

**PROJECT REPORT TITLED**

**DISPENSARY RECORD MANAGEMENT  
SYSTEM (DRMS)**

Submitted by:

NAZNEEN PATEL	161
ANKITA PHADKE	170

**B.Sc.-(I.T.), SEMESTER – VI EXAMINATION  
2014-2015**

Under the guidance of  
**PROF.SUMEDHA BARVE**



**PARLE TILAK VIDYALAYA ASSOCIATION'S**

**SATHAYE COLLEGE**

Mumbai-400 0057.

Parle Tilak Vidyalaya Association's  
**SATHAYE COLLEGE**

Vile-Parle (East), Mumbai – 400 057

T.Y.B.Sc. [I.T.] SEMESTER – VI

2014 – 2015

Project certificate for B. Sc. – IT students

This is to certify that the project entitled **Dispensary Record Management System (DRMS)** undertaken at **Dr. Rajesh Barve's Homeopathy Dispensary, Vile Parle (E)** by **Ms. Ankita Atul Phadke** Seat No **170** in partial fulfillment of B. Sc. I. T. degree (Semester- VI) examination had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

It is further certified that she has completed all required phases of the project.

Project Guide

Co-ordinator  
B.Sc. [I.T.]

Examiner

## Table of contents

1. Introduction .....	4
2. Objective and scope of project.....	5
2.1. Objective of the project.....	5
2.2. Scope of the project.....	5
3. History.....	6
4. Definition of problem.....	7
5. System Analysis and Design.....	9
6. PERT Chart.....	10
7. Detailed lifecycle of the project.....	11
7.1. ER Diagram.....	11
7.1.1. Connectivity and cardinality.....	11
7.1.2. ER Notation.....	12
7.1.3. ER Diagram.....	13
7.2. DFD.....	14
7.2.1. Introduction to DFD.....	14
7.2.2. Rules for DFD.....	15
7.2.3. Context level diagram.....	16
7.2.4. Level-1 DFD .....	16
8. Process involved.....	18

<b>9. Methodology adapted, system implementation.....</b>	<b>25</b>
<b>9.1. Methodology adapted and System implementation.....</b>	<b>25</b>
<b>9.2. Hardware specifications.....</b>	<b>30</b>
<b>9.3. Software specifications.....</b>	<b>30</b>
<b>9.3.1. Front end technology.....</b>	<b>30</b>
<b>9.3.2. Back end technology.....</b>	<b>35</b>
<b>10. Feasibility study.....</b>	<b>37</b>
<b>10.1. Economic feasibility.....</b>	<b>37</b>
<b>10.2. Technical feasibility.....</b>	<b>38</b>
<b>10.3. Operation feasibility.....</b>	<b>38</b>
<b>10.4. Cost-benefit feasibility.....</b>	<b>38</b>
<b>11. Methodology used for testing.....</b>	<b>39</b>
<b>12. Coding.....</b>	<b>42</b>
<b>13. Data dictionary.....</b>	<b>75</b>
<b>14. Future extension.....</b>	<b>78</b>
<b>15. Conclusion.....</b>	<b>78</b>
<b>16. References and bibliography.....</b>	<b>79</b>
<b>17. Acknowledgement.....</b>	<b>80</b>

# Dispensary Record Management System (DRMS)

# 1. INTRODUCTION

A dispensary generally refers to a small setup with basic medical facilities where a doctor can provide a primary level of care. It doesn't have a hospitalization facility and is generally owned by single doctor. A dispensary system has basic functioning to book appointment for patients, manages the visit histories and also it takes care of billing and other information.

The present system for managing the records is completely manual system. It contains high documentation for managing thousands of record manually. It contains various registers for maintaining records such as daily appointments, patients' history, payments, medicine stock etc. Every new patient gets unique PID (Patient ID) and based on the PID whole patient history is maintained. Each patient holds different case paper which contains history of patient. Each visit's information is noted down on same paper.

- There are some limitations to the above described system
  - System works on high documentation, making it difficult to maintain.
  - Any kind of changes or modifications are tedious and effort taking work.
  - Manual system is prone to human errors which are difficult to detect & correct.
  - It also involves high cost and time efforts to manage this system.

The Dispensary Record Management System (DRMS) Software's objective is to make record management an automated system by reducing manual documentation work. The new system will have two logins for doctor and receptionist. The Doctor can access visit history and view the daily appointments of the patients. The Receptionist makes registration for the new patient, Book and Cancel the appointment for the patient and can handle consultancy charges. Every new registration of patient will get sequential system auto generated PID which will hold unique identity of each patient. This unique identity for patient will form basis of all further medication process which takes place for the patient.

## **2. OBJECTIVE AND SCOPE OF THE PROJECT**

### **2.1 OBJECTIVES OF THE PROJECT**

The Dispensary Record Management System (DRMS) Software's objective is to make record management an automated system by reducing manual documentation work.

Below are the detailed objectives of the system-

- Minimize the documentation efforts of the older system, so that it works independent of any hard documentation, making it easier to maintain.
- reduce the efforts and tedious nature of the overall maintenance of the work.
- Making the system based on software would reduce human errors so that those are easier to detect & correct.
- Thereby, reducing the high cost and time efforts to manage this system.
- Modification becomes easy, as changes in single place will be reflected in other place. Also, the records will become more readable and manageable.

### **2.2 SCOPE OF THE PROJECT**

The scope of the project basically revolves around development of below software functionalities by managing separate logins for doctor and receptionist

- (1) The doctor can manage the visit history primarily, where he/she would be able to make a note of the illness description of the patient as well as treatment prescribed.
- (2) The receptionist would handle registering patient, booking appointments and taking care of payment and billing process.

### **3. HISTORY OF THE SYSTEM**

The dispensary operators/ medical practitioners feel the need of an automated system due to below reasons-

- (i) The manual registration process leads to data redundancy and also gives an additional workload to the person who is in charge. If the patients have received treatment from the doctor earlier; they need to find back the details on that particular patient in the archive.
- (ii) Also, there are many ready-made soft wares available in the market which also has the end to end solution ready for use in dispensary management. However the requirement of the dispensary functionality doesn't involve all those options which the ready-made soft wares give. Hence, we feel a need to develop the DRMS software which gives specific solution and functionalities as required by the specific physician.



## **4. DEFINITION OF PROBLEM**

The software product to be produced is software which will automate the major manual operations for Dispensary Record Management System (DRMS). The management system will consist of all the registering/ searching records of the patients who come to the dispensary to take medicines as well as their appointment details. It will include the billing/payment system for the DRMS management. It will manage the order and maintenance of the medical products for the management system.

Present system:

The present system for managing the records is completely manual system. It contains high documentation for managing thousands of record manually .It contains various registers for maintaining records such as daily appointments , patients history, payments , medicine stock etc. Every new patient gets unique PID (Patient ID), based on that PID the whole patient history is maintained. Each patient holds different case paper which contains history of patient and visit information is noted down on same paper.

New system:

The Record Management System (DRMS) Software's objective is to make record management an automated system by reducing manual documentation work. The new system will have various modules such as login for authorized users such as doctor or receptionist, patient's registration, search tab, visitor's detail, payment, printing bills. Modules such as daily appointments will be managed by receptionist where as modules such as patient's history or registration will be handled by doctor. Every new registration of patient will get sequential system auto generated PID which will hold unique identity of each patient. Other modules such as medicine stock will have information about current status of amount of stock and as per use the information will be updated automatically.

Issues with present system:

- System works on high documentation, making it difficult to maintain.
- Any kind of changes or modifications are tedious and effort taking work.
- Manual system is prone to human errors which are difficult to detect.

Advantages of DRMS over manual system:

- All records will be maintained and managed by system it will reduce human efforts.
- Automation of system will avoid human mistakes.
- Records will become more readable and manageable.
- Modification becomes easy, as changes in single place will be reflected in other place.

## **5. System Analysis and Design**

### **SUB-MODULES USED IN PROJECT**

This project includes the following sub-modules for development of the project. These are follows

#### **Login for Doctor and Receptionist**

Here doctor and receptionist enter valid user name and password that are checked in database so that doctor and receptionist can operate the application if valid or access is denied.

#### **Modules for Receptionist**

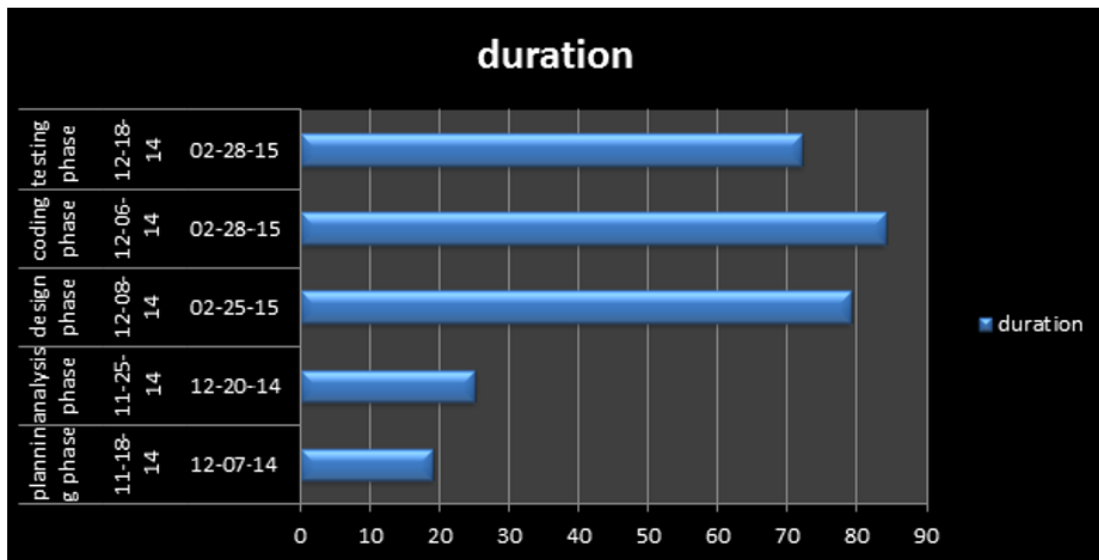
- Patient
  - Register Patients
  - Search Patient
- Appointments
  - Book New Appointment
  - Cancel Appointment
  - Search Appointment
  - Today's Appointment
- Medication
  - Bill Information
- Logout

#### **Modules for Doctor**

- Appointments
  - Today's Appointments
  - Visit History
- Logout

## 6. PERT Chart

### ➤ GANTT CHART:



Task Name	Start Date	End Date	duration
planning phase	11-18-14	12-07-14	19
analysis phase	11-25-14	12-20-14	25
design phase	12-08-14	02-25-15	79
coding phase	12-06-14	02-28-15	84
testing phase	12-18-14	02-28-15	72

## 7. Detailed lifecycle of the project

### 7.1 E-R DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationship. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the designer to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

#### 7.1.1 Connectivity and cardinality

The basic type of connectivity for relations is: one-to-one, one-to-many, and many-to-many. A one-to-one (1:1) Relationship is when at most one instance of an entity A is associated with one instance of entity B. For example, “employees in the company are each assigned their own office. For each employee there exists a unique office and for each office and for each office there exists a unique employee.

A one-to-many (1:N) relationship is when for one instance of entity A, there are zero, one, or many instances of entity B, but for one instance of entity B, there is only one instance of entity A, but for example of a 1:N relationships is a department has many employees each employee is assigned to one department.

A many-to-many (M:N) relationship, sometimes called non-specific, is when for one instance of entity A, there are zero, one, or many instances of entity B and for one instance of entity B there are zero, one, or many instances of entity A. the connectivity of a relationship describes the mapping of associated.

### **7.1.2 E R Notation :**

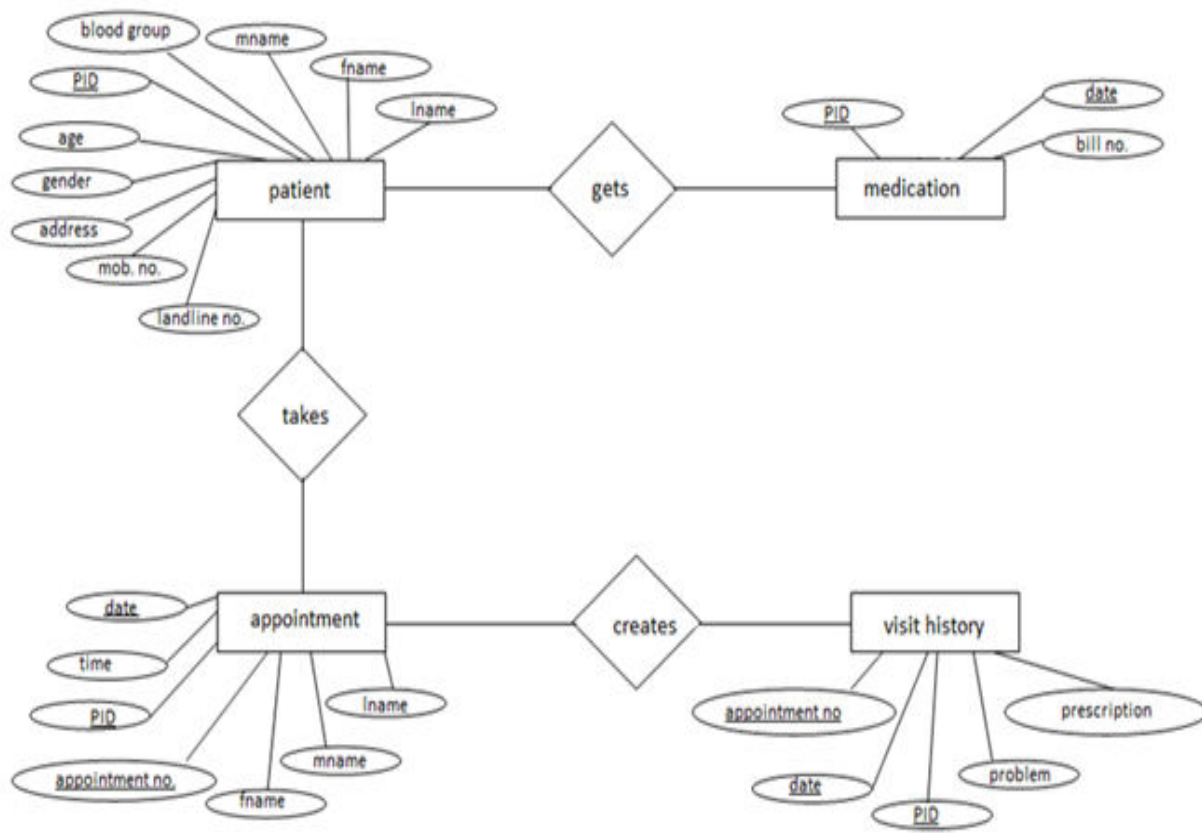
There is no standard for representing data objects in ER diagrams. Each modelling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publication by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX. All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in these documents is from Martin. The symbols used for the basic ER constructs are:

- Entities: are represented by labelled rectangles. The label is the name of the entity. Entity names should be singular nouns.
- Relationships: are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs.
- Attributes: when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined.

Attribute names should be singular nouns.

- Cardinality: of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.
- Existence: is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

### 7.1.3 ER Diagram



## **7.2 DATA FLOW DIAGRAM**

### **7.2.1 Introduction to DFD**

The DFD takes an input-process-output view of an system i.e. data objects flow into the software, are transformed by processing elements, and resultant data objects flow out of the software.

Data objects represented by labelled arrows and transformation are represented by circles also called as bubbles. DFD is presented in a hierarchical fashion i.e. the first data flow model represents the system as a whole. Subsequent DFD refine the context diagram (level 0 DFD), providing increasing details with each subsequent level.

The DFD enables the software engineer to develop models of the information domain & functional domain at the same time. As the DFD is refined into greater & functional domain at the same time, at the same time, the DFD refinement results in a corresponding refinement of the data as it moves through the process that embody the applications.

A context-level DFD for the system the primary external produce information for use by the system and consume information generated by the system. The labelled arrow represents data objects or object hierarchy.

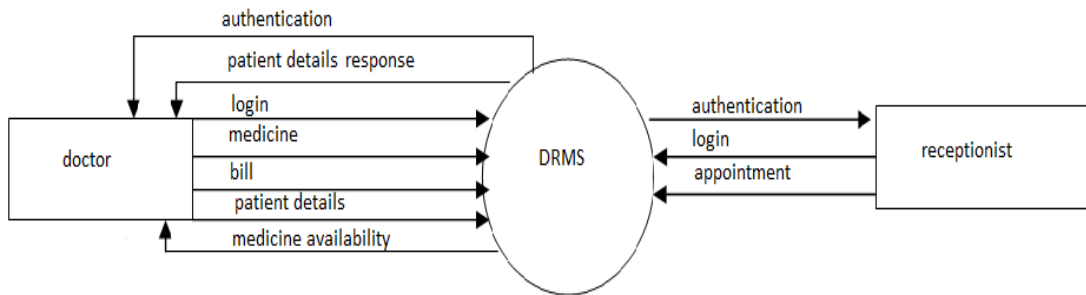


## 7.2.2 RULES FOR DFD:

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions.
- Reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to system with rounded circles.
- A process is required for all the data transformation and Transfers. Therefore, never connect a data store to a data Source or the destinations or another data store with just a Data flow arrow.
- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.
- Indicate external sources and destinations of the data, with Squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Records retrievals.

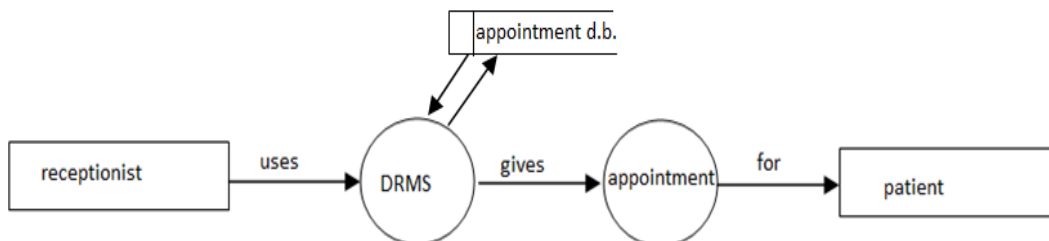
- Label data flows on each arrow.
- Use details flow on each arrow.

### 7.2.3 Context Level DFD

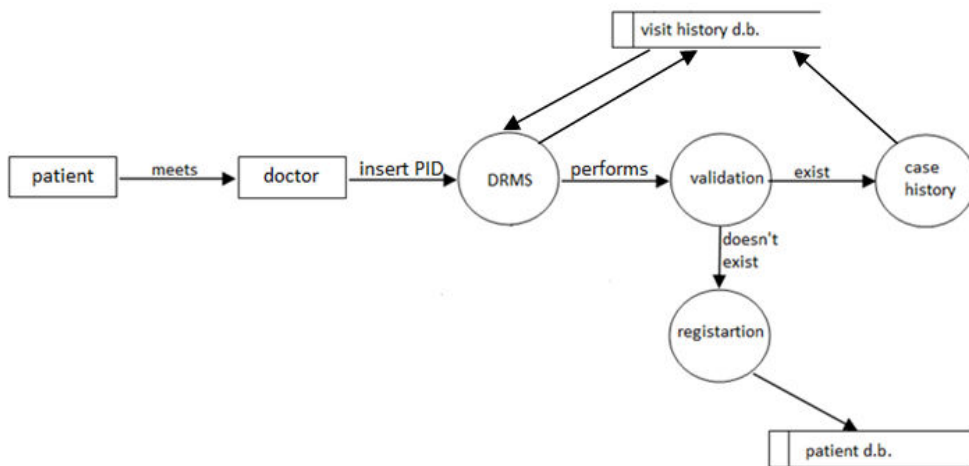


context level DFD

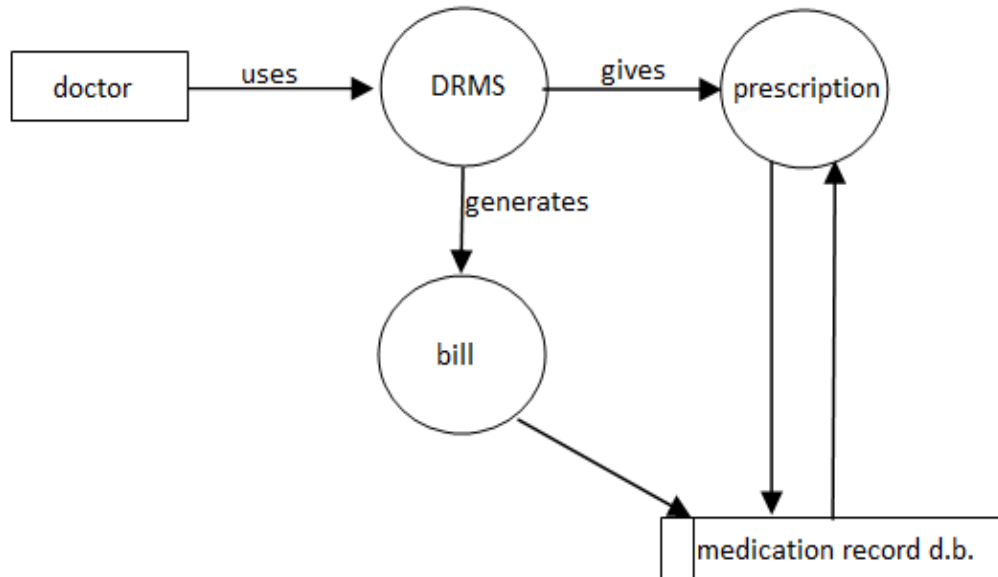
### 7.2.4 Level-One DFD



DFD for appointment of patient



**DFD for patient visit**

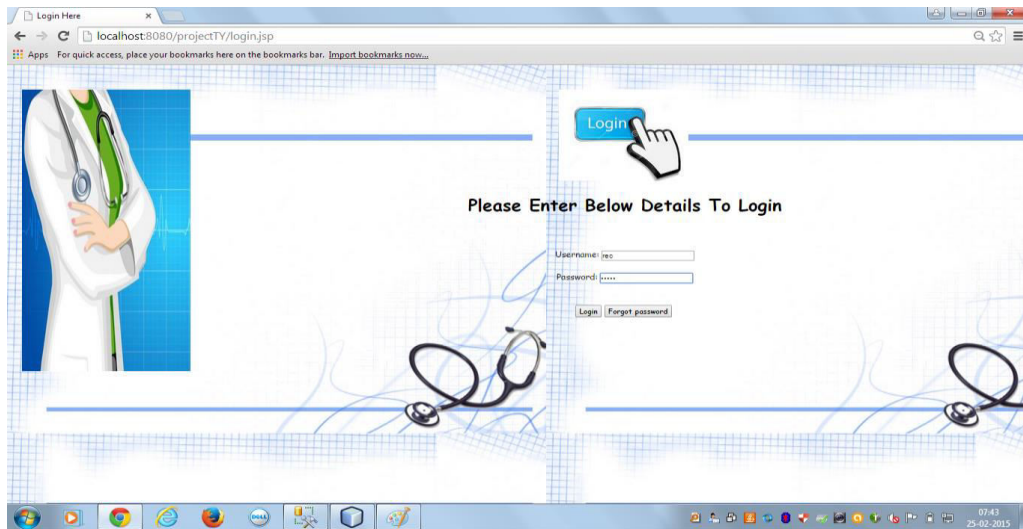


**DFD for medication**

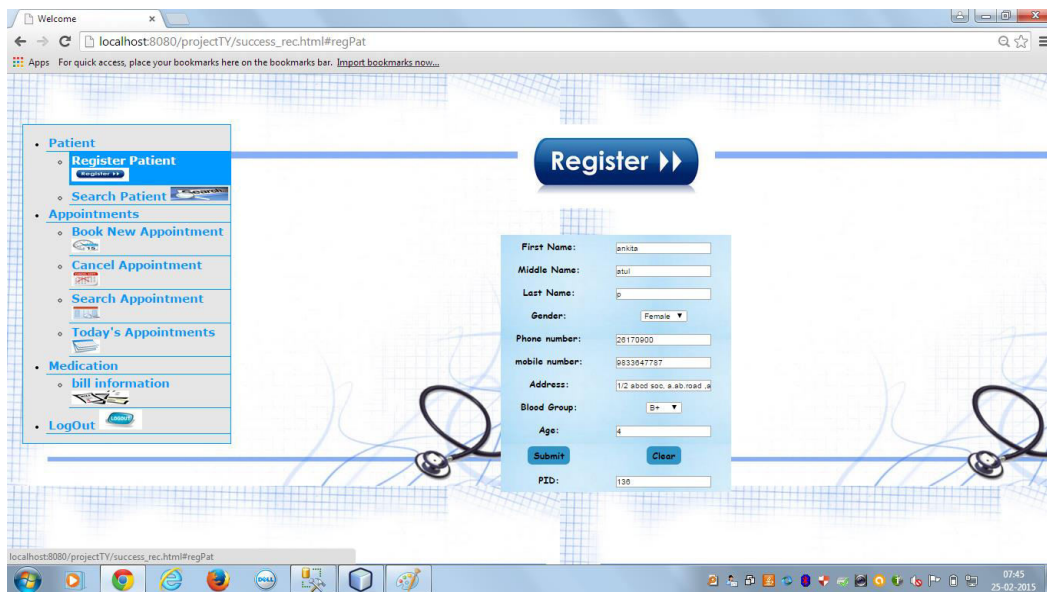
## 8. Process involved

### Process flow for receptionist:

1. **Receptionist Login Page**-this page lets the receptionist login to the system with his/her own role login.



2. **Register Patient**-this page registers a patient into the system, with his personal details, unique fields are first, middle, last name & mobile no. and generates a unique patient ID for the patient.



**3. Search Patient**-this page searches the patient ID in system using first, middle, last name and mobile no.

localhost:8080/projectTV/success\_rechtml#search\_patient

- Patient
  - Register Patient
  - **Search Patient**
- Appointments
  - Book New Appointment
  - Cancel Appointment
  - Search Appointment
  - Today's Appointments
- Medication
  - bill information
- LogOut

Search...

First Name: anjita  
Middle Name: ajul  
Last Name: p  
mobile no.: 9833047787  
patient id: 138

Search Clear

**4. Book New Appointment**-this page books new appointment, using selected date, appointment number and patient ID.

localhost:8080/projectTV/success\_rechtml#book

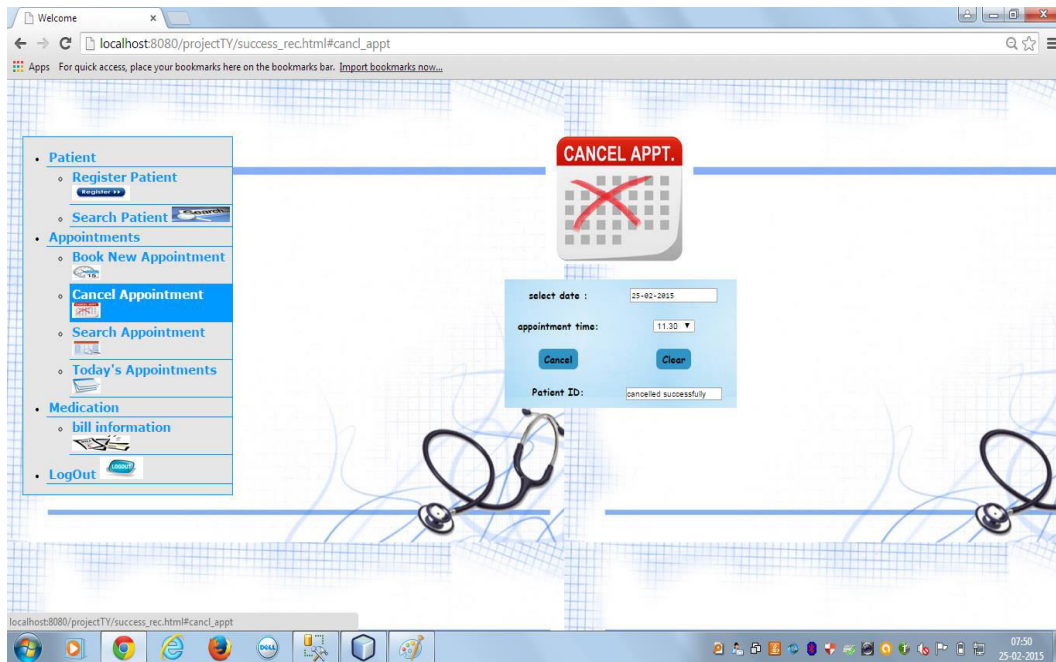
- Patient
  - Register Patient
  - Search Patient
- Appointments
  - **Book New Appointment**
  - Cancel Appointment
  - Search Appointment
  - Today's Appointments
- Medication
  - bill information
- LogOut

select date : 25-02-2015  
appointment time: 11:30  
Patient ID: 132

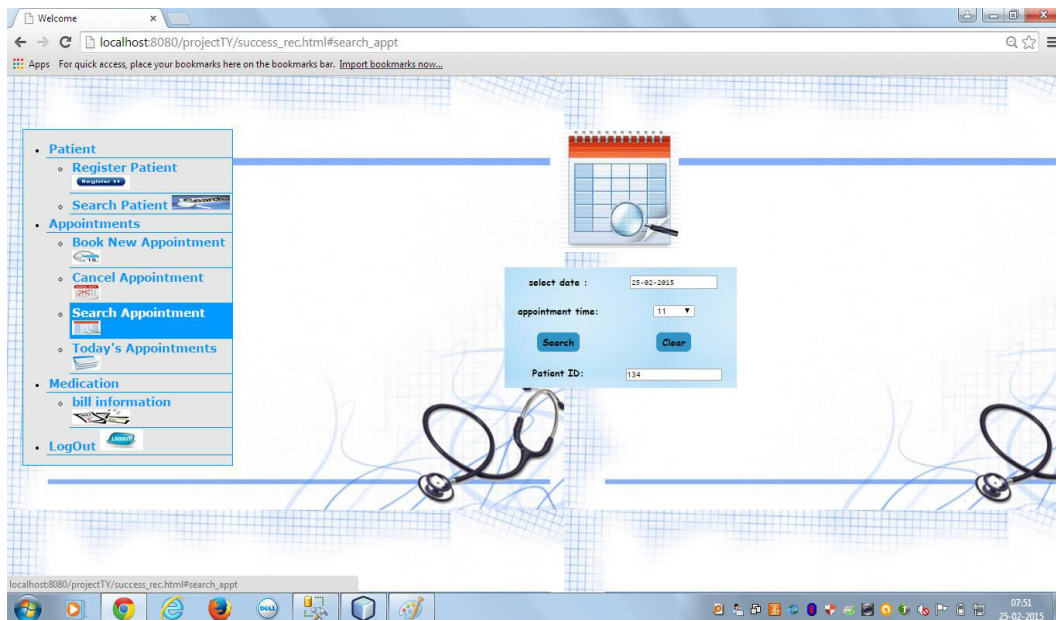
Book Clear

Booking Status: booked successfully

**5. Cancel Appointment-** this page cancels already booked appointment and deletes the record from database.

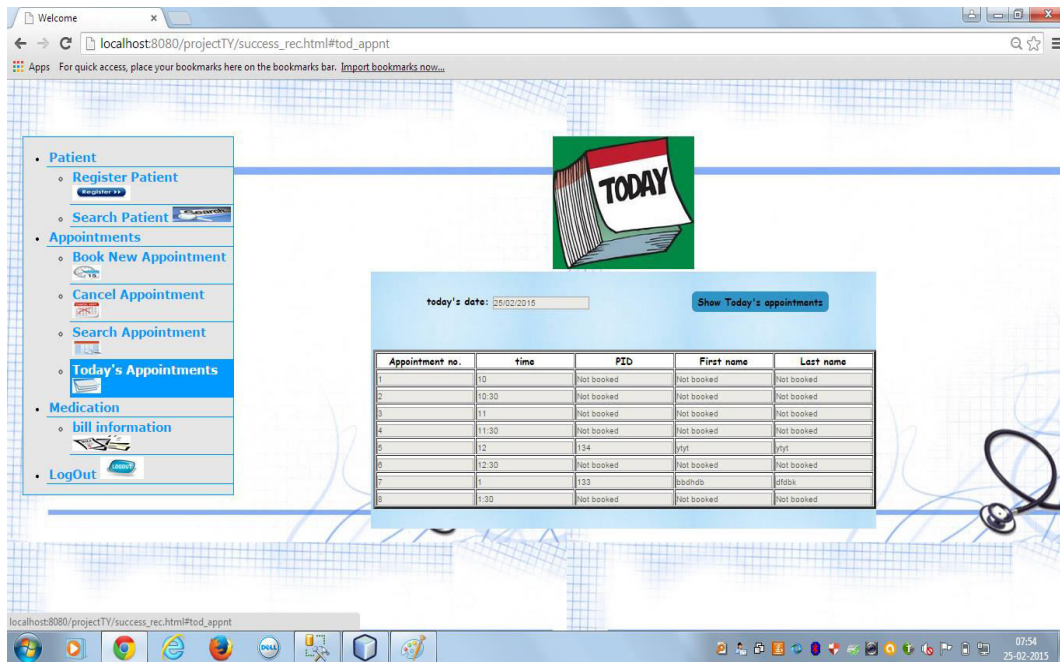


**6. Search Appointment-** this page allows the search for an already booked appointment, using appointment date and time.

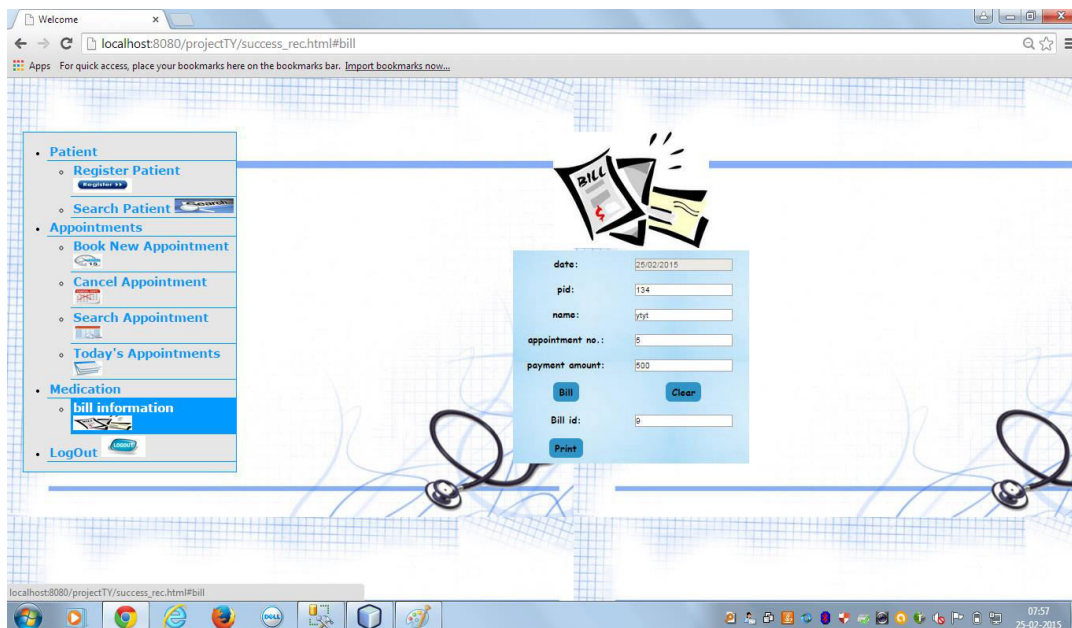




## 7. Today's Appointment-this page displays all the booked appointment for current system date.

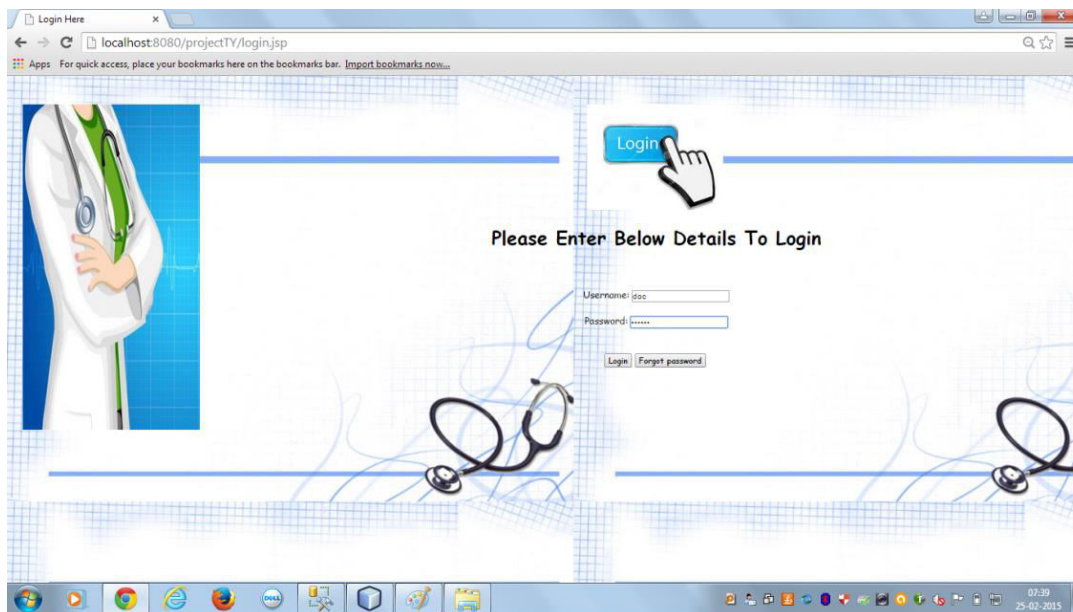


## 8. Bill Information-this page allows storing the payment amount and billing information and allows printing it.

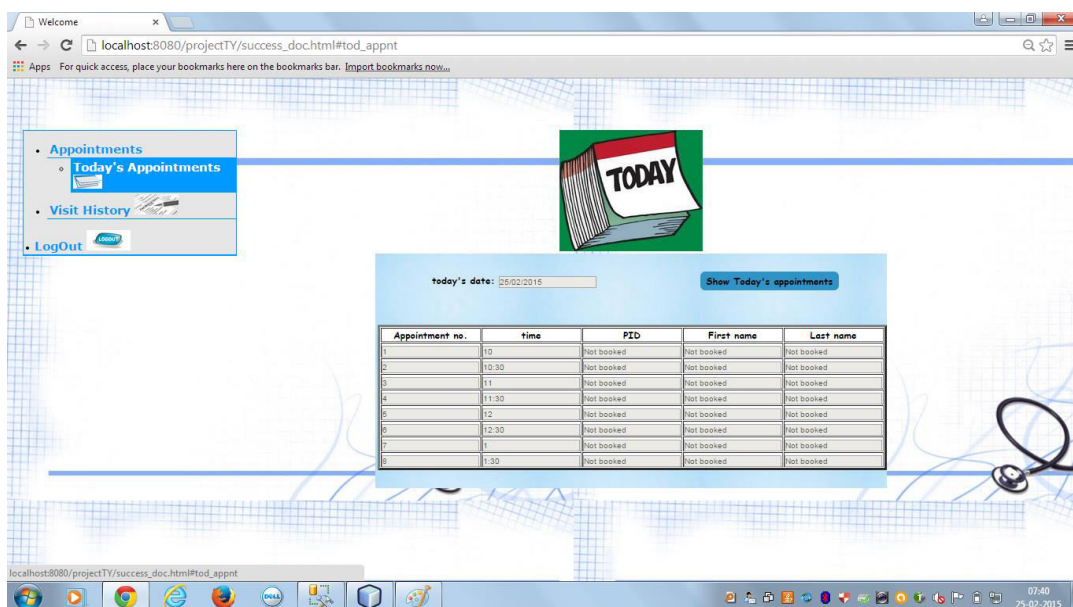


## Process flow for Doctor:

1. **Doctor Login-** this page lets the receptionist login to the system with his/her own role login.

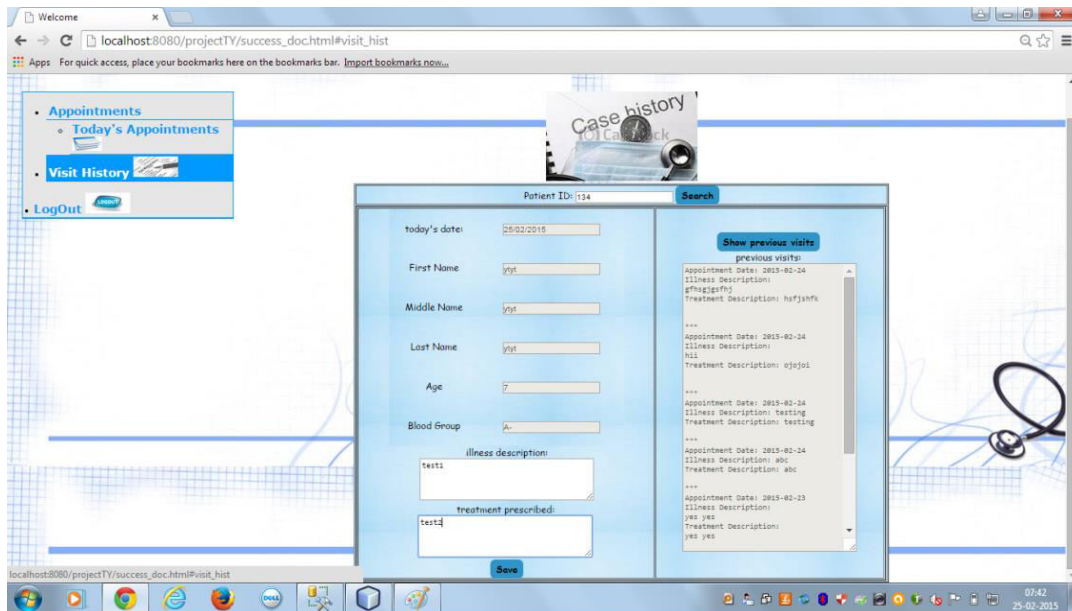


2. **Today's Appointment-** this page displays all the booked appointment for current system date.

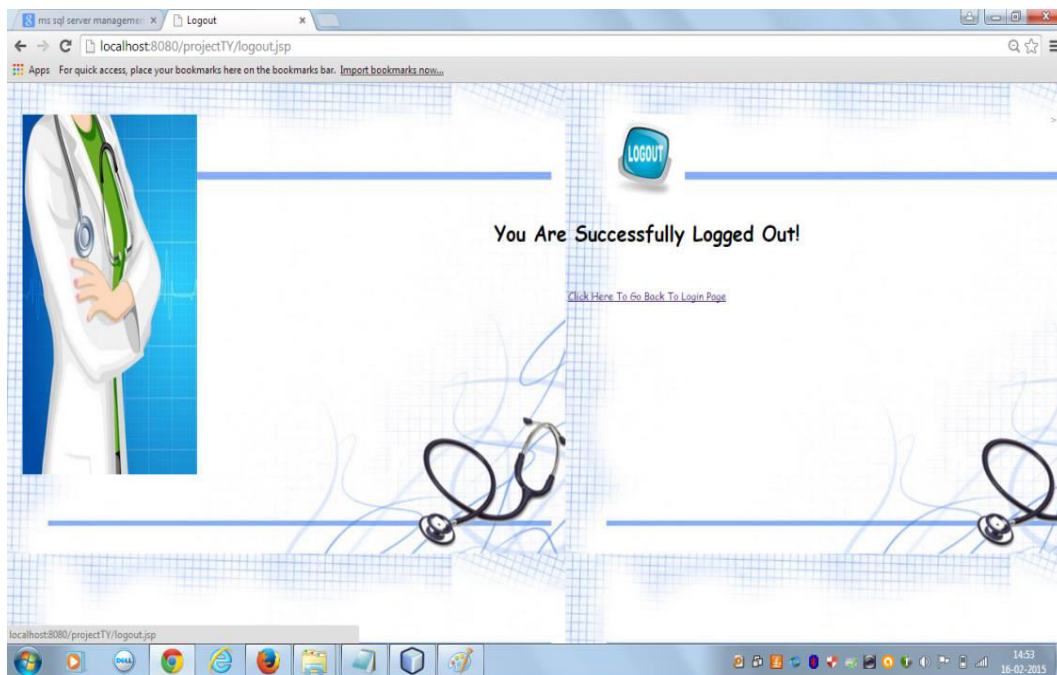




**3. Visit History** this page displays the previous visit details for the patient selected, as well as allows to save new details as on current date.



**4. Logout-** this page lets the user log out of the session



5. **Password Recovery**- this page lets the user recover the password by answering secret questions. The system does a look up on previously stored secret question-answer pairs and prompts the password only if the user trying to login is a genuine one.

JSP Page  
localhost:8080/projectTV/Forget.jsp

Apps For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Login

Please answer below questions to know your password

enter user id:  [Show questions](#)

1

answer here:

2

answer here:

[Show password](#)

11:16  
25-02-2015

## **9. Methodology adapted, system implementation, details of H/W and S/W**

### **9.1 Methodology adapted & System implementation**

The software methodology followed in this project includes the object-oriented methodology and the application system development methodologies. The description of these methodologies is given below:

#### **Application System Development- A Life Cycle Approach**

Although there are a growing number of applications (such as decision support systems) that should be developed using an experimental process strategy such as prototyping, a significant amount of a new development work continue to involve major operational applications of broad scope. The application systems are large highly structured. User task comprehension and developer task proficiency is usually high. These factors suggest a linear or iterative assurance strategy. The most common method for this stage class of problems is a system development life cycle modal in which each stage of development is well defined and has straight forward requirements for deliverables, feedback and sign off. The system development life cycle is described in detail since it continues to be an appropriate methodology for a significant part of new development work.

The phases in the life cycle for information system development are described differently by different writers, but the differences are primarily in the amount of necessity and manner of categorization. These are a general agreement on the flow of development steps and the necessity for control procedures at each stage.

The information system development cycle for an application consists of three major stages:

- Definition
- Development
- Installation and operation

The first stage of the process defines the information requirements for a feasible cost effective system. The requirements are then translated into a physical system of forms, procedures, programs etc., by the system design, computer programming and procedures development. The resulting system is test and put into operation. No system is perfect so there is always a need for maintenance changes. To complete the cycle, there should be a post audit of the system to evaluate how well it performs and how well it meets the cost and performance specifications. The stages of definition, development and installation and operation can therefore be divided into smaller steps or phrase as follows.

- Definition:
  - Proposed definition: preparation of request for proposed application.
  - Feasibility assessment: evaluation of feasibility and cost benefit of proposed system.
  - Information requirement analysis: determination of information needed.

- Design:
  - Conceptual design: User-oriented design of application development.
  - Physical system design: Detailed design of flows and processes in applications processing system and preparation of program specification.
- Development:
  - Program development: coding and testing of computer programs.
  - Procedure development: design of procedures and preparation of user instructions.
- Installation and operation:
  - Conversion: Final system test and conversion.
  - Operation and Maintenance: Month to month operation and maintenance.
  - Post audit: Evaluation of development process, application system and results of use at the completion of the each phase, formal approval sign-off is required from the users as well as from the manager of the managers of the project development.

# Iterative Model

## Definition

An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model.

## Advantages of Iterative model

- In iterative model we can only create a high-level design of the application before we actually begin to build the product and define the design solution for the entire product. Later on we can design and built a skeleton version of that, and then evolved the design based on what had been built.
- In iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.
- In iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.
- In iterative model less time is spent on documenting and more time is given for designing.

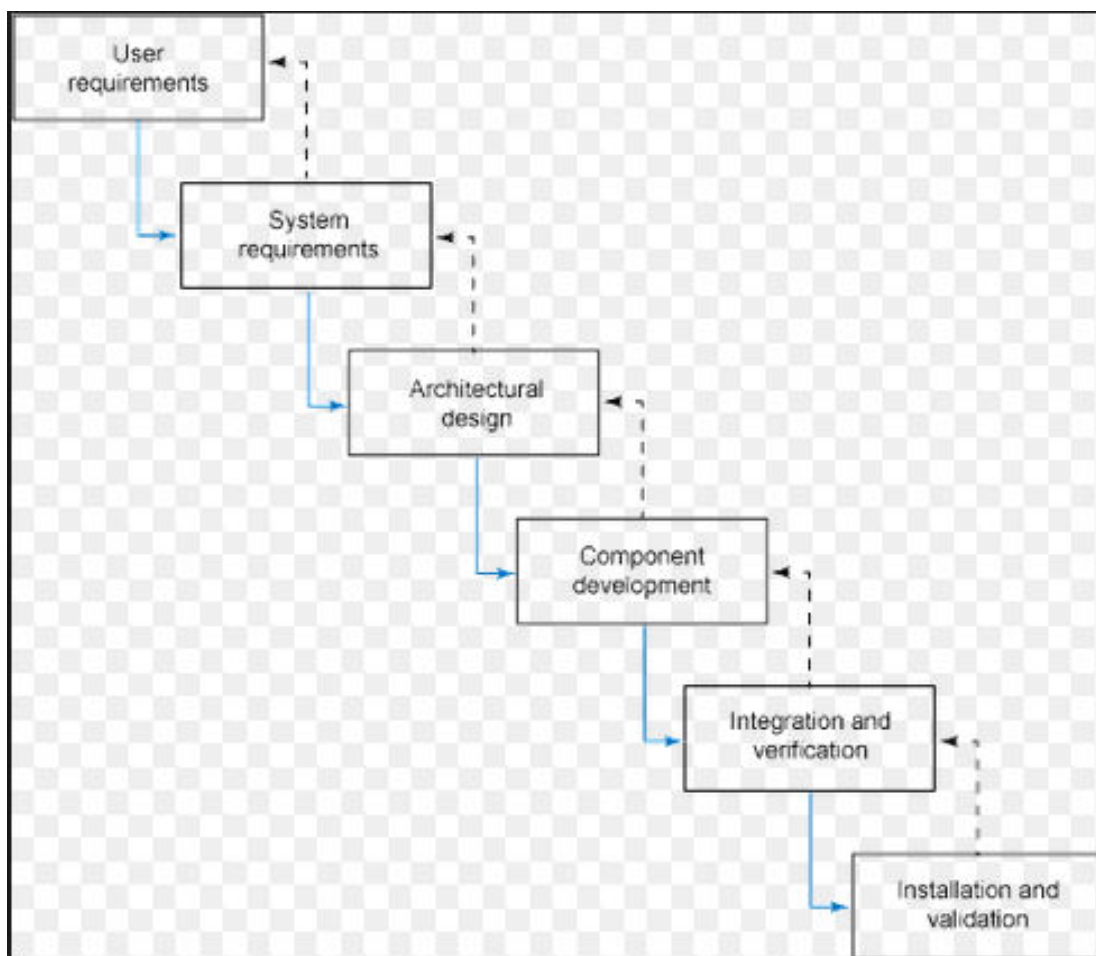
## Disadvantages of Iterative model

- Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire lifecycle
- Each phase of an iteration is rigid with no overlaps

## When to use iterative model

- Requirements of the complete system are clearly defined and understood.
- When the project is big.
- Major requirements must be defined; however, some details can evolve with time.

## Diagram



## **System specifications**

### **9.2 Hardware Requirements**

- 1 GB Ram
- Hard disk: Min. 1 GB
- Processor:- Intel Core 2 Duo
- Microsoft Compatible 101 or more Key Board

### **9.3 Software Requirements**

- Web Technologies : NetBeans IDE 7.1.1
- Language : JAVA/J2EE
- Database : SQL SERVER MANAGEMENT STUDIO 2008
- Web Server : Oracle Glass Fish Server
- Operating System : WINDOWS 7
- Browser: Google Chrome

#### **9.3.1 Front end technology**

- **Introduction to J2E Frame work:**

NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++ and HTML5. It is also an application platform framework for Java desktop applications and others. The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. The NetBeans Platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans Platform (including the NetBeans IDE itself) can be extended by third party developers.



- **J2E framework architecture:**

The Java EE Connector Architecture defines a standard for connecting a compliant application server to an EIS. It defines a standard set of system-level contracts between the Java EE application server and a resource adapter. The system contracts defined by Version 1.0 of the Java EE Connector Architecture are described by the specification as follows:

**Connection management**—Connection management enables an application server to pool connections to the underlying EIS and enables application components to connect. This leads to a scalable application environment that can support a large number of clients.

**Transaction management**—Transaction management enables an application server to use a transaction manager to manage transactions across multiple resource managers.

**Security management**—Security management reduces security threats to the EIS and protects valuable information resources managed by the EIS.

- **Introduction to JAVA:**

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is one of the most popular programming languages in use, particularly for client-server web applications.

Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

- **Introduction to SERVELETS**

- Provide dynamic content such as the results of a database query.
- Manage state information that does not exist in the stateless HTTP protocol, such as filling the articles into the shopping cart of the appropriate customer.
- Process or store data that was submitted from an HTML form.
- "Servlet" is a Java class in Java EE that conforms to the Java Servlet API, a standard for implementing Java classes which respond to requests.
- Servlets could in principle communicate over any client–server protocol, but they are most often used with the HTTP protocol.

- **Introduction to AJAX:**

- Ajax (Asynchronous JavaScript and XML) is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications.
- With Ajax, web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.
- Ajax is not a single technology, but a group of technologies.

- HTML and CSS can be used in combination to mark up and style information. The DOM is accessed with JavaScript to dynamically display – and allow the user to interact with – the information presented.
- JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

- **Introduction to JQUERY:**

- DOM manipulation based on CSS selectors that uses elements' names and attributes, such as id and class, as criteria to select nodes in the DOM.
- Provide Effects and animations.
- Deferred and Promise objects to control asynchronous processing.
- Provide Extensibility through plug-ins.
- Utilities, such as user agent information, feature detection.

- **Introduction to JSP:**

- Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified.
- JSP can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans.
- JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document.
- Like any other Java program, they must be executed within a Java virtual machine (JVM).

- JSPs are usually used to deliver HTML and XML documents, but through the use of OutputStream, they can deliver other types of data as well.

- **Features Of JAVA:**

- Simple :  
Java is Easy to write and more readable and eye catching.
- Secure :  
Java provides secure way to access web applications.
- Portable :  
Java programs can execute in any environment for which there is a Java run-time system (JVM). Java programs can be run on any platform (Linux, Window, Mac).
- Object-oriented :  
Like C++ java provides most of the object oriented features.
- Robust :  
Java encourages error-free programming by being strictly typed and performing run-time checks.
- Multithreaded :  
Java provides integrated support for multithreaded programming.
- Architecture-neutral :  
Java is not tied to a specific machine or operating system architecture. Machine Independent i.e. Java is independent of hardware.
- Interpreted :  
Java supports cross-platform code through the use of Java bytecode. Bytecode can be interpreted on any platform by JVM.
- High performance :  
Bytecodes are highly optimized. JVM can executed them much faster.
- Distributed :  
Java was designed with the distributed environment. Java can be transmit, run over internet.

### 9.3.2 Backend technology

- **Introduction to SQL Server Management Studio 2008:**

SQL Server Management Studio is a software application first launched with the Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server.

A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It also has an "express" version that can be freely downloaded.

It is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users. Its primary query languages are T-SQL and ANSI SQL.

Architecture: The protocol layer implements the external interface to SQL Server. All operations that can be invoked on SQL Server are communicated to it via a Microsoft-defined format, called Tabular Data Stream (TDS). TDS is an application layer protocol, used to transfer data between a database server and a client.

DataBase: Data storage is a database, which is a collection of tables with typed columns. SQL Server supports different data types, including primary types such as *Integer*, *Float*, *Decimal*, *Char* (including character strings), *Varchar* (variable length character strings), binary (for unstructured BLOBs of data), *Text* (for textual data) among others. The rounding of floats to integers uses either Symmetric Arithmetic Rounding or Symmetric Round Down (*Fix*) depending on arguments.

Microsoft SQL Server also allows user-defined composite types (UDTs) to be defined and used. It also makes server statistics available as virtual tables and views (called Dynamic Management Views or DMVs). In addition to tables, a database can also contain other objects including views, stored procedures, indexes and constraints, along with a transaction log.

## **10. Feasibility study**

Feasibility studies aim to objectively and rationally uncover the strengths and weaknesses of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospects for success. In its simplest terms, the two criteria to judge feasibility are cost required and value to be attained.

A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

A feasibility study evaluates the project's potential for success; therefore, perceived objectivity is an important factor in the credibility of the study for potential investors and lending institutions. It must therefore be conducted with an objective, unbiased approach to provide information upon which decisions can be based. The assessment is based on an outline design of system requirements,. When studying feasibility, following factors should be taken into account:

- A brief description of the software to assess more possible factors which could affect the study
- The part of the software being examined
- The human and economic factor
- The possible solutions to the problem

### **Types of feasibility study-**

#### **10.1 Economical feasibility:**

The software is developed using NetBeans IDE 7.1.1 and SQL Server Management Studio 2008 version which is freely available; so no extra cost is invested behind purchasing new System License. Hence the Dispensary Record Management System is Economically Feasible.

## **10.2 Technical Feasibility:**

For using this system the required hardware configuration is available on low cost and platform required for implementing system needs to be purchased which is one time investment. The system manages memory efficiently and gives better performance on low configuration of hardware. Therefore after investing once the benefit will be more after implementation of System. Hence the Dispensary Record Management System is Technically Feasible.

## **10.3 Operational Feasibility:**

The System provides GUI interface which is very easy to operate and a non-technical person can also operate the system efficiently. During development of the system, the care is taken to ensure smooth working of the system. The system will provide different activities for the doctor according to the patient and generate appropriate results for the same. Therefore the Dispensary Record Management System is operationally feasible.

## **10.4 Costs-Benefit Feasibility:**

Before the implementation of this system most of the work was done manually which made various tasks tedious and the records were kept in large registers; so maintaining the records and finding them was very tedious and time consuming.



## **11. Methodology used for testing**

### **Software Testing**

- **TESTING**

Testing is the most vital phase in completing a project. Testing mainly aims at checking the modularity, data flow and code thereby scrutinizing the intricacies of the system being developed.

A through testing has been carried out for this system. At all the required places appropriate validation has been done. User involvement at every stage has been propounded in the system. Testing is the crucial part of the system development to assure quality of the services, design and coding.

#### **TESTING OBJECTIVES**

Testing objective is the guideline that helps in carrying out proper testing of the system. The guidelines stated below are followed to carry out proper testing on the system

Testing is the process of executing a program with a goal to uncover the errors within the system. A successful test needs to be carried out in order to find undetected errors

The main objective is to design the tests that systematically uncover different types of errors with minimum amount of time and errors. Testing demonstrates that the system functions appear to be working in a proper manner and that the performance requirements are met.

## Formal Technical Review

A formal technical review is a form of a peer review in which a team of qualified personnel examines the suitability of the software product for its intended use and identifies discrepancies from specifications and standards. The purpose of technical reviews is to arrive at a technically superior version of the product reviewed, by correction of defects. There are various responsibilities assigned to the team members regarding the project.

- User participation is important for performing administrative tasks relative to the review, ensuring orderly conduct and ensuring that the reviews meet its objectives.
- One of the team documents anomalies, action items, decisions and recommendations made by the review team.
- Technical experts are the active participants in the review and evaluation of the software product.

## Test Plan

Test plan provides a road map for instituting software testing. This describes the overall testing strategy and the project management issues that are required to properly execute effective tests. Testing plan contains various test phases, start and end dates for each phase is described. Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item.

Software Test Plan (STP) is designed to prescribe the scope, approach, resource and schedule of all testing activities.

The plan must identify the items to be tested, the features to be tested, the types of testing to be performed, the personnel responsibilities for testing, the resources and schedule required to complete testing, and the risks associated with the test plan.

## Testing Methodology

- **Black box testing:**

**The first type** of testing was at user level where all the applications were tested for the users.

- **White box testing:**

The second level of testing was at functional level where each module of all the applications was tested. This testing continued throughout the entire project, evolving from component level (or unit) testing to integration testing.

- **Unit testing:**

This testing focuses effort on the smallest part of software design-the module or software component. After the code has been developed, reviewed and verified for correspondence to component level design, unit testing case design begins.

- **Integration testing:**

Unit testing ensures that all the modules are working fine independently but now we have to test the system as a whole; we have to develop a strategy, which sees that the system is functioning properly as a whole system.

Integration testing is systematic testing for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. We will use bottom up integration testing where we will integrate the modules by starting with the lower individual modules and moving upward through the control hierarchy.

## 12. Coding

### Register patient page coding-

#### regServlet.java: to handle the database transactions

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package servlets;

import java.awt.Window;
import java.sql.Statement;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Grish
 */
@WebServlet(name = "regServlet", urlPatterns = {"/regServlet"})
public class regServlet extends HttpServlet {

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on
    the left to edit the code.">
    /**
     * Handles the HTTP

```

```

* <code>GET</code> method.
*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        processRequest(request, response);
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(regServlet.class.getName()).log(Level.SEVERE, null, ex);
    } catch (SQLException ex) {
        Logger.getLogger(regServlet.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
* Handles the HTTP
* <code>POST</code> method.
*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    try {
        int pid1=0;
        PrintWriter out = response.getWriter();
        String fnm=request.getParameter("fname");
        String mnm=request.getParameter("mname");
        String lnm=request.getParameter("lname");
        String gend=request.getParameter("g1");
        String phn=request.getParameter("phnum");
        String mob=request.getParameter("mob");
        String addr=request.getParameter("addr");
        String bld_grp=request.getParameter("c");
        String age = request.getParameter("age");

```

```

try {

    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con=DriverManager.getConnection("jdbc:odbc:dsn1216");
    Statement st=con.createStatement();

    ResultSet rs=st.executeQuery("select * from patient where fname = '"+fnm+"' and
mname = '"+mnm+"' and lname = '"+lnm+"' and mob = '"+mob+"'");
    if(rs.next()==true)
    {
        pid1=Integer.parseInt(rs.getString("pid"));
        out.println("already registered with PID = ");
    }
    else
    {
        st.executeUpdate
        ("insert into patient
values('"+fnm+"','"+mnm+"','"+lnm+"','"+gend+"','"+phn+"','"+mob+"','"+addr+"','"+bld_grp+"','
"+age+"')");

        ResultSet rs2=st.executeQuery("select max(pid) as pid from patient");
        if(rs2.next()==true)
        {
            pid1=Integer.parseInt(rs2.getString("pid"));
        }
    }
    out.println(pid1);
}
finally
{
    out.close();
}
}
catch (ClassNotFoundException ex)
{
    Logger.getLogger(regServlet.class.getName()).log(Level.SEVERE, null, ex);
}
catch (SQLException ex)
{
    Logger.getLogger(regServlet.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

### **regPat.jsp: to handle to handle the frontend design of the form**

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" >
    <div>
        
    </div>
    <title> Register Patient</title>
    <script src="jquery-1.11.1.min.js"></script>
    <script type="text/javascript">
        var form = $('#f1');
        $('#calc').click(function(){
            if(document.getElementById("fname").value == null ||
document.getElementById("fname").value == "")
            {
                alert("Please enter first name!");
            }
            else if(document.getElementById("mname").value == null ||
document.getElementById("mname").value == "")
            {
                alert("Please enter middle name!");
            }
            else if(document.getElementById("lname").value == null ||
document.getElementById("lname").value == "")

```

```

{
    alert("Please enter last name!");
}
else if(document.getElementById("phnum").value == null ||
document.getElementById("phnum").value == "")
{
    alert("Please enter phone number!");
}
else if(document.getElementById("phnum").value != null ||
document.getElementById("phnum").value != "")
{
    var num = /^[0-9]+$/;
    if(phnum.value.match(num))
    {
        if(phnum.value.length != 8)
        {
            alert("Incorrect phone number!");
        }
        else
        {
            if(document.getElementById("mob").value == null ||
document.getElementById("mob").value == "")
            {
                alert("Please enter mobile number!");
            }
            else if(document.getElementById("mob").value != null ||
document.getElementById("mob").value != "")
            {
                var num = /^[0-9]+$/;
                if(mob.value.match(num))
                {
                    if(mob.value.length != 10)
                    {
                        alert("Incorrect mobile number!");
                    }
                    else
                    {
                        if(document.getElementById("addr").value == null ||

```



```

document.getElementById("addr").value == "")
    {
        alert("Please enter address!");
    }
    else if(document.getElementById("age").value == null ||
document.getElementById("age").value == "")
    {
        alert("Please enter age");
    }
    else if(document.getElementById("age").value != null ||
document.getElementById("age").value != "")
    {
        var num = /^[0-9]+$/;
        if(phnum.value.match(num))
        {
            if(document.getElementById("age").value <= 0)
            {
                alert("Invalid age format");
            }
            else
            {
                $.ajax({
                    type: form.attr('method'),
                    url: form.attr('action'),
                    data: $("#f1").serialize(),
                    success: function (data) {
                        var result=data;
                        $('#pid').attr("value",result);
                    }
                });
            }
        }
        else
        {
            alert("Incorrect age format!");
        }
    }
}

```

```

        }
        else
        {
            alert("Incorrect mobile number format!");
        }
    }
}
}
else
{
    alert("Incorrect phone number format!");
}
}
});
</script>
<script type="text/javascript">
var form = $('#f1');
    $('#clear').click(function(){
        alert("clear!");
        var result="";
        $.ajax({
            data: $('#f1').serialize(),
            success: function (data) {
                var result="";
                $('#fname').val("");
                $('#mname').val("");
                $('#lname').val("");
                $('#phnum').val("");
                $('#mob').val("");
                $('#addr').val("");
                $('#age').val("");
                $('#g1').val("Male").change();
                $('#c').val("O+").change();
            }
        });
    });
});
</script>

```

```

</head>

<body >
<form name="f1" id="f1" method="post" action="regServlet">
<table align="center" height="450" width="400" background="images/tbl_bckgrnd.jpg">
    <tr><td><b>First Name: </td><td><input type="text" name="fname"
id="fname"></td></tr>
    <tr><td><b>Middle Name: </td><td><input type="text" name="mname"
id="mname"></td></tr>
    <tr><td><b>Last Name: </td><td><input type="text"
name="lname" id="lname"></td></tr>
    <tr><td><b>Gender: </td><td><select name="g1">
    <option name="c1" value="male" selected="true">Male</option>
    <option name="c1" value="female">Female</option>
    </select></td></tr><br>
    <tr><td><b>Phone number: </td><td><input type="text" name="phnum"
id="phnum"></td></tr>

    <tr><td><b>mobile number: </td><td><input type="text" name="mob"
id="mob"></td></tr>
    <tr><td><b>Address: </td><td><input type="text" name="addr" width="100"
height="200" id="addr"></td></tr>
    <tr><td><b>Blood Group: </td><td><select name="c">
    <option name="c" value="O+" selected="true">O+</option>
    <option name="c" value="O-">O-</option>
    <option name="c" value="A+">A+</option>
    <option name="c" value="A-">A-</option>
    <option name="c" value="B+">B+</option>
    <option name="c" value="B-">B-</option>
    <option name="c" value="AB+">AB+</option>
    <option name="c" value="AB-">AB-</option>
    </select></td></tr><br>
    <tr><td><b> Age: </td><td><input type="text" name="age" id="age"
></td></tr>
    <tr><td>    <b><span id="calc" style="display:inline-block; padding:5px 10px
;background:#3194c6; -webkit-border-radius: 8px; -moz-border-radius: 8px;border-radius: 8px;
cursor:default">Submit</span></td><td>
    <b><span id="clear" style="display:inline-block; padding:5px 10px

```

```
;background:#3194c6; -webkit-border-radius: 8 px; -moz-border-radius: 8px;border-radius: 8px;
cursor:default">Clear</span> </td></tr>
```

```

                </td> <td> <b> <b> PID:</td><td><input type="text" name="pid" id="pid"
value=""></td></tr>
            </table>
        </form>

    </body>
</html>
```

## Book appointment page coding-

### bookAppt.java: to handle the database transactions

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Grish
 */
```

```

public class bookAppt extends HttpServlet {

    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        PrintWriter out = response.getWriter();
        try {
            /**
             * TODO output your page here. You may use following sample code.
             */
            String date=request.getParameter("date");
            String time=request.getParameter("a");
            int pid=Integer.parseInt(request.getParameter("pid"));

```

```

int pid_appt=0;
String pay_info = "0";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:dsn1216");

Statement st=con.createStatement();
ResultSet rs=st.executeQuery("select * from patient where pid = "+pid);
if(rs.next()==false)
{
    out.println("Invalid patient ID ");
}
else
{
    ResultSet rs2=st.executeQuery("select pid from appointment where appt_dt =
""+date+"" and appt_time = ""+time+"" ");

    if(rs2.next() == true)
    {
        pid_appt = Integer.parseInt(rs2.getString("pid"));
        out.println("Appointment is already booked for patient ID "+pid_appt);
    }
    else
    {
        int row_ins = st.executeUpdate("insert into appointment
values('"+date+"','"+time+"','"+pid+"','"+pay_info+"')");

        if(row_ins > 0)
        {
            out.println("booked successfully");
        }
        else
        {
            out.println("already booked");
        }
    }
}
}

```

```

    }
    catch(Exception e) {
        out.close();
    }
    finally {
        out.close();
    }
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

### **bookAppt.jsp: to handle to handle the frontend design of the form**

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        
        <title>JSP Page</title>
        <script src="jquery-1.11.1.min.js"></script>
        <script type="text/javascript">
            var form = $('#f3');
            $( "#calc" ).click(function(){
                if(document.getElementById("date").value ==null ||
document.getElementById("date").value == "")
                {

```

```

        alert("please select date!");
    }
    else if(document.getElementById("date").value !=null ||
document.getElementById("date").value != "")
    {
        if(document.getElementById("a").value ==null ||
document.getElementById("a").value == "")
        {
            alert("please select time!");
        }
        else if(document.getElementById("a").value !=null ||
document.getElementById("a").value != "")
        {
            if(document.getElementById("pid").value ==null ||
document.getElementById("pid").value == "")
            {
                alert("please select pid!");
            }
            else
            {

                $.ajax({
                    type: form.attr('method'),
                    url: form.attr('action'),
                    data: $("#f3").serialize(),
                    success: function (data) {
                        var result=data;
                        $('#bookstatus').attr("value",result);
                    }
                });

            }
        }
    }
    }
    }
);
</script>

```



```

</script>
<script type="text/javascript">
    function book()
    {
        if(document.getElementById("date").value ==null ||
document.getElementById("date").value == "")
        {
            alert("please select date!");
            //add additional check for selecting only future date!
        }
        else if(document.getElementById("date").value !=null ||
document.getElementById("date").value != "")
        {
            if(document.getElementById("a").value ==null ||
document.getElementById("a").value == "")
            {
                alert("please select time!");
            }
            else if(document.getElementById("a").value !=null ||
document.getElementById("a").value != "")
            {
                if(document.getElementById("pid").value ==null ||
document.getElementById("pid").value == "")
                {
                    alert("please select pid!");
                }
            }
            else
            {
                document.forms[0].action ="bookAppt";
                document.forms[0].method = "post";
                document.forms[0].submit();
            }
        }
    }
}
</script>
</head>

```

```

<body>
<br>
<br>
<br>
    <form method="post" name="f3" id="f3" action="bookAppt">
        <table align="center" height="200" width="400"
background="images/tbl_bckgrnd.jpg">
            <tr><td><b> select date :</td><td><input
type="date" expand="true" name="date" id="date"></td></tr>
            <tr><td><b> appointment time:</td><td><select
name="a" name="appt_time" id="a">
                <option name="a" value="10.00"
selected="true">10</option>
                <option name="a"
value="10.30">10.30</option>
                <option name="a"
value="11.00">11</option>
                <option name="a"
value="11.30">11.30</option>
                <option name="a"
value="12.00">12</option>
                <option name="a"
value="12.30">12.30</option>
                <option name="a"
value="1.00">1</option>
                <option name="a"
value="1.30">1.30</option></select></td></tr>
            <tr><td><b>Patient ID:</td><td><input
type="text" name="pid" id="pid"> </td></tr>
            <tr><td><b><span id="calc" style="display:inline-
block; padding: 5px 10px ;background:#3194c6; -webkit-border-radius: 8 px; -moz-border-
radius: 8px;border-radius: 8px; cursor:default">Book</span> </td><td>
                <b><span id="clear" style="
display:inline-block; padding: 5px 10px ;background:#3194c6; -webkit-border-radius: 8 px; -
moz-border-radius: 8px;border-radius: 8px; cursor:default">Clear</span></td></tr>
            <tr><td> Booking Status </td><td><input
type="text" id="bookstatus" name="bookstatus"> </td></tr>
        </table>

```

```

        </form>
    </body>
</html>

```

## Today's appointment page coding-

### tod\_appnt.jsp – to handle the front end coding

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <div>
            
            <title>JSP Page</title>
        </div>
        <script src="jquery-1.11.1.min.js"></script>
        <script>

            var today = new Date();
            var dd = today.getDate();
            var mm = today.getMonth()+1;

            var yyyy = today.getFullYear();
            if(dd<10){
                dd='0'+dd
            }
            if(mm<10){
                mm='0'+mm
            }
            var today = dd+'/'+mm+'/'+yyyy;
            document.getElementById("DATE").value = today;

        </script>
        <script type="text/javascript">

```

```

var form = $('#f7');
    $('#calc').click(function(){
alert("hi");
$.ajax({
    type: form.attr('method'),
    url: form.attr('action'),
    data: $('#f7').serialize(),
    success: function (data) {
        alert("content is loading");
        var result=data;
        alert(result);

        var a = result.split("*");
        alert(a);

        for( i=0; i<a.length;i++)
        {
            if(a[i].indexOf("10.00") >= 0)
            {
                $('#10id').attr("value",a[i+1]);
                $('#10fn').attr("value",a[i+2]);
                $('#10ln').attr("value",a[i+3]);
                continue;
            }
            else if(a[i].indexOf("10.30") >= 0)
            {
                $('#1030id').attr("value",a[i+1]);
                $('#1030fn').attr("value",a[i+2]);
                $('#1030ln').attr("value",a[i+3]);
                continue;
            }
            else if(a[i].indexOf("11.00") >= 0)
            {
                $('#11id').attr("value",a[i+1]);
                $('#11fn').attr("value",a[i+2]);
                $('#11ln').attr("value",a[i+3]);
                continue;
            }
        }
    }
});

```

```

    }
    else if(a[i].indexOf("11.30") >= 0)
    {
        $('#1130id').attr("value",a[i+1]);
        $('#1130fn').attr("value",a[i+2]);
        $('#1130ln').attr("value",a[i+3]);
        continue;
    }
    else if(a[i].indexOf("12.00") >= 0)
    {
        $('#12id').attr("value",a[i+1]);
        $('#12fn').attr("value",a[i+2]);
        $('#12ln').attr("value",a[i+3]);
        continue;
    }
    else if(a[i].indexOf("12.30") >= 0)
    {
        $('#1230id').attr("value",a[i+1]);
        $('#1230fn').attr("value",a[i+2]);
        $('#1230ln').attr("value",a[i+3]);
        continue;
    }
    else if(a[i].indexOf("1.00") >= 0)
    {
        $('#1id').attr("value",a[i+1]);
        $('#1fn').attr("value",a[i+2]);
        $('#1ln').attr("value",a[i+3]);
        continue;
    }
    else if(a[i].indexOf("1.30") >= 0)
    {
        $('#130id').attr("value",a[i+1]);
        $('#130fn').attr("value",a[i+2]);
        $('#130ln').attr("value",a[i+3]);
        continue;
    }
}

```

```

        }
    });

}

);

</script>
</head>
<body>
    <form method="post" name="f7" id="f7" action="TodayAppt">
<table align="center" height="400" width="400" background="images/tbl_bckgrnd.jpg">
    <tr><td>
        <b>today's date: <input type="text" id="DATE" disabled="true"></td>
        <td> <b><span id="calc" style="display:inline-block; padding:5px 10px
;background:#3194c6; -webkit-border-radius: 8px; -moz-border-radius: 8px;border-radius: 8px;
cursor:default">Show Today's appointments</span> </td> </tr>
    <tr><td colspan="2">

        <table border="5" align="center" bgcolor="white" >
        <tr>
            <b>
            <th>Appointment no.</th>
            <th>time</th>
            <th>PID</th>
            <th>First name</th>
            <th>Last name</th>
            <b>
        </tr>

```

```

<tr>
  <td><input type="text" value="1" disabled="true"></td>
  <td><input type="text" value="10" disabled="true"></td>
  <td><input type="text" name="10id" id="10id" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="10fn" id="10fn" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="10ln" id="10ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
  <td><input type="text" value="2" disabled="true"></td>
  <td><input type="text" value="10:30" disabled="true"></td>
  <td><input type="text" name="1030id" id="1030id" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="1030fn" id="1030fn" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="1030ln" id="1030ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
  <td><input type="text" value="3" disabled="true"></td>
  <td><input type="text" value="11" disabled="true"></td>
  <td><input type="text" name="11id" id="11id" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="11fn" id="11fn" value="Not booked"
disabled="true"></td>
  <td><input type="text" name="11ln" id="11ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
  <td><input type="text" value="4" disabled="true"></td>
  <td><input type="text" value="11:30" disabled="true"></td>
  <td><input type="text" name="1130id" id="1130id" value="Not booked"

```

```

disabled="true"></td>
    <td><input type="text" name="1130fn" id="1130fn" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="1130ln" id="1130ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
    <td><input type="text" value="5" disabled="true"></td>
    <td><input type="text" value="12" disabled="true"></td>
    <td><input type="text" name="12id" id="12id" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="12fn" id="12fn" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="12ln" id="12ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
    <td><input type="text" value="6" disabled="true"></td>
    <td><input type="text" value="12:30" disabled="true"></td>
    <td><input type="text" name="1230id" id="1230id" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="1230fn" id="1230fn" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="1230ln" id="1230ln" value="Not booked"
disabled="true"></td>
</tr>

<tr>
    <td><input type="text" value="7" disabled="true"></td>
    <td><input type="text" value="1" disabled="true"></td>
    <td><input type="text" name="1id" id="1id" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="1fn" id="1fn" value="Not booked"
disabled="true"></td>
    <td><input type="text" name="1ln" id="1ln" value="Not booked"
disabled="true"></td>

```



```

        </tr>

        <tr>
            <td><input type="text" value="8" disabled="true"></td>
            <td><input type="text" value="1:30" disabled="true"></td>
            <td><input type="text" name="130id" id="130id" value="Not booked"
disabled="true"></td>
            <td><input type="text" name="130fn" id="130fn" value="Not booked"
disabled="true"></td>
            <td><input type="text" name="130ln" id="130ln" value="Not booked"
disabled="true"></td>
        </tr>

    </table></td></tr>

</table>
</form>
</body>
</html>

```

### **TodayAppt.java: to handle the database transactions**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

```

```

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;

/**
 *
 * @author Grish
 */
public class TodayAppt extends HttpServlet {
    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)

```

```

throws ServletException, IOException {

    PrintWriter out = response.getWriter();
    try {

        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con=DriverManager.getConnection("jdbc:odbc:dsn1216");

        Statement st=con.createStatement();

        ResultSet rs2=st.executeQuery("select * from appointment,patient where appt_dt in
"
        + "(select CONVERT(date,getdate())) and appointment.pid= patient.pid order by
appt_time asc");

        while(rs2.next()== true)
        {
            out.println(rs2.getString("appt_time"));
            out.println("*");
            out.println(rs2.getInt("pid"));
            out.println("*");
            out.println(rs2.getString("fname"));
            out.println("*");
            out.println(rs2.getString("lname"));
            out.println("*");
        }

    }
    catch(Exception e){
        out.close();
    }
    finally {
        out.close();
    }
}

```

```

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}

```

## Visit History page coding-

### visit\_hist.jsp – to handle front end coding

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> <div>
        
    </div>
    <title> Visit History </title>
    <script src="jquery-1.11.1.min.js"></script>
    <script type="text/javascript">
        var form = $('#f12');

        $('#calc').click(function(){
            if(document.getElementById("pid").value == null || document.getElementById("pid").value
            == "")
            {
                alert("Please enter patient id!");
            }
            else
            {
                $.ajax({

```

```

        type: form.attr('method'),
        url: form.attr('action'),
        data: $("#f12").serialize(),
        success: function (data)
        {

            var result=data;
            var a = result.split(".");
            if(a[0].indexOf("Patient not registered") >= 0)
            {
                alert(a[0]);
            }

            else
            {
                alert("inside else");
                $('#fname').attr("value",a[0]);
                $('#mname').attr("value",a[1]);
                $('#lname').attr("value",a[2]);
                $('#bgrp').attr("value",a[3]);
                $('#age').attr("value",a[4]);
                //s$('#prev').val(" ");
                $('#ill').val("");
                $('#med').val("");
            }

        }
    });
}

});

</script>
<script type="text/javascript">
var form = $('#f12');

    $('#calc2').click(function(){
        if(document.getElementById("pid").value == null || document.getElementById("pid").value
        == "")
        {

```

```

        alert("Please enter patient id!");
    }
    else if(document.getElementById("ill").value == null ||
document.getElementById("ill").value == "")
    {
        alert("Please enter illness description!");
    }
    else if(document.getElementById("med").value == null ||
document.getElementById("med").value == "")
    {
        alert("Please enter treatment description!");
    }
    else
    {
        $.ajax({
            type: "post",
            url: "visit_hist2",
            data: $("#f12").serialize(),
            success: function (data)
            {
                var result=data;
                alert(result);
            }
        });
    }
};

</script>

<script type="text/javascript">
var form = $('#f12');

$("#calc3" ).click(function(){
if(document.getElementById("pid").value == null || document.getElementById("pid").value
== "")
{

```

```

        alert("Please enter patient id!");
    }

    else
    {
        $.ajax({
            type: "post",
            url: "vist_hist3",
            data: $("#f12").serialize(),
            success: function (data)
            {
                var result=data;
                alert(result);
                $('#prev').append(result);
            }
        });
    }
</script>

<script>
    var today = new Date();
var dd = today.getDate();
var mm = today.getMonth()+1; //January is 0!

var yyyy = today.getFullYear();
if(dd<10){
    dd='0'+dd
}
if(mm<10){
    mm='0'+mm
}
var today = dd+'/'+mm+'/'+yyyy;
document.getElementById("DATE").value = today;

</script>

```

```

</head>
<body>
  <form name="f12" id="f12" method="post" action="visitHist">
    <table border="2" align="center" width="900" height="500"
background="images/tbl_bckgrnd.jpg">
      <tr><td> Patient ID: <input type="text" name="pid" id="pid">
        <!-- <input type="submit" id="submit" value="search"></td></tr> -->
        <b><span id="calc" style="display:inline-block; padding:5px 10px ;background:#3194c6;
-webkit-border-radius: 8px; -moz-border-radius: 8px;border-radius: 8px;
cursor:default">Search</span></td></tr>
      <tr><td> <table border="2" align="left" width="900" height="500"
background="images/tbl_bckgrnd.jpg">

<tr >

      <td width="500">

        <table width="400" height="400" background="images/tbl_bckgrnd.jpg"
align="center">
          <tr><td> today's date:</td> <td> <input type="text" id="DATE" name="DATE"
disabled="true"><br></td></tr>
          <tr><td>First Name </td> <td><input type="text" disabled="true" value=""
name="fname" id="fname"><br></td></tr>
          <tr><td>Middle Name </td> <td> <input type="text" disabled="true" value=""
name="mname" id="mname"><br></td></tr>
          <tr><td>Last Name </td> <td> <input type="text" disabled="true" value=""
name="lname" id="lname"><br></td></tr>
          <tr><td>Age </td> <td> <input type="text" disabled="true" value="" name="age"
id="age"><br></td></tr>
          <tr><td>Blood Group </td> <td> <input type="text" disabled="true" value=""
name="bgrp" id="bgrp"><br></td></tr>

        </table>
        illness description:<br> <textarea rows="4" cols="40" id="ill" name="ill">

</textarea><br>
        treatment prescribed: <br><textarea rows="4" cols="40" id="med" name="med">

```



```

</textarea>

<br>
    <b><span id="calc2" style = "display:inline-block; padding :5px 10px ;background:#3194c6; -
webkit-border-radius: 8 px; -moz-border-radius: 8px;border-radius: 8px;
cursor:default">Save</span>
        </td>
<td>
    <b><span id="calc3" style = "display:inline-block; padding :5px 10px ;background:#3194c6; -
webkit-border-radius: 8 px; -moz-border-radius: 8px;border-radius: 8px; cursor:default">Show
previous visits</span></b> <br>
previous visits:
<br>
<textarea name="prev" id="prev" rows="30" cols="40" disabled="true" value=""></textarea>
</td>

</tr>

</table></td></tr>
</table>
</form>
</body>
</html>

```

### **visitHist.java: to handle the database transactions**

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package servlets;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;

```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author Grish
 */
public class visitHist extends HttpServlet {

    /**
     * Handles the HTTP
     * <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP
     * <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
    }
}

```

```

String pid=request.getParameter("pid");
String date1 = request.getParameter("DATE");
try {
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection con=DriverManager.getConnection("jdbc:odbc:dsn1216");
    PreparedStatement st=con.prepareStatement("select * from patient where pid =
"+pid);
    ResultSet rs=st.executeQuery();
    if(rs.next()==true)
    {

        out.println(rs.getString("fname"));
        out.println(".");
        out.println(rs.getString("mname"));
        out.println(".");
        out.println(rs.getString("lname"));
        out.println(".");
        out.println(rs.getString("bld_grp"));
        out.println(".");
        out.println(rs.getString("age"));

    }
    else
    {
        out.println("Patient not registered");
    }
}
catch(Exception e)
{

}
finally {
    out.close();
}
}

```

```
/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>
}
```

## 13. Data dictionary

### Database description

- Login Table: table name- “nlogin”

Column Name	Data Type	Constraints	Description
<b>Name</b>	Varchar(20)		Contains the Name of user
<b>Password</b>	Varchar(20)		Contains the Password of User
<b>Urole</b>	Varchar(20)		Contains the Role Information

- Patient Table: table name- “patient”

Column Name	Data Type	Constraints	Description
<b>ID</b>	Int	Primary Key	Contains the ID of Patient
<b>Fname</b>	Varchar(20)		Contains the First Name of Patient
<b>Mname</b>	Varchar(20)		Contains the Middle Name of Patient
<b>Lname</b>	Varchar(20)		Contains the Last Name of Patient
<b>Gender</b>	Varchar(20)		Define Gender Of Patient
<b>Phone</b>	Varchar(10)		Contain The Phone Number
<b>Mobile</b>	Varchar(10)		Contain The Mobile Number
<b>Addr</b>	Varchar(100)		Contain The Address Of Patient
<b>Bld_Grp</b>	Varchar(10)		Defines The Blood Group
<b>Age</b>	Varchar(10)		Contain The Age

- Bill Table: table name- “bill\_info”

Column Name	Data Type	Constraints	Description
<b>Bill_id</b>	Int		Contains The ID Of Bill
<b>PID</b>	Int		Contains The Patient ID
<b>P_name</b>	Varchar(100)		Contains The Name Of Patient
<b>Appt_dt</b>	Varchar(20)		Contain The Date Of Appointment
<b>Appt_no</b>	Varchar(10)		Contain The Number Of Appointment
<b>Payment_amount</b>	Int		Contain The Payment Amount

- Visit History Table: table name- “visits”

Column Name	Data Type	Constraints	Description
<b>PID</b>	Int		Contains the ID of Patient
<b>Appt_dt</b>	Varchar(20)		Contains The Appointment Date
<b>Appn_no</b>	Varchar(20)		Contains The Appointment Number Of Patient
<b>Illness_desc</b>	Varchar(20)		Contain The Description For Illness
<b>Treatment_desc</b>	Varchar(200)		Contain The Description For Treatment

- Appointment Table: table name- “appointment”

Column Name	Data Type	Constraints	Description
<b>Appt_dt</b>	Varchar(20)	Primary Key	Contains The Appointment Date
<b>Appt_time</b>	Varchar(20)	Primary Key	Contains The Appointment Time
<b>ID</b>	Int		Contains The ID Of Patient
<b>Payment_made</b>	Varchar(10)		Payment Made By Patient

- Password Recovery Table: table name- “password\_recovery”

Column Name	Data Type	Constraints	Description
<b>Secret_ques</b>	Varchar(200)		Contains The Secret Question for Security Purpose
<b>Secret_ans</b>	Varchar(20)		Contains The Answers For Security Questions
<b>Sr_no</b>	Int		Contains The Serial Numbers Of Questions

## 14. Future enhancement

Nothing is perfect in this world. Although we have tried our best to present the information efficiently, yet there can be further enhancement in the application.

Future enhancement ideas-

- **Use of E-Pen technology to integrate with the software project**  
Use E-pen can make the job of the doctor easier rather than using keyboard to type when consulting the patient.  
Also, it would give a paper-pen like experience so that even some medical diagrams or tabular data can be saved in the visit history page.
- **Stock inventory integration**  
The system can be further be expanded with more functionality integration like stock inventory- new stock, modify stock etc screens.



## 15. Conclusion

The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

- Automation of the entire system improves the efficiency.
- It provides a friendly graphical user interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permission.
- It effectively overcomes the delay in communications.
- Updating of information becomes so easier.
- System security, data security and reliability are the striking features.
- The system has adequate scope for modification in future if it is necessary.

## 16. References and bibliography

- [www.oracle.com/technetwork/java/](http://www.oracle.com/technetwork/java/)
- <https://netbeans.org/>
- [www.microsoft.com/en-in/download/details.aspx?id=1695](http://www.microsoft.com/en-in/download/details.aspx?id=1695)
- [api.jquery.com/jquery.ajax/](http://api.jquery.com/jquery.ajax/)
- <https://glassfish.java.net/>

## **17. Acknowledgement**

We would like to express our sincere gratitude towards the Information Technology Department of Sathaye College.

After months of hard work, finally we are very happy to present our final year project. The project making was full of new experiences and learning and difficult one too. Though a difficult job it was made simpler by the timely guidance received, which helped us greatly in the completion of our project. But it wouldn't be right to do so without thanking to those who have helped us in converting our thought into reality. So we would like to take full advantage of this opportunity to thank each and every person who has helped us throughout the completion of our project.

We are obliged to our parents & family members who always support us greatly and encouraged us in each and every step. We give our special thanks and sincere gratitude towards the Principal Mrs. Kavita Rege, Head of Department Prof. Arvind Khadye.

I owe my sincere thanks to our Project guide Prof. Sumedha Barve for her Constant support and encouragement without which the successful completion of this project would have been impossible. She has been instrumental for making us concentrate and focus our effort in this project.

We will be forever indebted to all friends who really encouraged us. And last but not the least the entire college staff. Finally we would like to thank each and every individual who was directly or indirectly contributing for this project.

-Thank you