

MAGADH UNIVERSITY

अमृतं तु विद्या
मगध विश्वविद्यालय
MAGADH UNIVERSITY, BODHGAYA

CYBOTECH CAMPUS [456]

An ISO 9001:2008 Certified Institute



PROJECT REPORT

Project Title

Veterinary Management System

**In partial fulfillment of the requirement for the award of
the degree(BCA) for the Session 2015-2018**

SUBMITTED BY:
SHUBHAM KUMAR
Reg. No. -154567005230



मगध विश्वविद्यालय बोधगया
MAGADH UNIVERSITY, BODHGAYA

Certificate of Project

TO WHOM IT MAY CONCERN

This is to certify that the Project entitled “**Veterinary Management System**” submitted by “**SHUBHAM KUMAR, Registration No. 154567005230**” in the partial fulfillment for the award of degree in **Bachelor of Computer Application (BCA)**, Magadh University, is a record of their own work carried out by them under my guidance & supervision for the academic year 2015–2018 and has not been submitted anywhere else except Magadh University [Bodhgaya, Gaya, Bihar].

SIGNATURE OF STUDENT

Name: Shubham Kumar
Course: BCA 3rd Year(MU)
Address: HCN Nagar, Patna-01

SIGNATURE OF GUIDE

Name: Mr. Alok Kumar Singh
Faculty of Computer Science
Cybotech Campus, Patna

DECLARATION

I hereby declare that the work which is being presented in the project entitled “***Veterinary Management System***” as part of the course curriculum of BCA, Magadh University, Bodhgaya; is an authentic record of my own work carried out during the period from August 2015 to June 2018 under the supervision of ***Mr. Alok Kumar Singh***, Technical Manager (IT) Cybotech Campus, Patna.

Shubham Kumar

BCA

Registration No.: 154567005230

ACKNOWLEDGEMENT

I express my deepest sense of gratitude to my supervisor **Mr. Alok Kumar Singh**, Faculty of Computer Science, Cybotech Campus, Patna for his valuable guidance, inspiration and constructive suggestions throughout the period of project work.

Moreover, his optimistic attitude, guidance and appreciation was such as to give impact us to our own thoughts and understandings, making us believe that all that accomplish was my own efforts for which I will ever remain obliged to him.

My family has been of great help and inspiration all along. I thank my family members for bearing me and putting up with my numerous tantrums during my creative pursuit.

I would like to thank my friends and staff members for their help and constructive criticism during development of this work.

Thank you all for your support.

Shubham Kumar

BCA

Registration No.: 154567005230

ABSTRACT

The premier invention of this century is the computer. In a relatively short span of time it has affected many areas of our lives. For example, computers help control our automobiles, act as challenging adversaries in electronic games, make possibly very sophisticated medical diagnostic tools such as Computerized Axial Tomography (CAT) scanner and most important is that computers have had a tremendous impact on the way information is processed in the organizations. Although information has been processed manually throughout history, modern management information system would not be possible without the computer.

Computers have altered completely the structure of business. Large volume of appointment, medical history and record keeping data can be manipulated, organized, stored, retrieved, and used for specific purposes. Medicine and statement can be processed and sent to patient in much less time and with less effort than would be required for the same processes done manually. Veterinary Management System projections are made with greater ease as the computer helps manager to organize and manipulate data. Any combinations of factors and their effect on the overall picture can be tested. Planning and decision making thus become more efficient and accurate.

With more organized and consistent Veterinary Management Systems, data management becomes a more efficient task. Now in India and also in other country are being used computerized Veterinary Management System in clinic and hospital. Modern Veterinary Management System would be impossible without the computer.



VETERINARY MANAGEMENT SYSTEM

CONTENTS

<u>TITLE</u>	<u>Page</u>
1. Introduction	1
1.1 Advantages of Computerized System	3
2. Object of Project	4
3. Requirement specification	6
3.1. Software and Hardware Requirement	7
3.2. Front End	8
3.3. Back End	11
4. Problem Statement	14
5. Complete Structure of Program	15
5.1. Requirement Analysis	15
5.2. System Analysis	21
5.3. Software Engineering Paradigm	23
5.4. DFD	25
5.5. E-R Diagram	29
6. Database Structure	31
7. Table Creation	37
8. Module Description	42
9. Designing	49
10. Coding	65
11. Testing Plan	135
12. Future Scope of Project	141
13. Conclusion	144
14. Limitation	145
15. Bibliography	146

INTRODUCTION

In broadest sense, a system is a set of components that interact to accomplish or to achieve some predefined goal. The word system is derived from the Greek word “SYSTEMA”. These systems are the independent unit which works in an organized way to achieve some specific objective.

This is the project on the “**Veterinary Management System**”. It is the management of the vet clinic having different kinds of works such as Joining/Relieving of Employee and Doctor, Appointment/Registration entry of Patient, Medicine management, Fee collection and also the ADDITIONAL FACILITIES like issuing of Creation of Patient Record for these problem that is being checked in clinic keeping in view of all the basic area that concerned for a Clinic.

All the facilities provided by the Veterinary Management System are the constructed measuring all about the various parameters taken into consideration according to the needs and the requirement for the user. The options selected here is an ideal working of any clinic where all the issuing and recovering facilities of the areas of concern are to be maintained in the database of the Veterinary Management System.

Veterinary Management System workings are of wide range. This is an attractive and user friendly system, which can reduce the work load and provide specified jobs. It is wholly a computerized management system in which all the jobs performed is being done and managed by computer.

It is capable of holding a number of records entries, maintaining it through, the data, in a database. This makes the task very easy, less time consuming and less error prone. Thus the accuracy of work in system is relatively very good.

It capable of doing the jobs performed in a computerized clinic system like Add New Patient, Edit Existing Patient record, Search for Patient, Add Doctor, Add Receptionist, Add Employee, Creation of Patient Treatment Record for

All which are created by Vet Clinic, Management of Medicine and Vaccination, Service Search, Service Maintenance.

Computerization of the Veterinary management system solves most of the problem and act as a very efficient tool in the enlistment of Clinic. The accuracy of the system is most accepted one. The style of working is extremely fast, and a very few number of workers are needed to operate this all. This all prove the significance of this system.

This project is made and designed for the efficient use by the users working with this system solves most of the problem faced during the course of veterinary management. The way this project coordinates with the user that it seems to be very friendly and convenient for him. This project is invariably fulfilling the needs of the operator and for the development of the Vet clinic. This all features and the act of response of this project makes this project to be different among the other.

ADVANTAGES OF COMPUTERIZED VETERINARY MANAGEMENT SYSTEM

We all know that the system which is time consuming and error prone. Especially in my project computer system is inevitable. It is calculated field where manual System consumes too much of time where as it is a matter of seconds with a computer system.

To deal with errors a programmer with at the time of creating software takes into account substantial amount of all kinds of unforeseen and unprecedented Possibilities. So that any one crops up while operating the program the computer already knows how to do it deal with it. Where as in a manual system it will take the necessary time to solve the problem and today's customer can offer anything but 'TIME'. So to make business success the system chosen for it is successful.

- Computerized system is user-friendly
- The computerized system enhanced the working of the work where everything is done manually.
- Records can be added and modified easily
- Operator doesn't need to know the coding. He/she only has to know, how to operate System.
- Details can be easily viewed.
- Validation is done at every step to the project.
- Operators can see the over-all records.
- Security of system enhances the quality of system.

OBJECTIVE

When the patient will visit in clinic, they will ask to provide their information and type or problems. This system consists of different specialized doctors so patient have multiple option to select doctor as their input during patient appointment. Patient will get an appointment letter. Each doctor having their unique doctor_id to differentiate their specialization. This clinic maintains a brief information of all the doctors involved in the system. The doctor will prescribe medicines using the type of problem patient is facing.

If the same patient will visit the clinic again then we provide facility that patient doesn't have to go through the whole registration process as clinic maintain old patient record also. So all details of the patient along with their previous check-up and test details will display on the screen by entering their patient id only.

There is provision of registration through telephone, by giving all required details of the patient.

It will provide the various advantages, such as

- Eliminate or reduce to minimum the manual work
- Quick accessing of data
- Time saving
- Instant services
- Take any detail of any type at any time
- A viable application.
- A robust application.
- A user friendly interface.
- Help users in searching and feeding information in application.
- Provide secured environment for secured data access wherever necessary.

- Retrieval of data from database using easy button and interface in a very fast and efficient manner.
- Updating of any changes in easy way.

Information about the Patient who took appointment in the clinic must be maintained in the data base. The main issue of the Veterinary Management System is to search for the information of Patient, Doctor, Receptionist and Employee, to modify the wrong or changed details of the Patient's and Doctor, and also make entry for all the Patient who check-up in Clinic. All the changes which are performed by the operator must be updated in the main data base.

These make the clinic working efficiently by giving the appropriate information of the Patient that whether he/she has been check or had check-up. All this makes the software in soft going and maintaining the data or information.

Project category

The Project category is desktop based application developed in java with Oracle 11g Language used is java.

REQUIREMENT SPECIFICATION

Software Engineering Concepts

Software engineering is the organized approach to the development, maintenance and retirement of software.

Obtaining economical software that is trustworthy and works professionally on real system by the organization and use of sound principles of engineering is called software engineering. It is the application of science and mathematics by which the capabilities of computer equipment are made useful to humans via computer programs procedures, and associated documentation.

The main purpose of the software engineering Software Maintenance, Correction, Adaption and Enhancement.

Software and Hardware Requirement

LANGUAGE AND SOFTWARE TOOL USED:

- Front End: java
- Operating System: Windows
- Back End: Oracle 11g
- IDE Used: Eclipse

MINIMUM HARDWARE REQUIRED:

- Intel dual core processor or more
- 256 MB RAM
- 40 GB HARD DISK.

MINIMUM SOFTWARE REQUIREMENT:

- O/S – WINDOWS XP.
- FRONT END TOOLS—JAVA
- BACK END TOOLS—ORACLE 11G

Front End: -

Object-orientation has become the predominant paradigm for virtually all modern software development. Java is one of the most popular languages in the field of software programming. The reason behind its widely acceptance in software programming is its large number of important and robust features. Java can be used to develop wide variety of dynamic, fully secure and platform independent applications. Java is a high-level, third generation programming language like C, FORTRAN, Perl and many others. Java can be used to write computer applications that crunch numbers, process words, play games, store data or do any thousands of other things computer software can do. The special thing about java is that it allows us to write special programs called applets that can be downloaded from the internet and played safely within a web browser.

Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere". Java is currently one of the most popular programming languages in use, and is widely used from application software to web applications.

Java 9.0 is the pure object-oriented language available today. In my project Java 9.0 is used as Front End. Java 9.0 was conceived by James Gosling, Chief Programmer, Patrick Naughton and others of Sun Microsystems in 1991, firstly known as "Oak" renamed as "Java 9.0" in 1995. Java 9.0 is an object-oriented programming language. Within a short span, Java 9.0 became spectacularly popular as an easy but strong language especially related to web or internet related applications. Java 9.0's motto is "Write Once, Run Anywhere".

Reasons for choosing Java 9.0 as Front End

Today the world is moving around the Object-Oriented technique. Java 9.0 is a completely object-oriented programming language and it is most popular language in the fast growing world.

Java 9.0 has various features, some of these which initiate me to use Java 9.0 as Front End for my project are as follows: -

- ◆ Simple.
- ◆ Object Oriented.
- ◆ Platform Independent.
- ◆ Robust.
- ◆ Secure.
- ◆ Portable.
- ◆ Dynamic.
- ◆ Interpreted.
- ◆ SIMPLE

Java 9.0 is easy to write and understand. In Java we don't have to work with pointers and doesn't need to manage memory explicitly. Java eliminated the use of pointers for security reasons. Its virtual machine is able to handle the memory management and so removes the occupied memory automatically when it is no longer referenced. We can now concentrate on the required application logic rather than wasting time and logic with these managements.

◆ Object-Oriented Technique: -

With object-oriented programming, the motto is “divide and conquer”, i.e. a program is divided into easily conceptualized parts. There are several languages like C and C++ which are not pure object oriented but Java is fully object oriented language which follows all the rules of OOPs like Inheritance, Encapsulation, Polymorphism etc. and everything can be treated in the form of objects.

To be an Object Oriented language, any language must follow at least the four characteristics.

♦ **Inheritance: -**

It is the process of creating the new classes and using the behaviour of the existing classes by extending them just to reuse the existing code and adding the additional features as needed.

♦ **Encapsulation: -**

It is the mechanism of combining the information and providing the abstraction.

♦ **Polymorphism: -**

As the name suggest one name multiple form, Polymorphism is the way of providing the different functionality by the functions having the same name based on the signatures of the methods.

♦ **Platform Independent: -**

The concept of “Write Once, Run Anywhere” i.e., WORA is one of the important key feature of java 9.0 language that makes java 9.0 a Platform Independent Language and by this feature Java 9.0 is the most powerful language.

♦ **Robust: -**

Programs are easy to write and debug because java 9.0 does not use the pointers explicitly. Java 9.0 has the strong memory allocation and automatic garbage collection mechanism as compare to other programming languages. All of the above features make the java 9.0 language robust and simple.

♦ **Secure: -**

As Java 9.0 uses OOps technique so it follows some OOps major feature that are Encapsulation, Data abstraction, Polymorphism, Inheritance etc. and Java 9.0 is a Compiler Interpreter based language, so its security is checked at all levels of program. And the Java 9.0 programs are made in its own environment i.e. JRE and its JVM can strictly monitor the program. So, Java 9.0 is a highly secured language.

♦ **Portable: -**

The feature Write-once-run-anywhere makes the java language portable provided that the system must have interpreter for the JVM. Java also has the standard data size irrespective of operating system or the processor. These features make the java as a portable language.

♦ **Dynamic: -**

While executing the java program the user can get the required files dynamically from a local drive or from computer thousands of miles away from the user just by connecting with the Internet.

♦ **Interpreted: -**

We all know that Java is an interpreted language as well. With an interpreted language such as Java, programs run directly from the source code. The interpreter program reads the source code and translates it on the fly into computations. Thus, Java as an interpreted language depends on an interpreter program.

The versatility of being platform independent makes Java to outshine from other languages. The source code to be written and distributed in platform independent. Another advantage of Java as an interpreted language is its error debugging quality. Due to this any error occurring in the program gets traced. This is how it is different to work with Java.

Back End: -

The Back End uses for my Project is Oracle 11g which is a Relational Database Management System. A Relational data model was invented by E.F. Coded and is based on one simple concept i.e. Relation. A relation is a combination of attributes and tuples.

RDBMS has three major parts: –

- Data that is presented as the form of relation in the database.

- Operators for manipulating relation in database.
- Integrity rules on relations.
- We can define Oracle as a multi-user relational database. But a database is called as relational database if it satisfies Dr. E.F. CODD's Rules. The CODD's rules for being an RDBMS are as follows:
 - Information Representation.
 - Guaranteed Access.
 - Systematic treatment of NULL value.
 - Data Description Rule.
 - Comprehensive data sub-language.
 - View updating.
 - High-level update, insert, delete.
 - Physical data independency.
 - Logical data independency.
 - The distribution rule.
 - Non subversion rule.
 - Integrity rule.

Reasons for choosing Oracle 11g as Back End

Oracle 11g has enormous features some of these are as follows for which I had chosen it for back end: -

- ✓ Define a database.
- ✓ Ease of Management
- ✓ Query from database.
- ✓ Flashback available at the row, transaction, table or database level.
- ✓ Insert, delete and update data in database as per requirement.
- ✓ Secure data from public access.
- ✓ Export and Import data.
- ✓ SQL compatibility.

- ✓ Efficient multi – user support and consistency.
- ✓ SQL is a structured query language that we use to communicate with ORACLE. It consists of a set of English words like SELECT, CREATE etc.

The standard set of SQL command fall into the following category: -

- Queries using select clause.
- Data Definition Language (DDL) commands which are for creating database.
- Data manipulation language (DML) commands for modifying database.
- Data control language (DCL) commands for controlling Database.

The idea of Oracle 11g is that the database will be able to dynamically "requisition" resources from the grid to meet levels of peak demand. Other enhancements for Oracle 11g are aimed at reducing costs and improving the quality of service by making the database easier to deploy and manage including enhancements to backup and recovery and data warehousing.

PROBLEM STATEMENT

We all know that the world now is high tech and demands speed, everything around is moving with fast pace. So, how can we ever think of a system that is slow and prone to error?

Especially, in fields where calculation and data handling play a major role, procedure can have affected the progress of the business because today's customer can offer anything but time.

A good example can be our project "Veterinary Management System" which evolves calculation as well as well data handling in every step.

Which have observed some of the most common errors and disadvantages of the system, which are listed below: -

- The universal disadvantage of a system is that it is a slow procedure
- Errors are often found in oral/written calculation
- In Every field where every calculation is involved is a procedure
- Takes plenty of time to reach to the output, hence it will provide curse in the business.
- There is lot of duplicate work and chances of mistake
- When there are any changes, they need to update each and every register.
- There is no security in current system; anybody can access any report and sensitive data.

Due to above disadvantages we can also conclude that it is an expensive method. The above stated disadvantages are the general one faced in all fields where manual system is being used.

COMPLETE STRUCTURE OF PROGRAM

Requirement Analysis

Identification of Need:

“Veterinary Management System” is made for one of the renowned vet clinic in Moti Nagar (New Delhi). Presently the whole system works manually but they want to automate it completely.

I have to first understand the problem and identify the end users involved in clinic that uses the software then produce a requirement documentation based on understanding of the problem: -

Client: Director of Vet Clinic.

End Users: Director, Doctor, Employee & Receptionist.

Now we begin to study the current system. After discussing with the Director, Doctor, Receptionist & other Employees of Vet Clinic, I found that the system operates as follows. There are four modules and its each department keeps their records manually and on a sheet of paper. They keep details of patient, doctor, receptionist and employees on a sheet of paper. They maintain attendance record of each and every Staff members and also they maintain their address, etc. of their staffs. Vet Clinic also keep records of Patient appointment/disease details, issue medicine & vaccination details, paid or dues fee details, etc.

However, it is not possible to share the data from multiple system in multi user environment, there is lot of duplicate work, and chances of mistakes.

When there make any changes, they need to update each and every register. There is no security; anybody can access any report and sensitive data. So, they want computerization of their entire system in their defined paradigm. Decisions were taken regarding the automated system should do and what the new environment should be: -

- The format of each entry should be similar to the one currently being used.
- Entries might have errors, so the system should be able to check for errors.
- The user will be able to generate reports easily and fast.
- Information about any departments will be kept in a database.
- Software is fully password protected and secure.

Software Requirement Specification (SRS)

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex, it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project us initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Purpose

The purpose of this document is to describe all external requirements for mobile task manager. It also describes the interfaces for the system.

FEASIBILITY ANALYSIS

The process followed in making a determination is called Feasibility Study. This type of study clearly specifies that, the project should be taken up or not. It is an evaluation of a proposal designed to determine the difficulty in carrying out a designated task. Generally, a feasibility study precedes technical development and project implementation. The objectives of a feasibility study are to find out if an information system project can be done (...is it possible? is it justified?) and to suggest possible alternative solutions. In other words, a feasibility study is an evaluation or analysis of the potential impact of a proposed project.

A feasibility study should provide management with enough information to decide:

- whether the project can be done
- whether the final product will benefit its intended users
- what are the alternatives among which a solution will be chosen (during subsequent phases)?
- is there a preferred alternative?

After a feasibility study, management makes a go/no go decision.

Types of feasibility study

There are various types of feasibility study but in my project I used the following feasibility studies: -

- Operational feasibility
- Technical Feasibility
- Economical Feasibility
- Schedule Feasibility

Operational Feasibility: -

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. It Define the urgency of the problem and the acceptability of any solution; If the system is developed, will it be used? Includes people-oriented and social issues: internal issues, such as manpower problems, labour objections, manager resistance, organizational conflicts and policies.

In nutshell we can say that it has following operational features: -

- ❖ Efficient tractability.
- ❖ Various reports can be generated.
- ❖ Fully protected by password and user name for unauthorized access

Technical Feasibility: -

The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Fields, Programs, and Procedures. This can be quantified in terms of volumes of data, trends, frequency of updating, etc. in order to estimate whether the new system will perform adequately or not. Technical feasibility is carried out to determine whether the company has the capability, in terms of software, hardware, personnel and expertise, to handle the completion of the project.

Economical Feasibility: -

The bottom line in many projects is economic feasibility. During the early phases of the project, economic feasibility analysis amounts to little more than judging whether the possible benefits of solving the problem are worthwhile. As soon as specific requirements and solutions have been Identified, the analyst can weigh the costs and benefits of each alternative.

Economic feasibility is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. It is important to identify cost and benefit factors like development costs, operating costs. This is an analysis of the costs to be incurred in the system and the benefits derivable out of the system.

Once the feasibility study is done then the project is approved or disapproved according to the results of the study. If the project seems feasible and desirable, then the project is finally approved otherwise no further work is done on it.

Schedule Feasibility: -

A project will fail if it takes too long to be completed before it is useful. Typically, this means estimating how long the system will take to develop, and if it can be completed in a given time period using some methods like payback period. Schedule feasibility is a measure of how reasonable the project timetable is. Given our technical expertise, are the project deadlines reasonable? Some projects are initiated with specific deadlines. We need to determine whether the deadlines are mandatory or desirable.

SYSTEM ANALYSIS

EXISTING SYSTEM:

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- what all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system.

During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analysed thoroughly in accordance with the needs.

System analysis can be categorized into four parts.

- ✓ System planning and initial investigation
- ✓ Information Gathering
- ✓ Applying analysis tools for structured analysis
- ✓ Feasibility study
- ✓ Cost/ Benefit analysis

In the existing system all the jobs of the library management is done manually. This is very easy to the operators who want to handle thousands of books and many students in a day. The currents system is that an operator

wants to keep the physical records of the books in his database and a separate record for the students.

PROPOSED SYSTEM:

The proposed system is very useful for the operators and passengers. This avoids the overheads for the operators. They can minimize the working stress and can keep essential documents related to the books and the students as a softcopy. The advantage of the proposed system is the reduction in the cost of the office equipment's and the transaction is done quickly. Any operator can answer if any seats for a particular route in a particular day are available or not.

Our proposed system has several advantages

- User friendly interface
- Fast access to database
- Less error
- More Storage Capacity
- Search facility
- Look and Feel Environment
- Quick transaction

SOFTWARE ENGINEERING PARADIGM

Process, method and tools, which are used to solve actual problem in an industry setting, a software engineer or a team of engineers must follow a development strategy. This strategy is often referred to as a software engineer paradigm. There are many types of these paradigms, but one of them is described below which is used in this software:-

Water Fall Model: -

This is the simplest software process model. In this model, the phases are organized in a linear order. The sequence of activities that Water Fall model contains is feasibility analysis, requirement analysis and project planning, system design, detailed design, coding and unit testing, system integration and testing.

At the end of each phase, validation and some certification of the intermediate work product is made to ensure that the output of a phase is consistent with its input, and the output of a phase is consistent with the overall requirements of the system.

The Water Fall model is most widely used process model. It is well suited for routine type of projects where all the requirements are well understood.

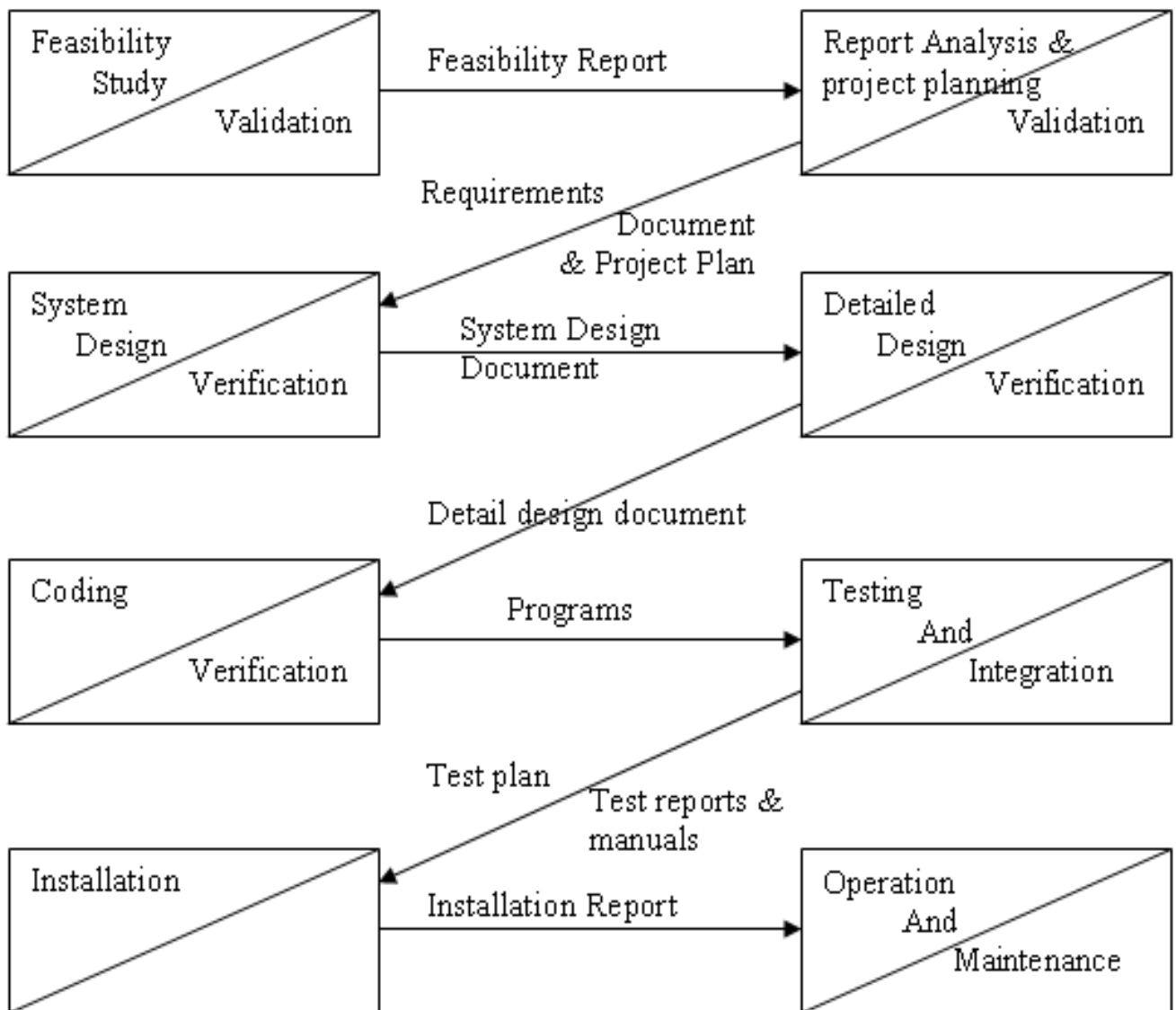


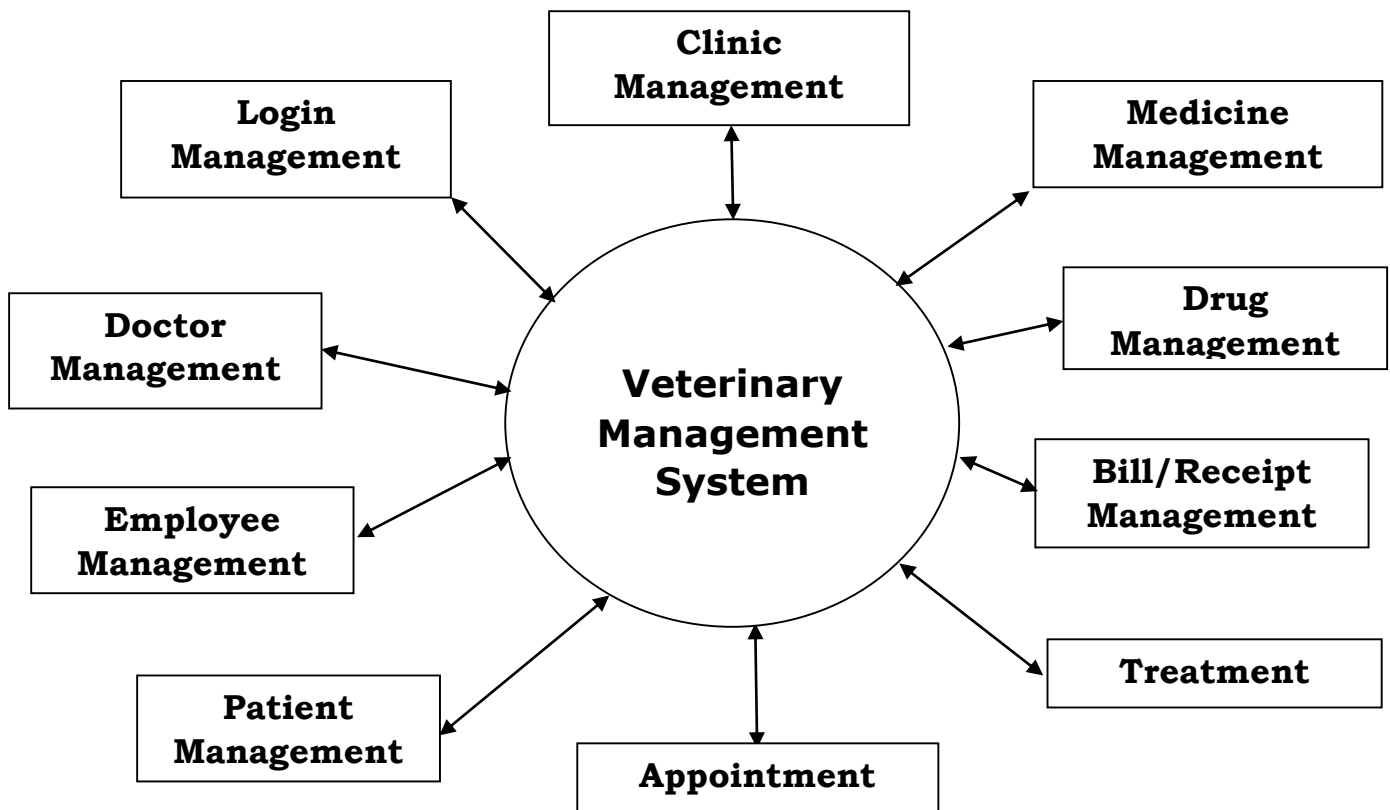
Fig: Water Fall Model

DATA FLOW DIAGRAM

This is the most important step in the design phase. DFD helps us to understand a system and can be effectively used during design as well as analysis phase. A DFD shows the flow of data through a system. It views a system as a function that transforms the inputs into desired output.

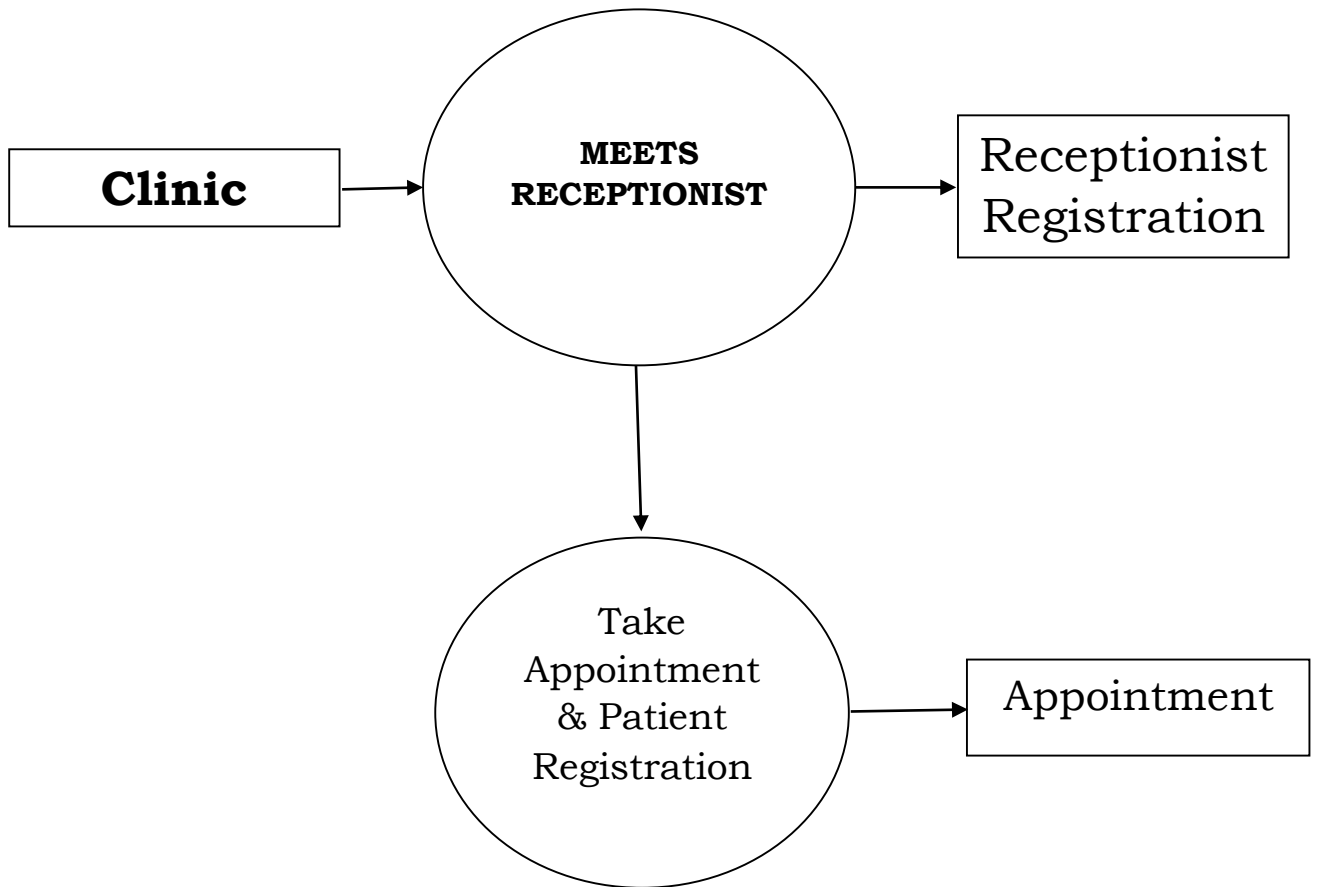
Symbols used in DFDs:

- Named circle- to represent processes.
- Data flows- Labelled arrows entering or leaving processes.
- Named Rectangles- represents the external entities.
- An open rectangle- represents a data store.

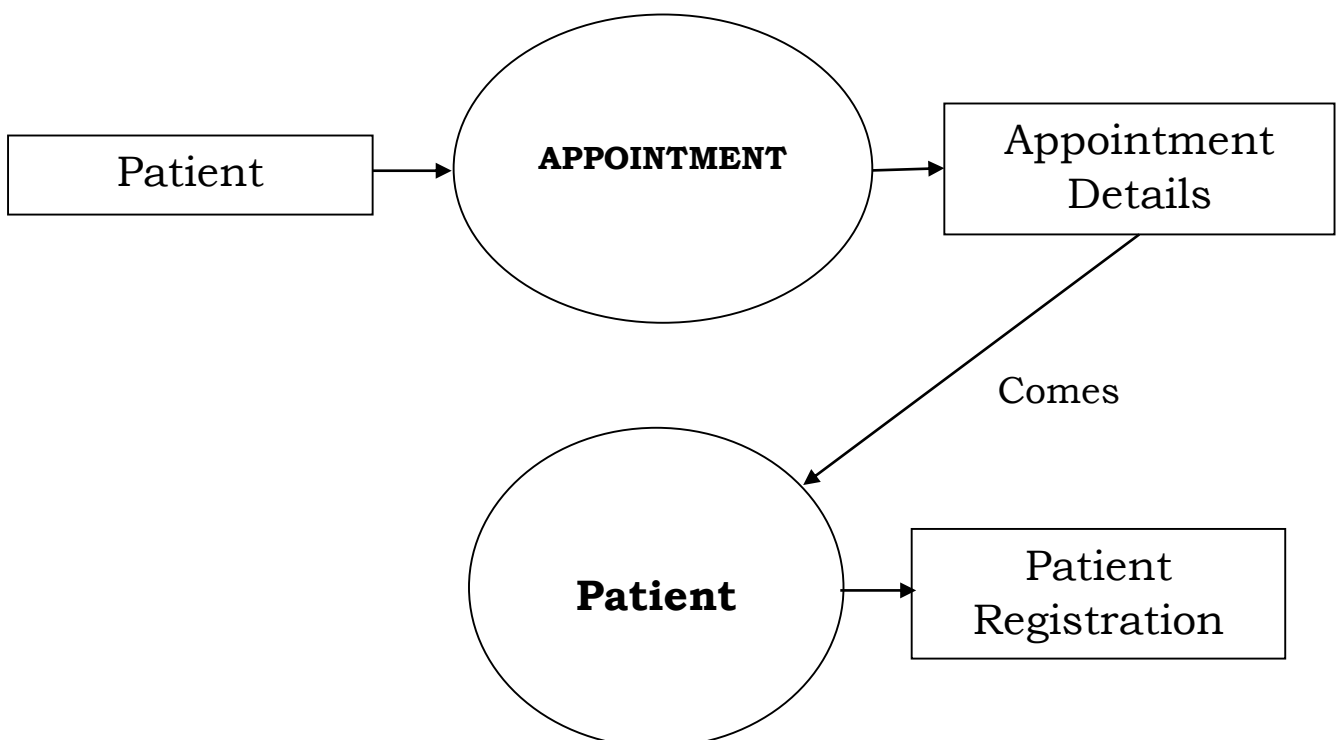


Top Level DFD:

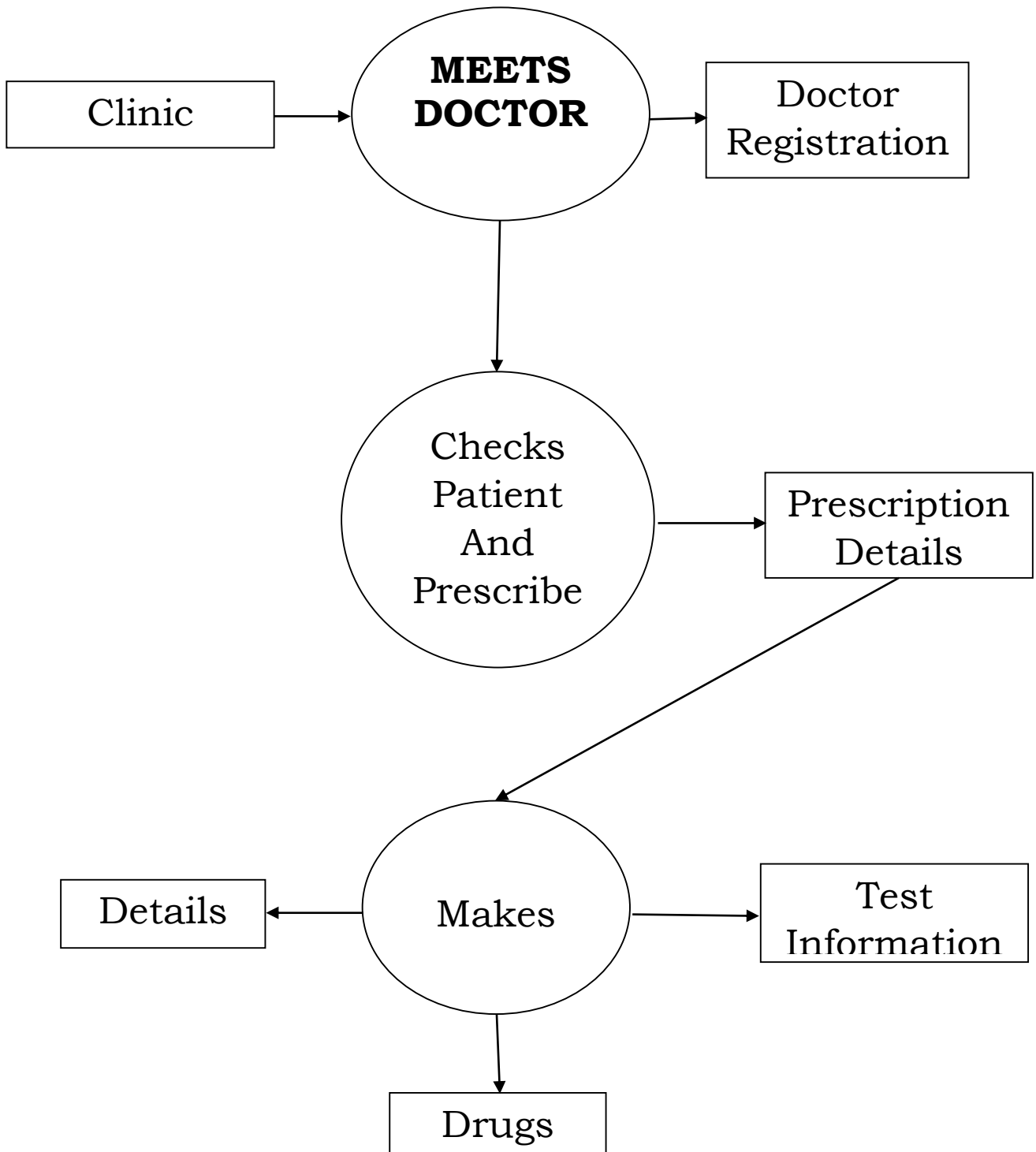
1.0. Receptionist Details: -



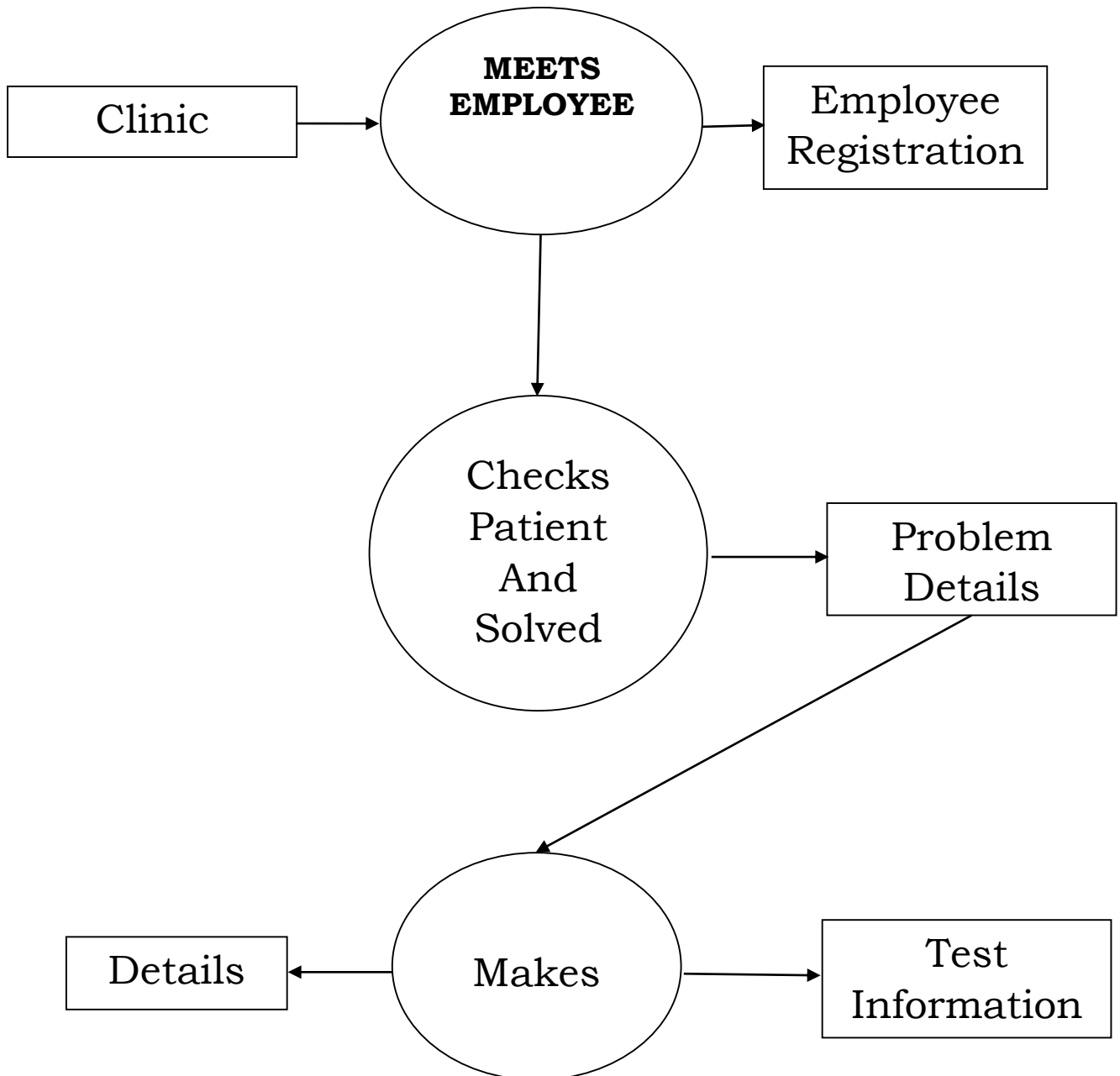
2.0. Patient Details: -



3.0. Doctor Details: -



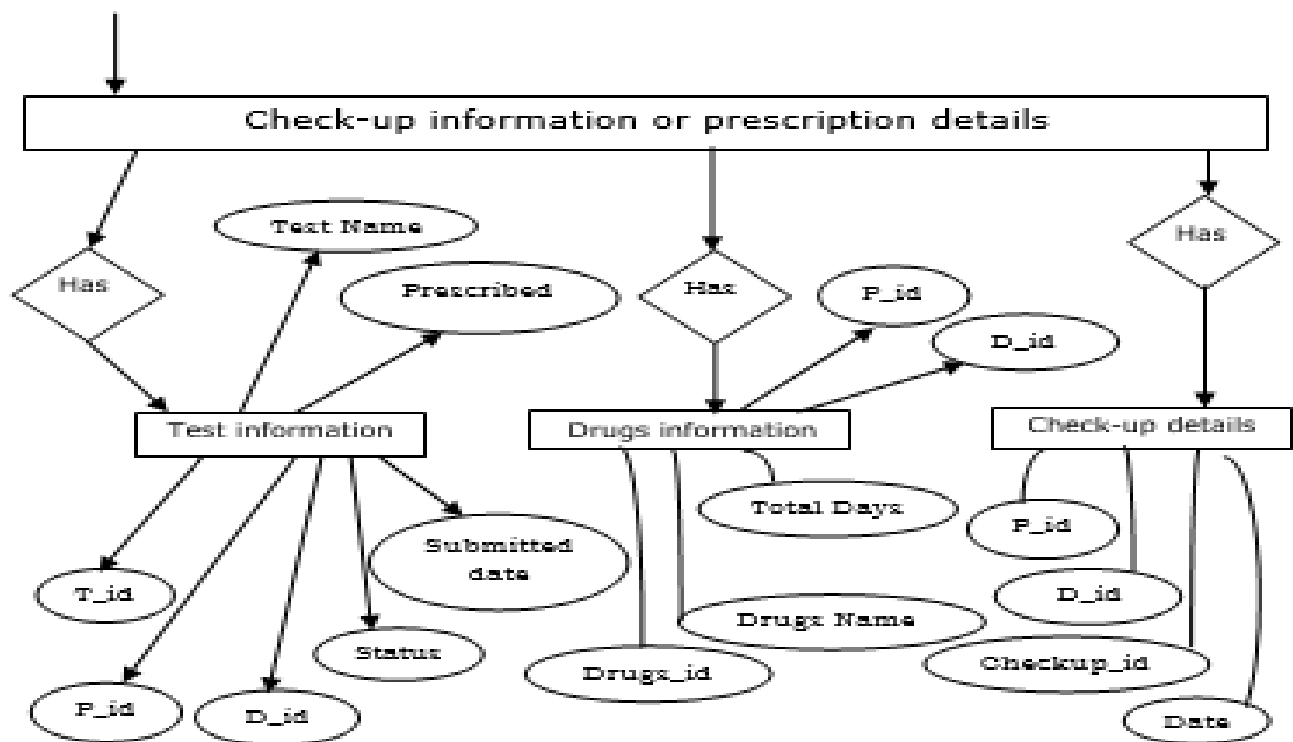
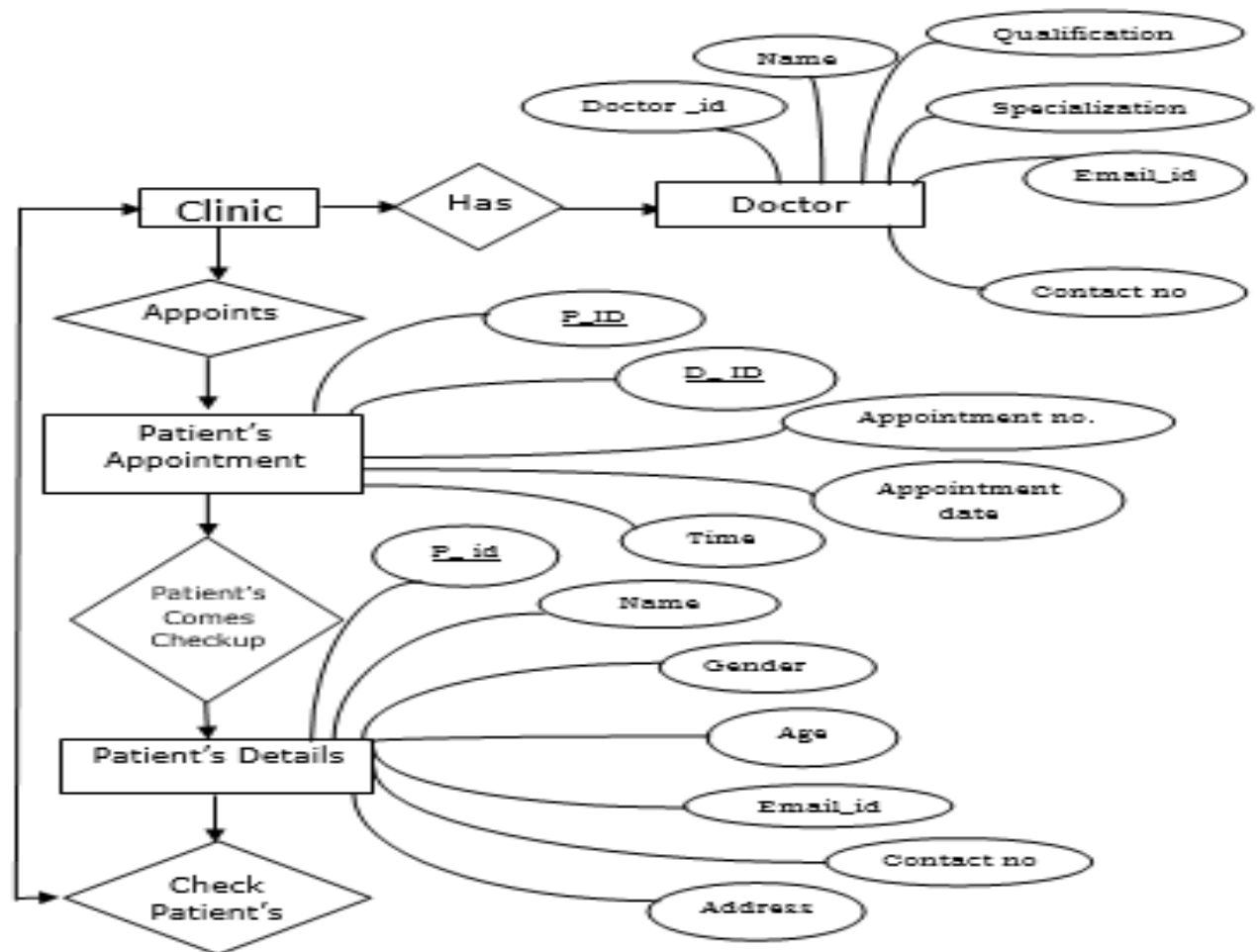
4.0. Employee Details: -



ENTITY RELATIONSHIP DIAGRAM

The E-R-Diagrams graphically represent the conceptual schema, in which the structure of the entity types, the relationship between entities, and constraints on relationships, are all represented.

I am illustrating the E-R-Diagram for the “School Automation System”. In this system, there are nine entity types: School, Student, Employee, Library, Books, Class, Payment, Fee receipt & Demand bill and Department (an entity type is represented by a rectangle). An attributes of an entity is shown as an oval attached to its entity type. Relationship types are shown as diamonds.



DATA DICTIONARY (DATABASE STRUCTURE)

Table: ADMIN_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
FULL NAME	VARCHAR2	30	
USER ID	VARCHAR2	20	
SECURITY_QUESTION	VARCHAR2	50	
ANSWER	VARCHAR2	20	
PASSWORD	VARCHAR2	20	

Table: RECEPTIONIST_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	20	
NAME	VARCHAR2	30	
DOB	DATE		
GENDER	VARCHAR2	20	
MOBILE_NO	VARCHAR2	13	
EMAILID	VARCHAR2	50	
ADDRESS	VARCHAR2	100	
USERID	VARCHAR2	30	
SECURITY_QUESTION	VARCHAR2	50	
ANSWER	VARCHAR2	30	
PASSWORD	VARCHAR2	30	

Table: DOCTOR_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	20	
NAME	VARCHAR2	30	
DOB	DATE		
GENDER	VARCHAR2	20	
QUALIFICATION	VARCHAR2	30	
SPECIALIZATION	VARCHAR2	30	
MOBILE_NO	VARCHAR2	13	
EMAILID	VARCHAR2	50	
ADDRESS	VARCHAR2	100	
USERID	VARCHAR2	30	
SECURITY_QUESTION	VARCHAR2	50	
ANSWER	VARCHAR2	30	
PASSWORD	VARCHAR2	30	

Table: MEDICINE_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	20	
NAME	VARCHAR2	40	
DRUGS	VARCHAR2	50	
TYPE	VARCHAR2	50	
COMPANY	VARCHAR2	30	

Table: EMPLOYEE_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	20	
NAME	VARCHAR2	30	
DOB	DATE		
GENDER	VARCHAR2	20	
QUALIFICATION	VARCHAR2	30	
POST	VARCHAR2	30	
MOBILE_NO	VARCHAR2	13	
EMAILID	VARCHAR2	50	
ADDRESS	VARCHAR2	100	
USERID	VARCHAR2	30	
SECURITY_QUESTION	VARCHAR2	50	
ANSWER	VARCHAR2	30	
PASSWORD	VARCHAR2	30	

Table: DRUGS_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
DRUGS_ID	VARCHAR2	20	
DRUGS_NAME	VARCHAR2	50	
DESCRIPTION	VARCHAR2	100	

Table: ANIMAL_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
REG_DATE	DATE		
REG_NO	VARCHAR2	20	
OWNER_NAME	VARCHAR2	30	
PET_NAME	VARCHAR2	20	
GENDER	VARCHAR2	20	
AGE	VARCHAR2	10	
SPECIES	VARCHAR2	20	
BREED	VARCHAR2	30	
HEIGHT	VARCHAR2	20	
WEIGHT	VARCHAR2	20	
MOBILE_NO	VARCHAR2	13	
EMAILID	VARCHAR2	50	
STATE	VARCHAR2	20	
ADDRESS	VARCHAR2	100	
CURRENT_CONDITION	VARCHAR2	100	
MEDICAL_HISTORY	VARCHAR2	250	

Table: VACCINATION_DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	30	
VACCINATION	VARCHAR2	50	

Table: APPOINTMENT

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
REG_NO	VARCHAR2	20	
CURDATE	DATE		
CURTIME	VARCHAR2	20	
APPOINTMENT_NO	VARCHAR2	40	
OWNER_NAME	VARCHAR2	30	
PET_NAME	VARCHAR2	20	
MOBILE_NO	VARCHAR2	20	
GENDER	VARCHAR2	20	
AGE	VARCHAR2	10	
HEIGHT	VARCHAR2	30	
WEIGHT	VARCHAR2	20	
APPOINTMENT_DATE	DATE	20	
APPOINTMENT_TIME	VARCHAR2	30	
PROBLEM	VARCHAR2	30	
SERVICES	VARCHAR2	70	

Table: SERVICE DETAILS

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
ID	VARCHAR2	20	
SERVICE	VARCHAR2	40	
FEE	VARCHAR2	50	

Table: DOC_TREATMENT

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
CURDATE	DATE		
PRESC_NO	VARCHAR2	20	
APMNT_NO	VARCHAR2	30	
REG_NO	VARCHAR2	30	
DOCNAME	VARCHAR2	50	
OWNER_NAME	VARCHAR2	50	
PET_NAME	VARCHAR2	30	
SERVICE	VARCHAR2	30	
PRESCRIPTION	VARCHAR2	70	

Table: EMP_TREATMENT

FIELD NAME	DATA TYPE	SIZE	CONSTRAINTS
CURDATE	DATE		
PRESC_NO	VARCHAR2	20	
APMNT_NO	VARCHAR2	30	
REG_NO	VARCHAR2	30	
EMPNAME	VARCHAR2	50	
OWNER_NAME	VARCHAR2	50	
PET_NAME	VARCHAR2	30	
SERVICE	VARCHAR2	30	
STATUS	VARCHAR2	30	

TABLE CREATION INSIDE DATABASE QUERY

CREATE TABLE ADMIN_DETAILS

```
(  
    FULLNAME VARCHAR2(30),  
    USERID VARCHAR2(20),  
    SECURITYQUESTION VARCHAR2(50),  
    ANSWER VARCHAR2(20),  
    PASSWORD VARCHAR2(20)  
);
```

CREATE TABLE RECEPTIONIST_DETAILS

```
(  
    ID VARCHAR2(20),  
    NAME VARCHAR2(30),  
    DOB DATE,  
    GENDER VARCHAR2(20),  
    MOBILE_NO VARCHAR2(13),  
    EMAILID VARCHAR2(50),  
    ADDRESS VARCHAR2(100),  
    USERID VARCHAR2(30),  
    SECURITY_QUESTION VARCHAR2(50),  
    ANSWER VARCHAR2(30),  
    PASSWORD VARCHAR2(30)  
);
```

CREATE TABLE DOCTOR_DETAILS

```
(  
    ID VARCHAR2(20),  
    NAME VARCHAR2(30),  
    DOB DATE,  
    GENDER VARCHAR2(20),  
    QUALIFICATION VARCHAR2(30),
```

```
SPECIALIZATION VARCHAR2(30),  
MOBILE_NO VARCHAR2(13),  
EMAILID VARCHAR2(50),  
ADDRESS VARCHAR2(100),  
USERID VARCHAR2(30),  
SECURITY_QUESTION VARCHAR2(30),  
ANSWER VARCHAR2(30),  
PASSWORD VARCHAR2(30)  
);
```

```
CREATE TABLE SERVICE_DETAILS  
(  
    ID VARCHAR2(30),  
    SERVICE VARCHAR2(80),  
    FEE VARCHAR2(50)  
);
```

```
CREATE TABLE EMPLOYEE_DETAILS  
(  
    ID VARCHAR2(20),  
    NAME VARCHAR2(30),  
    DOB DATE,  
    GENDER VARCHAR2(20),  
    QUALIFICATION VARCHAR2(30),  
    POST VARCHAR2(30),  
    MOBILE_NO VARCHAR2(13),  
    EMAILID VARCHAR2(50),  
    ADDRESS VARCHAR2(100),  
    USERID VARCHAR2(30),  
    SECURITY_QUESTION VARCHAR2(30),  
    ANSWER VARCHAR2(30),  
    PASSWORD VARCHAR2(30)  
);
```

```
CREATE TABLE MEDICINE_DETAILS
```

```
(  
    ID VARCHAR2(30),  
    NAME VARCHAR2(50),  
    DRUGS VARCHAR2(500),  
    TYPE VARCHAR2(50),  
    COMPANY VARCHAR2(30)  
);
```

```
CREATE TABLE ANIMAL_DETAILS
```

```
(  
    REG_DATE DATE,  
    REG_NO VARCHAR2(20),  
    OWNER_NAME VARCHAR2(30),  
    PET_NAME VARCHAR2(20),  
    GENDER VARCHAR2(20),  
    AGE VARCHAR2(10),  
    SPECIES VARCHAR2(20),  
    BREED VARCHAR2(30),  
    HEIGHT VARCHAR2(20),  
    WEIGHT VARCHAR2(20),  
    MOBILE_NO VARCHAR2(13),  
    EMAILID VARCHAR2(50),  
    STATE VARCHAR2(20),  
    ADDRESS VARCHAR2(150),  
    CURRENT_CONDITION VARCHAR2(250),  
    MEDICAL_HISTORY VARCHAR2(250)  
);
```

```
CREATE TABLE DRUGS_DETAILS
(
    DRUGS_ID VARCHAR2(20),
    DRUGS_NAME VARCHAR2(50),
    DESCRIPTION VARCHAR2(100)
);
```

```
CREATE TABLE APPOINTMENT
(
    REG_NO VARCHAR2(20),
    CURDATE DATE,
    CURTIME VARCHAR2(30),
    APPOINTMENT_NO VARCHAR2(40),
    OWNER_NAME VARCHAR2(20),
    PET_NAME VARCHAR2(20),
    MOBILE_NO VARCHAR2(20),
    GENDER VARCHAR2(20),
    AGE VARCHAR2(10),
    HEIGHT VARCHAR2(20),
    WEIGHT VARCHAR2(30),
    APPOINTMENT_DATE DATE,
    APPOINTMENT_TIME VARCHAR2(30),
    PROBLEM VARCHAR2(30),
    SERVICES VARCHAR2(70),
    FEE VARCHAR2(30),
    DOC_EMP_NAME VARCHAR2(20)
);
```

```
CREATE TABLE VACCINATION_DETAILS
(
    ID VARCHAR2(30),
    VACCINATION VARCHAR2(50)
);
```

```
CREATE TABLE EMP_TREATMENT
(
    CURDATE DATE,
    PRESC_NO VARCHAR2(20),
    APMNT_NO VARCHAR2(30),
    REG_NO VARCHAR2(30),
    EMPNAME VARCHAR2(50),
    OWNER_NAME VARCHAR2(50),
    PET_NAME VARCHAR2(30),
    SERVICE VARCHAR2(30),
    STATUS VARCHAR2(30)
);
```

```
CREATE TABLE DOC_TREATMENT
(
    CURDATE DATE,
    PRESC_NO VARCHAR2(20),
    APMNT_NO VARCHAR2(30),
    REG_NO VARCHAR2(30),
    EMPNAME VARCHAR2(50),
    OWNER_NAME VARCHAR2(50),
    PET_NAME VARCHAR2(30),
    SERVICE VARCHAR2(30),
    PRESCRIPTION VARCHAR2(200)
);
```


MODULE DESCRIPTION & PROCESS LOGIC

Login

This module is used to show the password protected screen and also login main parent form. This password protected screen offers another layer of protection by preventing other user from using “Veterinary Management System” when that user is away from screen.

Admin Portal

Doctor Register: -

In this module we can enter the Doctor details. By the help of this Module Clinic keeps full details about Doctor.

- Enter Doctor Name
- Enter Date Of Birth
- Select Gender
- Enter Qualification
- Enter Specialization
- Enter Mobile No.
- Enter Email Id
- Enter Address
- Enter Userid
- Enter Security Question
- Enter Answer
- Enter Password (Password Contain Uppercase, Lowercase, Number & Special Characters).

- Click Save button to save the entered details.
- If we want to edit the Doctor details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Employee Register: -

In this module we can enter the Employee details. By the help of this Module Clinic keeps full details about Employee.

- Enter Employee Name
- Enter Date Of Birth
- Select Gender
- Enter Qualification
- Enter Post
- Enter Mobile No.
- Enter Email Id
- Enter Address
- Enter Userid
- Enter Security Question
- Enter Answer
- Enter Password (Password Contain Uppercase, Lowercase, Number & Special Characters).
- Click Save button to save the entered details.
- If we want to edit the Employee details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Receptionist Register: -

In this module we can enter the Receptionist details. By the help of this Module Clinic keeps full details about Receptionist.

- Enter Receptionist Name

- Enter Date Of Birth
- Select Gender
- Enter Qualification
- Enter Specialization
- Enter Mobile No.
- Enter Email Id
- Enter Address
- Enter Userid
- Enter Security Question
- Enter Answer
- Enter Password (Password Contain Uppercase, Lowercase, Number & Special Characters).
- Click Save button to save the entered details.
- If we want to edit the Receptionist details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Pet's Record: -

In this module we can see and search the record of pet patient which can register by Receptionist.

Appointment: -

In this module we can see and search the record of pet patient which can appointment by Receptionist.

Treatment: -

In this module we can see and search the record of pet patient which can treat by which Doctor and Employee.

Medicine: -

In this module we can enter the Medicine details. By the help of this Module Clinic keeps full details about Medicine, and in this Module we can search a particular medicine.

- Enter Medicine Name
- Choose Type of Medicine
- Choose or Add New Drug
- Enter Company
- Click Save button to save the entered details.
- If we want to edit the Medicine details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Vaccination: -

In this module we can enter the Vaccination details. By the help of this Module Clinic keeps full details about Vaccine, and in this Module we can search a particular Vaccine.

- Enter Vaccine Name
- Enter Description
- Click Save button to save the entered details.
- If we want to edit the Vaccine details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Services: -

In this module we can enter the Service details. By the help of this Module Clinic keeps full details about Service, and in this Module we can search a particular medicine, Here the administrator provides their Service which are done by clinic.

- Enter Service Name
- Enter Fee
- Click Save button to save the entered details.
- If we want to edit the Service details, click Edit button. Make necessary modifications.
- Press Update button to save modification.

Receptionist Portal

Registration (New Patient): -

In this module we can register the new Patient. By the help of this Module Clinic keeps full details about Patient's Registration.

- Enter Owner Name
- Enter Pet Name
- Choose Gender
- Enter Age
- Choose Species
- Enter Breed Name
- Enter Height
- Enter Weight
- Enter Mobile Number
- Choose State
- Enter Email Id
- Enter Address
- Click Save button to save the entered details.
- Click Clear button to clear all the Text field

Appointment (Old Patient): -

In this module we can appoint the old Patient or new Patient. By the help of this Module Clinic keeps full details about Appointment.

If new Patient not necessary to fill all the text field, otherwise

- Enter Registration Number
- Enter Owner Name
- Enter Mobile Number
- Enter Pet Name
- Choose Gender
- Enter Age
- Enter Height
- Enter Weight
- Enter Appointment Date
- Enter Appointment Time
- Enter Problem
- Choose Service
- Choose Doctor or Employee
- Click Save button to save the entered details.
- Click Clear button to clear all the Text field

Doctor Portal

Treatment -

In this module we can treat or prescribed the Patient for their Problem. By the help of this Module Clinic keeps full details about Patient's Treatment. It is easily to use; in this module a table is given which show the appointment data for current date.

- Select Appointment data from Table
- Click on Add Medicine for Patient for given Problem
- Click on Add Vaccination for Patient for given Problem
- Click Save button to save the entered details.
- Click Clear button to clear all the Text field

Employee Portal

Treatment -

In this module we can solve the Patient for their Problem. By the help of this Module Clinic keeps full details about Patient's Problem.

It is easily to use; in this module a table is given which show the data for current date.

- Select the data from Table
- Choose Remark
- Click Save button to save the entered details.
- Click Clear button to clear all the Text field


DESIGNING


Splash Screen:-




Login Screen:-

LOGIN

USER 

USER ID 

PASSWORD 

[Forget Password?](#)


LOGIN

[About](#)



Logging In...

Forget Password Screen: -

 FORGET PASSWORD

VETERINARY MANAGEMENT SYSTEM
Forget Password

USER TYPE	<input type="text"/>
USER ID	admin
NAME	ADMINISTRATOR
SECURITY QUESTION	FAVOURITE COLOR
ANSWER	BLUE
PASSWORD
CONFIRM PASSWORD
<div>CHANGE PASSWORD</div> <div>CANCEL</div>	

ADMIN PORTAL SCREEN: -



ADMIN UPDATE PROFILE SCREEN: -

8 UPDATE PROFILE

VETERINARY MANAGEMENT SYSTEM
Update Profile

USER ID	<input type="text" value="admin"/>
NAME	<input type="text" value="ADMINISTRATOR"/>
SECURITY QUESTION	<input type="text" value="FAVOURITE COLOR"/>
ANSWER	<input type="text" value="BLUE"/>
PASSWORD	<input type="password" value="....."/>
CONFIRM PASSWORD	<input type="password" value="....."/>
<input type="button" value="EDIT"/> <input type="button" value="CANCEL"/>	

DOCTOR REGISTER SCREEN: -

REGISTER FORM

VETERINARY MANAGEMENT SYSTEM

Doctor Register

ID

DOC-1

NAME

RAM KUMAR

SEARCH

DATE OF BIRTH

08-Jun-1993

USER ID

GENDER

MALE

SECURITY QUESTION

QUALIFICATION

NONE

ANSWER

SPECILLIZATION

NONE

PASSWORD

MOBILE NO.

7894561230

CONFIRM PASSWORD

IMAGE

UPLOAD

E-Mail ID

ram@xyz.com

FULL ADDRESS

PUNE


NEW

UPDATE

CLEAR

ID	NAME	DOB	GEND...	QUALIF...	SPECI...	MOBIL...	EMAILID	ADDRE...
DOC-1	RAM ...	1993...	MALE	NONE	NONE	7894...	ram@...	PUNE

RECEPTIONIST REGISTER SCREEN: -


REGISTER FORM

VETERINARY MANAGEMENT SYSTEM

Receptionist Register

ID

REC-1

USER ID

SEARCH

NAME

ABC

SECURITY QUESTION

DATE OF BIRTH

01-Nov-1999

ANSWER

GENDER

FEMALE

PASSWORD

MOBILE NO.

1234567890

CONFIRM PASSWORD

IMAGE

UPLOAD

E-Mail ID

abc@xyz.com

FULL ADDRESS

DELHI

NEW

EDIT

CLEAR

ID	NAME	DOB	GENDER	MOBILE...	EMAILID	ADDRESS
REC-1	ABC	1999-1...	FEMALE	123456...	abc@xy...	DELHI

EMPLOYEE REGISTER SCREEN: -

REGISTER FORM

VETERINARY MANAGEMENT SYSTEM

Employee Register

ID

EMP-1

NAME

SHYAM KUMAR

SEARCH

DATE OF BIRTH

08-Jun-1995

USER ID

GENDER

MALE

SECURITY QUESTION

QUALIFICATION

NONE

ANSWER

POST

NONE

PASSWORD

MOBILE NO.

9874563210

CONFIRM PASSWORD

IMAGE

UPLOAD

E-Mail ID

shyam@xyz.com

FULL ADDRESS

PATNA

NEW

EDIT

CLEAR

ID	NAME	DOB	GEND...	QUALIF...	POST	MOBIL...	EMAILID	ADDRE...
EMP-1	SHYA...	1995...	MALE	NONE	NONE	9874...	shya...	PATNA

PATIENT DATABASE SCREEN:

Patient Database

VETERINARY MANAGEMENT SYSTEM

Patient Details

SEARCH

REG_...	REG_...	OWNE...	PET_N...	GEND...	AGE	SPECI...	BREED	HEIGHT	WEIGHT	MOBIL...	EMAILID	ADDR...	STATE	MEDIC...
201...	PET-1	SHU...	TUFY	MALE	15	DOG	GERMAN	015CM	050KG	897...	shub...	PATNA	BIHAR	11/...
201...	PET-2	MOH...	XYZ	MALE	10	CATS	UNK	015CM	005KG	789...	moha...	PATNA	BIHAR	11/...

53

APPOINTMENT DATABASE SCREEN: -

Appointment Database

VETERINARY MANAGEMENT SYSTEM

Appointment Details

SEARCH

CURDATE	CURTIME	REG_NO	APPOINT...	OWNER_...	PET_NAME	MOBILE_...	APPOINT...	APPOINT...	PROBLEM	DOC_EM...
2018-0...	08:17 PM	PET-1	VISITOR-1	SHUBHA...	TUFY	897546...	2018-0...	10:17 PM	NONE	RAM KUMAR
2018-0...	08:32 PM	PET-2	VISITOR-2	MOHAN ...	XYZ	789456...	2018-0...	10:32 PM	CLEANING	SHYAM ...

TREATMENT DATABASE SCREEN: -

TREATMENT DATABASE

VETERINARY MANAGEMENT SYSTEM

Treatment Details

DOCTOR

SEARCH

CURDATE	PRESC_NO	APMNT_NO	REG_NO	DOCNAME	OWNER_NA...	PET_NAME	SERVICE	PRESCRIPTI...
2018-06-...	PRESC-1	APPOINTM...	PET-1	RAM KUMAR	SHUBHAM ...	TUFY	NONE	XYZTABLE...

EMPLOYEE

SEARCH

CURDATE	PRESC_NO	APMNT_NO	REG_NO	EMPNAME	OWNER_NA...	PET_NAME	SERVICE	STATUS
2018-06-...	PRESC-1	VISITOR-2	PET-2	SHYAM KUMAR	MOHAN KUMAR	XYZ	TEETH CL...	SOLVED

TREATMENT DATABASE SCREEN: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM
Treatment Details

DATE	2018-06-11 00:00:...	PRESCRIPTION NO.	PRESC-1
APPOINTMENT ID	APPOINTMENT NO	REGISTRATION NO.	PET-1
OWNER NAME	SHUBHAM KUMAR	PET NAME	TUFY
DOCTOR/EMPLOYEE NAME	RAM KUMAR	PROBLEM	NONE
APPOINTMENT TIME	XYZTABLETS - MORNING - 1 , NOON - 1 , EVENING - 1 - 30 DAYS TEST1 - AFTER 15 DAYS - 90 DAYS		

MEDICINE DATABASE SCREEN: -

Medicine Database

VETERINARY MANAGEMENT SYSTEM
Medicine Details

MEDICINE ID	MED-1	SEARCH	
MEDICINE NAME	XYZ		
DRUGS	XYZ	ADD DRUGS	
TYPE	TABLETS		
COMPANY NAME	XYZ		
NEW	EDIT	CLEAR	

ID	NAME	DRUGS	TYPE	COMPANY
MED-1	XYZ	XYZ	TABLETS	XYZ
MED-2	QWE	QWER	CAPSULE	QWERTY

VACCINATION DATABASE SCREEN: -

Vaccination Database

VETERINARY MANAGEMENT SYSTEM
Vaccination Details

SEARCH

VACCINATION ID

VACCINATION NAME

ID	VACCINATION
VACC-1	TEST1

SERVICE DATABASE SCREEN: -

Service Database

VETERINARY MANAGEMENT SYSTEM
Service Details

SEARCH

SERVICE ID

SERVICE NAME

FEE

ID	SERVICE	FEE
SERV-1	COUNSALTENCY	500
SERV-2	GROOMING	600
SERV-3	TEETH CLEANING	400

RECEPTIONIST PORTAL SCREEN: -



The screenshot shows the 'RECEPTIONIST PORTAL SCREEN' of the 'VETERINARY MANAGEMENT SYSTEM'. The header is blue with a receptionist icon and the text 'VETERINARY MANAGEMENT SYSTEM' and 'Receptionist Portal'. Below the header, there is a profile section for 'ABC Receptionist' with an 'Update Profile' link. Two buttons, 'NEW PATIENT' and 'OLD PATIENT', are visible. On the right, there is a large image of a dog wearing a headset with a microphone.

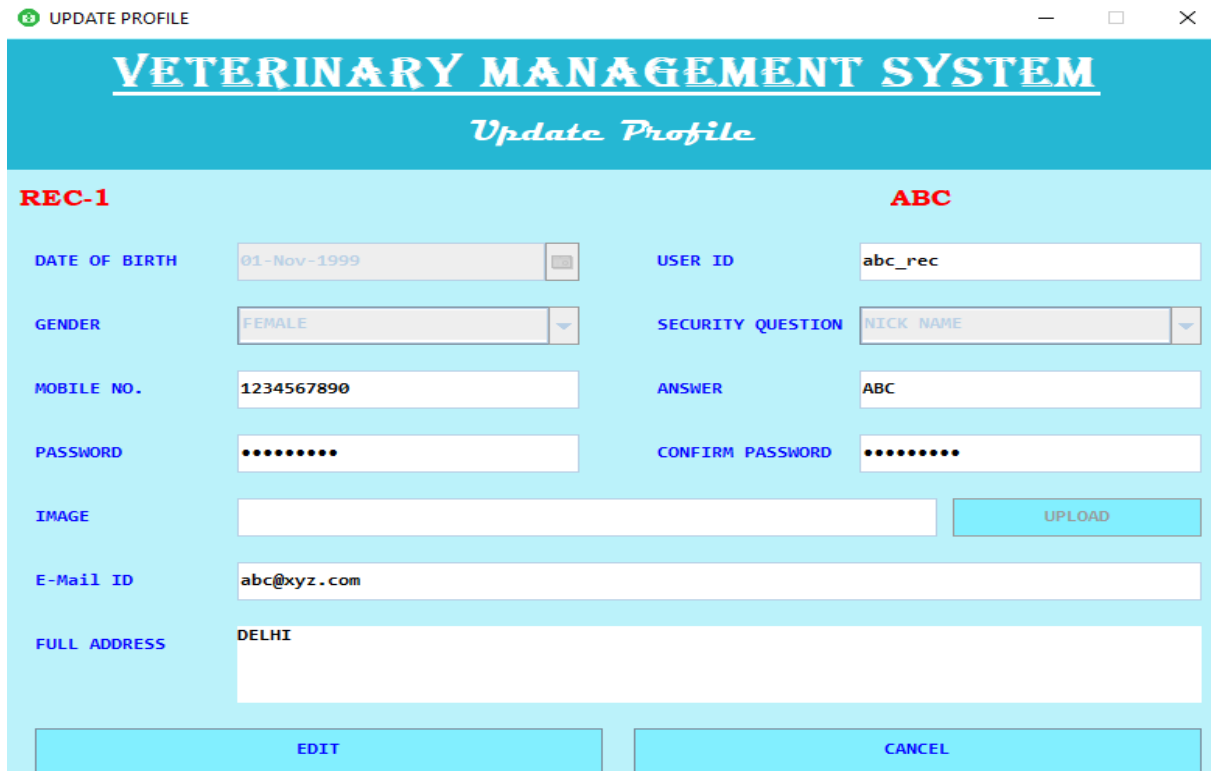
VETERINARY MANAGEMENT SYSTEM
Receptionist Portal

ABC
Receptionist
[Update Profile](#)

NEW PATIENT

OLD PATIENT

RECEPTIONIST UPDATE PROFILE SCREEN: -



The screenshot shows the 'RECEPTIONIST UPDATE PROFILE SCREEN' of the 'VETERINARY MANAGEMENT SYSTEM'. The header is blue with the text 'VETERINARY MANAGEMENT SYSTEM' and 'Update Profile'. Below the header, there is a form for updating the profile. The form is divided into two columns. The left column contains fields for 'DATE OF BIRTH', 'GENDER', 'MOBILE NO.', 'PASSWORD', 'IMAGE', 'E-Mail ID', and 'FULL ADDRESS'. The right column contains fields for 'USER ID', 'SECURITY QUESTION', 'ANSWER', and 'CONFIRM PASSWORD'. There is an 'UPLOAD' button next to the 'IMAGE' field. At the bottom, there are 'EDIT' and 'CANCEL' buttons.

UPDATE PROFILE

VETERINARY MANAGEMENT SYSTEM
Update Profile

REC-1 **ABC**

DATE OF BIRTH: 01-Nov-1999

GENDER: FEMALE

MOBILE NO.: 1234567890

PASSWORD:

IMAGE: [Image Field] **UPLOAD**

E-Mail ID: abc@xyz.com

FULL ADDRESS: DELHI

USER ID: abc_rec

SECURITY QUESTION: NICK NAME

ANSWER: ABC

CONFIRM PASSWORD:

EDIT **CANCEL**

PATIENT REGISTER DATABASE: -

REGISTER FORM

VETERINARY MANAGEMENT SYSTEM
Patient Register

REGISTRATION NO	PET-1	OLD	REGISTRATION DATE	11-Jun-2018	MEDICAL HISTORY
OWNER NAME	SHUBHAM KUMAR		PET NAME	TUFY	
GENDER	MALE		AGE	15	
SPECIES	DOG		BREED	GERMAN	
HEIGHT	015CM		WEIGHT	050KG	
MOBILE NO.	8975462310		STATE	BIHAR	
E-Mail ID	shubham@xyz.com				
FULL ADDRESS	PATNA				
SAVE			CLEAR		

APPOINTMENT SCREEN: -

BOOK APPOINTMENT

VETERINARY MANAGEMENT SYSTEM
Appointment

Jun 11, 2018 08:17 PM

REGISTRATION NO	PET-1		
APPOINTMENT ID	VISITOR-1	OWNER NAME	SHUBHAM KUMAR
MOBILE NO.	8975462310	PET NAME	TUFY
GENDER	MALE	AGE	15
HEIGHT	015CM	WEIGHT	050KG
APPOINTMENT DATE	11-Jun-2018	APPOINTMENT TIME	10:17 PM
PROBLEM	NONE	SERVICES	CONSULTANT CHARGES
DOCTOR/EMPLOYEE	RAM KUMAR	CHARGES	500
BOOK		CLEAR	

DOCTOR PORTAL SCREEN: -

**VETERINARY MANAGEMENT SYSTEM**
Doctor Portal



RAM KUMAR
DOCTOR
[Update Profile](#)

TREATMENT



DOCTOR UPDATE PROFILE SCREEN: -

UPDATE PROFILE

VETERINARY MANAGEMENT SYSTEM
Update Profile

DOC-1**RAM KUMAR**

DATE OF BIRTH08-Jun-1993

USER IDram_doc

GENDERMALE

SECURITY QUESTIONFavourite Color

MOBILE NO.7894561230

ANSWERBLUE

PASSWORD

CONFIRM PASSWORD

IMAGE

UPLOAD

E-Mail IDram@xyz.com

FULL ADDRESSPUNE

EDIT

CANCEL

TREATMENT FORM SCREEN: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM

Treatment Form

RAM KUMAR APPOINTMENT NO. Jun 11, 2018 08:21 PM

PRESCRIPTION NO. REGISTRATION NO. CURRENT CONDITION MEDICAL HISTORY

OWNER NAME PET NAME

AGE BREED

HEIGHT CM WEIGHT KG

PRESCRIPTION

ADD MEDICINE

ADD VACINATION

SAVE CLEAR

APPO...	REG...	OWN...	PET...	AGE	BREED	HEIG...	WEIG...	PROB...	MEDI...
VIS...	PET-1	SHU...	TUFY	15	GE...	015CM	050KG	NONE	

ADD MEDICINE SCREEN: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM

Treatment Form

RAM KUMAR APP Jun 11, 2018 08:21 PM

PRESCRIPTION NO. PRESC-1 REGISTRATION NO. CURRENT CONDITION MEDICAL HISTORY

OWNER NAME SHUBHAM KUMAR PET NAME

AGE 15 BREED

HEIGHT 015CM WEIGHT

PRESCRIPTION

ADD MEDICINE

ADD VACINATION

SAVE CLEAR

MEDICINE

VETERINARY MANAGEMENT SYSTEM

Medicine

MEDICINE NAME XYZ

TYPE TABLETS

MORNING NOON EVENING

DOSAGE 1 1 1

DURATION 30 Days

ADD CANCEL

REG...	OWN...	PET...	AGE	BREED	HEIG...	WEIG...	PROB...	MEDI...
PET-1	SHU...	TUFY	15	GE...	015CM	050KG	NONE	

ADD VACCINATION SCREEN: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM

Treatment Form

RAM KUMAR

APPOINTMENT NO VISITOR-1

Jun 11, 2018 08:21 PM

PRESCRIPTION NO PRESC-1

OWNER NAME SHUBHAM KUMAR

AGE 15

HEIGHT 015CM

PRESCRIPTION XYZTABLETS - MORNING - 1 , NOON - 1

REGISTRATION NO PET-1

PET NAME TUFY

BREED GERMAN

WEIGHT 050KG

CURRENT CONDITION

MEDICAL HISTORY

VACCINATION

VACCINATION NAME TEST1

TYPE after 15 Days

DURATION 90 Days

ADD CANCEL

ADD MEDICINE

ADD VACCINATION

SAVE CLEAR

APPO...	REG...	OWN...	PET...	AGE	BREED	HEIG...	WEIG...	PROB...	MEDI...
T-1	SHU...	TUFY	15	GE...	015CM	050KG	NONE		

AFTER FILL ALL DATA TREATMENT FORM SCREEN: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM

Treatment Form

RAM KUMAR

APPOINTMENT NO VISITOR-1

Jun 11, 2018 08:21 PM

PRESCRIPTION NO PRESC-1

OWNER NAME SHUBHAM KUMAR

AGE 15

HEIGHT 015CM

PRESCRIPTION XYZTABLETS - MORNING - 1 , NOON - 1 , EVENING - 1
TEST1 - AFTER 15 DAYS - 90 Days

REGISTRATION NO PET-1

PET NAME TUFY

BREED GERMAN

WEIGHT 050KG

CURRENT CONDITION NONE

MEDICAL HISTORY

ADD MEDICINE

ADD VACCINATION

SAVE CLEAR

APPO...	REG...	OWN...	PET...	AGE	BREED	HEIG...	WEIG...	PROB...	MEDI...
VIS...	PET-1	SHU...	TUFY	15	GE...	015CM	050KG	NONE	

EMPLOYEE PORTAL SCREEN: -



EMPLOYEE UPDATE PROFILE SCREEN: -

UPDATE PROFILE

VETERINARY MANAGEMENT SYSTEM
Update Profile

EMP-1 **SHYAM KUMAR**

DATE OF BIRTH	<input type="text" value="08-Jun-1995"/>	USER ID	<input type="text" value="shyam_emp"/>
GENDER	<input type="text" value="MALE"/>	SECURITY QUESTION	<input type="text" value="Nick Name"/>
MOBILE NO.	<input type="text" value="9874563210"/>	ANSWER	<input type="text" value="SHYAM"/>
PASSWORD	<input type="password" value="....."/>	CONFIRM PASSWORD	<input type="password" value="....."/>
IMAGE	<input type="text"/>		<input type="button" value="UPLOAD"/>
E-Mail ID	<input type="text" value="shyam@xyz.com"/>		
FULL ADDRESS	<input type="text" value="PATNA"/>		

EMPLOYEE WORK FORM: -

TREATMENT FORM

VETERINARY MANAGEMENT SYSTEM
Treatment Form

SHYAM KUMARVISITOR-2Jun 11, 2018 08:35 PM

PREScription NOPRESC-1REGISTRATION NOPET-2

OWNER NAMEMOHAN KUMARPET NAMEXYZ

SERVICESTEETH CLEANINGREMARK

SOLVED

APPOINTMENT_NO	REG_NO	OWNER_NAME	PET_NAME	SERVICES
VISITOR-2	PET-2	MOHAN KUMAR	XYZ	TEETH CLEANING

SAVECLEAR

CODING

Connection.java

```
import java.sql.*;
public class Connections
{
    public static Connection getConnections()
    {
        String url = "jdbc:oracle:thin:@localhost:1521:XE";
        String username = "vms";
        String password = "vms";
        Connection conn = null;
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            conn=DriverManager.getConnection(url,username,pasword);
        }
        catch (ClassNotFoundException e)
        {
            e.printStackTrace();
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        return conn;
    }
}
```

Base.Java

```
import java.sql.*;
public abstract class Base
{
    Connection conn;
    PreparedStatement ps;
    ResultSet rs;
    String Id, Name, DOB, Gender, Qualification, Drugs, Type, Company, Service,
    ExName,Specilaization,Post,Description,PrescriptionNo,Prescription,Emailid,Address,UserType,
    FullName, UserId, Password,Mobile_No, SecurityQuestion, Answer, image, RegDate,
    RegNo,Owner_Name, AnimalName, Age, Species, Breed,Height, Weight, State,
    Condition,MedicalHistory, Date, Time, AppointmentDate, AppointmentTime, Problem,Fee,
    Vaccination;
    static String Medical = null;
}
```

SplashScreen.Java

```
import java.awt.*;
import javax.swing.*;
class SplScreen
{
    public Label load_label;
    public JLabel name_label, ver_label, comp_label;
    public Frame frame;
```

```

public int i;
SplScreen()
{
    frame = new Frame();
    frame.setSize(758, 360);
    frame.setBackground(Color.black);
    frame.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
    name_label = new JLabel("Veterinary Management System");
    name_label.setFont(new Font("Script MT Bold", Font.BOLD, 44));
    name_label.setForeground(new Color(255, 255, 255));
    name_label.setBounds(85, 41, 635, 64);
    JLabel ver_label = new JLabel("Version 1.0");
    ver_label.setFont(new Font("Impact", Font.PLAIN, 18));
    ver_label.setForeground(new Color(255, 255, 255));
    ver_label.setBounds(95, 116, 124, 38);
    load_label = new JLabel();
    load_label.setForeground(Color.white);
    load_label.setFont(new Font("Consolas", Font.BOLD, 22));
    load_label.setBounds(570, 261, 225, 38);
    JLabel comp_label = new JLabel("\u00A9 2018 ShubhTechno Inc. : www.shubhtechno.com");
    comp_label.setFont(new Font("Bookman Old Style", Font.BOLD, 13));
    comp_label.setForeground(new Color(255, 255, 255));
    comp_label.setBounds(385, 328, 358, 14);
    Canvas canvas = new Canvas();
    canvas.setBackground(new Color(0, 100, 0));
    canvas.setForeground(Color.GREEN);
    canvas.setBounds(60, 10, 5, 360);
    Canvas canvas_1 = new Canvas();
    canvas_1.setBackground(new Color(0, 100, 0));
    canvas_1.setBounds(0, 309, 732, 5);
    frame.setLayout(null);
    frame.setUndecorated(true);
    frame.setAlwaysOnTop(false);
    Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
    int framewidth = frame.getSize().width; // get the width of the frame
    int frameheight = frame.getSize().height; // get the height of the frame
    int framelocationX = (dim.width - framewidth) / 2;
    int framelocationY = (dim.height - frameheight) / 2;
    frame.setLocation(framelocationX, framelocationY);
    frame.add(canvas);
    frame.add(canvas_1);
    frame.add(ver_label);
    frame.add(name_label);
    frame.add(load_label);
    frame.add(comp_label);
    frame.setVisible(true);
}

public void iterate()
{
    int i = 0;
    while (i <= 100)
    {
        load_label.setText("Loading    " + String.valueOf(i) + "%");
        i = i + 2;
        try
        {
            Thread.sleep(100);

```



```

        }
        catch (Exception e)
        {
        }
    }
    if (i == 102)
    {
        new Login();
        frame.dispose();
    }
}
}
public class SplashScreen
{
    public static void main(String[] args)
    {
        SplScreen m = new SplScreen();
        m.iterate();
    }
}

```

Login Screen

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.util.TimerTask;
import javax.swing.*;
import javax.swing.border.EmptyBorder;
class Log extends Base implements ActionListener
{
    public JFrame frm;
    public JPanel contentPane,pnl_bg,panel,login,panel_1,loader;
    public JTextField txt_userid;
    public JSeparator separator,separator_1;
    public JPasswordField passwordField;
    public JLabel label_1, lblX, lblLogin, lblForgetPassword, label_2, lblNewLabel, lblLoggingIn,
    lblUser, lblUserId, lblPassword, lblAbout;
    public JButton btnLogin;
    public Canvas canvas;
    public JComboBox<Object> comboBox;
    String arr[] = { "Receptionist", "Employee", "Doctor" };
    boolean flag;
    Log()
    {
        frm = new JFrame();
        frm.setSize(635, 324);
        frm.setLayout(null);
        frm.setUndecorated(true);
        frm.setAlwaysOnTop(false);
        frm.setBackground(Color.WHITE);
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int framelocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, framelocationY);
    }
}

```

```

frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
frm.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
contentPane = new JPanel();
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
frm.setContentPane(contentPane);
contentPane.setLayout(null);
pnl_bg = new JPanel();
pnl_bg.setBackground(Color.WHITE);
pnl_bg.setBounds(0, 0, 635, 324);
contentPane.add(pnl_bg);
pnl_bg.setLayout(null);
panel = new JPanel();
panel.setBackground(Color.WHITE);
panel.setBounds(0, 0, 635, 324);
pnl_bg.add(panel);
panel.setLayout(new CardLayout(0, 0));
login = new JPanel();
login.setBackground(Color.WHITE);
panel.add(login, "name_1246115112899");
login.setLayout(null);
panel_1 = new JPanel();
panel_1.setBackground(Color.WHITE);
panel_1.setBounds(601, 0, 34, 30);
login.add(panel_1);
panel_1.setLayout(null);
lblX = new JLabel(" X ");
lblX.addMouseListener(new MouseAdapter()
{
    public void mouseClicked(MouseEvent arg0)
    {
        System.exit(0);
    }
    @Override
    public void mouseEntered(MouseEvent e)
    {
        panel_1.setBackground(Color.RED);
        lblX.setForeground(Color.WHITE);
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        panel_1.setBackground(Color.WHITE);
        lblX.setForeground(Color.BLACK);
    }
});
lblX.setFont(new Font("Copperplate Gothic Bold", Font.BOLD, 10));
lblX.setForeground(Color.BLACK);
lblX.setBackground(Color.WHITE);
lblX.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblX.setBounds(0, 0, 29, 30);
panel_1.add(lblX);
lblLogin = new JLabel("Login");
lblLogin.setFont(new Font("AR JULIAN", Font.BOLD, 32));
lblLogin.setBounds(35, 32, 107, 40);
login.add(lblLogin);
lblUser = new JLabel("User");
lblUser.setForeground(new Color(0, 128, 0));

```

```

lblUser.setFont(new Font("Algerian", Font.PLAIN, 18));
lblUser.setBounds(120, 100, 69, 30);
login.add(lblUser);
comboBox = new JComboBox<Object>(arr);
comboBox.setBackground(new Color(255, 255, 255));
comboBox.setForeground(new Color(128, 128, 128));
comboBox.setFont(new Font("Tahoma", Font.PLAIN, 18));
comboBox.setSelectedIndex(-1);
comboBox.setBorder(null);
comboBox.setBounds(250, 100, 338, 30);
comboBox.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
login.add(comboBox);
label_2 = new JLabel("");
label_2.setIcon(new ImageIcon(Login.class.getResource("/images/contacts_18px.png")));
label_2.setBounds(220, 100, 18, 30);
login.add(label_2);
lblUserId = new JLabel("User Id");
lblUserId.setForeground(new Color(0, 128, 0));
lblUserId.setFont(new Font("Algerian", Font.PLAIN, 18));
lblUserId.setBounds(120, 140, 69, 30);
login.add(lblUserId);
label = new JLabel("");
label.setIcon(new ImageIcon(Login.class.getResource("/images/contacts_18px.png")));
label.setBounds(220, 143, 18, 28);
login.add(label);
txt_userid = new JTextField();
txt_userid.setFont(new Font("Tahoma", Font.PLAIN, 18));
txt_userid.setForeground(new Color(102, 102, 102));
txt_userid.setBackground(Color.WHITE);
txt_userid.setText("UserId");
txt_userid.setBounds(250, 140, 333, 30);
txt_userid.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txt_userid.setBorder(null);
txt_userid.addFocusListener(new FocusAdapter()
{
    @Override
    public void focusGained(FocusEvent arg0)
    {
        txt_userid.setText("");
    }
    @Override
    public void focusLost(FocusEvent e)
    {
        if(txt_userid.getText().equals(""))
        {
            txt_userid.setText("UserId");
        }
    }
});
login.add(txt_userid);
txt_userid.setColumns(10);
separator = new JSeparator();
separator.setForeground(new Color(0, 128, 0));
separator.setBackground(new Color(0, 128, 0));
separator.setBounds(250, 175, 333, 14);
login.add(separator);
lblPassword = new JLabel("Password");
lblPassword.setForeground(new Color(0, 128, 0));
lblPassword.setFont(new Font("Algerian", Font.PLAIN, 18));

```

```

lblPassword.setBounds(120, 181, 93, 30);
login.add(lblPassword);
label_1 = new JLabel("");
label_1.setIcon(new ImageIcon(Login.class.getResource("/images/unlock_18px.png")));
label_1.setBounds(220, 181, 18, 28);
login.add(label_1);
passwordField = new JPasswordField();
passwordField.setFont(new Font("Tahoma", Font.PLAIN, 18));
passwordField.setForeground(new Color(102, 102, 102));
passwordField.setBackground(Color.WHITE);
passwordField.setBounds(250, 181, 333, 30);
passwordField.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
passwordField.addFocusListener(new FocusAdapter()
{
    @Override
    public void focusGained(FocusEvent arg0)
    {
        passwordField.setText("");
    }
    @SuppressWarnings("deprecation")
    @Override
    public void focusLost(FocusEvent e)
    {
        if(passwordField.getText().equals(""))
        {
            passwordField.setText("Password");
        }
    }
});
passwordField.setBorder(null);
passwordField.setText("Password");
login.add(passwordField);
separator_1 = new JSeparator();
separator_1.setForeground(new Color(0, 128, 0));
separator_1.setBackground(new Color(0, 128, 0));
separator_1.setBounds(250, 214, 333, 20);
login.add(separator_1);
lblForgetPassword = new JLabel("Forget Password?");
lblForgetPassword.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent e)
    {
        lblForgetPassword.setForeground(Color.RED);
        canvas.setBackground(Color.RED);
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        lblForgetPassword.setForeground(UIManager.getColor("CheckBox.darkShadow"));
        canvas.setBackground(new Color(0, 128, 0));
    }
    @Override
    public void mouseClicked(MouseEvent e)
    {
        new ForgetPassword();
        frm.dispose();
    }
}

```

```

});
lblForgetPassword.setFont(new Font("Ebrima", Font.BOLD, 16));
lblForgetPassword.setForeground(UIManager.getColor("CheckBox.darkShadow"));
lblForgetPassword.setBounds(250, 252, 159, 22);
lblForgetPassword.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
login.add(lblForgetPassword);
canvas = new Canvas();
canvas.setBackground(new Color(0, 128, 0));
canvas.setBounds(250, 275, 133, 2);
login.add(canvas);
btnLogin = new JButton("LOGIN");
btnLogin.setForeground(new Color(0, 128, 0));
btnLogin.setFont(new Font("Algerian", Font.BOLD, 23));
btnLogin.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnLogin.addActionListener(this);
btnLogin.addMouseListener(new MouseAdapter()
{
    public void mouseEntered(MouseEvent e)
    {
        btnLogin.setBackground(new Color(0, 128, 0));
        btnLogin.setForeground(Color.WHITE);
    }
    @Override
    public void mouseExited(MouseEvent e)
    {
        btnLogin.setBackground(Color.WHITE);
        btnLogin.setForeground(new Color(0, 128, 0));
    }
});
btnLogin.setBackground(Color.WHITE);
btnLogin.setBounds(425, 250, 164, 40);
login.add(btnLogin);
lblAbout = new JLabel("About");
lblAbout.setBackground(new Color(192, 192, 192));
lblAbout.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseEntered(MouseEvent arg0)
    {
        lblAbout.setForeground(new Color(128, 128, 128));
    }
    @Override
    public void mouseExited(MouseEvent arg0)
    {
        lblAbout.setForeground(new Color(192, 192, 192));
    }
    @Override
    public void mouseClicked(MouseEvent arg0) {
    }
});
lblAbout.setFont(new Font("Tahoma", Font.BOLD, 11));
lblAbout.setForeground(new Color(192, 192, 192));
lblAbout.setBounds(585, 301, 46, 14);
lblAbout.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
login.add(lblAbout);
loader = new JPanel();
loader.setBackground(Color.WHITE);
panel.add(loader, "name_1277481775937");
loader.setLayout(null);

```

```

lblNewLabel = new JLabel("");
lblNewLabel.setIcon(new ImageIcon(Login.class.getResource("/images/ring.gif")));
lblNewLabel.setBounds(258, 63, 123, 119);
loader.add(lblNewLabel);
lblLoggingIn = new JLabel("Logging In...");
lblLoggingIn.setFont(new Font("SansSerif", Font.PLAIN, 18));
lblLoggingIn.setForeground(new Color(0, 128, 0));
lblLoggingIn.setBounds(271, 209, 110, 32);
loader.add(lblLoggingIn);
frm.setVisible(true);
}
@SuppressWarnings("deprecation")
@Override
public void actionPerformed(ActionEvent e)
{
    String text,pwd;
    text = txt_userid.getText();
    pwd= passwordField.getText();
    if(e.getSource().equals(btnLogin))
    {
        if (frmvalidate())
        {
            if (comboBox.getSelectedIndex() == -1)
            {
                LoginOperation ad = new LoginOperation(txt_userid.getText(),
                passwordField.getText());
                if (ad.admin_loginValidate())
                {
                    loader.show();
                    login.hide();
                    new java.util.Timer().schedule(new TimerTask()
                    {
                        @Override
                        public void run()
                        {
                            RegisterOperation ro = new
                            RegisterOperation();
                            ResultSet rs =
                            ro.admin_sec_record((txt_userid.getText()));
                            try
                            {
                                if (rs.next())
                                {
                                    Admin a = new Admin();
                                    a.idlabel.setText(rs.getString(1));
                                    a.namelabel.setText(rs.getString(2));
                                }
                            }
                            catch (SQLException e1)
                            {
                                e1.printStackTrace();
                            }
                            frm.dispose();
                        }
                    },1000*5);
                }
                else
                {

```

```

        JOptionPane.showMessageDialog(null, "UserId and
        Password Mismatch");
        loader.hide();
        login.show();
        txt_userid.setText(text);
        passwordField.setText(pwd);
        txt_userid.grabFocus();
    }
}
else if (comboBox.getSelectedItemAt().equals("Receptionist"))
{
    LoginOperation dl = new LoginOperation(txt_userid.getText(),
    passwordField.getText());
    if (dl.rec_loginValidate())
    {
        loader.show();
        login.hide();
        new java.util.Timer().schedule(new TimerTask()
        {
            @Override
            public void run()
            {
                RegisterOperation ro = new
                RegisterOperation();
                ResultSet rs =
                ro.rec_sec_record((txt_userid.getText()));
                try
                {
                    if (rs.next())
                    {
                        RecPortal rf = new RecPortal();
                        rf.idlabel.setText(rs.getString(1));
                        rf.namelabel.setText(rs.getString(2));
                    }
                }
                catch (SQLException e1)
                {
                    e1.printStackTrace();
                }
                frm.dispose();
            }
        }, 1000*5);
    }
    else
    {
        JOptionPane.showMessageDialog(null, "UserId and
        Password Mismatch");
        loader.hide();
        login.show();
        txt_userid.setText(text);
        passwordField.setText(pwd);
    }
}
else if (comboBox.getSelectedItemAt().equals("Doctor"))
{
    LoginOperation dl = new LoginOperation(txt_userid.getText(),
    passwordField.getText());
    if (dl.doc_loginValidate())
    {

```

```

        loader.show();
        login.hide();
        new java.util.Timer().schedule(new TimerTask()
        {
            @Override
            public void run()
            {
                RegisterOperation ro = new RegisterOperation();
                ResultSet rs =
                ro.doc_sec_record((txt_userid.getText()));
                try
                {
                    if (rs.next())
                    {
                        DocPortal doc = new DocPortal();
                        doc.idlabel.setText(rs.getString(1));
                        doc.namelabel.setText(rs.getString(2
                        ));
                    }
                }
                catch (SQLException e1)
                {
                    e1.printStackTrace();
                }
                frm.dispose();
            }
        },1000*5);
    }
    else
    {
        JOptionPane.showMessageDialog(null, "UserId and
        Password Mismatch");
        loader.hide();
        login.show();
        txt_userid.setText(text);
        passwordField.setText(pwd);
        txt_userid.grabFocus();
    }
}
else if (comboBox.getSelectedItem().equals("Employee"))
{
    LoginOperation dl = new LoginOperation(txt_userid.getText(),
    passwordField.getText());
    if (dl.emp_loginValidate())
    {
        loader.show();
        login.hide();
        new java.util.Timer().schedule(new TimerTask()
        {
            @Override
            public void run()
            {
                RegisterOperation ro = new
                RegisterOperation();
                ResultSet rs =
                ro.emp_sec_record((txt_userid.getText()));
                try
                {

```



```

        if (rs.next())
        {
            EmpPortal emp = new EmpPortal();
            emp.idlabel.setText(rs.getString(1));
            emp.namelabel.setText(rs.getString(
                2));
        }
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
    }
    frm.dispose();
}
},1000*5);
}
else
{
    JOptionPane.showMessageDialog(null, "UserId and
    Password Mismatch");
    loader.hide();
    login.show();
    txt_userid.setText(text);
    passwordField.setText(pwd);
    txt_userid.grabFocus();
}
}
}
}
}
}
@SuppressWarnings("deprecation")
public boolean frmvalidate()
{
    // TODO Auto-generated method stub
    flag = true;
    if (txt_userid.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null, "ENTER USER NAME");
        txt_userid.grabFocus();
        flag = false;
    }
    else if (passwordField.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null, "ENTER USER PASSWORD");
        passwordField.grabFocus();
        flag = false;
    }
    return flag;
}
}
}
public class Login
{
    Log l = new Log();
}

```

AdminPortal.Java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
class Admin extends Base implements ActionListener
{
    public JFrame frm;
    public JLabel Fullimg, userlabel, namelabel, idlabel, lblupdateprofile, adminlabel, lbldocicon,
    lblrecicon, lblVeterinaryManagementSystem, lblAdminPortal, barlabel;
    public JButton docbtn, recbtn, empbtn, petbtn, drtimebtn, treatmentbtn, vaccbtn, medbtn,
    servbtn, appbtn, btnlogof, btnExit;
    Admin()
    {
        frm = new JFrame();
        frm.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        Toolkit toolkit = frm.getToolkit();
        Dimension size = toolkit.getScreenSize();
        frm.setSize(size.width, size.height);
        frm.setLayout(null);
        frm.setResizable(false);
        frm.setUndecorated(true);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int framelocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, framelocationY);
        frm.getContentPane().setBackground(Color.WHITE);
        frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
        adminlabel = new JLabel("");
        Image adminimg = new
        ImageIcon(this.getClass().getResource("/images/admin.png")).getImage();
        adminlabel.setIcon(new ImageIcon(adminimg));
        adminlabel.setBounds(0, 5, 128, 128);
        frm.add(adminlabel);
        lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
        lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
        lblVeterinaryManagementSystem.setForeground(Color.WHITE);
        lblVeterinaryManagementSystem.setBounds(350, 22, 632, 49);
        frm.add(lblVeterinaryManagementSystem);
        Canvas canvas = new Canvas();
        canvas.setBackground(Color.WHITE);
        canvas.setForeground(Color.RED);
        canvas.setBounds(350, 71, 630, 4);
        frm.add(canvas);
        lblAdminPortal = new JLabel("Admin Portal");
        lblAdminPortal.setForeground(Color.WHITE);
        lblAdminPortal.setFont(new Font("Magneto", Font.BOLD, 27));
        lblAdminPortal.setBounds(593, 88, 204, 27);
        frm.add(lblAdminPortal);
        userlabel = new JLabel();
        Image userimg = new
        ImageIcon(this.getClass().getResource("/images/usernew.png")).getImage();
        userlabel.setIcon(new ImageIcon(userimg));
        userlabel.setBounds(30, 145, 200, 200);
```

```

frm.add(userlabel);
namelabel = new JLabel();
namelabel.setFont(new Font("Agency FB", Font.BOLD, 32));
namelabel.setBounds(250, 160, 300, 60);
frm.add(namelabel);
idlabel = new JLabel();
userlabel = new JLabel("Admin");
userlabel.setFont(new Font("Agency FB", Font.BOLD, 32));
userlabel.setBounds(250, 230, 200, 40);
frm.add(userlabel);
lblupdateprofile = new JLabel("Update Profile");
lblupdateprofile.setForeground(new Color(128, 0, 0));
lblupdateprofile.setFont(new Font("Agency FB", Font.BOLD, 32));
lblupdateprofile.setBounds(250, 280, 200, 40);
lblupdateprofile.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblupdateprofile.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        RegisterOperation ro = new RegisterOperation();
        ResultSet rs = ro.admin_sec_record((idlabel.getText()));
        try
        {
            if (rs.next())
            {
                AdminUpdate au = new AdminUpdate();
                au.txtUser_Id.setText(rs.getString(1));
                au.txtName.setText(rs.getString(2));
                switch (rs.getString(3))
                {
                    case "NICK NAME":
                        au.cmbSecQueston.setSelectedIndex(0);
                        break;
                    case "FAVOURITE COLOR":
                        au.cmbSecQueston.setSelectedIndex(1);
                        break;
                    case "FAVOURITE FOOD":
                        au.cmbSecQueston.setSelectedIndex(2);
                        break;
                    case "FAVOURITE PLACE":
                        au.cmbSecQueston.setSelectedIndex(3);
                        break;
                    default:
                        break;
                }
                au.cmbSecQueston.setSelectedItem(rs.getString(3));
                au.txtsAnswer.setText(rs.getString(4));
                au.txtPASS.setText(rs.getString(5));
                au.txt_ConfPASS.setText(rs.getString(5));
            }
        }
        catch (SQLException e1)
        {
        }
    }
});
frm.add(lblupdateprofile);
Canvas canvaschpass = new Canvas();

```

```

canvaschpass.setBackground(new Color(128, 0, 0));
canvaschpass.setBounds(250, 325, 150, 4);
frm.add(canvaschpass);
barlabel = new JLabel("");
Image barimg = new
ImageIcon(this.getClass().getResource("/images/BarNew.png")).getImage();
barlabel.setIcon(new ImageIcon(barimg));
barlabel.setBounds(91, 19, 1293, 102);
frm.add(barlabel);
docbtn = new JButton();
Image docimg = new
ImageIcon(this.getClass().getResource("/images/doctor.png")).getImage();
docbtn.setIcon(new ImageIcon(docimg));
docbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
docbtn.setBounds(20, 470, 245, 60);
docbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
docbtn.addActionListener(this);
frm.add(docbtn);
rechtn = new JButton();
Image recimg = new
ImageIcon(this.getClass().getResource("/images/Receptionist1.png")).getImage();
rechtn.setIcon(new ImageIcon(recimg));
rechtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
rechtn.setBounds(330, 470, 245, 60);
rechtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
rechtn.addActionListener(this);
frm.add(rechtn);
empbtn = new JButton();
Image empimg = new
ImageIcon(this.getClass().getResource("/images/Employee.png")).getImage();
empbtn.setIcon(new ImageIcon(empimg));
empbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
empbtn.setBounds(650, 470, 245, 60);
empbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
empbtn.addActionListener(this);
frm.add(empbtn);
petbtn = new JButton();
Image petimg = new
ImageIcon(this.getClass().getResource("/images/petsrecord.png")).getImage();
petbtn.setIcon(new ImageIcon(petimg));
petbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
petbtn.setBounds(20, 570, 245, 60);
petbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
petbtn.addActionListener(this);
frm.add(petbtn);
appbtn = new JButton();
Image appimg = new
ImageIcon(this.getClass().getResource("/images/appointment.png")).getImage();
appbtn.setIcon(new ImageIcon(appimg));
appbtn.setBounds(330, 570, 245, 60);
appbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
appbtn.addActionListener(this);
frm.add(appbtn);
treatmentbtn = new JButton();
Image drugimg = new
ImageIcon(this.getClass().getResource("/images/treatment.png")).getImage();
treatmentbtn.setIcon(new ImageIcon(drugimg));
treatmentbtn.setBounds(650, 570, 245, 60);
treatmentbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));

```

```

treatmentbtn.addActionListener(this);
frm.add(treatmentbtn);
medbtn = new JButton();
Image medimg = new
ImageIcon(this.getClass().getResource("/images/medicine.png")).getImage();
medbtn.setIcon(new ImageIcon(medimg));
medbtn.setBounds(20, 670, 245, 60);
medbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
medbtn.addActionListener(this);
frm.add(medbtn);
vaccbtn = new JButton();
Image vaccimg = new
ImageIcon(this.getClass().getResource("/images/vaccination.png")).getImage();
vaccbtn.setIcon(new ImageIcon(vaccimg));
vaccbtn.setBounds(330, 670, 245, 60);
vaccbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
vaccbtn.addActionListener(this);
frm.add(vaccbtn);
servbtn = new JButton();
Image servimg = new
ImageIcon(this.getClass().getResource("/images/services.png")).getImage();
servbtn.setIcon(new ImageIcon(servimg));
servbtn.setBounds(650, 670, 245, 60);
servbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
servbtn.addActionListener(this);
frm.add(servbtn);
Fullimg = new JLabel();
Image img5 = new
ImageIcon(this.getClass().getResource("/images/ad_bg.png")).getImage();
Fullimg.setIcon(new ImageIcon(img5));
Fullimg.setBounds(0, 0, framewidth, frameheigth);
frm.add(Fullimg);
btnlogof = new JButton();
Image logoffimg = new
ImageIcon(this.getClass().getResource("/images/logog.jpg")).getImage();
btnlogof.setBounds(1280, 725, 35, 35);
btnlogof.setIcon(new ImageIcon(logoffimg));
Fullimg.add(btnlogof);
btnlogof.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnlogof.addActionListener(this);
btnExit = new JButton();
Image exitimg = new
ImageIcon(this.getClass().getResource("/images/Exit.jpg")).getImage();
btnExit.setBounds(1322, 725, 35, 35);
btnExit.setIcon(new ImageIcon(exitimg));
Fullimg.add(btnExit);
btnExit.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnExit.addActionListener(this);
frm.setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e)
{
    if (e.getSource().equals(docbtn))
    {
        new Doctor();
    }
    else if (e.getSource().equals(recbtn))
    {

```

```

        new Receptionist();
    }
    else if (e.getSource().equals(empbtn))
    {
        new Employee();
    }
    else if (e.getSource().equals(petbtn))
    {
        new AnimalZone();
    }
    else if (e.getSource().equals(appbtn))
    {
        new AppointmentZone();
    }
    else if (e.getSource().equals(treatmentbtn))
    {
        new Treatment();
    }
    else if (e.getSource().equals(medbtn))
    {
        new Medicine();
    }
    else if (e.getSource().equals(vaccbtn))
    {
        new Vaccination();
    }
    else if (e.getSource().equals(servbtn))
    {
        new Service();
    }
    else if (e.getSource().equals(btnlogof))
    {
        new Login();
        frm.dispose();
    }
    else if (e.getSource().equals(btnExit))
    {
        System.exit(0);
    }
}
}
public class AdminPortal
{
}

```

Doctor.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.text.*;
import java.util.Date;
import javax.swing.*;
import javax.swing.table.*;
import com.toedter.calendar.JDateChooser;
import net.proteanit.sql.DbUtils;
class Doc extends ValidateForm implements ActionListener
{

```

```

public JFrame frm;
public PreparedStatement ps;
public ResultSet rs;
public Connection conn;
public JButton btnNEW, btnUploade, btnEDIT, btnCLAR;
public DefaultTableModel jtable;
public JTable tabview;
public JLabel lblId, lbluser, lblsAnswer, lblpass, lblconfpass, lblSearch, barlabel,
lblVeterinaryManagementSystem, lblsQueston, lblDoctorRegister, lblQualification, lblEmailId,
lblSpez, lblName, lblDOB, lblGender, lblFull_Address, lblMob_No, lblimg;
public JTextField txtId, txtUser_Id, txtsAnswer, txtName, txtimg, txtDOB, txtSearch,
txtQualification, txtMob_No, txtEmailId, txtSpez;
public JComboBox<Object> cmbGender, cmbSecQuestion;
public JTextArea txtFull_Address;
public JPasswordField txtPASS, txt_ConfPASS;
public Color hexa1;
public JScrollPane js, jsAddress;
public TableRowSorter<TableModel> rowSorter;
public JDateChooser dateChooser;
String arr[] = { "MALE", "FEMALE" };
String arr1[] = { "NICK NAME", "FAVOURITE COLOR", "FAVOURITE FOOD", "FAVOURITE
PLACE" };
Font f = new Font("Consolas", Font.BOLD, 13);
Doc()
{
    frm = new JFrame();
    frm.setSize(1300, 630);
    Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
    int framewidth = frm.getSize().width;
    int frameheight = frm.getSize().height;
    int framelocationX = (dim.width - framewidth) / 2;
    int frmlocationY = (dim.height - frameheight) / 2;
    frm.setLocation(framelocationX, frmlocationY);
    frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
    frm.setResizable(false);
    Color hexa = Color.decode("#BBF2FA");
    frm.getContentPane().setBackground(hexa);
    frm.setLayout(null);
    frm.setTitle("REGISTER FORM");
    hexa1 = Color.decode("#82efff");
    lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
    lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
    lblVeterinaryManagementSystem.setForeground(Color.WHITE);
    lblVeterinaryManagementSystem.setBounds(320, 12, 650, 30);
    frm.add(lblVeterinaryManagementSystem);
    Canvas canvas = new Canvas();
    canvas.setBackground(Color.WHITE);
    canvas.setForeground(Color.RED);
    canvas.setBounds(320, 40, 635, 5);
    frm.add(canvas);
    lblDoctorRegister = new JLabel("Doctor Register");
    lblDoctorRegister.setForeground(Color.WHITE);
    lblDoctorRegister.setFont(new Font("Magneto", Font.BOLD, 23));
    lblDoctorRegister.setBounds(550, 60, 204, 27);
    frm.add(lblDoctorRegister);
    barlabel = new JLabel();
    Image usingimg = new
    ImageIcon(this.getClass().getResource("/images/doc_bar.png")).getImage();

```

```

barlabel.setIcon(new ImageIcon(userimg));
barlabel.setBounds(0, 0, 1300, 105);
frm.add(barlabel);
lblId = new JLabel("ID");
lblId.setBounds(20, 130, 100, 30);
lblId.setFont(f);
lblId.setForeground(Color.BLUE);
frm.add(lblId);
txtId = new JTextField();
txtId.setBounds(150, 130, 220, 30);
txtId.setFont(f);
txtId.setBackground(Color.WHITE);
txtId.setForeground(Color.BLACK);
txtId.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtId);
lblName = new JLabel("NAME");
lblName.setBounds(420, 130, 100, 30);
lblName.setFont(f);
lblName.setForeground(Color.BLUE);
frm.add(lblName);
txtName = new JTextField();
txtName.setBounds(550, 130, 220, 30);
txtName.setFont(f);
txtName.setBackground(Color.WHITE);
txtName.setForeground(Color.BLACK);
txtName.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtName);
lblDOB = new JLabel("DATE OF BIRTH");
lblDOB.setBounds(20, 170, 100, 30);
lblDOB.setFont(f);
lblDOB.setForeground(Color.BLUE);
frm.add(lblDOB);

dateChooser = new JDateChooser();
dateChooser.setBounds(150, 170, 220, 30);
dateChooser.setFont(f);
dateChooser.setBackground(Color.WHITE);
dateChooser.setForeground(Color.BLACK);
dateChooser.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(dateChooser);
lbluser = new JLabel("USER ID");
lbluser.setBounds(420, 170, 80, 30);
lbluser.setFont(f);
lbluser.setForeground(Color.BLUE);
frm.add(lbluser);
txtUser_Id = new JTextField();
txtUser_Id.setBounds(550, 170, 220, 30);
txtUser_Id.setFont(f);
txtUser_Id.setBackground(Color.WHITE);
txtUser_Id.setForeground(Color.BLACK);
txtUser_Id.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtUser_Id);
lblGender = new JLabel("GENDER");
lblGender.setBounds(20, 210, 100, 30);
lblGender.setFont(f);
lblGender.setForeground(Color.BLUE);
frm.add(lblGender);
cmbGender = new JComboBox<Object>(arr);
cmbGender.setBounds(150, 210, 220, 30);

```



```

cmbGender.setFont(f);
cmbGender.setSelectedIndex(-1);
cmbGender.setBackground(Color.WHITE);
cmbGender.setForeground(Color.BLACK);
cmbGender.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(cmbGender);
lblsQuestion = new JLabel("SECURITY QUESTION");
lblsQuestion.setBounds(420, 210, 150, 30);
lblsQuestion.setFont(f);
lblsQuestion.setForeground(Color.BLUE);
frm.add(lblsQuestion);
cmbSecQuestion = new JComboBox<Object>(arr1);
cmbSecQuestion.setBounds(550, 210, 220, 30);
cmbSecQuestion.setFont(f);
cmbSecQuestion.setSelectedIndex(-1);
cmbSecQuestion.setBackground(Color.WHITE);
cmbSecQuestion.setForeground(Color.BLACK);
cmbSecQuestion.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(cmbSecQuestion);
lblQualification = new JLabel("QUALIFICATION");
lblQualification.setBounds(20, 250, 100, 30);
lblQualification.setFont(f);
lblQualification.setForeground(Color.BLUE);
frm.add(lblQualification);
txtQualification = new JTextField();
txtQualification.setBounds(150, 250, 220, 30);
txtQualification.setFont(f);
txtQualification.setBackground(Color.WHITE);
txtQualification.setForeground(Color.BLACK);
txtQualification.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtQualification);
lblsAnswer = new JLabel("ANSWER");
lblsAnswer.setBounds(420, 250, 150, 30);
lblsAnswer.setFont(f);
lblsAnswer.setForeground(Color.BLUE);
frm.add(lblsAnswer);
txtsAnswer = new JTextField();
txtsAnswer.setBounds(550, 250, 220, 30);
txtsAnswer.setFont(f);
txtsAnswer.setBackground(Color.WHITE);
txtsAnswer.setForeground(Color.BLACK);
txtsAnswer.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtsAnswer);
lblSpez = new JLabel("SPECILLIZATION");
lblSpez.setBounds(20, 290, 100, 30);
lblSpez.setFont(f);
lblSpez.setForeground(Color.BLUE);
frm.add(lblSpez);
txtSpez = new JTextField();
txtSpez.setBounds(150, 290, 220, 30);
txtSpez.setFont(f);
txtSpez.setBackground(Color.WHITE);
txtSpez.setForeground(Color.BLACK);
txtSpez.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtSpez);
lblpass = new JLabel("PASSWORD");
lblpass.setBounds(420, 290, 80, 30);
lblpass.setFont(f);
lblpass.setForeground(Color.BLUE);

```

```

frm.add(lblpass);
txtPASS = new JPasswordField();
txtPASS.setBounds(550, 290, 220, 30);
txtPASS.setFont(f);
txtPASS.setBackground(Color.WHITE);
txtPASS.setForeground(Color.BLACK);
txtPASS.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtPASS);
lblMob_No = new JLabel("MOBILE NO.");
lblMob_No.setBounds(20, 330, 100, 30);
lblMob_No.setFont(f);
lblMob_No.setForeground(Color.BLUE);
frm.add(lblMob_No);
txtMob_No = new JTextField();
txtMob_No.setBounds(150, 330, 220, 30);
txtMob_No.setFont(f);
txtMob_No.setBackground(Color.WHITE);
txtMob_No.setForeground(Color.BLACK);
txtMob_No.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtMob_No);
lblconfpas = new JLabel("CONFIRM PASSWORD");
lblconfpas.setBounds(420, 330, 150, 30);
lblconfpas.setFont(f);
lblconfpas.setForeground(Color.BLUE);
frm.add(lblconfpas);
txt_ConfPASS = new JPasswordField();
txt_ConfPASS.setBounds(550, 330, 220, 30);
txt_ConfPASS.setFont(f);
txt_ConfPASS.setBackground(Color.WHITE);
txt_ConfPASS.setForeground(Color.BLACK);
txt_ConfPASS.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txt_ConfPASS);
lblimg = new JLabel("IMAGE");
lblimg.setBounds(20, 370, 100, 30);
lblimg.setFont(f);
lblimg.setForeground(Color.BLUE);
frm.add(lblimg);
txtimg = new JTextField();
txtimg.setBounds(150, 370, 450, 30);
txtimg.setFont(f);
txtimg.setBackground(Color.WHITE);
txtimg.setForeground(Color.BLACK);
frm.add(txtimg);
btnUploade = new JButton("UPLOAD");
btnUploade.setBounds(610, 370, 160, 30);
btnUploade.setFont(f);
btnUploade.setForeground(Color.BLUE);
btnUploade.setBackground(hexa1);
btnUploade.setToolTipText("Temporirly,The button doesnot work");
btnUploade.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(btnUploade);
btnUploade.setEnabled(false);
lblEmailId = new JLabel("E-Mail ID");
lblEmailId.setBounds(20, 410, 100, 30);
lblEmailId.setFont(f);
lblEmailId.setForeground(Color.BLUE);
frm.add(lblEmailId);
txtEmailId = new JTextField();
txtEmailId.setBounds(150, 410, 620, 30);

```

```

txtEmailId.setFont(f);
txtEmailId.setBackground(Color.WHITE);
txtEmailId.setForeground(Color.BLACK);
txtEmailId.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtEmailId);
lblFull_Address = new JLabel("FULL ADDRESS");
lblFull_Address.setBounds(20, 450, 100, 30);
lblFull_Address.setFont(f);
lblFull_Address.setForeground(Color.BLUE);
frm.add(lblFull_Address);
jsAddress = new JScrollPane();
txtFull_Address = new JTextArea();
jsAddress.setBounds(150, 450, 620, 60);
txtFull_Address.setFont(f);
txtFull_Address.setBackground(Color.WHITE);
txtFull_Address.setForeground(Color.BLACK);
txtFull_Address.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
jsAddress.setViewportView(txtFull_Address);
frm.add(jsAddress);
btnNEW = new JButton("NEW");
btnNEW.setBounds(20, 530, 235, 35);
btnNEW.setFont(f);
btnNEW.setForeground(Color.BLUE);
btnNEW.setBackground(hexa1);
btnNEW.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnNEW.addActionListener(this);
frm.add(btnNEW);
btnEDIT = new JButton("EDIT");
btnEDIT.setBounds(275, 530, 235, 35);
btnEDIT.setFont(f);
btnEDIT.setForeground(Color.BLUE);
btnEDIT.setBackground(hexa1);
btnEDIT.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnEDIT.addActionListener(this);
frm.add(btnEDIT);
btnCLAR = new JButton("CLEAR");
btnCLAR.setBounds(535, 530, 235, 35);
btnCLAR.setFont(f);
btnCLAR.setForeground(Color.BLUE);
btnCLAR.setBackground(hexa1);
btnCLAR.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnCLAR.addActionListener(this);
frm.add(btnCLAR);
lblSearch = new JLabel("SEARCH");
lblSearch.setBounds(860, 130, 100, 30);
lblSearch.setForeground(Color.BLUE);
lblSearch.setFont(f);
lblSearch.setBackground(Color.BLUE);
frm.add(lblSearch);
jtable = new DefaultTableModel();
tabview = new JTable(jtable);
js = new JScrollPane();
js.setBounds(790, 170, 480, 400);
tabview.setFont(f);
tabview.setForeground(Color.blue);
tabview.setBackground(Color.WHITE);
Object[] columns = { "ID", "NAME", "Date Of Birth", "GENDER", "QUALIFICATION",
"SPECIALIZATION", "MOBILE NO", "EMAIL ID", "ADDRESS" };
jtable.setColumnIdentifiers(columns);

```

```

tabview.setModel(jtable);
tabview.setRowHeight(30);
tabview.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
ShowTable();
tabview.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        int i = tabview.getSelectedRow();
        TableModel model = tabview.getModel();
        txtId.setText(model.getValueAt(i, 0).toString());
        txtName.setText(model.getValueAt(i, 1).toString());
        try
        {
            Date date = new SimpleDateFormat("yyyy-MM-dd").parse(model.getValueAt(i, 2).toString());
            dateChooser.setDate(date);
        }
        catch (ParseException e1)
        {
            e1.printStackTrace();
        }
        cmbGender.setSelectedItem(model.getValueAt(i, 3));
        txtQualification.setText(model.getValueAt(i, 4).toString());
        txtSpez.setText(model.getValueAt(i, 5).toString());
        txtMob_No.setText(model.getValueAt(i, 6).toString());
        txtEmailId.setText(model.getValueAt(i, 7).toString());
        txtFull_Address.setText(model.getValueAt(i, 8).toString());
        txtName.grabFocus();
    }
});
txtSearch = new JTextField();
txtSearch.setBounds(930, 130, 230, 30);
txtSearch.setFont(f);
txtSearch.setBackground(Color.WHITE);
txtSearch.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtSearch.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyReleased(KeyEvent e)
    {
        DefaultTableModel table = (DefaultTableModel) tabview.getModel();
        String search = txtSearch.getText();
        TableRowSorter<DefaultTableModel> tr = new
        TableRowSorter<DefaultTableModel>(table);
        tabview.setRowSorter(tr);
        tr.setRowFilter(RowFilter.regexFilter(search));
    }
});
frm.add(txtSearch);
js.setViewPortView(tabview);
frm.add(js);
disabled();
frm.setVisible(true);
}
public void ShowTable()
{

```

```

String sqlqry = "SELECT
ID,NAME,DOB,GENDER,QUALIFICATION,SPECIALIZATION,MOBILE_NO,EMAILID,ADD
RESS FROM DOCTOR_DETAILS";
conn = Connections.getConnections();
try
{
    ps = conn.prepareStatement(sqlqry);
    rs = ps.executeQuery();
    tabview.setModel(DbUtils.resultSetToTableModel(rs));
}
catch (SQLException e)
{
    e.printStackTrace();
}
}
@SuppressWarnings("deprecation")
@Override
public void actionPerformed(ActionEvent e)
{
    if (e.getSource().equals(btnNEW))
    {
        if (e.getActionCommand().equalsIgnoreCase("NEW"))
        {
            btnNEW.disable();
            enabled();
            getid();
            btnNEW.setText("SAVE");
        }
        else if (e.getActionCommand().equalsIgnoreCase("SAVE"))
        {
            SimpleDateFormat dateFormat = new
            SimpleDateFormat("dd/MMM/yyyy");
            String ddate = dateFormat.format(dateChooser.getDate());
            if (frmvalidate())
            {
                RegisterOperation ro = new RegisterOperation(txtId.getText(),
                txtName.getText().toUpperCase(),ddate,
                cmbGender.getSelectedItem().toString().toUpperCase(),
                txtQualification.getText().toUpperCase(),
                txtSpez.getText().toUpperCase(), txtMob_No.getText(),
                txtEmailId.getText().toLowerCase(),
                txtFull_Address.getText().toUpperCase(), txtUser_Id.getText(),
                cmbSecQestion.getSelectedItem().toString(), txtsAnswer.getText(),
                txtPASS.getText());
                if (ro.doc_userValidate())
                {
                    if (strongPassword(txtPASS.getText()))
                    {
                        if (ro.doc_saverecord() == 1)
                        {
                            JOptionPane.showMessageDialog(null,
                            "USER SUCCESFULL REGISTER");
                            ShowTable();
                            clear();
                            disabled();
                            btnNEW.setText("NEW");
                        }
                    }
                    else
                    {

```

```

JOptionPane.showMessageDialog(null,
"USER NOT REGISTER");
    }
    }
    else
    {
        JOptionPane.showMessageDialog(null,"Week
        Password!Password Must Contain Capital
        Letter,Special Symbol & Digit[N/0/@]");
        txtPASS.setText("");
        txtPASS.grabFocus();
    }
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Record    Already
        Inserted");
    }
    }
}
}
else if (e.getSource().equals(btnEDIT))
{
    if (e.getActionCommand().equalsIgnoreCase("EDIT"))
    {
        btnEDIT.disable();
        editenabled();
        btnEDIT.setText("UPDATE");
    }
    else if (e.getActionCommand().equalsIgnoreCase("UPDATE"))
    {
        SimpleDateFormat dateFormat = new
        SimpleDateFormat("dd/MMM/yyyy");
        String ddate = dateFormat.format(dateChooser.getDate());
        if (frmvalidateforupdate())
        {
            RegisterOperation ro = new
            RegisterOperation(txtId.getText().toUpperCase(),
            txtName.getText().toUpperCase(), ddate,
            cmbGender.getSelectedItem().toString().toUpperCase(),
            txtQualification.getText().toUpperCase(),
            txtSpez.getText().toUpperCase(), txtMob_No.getText(),
            txtEmailId.getText().toLowerCase(),
            txtFull_Address.getText().toUpperCase());
            if (ro.doc_updatebyadmin() == 1)
            {
                JOptionPane.showMessageDialog(null, "SUCESSFULLY
                UPDATED");
                ShowTable();
                btnEDIT.setText("EDIT");
                btnEDIT.setEnabled(true);
                disabled();
                clear();
            }
            else
            {
                JOptionPane.showMessageDialog(null, "NOT UPDATED");
            }
        }
    }
}
}

```

```

        }
    }
    else if (e.getSource().equals(btnCLAR))
    {
        clear();
    }
}
public void getid()
{
    RegisterOperation ro = new RegisterOperation();
    txtId.setText(ro.doc_getid());
}
public void clear()
{
    txtId.setText("");
    txtName.setText("");
    dateChooser.setDate(null);
    txtUser_Id.setText("");
    cmbGender.setSelectedIndex(-1);
    cmbSecQueston.setSelectedIndex(-1);
    txtsAnswer.setText("");
    txtQualification.setText("");
    txtSpez.setText("");
    txtMob_No.setText("");
    txtimg.setText("");
    txtEmailId.setText("");
    txtPASS.setText("");
    txt_ConfPASS.setText("");
    txtFull_Address.setText("");
}
public void editenabled()
{
    txtName.setEditable(true);
    dateChooser.setEnabled(true);
    cmbGender.setEditable(true);
    cmbGender.setEnabled(true);
    txtQualification.setEditable(true);
    txtSpez.setEditable(true);
    txtMob_No.setEditable(true);
    txtEmailId.setEditable(true);
    txtFull_Address.setEditable(true);
}
public void enabled()
{
    txtName.setEditable(true);
    dateChooser.setEnabled(true);
    txtUser_Id.setEditable(true);
    txtMob_No.setEditable(true);
    txtEmailId.setEditable(true);
    cmbGender.setEnabled(true);
    cmbGender.setEditable(true);
    cmbSecQueston.setEnabled(true);
    cmbSecQueston.setEditable(true);
    txtQualification.setEditable(true);
    txtSpez.setEditable(true);
    txtsAnswer.setEditable(true);
    txtPASS.setEditable(true);
    txt_ConfPASS.setEditable(true);
    txtFull_Address.setEditable(true);
}

```

```

}
public void disabled()
{
    txtId.setEditable(false);
    txtName.setEditable(false);
    dateChooser.setEnabled(false);
    txtUser_Id.setEditable(false);
    txtMob_No.setEditable(false);
    txtimg.setEditable(false);
    txtEmailId.setEditable(false);
    cmbGender.setEnabled(false);
    cmbSecQueston.setEnabled(false);
    txtQualification.setEditable(false);
    txtSpez.setEditable(false);
    txtsAnswer.setEditable(false);
    txtPASS.setEditable(false);
    txt_ConfPASS.setEditable(false);
    txtFull_Address.setEditable(false);
}
@SuppressWarnings("deprecation")
public boolean frmvalidate()
{
    boolean flag = true;
    if (ValidateForm.validateAnyString(txtName.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Name ");
        txtName.grabFocus();
        flag = false;
    }
    else if (cmbGender.getSelectedIndex() == -1)
    {
        JOptionPane.showMessageDialog(null, "Choose Your Gender");
        cmbGender.grabFocus();
        flag = false;
    }
    else if (ValidateForm.validateMobileNo(txtMob_No.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Correct phone no..");
        txtMob_No.grabFocus();
        flag = false;
    }
    else if (ValidateForm.validateEmail(txtEmailId.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter email id..");
        txtEmailId.grabFocus();
        flag = false;
    }
    else if (txtUser_Id.getText().equals(""))
    {
        JOptionPane.showMessageDialog(null, "Enter UserId");
        txtUser_Id.grabFocus();
        flag = false;
    }
    else if (cmbSecQueston.getSelectedIndex() == -1)
    {
        JOptionPane.showMessageDialog(null, "Choose Security Question");
        cmbSecQueston.grabFocus();
        flag = false;
    }
}

```



```

else if (cmbSecQueston.getSelectedIndex() == -1)
{
    JOptionPane.showMessageDialog(null, "Choose Security Question");
    cmbSecQueston.grabFocus();
    flag = false;
}
else if (txtFull_Address.getText().length() == 0)
{
    JOptionPane.showMessageDialog(null, "Enter Address");
    cmbSecQueston.grabFocus();
    flag = false;
}
else if (txtPASS.getText().length() == 0)
{
    JOptionPane.showMessageDialog(null, "ENTER PASSWORD");
    txtPASS.grabFocus();
    flag = false;
}
else if (txt_ConfPASS.getText().length() == 0)
{
    JOptionPane.showMessageDialog(null, "Enter Confirm Password");
    txt_ConfPASS.grabFocus();
    flag = false;
}
else if (!txtPASS.getText().equals(txt_ConfPASS.getText()))
{
    JOptionPane.showMessageDialog(null, "PASSWORD AND CONFIRM PASSWORD
DO NOT MATCH");
    txtPASS.grabFocus();
    flag = false;
}
return flag;
}
public boolean frmvalidateforupdate()
{
    boolean flag = true;
    if (ValidateForm.validateAnyString(txtName.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Name ");
        txtName.grabFocus();
        flag = false;
    }
    else if (cmbGender.getSelectedIndex() == -1)
    {
        JOptionPane.showMessageDialog(null, "Choose Your Gender");
        cmbGender.grabFocus();
        flag = false;
    }
    else if (ValidateForm.validateAnyString(txtQualification.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Qualification ");
        txtName.grabFocus();
        flag = false;
    }
    else if (ValidateForm.validateAnyString(txtSpez.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Specialization ");
        txtName.grabFocus();
    }
}

```

```

        flag = false;
    }
    else if (ValidateForm.validateMobileNo(txtMob_No.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter Correct phone no..");
        txtMob_No.grabFocus();
        flag = false;
    }
    else if (ValidateForm.validateEmail(txtEmailId.getText()) == false)
    {
        JOptionPane.showMessageDialog(null, "Enter email id..");
        txtEmailId.grabFocus();
        flag = false;
    }
    else if (txtFull_Address.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Address");
        cmbSecQueston.grabFocus();
        flag = false;
    }
    return flag;
}
public boolean strongPassword(String pass)
{
    boolean sflag = false;
    int spacecount, digitcount, capcount;
    spacecount = digitcount = capcount = 0;
    char[] ch = pass.toCharArray();
    for (char c : ch)
    {
        if (c >= 65 && c <= 90)
        {
            capcount++;
        }
        if (c > 48 && c <= 57)
        {
            digitcount++;
        }
        if (!(c >= 48 && c <= 57) && !(c >= 65 && c <= 90) && !(c >= 97 && c <= 122))
        {
            spacecount++;
        }
        if (spacecount > 0 && digitcount > 0 && capcount > 0)
        {
            sflag = true;
        }
    }
    return sflag;
}
}
public class Doctor
{
    Doc d = new Doc();
}

```

AnimalZone.Java

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.text.SimpleDateFormat;
import javax.swing.*;
import javax.swing.table.*;
import net.proteanit.sql.DbUtils;

final class AnimZone extends Base
{
    public PreparedStatement ps;
    public ResultSet rs;
    public Connection conn;
    public DefaultTableModel jtable = new DefaultTableModel();
    public JFrame frm;
    public JLabel lblVeterinaryManagementSystem, lblPatientDetails, barlabel, lblSearch;
    public JTextField txtSearch;
    public JTable tabview;

    AnimZone()
    {
        frm = new JFrame();
        frm.setSize(800, 650);
        frm.setResizable(false);
        frm.setLayout(null);
        frm.setResizable(false);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int framelocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, framelocationY);
        frm.getContentPane().setBackground(Color.WHITE);
        frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
        frm.setTitle("Patient Database");
        Color hexa = Color.decode("#c1fff0");
        frm.getContentPane().setBackground(hexa);
        Font f = new Font("Consolas", Font.BOLD, 13);
        lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
        lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
        lblVeterinaryManagementSystem.setForeground(Color.WHITE);
        lblVeterinaryManagementSystem.setBounds(80, 12, 650, 30);
        frm.add(lblVeterinaryManagementSystem);
        Canvas canvas = new Canvas();
        canvas.setBackground(Color.WHITE);
        canvas.setForeground(Color.RED);
        canvas.setBounds(80, 40, 635, 5);
        frm.add(canvas);
        lblPatientDetails = new JLabel("Patient Details");
        lblPatientDetails.setForeground(Color.WHITE);
        lblPatientDetails.setFont(new Font("Magneto", Font.BOLD, 23));
        lblPatientDetails.setBounds(300, 60, 300, 27);
        frm.add(lblPatientDetails);
        barlabel = new JLabel();
        Image usingimg = new
        ImageIcon(this.getClass().getResource("/images/recbar.png")).getImage();
```

```

barlabel.setIcon(new ImageIcon(userimg));
barlabel.setBounds(0, 0, 800, 105);
frm.add(barlabel);
lblSearch = new JLabel("SEARCH");
lblSearch.setBounds(460, 130, 100, 30);
lblSearch.setForeground(Color.BLUE);
lblSearch.setFont(f);
lblSearch.setBackground(Color.BLUE);
frm.add(lblSearch);
tabview = new JTable(jtable);
JScrollPane js = new JScrollPane(tabview);
js.setBounds(8, 180, 770, 425);
tabview.setFont(f);
tabview.setForeground(Color.blue);
Object[] columns = { "DATE", "REGISTRATION NO.", "OWNER", "ANIMAL NAME", "AGE",
"SPECIES", "MOBILE NO", "EMAIL ID", "ADDRESS", "MEDICAL HISTORY" };
jtable.setColumnIdentifiers(columns);
tabview.setModel(jtable);
tabview.setRowHeight(30);
tabview.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
tabview.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        int i = tabview.getSelectedRow();
        TableModel model = tabview.getModel();
        ProfileAnimal pa = new ProfileAnimal();
        try
        {
            String regdate = new SimpleDateFormat("MMM dd,
            yyyy").format(model.getValueAt(i, 0));
            pa.dateChooser.setText(regdate);
        }
        catch (Exception e1)
        {
        }
        pa.txtregistration.setText(model.getValueAt(i, 1).toString());
        pa.txtName.setText(model.getValueAt(i, 2).toString());
        pa.txtAnimalName.setText(model.getValueAt(i, 3).toString());
        pa.cmbGender.setText(model.getValueAt(i, 4).toString());
        pa.txtAge.setText(model.getValueAt(i, 5).toString());
        pa.cmbSpecies.setText(model.getValueAt(i, 6).toString());
        pa.txtBreed.setText(model.getValueAt(i, 7).toString());
        pa.txtHeight.setText(model.getValueAt(i, 8).toString());
        pa.txtWeight.setText(model.getValueAt(i, 9).toString());
        pa.txtMob_No.setText(model.getValueAt(i, 10).toString());
        pa.txtEmailId.setText(model.getValueAt(i, 11).toString());
        pa.txtFull_Address.setText(model.getValueAt(i, 12).toString());
        pa.cmbState.setText(model.getValueAt(i, 13).toString());
        pa.txtMedicalHistory.setText(model.getValueAt(i, 14).toString());
    }
});
txtSearch = new JTextField();
txtSearch.setBounds(530, 130, 240, 30);
txtSearch.setFont(f);
txtSearch.setBackground(Color.WHITE);
txtSearch.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtSearch.addKeyListener(new KeyAdapter()

```

```

        {
            @Override
            public void keyReleased(KeyEvent e)
            {
                DefaultTableModel table = (DefaultTableModel) tabview.getModel();
                String search = txtSearch.getText();
                TableRowSorter<DefaultTableModel> tr = new TableRowSorter<>(table);
                tabview.setRowSorter(tr);
                tr.setRowFilter(RowFilter.regexFilter(search));
            }
        });
        frm.add(txtSearch);
        frm.add(js);
        ShowTable();
        frm.setVisible(true);
    }
    public void ShowTable()
    {
        String sqlqry = "SELECT
        REG_DATE,REG_NO,OWNER_NAME,PET_NAME,GENDER,AGE,SPECIES,BREED,HEIGHT,WEIGHT,MOBILE_NO,EMAILID,ADDRESS,STATE,MEDICAL_HISTORY FROM
        ANIMAL_DETAILS";
        conn = Connections.getConnections();
        try
        {
            ps = conn.prepareStatement(sqlqry);
            rs = ps.executeQuery();
            tabview.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch (SQLException e)
        {
        }
    }
}
public class AnimalZone
{
    AnimZone az = new AnimZone();
}

```

Medicine.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.*;
import net.proteanit.sql.DbUtils;
class Med extends ValidateForm implements ActionListener, WindowListener
{
    public JFrame frm;
    public PreparedStatement ps;
    public ResultSet rs;
    public Connection conn;
    public JLabel lblmid, lblVeterinaryManagementSystem, lblmedname, lblSearch,
    lblMedicinePortal, lbldrugs, barlabel, lbltype, lblcomp;
    public JTextField txtmid, txtmedname, txtdosremarks, txtduration, txtcompny, txtSearch;
    public JComboBox<String> cmbdrug, cmbtype, cmbmorning, cmbnoon, cmbeve;
    public JButton btnNew, btndrugs, btnEDIT, btnClear;
}

```

```

public DefaultTableModel jTable;
public JTable tabview;
public String drugstype[] = { "TABLETS", "CAPSULE", "SYRUP", "POWDER", "INJECTION" };
public TableRowSorter<TableModel> rowSorter;
public String strdrugs, strtype, strdosage;
Font f = new Font("consolas", Font.BOLD, 14);
Med()
{
    frm = new JFrame();
    frm.setSize(720, 370);
    frm.setResizable(false);
    frm.setLayout(null);
    frm.setResizable(false);
    Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
    int framewidth = frm.getSize().width;
    int frameheight = frm.getSize().height;
    int framelocationX = (dim.width - framewidth) / 2;
    int framelocationY = (dim.height - frameheight) / 2;
    frm.setLocation(framelocationX, framelocationY);
    frm.getContentPane().setBackground(Color.WHITE);
    frm.setTitle("Medicine Database");
    Color hexa = Color.decode("#BBF2FA");
    frm.getContentPane().setBackground(hexa);
    frm.addWindowListener(this);
    frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
    lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
    lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 18));
    lblVeterinaryManagementSystem.setForeground(Color.WHITE);
    lblVeterinaryManagementSystem.setBounds(180, 12, 370, 20);
    frm.add(lblVeterinaryManagementSystem);
    Canvas canvas = new Canvas();
    canvas.setBackground(Color.WHITE);
    canvas.setForeground(Color.RED);
    canvas.setBounds(180, 33, 335, 2);
    frm.add(canvas);
    lblMedicinePortal = new JLabel("Medicine Details");
    lblMedicinePortal.setForeground(Color.WHITE);
    lblMedicinePortal.setFont(new Font("Magneto", Font.BOLD, 17));
    lblMedicinePortal.setBounds(280, 40, 204, 27);
    frm.add(lblMedicinePortal);
    barlabel = new JLabel();
    Image usingimg = new
    ImageIcon(this.getClass().getResource("/images/emp_bar.png")).getImage();
    barlabel.setIcon(new ImageIcon(usingimg));
    barlabel.setBounds(0, 0, 830, 70);
    frm.add(barlabel);
    lblmid = new JLabel("MEDICINE ID");
    lblmid.setBounds(20, 80, 90, 30);
    frm.add(lblmid);
    lblmid.setFont(f);
    lblmid.setForeground(Color.blue);
    txtmid = new JTextField();
    txtmid.setBounds(140, 80, 200, 30);
    frm.add(txtmid);
    txtmid.setEditable(false);
    txtmid.setBackground(Color.WHITE);
    txtmid.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    txtmid.setFont(f);
}

```

```

lblmedname = new JLabel("MEDICINE NAME");
lblmedname.setBounds(20, 120, 110, 30);
frm.add(lblmedname);
lblmedname.setFont(f);
lblmedname.setForeground(Color.blue);
txtmedname = new JTextField();
txtmedname.setBounds(140, 120, 200, 30);
txtmedname.setBackground(Color.WHITE);
txtmedname.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtmedname);
txtmedname.setFont(f);
lbldrugs = new JLabel("DRUGS");
lbldrugs.setBounds(20, 160, 90, 30);
frm.add(lbldrugs);
lbldrugs.setFont(f);
lbldrugs.setForeground(Color.blue);
cmbdrug = new JComboBox<String>();
cmbdrug.setBounds(140, 160, 80, 30);
cmbdrug.setBackground(Color.WHITE);
cmbdrug.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(cmbdrug);
cmbdrug.setFont(f);
cmbdrug.setEditable(true);
btndrugs = new JButton("ADD DRUGS");
btndrugs.setBounds(230, 160, 110, 30);
btndrugs.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(btndrugs);
btndrugs.addActionListener(this);
btndrugs.setBackground(Color.decode("#88d2f6"));
lbltype = new JLabel("TYPE");
lbltype.setBounds(20, 200, 80, 30);
frm.add(lbltype);
lbltype.setFont(f);
lbltype.setForeground(Color.blue);
cmbtype = new JComboBox<String>(drugstype);
cmbtype.setBounds(140, 200, 200, 30);
cmbtype.setBackground(Color.WHITE);
cmbtype.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
cmbtype.setSelectedIndex(-1);
frm.add(cmbtype);
cmbtype.setFont(f);
cmbtype.setEditable(true);
lblcomp = new JLabel("COMPANY NAME");
lblcomp.setBounds(20, 240, 100, 30);
frm.add(lblcomp);
lblcomp.setFont(f);
lblcomp.setForeground(Color.blue);
txtcompny = new JTextField();
txtcompny.setBounds(140, 240, 200, 30);
txtcompny.setBackground(Color.WHITE);
txtcompny.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtcompny);
txtcompny.setFont(f);
btnNew = new JButton("NEW");
btnNew.setBounds(20, 290, 100, 30);
btnNew.setBackground(Color.decode("#88d2f6"));
btnNew.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnNew.addActionListener(this);
frm.add(btnNew);

```

```

btnEDIT = new JButton("EDIT");
btnEDIT.setBounds(130, 290, 100, 30);
btnEDIT.setBackground(Color.decode("#88d2f6"));
btnEDIT.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnEDIT.addActionListener(this);
frm.add(btnEDIT);
btnClear = new JButton("CLEAR");
btnClear.setBounds(240, 290, 100, 30);
btnClear.setBackground(Color.decode("#88d2f6"));
btnClear.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnClear.addActionListener(this);
frm.add(btnClear);
lblSearch = new JLabel("SEARCH");
lblSearch.setBounds(370, 80, 125, 30);
lblSearch.setForeground(Color.BLUE);
lblSearch.setFont(f);
lblSearch.setBackground(Color.BLUE);
frm.add(lblSearch);
jtable = new DefaultTableModel();
tabview = new JTable(jtable);
JScrollPane js = new JScrollPane(tabview);
js.setBounds(350, 120, 340, 200);
tabview.setFont(f);
tabview.setForeground(Color.blue);
tabview.setBackground(Color.WHITE);
Object[] columns = { "Id", "MEDICINE", "DRUGS", "TYPE", "COMPANY" };
jtable.setColumnIdentifiers(columns);
tabview.setModel(jtable);
tabview.setRowHeight(30);
tabview.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
ShowTable();
tabview.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        int i = tabview.getSelectedRow();
        TableModel model = tabview.getModel();
        txtmid.setText(model.getValueAt(i, 0).toString());
        txtmedname.setText(model.getValueAt(i, 1).toString());
        cmbdrug.setSelectedItem(model.getValueAt(i, 2));
        cmbtype.setSelectedItem(model.getValueAt(i, 3));
        txtcompny.setText(model.getValueAt(i, 4).toString());
        txtmedname.grabFocus();
    }
});
txtSearch = new JTextField();
txtSearch.setBounds(450, 80, 230, 30);
txtSearch.setFont(f);
txtSearch.setBackground(Color.WHITE);
txtSearch.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtSearch.addKeyListener(new KeyAdapter()
{
    @Override
    public void keyReleased(KeyEvent e)
    {
        DefaultTableModel table = (DefaultTableModel) tabview.getModel();
        String search = txtSearch.getText();
    }
}

```



```

        TableRowSorter<DefaultTableModel> tr = new
        TableRowSorter<DefaultTableModel>(table);
        tabview.setRowSorter(tr);
        tr.setRowFilter(RowFilter.regexFilter(search));
    }
});
frm.add(txtSearch);
frm.add(js);
disabled();
frm.setVisible(true);
}
public void ShowTable()
{
    String sqlqry = "SELECT ID,NAME,DRUGS,TYPE,COMPANY FROM MEDICINE_DETAILS
    ORDER BY ID ASC";
    conn = Connections.getConnections();
    try
    {
        ps = conn.prepareStatement(sqlqry);
        rs = ps.executeQuery();
        tabview.setModel(DbUtils.resultSetToTableModel(rs));
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
@SuppressWarnings("deprecation")
@Override
public void actionPerformed(ActionEvent e)
{
    if (e.getSource().equals(btnNew))
    {
        if (e.getActionCommand().equalsIgnoreCase("NEW"))
        {
            btnNew.disable();
            enabled();
            getId();
            btnNew.setText("SAVE");
        }
        else if (e.getActionCommand().equalsIgnoreCase("SAVE"))
        {
            if (frmvalidate())
            {
                MedicalOperations mo = new
                MedicalOperations(txtmid.getText().toUpperCase(),txtmedname.get
                Text().toUpperCase(),
                cmbdrug.getSelectedItem().toString().toUpperCase(),cmbtype.getSe
                lectedItem().toString().toUpperCase(),
                txtcompny.getText().toUpperCase());
                if (mo.Med_userValidate())
                {
                    if (mo.Med_saveRecord() == 1)
                    {
                        JOptionPane.showMessageDialog(null, "ADD
                        SUCCESSFULLY");
                        ShowTable();
                        clear();
                        disabled();
                    }
                }
            }
        }
    }
}

```

```

        btnNew.setText("NEW");
    }
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Record
        Already Inserted");
    }
}
}
else if (e.getSource().equals(btnEDIT))
{
    if (e.getActionCommand().equalsIgnoreCase("EDIT"))
    {
        btnEDIT.disable();
        enabled();
        btnEDIT.setText("UPDATE");
    }
    else if (e.getActionCommand().equalsIgnoreCase("UPDATE"))
    {
        if (frmvalidate())
        {
            MedicalOperations mo = new
            MedicalOperations(txtmid.getText().toUpperCase(),txtmedname.get
            Text().toUpperCase(),
            cmbdrug.getSelectedItem().toString().toUpperCase(),cmbtype.getSe
            lectedItem().toString().toUpperCase(),
            txtcompny.getText().toUpperCase());
            if (mo.Med_update() == 1)
            {
                JOptionPane.showMessageDialog(null, "SUCESSFULLY
                UPDATED");
                ShowTable();
                btnEDIT.setText("EDIT");
                btnEDIT.setEnabled(true);
                disabled();
                clear();
            }
            else
            {
                JOptionPane.showMessageDialog(null, "NOT UPDATED");
            }
        }
    }
}
else if (e.getSource().equals(btnClear))
{
    clear();
}
else if (e.getSource().equals(btndrugs))
{
    new Drugs();
}
}
public void clear()
{
    txtmid.setText("");
    txtmedname.setText("");
}

```

```

        cmbdrug.setSelectedIndex(-1);
        cmbtype.setSelectedIndex(-1);
        txtcompny.setText("");
    }
    public void disabled()
    {
        txtmid.setEditable(false);
        txtmedname.setEditable(false);
        cmbdrug.setEnabled(false);
        cmbtype.setEnabled(false);
        txtcompny.setEditable(false);
    }
    public void enabled()
    {
        txtmedname.setEditable(true);
        cmbdrug.setEnabled(true);
        cmbtype.setEnabled(true);
        txtcompny.setEditable(true);
    }
    public boolean frmvalidate()
    {
        boolean flag = true;
        if (ValidateForm.validateAnyString(txtmedname.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Medicine Name ");
            txtmedname.grabFocus();
            flag = false;
        }
        else if (cmbdrug.getSelectedIndex() == -1)
        {
            JOptionPane.showMessageDialog(null, "Choose Your Drugs");
            cmbdrug.grabFocus();
            flag = false;
        }
        else if (cmbtype.getSelectedIndex() == -1)
        {
            JOptionPane.showMessageDialog(null, "Choose Type");
            cmbtype.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateAnyString(txtcompny.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Company Name ");
            txtcompny.grabFocus();
            flag = false;
        }
        return flag;
    }
    public void srccour()
    {
        String sqlqry = "select DRUGS_NAME from DRUGS_DETAILS";
        conn = Connections.getConnections();
        try
        {
            ps = conn.prepareStatement(sqlqry);
            rs = ps.executeQuery();
            while (rs.next())
            {
                cmbdrug.addItem(rs.getString(1));
            }
        }
    }

```

```

        cmbdrug.setSelectedIndex(-1);
    }
}
catch (SQLException e)
{
    e.printStackTrace();
}
}
public void getid()
{
    MedicalOperations mo = new MedicalOperations();
    txtmid.setText(mo.Med_getid());
}
@Override
public void windowActivated(WindowEvent e)
{
    srccour();
}
@Override
public void windowClosed(WindowEvent e)
{
}
@Override
public void windowClosing(WindowEvent e)
{
}
@Override
public void windowDeactivated(WindowEvent e)
{
    cmbdrug.removeAllItems();
}
@Override
public void windowDeiconified(WindowEvent e)
{
}
@Override
public void windowIconified(WindowEvent e)
{
    // TODO Auto-generated method stub
}
@Override
public void windowOpened(WindowEvent e)
{
}
}
public class Medicine
{
    Med am = new Med();
}

```

ReceptionistPortal.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;

```

```

import java.util.Date;
import javax.swing.*;
class RecPortal extends Base implements ActionListener
{
    public JFrame frm;
    public JLabel Fullimg, userlabel, idlabel, namelabel, lblupdateprofile, adminlabel,
    lblVeterinaryManagementSystem, lblAdminPortal, barlabel;
    public JButton oldpatientbtn, newpatientbtn, btnlogof, btnExit;
    RecPortal()
    {
        frm = new JFrame();
        frm.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        Toolkit toolkit = frm.getToolkit();
        Dimension size = toolkit.getScreenSize();
        frm.setSize(size.width, size.height);
        frm.setLayout(null);
        frm.setResizable(false);
        frm.setUndecorated(true);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int framelocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, framelocationY);
        frm.getContentPane().setBackground(Color.WHITE);
        frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
        adminlabel = new JLabel("");
        Image adminimg = new
        ImageIcon(this.getClass().getResource("/images/recep_icon.png")).getImage();
        adminlabel.setIcon(new ImageIcon(adminimg));
        adminlabel.setBounds(0, 5, 128, 128);
        frm.add(adminlabel);
        lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
        lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
        lblVeterinaryManagementSystem.setForeground(Color.WHITE);
        lblVeterinaryManagementSystem.setBounds(350, 22, 632, 49);
        frm.add(lblVeterinaryManagementSystem);
        Canvas canvas = new Canvas();
        canvas.setBackground(Color.WHITE);
        canvas.setForeground(Color.RED);
        canvas.setBounds(350, 71, 630, 4);
        frm.add(canvas);
        lblAdminPortal = new JLabel("Receptionist Portal");
        lblAdminPortal.setForeground(Color.WHITE);
        lblAdminPortal.setFont(new Font("Magneto", Font.BOLD, 27));
        lblAdminPortal.setBounds(560, 88, 300, 27);
        frm.add(lblAdminPortal);
        userlabel = new JLabel();
        Image userimg = new
        ImageIcon(this.getClass().getResource("/images/usernew.png")).getImage();
        userlabel.setIcon(new ImageIcon(userimg));
        userlabel.setBounds(30, 145, 200, 200);
        frm.add(userlabel);
        namelabel = new JLabel();
        namelabel.setFont(new Font("Agency FB", Font.BOLD, 32));
        namelabel.setBounds(250, 160, 300, 60);
        frm.add(namelabel);
        userlabel = new JLabel("Receptionist");
    }
}

```

```

userlabel.setFont(new Font("Agency FB", Font.BOLD, 32));
userlabel.setBounds(250, 220, 200, 40);
frm.add(userlabel);
idlabel = new JLabel();
lblupdateprofile = new JLabel("Update Profile");
lblupdateprofile.setForeground(new Color(128, 0, 0));
lblupdateprofile.setFont(new Font("Agency FB", Font.BOLD, 32));
lblupdateprofile.setBounds(250, 280, 200, 40);
lblupdateprofile.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
lblupdateprofile.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        RegisterOperation ro = new RegisterOperation();
        ResultSet rs = ro.rec_sec_recordupdate((idlabel.getText()));
        try
        {
            if (rs.next())
            {
                RecUpdate ru = new RecUpdate();
                ru.lblId.setText(rs.getString(1));
                ru.lblName.setText(rs.getString(2));
                try
                {
                    Date date = new SimpleDateFormat("yyyy-MM-dd").parse(rs.getString(3));
                    ru.dateChooser.setDate(date);
                }
                catch (ParseException e1)
                {
                    e1.printStackTrace();
                }
                if (rs.getString(4).equals("MALE"))
                {
                    ru.cmbSecQueston.setSelectedIndex(0);
                }
                else if (rs.getString(4).equals("FEMALE"))
                {
                    ru.cmbSecQueston.setSelectedIndex(1);
                }
                ru.cmbGender.setSelectedItem(rs.getString(4));
                ru.txtMob_No.setText(rs.getString(5));
                ru.txtUser_Id.setText(rs.getString(6));
                if (rs.getString(7).equals("NICK NAME"))
                {
                    ru.cmbSecQueston.setSelectedIndex(0);
                }
                else if (rs.getString(7).equals("FAVOURITE COLOR"))
                {
                    ru.cmbSecQueston.setSelectedIndex(1);
                }
                else if (rs.getString(7).equals("FAVOURITE FOOD"))
                {
                    ru.cmbSecQueston.setSelectedIndex(2);
                }
                else if (rs.getString(7).equals("FAVOURITE PLACE"))
                {
                    ru.cmbSecQueston.setSelectedIndex(3);
                }
            }
        }
    }
});

```

```

        }
        ru.cmbSecQueston.setSelectedItem(rs.getString(7));
        ru.txtsAnswer.setText(rs.getString(8));
        ru.txtPASS.setText(rs.getString(9));
        ru.txt_ConfPASS.setText(rs.getString(9));
        ru.txtEmailId.setText(rs.getString(10));
        ru.txtFull_Address.setText(rs.getString(11));
    }
    } catch (SQLException e1) {
        e1.printStackTrace();
    }
}

});
frm.add(lblupdateprofile);
Canvas canvaschpass = new Canvas();
canvaschpass.setBackground(new Color(128, 0, 0));
canvaschpass.setBounds(250, 325, 150, 4);
frm.add(canvaschpass);
barlabel = new JLabel("");
Image barimg = new
ImageIcon(this.getClass().getResource("/images/recbar.png")).getImage();
barlabel.setIcon(new ImageIcon(barimg));
barlabel.setBounds(91, 19, 1293, 102);
frm.add(barlabel);
newpatientbtn = new JButton();
Image recimg = new
ImageIcon(this.getClass().getResource("/images/newpatientbtn.png")).getImage();
newpatientbtn.setIcon(new ImageIcon(recimg));
newpatientbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
newpatientbtn.setBounds(30, 450, 370, 60);
newpatientbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
newpatientbtn.addActionListener(this);
frm.add(newpatientbtn);
oldpatientbtn = new JButton();
Image docimg = new
ImageIcon(this.getClass().getResource("/images/oldpatientbtn.png")).getImage();
oldpatientbtn.setIcon(new ImageIcon(docimg));
oldpatientbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
oldpatientbtn.setBounds(30, 550, 370, 60);
oldpatientbtn.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
oldpatientbtn.addActionListener(this);
frm.add(oldpatientbtn);
Fullimg = new JLabel();
Image img5 = new
ImageIcon(this.getClass().getResource("/images/rec_bg.png")).getImage();
Fullimg.setIcon(new ImageIcon(img5));
Fullimg.setBounds(0, 0, framewidth, frameheight);
btnlogof = new JButton();
Image logoffimg = new
ImageIcon(this.getClass().getResource("/images/logog.jpg")).getImage();
btnlogof.setBounds(1280, 725, 35, 35);
btnlogof.setIcon(new ImageIcon(logoffimg));
Fullimg.add(btnlogof);
btnlogof.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
btnlogof.addActionListener(this);
btnExit = new JButton();
Image exiting = new
ImageIcon(this.getClass().getResource("/images/Exit.jpg")).getImage();
btnExit.setBounds(1322, 725, 35, 35);

```

```

        btnExit.setIcon(new ImageIcon(exitimg));
        Fullimg.add(btnExit);
        btnExit.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnExit.addActionListener(this);
        frm.add(Fullimg);
        frm.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(oldpatientbtn))
        {
            new Appointment();
        }
        else if (e.getSource().equals(newpatientbtn))
        {
            new AnimalDetails();
        }
        else if (e.getSource().equals(btnExit))
        {
            System.exit(0);
        }
        else if (e.getSource().equals(btnlogout))
        {
            new Login();
            frm.dispose();
        }
    }
}
}
public class ReceptionistPortal
{
    RecPortal rf = new RecPortal();
}

```

Appointment.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.text.*;
import java.util.Calendar;
import java.util.Date;
import javax.swing.*;
import javax.swing.text.MaskFormatter;
import com.toedter.calendar.JDateChooser;
class Appt extends ValidateForm implements ActionListener, FocusListener, ItemListener
{
    public PreparedStatement ps;
    public ResultSet rs;
    public Connection conn;
    public JFrame frm;
    public JButton btnBOOK, btnClear;
    public JLabel lblVeterinaryManagementSystem, lblAppointment, barlabel, lblregistration,
    lblTime, lblAppmntId, lblAppmntDate, lblVtime, lblAnimalName, lblFee,
    lblMedicalHistory, lblWeight, lblDoc_Emp, lblProblem, lblAge, lblService, lblHeight,
    lblName, lblDate, lblGender, lblMob_No;
    public JTextField txtregistration, txtAppmntId, txtAppmntDate, txtVtime, txtAge,
    txtProblem, txtFee, txtWeight, txtAnimalName, txtName, txtMob_No, txtHeight;
    public JComboBox<Object> cmbGender, cmbServices, cmbDoc_Emp;
}

```



```

public JDateChooser dateChooser;
public JSpinner spinner;
public Color hexa1;
String gender[] = { "MALE", "FEMALE" };
String pet[] = { "BIRDS", "CATS", "DOG", "RABBITS" };
public static String date, time;
Font f = new Font("Consolas", Font.BOLD, 13);
Font heading = new Font("Bookman Old Style", Font.BOLD, 18);
public MaskFormatter mf1, mf2, mf3, mf4;
Appt()
{
    frm = new JFrame();
    frm.setSize(870, 620);
    Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
    int framewidth = frm.getSize().width;
    int frameheight = frm.getSize().height;
    int framelocationX = (dim.width - framewidth) / 2;
    int frmlocationY = (dim.height - frameheight) / 2;
    frm.setLocation(framelocationX, frmlocationY);
    frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/images/icon.png")));
    frm.setResizable(false);
    Color hexa = Color.decode("#BBF2FA");
    frm.getContentPane().setBackground(hexa);
    frm.setLayout(null);
    frm.setTitle("BOOK APPOINTMENT");
    hexa1 = Color.decode("#82efff");
    lblVeterinaryManagementSystem = new JLabel("Veterinary Management System");
    lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
    lblVeterinaryManagementSystem.setForeground(Color.WHITE);
    lblVeterinaryManagementSystem.setBounds(80, 12, 650, 30);
    frm.add(lblVeterinaryManagementSystem);
    Canvas canvas = new Canvas();
    canvas.setBackground(Color.WHITE);
    canvas.setForeground(Color.RED);
    canvas.setBounds(80, 40, 635, 5);
    frm.add(canvas);
    lblAppointment = new JLabel("Appointment");
    lblAppointment.setForeground(Color.WHITE);
    lblAppointment.setFont(new Font("Magneto", Font.BOLD, 23));
    lblAppointment.setBounds(300, 60, 300, 27);
    frm.add(lblAppointment);
    barlabel = new JLabel();
    Image userimg = new
    ImageIcon(this.getClass().getResource("/images/recbar.png")).getImage();
    barlabel.setIcon(new ImageIcon(userimg));
    barlabel.setBounds(0, 0, 870, 105);
    frm.add(barlabel);
    String curdate = new SimpleDateFormat("MMM dd,
    yyyy").format(Calendar.getInstance().getTime());
    String curtime = new SimpleDateFormat("hh:mm a").format(new Date()).toString();
    lblDate = new JLabel();
    lblDate.setBounds(595, 110, 150, 30);
    lblDate.setFont(heading);
    lblDate.setText(curdate);
    lblDate.setForeground(Color.WHITE);
    frm.add(lblDate);
    lblTime = new JLabel();

```

```

lblTime.setBounds(730, 110, 150, 30);
lblTime.setFont(heading);
lblTime.setText(curtime);
lblTime.setForeground(Color.WHITE);
frm.add(lblTime);
lblregistration = new JLabel("REGISTRATION NO");
lblregistration.setBounds(150, 130, 200, 30);
lblregistration.setFont(new Font("Consolas", Font.BOLD, 20));
lblregistration.setForeground(Color.BLUE);
frm.add(lblregistration);
txtregistration = new JTextField();
txtregistration.setBounds(340, 130, 230, 30);
txtregistration.setFont(f);
txtregistration.setBackground(Color.WHITE);
txtregistration.setForeground(Color.BLACK);
txtregistration.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtregistration.addFocusListener(this);
frm.add(txtregistration);
lblAppmntId = new JLabel("APPOINTMENT ID");
lblAppmntId.setBounds(20, 180, 100, 30);
lblAppmntId.setFont(f);
lblAppmntId.setForeground(Color.BLUE);
frm.add(lblAppmntId);
txtApmntId = new JTextField();
txtApmntId.setBounds(150, 180, 250, 30);
txtApmntId.setFont(f);
txtApmntId.setBackground(Color.WHITE);
txtApmntId.setForeground(Color.BLACK);
txtApmntId.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
getidText();
frm.add(txtApmntId);
lblName = new JLabel("OWNER NAME");
lblName.setBounds(450, 180, 100, 30);
lblName.setFont(f);
lblName.setForeground(Color.BLUE);
frm.add(lblName);
txtName = new JTextField();
txtName.setBounds(580, 180, 250, 30);
txtName.setFont(f);
txtName.setBackground(Color.WHITE);
txtName.setForeground(Color.BLACK);
txtName.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtName);
lblMob_No = new JLabel("MOBILE NO.");
lblMob_No.setBounds(20, 230, 100, 30);
lblMob_No.setFont(f);
lblMob_No.setForeground(Color.BLUE);
frm.add(lblMob_No);
txtMob_No = new JTextField();
txtMob_No.setBounds(150, 230, 250, 30);
txtMob_No.setFont(f);
txtMob_No.setBackground(Color.WHITE);
txtMob_No.setForeground(Color.BLACK);
txtMob_No.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtMob_No);
lblAnimalName = new JLabel("PET NAME");
lblAnimalName.setBounds(450, 230, 100, 30);
lblAnimalName.setFont(f);
lblAnimalName.setForeground(Color.BLUE);

```

```

frm.add(lblAnimalName);
txtAnimalName = new JTextField();
txtAnimalName.setBounds(580, 230, 250, 30);
txtAnimalName.setFont(f);
txtAnimalName.setBackground(Color.WHITE);
txtAnimalName.setForeground(Color.BLACK);
txtAnimalName.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtAnimalName);
lblGender = new JLabel("GENDER");
lblGender.setBounds(20, 280, 100, 30);
lblGender.setFont(f);
lblGender.setForeground(Color.BLUE);
frm.add(lblGender);
cmbGender = new JComboBox<Object>(gender);
cmbGender.setBounds(150, 280, 250, 30);
cmbGender.setFont(f);
cmbGender.setSelectedIndex(-1);
cmbGender.setBackground(Color.WHITE);
cmbGender.setForeground(Color.BLACK);
cmbGender.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(cmbGender);
lblAge = new JLabel("AGE");
lblAge.setBounds(450, 280, 150, 30);
lblAge.setFont(f);
lblAge.setForeground(Color.BLUE);
frm.add(lblAge);
txtAge = new JTextField();
txtAge.setBounds(580, 280, 250, 30);
txtAge.setFont(f);
txtAge.setBackground(Color.WHITE);
txtAge.setForeground(Color.BLACK);
txtAge.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtAge);
lblHeight = new JLabel("HEIGHT");
lblHeight.setBounds(20, 330, 100, 30);
lblHeight.setFont(f);
frm.add(lblHeight);
lblHeight.setForeground(Color.BLUE);
try
{
    mf3 = new MaskFormatter("###CM");
}
catch (ParseException e)
{
    e.printStackTrace();
}
txtHeight = new JFormattedTextField(mf3);
txtHeight.setBounds(150, 330, 250, 30);
txtHeight.setFont(f);
txtHeight.setBackground(Color.WHITE);
txtHeight.setForeground(Color.BLACK);
txtHeight.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtHeight);
lblWeight = new JLabel("WEIGHT");
lblWeight.setBounds(450, 330, 150, 30);
lblWeight.setFont(f);
lblWeight.setForeground(Color.BLUE);
frm.add(lblWeight);
try

```

```

{
    mf4 = new MaskFormatter("###KG");
}
catch (ParseException e)
{
    e.printStackTrace();
}
txtWeight = new JFormattedTextField(mf4);
txtWeight.setBounds(580, 330, 250, 30);
txtWeight.setFont(f);
txtWeight.setBackground(Color.WHITE);
txtWeight.setForeground(Color.BLACK);
txtWeight.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtWeight);
lblAppmntDate = new JLabel("APPOINTMENT DATE");
lblAppmntDate.setBounds(20, 380, 150, 30);
lblAppmntDate.setFont(f);
lblAppmntDate.setForeground(Color.BLUE);
frm.add(lblAppmntDate);
dateChooser = new JDateChooser();
dateChooser.setBounds(150, 380, 250, 30);
dateChooser.setFont(f);
dateChooser.setBackground(Color.WHITE);
dateChooser.setForeground(Color.BLACK);
dateChooser.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(dateChooser);
lblVtime = new JLabel("APPOINTMENT TIME");
lblVtime.setBounds(450, 380, 150, 30);
lblVtime.setFont(f);
lblVtime.setForeground(Color.BLUE);
frm.add(lblVtime);
Date time = new Date();
SpinnerDateModel sm = new SpinnerDateModel(time, null, null,
Calendar.HOUR_OF_DAY);
spinner = new JSpinner(sm);
spinner.setBounds(580, 380, 250, 30);
JSpinner.DateEditor de = new JSpinner.DateEditor(spinner, "hh:mm a");
spinner.setEditor(de);
spinner.setBackground(Color.WHITE);
spinner.setForeground(Color.BLACK);
spinner.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(spinner);
lblProblem = new JLabel("PROBLEM");
lblProblem.setBounds(20, 430, 150, 30);
lblProblem.setFont(f);
lblProblem.setForeground(Color.BLUE);
frm.add(lblProblem);
txtProblem = new JTextField();
txtProblem.setBounds(150, 430, 250, 30);
txtProblem.setFont(f);
txtProblem.setBackground(Color.WHITE);
txtProblem.setForeground(Color.BLACK);
txtProblem.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtProblem);
lblService = new JLabel("SERVICES");
lblService.setBounds(450, 430, 100, 30);
lblService.setFont(f);
lblService.setForeground(Color.BLUE);
frm.add(lblService);

```

```

        cmbServices = new JComboBox<Object>();
        cmbServices.setBounds(580, 430, 250, 30);
        cmbServices.setFont(f);
        cmbServices.setBackground(Color.WHITE);
        cmbServices.setForeground(Color.BLACK);
        cmbServices.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        cmbServices.setSelectedIndex(-1);
        srvcour();
        cmbServices.addItemListener(this);
        frm.add(cmbServices);
        lblDoc_Emp = new JLabel("DOCTOR/EMPLOYEE");
        lblDoc_Emp.setBounds(20, 480, 150, 30);
        lblDoc_Emp.setFont(f);
        lblDoc_Emp.setForeground(Color.BLUE);
        frm.add(lblDoc_Emp);
        cmbDoc_Emp = new JComboBox<Object>();
        cmbDoc_Emp.setBounds(150, 480, 250, 30);
        cmbDoc_Emp.setFont(f);
        cmbDoc_Emp.setBackground(Color.WHITE);
        cmbDoc_Emp.setForeground(Color.BLACK);
        cmbDoc_Emp.setSelectedIndex(-1);
        cmbDoc_Emp.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        frm.add(cmbDoc_Emp);
        lblFee = new JLabel("CHARGES");
        lblFee.setBounds(450, 480, 150, 30);
        lblFee.setFont(f);
        lblFee.setForeground(Color.BLUE);
        frm.add(lblFee);
        txtFee = new JTextField();
        txtFee.setBounds(580, 480, 250, 30);
        txtFee.setFont(f);
        txtFee.setEditable(false);
        txtFee.setBackground(Color.WHITE);
        txtFee.setForeground(Color.BLACK);
        txtFee.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        frm.add(txtFee);
        btnBOOK = new JButton("BOOK");
        btnBOOK.setBounds(20, 530, 392, 35);
        btnBOOK.setFont(f);
        btnBOOK.setForeground(Color.BLUE);
        btnBOOK.setBackground(hexa1);
        btnBOOK.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnBOOK.addActionListener(this);
        frm.add(btnBOOK);
        btnClear = new JButton("CLEAR");
        btnClear.setBounds(440, 530, 392, 35);
        btnClear.setFont(f);
        btnClear.setForeground(Color.BLUE);
        btnClear.setBackground(hexa1);
        btnClear.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnClear.addActionListener(this);
        frm.add(btnClear);
        frm.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(btnBOOK))
        {

```

```

SimpleDateFormat dateFormat = new
SimpleDateFormat("dd/MMM/yyyy");
String ddate = dateFormat.format(dateChooser.getDate());
String curdate = new
SimpleDateFormat("dd/MMM/yyyy").format(Calendar.getInstance().getTi
me());
date = curdate;
SimpleDateFormat timeFormat = new SimpleDateFormat("hh:mm a");
String appointmenttime = timeFormat.format(spinner.getValue());
if (frmvalidate())
{
    AppointmentOperation ao = new
AppointmentOperation(txtregistration.getText().toUpperCase(),
date, lblTime.getText(), txtApmntId.getText(),
txtName.getText().toUpperCase(),
txtAnimalName.getText().toUpperCase(),
txtMob_No.getText().toUpperCase(),
cmbGender.getSelectedItem().toString().toUpperCase(),
txtAge.getText(), txtHeight.getText().toUpperCase(),
txtWeight.getText().toUpperCase(), ddate, appointmenttime,
txtProblem.getText().toUpperCase(),
cmbServices.getSelectedItem().toString().toUpperCase(),
txtFee.getText().toUpperCase(),
cmbDoc_Emp.getSelectedItem().toString());
    if (ao.userValidate())
    {
        if (ao.saverecord() == 1)
        {
            JOptionPane.showMessageDialog(null,
"APPOINTMENT SUCCESFULL BOOK");
            txtregistration.setText("");
            clearForm();
            txtregistration.setEditable(true);
            frm.dispose();
        }
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Record
Already Inserted");
    }
}
}
else if (e.getSource().equals(btnClear))
{
    clearForm();
}
}
public void clearForm()
{
    txtregistration.setText("");
    txtName.setText("");
    txtMob_No.setText("");
    txtAnimalName.setText("");
    cmbGender.setSelectedIndex(-1);
    txtAge.setText("");
    txtHeight.setText("");
    txtWeight.setText("");
    dateChooser.setDate(null);
}

```

```

        txtProblem.setText("");
        cmbServices.setSelectedIndex(-1);
        cmbDoc_Emp.setSelectedIndex(-1);
        txtFee.setText("");
    }
    public boolean frmvalidate()
    {
        boolean flag = true;
        if (txtregistration.getText().equals(""))
        {
            JOptionPane.showMessageDialog(null, "Enter Registration No. ");
            txtregistration.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateAnyString(txtName.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Owner Name ");
            txtName.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateMobileNo(txtMob_No.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Correct phone no..");
            txtMob_No.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateAnyString(txtAnimalName.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Animal name ");
            txtAnimalName.grabFocus();
            flag = false;
        }
        else if (cmbGender.getSelectedIndex() == -1)
        {
            JOptionPane.showMessageDialog(null, "Choose Your Gender");
            cmbGender.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateAnyNo(txtAge.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Age ");
            txtAge.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateheiWei(txtHeight.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Choose Species");
            txtHeight.grabFocus();
            flag = false;
        }
        else if (ValidateForm.validateheiWei(txtWeight.getText()) == false)
        {
            JOptionPane.showMessageDialog(null, "Enter Breed ");
            txtWeight.grabFocus();
            flag = false;
        }
        return flag;
    }
}
@Override

```

```

public void focusGained(FocusEvent e)
{
}
@Override
public void focusLost(FocusEvent e)
{
    AnimalDetailsOperations ado = new AnimalDetailsOperations();
    ResultSet rs = ado.sec_recordforApp(txtregistration.getText());
    try
    {
        if (rs.next())
        {
            txtName.setText(rs.getString(1));
            txtAnimalName.setText(rs.getString(2));
            if (rs.getString(3).equals(" MALE"))
            {
                cmbGender.setSelectedIndex(0);
            }
            else if (rs.getString(3).equals(" FEMALE"))
            {
                cmbGender.setSelectedIndex(1);
            }
            cmbGender.setSelectedItem(rs.getString(3));
            txtAge.setText(rs.getString(4));
            txtHeight.setText(rs.getString(5));
            txtWeight.setText(rs.getString(6));
            txtMob_No.setText(rs.getString(7));
        }
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
    }
}

public void srvcour()
{
    String sqlqry = "select SERVICE from SERVICE_DETAILS";
    conn = Connections.getConnections();
    try
    {
        ps = conn.prepareStatement(sqlqry);
        rs = ps.executeQuery();
        while (rs.next())
        {
            cmbServices.addItem(rs.getString(1));
            cmbServices.setSelectedIndex(-1);
        }
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}

public void fee(String serv)
{
    String query = "select FEE from SERVICE_DETAILS WHERE SERVICE='" + serv
    + "'";
    conn = Connections.getConnections();
    try

```



```

        {
            ps = conn.prepareStatement(query);
            rs = ps.executeQuery();
            while (rs.next())
            {
                txtFee.setText(rs.getString(1));
            }
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
        txtFee.getText();
    }
    public void getidText()
    {
        AppointmentOperation ao = new AppointmentOperation();
        txtApmntId.setText(ao.getid());
        txtApmntId.setEditable(false);
    }
    public void doccour()
    {
        String sqlqry = "select NAME from DOCTOR_DETAILS";
        conn = Connections.getConnections();
        try
        {
            ps = conn.prepareStatement(sqlqry);
            rs = ps.executeQuery();
            while (rs.next())
            {
                cmbDoc_Emp.addItem(rs.getString(1));
                cmbDoc_Emp.setSelectedIndex(-1);
            }
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
    public void empcour()
    {
        String sqlqry = "select NAME from EMPLOYEE_DETAILS";
        conn = Connections.getConnections();
        try
        {
            ps = conn.prepareStatement(sqlqry);
            rs = ps.executeQuery();
            while (rs.next())
            {
                cmbDoc_Emp.addItem(rs.getString(1));
                cmbDoc_Emp.setSelectedIndex(-1);
            }
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
}
@Override

```

```

public void itemStateChanged(ItemEvent e)
{
    cmbDoc_Emp.removeAllItems();
    if ((cmbServices.getSelectedItem().toString()).equals("CONSULTANT CHARGES"))
    {
        doccour();
        fee(cmbServices.getSelectedItem().toString());
    }
    else
    {
        empcour();
        fee(cmbServices.getSelectedItem().toString());
    }
}
}
public class Appointment
{
    Appt ap = new Appt();
}

```

DoctorPortal.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import javax.swing.*;
class DocPortal extends Base implements ActionListener
{
    public JFrame frm;
    public JLabel Fulling, userlabel, idlabel, namelabel, lblupdateprofile, doclabel,
    lblVeterinaryManagementSystem, lblAdminPortal, barlabel;
    public JButton appbtn, treatbtn, empbtn, drtimebtn, drugbtn, vaccbtn, servbtn, app1btn,
    btnlok, btnlogof, btnExit;
    DocPortal()
    {
        frm = new JFrame();
        frm.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        Toolkit toolkit = frm.getToolkit();
        Dimension size = toolkit.getScreenSize();
        frm.setSize(size.width, size.height);
        frm.setLayout(null);
        frm.setResizable(false);
        frm.setUndecorated(true);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int framelocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, framelocationY);
        frm.getContentPane().setBackground(Color.WHITE);
        frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/i
        mages/icon.png")));
        doclabel = new JLabel("");
    }
}

```

```

Image adminimg = new
ImageIcon(this.getClass().getResource("/images/doc_icon.png")).getImage();
doclabel.setIcon(new ImageIcon(adminimg));
doclabel.setBounds(0, 5, 128, 128);
frm.add(doclabel);
lblVeterinaryManagementSystem = new JLabel("Veterinary Management
System");
lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
lblVeterinaryManagementSystem.setForeground(Color.WHITE);
lblVeterinaryManagementSystem.setBounds(350, 22, 632, 49);
frm.add(lblVeterinaryManagementSystem);
Canvas canvas = new Canvas();
canvas.setBackground(Color.WHITE);
canvas.setForeground(Color.RED);
canvas.setBounds(350, 71, 630, 4);
frm.add(canvas);
lblAdminPortal = new JLabel("Doctor Portal");
lblAdminPortal.setForeground(Color.WHITE);
lblAdminPortal.setFont(new Font("Magneto", Font.BOLD, 27));
lblAdminPortal.setBounds(560, 88, 300, 27);
frm.add(lblAdminPortal);
userlabel = new JLabel();
Image userimg = new
ImageIcon(this.getClass().getResource("/images/usernew.png")).getImage();
userlabel.setIcon(new ImageIcon(userimg));
userlabel.setBounds(30, 145, 200, 200);
frm.add(userlabel);
namelabel = new JLabel();
namelabel.setFont(new Font("Agency FB", Font.BOLD, 32));
namelabel.setBounds(250, 160, 300, 60);
frm.add(namelabel);
idlabel = new JLabel();
userlabel = new JLabel("DOCTOR");
userlabel.setFont(new Font("Agency FB", Font.BOLD, 32));
userlabel.setBounds(250, 220, 200, 40);
frm.add(userlabel);
lblupdateprofile = new JLabel("Update Profile");
lblupdateprofile.setForeground(new Color(128, 0, 0));
lblupdateprofile.setFont(new Font("Agency FB", Font.BOLD, 32));
lblupdateprofile.setBounds(250, 280, 200, 40);
lblupdateprofile.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
;
lblupdateprofile.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)
    {
        RegisterOperation ro = new RegisterOperation();
        ResultSet rs = ro.doc_sec_recordUpdate((idlabel.getText()));
        try
        {
            if (rs.next())
            {
                DocUpdate ru = new DocUpdate();
                ru.lblId.setText(rs.getString(1));
                ru.lblName.setText(rs.getString(2));
                try
                {

```

```

        Date date = new SimpleDateFormat("yyyy-MM-dd").parse(rs.getString(3));
        ru.dateChooser.setDate(date);
    }
    catch (ParseException e1)
    {
        e1.printStackTrace();
    }
    if (rs.getString(4).equals("MALE"))
    {
        ru.cmbSecQueston.setSelectedIndex(0);
    }
    else if (rs.getString(4).equals("FEMALE"))
    {
        ru.cmbSecQueston.setSelectedIndex(1);
    }
    ru.cmbGender.setSelectedItem(rs.getString(4));
    ru.txtMob_No.setText(rs.getString(5));
    ru.txtUser_Id.setText(rs.getString(6));
    if (rs.getString(7).equals("NICK NAME"))
    {
        ru.cmbSecQueston.setSelectedIndex(0);
    }
    else if (rs.getString(7).equals("FAVOURITE COLOR"))
    {
        ru.cmbSecQueston.setSelectedIndex(1);
    }
    else if (rs.getString(7).equals("FAVOURITE FOOD"))
    {
        ru.cmbSecQueston.setSelectedIndex(2);
    }
    else if (rs.getString(7).equals("FAVOURITE PLACE"))
    {
        ru.cmbSecQueston.setSelectedIndex(3);
    }
    ru.cmbSecQueston.setSelectedItem(rs.getString(7));
    ru.txtsAnswer.setText(rs.getString(8));
    ru.txtPASS.setText(rs.getString(9));
    ru.txt_ConfPASS.setText(rs.getString(9));
    ru.txtEmailId.setText(rs.getString(10));
    ru.txtFull_Address.setText(rs.getString(11));
    }
    }
    catch (SQLException e1)
    {
        e1.printStackTrace();
    }
    }
});
frm.add(lblupdateprofile);
Canvas canvaschpass = new Canvas();
canvaschpass.setBackground(new Color(128, 0, 0));
canvaschpass.setBounds(250, 325, 150, 4);
frm.add(canvaschpass);
barlabel = new JLabel("");
Image barimg = new
ImageIcon(this.getClass().getResource("/images/doc_bar.png")).getImage();
barlabel.setIcon(new ImageIcon(barimg));

```

```

        barlabel.setBounds(91, 19, 1293, 102);
        frm.add(barlabel);
        treatbtn = new JButton();
        Image recimg = new
        ImageIcon(this.getClass().getResource("/images/treatmentdoc.png")).getImage();
        treatbtn.setIcon(new ImageIcon(recimg));
        treatbtn.setFont(new Font("Bookman Old Style", Font.BOLD, 20));
        treatbtn.setBounds(30, 480, 370, 60);
        treatbtn.addActionListener(this);
        frm.add(treatbtn);
        Fullimg = new JLabel();
        Image img5 = new
        ImageIcon(this.getClass().getResource("/images/doc_img.png")).getImage();
        Fullimg.setIcon(new ImageIcon(img5));
        Fullimg.setBounds(0, 0, framewidth, frameheight);
        btnlogof = new JButton();
        Image logoffimg = new
        ImageIcon(this.getClass().getResource("/images/logog.jpg")).getImage();
        btnlogof.setBounds(1280, 725, 35, 35);
        btnlogof.setIcon(new ImageIcon(logoffimg));
        Fullimg.add(btnlogof);
        btnlogof.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnlogof.addActionListener(this);
        btnExit = new JButton();
        Image exiting = new
        ImageIcon(this.getClass().getResource("/images/Exit.jpg")).getImage();
        btnExit.setBounds(1322, 725, 35, 35);
        btnExit.setIcon(new ImageIcon(exiting));
        Fullimg.add(btnExit);
        btnExit.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
        btnExit.addActionListener(this);
        frm.add(Fullimg);
        frm.setVisible(true);
    }
    @Override
    public void actionPerformed(ActionEvent e)
    {
        if (e.getSource().equals(treatbtn))
        {
            String curdate = new
            SimpleDateFormat("dd/MMM/yyyy").format(Calendar.getInstance().getTi
            me());
            Presc pr = new Presc();
            pr.lblDoctorName.setText(namelabel.getText());
            pr.ShowTable(curdate, namelabel.getText());
        }
        else if (e.getSource().equals(btnExit))
        {
            System.exit(0);
        }
        else if (e.getSource().equals(btnlogof))
        {
            new Login();
            frm.dispose();
        }
    }
}
public class DoctorPortal
{

```

```

        DocPortal d = new DocPortal();
    }

```

Prescription.Java

```

import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.text.*;
import java.util.Calendar;
import java.util.Date;
import javax.swing.*;
import javax.swing.table.*;
import javax.swing.text.MaskFormatter;
import net.proteanit.sql.DbUtils;
class Presc extends Base implements ActionListener, WindowListener
{
    public static String abc = null;
    public JFrame frm;
    public PreparedStatement ps;
    public ResultSet rs;
    public Connection conn;
    public JButton btnSave, btnMedicine, btnVaccin, btnUpdate, btnClear;
    public JLabel lblVeterinaryManagementSystem, barlabel, lblPrescribe, lblprescription,
    lblDoctorName, lblRegNo, lblTime, lblAnimalName, lblOwnerName, lblMedicalHistory,
    lblWeight, lblState, lblAge, lblEmailId, lblHeight, lblAppointmentNo, lblDate, lblSpecies,
    lblBreed, lblPrescription, lblMob_No, lblcurrentCondition, lbllogo;
    public JTextField txtprescription, txtOwnerName, txtDoctorName, txtRegNo, txtAge,
    txtWeight, txtAnimalName, txtBreed, txtAppointmentNo, txtDate, txtMob_No, txtEmailId,
    txtHeight;
    public JComboBox<Object> cmbSpecies, cmbState;
    public JTextArea txtPrescription, txtCurrentCondition, txtMedicalHistory;
    public Color hexa1;
    public DefaultTableModel jTable;
    public JTable tabview;
    public boolean start = true;
    public JScrollPane jsCurrentCondition, jsMedicalHistory, JSPrescription;
    public static String date;
    Font heading = new Font("Bookman Old Style", Font.BOLD, 18);
    Font f = new Font("Consolas", Font.BOLD, 13);
    public MaskFormatter mf1, mf2, mf3;
    Presc()
    {
        frm = new JFrame();
        frm.setSize(1200, 560);
        Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
        int framewidth = frm.getSize().width;
        int frameheight = frm.getSize().height;
        int framelocationX = (dim.width - framewidth) / 2;
        int frmlocationY = (dim.height - frameheight) / 2;
        frm.setLocation(framelocationX, frmlocationY);
        frm.setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("/i
        mages/icon.png")));
        frm.setResizable(false);
        Color hexa = Color.decode("#BBF2FA");
        frm.getContentPane().setBackground(hexa);
        frm.setLayout(null);
        frm.setTitle("TREATMENT FORM");
    }

```

```

hexa1 = Color.decode("#82efff");
lblVeterinaryManagementSystem = new JLabel("Veterinary Management
System");
lblVeterinaryManagementSystem.setFont(new Font("Algerian", Font.BOLD, 34));
lblVeterinaryManagementSystem.setForeground(Color.WHITE);
lblVeterinaryManagementSystem.setBounds(280, 12, 650, 30);
frm.add(lblVeterinaryManagementSystem);
Canvas canvas = new Canvas();
canvas.setBackground(Color.WHITE);
canvas.setForeground(Color.RED);
canvas.setBounds(280, 40, 635, 5);
frm.add(canvas);
lblPrescribe = new JLabel("Treatment Form");
lblPrescribe.setForeground(Color.WHITE);
lblPrescribe.setFont(new Font("Magneto", Font.BOLD, 23));
lblPrescribe.setBounds(510, 60, 300, 27);
frm.add(lblPrescribe);
barlabel = new JLabel();
Image userimg = new
ImageIcon(this.getClass().getResource("/images/doc_bar.png")).getImage();
barlabel.setIcon(new ImageIcon(userimg));
barlabel.setBounds(0, 0, 1260, 105);
frm.add(barlabel);
lblDoctorName = new JLabel();
lblDoctorName.setBounds(20, 120, 200, 30);
lblDoctorName.setFont(heading);
lblDoctorName.setForeground(Color.WHITE);
frm.add(lblDoctorName);
String curdate = new SimpleDateFormat("MMM dd,
yyyy").format(Calendar.getInstance().getTime());
String curtime = new SimpleDateFormat("hh:mm a").format(new
Date()).toString();
lblDate = new JLabel();
lblDate.setBounds(935, 120, 150, 30);
lblDate.setFont(heading);
lblDate.setText(curdate);
lblDate.setForeground(Color.WHITE);
frm.add(lblDate);
lblTime = new JLabel();
lblTime.setBounds(1070, 120, 150, 30);
lblTime.setFont(heading);
lblTime.setText(curtime);
lblTime.setForeground(Color.WHITE);
frm.add(lblTime);
lblAppointmentNo = new JLabel("APPOINTMENT NO");
lblAppointmentNo.setBounds(380, 120, 200, 30);
lblAppointmentNo.setFont(new Font("Consolas", Font.BOLD, 20));
lblAppointmentNo.setForeground(Color.RED);
frm.add(lblAppointmentNo);
txtAppointmentNo = new JTextField();
txtAppointmentNo.setBounds(590, 120, 200, 30);
txtAppointmentNo.setFont(new Font("Consolas", Font.BOLD, 20));
txtAppointmentNo.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSO
R));
txtAppointmentNo.setForeground(Color.BLACK);
frm.add(txtAppointmentNo);

lblprescription = new JLabel("PRESCRIPTION NO");
lblprescription.setBounds(20, 170, 150, 30);

```

```

lblprescription.setFont(f);
lblprescription.setForeground(Color.BLUE);
frm.add(lblprescription);
txtprescription = new JTextField();
txtprescription.setBounds(150, 170, 200, 30);
txtprescription.setFont(f);
txtprescription.setBackground(Color.WHITE);
txtprescription.setForeground(Color.BLACK);
txtprescription.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtprescription.setEditable(false);
frm.add(txtprescription);
lblRegNo = new JLabel("REGISTRATION NO");
lblRegNo.setBounds(380, 170, 150, 30);
lblRegNo.setFont(f);
lblRegNo.setForeground(Color.BLUE);
frm.add(lblRegNo);
txtRegNo = new JTextField();
txtRegNo.setBounds(500, 170, 200, 30);
txtRegNo.setFont(f);
txtRegNo.setForeground(Color.BLACK);
txtRegNo.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtRegNo);
lblOwnerName = new JLabel("OWNER NAME");
lblOwnerName.setBounds(20, 220, 100, 30);
lblOwnerName.setFont(f);
lblOwnerName.setForeground(Color.BLUE);
frm.add(lblOwnerName);
txtOwnerName = new JTextField();
txtOwnerName.setBounds(150, 220, 200, 30);
txtOwnerName.setFont(f);
txtOwnerName.setForeground(Color.BLACK);
txtOwnerName.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtOwnerName);
lblAnimalName = new JLabel("PET NAME");
lblAnimalName.setBounds(380, 220, 100, 30);
lblAnimalName.setFont(f);
lblAnimalName.setForeground(Color.BLUE);
frm.add(lblAnimalName);
txtAnimalName = new JTextField();
txtAnimalName.setBounds(500, 220, 200, 30);
txtAnimalName.setFont(f);
txtAnimalName.setForeground(Color.BLACK);
txtAnimalName.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtAnimalName);
lblAge = new JLabel("AGE");
lblAge.setBounds(20, 270, 150, 30);
lblAge.setFont(f);
lblAge.setForeground(Color.BLUE);
frm.add(lblAge);
txtAge = new JTextField();
txtAge.setBounds(150, 270, 200, 30);
txtAge.setFont(f);
txtAge.setForeground(Color.BLACK);
txtAge.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtAge);
lblBreed = new JLabel("BREED");
lblBreed.setBounds(380, 270, 100, 30);
lblBreed.setFont(f);
lblBreed.setForeground(Color.BLUE);

```



```

frm.add(lblBreed);
txtBreed = new JTextField();
txtBreed.setBounds(500, 270, 200, 30);
txtBreed.setFont(f);
txtBreed.setBackground(Color.WHITE);
txtBreed.setForeground(Color.BLACK);
txtBreed.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtBreed);
lblHeight = new JLabel("HEIGHT");
lblHeight.setBounds(20, 320, 100, 30);
lblHeight.setFont(f);
lblHeight.setForeground(Color.BLUE);
frm.add(lblHeight);
try
{
    mf2 = new MaskFormatter("###CM");
}
catch (ParseException e)
{
    e.printStackTrace();
}
txtHeight = new JFormattedTextField(mf2);
txtHeight.setBounds(150, 320, 200, 30);
txtHeight.setFont(f);
txtHeight.setForeground(Color.BLACK);
txtHeight.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtHeight);
lblWeight = new JLabel("WEIGHT");
lblWeight.setBounds(380, 320, 80, 30);
lblWeight.setFont(f);
lblWeight.setForeground(Color.BLUE);
frm.add(lblWeight);
try
{
    mf3 = new MaskFormatter("###KG");
}
catch (ParseException e)
{
    e.printStackTrace();
}
txtWeight = new JFormattedTextField(mf3);
txtWeight.setBounds(500, 320, 200, 30);
txtWeight.setFont(f);
txtWeight.setForeground(Color.BLACK);
txtWeight.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(txtWeight);
lblPrescription = new JLabel("PRESCRIPTION");
lblPrescription.setBounds(20, 370, 100, 30);
lblPrescription.setFont(f);
lblPrescription.setForeground(Color.BLUE);
frm.add(lblPrescription);
JSPrescription = new JScrollPane();
txtPrescription = new JTextArea();
txtPrescription.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
JSPrescription.setBounds(150, 370, 350, 80);
txtPrescription.setFont(f);
txtPrescription.setForeground(Color.BLACK);
JSPrescription.setViewportViewView(txtPrescription);
frm.add(JSPrescription);

```

```

btnMedicine = new JButton("ADD MEDICINE");
btnMedicine.setBounds(510, 370, 190, 35);
btnMedicine.setFont(f);
btnMedicine.setForeground(Color.BLUE);
btnMedicine.setBackground(hexa1);
btnMedicine.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(btnMedicine);
btnMedicine.addActionListener(this);
btnVaccin = new JButton("ADD VACINATION");
btnVaccin.setBounds(510, 415, 190, 35);
btnVaccin.setFont(f);
btnVaccin.setForeground(Color.BLUE);
btnVaccin.setBackground(hexa1);
btnVaccin.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
frm.add(btnVaccin);
btnVaccin.addActionListener(this);
lblcurrentCondition = new JLabel("CURRENT CONDITION");
lblcurrentCondition.setBounds(720, 170, 170, 30);
lblcurrentCondition.setForeground(Color.BLUE);
frm.add(lblcurrentCondition);
jsCurrentCondition = new JScrollPane();
jsCurrentCondition.setBounds(720, 200, 210, 90);
txtCurrentCondition = new JTextArea();
txtCurrentCondition.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtCurrentCondition.setFont(f);
txtCurrentCondition.setForeground(Color.BLACK);
txtCurrentCondition.setEditable(false);
jsCurrentCondition.setViewportViewView(txtCurrentCondition);
frm.add(jsCurrentCondition);
lblMedicalHistory = new JLabel("MEDICAL HISTORY");
lblMedicalHistory.setBounds(955, 170, 150, 30);
lblMedicalHistory.setForeground(Color.BLUE);
frm.add(lblMedicalHistory);
jsMedicalHistory = new JScrollPane();
txtMedicalHistory = new JTextArea();
jsMedicalHistory.setBounds(955, 200, 210, 90);
txtMedicalHistory.setFont(f);
txtMedicalHistory.setForeground(Color.BLACK);
txtMedicalHistory.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
txtMedicalHistory.setEditable(false);
jsMedicalHistory.setViewportViewView(txtMedicalHistory);
frm.add(jsMedicalHistory);
jtable = new DefaultTableModel();
tabview = new JTable(jtable);
JScrollPane js = new JScrollPane(tabview);
js.setBounds(720, 300, 450, 210);
tabview.setFont(f);
tabview.setForeground(Color.blue);
tabview.setBackground(Color.WHITE);
tabview.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
Object[] columns = { "Id", "MEDICINE", "DRUGS", "TYPE", "COMPANY" };
jtable.setColumnIdentifiers(columns);
tabview.setModel(jtable);
tabview.addMouseListener(new MouseAdapter()
{
    @Override
    public void mouseClicked(MouseEvent e)

```

```

        {
            getidtext();
            int i = tabview.getSelectedRow();
            TableModel model = tabview.getModel();
            txtAppointmentNo.setText(model.getValueAt(i, 0).toString());
            txtRegNo.setText(model.getValueAt(i, 1).toString());
            txtOwnerName.setText(model.getValueAt(i, 2).toString());
            txtAnimalName.setText(model.getValueAt(i, 3).toString());
            txtAge.setText(model.getValueAt(i, 4).toString());
            txtBreed.setText(model.getValueAt(i, 5).toString());
            txtHeight.setText(model.getValueAt(i, 6).toString());
            txtWeight.setText(model.getValueAt(i, 7).toString());
            txtCurrentCondition.setText(model.getValueAt(i, 8).toString());
        }
    });
    frm.add(js);
    btnSAVE = new JButton("SAVE");
    btnSAVE.setBounds(20, 470, 330, 40);
    btnSAVE.setFont(f);
    btnSAVE.setForeground(Color.BLUE);
    btnSAVE.setBackground(hexa1);
    btnSAVE.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    frm.add(btnSAVE);
    btnSAVE.addActionListener(this);
    btnClear = new JButton("CLEAR");
    btnClear.setBounds(370, 470, 330, 40);
    btnClear.setFont(f);
    btnClear.setForeground(Color.BLUE);
    btnClear.setBackground(hexa1);
    btnClear.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
    frm.add(btnClear);
    btnClear.addActionListener(this);
    frm.setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e)
{
    if (e.getSource().equals(btnSAVE))
    {
        String curdate = new
        SimpleDateFormat("dd/MMM/yyyy").format(Calendar.getInstance().getTi
        me());
        date = curdate;
        String MedicalHistory = txtMedicalHistory.getText();
        String history = date + "Problem - " +
        txtCurrentCondition.getText().toUpperCase() + "Solved - " +
        txtPrescription.getText().toUpperCase() + MedicalHistory;
        if (frmvalidate())
        {
            PrescriptionOperation po = new PrescriptionOperation(date,
            txtprescription.getText().toUpperCase(),
            lblAppointmentNo.getText().toUpperCase(),
            txtRegNo.getText().toUpperCase(), lblDoctorName.getText().toUpper
            Case(), txtOwnerName.getText().toUpperCase(),
            txtAnimalName.getText().toUpperCase(),
            txtCurrentCondition.getText().toUpperCase(),
            txtPrescription.getText().toUpperCase());
            if (po.doc_saverecord() == 1)
            {

```

```

        AnimalDetailsOperations ado = new
        AnimalDetailsOperations(txtRegNo.getText().toUpperCase(
        ),history);
        if (ado.animal_update() == 1)
        {
            JOptionPane.showMessageDialog(null, "SAVED
            SUCCESSFULLY");
            frm.dispose();
        }
    }
    else
    {
        JOptionPane.showMessageDialog(null, "NOT
        SAVED");
    }
}

}
else if (e.getSource().equals(btnClear))
{
    clearForm();
}
else if (e.getSource().equals(btnMedicine))
{
    new AddMedicine();
    frm.addWindowListener(this);
}
else if (e.getSource().equals(btnVaccin))
{
    new AddVaccination();
    frm.addWindowListener(this);
}
}

public void clearForm()
{
    txtAppointmentNo.setText("");
    txtRegNo.setText("");
    txtOwnerName.setText("");
    txtAnimalName.setText("");
    txtBreed.setText("");
    txtAge.setText("");
    txtHeight.setText("");
    txtWeight.setText("");
    txtPrescription.setText("");
    txtCurrentCondition.setText("");
}

public boolean frmvalidate()
{
    boolean flag = true;
    if (txtAppointmentNo.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Appointment No.");
        txtAppointmentNo.grabFocus();
        flag = false;
    }
    else if (txtRegNo.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Registration No.");
        txtRegNo.grabFocus();
        flag = false;
    }
}

```

```

    }
    else if (txtOwnerName.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Owner Name");
        txtOwnerName.grabFocus();
        flag = false;
    }
    else if (txtAnimalName.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Pet Name");
        txtAnimalName.grabFocus();
        flag = false;
    }
    else if (txtAge.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Age");
        txtAge.grabFocus();
        flag = false;
    }
    else if (txtBreed.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Breed");
        txtBreed.grabFocus();
        flag = false;
    }
    else if (txtHeight.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Height");
        txtHeight.grabFocus();
        flag = false;
    }
    else if (txtWeight.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Weight");
        txtWeight.grabFocus();
        flag = false;
    }
    else if (txtPrescription.getText().length() == 0)
    {
        JOptionPane.showMessageDialog(null, "Enter Prescription");
        txtPrescription.grabFocus();
        flag = false;
    }
    return flag;
}
public void getidtext()
{
    PrescriptionOperation po = new PrescriptionOperation();
    txtprescription.setText(po.doc_getid());
}
public void ShowTable(String date, String Name)
{
    String curdate = new
    SimpleDateFormat("dd/MMM/yyyy").format(Calendar.getInstance().getTime());
    String sqlqry = "select
    APPOINTMENT.APPOINTMENT_NO,APPOINTMENT.REG_NO,APPOINTMENT.OW
    NER_NAME,APPOINTMENT.PET_NAME,APPOINTMENT.AGE,ANIMAL_DETAILS.
    BREED,APPOINTMENT.HEIGHT,APPOINTMENT.WEIGHT,APPOINTMENT.PROBL
    EM,ANIMAL_DETAILS.MEDICAL_HISTORY from APPOINTMENT INNER JOIN

```

```

        ANIMAL_DETAILS ON APPOINTMENT.REG_NO=ANIMAL_DETAILS.REG_NO
        WHERE APPOINTMENT_DATE='"+ curdate + "' AND DOC_EMP_NAME='"+ Name
        + "'";
        conn = Connections.getConnections();
        try
        {
            ps = conn.prepareStatement(sqlqry);
            rs = ps.executeQuery();
            tabview.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch (SQLException e)
        {
            e.printStackTrace();
        }
    }
    @Override
    public void windowOpened(WindowEvent e)
    {
    }
    @Override
    public void windowClosing(WindowEvent e)
    {
    }
    @Override
    public void windowClosed(WindowEvent e)
    {
    }
    @Override
    public void windowIconified(WindowEvent e)
    {
    }
    @Override
    public void windowDeiconified(WindowEvent e)
    {
    }
    @SuppressWarnings("static-access")
    @Override
    public void windowActivated(WindowEvent e)
    {
        String value = super.Medical;
        txtPrescription.setText(abc + value);
    }
    @Override
    public void windowDeactivated(WindowEvent e)
    {
        abc = txtPrescription.getText();
    }
}
public class prescription
{
    Presc ps = new Presc();
}

```

TESTING PLAN

UNIT TESTING: -

Unit testing focuses verification efforts on the smallest unit of software design- a software component or module where important paths are tested to uncover errors within the boundary of the module. This is the white-box oriented test & can be conducted in parallel for multiple modules.

INTEGRATION TESTING: -

After conducting the unit testing, where all modules will be tested, can be it said that it will work when we put them together? There may be problem in facing among different modules. Hence, I feel to conduct the following integration testing on our project.

VALIDATION TESTING: -

We will ensure whether the software developed by me functions in a manner that are expected by the customer in the software requirements specifications. It will virtually impossible for me to know how the user will really use the software. Instruction for use may be misinterpreted; strange combination of data may be used, output seemed to me intangible to the users. For this problem we will conduct Alpha & Beta testing.

ALPHA TEST: -

Will be conduct at my working site by the user of the organization and errors be recorded by me for overcoming.

BETA TEST: -

Will be conducted at the organization site and user will frequently reports me for correction.

In both the above tests our work will go right by any one of the test because our software is just an application program not a system program.

SOFTWARE TESTING

TESTING OBJECTIVES: -

- i. Testing is a process of executing a program with the intent of finding an error.
- ii. A good test case is one that has a high probability of finding an undiscovered error.
- iii. A successful test is one that uncovers as yet undiscovered error.

TESTING TECHNIQUES: -

Once source code has been generated, software must be tested to uncover (and correct) as many errors as possible before delivery to our customer. Our goal is to design a series of test cases that have a high likelihood of finding errors.

SOFTWARE IS TESTED FROM TWO DIFFERENT PERSPECTIVES: -

- a. Internal program logic is exercised using “White Box” test case design techniques.
- b. Software requirements are exercised using “Black box” test case design techniques. In both cases, the intent is to find the maximum number of errors with the minimum amount of effort and time.

WHITE BOX TESTING:-

White box testing sometimes called Glass box testing, is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- I. Guarantee that all independent paths within a module have been exercised at least once.
- II. Exercise all logical decisions on their true and false sides.
- III. Exercise internal data structures to ensure their validity. Logic errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed. Errors tend to creep into our work when we design and implement function, conditions, or control that is out of the mainstream.

Typographical errors are random. When a program is translated into programming language source code, it is likely that some typing errors will occur. Many will be uncovered by syntax and type checking mechanisms, but others may go undetected until testing begins. White-Box Testing is far more likely to uncover them.

BLACK- BOX TESTING:-

Black Box testing, also called Behavioural Testing, focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black Box testing is not an alternative to white-box techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods. Black Box testing attempts to find errors in the following categories: -

- Incorrect or missing functions.
- Interface errors.
- Errors in data structures or external data base access.
- Behaviour or performance errors.
- Initialization and termination errors.

TESTING PROCEDURE :-

Different type of checks like duplicate checks, completeness check, validity, checks etc. are incorporated in this system, as the data has to be entered in different forms. The user is not familiar with new system the data entry screens are designed in such a way that they are

- ◆ Consistent
- ◆ Compatible
- ◆ Easy to use
- ◆ Had quick response

The following conventions are used while designing of the various screens to make the system user friendly

- ◆ All the items that are logically related are together.
- ◆ Error and validation messages are provided whenever required.
- ◆ System testing is against its initial objectives, it is done in a simulated environment.

TEST REVIEW :-

Test review is the process, which ensures that testing is carried out, as planned test review decides whether or not the program is ready to ship out for the implementation. For each data entry screen, we prepared test data with extreme values and under all relevant data- entry screen against real this process helped in rectifying the modules time.

Test Plan:

The test plan is an important, but often overlooked, product of software design. A test plan prescribes various kinds of activities that will be performed to demonstrate that the software product meets its requirements.

TEST PLAN SPECIFIES

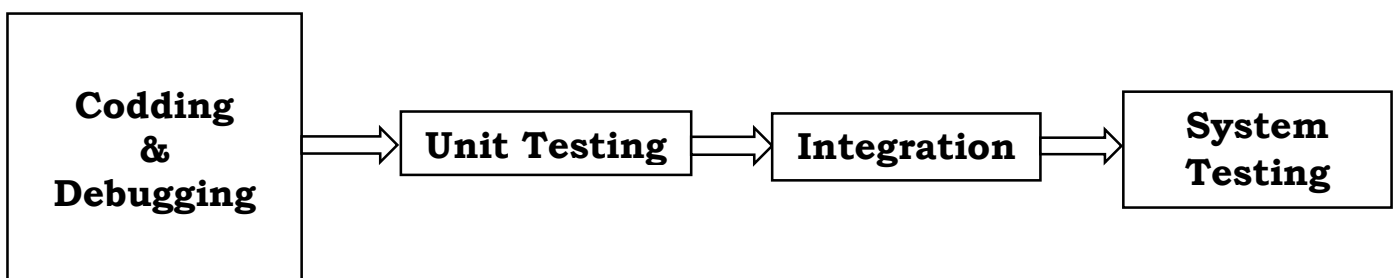
1. **THE OBJECTIVE OF TESTING:** It achieves error-free operation under stated conditions for the state period of time.
2. **THE TEST COMPLETION CRITERIA:** It helps to achieve a specified rate of error exposure, and a specified percent of logical path coverage.
3. **THE SYSTEM INTEGRATION PLAN:** It helps to specify strategy, schedule and responsible individuals of test
4. **METHOD TO BE USED:** It defines method of the test plan such as walkthroughs, inspections, static analysis, dynamic tests, formal verification and the particular test cases to be used.

Testing Level:

Three levels of testing were carried out in each phase:

- Unit testing : It tests a code function.
- Integration function : It checks interdependency of different modules.
- System Testing : It tests the applications' functionality as was specified in the detailed functionality detail.

THE TESTING LEVEL SITUATION IS ILLUSTRATED BELOW:



FUTURE SCOPE OF PROJECT

This “Veterinary Management System” is best suited to automate the Clinic. It can retain records for future references. This project has very much capacity to retaining records and query support is very powerful at each and every level. Accessing is done easily and efficiently because use of ORDBMS as back end. It can be accessed or manipulated according to our need. In future one can retrieve the old records.

This software has many other hidden advantages over the other software. Future applications of this software are limitless. This software can be used at any Clinic. This project is prepared in such a manner that, a few changes can be also done in less effort.

Following are the main future application of this software:

1. This software can be used as a basis to develop a full fledged Clinic Information System.
2. The proposed software can be further upgraded to handle other operations, if introduced in future by Clinic.
3. This software can be further developed for Routine management of Clinic.
4. In future the whole systems going to be online, fully java application.

IDENTIFICATION OF NEED

To identify the need for software we use the Principles of Requirement Engineering. Requirement engineering provides the appropriate mechanism for understanding what the customer wants, analysing need, assessing feasibility, negotiating a reasonable solution, specifying the solution unambiguously, validating the specification and managing the requirement as they are transformed into an operational system. The requirement engineering process can be described in five distinct steps: -

- Requirement elicitation.
- Requirement analysis & negotiation.
- Requirement specification.
- System Modelling.
- Requirement validation.
- Requirement Management.

In other words we can say that requirement analysis is a software task that bridges the gap between system level requirement engineering and software design. Requirement analysis allows the software engineering to refine the software allocation and build models of the data, functional and behavioral domains that will be treated by software. Requirement analysis provides the software designer with a representation of information, function and behavior that can be translated into data, architectural, interface and component level design; finally the requirement specification provides the developer and the customer with the means to assess quality once software is built.

The most commonly used requirement elicitation technique is to conduct a meeting or interview. The first meeting between a software engineer and

customer can be likened to the awkwardness of a first date between two adolescents. Neither person knows what to say or ask, Both are worried that what they do say will be misinterpreted, both are thinking about where it might lead (Both likely have radically different expectations here) Both want to get the thing over with, but at the same time, both want it to be a success.

CONCLUSION

The VETERINARY MANAGEMENT SYSTEM is a great improvement over the manual system using case fields and paper. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The Retail Shop management system was thoroughly checked and tested with dummy data and thus is found to be very reliable.

ADVANTAGES

- It is fast, efficient and reliable
- Avoids data redundancy and inconsistency
- Very user-friendly
- Easy accessibility of data
- Number of personnel required is considerably less
- Provides more security and integrity to data

LIMITATION

No project is complete and each and every project has its limitations. Also this project has some limitations. And in near future any one can improve this project and decrease its limitations. Limitations in this project are :

1. The system is not portable as it is a standalone application running on single system with no shared database.
2. The system is only applicable for outdoor patient.

BIBLIOGRAPHY

BOOKS

1. JAVA THE COMPLETE REFERENCE-Herbert Schildt
2. JAVA Black Book – R. Nageswara Rao
3. Database Management System – Pranab Kumar Das Gupta
4. Java Black book
5. Google the search engine

WEB SITES

1. www.tutorialspoint.com/java/
2. www.javatpoint.com/java-tutorial
3. www.w3schools.com/sql/
4. www.tutorialspoint.com/plsql/