password

Table description: This table keeps information Appointment of the patient

TABLERECEIPTNIST REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Receiptnist id
NAME	VARCHAR	50	-		Receiptnist name
ADDRESS	VARCHAR	150	-		Receiptnist address
CONTACT	INT	12	-		Receiptnist contact
EMAIL	VARCHAR	50	-		Receiptnist email
QUALIFICATION	VARCHAR	20	-		Receiptnist qualification
GENDER	VARCHAR	6	-		Receiptnist gender
STATUS	VARCHAR	20	-		Receiptnist status
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Receiptnist username
PASSWORD	VARCHAR	50	-		Receiptnist password

Table description: This table keeps information of Receiptnist Registration.

TABLE:MEDICAL REGISTRATION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Medical id
NAME	VARCHAR	50	-		Medical name
ADDRESS	VARCHAR	150	-		Medical address
CONTACT	INT	12	-		Medical contact
EMAIL	VARCHAR	50	-		Medical email
DATE	DATE	10	-		Date
USERNAME	VARCHAR	50	-		Medical username
PASSWORD	VARCHAR	50	-		Medical password

Table description: This table keeps information medical registration.

TABLE:MASTER PRESCRITION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Prescription id
DATE	DATE	10	-	-	Date
DOCTOR_ID	INT	10	-	-	Doctor id
PATIENT_ID	INT	10	-	-	Patient id

Table description: This table keeps information of master prescription.

TABLE :-PRESCRIPTION

Column	Data Type	Length	Precision	Primary Key	Description
ID	INT	10	-	1	Incremented id
P_ID	INT	10	-	1	Prescription id
NAME	VARCHAR	50	-		Medical name
QUANTITY	VARCHAR	50	-		Medicine quantity
MORNING	VARCHAR	50	-		Dose1
AFTERNOON	VARCHAR	50	-		Dose2
EVENING	VARCHAR	50	-		Dose3

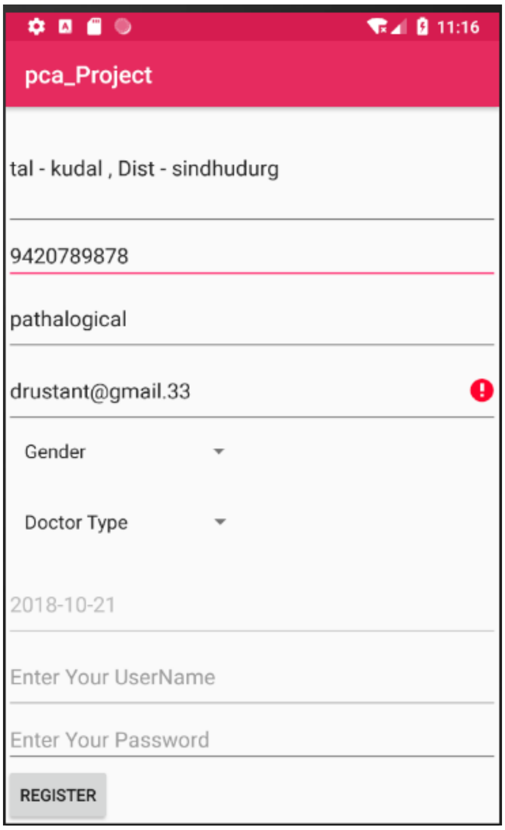
Table description: This table keeps information of prescription.

Validation

While any application can be designed with sound logic and good technology and deliver high performance with accuracy, some errors could still creep into it. This could be due to wrong inputs by users. While the programmer may have taken care of all the exception it could cause a loss of business good will if a customer is confronted with an error message after he has enter a valid "please enter a valid" or so on.

The type property can have any of the following values:

- String
- Integer
- Double
- Date


11:16

pca_Project

tal - kudal , Dist - sindhudurg

9420789878

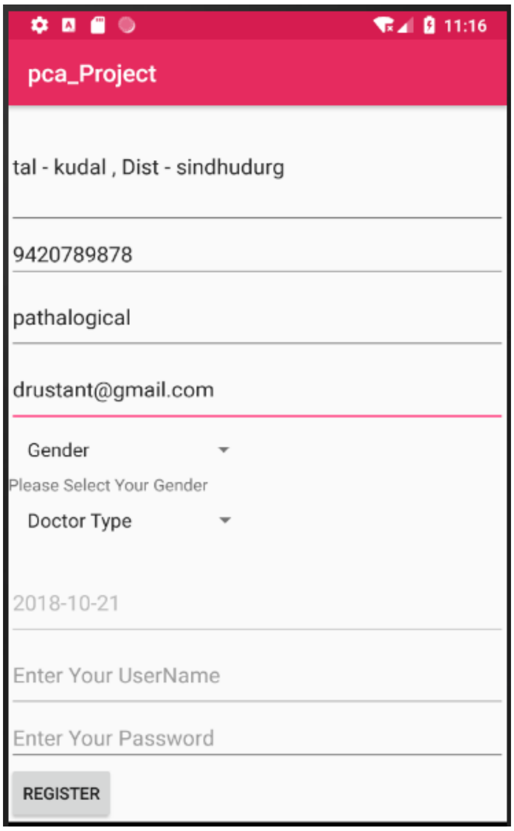
pathalogical

drustant@gmail.33 ❗

2018-10-21

Enter Your UserName

Enter Your Password


11:16

pca_Project

tal - kudal , Dist - sindhudurg

9420789878

pathalogical

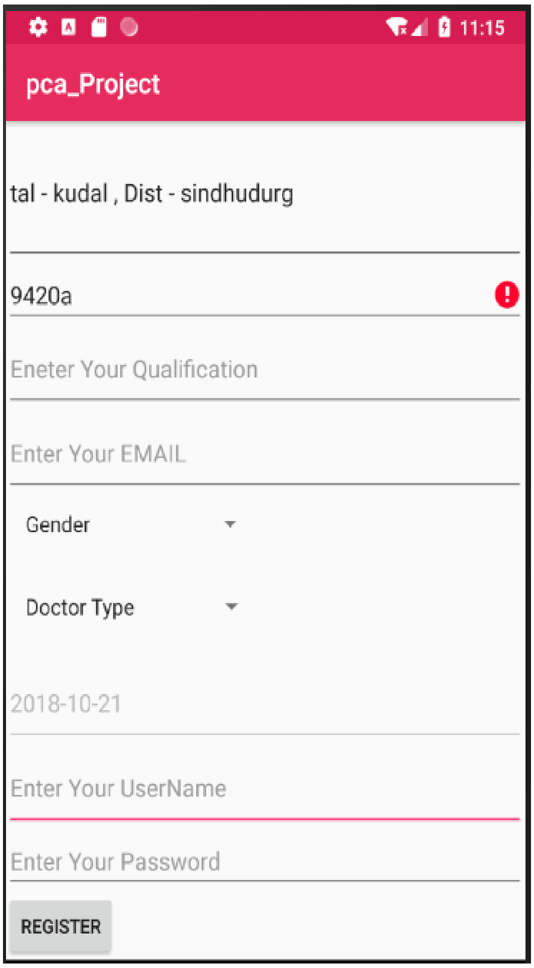
drustant@gmail.com

Please Select Your Gender

2018-10-21

Enter Your UserName

Enter Your Password



pca_Project

tal - kudal , Dist - sindhudurg

9420a

Eneter Your Qualification

Enter Your EMAIL

Gender

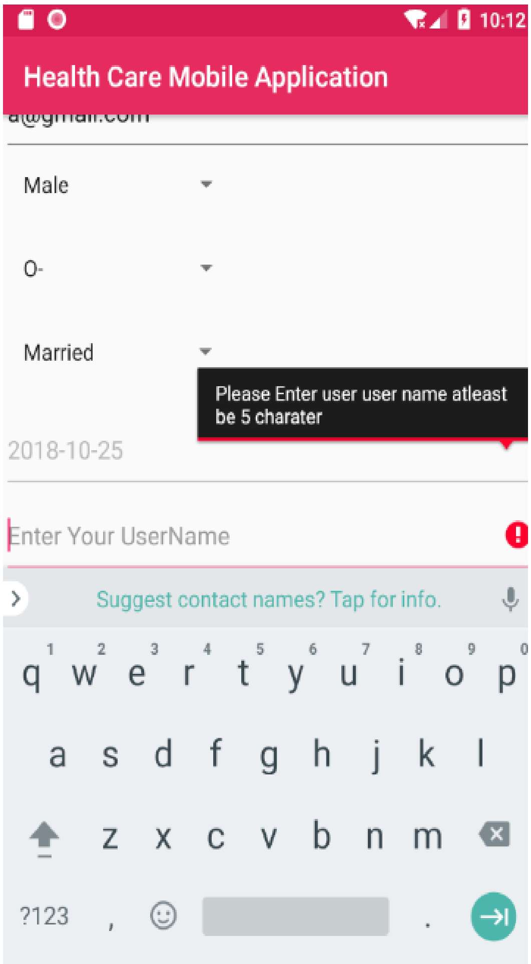
Doctor Type

2018-10-21

Enter Your UserName

Enter Your Password

REGISTER



Health Care Mobile Application

a@gmail.com

Male

O-

Married

Please Enter user user name atleast be 5 charater

2018-10-25

Enter Your UserName

Suggest contact names? Tap for info.

Validation Code :

```
   @Override
public void onClick(View view)
   {
      if(view==reg)
      {
         String tid=id.getText().toString();
         String tname=name.getText().toString();
         String taddress=address.getText().toString();
         String tcont=cont.getText().toString();
         String tqua=qua.getText().toString();
         String tuser_name=user_name.getText().toString();
         String tpassword=password.getText().toString();
         String temail=email.getText().toString();
         String sgender=gender.getSelectedItem().toString();
         String stype_doctor=type_doctor.getSelectedItem().toString();
         String tdate=date.getText().toString();
if(!isValidateId(tid))
         {
id.setError("please enter the more then 5 didit ID");
         }
         else
         {
if(!isValidateName(tname))
            {
name.setError("Please Enter Your Name");
            }
            else
            {
if(address.length()==0)
```

```
          {
address.setError("Please Enter Your Address");
          }
          else
          {
if(!isValidatePhone(tcont))
            {
cont.setError("Please Enter Your Contact Number");
            }
            else
            {
if(qua.length()==0)
              {
qua.setError("Please Enter Your Qualification");
              }
              else
              {
if(!isValidEmail(temail))
                {
email.setError("Please Enter Your Email Address");
                }
                else
                {
if(gender.getSelectedItemPosition()==0)
                  {
t1.setText("Please Select Your Gender");
                  }
                  else
                  {
t1.setText("");
if(type_doctor.getSelectedItemPosition()==0)
                    {
```

```
t2.setText("Please Doctor Type");
                    }
                    else
                    {
t2.setText("");
if(user_name.length()<=5)
                      {
                          user_name.setError("Please Enter user user name atleast be 5 charater");
                      }
                      else
                      {
if(password.length()<=5)
                        {
password.setError("please Enter Your Password atleast more than 5 charater");
                        }
                        else
                        {
t2.setText("");
                          String type="doctor_register";
                          Doctor_Registration_Form_Back drb=new
                              Doctor_Registration_Form_Back(context);
drb.execute(type,tid,tname,taddress,tcont,temail,tqua,sgender,
                              stype_doctor,tdate,tuser_name,
                              tpassword);
Toast.makeText(this,"Record Added",Toast.LENGTH_LONG).show();
                        }
                      }
                    }
                  }
                }
              }
            }
```

```
                    }
                }
            }
        }
    }
    // validations function
private boolean isValidEmail(String email)
    {
        String email_pattern="^[A-Za-z0-9._%+\\-]+@[A-Za-z0-9.\\-]+\\.[A-Za-z]{2,4}$";
        Pattern pattern=Pattern.compile(email_pattern);
        Matcher matcher=pattern.matcher(email);
return matcher.matches();
    }
private boolean isValidateName(String name)
    {
        String name_pattern="[a-zA-z]+([ '-][a-zA-Z]+)*";
        Pattern pattern=Pattern.compile(name_pattern);
        Matcher matcher=pattern.matcher(name);
return matcher.matches();
    }
private boolean isValidatePhone(String phone)
    {
final String phone_pattern="\\d{10}";
        Pattern pattern=Pattern.compile(phone_pattern);
        Matcher matcher=pattern.matcher(phone);
return matcher.matches();
    }
private boolean isValidateId(String id)
    {
final String id_pattern="\\d{6}";
        Pattern pattern=Pattern.compile(id_pattern);
        Matcher matcher=pattern.matcher(id);
```

```
return matcher.matches();

    }

// Doctor name fetch in spinner

    @Override

    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {

    }

    @Override

    public void onNothingSelected(AdapterView<?> adapterView) {

    }

}
```

Doctor Registration Form

```
package com.example.drustant.pca_project;
import android.content.Context;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONObject;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Doctor_Registration_Form extends AppCompatActivity implements
View.OnClickListener,AdapterView.OnItemSelectedListener
{
    Button reg;
    EditText id,name,address,cont,email,user_name,password,qua,date;
    Spinner gender,type_doctor;
    Context context;
    TextView t1,t2;

    InputStream is1234 = null;
    String line1234 = null;
    String result1234 = null;
    String[] data;
    String address1234 = "http://10.0.2.2/pca/date_script.php";

    String[] gender_array={"Gender","Male","Female"};

    String[] doc_type_array={"Doctor
Type","psychatrist","Dentist","Pathalogist","Pharmacist","Neurologist"};
    ArrayAdapter<String> gender_adapter,doc_type,adapter;

    InputStream is=null;
    String line=null;
    String result=null;
 //   String[] data;
    String file_path="http://10.0.2.2/pca/Doctor_max_id.php";

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_doctor__registration__form);

        context=this;
        super.setTitle("Health Care Mobile Application");

        id=(EditText)findViewById(R.id.editText);
        name=(EditText)findViewById(R.id.name);
        address=(EditText)findViewById(R.id.address);
        cont=(EditText)findViewById(R.id.cont);
        email=(EditText)findViewById(R.id.email);
        qua=(EditText)findViewById(R.id.qua);
        user_name=(EditText)findViewById(R.id.uname);
        password=(EditText)findViewById(R.id.upass);
        gender=(Spinner)findViewById(R.id.gender);
        type_doctor=(Spinner)findViewById(R.id.type);
        t1=(TextView)findViewById(R.id.t1);
        t2=(TextView)findViewById(R.id.t2);
        date=(EditText)findViewById(R.id.date);

        id.setEnabled(false);
        date.setEnabled(false);
        gender_adapter=new
ArrayAdapter<String>(getApplicationContext(),android.R.layout
                .simple_list_item_1,gender_array);
        doc_type=new
ArrayAdapter<String>(getApplicationContext(),android.R.layout
                .simple_list_item_1,doc_type_array);

        gender.setAdapter(gender_adapter);
        type_doctor.setAdapter(doc_type);

        StrictMode.setThreadPolicy(new
StrictMode.ThreadPolicy.Builder().permitNetwork().build());
        getData();
        getData1234();

        type_doctor.setOnItemSelectedListener(this);
        gender.setOnItemSelectedListener(this);
        reg=(Button)findViewById(R.id.reg);
        reg.setOnClickListener(this);
    }

    private void getData1234()
    {
        try {
           // String find=tid.getText()+"";
            URL url=new URL(address1234);
            HttpURLConnection con=(HttpURLConnection) url.openConnection();
            con.setRequestMethod("GET");
            is1234=new BufferedInputStream(con.getInputStream());


        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }


        try
        {
            BufferedReader br=new BufferedReader(new
InputStreamReader(is1234));
            StringBuilder sb=new StringBuilder();
            while ((line1234=br.readLine())!=null)
            {
                sb.append(line1234+"\n");
            }
            is1234.close();
            result1234=sb.toString();
        }
        catch(Exception e)
        {
        }
        try
        {
            JSONArray ja=new JSONArray(result1234);
            JSONObject jo=null;
            data=new String[ja.length()];
            //    tid.setText(data[0]=jo.getString("id"));
            jo=ja.getJSONObject(0);
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));


            for(int i=0;i<ja.length();i++)
            {
                jo=ja.getJSONObject(i);

                date.setText(data[i]=jo.getString("curdate()")+"");
                // tid.setText(data[i]=jo.getString("id")+"");
            }


        }
        catch (Exception e)
        {

        }

    }




    private void getData()
    {
```

```
    try {

       // String find=tid.getText()+"";
        URL url=new URL(file_path);
        HttpURLConnection con=(HttpURLConnection) url.openConnection();
        con.setRequestMethod("GET");
        is=new BufferedInputStream(con.getInputStream());


    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }


    try
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(is));
        StringBuilder sb=new StringBuilder();
        while ((line=br.readLine())!=null)
        {
            sb.append(line+"\n");
        }
        is.close();
        result=sb.toString();
    }
    catch(Exception e)
    {
    }
    try
    {
        JSONArray ja=new JSONArray(result);
        JSONObject jo=null;
        data=new String[ja.length()];
        //    tid.setText(data[0]=jo.getString("id"));
        jo=ja.getJSONObject(0);
        //tid.setText(data[0]=jo.getString("id"));
        //tid.setText(data[0]=jo.getString("id"));
        //tid.setText(data[0]=jo.getString("id"));

        for(int i=0;i<ja.length();i++)
        {
            jo=ja.getJSONObject(i);

            // tid.setText(data[i]=jo.getString("curdate()")+"");
            id.setText(data[i]=jo.getString("max(id)+1")+"");
        }


    }
    catch (Exception e)
    {

    }

}



@Override
public void onClick(View view)
{
        if(view==reg)
        {
            String tid=id.getText().toString();
            String tname=name.getText().toString();
            String taddress=address.getText().toString();
            String tcont=cont.getText().toString();
            String tqua=qua.getText().toString();
            String tuser_name=user_name.getText().toString();
            String tpassword=password.getText().toString();
            String temail=email.getText().toString();
            String sgender=gender.getSelectedItem().toString();
            String stype_doctor=type_doctor.getSelectedItem().toString();
            String tdate=date.getText().toString();


            if(!isValidateId(tid))
            {
                id.setError("please enter the more then 5 didit ID");
            }
            else
            {
                if(!isValidateName(tname))
                {
                    name.setError("Please Enter Your Name");
                }
                else
                {
                    if(address.length()==0)
                    {
                        address.setError("Please Enter Your Address");
                    }
                    else
                    {
                        if(!isValidatePhone(tcont))
                        {
                            cont.setError("Please Enter Your Contact Number");
                        }
                        else
                        {
                            if(qua.length()==0)
                            {
                                qua.setError("Please Enter Your
Qualification");
                            }
                            else
                            {
                                if(!isValidEmail(temail))
                                {
                                    email.setError("Please Enter Your Email
Address");
                                }
                                else
                                {
                                    if(gender.getSelectedItemPosition()==0)
                                    {
                                        t1.setText("Please Select Your
Gender");
                                    }
                                    else
                                    {
                                        t1.setText("");

if(type_doctor.getSelectedItemPosition()==0)
                                        {
                                            t2.setText("Please Doctor Type");
                                        }
                                        else
                                        {
                                            t2.setText("");
                                            if(user_name.length()<=5)
                                            {
                                                user_name.setError("Please 
Enter user user name atleast be 5 charater");
                                            }
                                            else
                                            {
                                                if(password.length()<=5)
                                                {
                                                    password.setError("please 
Enter Your Password atleast more than 5 charater");
                                                }
                                                else
                                                {
                                                    t2.setText("");
                                                    String 
type="doctor_register";

Doctor_Registration_Form_Back drb=new 

Doctor_Registration_Form_Back(context);

drb.execute(type,tid,tname,taddress,tcont,temail,tqua,sgender,

stype_doctor,tdate,tuser_name,
                                                            tpassword);
                                                    Toast.makeText(this,"Record 
Added",Toast.LENGTH_LONG).show();
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
                    /*
                    else
                    {
                        t1.setText("");
                    }

                    else
                    {
                        t2.setText("");
                    }
*/

        }
    }
    // validations function
    private boolean isValidEmail(String email)
    {
        String email_pattern="^[A-Za-z0-9._%+\\-]+@[A-Za-z0-9.\\-]+\\.[A-Za-
z]{2,4}$";
        Pattern pattern=Pattern.compile(email_pattern);
        Matcher matcher=pattern.matcher(email);
        return matcher.matches();
    }
    private boolean isValidateName(String name)
    {
        String name_pattern="[a-zA-z]+([ '-][a-zA-Z]+)*";
        Pattern pattern=Pattern.compile(name_pattern);
        Matcher matcher=pattern.matcher(name);
        return matcher.matches();
    }
    private boolean isValidatePhone(String phone)
    {
        final String phone_pattern="\\d{10}";
        Pattern pattern=Pattern.compile(phone_pattern);
        Matcher matcher=pattern.matcher(phone);
        return matcher.matches();
    }
    private boolean isValidateId(String id)
    {
        final String id_pattern="\\d{6}";
        Pattern pattern=Pattern.compile(id_pattern);
        Matcher matcher=pattern.matcher(id);
        return matcher.matches();
    }

    // Doctor name fetch in spinner

    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i,
long l) {

    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {

    }
}
```

Doctor registration form back

```
package com.example.drustant.pca_project;

import android.app.AlertDialog;
import android.content.Context;
import android.os.AsyncTask;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
public class Doctor_Registration_Form_Back extends
AsyncTask<String,Void,String>
{
    Context context;
    AlertDialog alertDialog;
    Doctor_Registration_Form_Back(Context ctx)
    {
        context=ctx;
    }
    @Override
    protected String doInBackground(String... params)
    {
        String type=params[0];
        //String type="login";
        System.out.println("emulator ip address");
        String login_url="http://10.0.2.2/pca/doctor_registration.php";

        // String login_url="http://192.168.1.101/login.php";

        if(type.equals("doctor_register"))
        {
            try {

                String tid=params[1];
                String tname=params[2];
                String taddress=params[3];
                String tcont=params[4];
                String temail=params[5];
                String tqua=params[6];
                String sgender=params[7];
                String stype_doctor=params[8];
                String tdate=params[9];
                String tuser_name=params[10];
                String tpassword=params[11];


                URL url=new URL(login_url);
                HttpURLConnection
httpURLConnection=(HttpURLConnection)url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                httpURLConnection.setDoInput(true);
                OutputStream outputStream=httpURLConnection.getOutputStream();
                BufferedWriter bufferedWriter=new BufferedWriter(new
OutputStreamWriter(outputStream,"UTF-8"));

                String post_data= URLEncoder.encode("tid","UTF-
8")+"="+URLEncoder.encode
                        (tid,"UTF-8")+
                        "&&"+URLEncoder.encode("tname","UTF-
8")+"="+URLEncoder.encode
                        (tname,"UTF-8")+
                        "&&"+URLEncoder.encode("taddress","UTF-
8")+"="+URLEncoder.encode(taddress,
                        "UTF-8")+
                        "&&"+URLEncoder.encode("tcont","UTF-
8")+"="+URLEncoder.encode
                        (tcont,"UTF-8")+
                        "&&"+URLEncoder.encode("temail","UTF-
8")+"="+URLEncoder.encode(temail,
                        "UTF-8")+
                        "&&"+URLEncoder.encode("tqua","UTF-
8")+"="+URLEncoder.encode
                        (tqua,"UTF-8")+
                        "&&"+URLEncoder.encode("sgender","UTF-
8")+"="+URLEncoder.encode
                        (sgender,"UTF-8")+
                        "&&"+URLEncoder.encode("stype_doctor","UTF-
8")+"="+URLEncoder.encode
```

```
                         (stype_doctor,"UTF-8")+
                         "&&"+URLEncoder.encode("date","UTF-
8")+"="+URLEncoder.encode
                         (tdate,"UTF-8")+
                         "&&"+URLEncoder.encode("tuser_name","UTF-
8")+"="+URLEncoder.encode
                         (tuser_name,
                                 "UTF-8")+
                         "&&"+URLEncoder.encode("tpassword","UTF-
8")+"="+URLEncoder.encode
                         (tpassword,
                                 "UTF-8");

                bufferedWriter.write(post_data);
                bufferedWriter.flush();;
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream=httpURLConnection.getInputStream();
                BufferedReader bufferedReader=new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
                String result="";
                String line="";

                while((line=bufferedReader.readLine())!=null)
                {
                    result+=line;
                }
                bufferedReader.close();
                inputStream.close();
                httpURLConnection.disconnect();
                return result;
            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return null;
    }
    @Override
    protected void onPreExecute() {
        // super.onPreExecute();
        alertDialog=new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Login Status.......");
    }
    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        //alertDialog=new AlertDialog.Builder(context).create();
        // alertDialog.setTitle("Login Status..");
        alertDialog.setMessage(result);
        alertDialog.show();
    }
    @Override
    protected void onProgressUpdate(Void... values) {
        super.onProgressUpdate(values);

    }
}
```

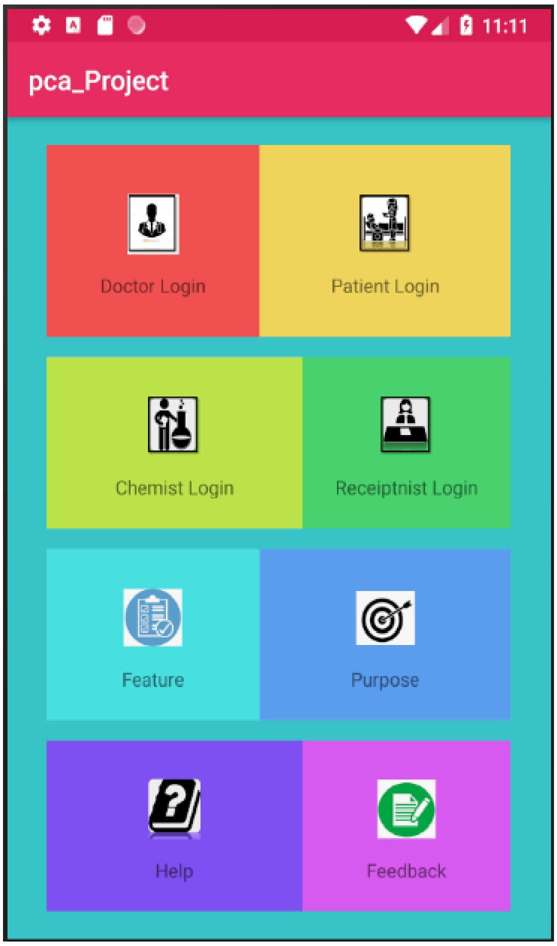
Doctor registration form xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
```

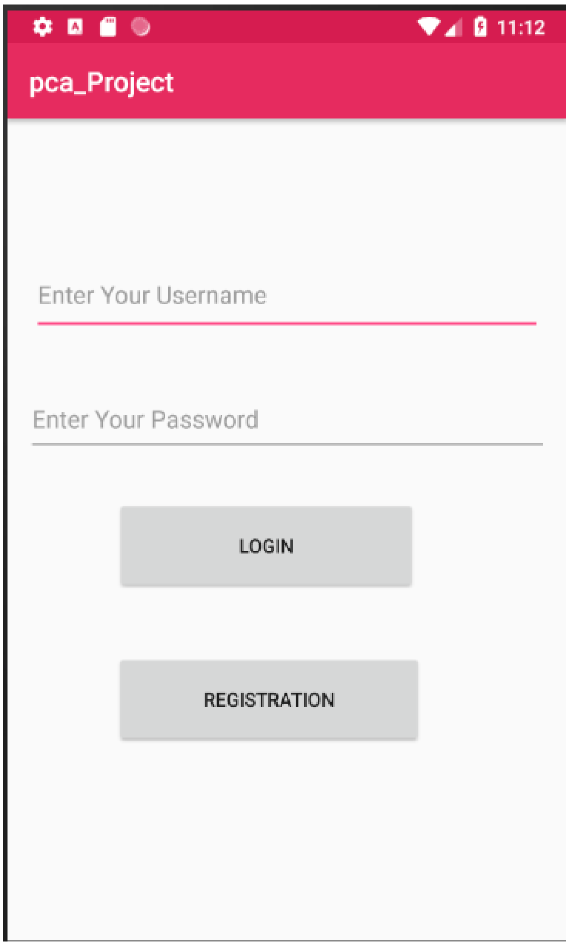
```
android:layout_height="match_parent"
android:layout_width="match_parent"
xmlns:android="http://schemas.android.com/apk/res/android" >



<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    tools:context=".Medical_Registration_Form"
    android:orientation="vertical"
    tools:showIn="@layout/activity_doctor__registration__form">



    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="61dp"
        android:ems="10"
        android:hint="Enter Your ID"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="61dp"
        android:ems="10"
        android:hint="enter_your_name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/address"
        android:layout_width="match_parent"
        android:layout_height="103dp"
        android:ems="10"
        android:hint="Enter Your Address"
        android:inputType="textPostalAddress" />

    <EditText
        android:id="@+id/cont"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Contact Number"
        android:inputType="textPersonName" />
    <EditText
        android:id="@+id/qua"
        android:layout_width="match_parent"
        android:layout_height="61dp"
        android:ems="10"
        android:hint="Eneter Your Qualification"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="61dp"
        android:ems="10"
        android:hint="Enter Your EMAIL"
        android:inputType="textPersonName" />
        <Spinner
            android:id="@+id/gender"
            android:layout_width="202dp"
            android:layout_height="40dp">

        </Spinner>

        <TextView
            android:id="@+id/t1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <Spinner
            android:id="@+id/type"
            android:layout_width="203dp"
            android:layout_height="40dp">

        </Spinner>


        <TextView
            android:id="@+id/t2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <EditText
            android:id="@+id/date"
            android:layout_width="match_parent"
            android:layout_height="61dp"
            android:ems="10"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/uname"
            android:layout_width="match_parent"
            android:layout_height="61dp"
            android:ems="10"
            android:hint="Enter Your UserName"
            android:inputType="textPersonName" />

        <EditText
            android:id="@+id/upass"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="Enter Your Password"
            android:inputType="textPassword" />

        <Button
            android:id="@+id/reg"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Register" />



    </LinearLayout>
</ScrollView>
```

SCREEN LAYOUTS & REPORT LAYOUTS

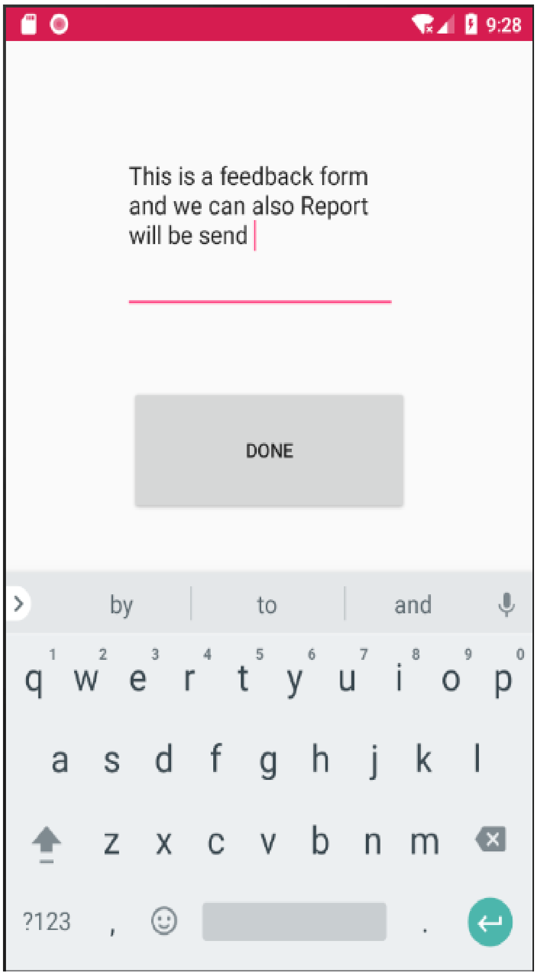
HOME PAGE



LOGIN PAGE

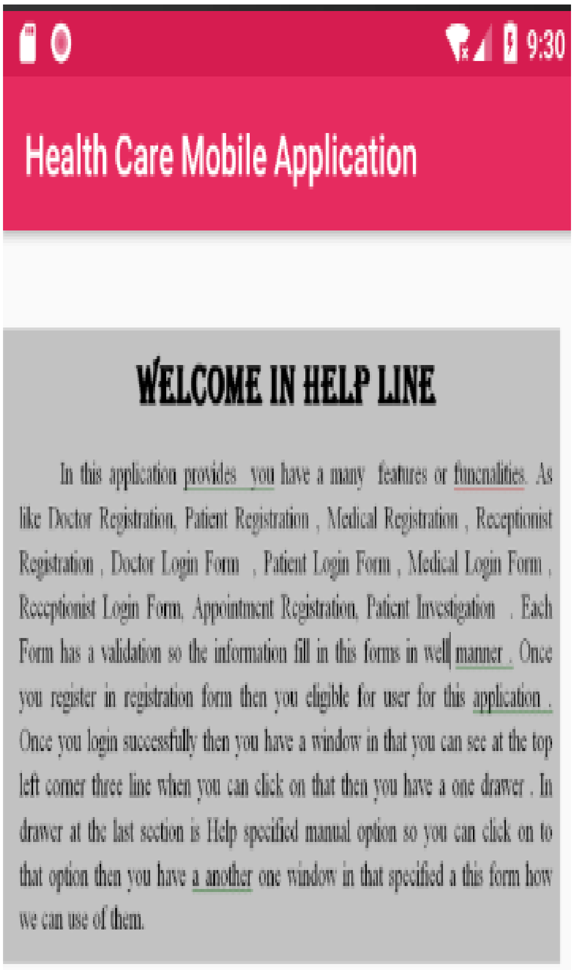


FEEEDBACK FORM



This is a feedback form and we can also Report will be send

HELP PAGE

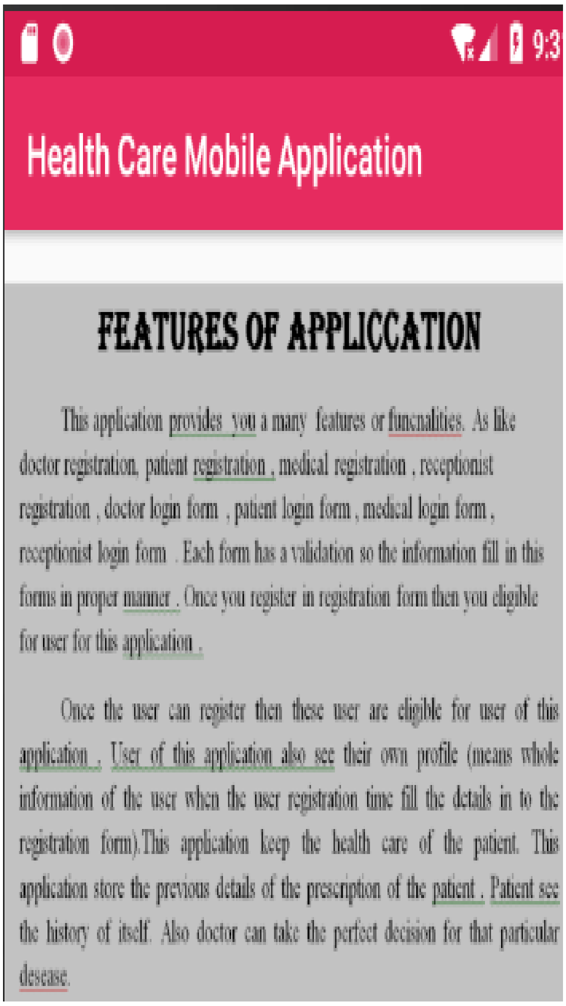


Health Care Mobile Application

WELCOME IN HELP LINE

In this application provides you have a many features or funcnalities. As like Doctor Registration, Patient Registration , Medical Registration , Receptionist Registration , Doctor Login Form , Patient Login Form , Medical Login Form , Receptionist Login Form, Appointment Registration, Patient Investigation . Each Form has a validation so the information fill in this forms in well manner . Once you register in registration form then you eligible for user for this application . Once you login successfully then you have a window in that you can see at the top left corner three line when you can click on that then you have a one drawer . In drawer at the last section is Help specified manual option so you can click on to that option then you have a another one window in that specified a this form how we can use of them.

FEATURE PAGE



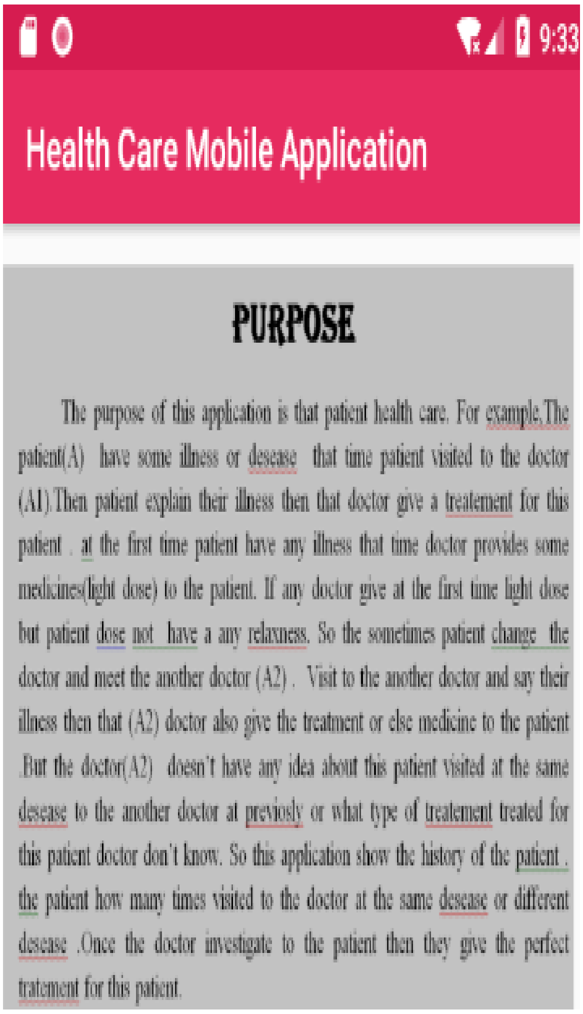
Health Care Mobile Application

FEATURES OF APPLICCATION

This application provides you a many features or funcnalities. As like doctor registration, patient registration , medical registration , receptionist registration , doctor login form , patient login form , medical login form , receptionist login form . Each form has a validation so the information fill in this forms in proper manner . Once you register in registration form then you eligible for user for this application .

Once the user can register then these user are eligible for user of this application . User of this application also see their own profile (means whole information of the user when the user registration time fill the details in to the registration form).This application keep the health care of the patient. This application store the previous details of the prescription of the patient . Patient see the history of itself. Also doctor can take the perfect decision for that particular desease.

PURPOSE PAGE

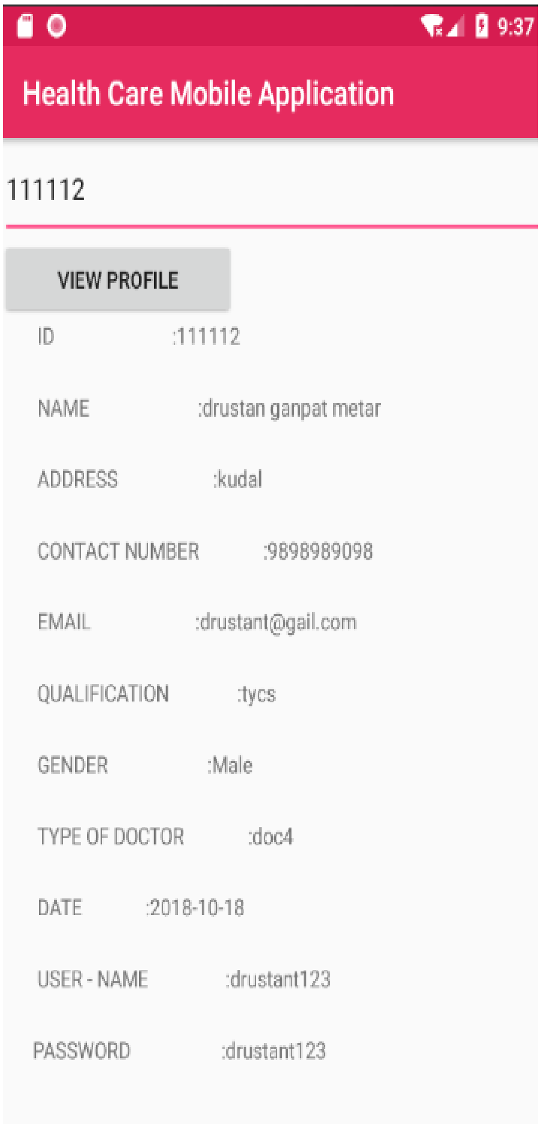


Health Care Mobile Application

PURPOSE

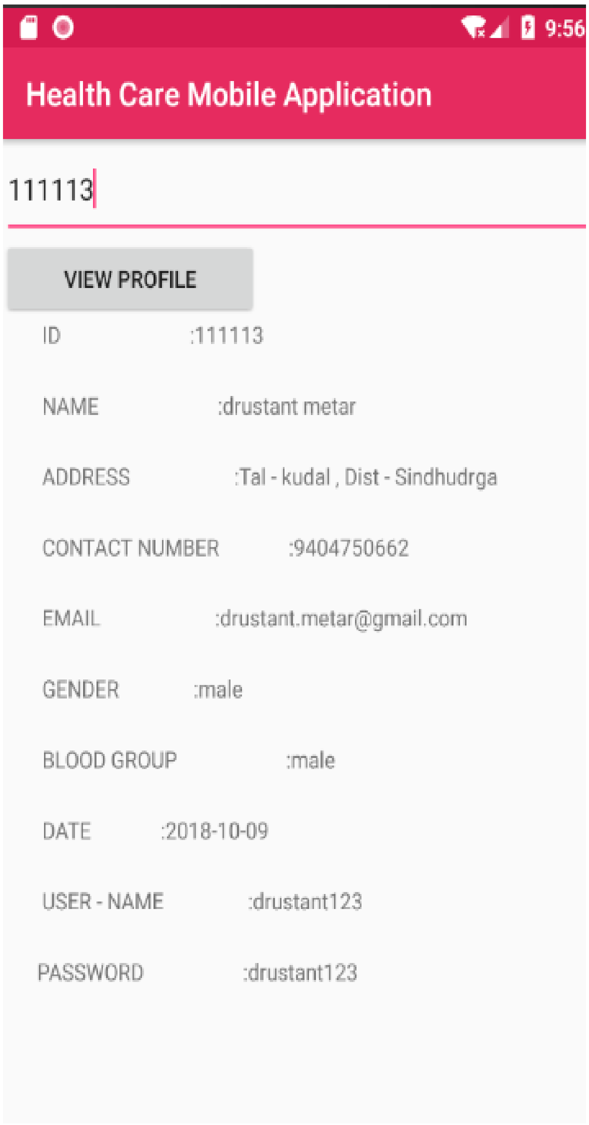
The purpose of this application is that patient health care. For example.The patient(A) have some illness or desease that time patient visited to the doctor (A1).Then patient explain their illness then that doctor give a treatement for this patient . at the first time patient have any illness that time doctor provides some medicines(light dose) to the patient. If any doctor give at the first time light dose but patient dose not have a any relaxness. So the sometimes patient change the doctor and meet the another doctor (A2) . Visit to the another doctor and say their illness then that (A2) doctor also give the treatment or else medicine to the patient .But the doctor(A2) doesn't have any idea about this patient visited at the same desease to the another doctor at previosly or what type of treatement treated for this patient doctor don't know. So this application show the history of the patient . the patient how many times visited to the doctor at the same desease or different desease .Once the doctor investigate to the patient then they give the perfect tratement for this patient.

DOCTOR PROFILE



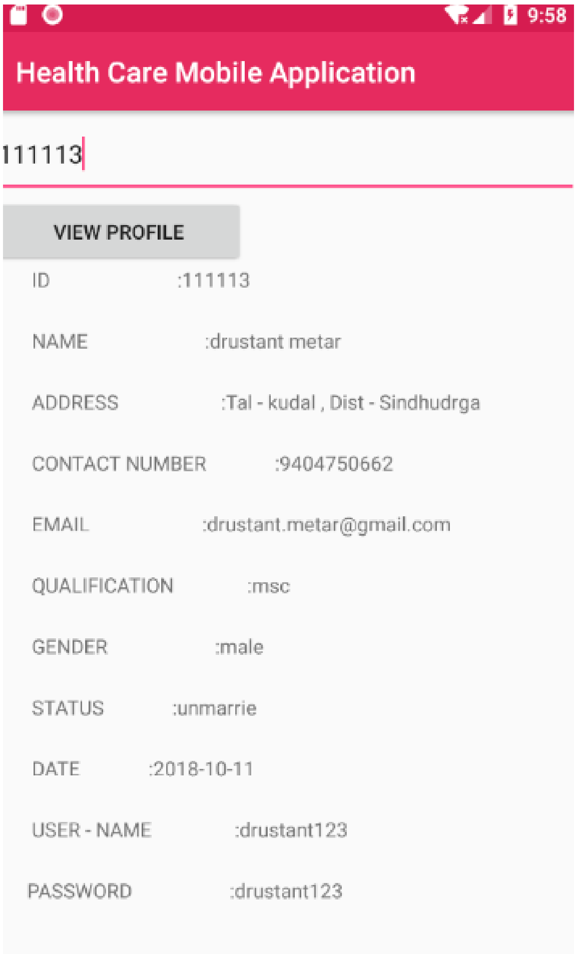
ID	:111112
NAME	:drustan ganpat metar
ADDRESS	:kudal
CONTACT NUMBER	:9898989098
EMAIL	:drustant@gail.com
QUALIFICATION	:tycs
GENDER	:Male
TYPE OF DOCTOR	:doc4
DATE	:2018-10-18
USER - NAME	:drustant123
PASSWORD	:drustant123

PATIENT PROFILE



ID	:111113
NAME	:drustant metar
ADDRESS	:Tal - kudal , Dist - Sindhudrga
CONTACT NUMBER	:9404750662
EMAIL	:drustant.metar@gmail.com
GENDER	:male
BLOOD GROUP	:male
DATE	:2018-10-09
USER - NAME	:drustant123
PASSWORD	:drustant123

RECEIPTNIST PROFILE



111113

ID :111113

NAME :drustant metar

ADDRESS :Tal - kudal , Dist - Sindhudrga

CONTACT NUMBER :9404750662

EMAIL :drustant.metar@gmail.com

QUALIFICATION :msc

GENDER :male

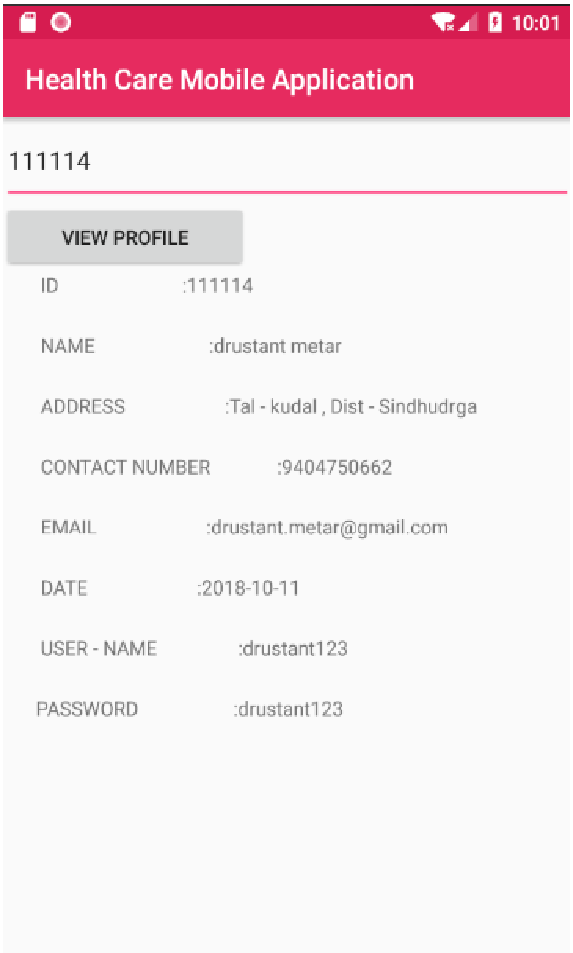
STATUS :unmarrie

DATE :2018-10-11

USER - NAME :drustant123

PASSWORD :drustant123

MEDICAL PROFILE



111114

ID :111114

NAME :drustant metar

ADDRESS :Tal - kudal , Dist - Sindhudrga

CONTACT NUMBER :9404750662

EMAIL :drustant.metar@gmail.com

DATE :2018-10-11

USER - NAME :drustant123

PASSWORD :drustant123

CODING

Login_Page.java

```
package com.example.drustant.pca_project;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;


public class Doctor_Login_Form extends AppCompatActivity implements
View.OnClickListener
{
    EditText user,pwd;
    Button log,reg;
    Context context;


    InputStream is=null;
    String line=null;
    String result=null;
String[] data;
    String idfetch;
    TextView id12;
    String file_path="http://10.0.2.2/pca/doctor_login_form.php";
    @Override
protected void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_doctor__login__form);
        context=this;
user=(EditText)findViewById(R.id.username);
pwd=(EditText)findViewById(R.id.password);
       // id12=(TextView)findViewById(R.id.id12);
log=(Button)findViewById(R.id.b1);
        reg=(Button)findViewById(R.id.reg);
        //StrictMode.setThreadPolicy(new
StrictMode.ThreadPolicy.Builder().permitNetwork().build());
        //getData();
log.setOnClickListener(this);
reg.setOnClickListener(this);
    }
    @Override
public void onClick(View view)
```

```
    {
        String tlog=user.getText().toString();
        String tpwd=pwd.getText().toString();


        if(view==log)
        {
            String type="doctor_login";
            Doctor_Login_Form_Back drfb=new Doctor_Login_Form_Back(context);
drfb.execute(type,tlog,tpwd);
user.setText("");
pwd.setText("");
            //   Toast.makeText(this,"login sucess",Toast.LENGTH_LONG).show();
        }
        if(view==reg)
        {
            Intent in=new
Intent("com.example.drustant.pca_project.Doctor_Registration_Form");
context.startActivity(in);
        }
    }
}
```

Login_Page_Back.java

```
package com.example.drustant.pca_project;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
public class Doctor_Login_Form_Back extends AsyncTask<String,Void,String> {
    Context context;
    AlertDialog alertDialog;
    // Intent in;
    Doctor_Login_Form_Back(Context ctx) {
        context = ctx;
    }
    String tusername, tpassword;
```

```
  @Override
protected String doInBackground(String... params) {
        String type = params[0];
        //String type="login";
        System.out.println("emulator ip address");
        //String login_url = "http://10.0.2.2/pca/adminlogin.php";
        String login_url = "http://10.0.2.2/pca/doctor_login_form.php";
      //  String login_url = "http://10.0.2.2/pca/medical_login.php";
        // String login_url="http://192.168.1.101/login.php";
tusername = params[1];
tpassword = params[2];
if (type.equals("doctor_login")) {
            try {
tusername = params[1];
tpassword = params[2];
                // Toast.makeText(this."username"+username,Toast.LENGTH_LONG).show();
                URL url = new URL(login_url);
                HttpURLConnection httpURLConnection = (HttpURLConnection)
url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);
                httpURLConnection.setDoInput(true);
                OutputStream outputStream = httpURLConnection.getOutputStream();
                BufferedWriter bufferedWriter = new BufferedWriter(new
OutputStreamWriter(outputStream, "UTF-8"));
                String post_data = URLEncoder.encode("username", "UTF-8") + "=" +
URLEncoder
                        .encode
                                (tusername, "UTF-8") +
                        "&&" + URLEncoder.encode("password", "UTF-8") + "=" +
URLEncoder.encode
```

```
                    (tpassword, "UTF-8");
                bufferedWriter.write(post_data);
                bufferedWriter.flush();
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream = httpURLConnection.getInputStream();
                BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "iso-8859-1"));
                String result = "";
                String line = "";
while ((line = bufferedReader.readLine()) != null) {
                    result += line;
                }
                bufferedReader.close();
                inputStream.close();
                httpURLConnection.disconnect();
                return result;
            } catch (MalformedURLException e) {
e.printStackTrace();
            } catch (IOException e) {
e.printStackTrace();
            }
        }
        return null;
    }
    @Override
protected void onPreExecute() {
        // super.onPreExecute();
alertDialog = new AlertDialog.Builder(context).create();
```

```
        //alertDialog.setTitle("Login Status........");
    }
    @Override
protected void onPostExecute(String ss) {
        //super.onPostExecute(result);
        //Toast.makeText(context,"post execution "+result,Toast.LENGTH_LONG).show();
switch (ss.charAt(0)) {
            case 'a':
Toast.makeText(context, "Succefully logged in..", Toast.LENGTH_LONG).show();
            case 'b':
Toast.makeText(context, "Succefully Updated Profile", Toast.LENGTH_LONG).show();
            case 'c':
Toast.makeText(context, "Login Sucessfully!", Toast.LENGTH_LONG).show();
                Intent in=new
Intent("com.example.drustant.pca_project.Doctor_Navigation_Drawer");
in.putExtra("username",tusername);
context.startActivity(in);
            default:
Toast.makeText(context, ss, Toast.LENGTH_LONG).show();
        }
    }
    @Override
    protected void onProgressUpdate (Void...values)
    {
super.onProgressUpdate(values);
    }
}
```

<u>Registration_page.xml</u>

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
android:layout_height="match_parent"
android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android" >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Medical_Registration_Form"
android:orientation="vertical"
        tools:showIn="@layout/activity_doctor__registration__form">
<EditText
            android:id="@+id/editText"
android:layout_width="match_parent"
android:layout_height="61dp"
android:ems="10"
android:hint="Enter Your ID"
android:inputType="textPersonName" />
<EditText
            android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="61dp"
```

```
android:ems="10"
android:hint="enter_your_name"
android:inputType="textPersonName" />

<EditText
        android:id="@+id/address"
android:layout_width="match_parent"
android:layout_height="103dp"
android:ems="10"
android:hint="Enter Your Address"
android:inputType="textPostalAddress" />
<EditText
        android:id="@+id/cont"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10"
android:hint="Contact Number"
android:inputType="textPersonName" />
<EditText
        android:id="@+id/qua"
android:layout_width="match_parent"
android:layout_height="61dp"
android:ems="10"
android:hint="Eneter Your Qualification"
android:inputType="textPersonName" />
<EditText
        android:id="@+id/email"
android:layout_width="match_parent"
android:layout_height="61dp"
```

```
android:ems="10"
android:hint="Enter Your EMAIL"
android:inputType="textPersonName" />

<Spinner
        android:id="@+id/gender"
android:layout_width="202dp"
android:layout_height="40dp">
</Spinner>
<TextView
        android:id="@+id/t1"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
<Spinner
        android:id="@+id/type"
android:layout_width="203dp"
android:layout_height="40dp">
</Spinner>
<TextView
        android:id="@+id/t2"
android:layout_width="match_parent"
android:layout_height="wrap_content" />
<EditText
        android:id="@+id/date"
android:layout_width="match_parent"
android:layout_height="61dp"
android:ems="10"
android:inputType="textPersonName" />
<EditText
```

```
            android:id="@+id/uname"
android:layout_width="match_parent"
android:layout_height="61dp"
android:ems="10"
android:hint="Enter Your UserName"
android:inputType="textPersonName" />
<EditText
            android:id="@+id/upass"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10"
android:hint="Enter Your Password"
android:inputType="textPassword" />
<Button
            android:id="@+id/reg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
            android:text="Register" />
</LinearLayout>
</ScrollView>
```

Registration_Page.java

```
package com.example.drustant.pca_project;
import android.content.Context;
import android.os.StrictMode;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class Doctor_Registration_Form extends AppCompatActivity implements
View.OnClickListener,AdapterView.OnItemSelectedListener
{
    Button reg;
    EditText id,name,address,cont,email,user_name,password,qua,date;
    Spinner gender,type_doctor;
    Context context;
    TextView t1,t2;
    InputStream is1234 = null;
    String line1234 = null;
    String result1234 = null;
String[] data;
    String address1234 = "http://10.0.2.2/pca/date_script.php";
String[] gender_array={"Gender","Male","Female"};
    String[] doc_type_array={"Doctor Type","doc1","doc2","doc3","doc4","doc5","doc6",
            "doc7"};
    ArrayAdapter<String> gender_adapter,doc_type,adapter;
    InputStream is=null;
    String line=null;
    String result=null;
 //   String[] data;
    String file_path="http://10.0.2.2/pca/Doctor_max_id.php";
    @Override
protected void onCreate(Bundle savedInstanceState)
    {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_doctor__registration__form);
        context=this;
```

```
id=(EditText)findViewById(R.id.editText);
name=(EditText)findViewById(R.id.name);
address=(EditText)findViewById(R.id.address);
cont=(EditText)findViewById(R.id.cont);
email=(EditText)findViewById(R.id.email);
        qua=(EditText)findViewById(R.id.qua);
        user_name=(EditText)findViewById(R.id.uname);
password=(EditText)findViewById(R.id.upass);
gender=(Spinner)findViewById(R.id.gender);
        type_doctor=(Spinner)findViewById(R.id.type);
        t1=(TextView)findViewById(R.id.t1);
        t2=(TextView)findViewById(R.id.t2);
date=(EditText)findViewById(R.id.date);
id.setEnabled(false);
date.setEnabled(false);
        gender_adapter=new ArrayAdapter<String>(getApplicationContext(),android.R.layout
                .simple_list_item_1,gender_array);
        doc_type=new ArrayAdapter<String>(getApplicationContext(),android.R.layout
                .simple_list_item_1,doc_type_array);
gender.setAdapter(gender_adapter);
        type_doctor.setAdapter(doc_type);
StrictMode.setThreadPolicy(new
StrictMode.ThreadPolicy.Builder().permitNetwork().build());
getData();
        getData1234();
        type_doctor.setOnItemSelectedListener(this);
gender.setOnItemSelectedListener(this);
        reg=(Button)findViewById(R.id.reg);
reg.setOnClickListener(this);
```

```
    }
    private void getData1234()
    {
        try {
           // String find=tid.getText()+"";
            URL url=new URL(address1234);
            HttpURLConnection con=(HttpURLConnection) url.openConnection();
con.setRequestMethod("GET");
            is1234=new BufferedInputStream(con.getInputStream());
        } catch (MalformedURLException e) {
e.printStackTrace();
        } catch (IOException e) {
e.printStackTrace();
        }
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader(is1234));
            StringBuilder sb=new StringBuilder();
while ((line1234=br.readLine())!=null)
            {
sb.append(line1234+"\n");
            }
            is1234.close();
            result1234=sb.toString();
        }
catch(Exception e)
        {
        }
        try
        {
            JSONArray ja=new JSONArray(result1234);
            JSONObject jo=null;
data=new String[ja.length()];
            //    tid.setText(data[0]=jo.getString("id"));
jo=ja.getJSONObject(0);
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));
for(int i=0;i<ja.length();i++)
            {
jo=ja.getJSONObject(i);
date.setText(data[i]=jo.getString("curdate()")+"");
                // tid.setText(data[i]=jo.getString("id")+"");
            }
        }
        catch (Exception e)
        {
        }
    }
private void getData()
    {
        try {
           // String find=tid.getText()+"";
            URL url=new URL(file_path);
            HttpURLConnection con=(HttpURLConnection) url.openConnection();
con.setRequestMethod("GET");
is=new BufferedInputStream(con.getInputStream());
        } catch (MalformedURLException e) {
```

```
e.printStackTrace();
        } catch (IOException e) {
e.printStackTrace();
        }
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader(is));
            StringBuilder sb=new StringBuilder();
while ((line=br.readLine())!=null)
            {
sb.append(line+"\n");
            }
is.close();
result=sb.toString();
        }
catch(Exception e)
        {
        }
        try
        {
            JSONArray ja=new JSONArray(result);
            JSONObject jo=null;
data=new String[ja.length()];
            //    tid.setText(data[0]=jo.getString("id"));
jo=ja.getJSONObject(0);
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));
            //tid.setText(data[0]=jo.getString("id"));
```

```
for(int i=0;i<ja.length();i++)
            {
jo=ja.getJSONObject(i);
                // tid.setText(data[i]=jo.getString("curdate()")+"");
id.setText(data[i]=jo.getString("max(id)+1")+"");
 }
        }
        catch (Exception e)
        {
        }
    }
    @Override
public void onClick(View view)
    {
        if(view==reg)
        {
            String tid=id.getText().toString();
            String tname=name.getText().toString();
            String taddress=address.getText().toString();
            String tcont=cont.getText().toString();
            String tqua=qua.getText().toString();
            String tuser_name=user_name.getText().toString();
            String tpassword=password.getText().toString();
            String temail=email.getText().toString();
            String sgender=gender.getSelectedItem().toString();
            String stype_doctor=type_doctor.getSelectedItem().toString();
            String tdate=date.getText().toString();
if(!isValidateId(tid))
            {
```

```
id.setError("please enter the more then 5 didit ID");
            }
            else
            {
if(!isValidateName(tname))
                {
name.setError("Please Enter Your Name");
                }
                else
                {
if(address.length()==0)
                    {
address.setError("Please Enter Your Address");
                    }
                    else
                    {
if(!isValidatePhone(tcont))
                        {
cont.setError("Please Enter Your Contact Number");
                        }
                        else
                        {
if(qua.length()==0)
                            {
qua.setError("Please Enter Your Qualification");
                            }
                            else
                            {
if(!isValidEmail(temail))
```

```
                        {
email.setError("Please Enter Your Email Address");
                        }
                        else
                        {
if(gender.getSelectedItemPosition()==0)
                           {
t1.setText("Please Select Your Gender");
                           }
                           else
                           {
t1.setText("");
if(type_doctor.getSelectedItemPosition()==0)
                              {
t2.setText("Please Doctor Type");
                              }
                              else
                              {
t2.setText("");
if(user_name.length()<=5)
                                 {
                                    user_name.setError("Please Enter user user name atleast be 5
charater");
                                 }
                                 else
                                 {
if(password.length()<=5)
                                    {
password.setError("please Enter Your Password atleast more than 5 charater");
```

```
                                                }
                                                else
                                                {
t2.setText("");
                                                    String type="doctor_register";
                                                    Doctor_Registration_Form_Back drb=new
                                                            Doctor_Registration_Form_Back(context);
drb.execute(type,tid,tname,taddress,tcont,temail,tqua,sgender,
                                                            stype_doctor,tdate,tuser_name,
                                                            tpassword);
Toast.makeText(this,"Record Added",Toast.LENGTH_LONG).show();
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    // validations function
private boolean isValidEmail(String email)
    {
        String email_pattern="^[A-Za-z0-9._%+\\-]+@[A-Za-z0-9.\\-]+\\.[A-Za-z]{2,4}$";
        Pattern pattern=Pattern.compile(email_pattern);
        Matcher matcher=pattern.matcher(email);
```

```
return matcher.matches();
   }
private boolean isValidateName(String name)
   {
      String name_pattern="[a-zA-z]+([ '-][a-zA-Z]+)*";
      Pattern pattern=Pattern.compile(name_pattern);
      Matcher matcher=pattern.matcher(name);
return matcher.matches();
   }
private boolean isValidatePhone(String phone)
   {
final String phone_pattern="\\d{10}";
      Pattern pattern=Pattern.compile(phone_pattern);
      Matcher matcher=pattern.matcher(phone);
return matcher.matches();
   }
private boolean isValidateId(String id)
   {
final String id_pattern="\\d{6}";
      Pattern pattern=Pattern.compile(id_pattern);
      Matcher matcher=pattern.matcher(id);
return matcher.matches();
   }
   // Doctor name fetch in spinner
   @Override
   public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
   }
   @Override
   public void onNothingSelected(AdapterView<?> adapterView) {
    }
}
```

Regisration_Page_Back_Class.java

```
package com.example.drustant.pca_project;
import android.app.AlertDialog;
import android.content.Context;
import android.os.AsyncTask;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
public class Doctor_Registration_Form_Back extends AsyncTask<String,Void,String>
{
    Context context;
    AlertDialog alertDialog;
    Doctor_Registration_Form_Back(Context ctx)
    {
        context=ctx;
    }
    @Override
protected String doInBackground(String... params)
    {
        String type=params[0];
        //String type="login";
        System.out.println("emulator ip address");
        String login_url="http://10.0.2.2/pca/doctor_registration.php";
        // String login_url="http://192.168.1.101/login.php";
if(type.equals("doctor_register"))
        {
            try {
                String tid=params[1];
                String tname=params[2];
                String taddress=params[3];
                String tcont=params[4];
                String temail=params[5];
                String tqua=params[6];
                String sgender=params[7];
                String stype_doctor=params[8];
                String tdate=params[9];
                String tuser_name=params[10];
   String tpassword=params[11];
                URL url=new URL(login_url);
                HttpURLConnection
httpURLConnection=(HttpURLConnection)url.openConnection();
                httpURLConnection.setRequestMethod("POST");
                httpURLConnection.setDoOutput(true);

                httpURLConnection.setDoInput(true);

                OutputStream outputStream=httpURLConnection.getOutputStream();

                BufferedWriter bufferedWriter=new BufferedWriter(new
OutputStreamWriter(outputStream,"UTF-8"));

                String post_data= URLEncoder.encode("tid","UTF-8")+"="+URLEncoder.encode

                    (tid,"UTF-8")+

                    "&&"+URLEncoder.encode("tname","UTF-8")+"="+URLEncoder.encode

                    (tname,"UTF-8")+

                    "&&"+URLEncoder.encode("taddress","UTF-
8")+"="+URLEncoder.encode(taddress,

                    "UTF-8")+

                    "&&"+URLEncoder.encode("tcont","UTF-8")+"="+URLEncoder.encode

                    (tcont,"UTF-8")+

                    "&&"+URLEncoder.encode("temail","UTF-
8")+"="+URLEncoder.encode(temail,

                    "UTF-8")+

                    "&&"+URLEncoder.encode("tqua","UTF-8")+"="+URLEncoder.encode

                    (tqua,"UTF-8")+

                    "&&"+URLEncoder.encode("sgender","UTF-8")+"="+URLEncoder.encode

                    (sgender,"UTF-8")+

                    "&&"+URLEncoder.encode("stype_doctor","UTF-
8")+"="+URLEncoder.encode

                    (stype_doctor,"UTF-8")+

                    "&&"+URLEncoder.encode("date","UTF-8")+"="+URLEncoder.encode

                    (tdate,"UTF-8")+

                    "&&"+URLEncoder.encode("tuser_name","UTF-
8")+"="+URLEncoder.encode

                    (tuser_name,

                        "UTF-8")+

                    "&&"+URLEncoder.encode("tpassword","UTF-8")+"="+URLEncoder.encode
```

```
                        (tpassword,
                                "UTF-8");
                bufferedWriter.write(post_data);
bufferedWriter.flush();;
                bufferedWriter.close();
                outputStream.close();
                InputStream inputStream=httpURLConnection.getInputStream();
                BufferedReader bufferedReader=new BufferedReader(new
InputStreamReader(inputStream,"iso-8859-1"));
                String result="";
                String line="";
while((line=bufferedReader.readLine())!=null)
                {
                    result+=line;
                }
                bufferedReader.close();
                inputStream.close();
                httpURLConnection.disconnect();
                return result;
            } catch (MalformedURLException e) {
e.printStackTrace();
            } catch (IOException e) {
e.printStackTrace();
            }
        }
        return null;
    }
    @Override
protected void onPreExecute() {
        // super.onPreExecute();
alertDialog=new AlertDialog.Builder(context).create();
        alertDialog.setTitle("Login Status........");
    }
    @Override
protected void onPostExecute(String result) {
super.onPostExecute(result);
        //alertDialog=new AlertDialog.Builder(context).create();
        // alertDialog.setTitle("Login Status..");
        alertDialog.setMessage(result);
        alertDialog.show();
    }
    @Override
protected void onProgressUpdate(Void... values) {
super.onProgressUpdate(values);
    }
}
```

CHAPTER – V

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

Following are hardware & software requirements to run this software

Minimum Software Requirements:-

- Android Studio
- Wamp server/ xmap server

Minimum Hardware Requirements:

- RAM : 8 GB RAM
- Processor : Dual Core
- Hard Disk : 1 GB

CHAPTER – VI

FUTURE ENHANCEMENT

FUTURE ENHANCEMENT

- The smart card replace by thumb scanner.
- Doctor registration will be more accurate.
- Dynamic Prescription

CHAPTER - VII

REFERENCES AND BIBLIOGRAPHY

REFERENCES AND BIBLIOGRAPHY

- www.google.com
- www.youtube.com
- www.w3school.com
- www.javatpoint.com