

A PROJECT REPORT  
ON  
**WINDOWS APPLICATION ABOUT PATIENT FOLLOWUP FORM  
(DOCPRO)**

SUBMITTED BY

**Mr. SALIL RAVINDRA BHAGAT**

IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE

OF

**BACHELOR OF SCIENCE**

IN

**COMPUTER SCIENCE**

UNDER THE GUIDANCE OF

**PROF. VIPUL SALUJA**

**DEPARTMENT OF COMPUTER SCIENCE**



**R. D. & S.H. National College of Arts and Commerce**

**& W.A. Science College , Bandra, Mumbai - 400050.**

**(Sem-V)**

**(2019- 2020)**



# R.D. & S.H. NATIONAL COLLEGE & S. W.A. SCIENCE COLLEGE,



Bandra, Mumbai - 400050.

Department of Computer Science

## PROJECT CERTIFICATE

This is to certify that **Mr. SALIL RAVINDRA BHAGAT** of **T.Y.B.Sc. CS (Sem V)** class has satisfactorily completed the Project entitled **WINDOWS APPLICATION ABOUT PATIENT FOLLOWUP FORM (DOCPRO)**, to be submitted in the partial fulfillment for the award of **Bachelor of Science in Computer Science (Semester V)** during the academic year **2019 - 2020**.

It is further certified that the project is an original work and it has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

**Date of Submission:**

**Internal Project Guide**

**Co-ordinator,**

**Department of Computer Science**

**Signature of External Examiner**

### **DECLARATION**

I, Mr. SALIL BHAGAT, hereby declare that the project entitled “**WINDOWS APPLICATION ABOUT PATIENT FOLOWUP FORM (DOCPRO)**” submitted in the partial fulfilment for the award of **Bachelor of Science in Computer Science** during the academics year 2019-2020 is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles .

**Signature of the student:**

**Place:**

**Date:**

## ACKNOWLEDGEMENT

This project could not have been completed without **Prof. Vipul Saluja** who not only served as my project guide but also encouraged and challenged me throughout my academic program. He guided me through the entire project process, never accepting less than my best efforts. I thank them all.

I would also like to thank my family and my brother , who helped and supported me throughout the way. And a great contribution of internet, specifically specifiedcoding, youtube, google, android hub and so on.

## INDEX

### **CHAPTER 1: PRELIMINARY INVESTIGATION**

- 1.1 Synopsis
- 1.2 Organizational Overview
- 1.3 Working of the current system
- 1.4 Limitations of the current system
- 1.5 The Proposed System
- 1.6 Advantages of the current system
- 1.7 Tools and Technologies to be used
- 1.8 Feasibility study

### **CHAPTER 2: SYSTEM ANALYSIS**

- 2.1 Event Table
- 2.2 Entity Relationship Diagram
- 2.3 Class Diagram
- 2.4 Object Diagram
- 2.5 Use Case Diagram
- 2.6 Activity Diagram
- 2.7 State Chart Diagram
- 2.8 Sequence Diagram
- 2.9 Collaboration Diagram

### **CHAPTER 3: SYSTEM DESIGN**

- 3.1 Component Diagram
- 3.2 System Flow Chart
- 3.3 Structured Chart
- 3.4 Deployment Diagram

### **CHAPTER 4: SYSTEM CODING**

- 4.1 Data Dictionary
- 4.2 Source Code

### **CHAPTER 5: SYSTEM IMPLEMENTATION**

- 5.1 Hardware and Software Requirements
- 5.2 Screen Layouts

### **CHAPTER 6: FUTURE ENHANCEMENTS**

### **CHAPTER 7: REFERENCES AND BIBLIOGRAPHY**

## **CHP 1-PRELIMINARY INVESTIGATION**

1. Synopsis
2. Organization Overview
3. Working of the current system
4. Limitations of the current system
5. The Proposed System
6. The Advantages of the Proposed System
7. Tools and Technologies to be used
8. Feasibility Study

## **1. PRELIMINARY INVESTIGATION**

### **1.1: SYNOPSIS**

#### **PATIENT FOLLOW-UP FORM**

##### **INTRODUCTION:-**

- **TILL THIS DATE DOCTORS USE PAPERS AND BOOKS FOR MAINTAINING THE RECORD OF THE PATIENTS.**
- **I WILL BE DEVELOPING THE THE WINDOWS APP NAMED “DOCPRO”**
- **IT INCLUDES THE FORM WHERE THE DOCTOR WRITES THE INFORMATION ABOUT PATIENT AND IT IS STORED**
- **THE INFORMATION CAN BE STORED IN THE FORM OF TABLE**
- **DAILY RECORD OF PATIENTS IS ALSO AVAILABLE**

##### **ADVANTAGES:-**

- **DOCTORS CAN ACCESS THE PATIENTS RECORDS ANYWHERE WITH THE HELP OF THIS APP**
- **IT CAN PREVENT THE LOSS OF INFORMATION**
- **IT CAN REDUCE THE DEPENDENCY OF PAPERS**

##### **MODULES:-**

- **HISTORY**
- **CHIEF COMPLAINT**
- **SYMPTOMS**
- **GENERAL SENSE OF WELL BIENG**
- **POSSIBILITIES**
- **NAMES OF DOCORS,CLINIC,ETC.**
- **PATIENT INFORMATION**
- **TREATMENT**
- **ACTION TO TAKE**
- **REMARKS**

## **1.2 Organizational Overview:**

Few reasons why 'DOCPRO' is loved by many:

- 1) **Quality** – This app is used to manage the patient records. User interface is priority
- 2) **Green** : to give less stress on environment by eliminating paper use
- 3) **Security** : Information is secured and cannot be deleted unless by user

## **1.3 Working of the current system :**

- 1) The user needs to enter the patient data in system
- 2) When the user log-in he get to choose weather to store,delete or view the data
- 3) To store the data form button is to be used, to delete the record delete button is to be used and to view the record view the record is to be used
- 4) Thus these functions are present in this software

## **1.4 Limitations of the current systems**

- 1) You need to adjust the table while viewing the record
- 2) Cant do backup yet
- 3) Cant transform record in readable file yet
- 4) Some minor bugs still present

## **1.5 The Proposed Systems**

- 1 This windows app is the example of softwares engaging in different fields
- 2 This app is best to save space,time,manpower,etc
- 3 The application makes the whole process of managing the patients records easier
- 4 This app as good user interface

## **1.6 Advantages of the Proposed system:**

- 1) User can access the records of the patient anytime with the help of this app
- 2) It can prevent loss of information
- 3) It can reduce dependency on papers

## **1.7 Tools and Technologies to be used:**

- 1) MYSQL
- 2) NETBEANS



## MySql

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

MySQL is a central component of the LAMP open-source web application software stack (and other "AMP" stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl /PHP/ Python". Applications that use the MySQL database include: TYPO3, MODx, Joomla, WordPress, Simple Machines Forum, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

## NETBEANS

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called *modules*. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5,<sup>[4]</sup> and JavaScript. Applications based on NetBeans, including the NetBeans IDE, can be extended by third party developers

NetBeans began in 1996 as Xelfi (word play on *Delphi*) a Java IDE student project under the guidance of the Faculty of Mathematics and Physics at Charles University in Prague. In 1997, Roman Staněk formed a company around the project and produced commercial versions of the NetBeans IDE until it was bought by Sun Microsystems in 1999. Sun open-sourced the NetBeans IDE in June of the following year. Since then, the NetBeans community has continued to grow/ In 2010, Sun (and thus NetBeans) was acquired by Oracle Corporation. Under Oracle, NetBeans competed with JDeveloper, a freeware IDE that has historically been a product of the company. In September 2016, Oracle submitted a proposal to donate the NetBeans project to the Apache Software Foundation, stating that it was "opening up the NetBeans governance model to give NetBeans constituents a greater voice in the project's direction and future success through the upcoming release of Java 9 and NetBeans 9 and beyond". The move was endorsed by Java creator James Gosling.<sup>[9]</sup> The project entered the Apache Incubator in October 2016.

## 1.8 Feasibility study:

### **Introduction:**

- A feasibility study is an analysis of how successfully a project can be completed.
- It is the initial design stage of any project, which brings together the elements of knowledge.

### **Technical feasibility:-**

DOCPRO is an windows application which developed under netbeans between And MySQL databases for database connectivity And for data storage access

The use of the MySQL was preferred because of its simplicity

Technical feasibility is concerned with specifying the equipments and the software to satisfy the user requirements.

### **Operational feasibility:-**

The Application has been developed for windows operating system and will work on all the windows Device . The database which is used for this application is been hosted using MySQL

The overall response of the system will also increase as there will be more number of users affiliated with the system in the near future.

### **Economic feasibility:-**

The Application is been developed for the Third Year College Project.

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system.

The proposed system can be developed at a minimum cost and resource.

The system can assure a good beneficial cost to the organisation.

The savings that would arise from the beneficial cost of the system can be used to improve the system's performance in future.

## CHP 2-SYSTEM ANALYSIS

- 2.1 Event Table
- 2.2 Entity Relationship Diagram
- 2.3 Class Diagram
- 2.4 Object Diagram
- 2.5 Use Case Diagram
- 2.6 Activity Diagram
- 2.7 State Chart Diagram
- 2.8 Sequence Diagram
- 2.9 Collaboration Diagram

### 2.1 Event Table

Events are objects or messages used when a software components wants to notify a state change to other components.

An Event model is a software architecture (a set of classes and interfaces)that determines how components occur.

On the event source side:-

- create and describe events
- trigger (or fire)events
- distribute events to interested components

On the event listener side:-

- subscribe to event sources
- react to events when received
- remove the subscription to event sources when desired

Terminology often used refers to:-

- Event Source or Provider :-the sender of events
- Event :-the object sent
- Event Listener or Event Sink or Consumer :-the receiver of events

### FOR REGISTERED USER :

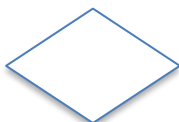
EVENT	SOURCE	TRIGGER	ACTIVITY	DESTINATION
Login	Registered User	Login	Display Login Page	Registered User
Visits the choose option	Registered User	Choose	Display page	Registered User
Sign Up	Registered User	Registration	Display Registration page	Registered User
View data	Registered User	Viewing of data	Display view record page	Registered User
Delete records	Registered User	Delete records	Display delete record page	Registered User
Insert records	Registered User	Insert records	Insert records page	Registered User
Logs Out	Registered User	Log Out	Display Login Page	Registered User

## 2.2 Entity Relationship Diagram

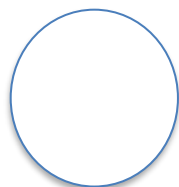
An entity- relationship model is a systematic way of describing and defining a business process. The process is modelled as components (entites) that are linked wid each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. Entites may have various properties (attributes) that characterize them. Diagram created to represent these entities, attributes and relationships graphically are called entity -relationship diagrams.



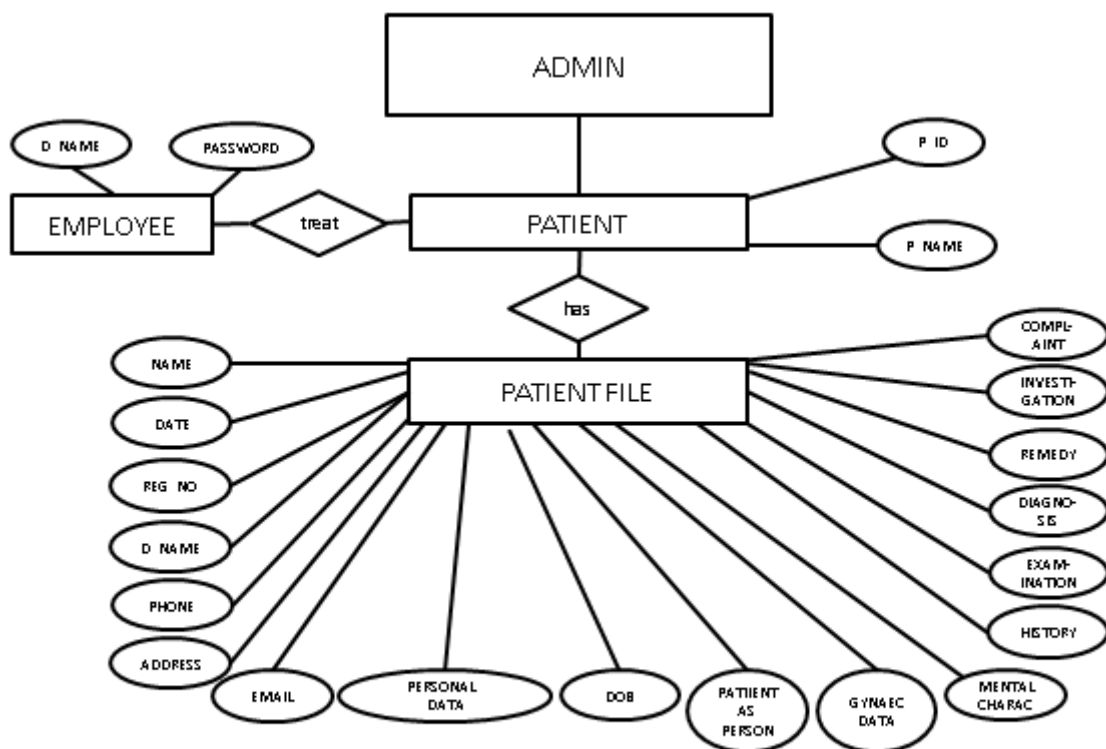
**Entity**



**Relationship**



**Attribute**



## 2.3 Class Diagram

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualising, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a structural diagram.

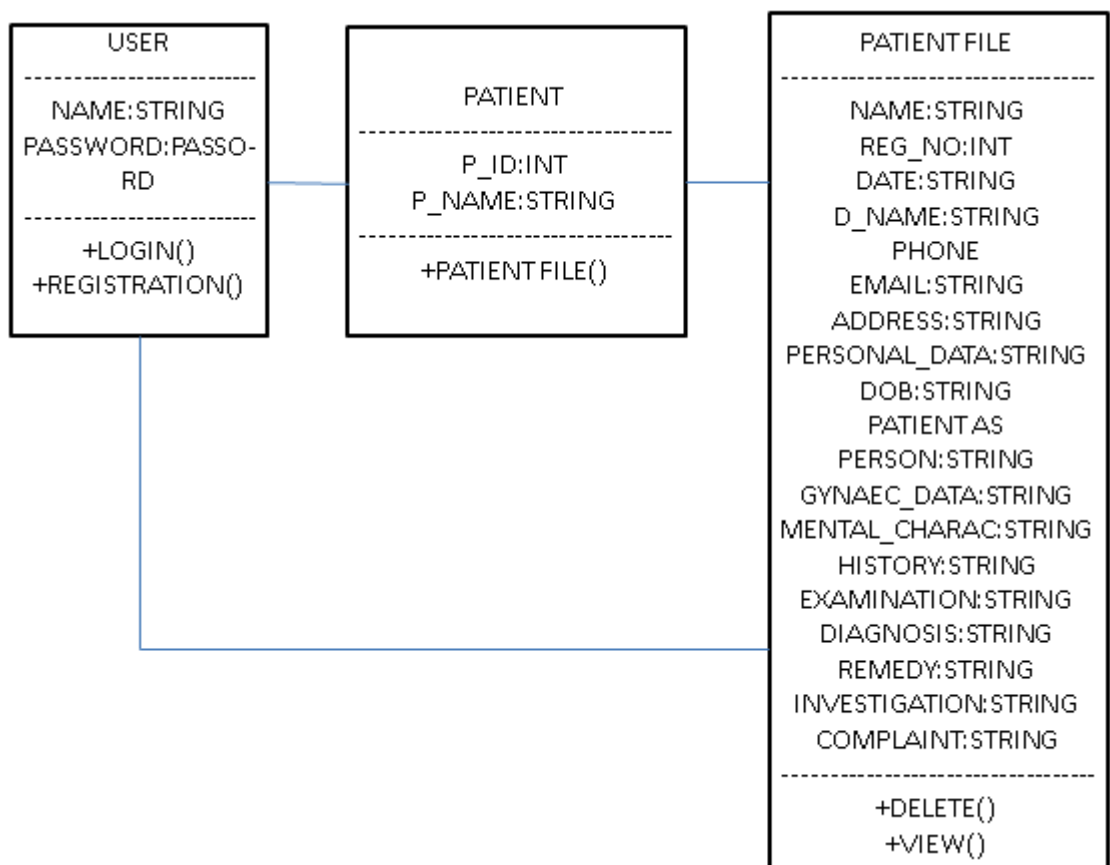
### Purpose:-

The purpose of the class diagram is to model the static view of an application. The class diagrams are the only diagrams which can be directly mapped with object oriented languages and thus widely used at the time of construction.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

So the purpose of the class diagram can be summarised as :-

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.



## 2.4 Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.

Object diagrams represent an instance of class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.

Object diagrams are used to render a set of objects and their relationships as an instance.

### Purpose:-

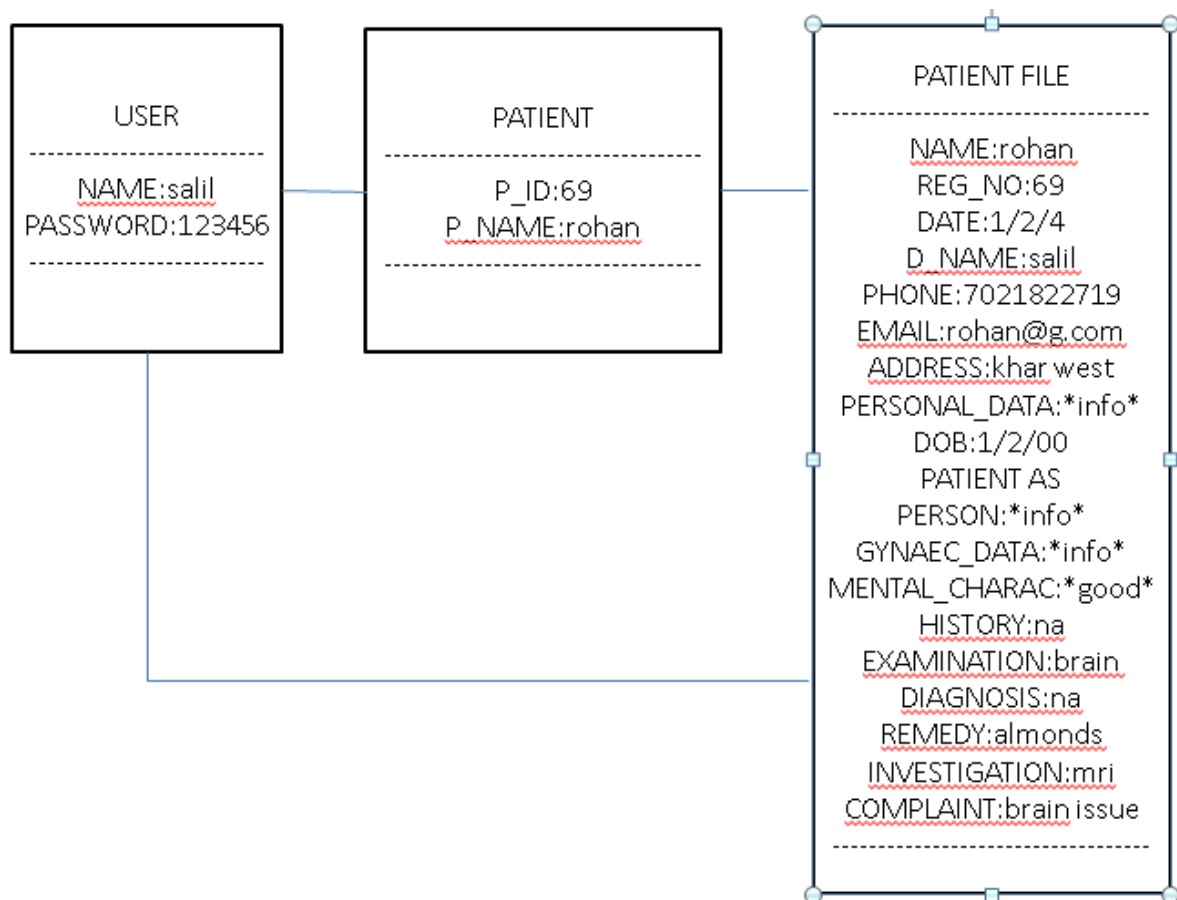
The purpose of a diagram should be understood clearly to implement it practically. The purpose of object diagrams are similar to class diagrams.

The difference is that a class diagram represents an abstract model consisting of classes and their relationships. But an object diagram represents an instance at a particular moment which is concrete in nature.

It means the object diagram is more close to the actual system behaviour. The purpose is to capture the static view of a system at a particular moment.

So the purpose of the object diagram can be summarised as:-

- Forward and reverse engineering
- Object relationships of a system
- Static view of an interaction
- Understand object behaviour and their relationship from practical perspective.



## 2.5 Use Case Diagram

To model a system the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.

So to model the entire system numbers of use case diagrams are used.

### **Purpose:-**

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose.

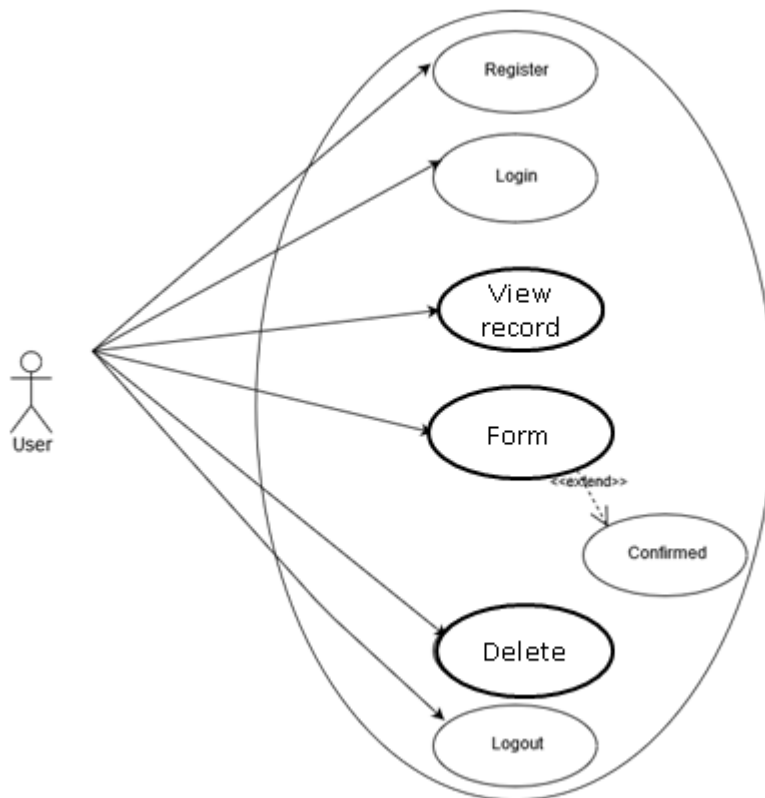
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So in brief the purpose of use case diagram can be as follows:-

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interaction among the requirements and actors





## 2.6 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system.

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagram deals with all type of flow control by using different elements like fork, join etc.

### Purpose:

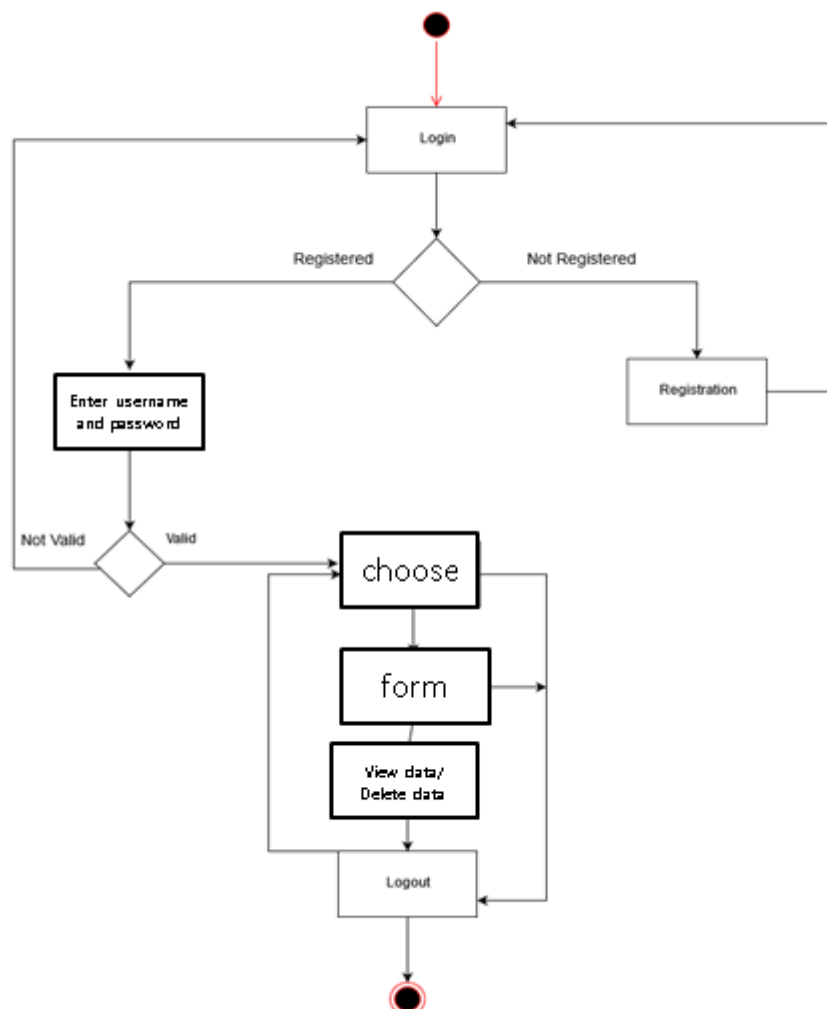
It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagram looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

So the purpose can be described as:

- Draw the activity flow of system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.



## 2.7 State Chart Diagram

The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.

A State Chart Diagram describes a state machine. Now to clarify it state machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

As State Chart Diagram defines states it is used to model lifetime of an object.

### **Purpose:-**

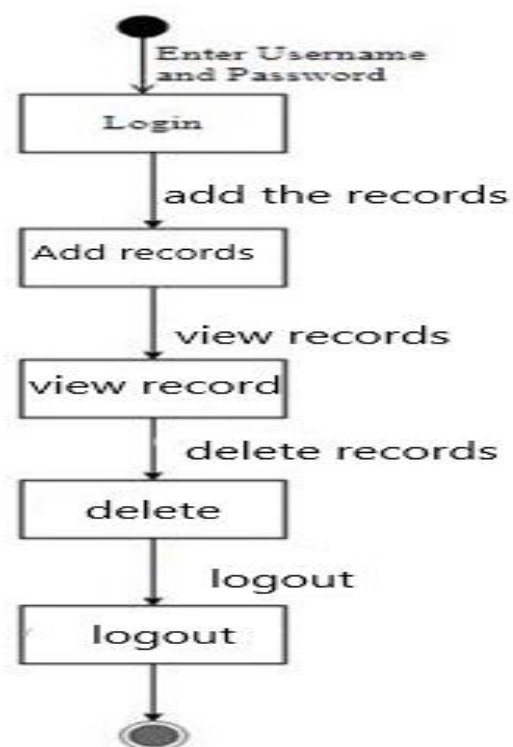
State Chart Diagram is one of the five UML diagrams used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So State Chart Diagrams are useful to model reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

State Chart Diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. So the most important purpose of State Chart Diagram is to model life time of an object from creation to termination.

State Chart Diagram are also used for forward and reverse engineering of a system. But the main purpose is to model reactive system.

Following are the main purpose of using State Chart Diagrams:-

- To model dynamic aspect of a system.
- To model life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model of an object.

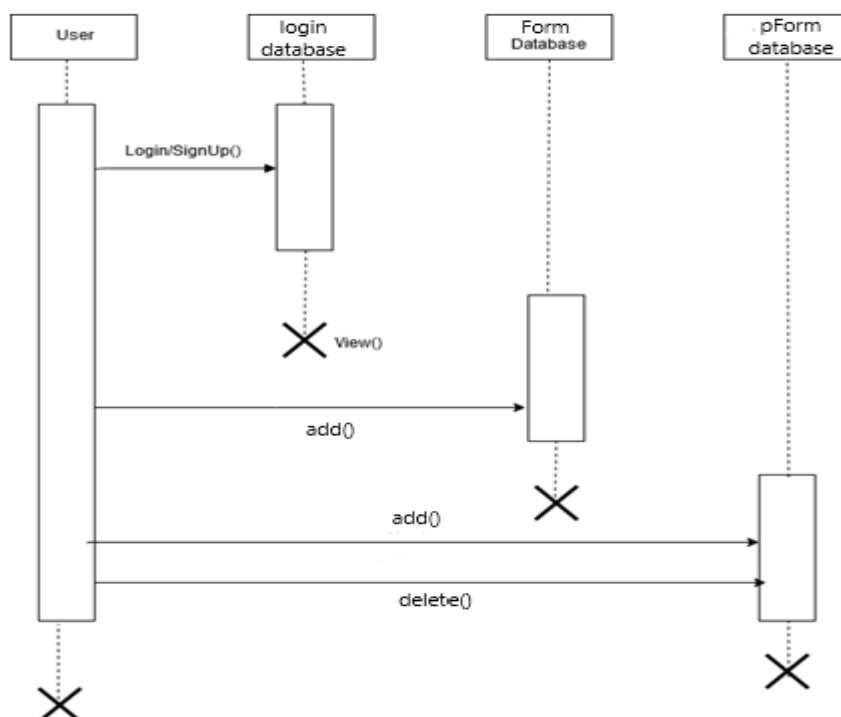


## 2.8 Sequence Diagram

A Sequence Diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A Sequence Diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence Diagrams are sometimes called event diagrams or event scenarios.

A Sequence Diagram shows parallel vertical lines(lifelines),different processes or objects that live simulatenously and as horizontal arrows,the messgaes exchanged between them,in order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

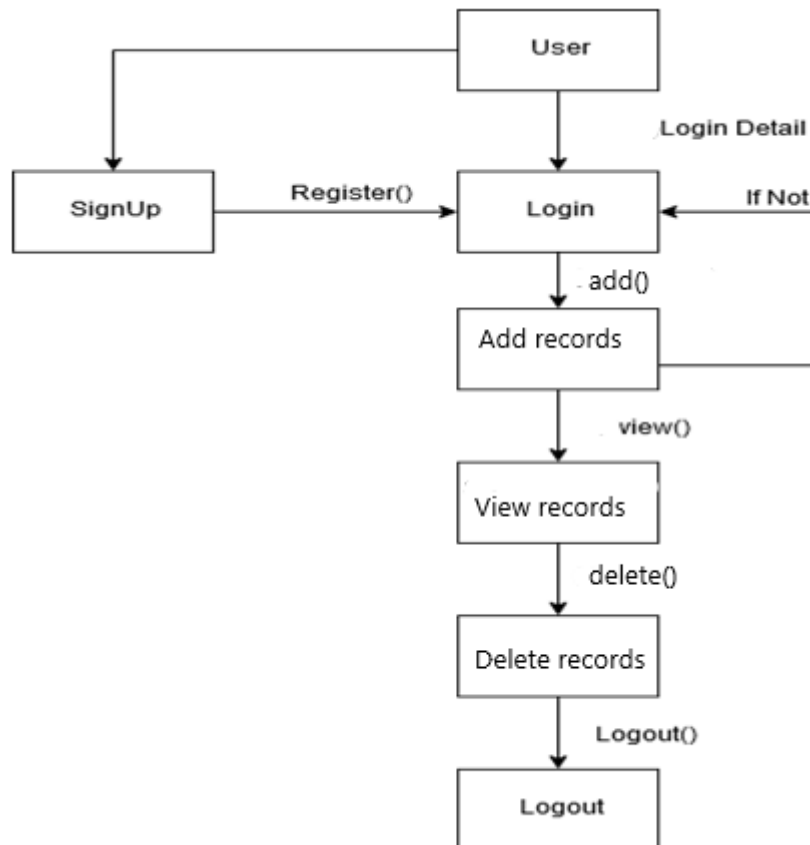
If the lifeline is that of an object,it demonstrates a role.Leaving the instance name blank can represent anonymous and unnamed instances.Messages,written with horizontal arrows with the message name written above them,display interaction.Solid arrow heads represent synchronous calls,open arrow heads represent asynchronous messages, and dashed lines represent reply messages.If a caller sends a synchronous message,it must wait until the message is done,such as invoking a subroutine. If a caller sends an asynchronous message,it can continue processing and doesn't have to wait for a response.Asynchronous calls are present in multithreaded applications and in message-oriented middleware.Activation boxes,or method-call boxes,ar opaque rectangles drawn on top of lifelines to represent that processes are being performed response to the message(Execution Specification in UML)



## 2.9 Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the UML.

Collaboration diagrams are best suited to the portrayal of simple interactions among relatively small numbers of objects. As the number of objects and messages grows, a collaboration diagram can become difficult to read.



## CHP 3 :- System Diagram

3.1 Component Diagram

3.2 System Flow Chart

3.3 Structured Chart

3.4 Deployment Diagram

3.5 Package Diagram

### 3.1 Component Diagram

Component Diagram are different in terms of nature and behaviour. Component diagrams are used to model physical aspects of a system.

Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

#### **Purpose:-**

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the component used to make those functionalities

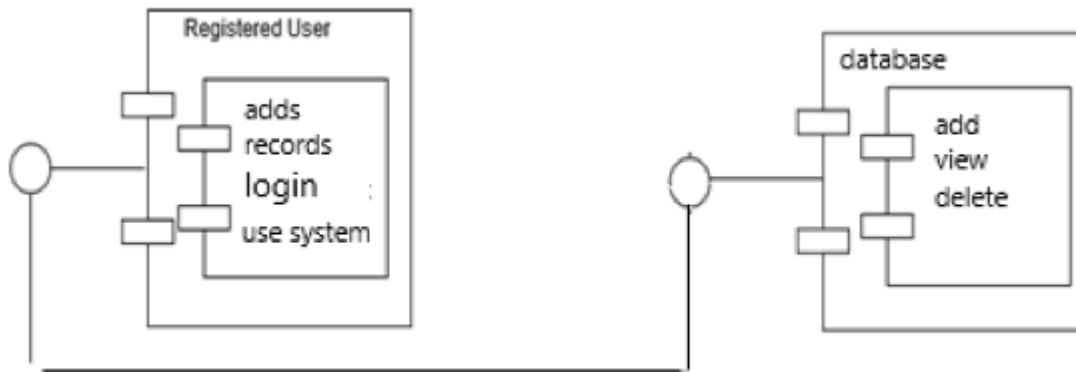
So from that point component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of system. Static implementation represents the organization of the components at a particular moment.

A single Component Diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarized as:

- Visualize the components of a system
- Construct executables by using forward and reverse engineering
- Describe the organization and relationships of the components.



### 3.2 System Flow Chart Diagram

- System flow charts are a way of displaying how data flows in a system and how decisions are made to control events.

- To illustrate this, symbols are used. They are connected together to show what happens to data and where it goes. The basic ones include: symbols used in flow charts.

- The flow of data generally goes from top to bottom and left to right and depicts the sequence of processing steps along these data lines.

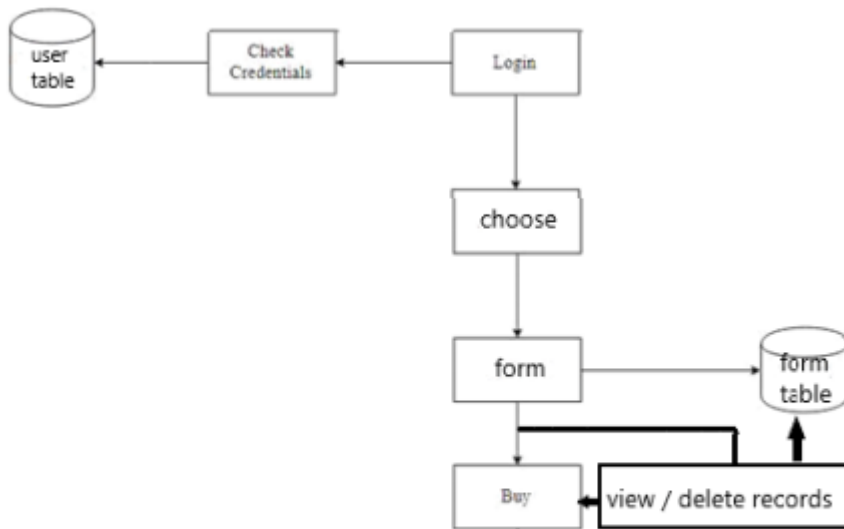
- The following are examples of some of the symbols used in systems flowcharts:

A systems flowchart shows the key input and outputs associated with the program. The shape of the symbols indicates the types of input or output devices.

- The type of diagram dictates the flow chart symbols that are used. The terminator symbols marks the starting or ending point of the system.

- A flow chart is a formalized graphic representation of a program logic sequence, work or manufacturing process, organization

- A graphical representation of the sequence of operations in an information system or program. Information system flow charts show how data flows from source.



### 3.3 Structure Chart Diagram

A structure chart (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels.

They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contain the module's name. A structure chart (SC) in software engineering and organizational theory, is a chart which shows the breakdown of a system to its lowest manageable levels.

They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

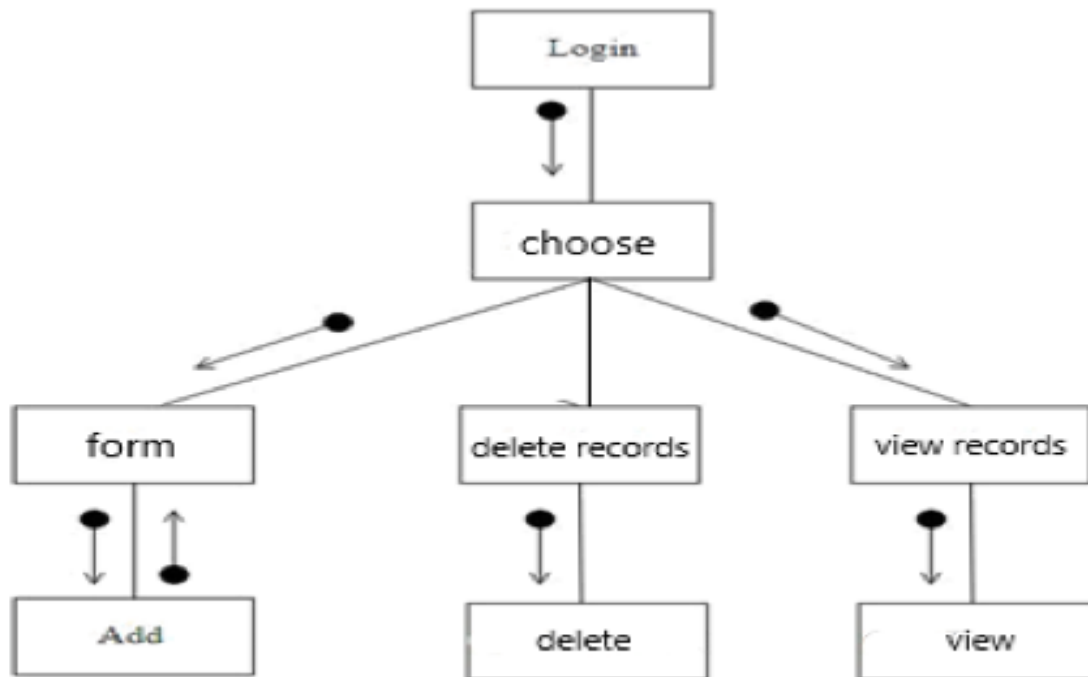
Structure diagram is a chart derived from data flow diagram. The system structure chart represents hierarchical structure of modules.

A structure chart depicts

The size and complexity of the system, and number of a readily identifiable functions and modules within each function and whether each identifiable function is a manageable entity or should be broken down into smaller components.

A structure chart is also used to diagram associated elements that comprise a run stream or thread. It is often developed as a hierarchal diagram, but other representations are allowable.





### 3.4 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

#### **Purpose:-**

The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.

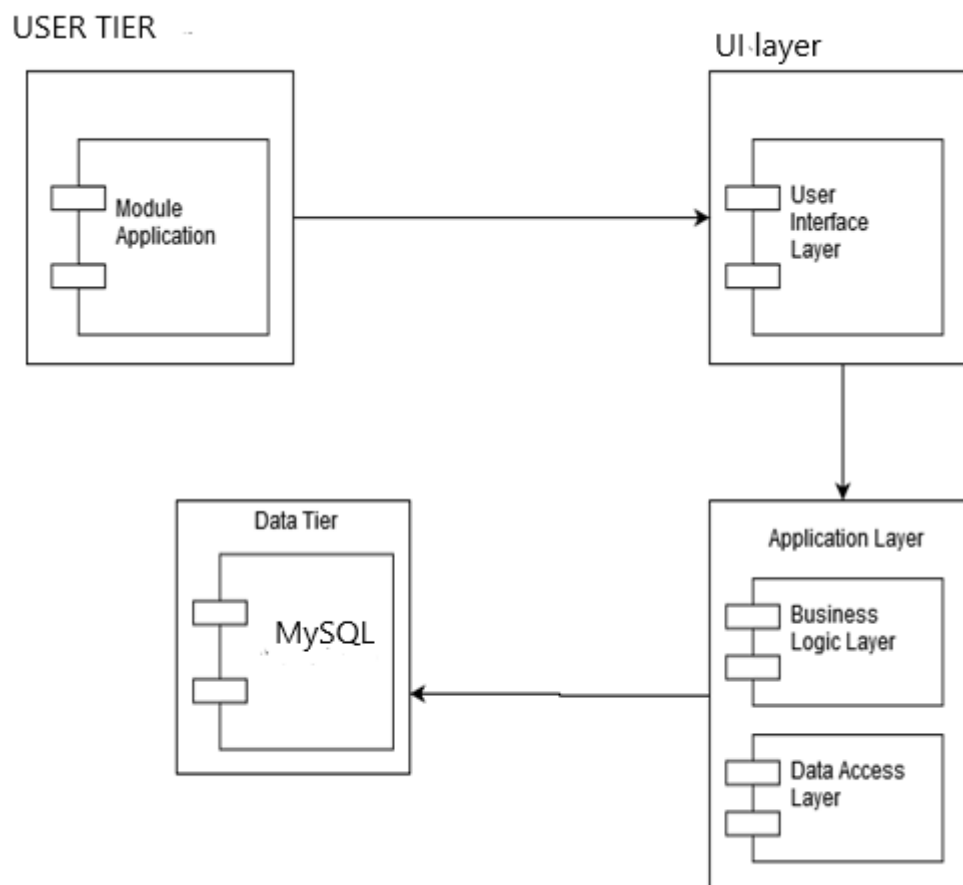
Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So most of the UML diagrams are used to handle logical components but deployment are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

It can be described as follows:-

- Visualize hardware topology of a system.
- Describe the hardware components used to deploy software components.
- Describe runtime processing nodes



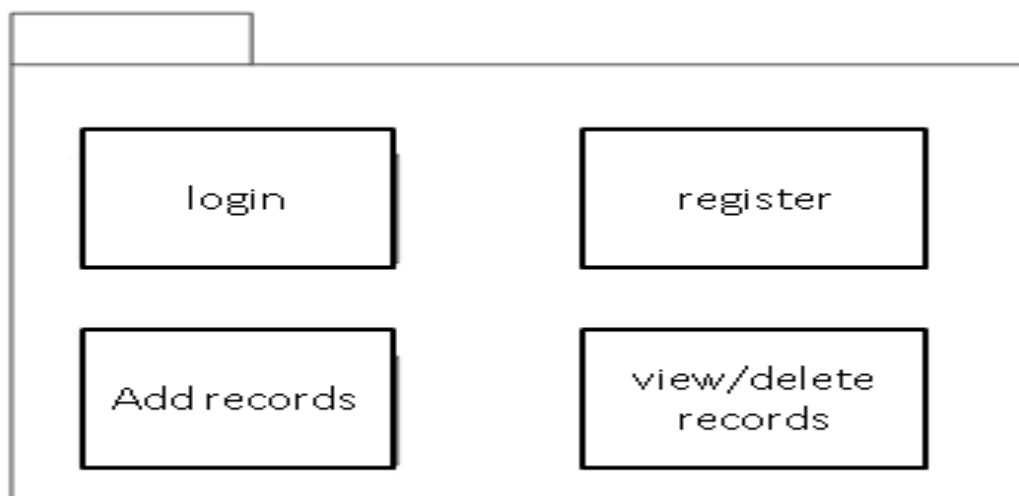
### 3.5 Package Diagram

1) When modeling a large scale system, you would probably be working with a high volume of model elements. They describe a model from different views and different phases, hence are in different types.

2) UML package helps to organize and arrange model elements and diagrams into logical groups, through which you can manage a chunk of project data together.

3) You can also use packages to represent different views of the systems architecture .In addition, developers can use package to model the physical package or namespace structure of the application to build.

4) Package Diagram visualizes packages and depicts the dependency, Import, access, generalization, realization and merge relationships between them. Package diagram enables you to gain a high level understanding of the collaboration among model elements through analyzing the relationships among their parent package. This also helps explain the systems architecture from a broad view.



## CHAPTER 4: SYSTEM CODING

4.1 Data Dictionary

4.2 Source Code

### 4.1 DATA DICTIONARY

# 1) PFORM TABLE

Field	Type	Null	Key	Default	Extra
regno	int(11)	YES	UNI	NULL	
docname	varchar(255)	YES		NULL	
date	varchar(255)	YES		NULL	
patientname	varchar(255)	YES		NULL	
phoneno	varchar(255)	YES		NULL	
address	varchar(255)	YES		NULL	
email	varchar(255)	YES		NULL	
age	varchar(255)	YES		NULL	
sex	varchar(255)	YES		NULL	
dob	varchar(255)	YES		NULL	
occupation	varchar(255)	YES		NULL	
marry	varchar(255)	YES		NULL	
religion	varchar(255)	YES		NULL	
complaint	varchar(255)	YES		NULL	
diet	varchar(255)	YES		NULL	
habit	varchar(255)	YES		NULL	
appetite	varchar(255)	YES		NULL	
likes	varchar(255)	YES		NULL	
dislikes	varchar(255)	YES		NULL	
thirst	varchar(255)	YES		NULL	
stool	varchar(255)	YES		NULL	
urine	varchar(255)	YES		NULL	
sweat	varchar(255)	YES		NULL	
sleep	varchar(255)	YES		NULL	
dreams	varchar(255)	YES		NULL	
gynac	varchar(255)	YES		NULL	
childdev	varchar(255)	YES		NULL	
mental	varchar(255)	YES		NULL	
family	varchar(255)	YES		NULL	
parent	varchar(255)	YES		NULL	
examination	varchar(255)	YES		NULL	
investigation	varchar(255)	YES		NULL	
diagno	varchar(255)	YES		NULL	
remedy	varchar(225)	YES		NULL	

## RFORM

Field	Type	Null	Key	Default	Extra
regno	int(11)	YES		NULL	
docname	varchar(255)	YES		NULL	
date	varchar(255)	YES		NULL	
patientname	varchar(255)	YES		NULL	
age	varchar(255)	YES		NULL	
sex	varchar(255)	YES		NULL	
diagno	varchar(255)	YES		NULL	
remedy	varchar(225)	YES		NULL	

## LOGIN

Field	Type	Null	Key	Default	Extra
username	varchar(225)	YES		NULL	
password	varchar(255)	YES		NULL	

### 4.3 Source code

**NewJFrameload:** welcome.java page

```
public static void main(String[] args) {  
    NewJFrameload n = new NewJFrameload();  
    n.setVisible(true);  
    NewJFramestart a = new NewJFramestart();  
    try{  
        for(int x = 0; x <=100; x++)  
        {  
            Thread.sleep(10);  
            NewJFrameload.per.setText(Integer.toString(x)+"%");  
            NewJFrameload.pro.setValue(x);  
            if(x==100){  
                n.setVisible(false);  
                a.setVisible(true);  
            }  
        }  
    }catch (Exception e){  
    }  
}
```

### NewJFramestart.java: TO LOGIN USER

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jTextField1.setText("");  
    jPasswordField1.setText("");  
}  
  
private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {  
    conn=MySqlConnection.ConnectDB();  
    String Sql="Select * from login where username=? and password=?";  
    try{  
        pst=conn.prepareStatement(Sql);  
        pst.setString(1,jTextField1.getText());  
        pst.setString(2,jPasswordField1.getText());  
        rs=pst.executeQuery();  
        if(rs.next()){  
            JOptionPane.showMessageDialog(null,"LOGIN SUCCESSFULL");  
            new NewJFramechoose().setVisible(true);  
            this.setVisible(false);  
        }  
        else{  
            JOptionPane.showMessageDialog(null,"Invalid      username      or  
password","Access Denied",JOptionPane.ERROR_MESSAGE);  
        }  
    }catch(Exception e){
```

```

        JOptionPane.showMessageDialog(null, e);
    }
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    new NewJFrameregis().setVisible(true);
    this.setVisible(false);
}

```

### **NewJFrameregis.java: TO REGISTER USER**

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
        Class.forName("com.mysql.jdbc.Driver");

        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");

        String sql = "insert into login values (?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, username.getText());
        pstmt.setString(2, password.getText());
        pstmt.executeUpdate();

        JOptionPane.showMessageDialog(null, "success");
        conn.close();
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}

```

### **NewJFrameform.java: TO ENTER PATIENT DATA**

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        Class.forName("com.mysql.jdbc.Driver");  
  
        Connection conn =  
DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");  
  
        String sql = "insert into pform values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,  
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";  
  
        PreparedStatement pstmt = conn.prepareStatement(sql);  
  
        pstmt.setInt(1, Integer.parseInt(regno.getText()));  
  
        pstmt.setString(2, docname.getText());  
  
        pstmt.setString(3, date.getText());  
  
        pstmt.setString(4, patientname.getText());  
  
        pstmt.setString(5, phoneno.getText());  
  
        pstmt.setString(6, address.getText());  
  
        pstmt.setString(7, email.getText());  
  
        pstmt.setString(8, age.getText());  
  
        pstmt.setString(9, sex.getText());  
  
        pstmt.setString(10, dob.getText());  
  
        pstmt.setString(11, occupation.getText());  
  
        pstmt.setString(12, marry.getText());  
  
        pstmt.setString(13, religion.getText());  
  
        pstmt.setString(14, complaint.getText());  
  
        pstmt.setString(15, diet.getText());  
  
        pstmt.setString(16, habit.getText());  
  
        pstmt.setString(17, appetite.getText());  
  
        pstmt.setString(18, likes.getText());
```



```
pstmt.setString(19, dislikes.getText());  
pstmt.setString(20, thirst.getText());  
pstmt.setString(21, stool.getText());  
pstmt.setString(22, urine.getText());  
pstmt.setString(23, sweat.getText());  
pstmt.setString(24, sleep.getText());  
pstmt.setString(25, dreams.getText());  
pstmt.setString(26, gynac.getText());  
pstmt.setString(27, childdev.getText());  
pstmt.setString(28, mental.getText());  
pstmt.setString(29, family.getText());  
pstmt.setString(30, parent.getText());  
pstmt.setString(31, examination.getText());  
pstmt.setString(32, investigation.getText());  
pstmt.setString(33, diagno.getText());  
pstmt.setString(34, remedy.getText());  
pstmt.executeUpdate();  
  
JOptionPane.showMessageDialog(null, "success");  
  
conn.close();  
}catch(Exception e)  
{  
    JOptionPane.showMessageDialog(null, e);  
}  
  
try{  
    Class.forName("com.mysql.jdbc.Driver");
```

```

        Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");

        String pql = "insert into rform values (?, ?, ?, ?, ?, ?, ?, ?)";

        PreparedStatement dstmt = con.prepareStatement(pql);

        dstmt.setInt(1, Integer.parseInt(regno.getText()));

        dstmt.setString(2, docname.getText());

        dstmt.setString(3, date.getText());

        dstmt.setString(4, patientname.getText());

        dstmt.setString(5, age.getText());

        dstmt.setString(6, sex.getText());

        dstmt.setString(7, diagno.getText());

        dstmt.setString(8, remedy.getText());

        dstmt.executeUpdate();

        JOptionPane.showMessageDialog(null, "success");

        con.close();

    }catch(Exception e)

    {

        JOptionPane.showMessageDialog(null, e);

    }

    new NewJFrameform().setVisible(true);

    this.setVisible(false);

}

```

### **NewJFramereg.java: TO DISPLAY TABLE**

```

public NewJFramereg() {

    initComponents();

    DisplayTable();
}

```

```

    }

    private void DisplayTable()
    {
        try{
            Class.forName("com.mysql.jdbc.Driver");

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");

            String sql = "select * from rform";

            PreparedStatement pstmt = conn.prepareStatement(sql);

            ResultSet rs = pstmt.executeQuery();

            t1.setModel(DbUtils.resultSetToTableModel(rs));

        }

        catch(Exception e){

            JOptionPane.showMessageDialog(null, e);

        }
    }

```

---

## TO DELETE DATA

```

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        try{

            Class.forName("com.mysql.jdbc.Driver");

            Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");

            String sql = "delete from rform where regno = ?";

            PreparedStatement pstmt = conn.prepareStatement(sql);

            pstmt.setInt(1, Integer.parseInt(dbar.getText()));

            pstmt.executeUpdate();

            JOptionPane.showMessageDialog(null, "record deleted");

        }
    }

```

```

        new NewJFramereg().setVisible(true);

        this.setVisible(false);

        conn.close();

    }

    catch(Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

    try{

        Class.forName("com.mysql.jdbc.Driver");

        Connection conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/form","root","root");

        String sql = "delete from pform where regno = ?";

        PreparedStatement pstmt = conn.prepareStatement(sql);

        pstmt.setInt(1, Integer.parseInt(dbar.getText()));

        pstmt.executeUpdate();

        new NewJFramereg().setVisible(true);

        this.setVisible(false);

        conn.close();

    }

    catch(Exception e){

        JOptionPane.showMessageDialog(null, e);

    }

}

```

### **NewJFrameview.java : TO VIEW DATA**

```

private void t1MouseClicked(java.awt.event.MouseEvent evt) {

    int i = t1.getSelectedRow();

```

```
TableModel model = t1.getModel();

regno.setText(model.getValueAt(i,0).toString());

docname.setText(model.getValueAt(i,1).toString());

date.setText(model.getValueAt(i,2).toString());

patientname.setText(model.getValueAt(i,3).toString());

phoneno.setText(model.getValueAt(i,4).toString());

address.setText(model.getValueAt(i,5).toString());

email.setText(model.getValueAt(i,6).toString());

age.setText(model.getValueAt(i,7).toString());

sex.setText(model.getValueAt(i,8).toString());

dob.setText(model.getValueAt(i,9).toString());

occupation.setText(model.getValueAt(i,10).toString());

marry.setText(model.getValueAt(i,11).toString());

religion.setText(model.getValueAt(i,12).toString());

complaint.setText(model.getValueAt(i,13).toString());

diet.setText(model.getValueAt(i,14).toString());

habit.setText(model.getValueAt(i,15).toString());

appetite.setText(model.getValueAt(i,16).toString());

likes.setText(model.getValueAt(i,17).toString());

dislikes.setText(model.getValueAt(i,18).toString());

thirst.setText(model.getValueAt(i,19).toString());

stool.setText(model.getValueAt(i,20).toString());

urine.setText(model.getValueAt(i,21).toString());

sweat.setText(model.getValueAt(i,22).toString());

sleep.setText(model.getValueAt(i,23).toString());
```

```
dreams.setText(model.getValueAt(i,24).toString());  
gynac.setText(model.getValueAt(i,25).toString());  
childdev.setText(model.getValueAt(i,26).toString());  
mental.setText(model.getValueAt(i,27).toString());  
family.setText(model.getValueAt(i,28).toString());  
parent.setText(model.getValueAt(i,29).toString());  
examination.setText(model.getValueAt(i,30).toString());  
investigation.setText(model.getValueAt(i,31).toString());  
diagno.setText(model.getValueAt(i,32).toString());  
remedy.setText(model.getValueAt(i,33).toString());  
}
```

## CHAP 5: SYSTEM IMPLEMENTATION

### 5.1 Hardware and Software Requirements

#### 5.2 Screen/report Layouts

### 5.1 Hardware & Software Requirements

#### Hardware Requirements:

The minimum Hardware and system Software requirements for development and using this is:

Processor: 1.2GHz and above

RAM: Minimum requirement is 2 GB

#### Software requirements:

##### Client side requirements

- WINDOWS OS

## Sever side requirements


- Windows 10
- MySQL

## 5.2 Screen Layouts

DOCPRO-LOADING

♥ **DOCPRO - DOCTOR'S ASSISTANT** ♥

LOADING DOCPRO....PLEASE WAIT....



TIPS AND NOTICE FROM DEVELOPER  
\*USE THE TAB BUTTON TO MOVE TO NEXT FIELD  
\*USE THE FORM OPTION TO ADD NEW DATA  
\*

43%

DOCPRO-LOGIN

♥ **DOCPRO - DOCTOR'S ASSISTANT** ♥

**LOGIN**


USERNAME:-

PASSWORD:-

LOGIN

RESET

EXIT



DOCPRO-REGISTER YOURSELF

DOCPRO - DOCTOR'S ASSISTANT

SIGNUP


USERNAME

PASSWORD

CREATE

BACK

EXIT



DOCPRO-CHOOSE

DOCPRO - DOCTOR'S ASSISTANT

FORM

DELETE RECORDS

VIEW PATIENT DATA

BACK

LOGOUT





## **CHAP 6: FUTURE ENHANCEMENT**

### **6. FUTURE ENHANCEMENT**

Future development of the project will depend on the feedback received from the user and developers ideas

Future enhancements include developing the following as stated below:

- Save records in text files
- Cloud saving of data
- Online service
- Signup using email
- Improve user interface
- Availability on various softwares

The above mentioned points are the enhancements which can be done increase the applicability and usage of this project.

Here we can store the records of the patients in the database using this software

## CHAP 7: REFERENCES AND BIBLIOGRAPHY

### **7. REFERENCES AND BIBLIOGRAPHY**

#### **Website:**

- <https://www.youtube.com>
- <https://www.google.com>
- <https://www.tutorialspoint.com>

#### **Books:**

- JAVA
- MYSQL FOR STARTERS