A

PROJECT REPORT

ON

Euphoria Health Care

Submitted by

Banshita Gangwar (166140316027)
Pallavi Kher (166140316040)
Grishma Makadiya (166140316048)
Apexa Patel (166140316065)

In partial fulfillment for the award of the degree

Of

Diploma of Engineering

In

Information Technology

Government Polytechnic for Girls, Ahmedabad

Gujarat Technological University

October, 2018

Government Polytechnic for Girls,
Ahmedabad.

DECLARATION

I hereby declare that the Project Report submitted, along with the project entitled “ **EUPHORIA Health Care ”** submitted in partial fulfillment for the degree of **diploma in Engineering** in **IT** to Gujarat Technological University, Ahmedabad, is a bonafide record of the project work carried out at **Government Polytechnic for Girls, Ahmedabad** under the supervision of **Ms. Hemali Vithalani** and that no part of any of these project reports has been directly copied from any students’ reports or taken from any other source, without providing due reference.

Banshita Gangwar
Pallavi Kher
Grisham Makadiya
Apexa Patel

Government Polytechnic for Girls,
Ahmedabad.

CERTIFICATE

This is to certify that the project reports, submitted along with the project entitled **EUPHORIA Health Care** has been carried out by **Banshita Gangwar (166140316027), Pallavi Kher (166140316027), Grishma Makadiya (166140316048), Apexa Patel (166140316065)** under my guidance in partial fulfillment for the degree of **Diploma of Engineering** in **Information Technology** 5th **Semester** of Gujarat Technological University, Ahmedabad during the academic year 2018-19. These students have successfully completed project activity under my guidance.

Ms. Hemali Vithalani
Internal Guide
IT Department
GPG, Ahmedabad.

Ms. Priti Parikh
Head of Department
IT Department
GPG, Ahmedabad.

ACKNOWLEDGMENT

I take this opportunity to express my profound gratitude and deep regards to my guide Mrs. Hemali Vithalani for her exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to Ms. Priti Parikh H.O.D of Information Technology department for her cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am obliged to staff members of IT department, for the valuable information provided by them. I am grateful for their cooperation during the period of my project.

Lastly, I thank almighty, my parents, brother, sisters and friends for their constant encouragement without which this assignment would not be possible.

ABSTRACT

Usually people face certain problem in finding a skilled doctor in their nearby area. Nowadays people are having a hectic life, due to this sometimes they are unaware of their nearby area. Thus it’s quite difficult to find a skilled doctor and to maintain different medical reports. Therefore there is a need of an application which will resolve this problem. Team EUPHORIA comes with the solution. Team EUPHORIA’s aim is to provide a skilled doctor to its each and every user. With the help of EUPHORIA HEALTH CARE one can maintain their all report using QR code. QR code will be generated for each patient and it will contain the past report of the patient this will make easy for doctor to diagnose one disease and for patient maintaining their medical history. EUPHORIA HEALTH CARE is connected to the nearby medical store too. This will make one to get prescribed medicine easily. Other then Doctor and Patient, there are two modules of the application which are Pharmacist and Medical Representative. Medical representative make sure that ever area have a medical store and if any area does not any medical store then he/she submit a report to the EUPHORIA HEALTH CARE. He will also make a survey in area and gather the information about the disease spreading nowadays. Team EUPHORIA give awareness to the people about disease and how to the people take precaution about disease. EUPHORIA will give details about how to do first aid. In future we will try to connect blood bank to our application which will also make easy to donate blood and to get blood.

List of Content

Chapter Number **Page Number**

Declaration
Certificate
Acknowledgement
Abstract
Index / List Of Contents
List of Figures
List of Tables

Chapter 1: Introduction……………………………………………………… 10
1.1 Project Summary……………………………………….. 10
1.2 Project Purpose………………………………………. 10
1.3 Project Scope……………………………………………. 11
1.4 Literature Review……………………………………….. 11

Chapter 2: System Requirement Analysis and Project Management…….. 18
2.1 Tools and Technology…………………………………... 18
2.2 Hardware and Software Requirement…………………... 18
2.3 Software Process Model………………………………… 19
2.4 Project planning and scheduling………………………... 21
2.4.1 Project development approach…………………… 21
2.4.2 Project plan………………………………………. 22
2.4.3 Schedule Representation…………………………. 26

Chapter 3: System Analysis…………………………………………………….. 28
3.1 Requirement of new system…………………………….. 28
3.2 Features of New System………………………………... 29
3.3 Feasibility study………………………………………… 30
3.3.1 Technical feasibility……………………………... 30
3.3.2 Economical feasibility…………………………… 30
3.3.3 Operational feasibility…………………………… 30
3.4Database Schema Design………………………………... 31
3.4.1 Data Dictionary…………………………………... 31
3.5 Data Modeling…………………………………………… 38
3.5.1 ER Diagram……………………………………… 38
3.5.2 Activity Diagram………………………………… 39
3.5.3 Class Diagram ……………..……………………. 44

3.6 Functional and Behavioral modeling…………………….. 45
3.6.1 DFD Level 0 …………………………………….. 45
3.6.2 DFD Level 1 …………………………………….. 46
3.7 Functions of System ……………………………………. 47
3.7.1 Use Case Diagram……………………………….. 47

Chapter 4: Implementation …………………………………………………. 53

Chapter 5: Conclusion and future work……………………………………. 55

References……………………………………………………….. 56
List of Web References 56
List of Text References. 56

List of figure

FIGURE NO	NAME OF FIG	PAGE NO
Fig 1.1	MVC model	13
Fig 1.2	MVC model architecture	14
Fig 1.3	Hibernate architecture	15
Fig 2.1	Incremental model	20
Fig 2.2	Incremental model 2	22
Fig 2.3	Project Schedule	26
Fig 3.1	ER diagram	38
Fig 3.2	Activity Diagram for admin	39
Fig 3.3	Activity Diagram for patient	40
Fig 3.4	Activity Diagram for doctor	41
Fig 3.5	Activity Diagram for pharmacist	42
Fig 3.6	Activity Diagram for medical representative	43
Fig 3.7	Class diagram	44
Fig 3.8	0 level dfd	45
Fig 3.9	1 level dfd	46
Fig 3.10	Use case for admin	47
Fig 3.11	Use case for Doctor	48
Fig 3.12	Use case for medical representative	49
Fig 3.13	Use case for patient	50
Fig 3.14	Use case for pharmacist	51
Fig 3.15	Home page	53
Fig 3.16	Login page for doctor	53

List of Tables

TABLE NO	TABLE NAME	PAGE NO
Table 3.1	Address	31
Table 3.2	Admin	31
Table 3.3	Analysis	32
Table 3.4	Appointment	32
Table 3.5	Doctor	33
Table 3.6	Emergency Services	33
Table 3.7	Feedback	34
Table 3.8	Health Tracker	34
Table 3.9	Medical Representative	35
Table 3.10	Patient	35
Table 3.11	Payment	36
Table 3.12	Pharmacist	36
Table 3.13	Pharmacy	37
Table 3.14	Prescription	37
Table 3.15	Report	37

Chapter: 1

INTRODUCTION

1.1 Project Summary

People faces problem in finding a skilled doctor in their nearby area. Nowadays people are having a hectic life, due to this sometimes they are unaware of their nearby area. Thus it’s quite difficult to find a skilled doctor and to maintain different medical reports. Therefore there is a need of a application which will resolve this problem.

Team EUPHORIA comes with the solution. Team EUPHORIA’s aim is to provide a skilled doctor to its each and every user. With the help of EUPHORIA HEALTH CARE one can maintain their all report using QR code. QR code will be generated for each patient and it will contain the past report of the patient this will make easy for doctor to diagnose one disease and for patient maintaining their medical history. EUPHORIA HEALTH CARE is connected to the nearby medical store too. This will make one to get prescribed medicine easily. Other then Doctor and Patient, there are two modules of the application which are Pharmacist and Medical Representative. Medical representative make sure that ever area have a medical store and if any area does not any medical store then he/she submit a report to the EUPHORIA HEALTH CARE. He will also make a survey in area and gather the information about the disease spreading nowadays.

Team EUPHORIA give awareness to the people about disease and how to the people take precaution about disease. EUPHORIA will give details about how to do first aid. In future we will try to connect blood bank to our application which will also make easy to donate blood and to get blood.

1.2 Project Purpose

- The purpose of this project is to connect end user to with this organization for the medical remedies.
- We have seen many remote areas are not served gracefully especially village & slum area are overlook & treated as secondary aspect.
- Our purpose is to connect this people with the organization via different entities like doctor, medical representative and pharmacist.
- Another purpose is to give ontime service in any area.
- Prompt delivery of medicines.
- Our main purpose is to see our city disease free.
- Precise survey of disease and finding accurate cure for that disease
- Dynamic report updation, dynamic checking.

1.3 Project Scope

- Main aim of our application is to serve patients with online medical facilities like online appointment, online doctor consultation, online patient track record, online doctor rating and optional feature of medicines' home delivery.

- The patients who can't travel to doctor can directly consult with them through online doctor consultation feature provided in our application.

- Patients can check the ratings of doctors and select the best doctor for themselves.

- Some patient can't invest a lot of time waiting in a long queue in doctors' clinic can use our application for online appointment.

- In some cases, patients forget to take their previous reports, x-rays, ECG, sonography so to overcome this problem our application serves with online patient track record which will store the x-rays, ECG, sonography reports form the day the user will start using our application.

- Some patients who want to investigate the ingredients of the medicine and the proportion of the ingredient used can surf our application.

- Patients who are capable of walking properly and don't want home delivery can use our offline medicines option in which we will provide the nearest shop address and their number.

- After a patient has consulted the doctor, the doctor will send the prescription to the patient online, and the user will have to decide that he wants online medicines or offline medicines.

1.4 Technology & Literature Review

The system uses J2EE Environment and JAVA1.8 as code language, Hibernate and Apache Maven as an Extended Frameworks, back end side My SQL Database, Development Tool as Eclipse Oxygen, database management Tool as SQLyog Enterprise.
Below there is some explanation about these Terminologies.

➢ Apache Tomcat Server 8.0

The Apache Software Foundation provides support for the Apache community of open-source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field.

Apache Tomcat is the Servlet container that is used in the official Reference Implementation for the Java Servlet and Java Server Pages technologie**s.**

- **Advantages:--**
  - Performance optimizations and reduced garbage collection
  - Refactored application deplorer, with an optional standalone deplorer allowing validation and compilation of a web application before putting it in production.
  - Complete server monitoring using JMX and the manager web application
  - Scalability and reliability enhancements
  - Improved Taglibs handling, including advanced pooling and tag plugins
  - Improved platform integration, with native Windows and Unix wrappers
  - Embedding using (Java Management Extension) JMX
  - Enhanced Security Manager support
  - Integrated session clustering.

➢ J2EE platform

The Java 2 Platform, Enterprise Edition (J2EE), developed by Sun Microsystems, is a platform-independent development environment for Java based enterprise applications. J2EE provides support for a number of components which extend the Java 2 Platform to facilitate the development of web-based applications; namely Enterprise 2005)].

- **Software design patterns:**

  Software design patterns are pre-defined solutions to recurring problems in software engineering. Although many patterns exist which can be applied to a wide range of platforms, the Sun JavaCenter has also published a set of patterns relating specifically to J2EE development projects [Alur, et al (2003)]. As Winn and Calder (2002) explain, adherence to software design patterns can significantly aid the process of development by providing expertly verified solutions and accepted standards for good practice.

- **The Model View Controller (MVC) pattern:**

  One design pattern which has been adopted across a wide range of software development projects and platforms is the *Model View Controller (MVC) pattern*. The main concept behind MVC is the separation of design concerns in application development into three layers – the Model, View and Controller. The model component manages the business logic and application state, the view renders the model for display to the user and the controller translates interactions with the view into actions performed by the model. Figure 1.1 shows how the MVC layers are connected.

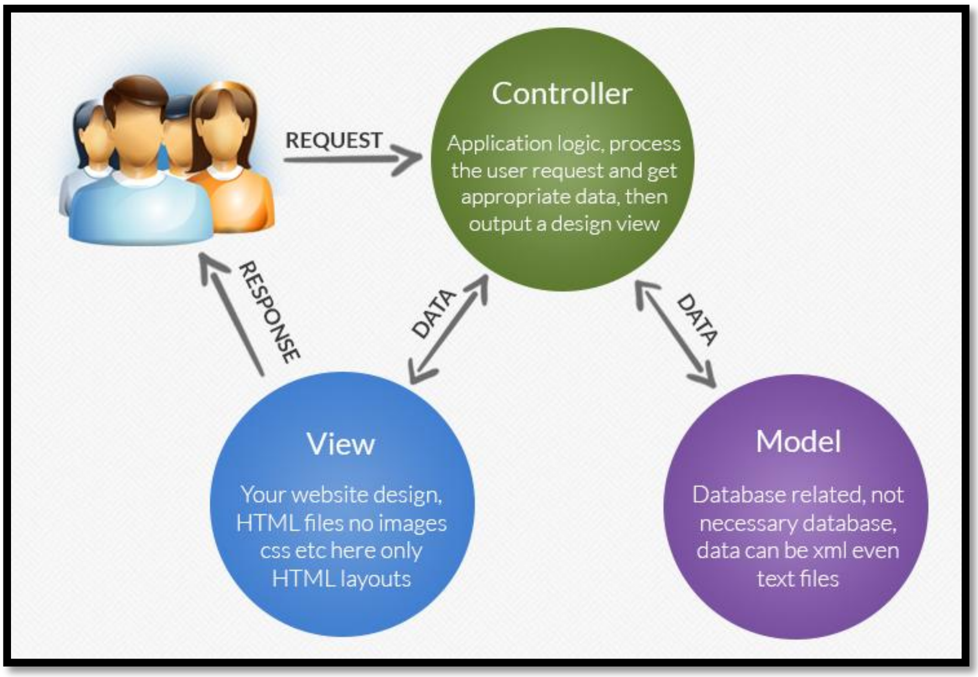
REQUEST
Controller
Application logic, process the user request and get appropriate data, then output a design view
RESPONSE
DATA
DATA
View
Your website design, HTML files no images css etc here only HTML layouts
Model
Database related, not necessary database, data can be xml even text files

Fig 1.1 :- MVC Model

The separation of design concerns through the use of MVC exposes a number of possibilities, including the ability to build a number of alternative view components which utilize the same model and controller. In addition, collaborative developments can be approached with more confidence and ease when the various components have been clearly separated.

As we mentioned above, J2EE technologies can be combined in order to achieve more effective application designs. In terms of J2EE developments, the MVC design pattern is embodied by the so-called *Model 2* architecture where Java Beans are used to represent the model layer, JSP pages for the view and Java Servlets for the application controller. When a request is sent to a *Model 2* application, the *controller servlet* initiates state changes in the model though the instantiation and manipulation of one or more JavaBeans. At the stage where a response is to be generated, the*controller servlet* selects an appropriate JSP page to supply the view. Figure 1.2 provides a diagrammatic representation of a typical *Model 2* application.

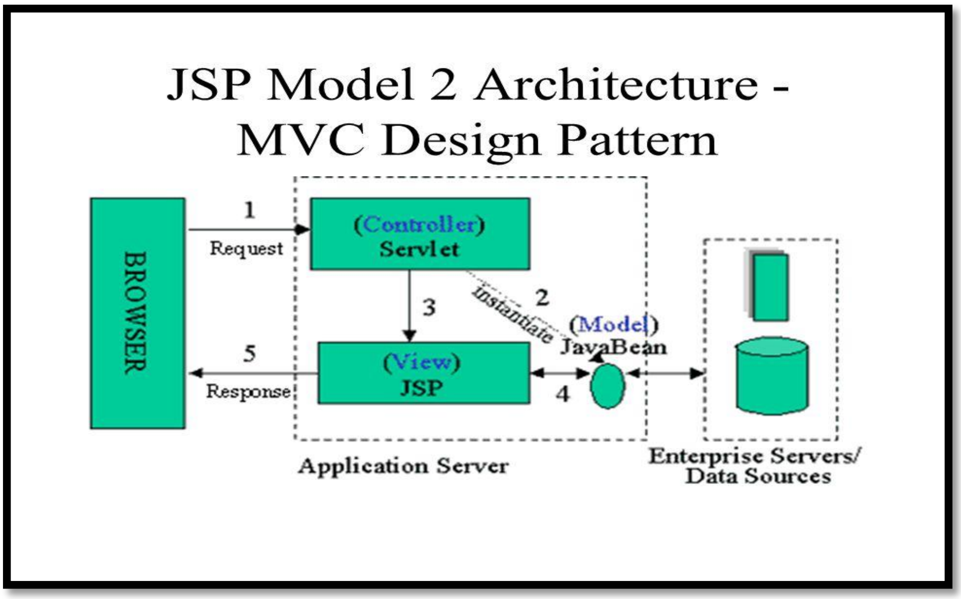

Fig 1.2 - MVC Model 2 application

Although the typical *Model 2* framework provides an effective means of decoupling the MVC design concerns; the controller component can be criticised for having a lack of maintainability. Utilising a collection of servlets to provide the application controller exposes the opportunity to reuse model and view components for different operations. However, this carries the drawback of having replicated code in each servlet for performing common controller operations. The repercussion of replication, more pertinent in larger developments, is a reduced level of maintainability - where more components of the application will be impacted when a change needs to be made.

➢ JSP

Java Server Pages (JSP) is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. JSP was designed to address the perception that the Java programming environment didn't provide developers with enough support for the Web.

JSP is a high-end technology that helps developers insert java code in HTML pages by making use of special JSP tags. The JSP are HTML pages but do not automatically have .html as file extension. JSP files have .jsp as extension.

The main features of JSP technology are as follows:

- A language for developing JSP pages, which are text-based documents that describe how to process a request and construct a response
- An expression language for accessing server-side objects
- Mechanisms for defining extensions to the JSP language

➢ Hibernate

Hibernate is an open source object/relational mapping tool for Java. Hibernate lets you develop persistent classes following common Java idiom - including association, inheritance, polymorphism, composition and the Java collections framework. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities and can significantly reduce development time otherwise spent with manual data handling in SQL and JDBC.

Hibernates goal is to relieve the developer from 95 percent of common data persistence related programming tasks. Hibernate is Free Software. The LGPL license is sufficiently flexible to allow the use of Hibernate in both open source and commercial projects (see the LicenseFAQ for details). Hibernate is available for download at http://www.hibernate.org/. This tutorial aims to provide insight into Hibernate version 3.0RC and its usage

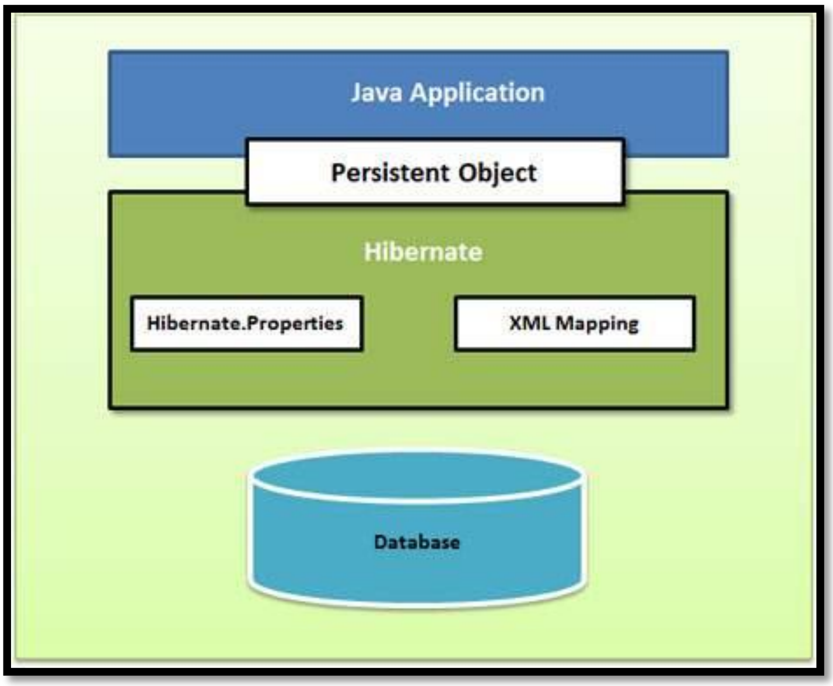

Fig 1.3: Hibernate Architecture

Hibernate architecture has three main components:

- **Connection Management:**
  Hibernate Connection management service provide efficient management of the database connections. Database connection is the most expensive part of interacting with the database as it requires a lot of resources of open and close the database connection.

- **Transaction management:**
  Transaction management service provides the ability to the user to execute more than one database statements at a time.

- **Object relational mapping:**
  Object relational mapping is technique of mapping the data representation from an object model to a relational data model. This part of the hibernate is used to select, insert, update and delete the records form the underlying table. When we pass an object to a session.save () method, Hibernate reads the state of the variables of that object and executes the necessary query.

Chapter:2

SYSTEM REQUIREMENT ANALYSIS AND PROJECT MANAGEMENT

2.1 Tools and Technology

The system uses J2EE Environment and JAVA1.8 as code language, Hibernate and Apache Maven as an Extended Frameworks, back end side My SQL Database, Development Tool as Eclipse Oxygen, database management Tool as SQLyog Enterprise.
Below there is some explanation about these Terminologies.

The technologies which are used in this project are :-

- The **Java** 2 Platform, Enterprise Edition (J2EE), developed by Sun Microsystems, is a platform-independent development environment for Java based enterprise applications. J2EE provides support for a number of components which extend the Java 2 Platform to facilitate the development of web-based applications; namely Enterprise 2005)].

- The **Apache Software** Foundation provides support for the Apache community of open-source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field.

- **Java Server Pages** (JSP) is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. JSP was designed to address the perception that the Java programming environment didn't provide developers with enough support for the Web.

2.2 Hardware and Software Requirement

➢ ***Hardware Configuration of Server Side***

Processor	:	Intel core i3 Processor
RAM	:	4 GB
Hard Disk Drive	:	500 GB
Key Board	:	Standard 101/102 or Digi Sync Family
Monitor	:	Display Panel (1366 X 768)
Mouse	:	Serial Mouse
Internet Connection	:	Yes

- ***Software Requirement for Server Side***

Technology	:	JSP
Frameworks	:	Hibernate
Server	:	Apache Tomcat Server
Basic Features	:	Html, CSS
Other Tools	:	Apache Maven Repository
Database	:	MySQL 5.1
Development IDE	:	Eclipse Oxygen

- ***Minimum System Configuration of Client Side***

Processor	:	INTEL Pentium 4 or Higher
RAM	:	512MB
Hard Disk Drive	:	40GB
Monitor	:	Display Panel (1024 X 764)
Mouse	:	Serial Mouse
Internet Connection	:	Yes
Browser	:	Google Chrome, Mozilla Firefox 8.0 or higher Internet Explorer 6.0 or higher

2.3 Software Process Model

The Requirements provided by the users are converted into Users Requirement Specification as described above. The URS documents are then revised, validated, authorized and approved by the users. The development commences after the approval phase i.e. after the signing off of the URS documents. Thus, the URS is concerned to be the most important document from user and developer prospective. The Developer will try to adhere to the requirements specified in the URS documents in order to develop the required application.

The model used in this project is Incremental & iterative development cycle. Incremental development is a staging and scheduling strategy in which various parts of the system are developed at different times or rates, and integrated as they are completed.

Iterative development is a rework scheduling strategy in which time is set aside to revise and improve parts of the system. It is important to notice that neither strategy presupposes, requires or implies the other.

➢ Iterative and Incremental Development

In incremental development, we do each of those activities multiple times. That is, we go around the requirements – design – Implementation – testing – integration – delivery cycle multiple times. We "iterate" through that cycle multiple times.

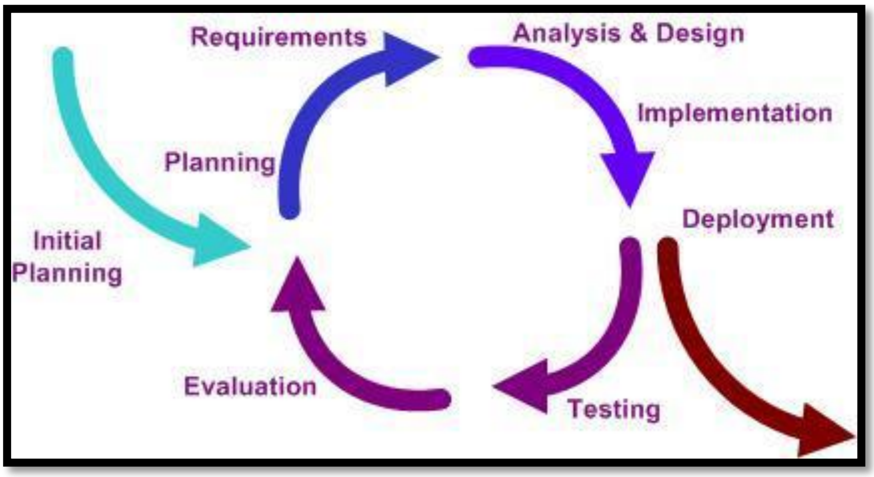

Fig 2.1 incremental model

This model contains six task regions:

- **Initial Planning/ Customer communication:** Tasks required establishing effective communication between developer and customer. During this phase the objectives, alternatives and constraints are elaborated.
- **Planning:** Tasks required defining resources, timelines and other project related information.
- **Analysis and Design:** A detailed analysis of the requirements gathered is assessed for both technical and management risks. A design of the product is carefully created which is supposed to fulfill the client specifications.
- **Implementation and Testing:** The actual coding starts from here according to the prototypes proposed during the design phase. Each application module is thoroughly tested for consistency and requirement fulfillment. Tasks required building one or more representations of the application.
- **Deployment and release:** Tasks required constructing, testing, installing and providing user support (e.g. documentation and training). Customer evaluation: Tasks required obtaining customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage (Pressman 1997).

- **Advantages of Incremental Model**
  - Generates working software quickly and early during the software life cycle.
  - More flexible – less costly to change scope and requirements.
  - Easier to test and debug during a smaller iteration.
  - Easier to manage risk because risky pieces are identified and handled during its iteration.
  - Each iteration is an easily managed milestone.

- **Disadvantages of Incremental Model**
  - Each phase of an iteration is rigid and do not overlap each other.
  - Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

2.4 Project planning and scheduling

2.4.1 Project development approach

The activities we followed for this project is listed below:

- Planning the work or objectives
- Analysis & Design of objectives
- Assessing and controlling risk
- Allocation of resources
- Organizing the work
- Database Designing
- Form Design

The Process Paradigm we used for our project is Incremental Model.

➢ The Incremental Software Process Model

The Incremental Model combines elements of the linear sequential model with the iterative philosophy of prototyping. The incremental model applies linear sequences in a staged fashion as calendar time progresses.

Each linear sequence produces a deliverable “increment” of the software. For example, word processing software developed using the incremental paradigm might deliver basic file management, editing and document production functions in the first increment; more sophisticated editing and document production capabilities in the second increment; spelling and grammar checking in the third increment; and advanced page layout capability in the fourth increment.

It should be noted that the process flow for any increment can incorporate the prototyping paradigm.

When an incremental model is used, the first increment is often a core product. That is, basic requirements are addressed, but many supplementary features remain undelivered. The core product is used by the customer. As a result of use and/or evaluation, a plan is developed for the next increment. The plan addresses the modification of the core product to better meet the needs of the customer and the delivery of additional features and functionality.

This process is repeated following the delivery of each increment, until the complete product is produced. The Incremental process model, like prototyping and other evolution approaches, is iterative in nature.

But unlike prototyping, the incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down versions of the final product, but they do provide capability that serves the user and also provide a platform for evaluation by the user.

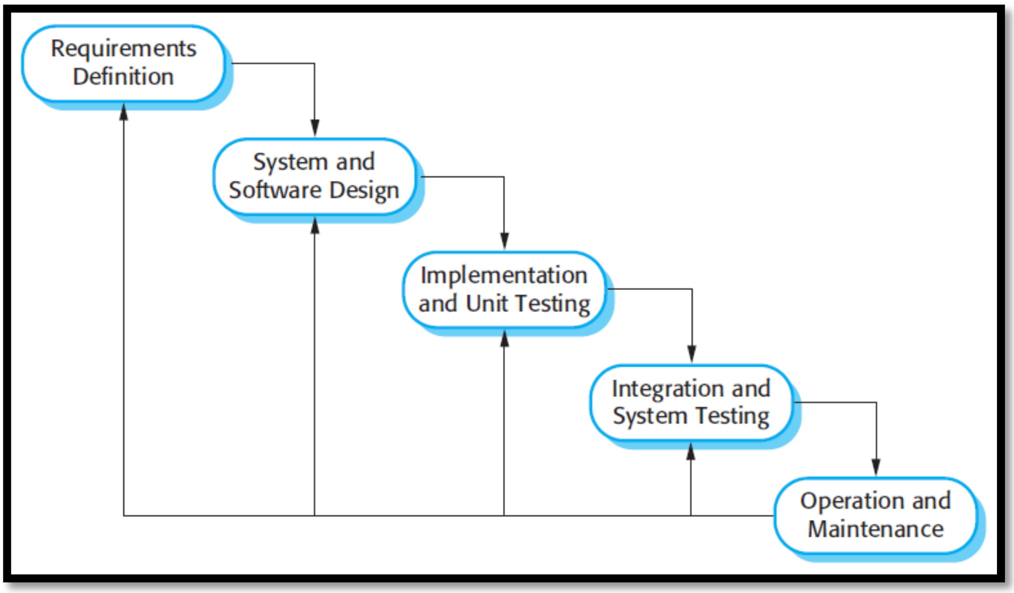
Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Fig:-2.2 incremental model 2

2.4.2 Project plan

LSS has a very long-standing relationship at client side. It has provided its expertise to various Government agencies, working with numerous hardware and software technologies, to improve its overall image and provide better services to its constituents. LSS will bring its vast domain expertise in legacy and low firm project, knowledge of the best practices and lessons learnt in various implementations at client side to this project. For this specific requirement analysis, LSS serve with the help of designated personnel from CLIENT would familiarize themselves with the overall structure of the office and it's inter-linkages with other departments/systems and branch. They will understand the broad working of the Company and the other related departments, its functions, manpower employed, processes etc.

LSS shall meet the representative personnel and document the detailed gaps between the current system and the proposed solution based on the understanding of the activities and processes, as they are currently being carried out in the office.

The gaps would be documented and analyzed, and if required and possible reengineering of process in concurrence with CLIENT to leverage the benefits of the product would be suggested.

Design Constraints:

Design specifications that are followed in the organization have to be followed which are as follows:

Programming Convention:

- Programming style should be modular in the sense that it is easy to identify the functionality of the module and its relationship with other modules.
- Module should be predetermined and be clearly defined.
- Intercommunication within same module should be high but intercommunication between different modules should be minimum, i.e. (Cohesion should be high but coupling should be low.)
- Reusable routines should be used to maximize productivity.
- Code should not contain any hard-coded components.
- Program should be readable, well indented, and visible within the screen by writing the command in a structured format and using word-wrap to next line for long statements.
- Appropriate comments should be used to describe the purpose of the program, statement, routines, and variables.
- User log should be maintained for concurrency control.
- The system should provide some sort of authentication for diff. categories of user.

Look And Feel

- Look and feel of the system should be consistent throughout the system.
- Same color combination for each page as well as fonts and use of safe color for web-based applications.
- Similar look and Standard navigation style.
- Consistent usage of caption for same purpose controls.
- Interface should be user friendly and self-explanatory.
- Appropriate use of tooltip for displaying relevant information to the user.
- Use of appropriate titles and metadata keywords/description for each page.
- Entire screen layout should be accommodated in one screen and if required should be vertically scrolled.

- **Input Output Conventions**
  - Take maximum input by selection and minimize text input that would help in reducing user level data entry errors.
  - System should take minimum input from user and reduce redundant data input. E.g. if once birth date is entered system should not take age from user but it should be calculated automatically.
  - All report layouts should be formatted to accommodate printing in A4 size paper only.
  - All report should have feature to view on screen and save in Excel format.
  - All queries /reports should have facility to extract the data between specified filter. E.g. Between range of dates, and complex search.

- **Database Design**
  - Table should be normalized to have minimum redundancy and exhibit consistency.
  - Relationships should be well defined, easy to identify and maintain.
  - Appropriate nomenclature should be done such that it indicates what type of data it contains.
  - Fields should be of proper data types.
  - Use proper data types as required. E.g. -Use date data type wherever date is required instead of varchar.
  - Proper backup strategy should be maintained, script files for the tables should be generated.
  - Primary keys and foreign key constraints should be appropriately defined.
  - Indexing mechanisms should be used for faster retrieval of information.
  - Use of Views should be done for security and precise output.

- **Standards for Development Environment**
  - Centralized storage in server i.e. No client level replica
  - Meaningful project name
  - Meaningful main folder naming convention.
  - Nature of object based separation.
  - Duplicate object with different name should be avoided.
  - No direct changes in database level object. Source file should be modified first.
  - Final source should be migrated to central software library.
  - Copy of technical document always with the source should be provided.

  - Maintain a document to describe the change history date wise.
  - No direct changes to the live server should be made; changes should be incorporated first in the development and should be migrated to live only after testing and finalizing.

- **Database Connectivity**
  - A global file like spring config. file should be maintained for database connection string for flexibility, updating and only reference to that file should be used while programming.

- **Quality and Testing**
  - Test plans and Test data should be well documented.
  - Unit testing should be performed at programmer level.
  - A senior programmer to check working of the system as a whole should do integration testing.
  - Functional testing and Boundary case testing should be done.

- **Performance**
  - Minimum usage of graphics is required for fast access to pages.
  - Application should be platform independent.
  - Low response time and security measures for user information and other important information.
  - User information should be used to give user a personalized environment.

- **System Security**
  - Security measures for user information and other important information.
  - User should be authenticated using a common login procedure to use the system.
  - A user log should be maintained in order to keep a check on proper usage of the system.
  - All the sensitive information stored in database such as password should be in encrypted form.

2.4.3 Schedule Representation

Scheduling the project is an important activity. It involves deciding which tasks would be taken up when .Project Manager of project has done these following tasks.

- Break down large tasks into small activities.

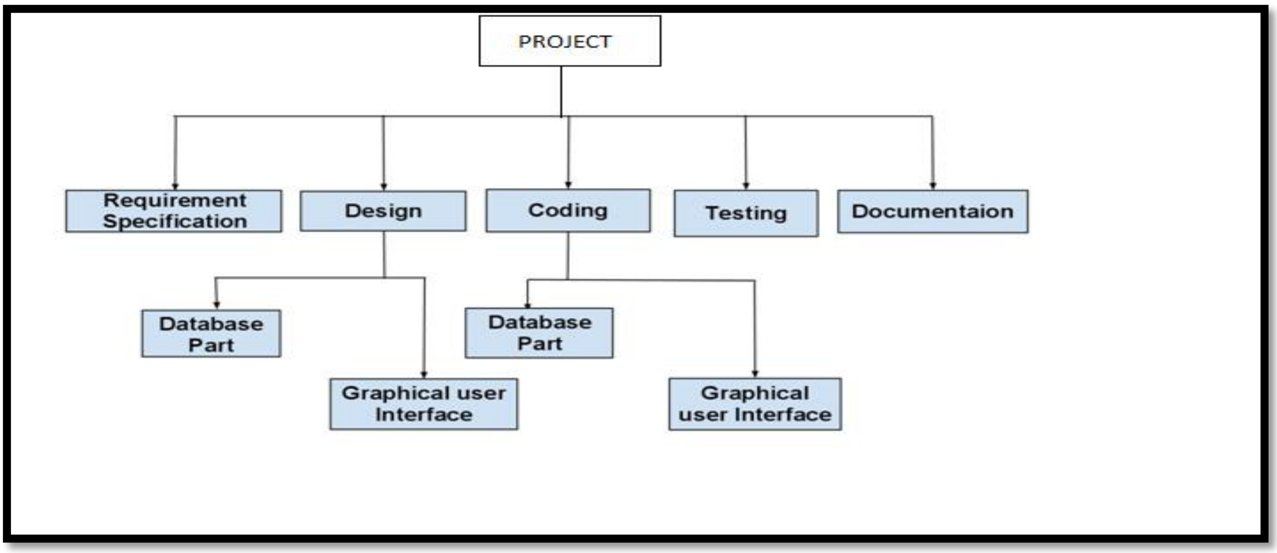

Fig 2.3 Project Schedule

Chapter : 3

SYSTEM ANALYSIS

3.1 Requirement of new system

- Main aim of our application is to serve patients with online medical facilities like online appointment, online doctor consultation, online patient track record, online doctor rating and optional feature of medicines' home delivery.

- The patients who can't travel to doctor can directly consult with them through online doctor consultation feature provided in our application.

- Patients can check the ratings of doctors and select the best doctor for themselves.

- Some patient can't invest a lot of time waiting in a long queue in doctors' clinic can use our application for online appointment.

- In some cases, patients forget to take their previous reports, x-rays, ECG, sonography so to overcome this problem our application serves with online patient track record which will store the x-rays, ECG, sonography reports from the day the user will start using our application.

- Some patients who want to investigate the ingredients of the medicine and the proportion of the ingredient used can surf our application.

- Patients who are capable of walking properly and don't want home delivery can use our offline medicines option in which we will provide the nearest shop address and their number.

- After a patient has consulted the doctor, the doctor will send the prescription to the patient online, and the user will have to decide that he wants online medicines or offline medicines.

- These unique features of our application differs our application from other applications.

- Our aim is to serve our users with smooth and trouble free medical services, which can make India Digital as it is the motto of our Prime minister to make India "Digital India".

3.2 Features of new System

- **Book Health check-up:**
  Choose the healthcare plan according to your age and need. You can book it online or through our Toll free number. We shall fix an appointment for you in our network hospital of your choice. Visit the hospital on the date of appointment and get the check-up done, meet the consultant physician and go back home with your reports

- **Oral Hygiene pack (Dental Solutions):**
  Euphoria Health Care has Dental pack is designed to take care of your oral health. Meet a dentist, fill cavities if any, get a tooth X-Ray if required, get your teeth cleaned and polished and go back home.

- **Doctor Consultation:**
  Euphoria Health Care has partnered with General Physicians in all over the city. All you have to do is, call us or book online an appointment and meet the Physician nearest to you.

- **Track your health:**
  Euphoria Health Care has various health trackers for you to keep a track of your Blood Pressure, Blood Glucose levels, weight and other health parameters which you can share it with your Physician at a later date. You can also know your BMI, Blood Volume, Water intake etc....

- **First Aid Information:**
  Simple step-by-step instructions guide for the first aid of sudden illness, health issues and injuries

- **Blood Bank and Lab Information:**
  Locate and interact with the Blood banks (currently limited to demo) nearest to you. Currently, we will be providing our services in demo mode only. In Days to come, we shall be present in other cities across India.

- **The Medical representative:**
  The Medical representative’s role is to survey market and areas and find uncertain calamities and diseases if any and report it to Euphoria Health Care. He will also make survey in the different area and assure that every area must have a medical store and if it is not then he will upload the report on Euphoria care.

- **Direct connection to medical store:**
  Medical store are directly connected to Euphoria, by which patients get discount and direct home delivery of medicines. Medical stores who receives prescription from the Doctor and notify patient for delivery of medicine

- **Cost efficient :**
  Prescribe any patients based on reports is also cost effective.

- **QR code:**
  Patient's entire medical record will be maintained using QR code. Each patient will get a unique QR code which will contain pervious medical history and the current reports of the patient.

- **Emergency services:**
  Emergency services, Direct Helpline etc are also part of Euphoria.

3.3 Feasibility study

3.3.1 Technical Feasibility

This type of the feasibility includes that can the work for the project is done with current equipment, existing software technology, and available market. If new technology is required, what is the likelihood that it can be developed?

We are going to use current technologies .Java and its framework, for frontend Designing and MySQL for back-end designing.

3.3.2 Economical Feasibility

This type of the feasibility include that a system that can be developed precisely and that will be used if installed must still be a good information for the advertising.

During the study of economic feasibility there are some of the questions which are generated by us are as follows:

- What is the rate of the hardware and software?
- How much manpower required?

Here manpower comes indirectly into the cost of developing the system so.

3.3.3 Operational Feasibility

This type of feasibility is troubled with whether after the installation of the completed system it will operate properly or not. System will be beneficial to the organization if it can operate properly and it supports all specified requirements of the user. This part of the feasibility study was tough because of two reasons: Software development is done at some other place and implementation on was done at other place.

We do not have any contacts with the would-be users of our system.

3.4Database Schema Design

3.4.1 Data Dictionary

- **Address: -** It stores the details of user’s address.

Address				
Fieldname	**Datatype**	**Size**	**Constraint**	**Description**
add_id	bigint	-	primary key	It indicate the id of address.
add_detail	varchar	250	not null	It indicate the address of the pateint.
area	varchar	50	-	It indicate the area of the address.
landmark	varchar	50	-	It indicates the landmark of patients address.
city	varchar	50	not null	It indicate the address's city of the pateint.
pincode	numeric	6	not null	It indicate the address of the pateint.
type	varchar	15	not null	It indicate the type of address (either professional or private).
ifd	bit	1	not null	It Describe the data is deleted or not (IsFlagDeleted).

Table 3.1: Address

- **Admin:-** It stores the details of admin’s details.

ADMIN				
FieldName	**DataType**	**Size**	**Constraint**	**Description**
ad_id	bigint	-	Primary Key	It indicates the admin id
log_email	varchar	50	not null	It indicate user's email.
log_password	varchar	13	not null	It indicate user's password.

Table 3.2: Admin

- **Analysis:-** It stores the details of analysis uploaded by medical representative.

ANALYSIS				
Fieldname	**Data type**	**Size**	**Constraint**	**Descripiton**
ana_id	bigint	-	primary key	It indicates the id of Analysis.
ana_descrption	text	500	not null	It indicates the description of Analysis.
ana_date	date	-	not null	It indicates the date of Analysis.
Ifd	bit	1	not null	It Describe the data is deleted or not (IsFlagDeleted).

Table 3.3: Analysis

- **Appointment:-** It stores the details of patient's appointment.

APPOINTMENT				
Fieldname	**Data type**	**Size**	**Constraint**	**Description**
app_id	bigint	-	primary key	It indicate the id of appointment.
Patient	bigint	-	foreign key	it indicate that appointment belong to which patient.
Doctor	bigint	-	foreign key	it indicates that appoinment is for which doctor.
booking_date	dateTime	-	not null	It indicate the date of booking.
appointment_date	dateTime	-	not null	It indicate the date of appointment.

Table 3. 4: Appointment

- **Doctor:-** It stores the details of doctor.

DOCTOR				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
dr_id	bigint	-	primary key	It indicate the id of doctor.
log_email	varchar	50	not null	It indicate user's email.
log_password	varchar	50	not null	It indicate user's password.
dr_name	varchar	20	not null	It indicate the name of doctor.
dr_gender	varchar	15	not null	It indicate the doctor's gender
dr_qualificaton	varchar	50	not null	It indicate the Qualification of doctor.
personal_address	bigint	-	foreign key	It indicate the personal address of doctor.
clinic_address	bigint	-	foreign key	It indicate the the address of doctor's clinic.
dr_mobile	numeric	10	not null	It indicate the contact number of doctor.
dr_dob	date	-	not null	It indicate the doctor's date of birth.
Idf	bit	1	-	It Describe the data is deleted or not (IsFlagDeleted)

Table 3.5: Doctor

- **EmergencyServices:-** It stores the details of Emergency Services.

EMERGENCY SERVICES				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
es_id	bigint	-	Primary key	it indicate the id of emergency services
es_name	varchar	20	not null	it indicate the name of emergency services
es_mobile	numeric	10	not null	it indicate the mobile number of emergency services
Ifd	bit	-	not null	It Describe the data is deleted or not (IsFlagDeleted)

Table 3.6: Emergency Services

- **Feedback:-** It stores the details of feedback.

FEED BACK				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
f_id	bigint	-	primary key	it indicate the id of feedback
f_email	varchar	50	not null	it indicate the email address of the user who post the feedback
f_desc	varchar	150	not null	it indicate the description of the feedback
f_date	date	10	not null	it indicates the date of feedback

Table 3.7: feedback

- **HealthTracker:-** It stores the details of patient's health.

HEALTH_TRACKER				
Field Name	**Data Type**	**Size**	**Constraint**	**Descrption**
h_id	bigint	-	primary key	It indicate the id of health.
Patient	bigint	-	foreign key	it indicate the id of patient
blood_glucose	float	5,2	-	it indicates blood glucose of patient
blood_pressure	float	5,2	-	it indicates blood pressure of patient
blood_volume	float	5,2	-	it indicate blood volume of patient
Weight	float	5,2	-	it indicates weight of patient
Height	float	5,2	-	it indicates height of patient
created_date	date	-	not null	it indicates the current date of the record insertion.

Table 3.8: Health_Tracker

- **MedicalRepresentative:-** It stores the details of Medical Representative.

MEDICAL REPRESENTATIVE				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
mr_id	Bigint	-	primary key	it indicates the id of medical representative.
log_email	varchar	50	not null	It indicate user's email.
log_password	varchar	13	not null	It indicate user's password.
mr_name	varchar	20	not null	it indicate the name of medical representative
mr_workarea	varchar	150	not null	it indicate the workarea of medical repesentative
mr_dob	date	-	not null	it indicate the date of birth of medical representative
mr_gender	varchar	15	not null	it indicate the gender of medical representative
Address	bigint	-	foregin key	it indicates the address of medical representative.
Ifd	bit	1	not null	It Describe the data is deleted or not (IsFlagDeleted)

Table 3.9: Medical representative

- **Patient:-** It stores the details of patient.

PATIENT				
Field Name	**DataType**	**Size**	**Constraint**	**Description**
patient_id	bigint	-	Primary key	It indicates the patient id
log_email	varchar	50	not null	It indicate user's email.
log_password	varchar	13	not null	It indicate user's password.
patient_name	varchar	20	not null	It indicates the name of patient.
Address	bigint	-	foreign key	It indicates the id of user's address in address table.
patient_dob	date	-	not null	It indicates the patient date of birth
patient_qrcode	varchar	500	not null	It indicates the path of patient’s QR code
patient_mobile	int	-	not null	It indicates the patient mobile number
patient_gender	varchar	15	not null	It indicates the patient gender
Ifd	bit	1	not null	It describe the data is deleted or not (IsFlagDeleted)

Table 3.10: Patient

- **Payment:-** It stores the details of payment.

PAYMENT				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
p_id	bigint	-	primary key	It indicate the id of payment.
pharmacy	bigint	-	foreign key	It indicate the id of pharmacy.
Amount	float	10,2	not null	It indicate the total amount of medicine and the bitges of doctor.
p_date	datetime	-	not null	It indicate the date of payment.
prescription	bigint	-	foreign key	It indicate the id of prescription.

Table 3.11: Payment

- **Pharmacist:-** It stores the details of pharmacist.

PHARMACIST				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
phst_id	bigint	-	Primary key	It indicates the pharmacist id
phst_name	varchar	20	not null	It indicates the pharmacist name
Address	bigint	-	foreign key	It indicates the pharmacist address
phst_mobile	numeric	10	not null	It indicates the pharmacist mobile number
phst_dob	date	-	not null	It indicates the pharmacist date of birth
phst_gender	varchar	15	not null	It indicates the pharmacist gender
log_email	varchar	50	not null	It indicate user's email.
log_password	varchar	13	not null	It indicate user's password.
Ifd	bit	1	not null	It Describe the data is deleted or not (IsFlagDeleted)

Table 3.12: Pharmacist

- **Pharmacy:-** It stores the details of pharmacy.

PHARMACY				
FieldName	**Data Type**	**Size**	**Constraint**	**Description**
phcy_id	bigint	-	primary key	It indicates the pharmacy id
phcy_name	varchar	20	not null	It indicates the pharmacy name
Address	varchar	50	foreign key	It indicates the pharmacy address
pharmacist	bigint	-	foreign key	It indicates the pharmacist id (owner of record)
phcy_mobile	numeric	10	not null	It indicates the pharmacy mobile number

Table 3.13: Pharmacy

- **Prescription:-** It stores the details of prescription.

PRESCRIPTION				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
pr_id	Bigint	-	primary key	It indicate the id of prescription.
prescribed_by	Bigint	-	foreign key	It indicate the id of doctor who prescribed the prescription.
prescribed_to	bigint	-	foreign key	It indicate the id of pateint to whom the prescription is prescription.
pr_path	varchar	250	not null	It indicate the id of doctor who prescribed the prescription.
created_date	date	-	not null	It indicate the date of creation.

Table 3.14: Pharmacy

- **Report:-** It stores the details of patient’s report.

REPORT				
Field Name	**Data Type**	**Size**	**Constraint**	**Description**
r_id	bigint	-	primary key	It indicate the id of report.
Patient	bigint	-	foreign key	It indicate the id of patient.
Doctor	bigint	-	foreign key	It indicate the id of doctor who suugested the user to have a report.
r_date	date	-	not null	It indicate the uploaded date of report.
r_path	varchar	500	not null	It indicate the path of report.

Table 3.15: Report

3.5 Data Modeling

3.5.1 ER Diagram

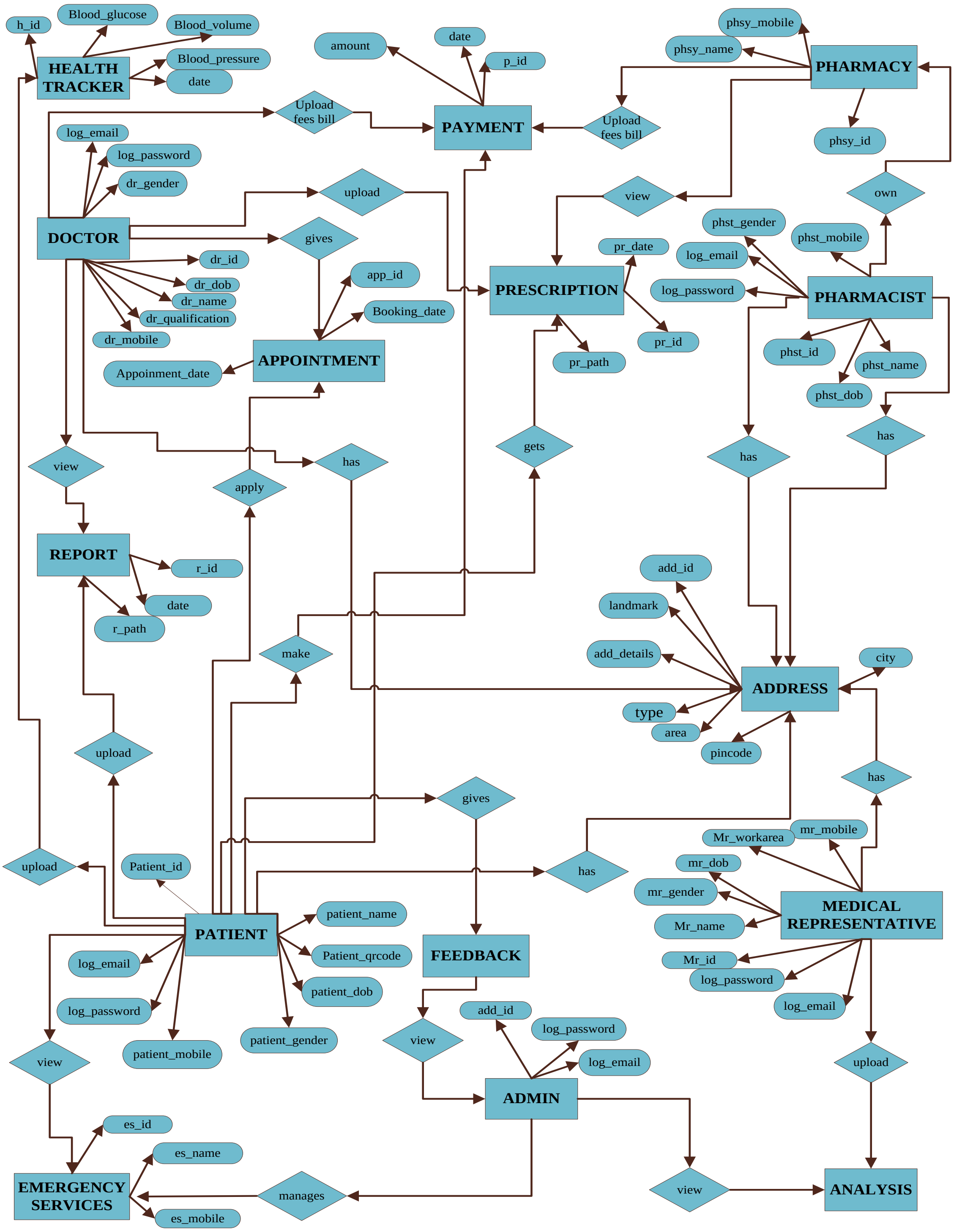

Fig:3.1 ER diagram

3.5.2 Activity Diagram

- **Activity for admin**

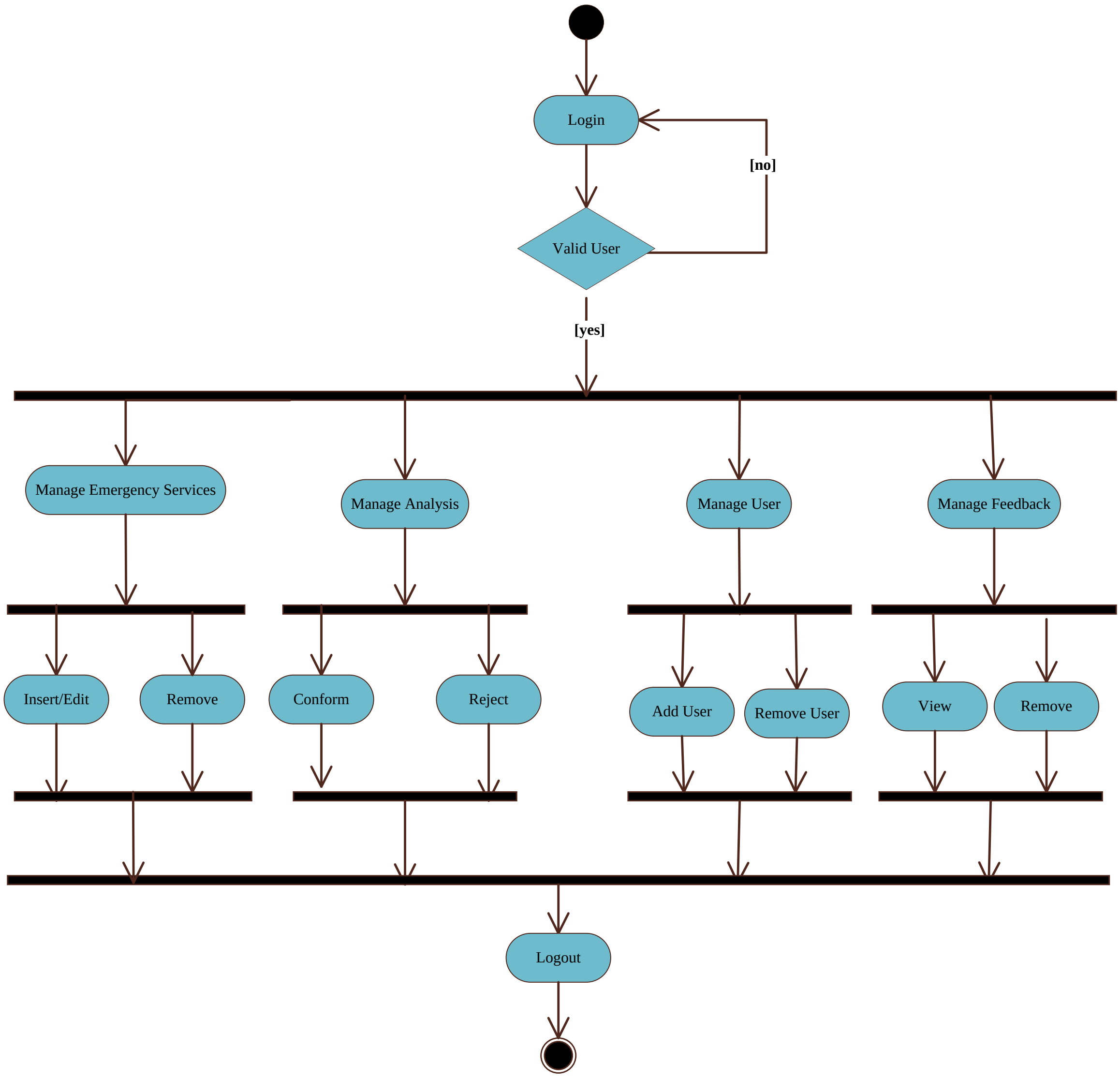

Fig 3.2 activity diagram for admin

➢ Activity diagram for Patient.

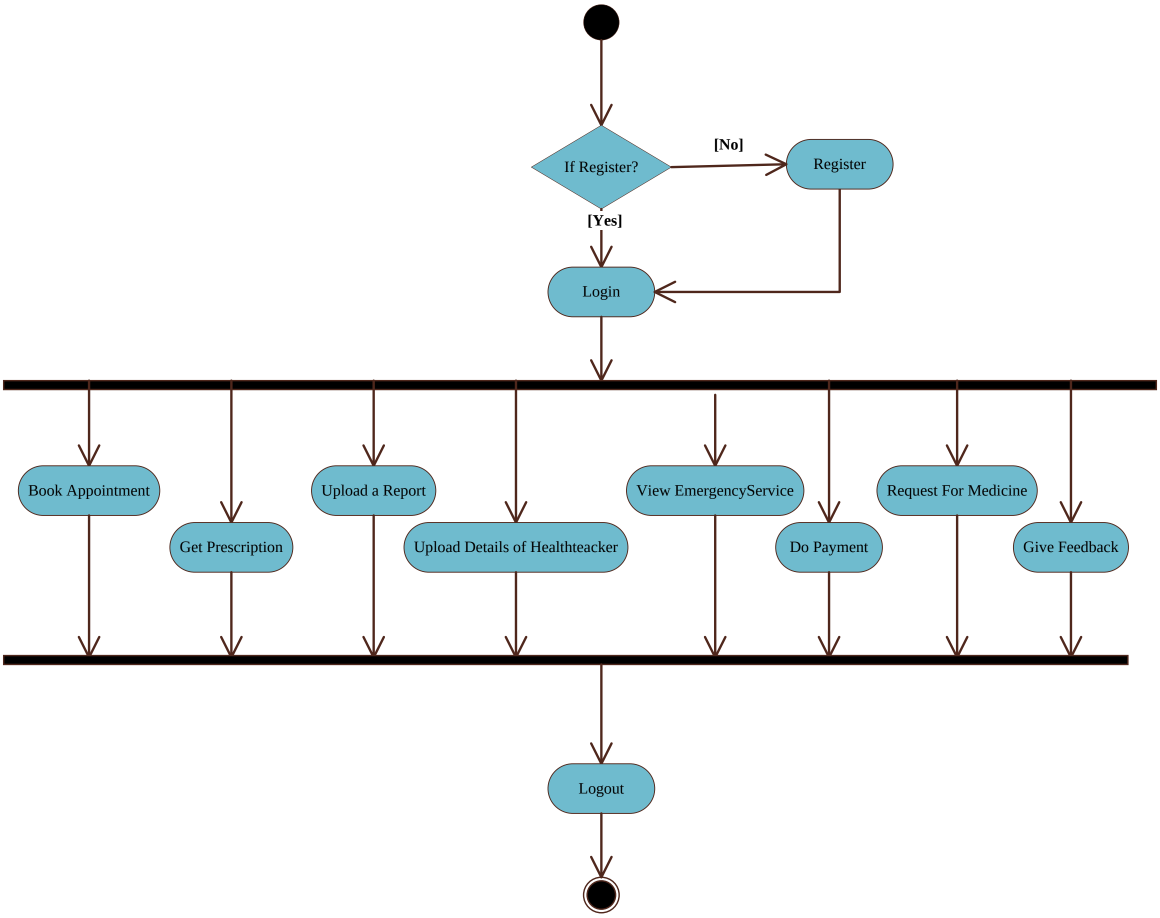

Fig 3.3 activity diagram for patient

➢ Activity diagram for Doctor.

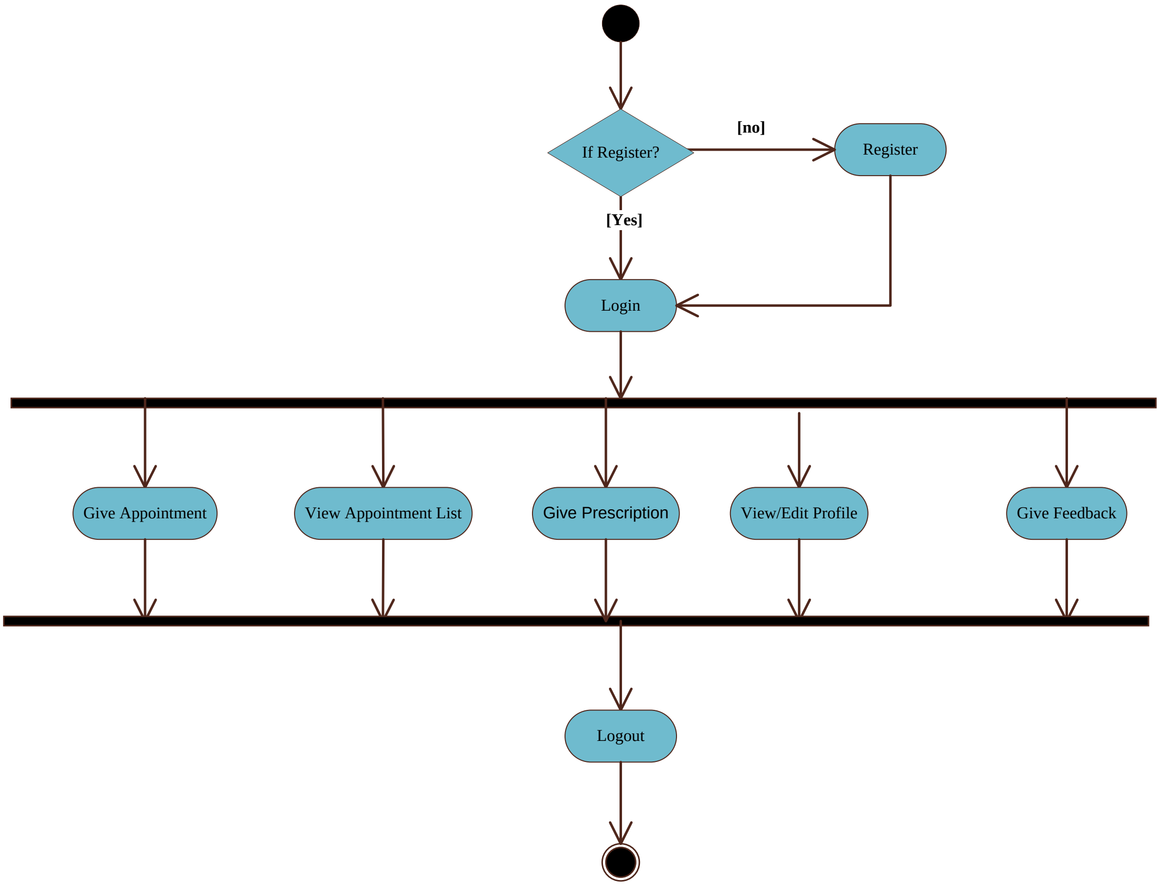

Fig 3.4 activity diagram for doctor

➢ Activity diagram for Pharmacist.

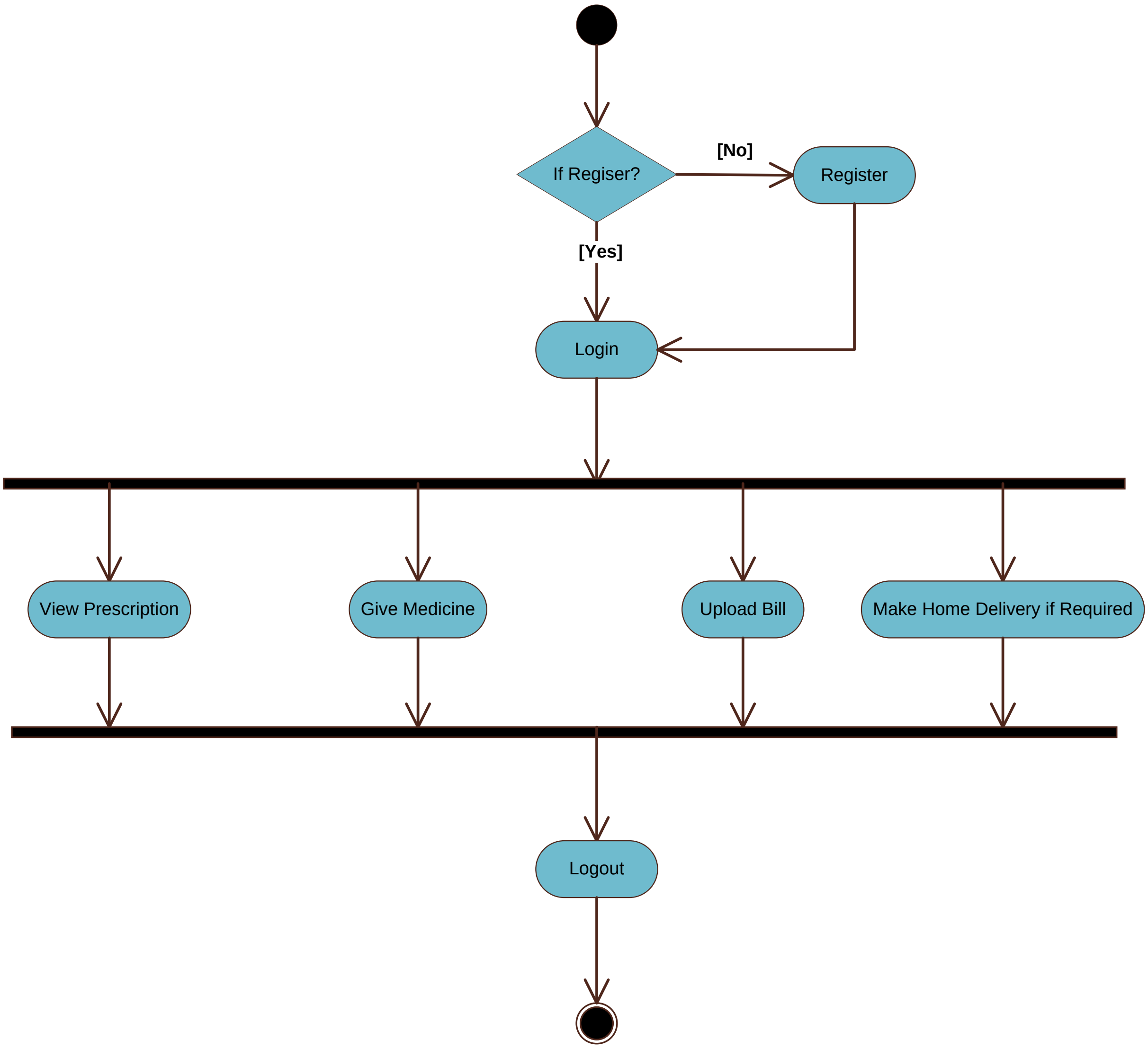

Fig 3.5 activity diagram for pharmacist

➢ Activity diagram for Medical Representative.

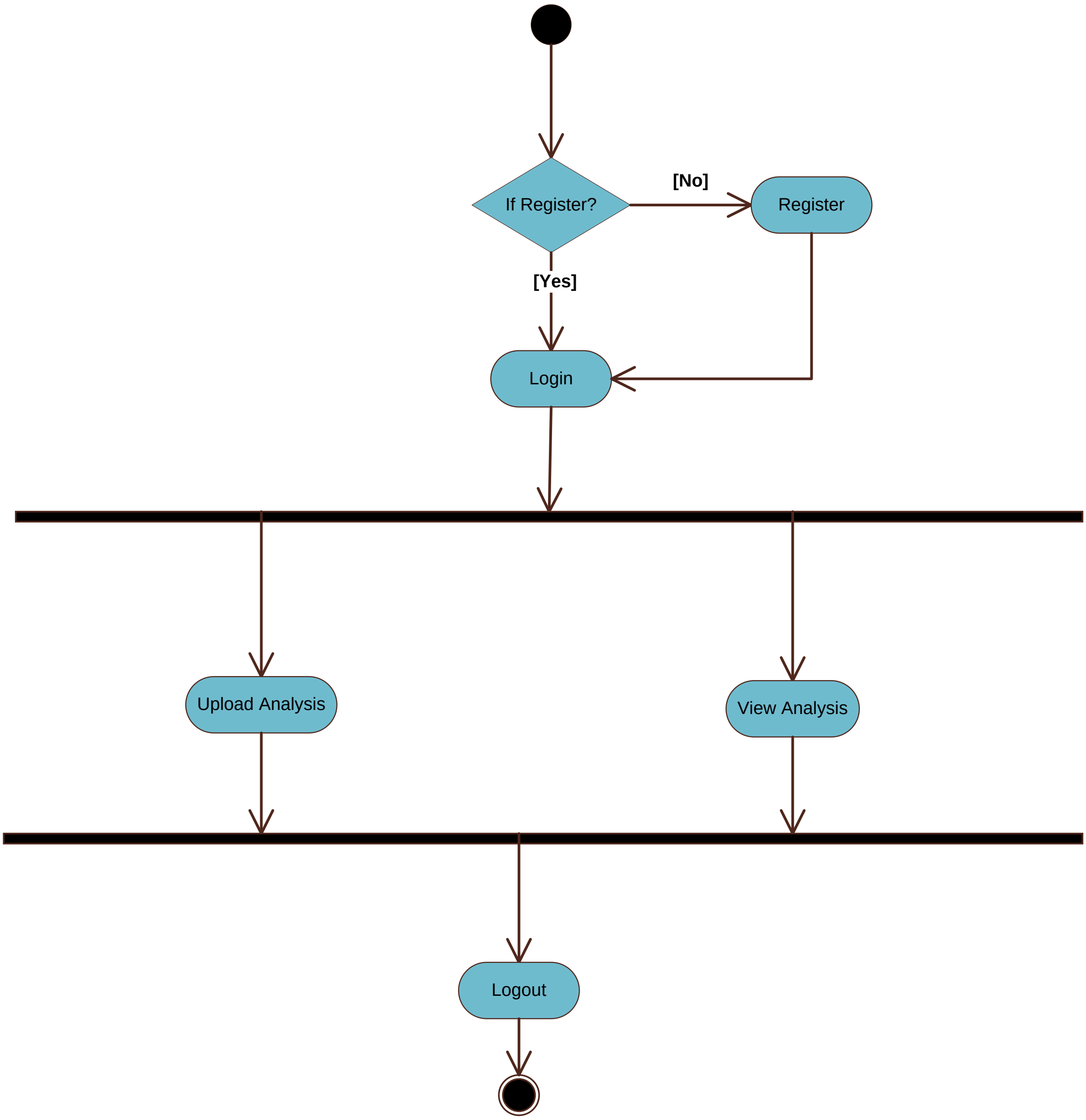

Fig 3.6 activity diagram for medical representative

5.3.4 Class diagram

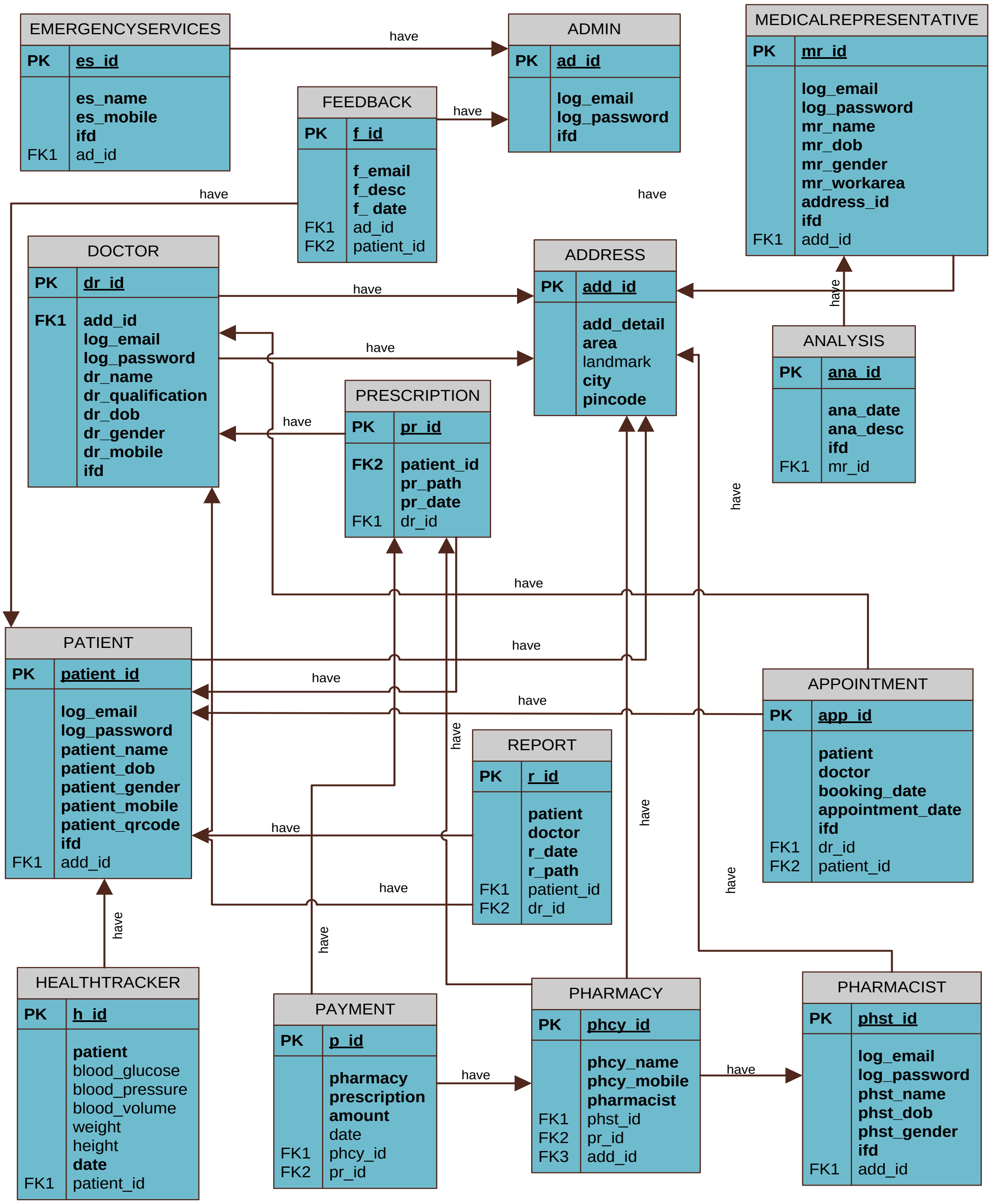

Fig 3.7 class diagram

3.6 Functional and Behavioral modeling

3.6.1 DFD Level 0

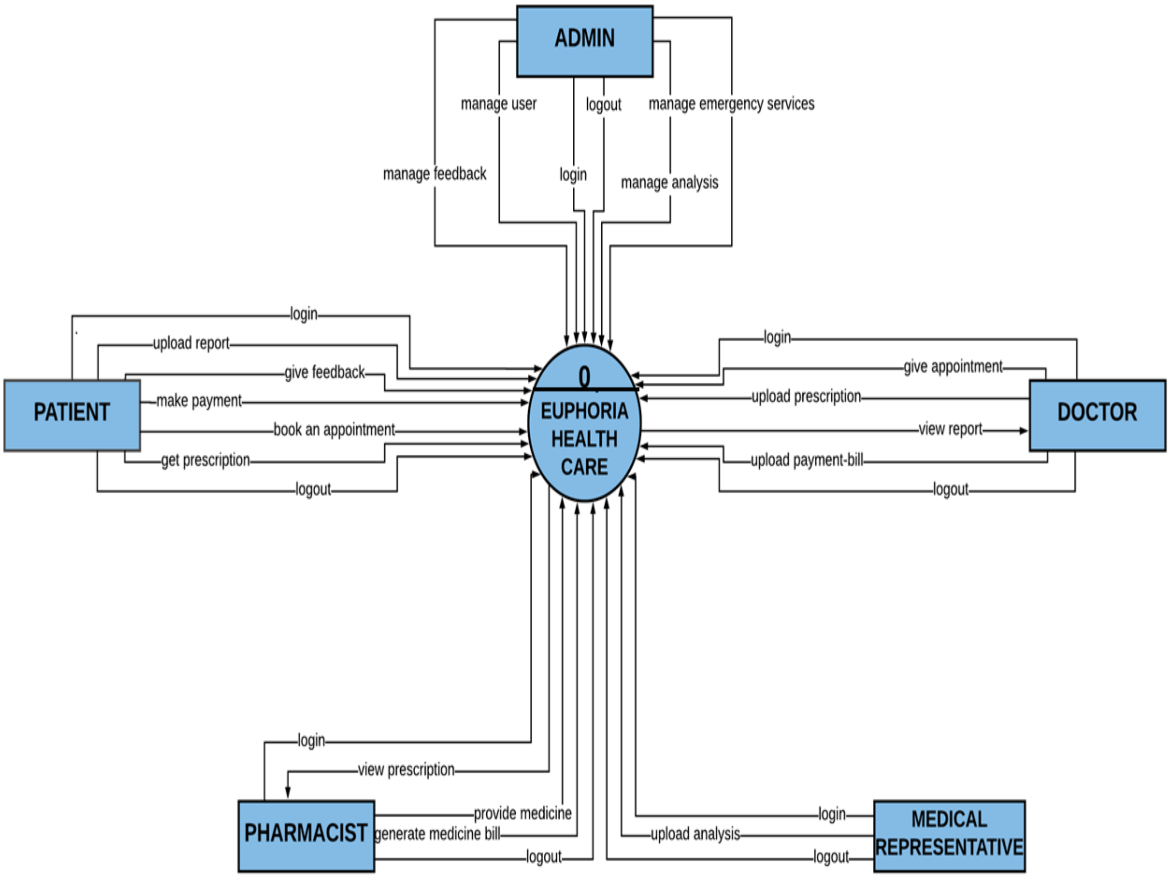

Fig 3.8 context level dfd

3.6.2 DFD Level 1

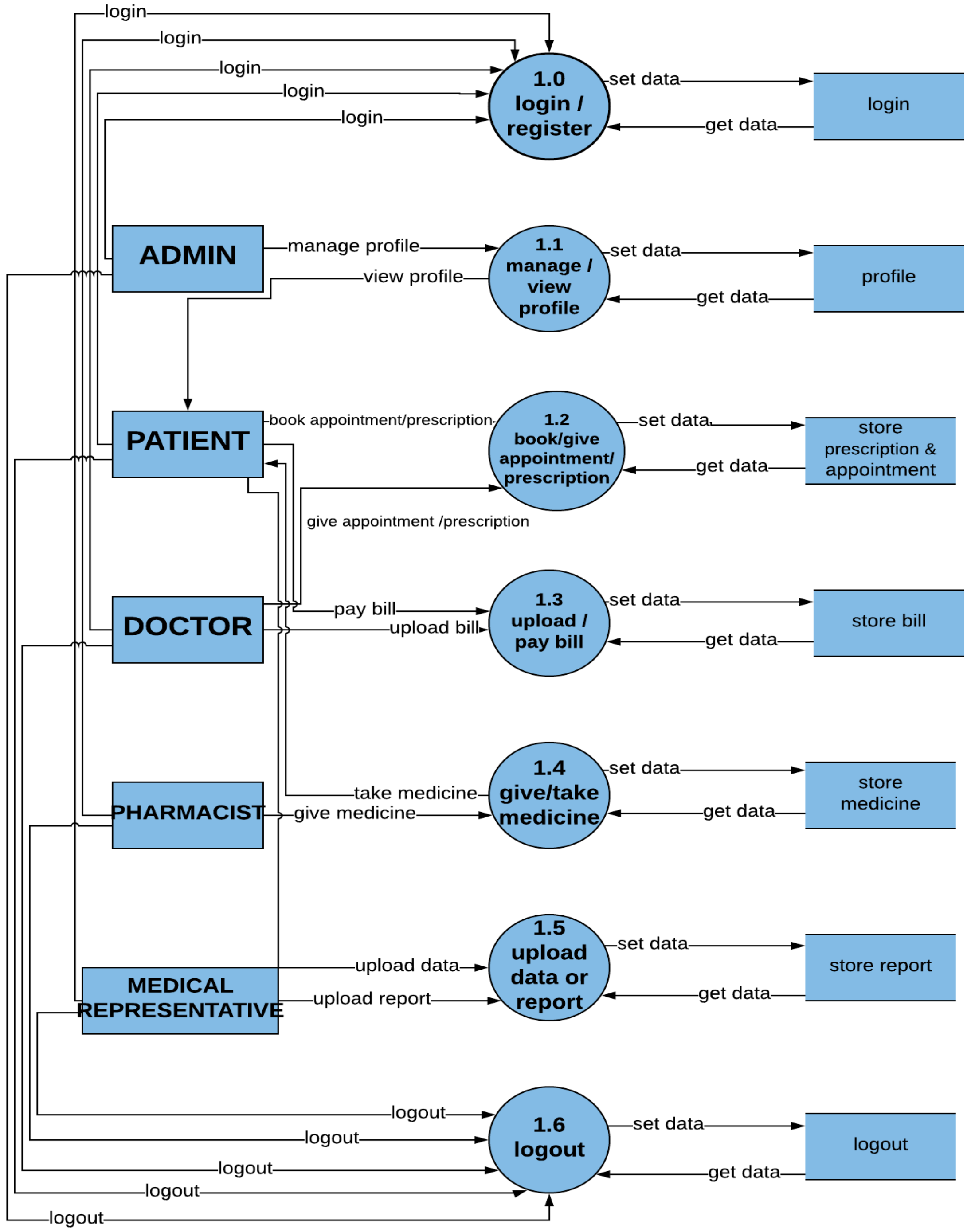

Fig 3.9 level -1 dfd

3.7 Functions of System

3.7.1 Use Case Diagram

- **Use case for admin.**

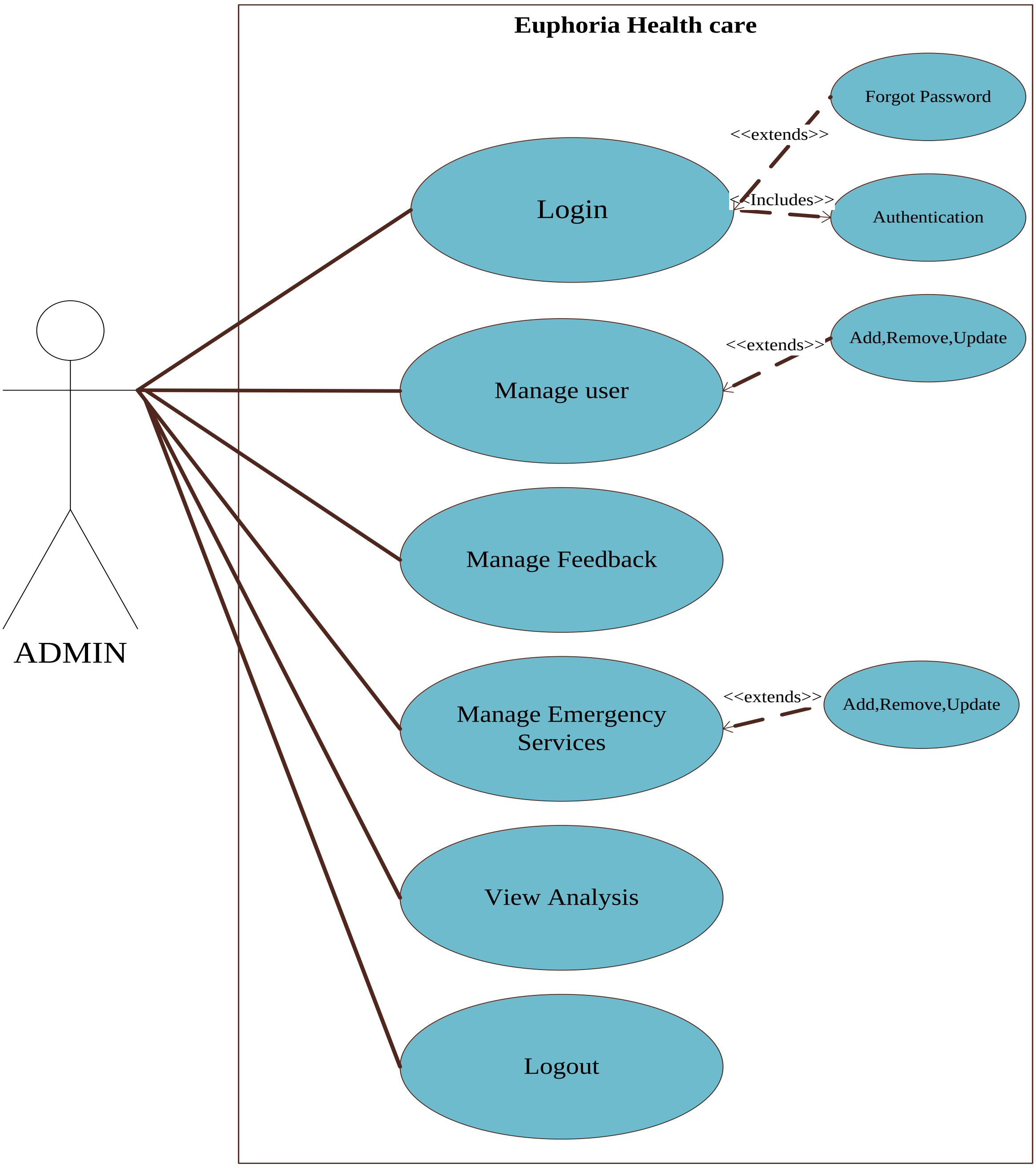

Fig 3.10 use case for admin

- **Use case for Doctor.**

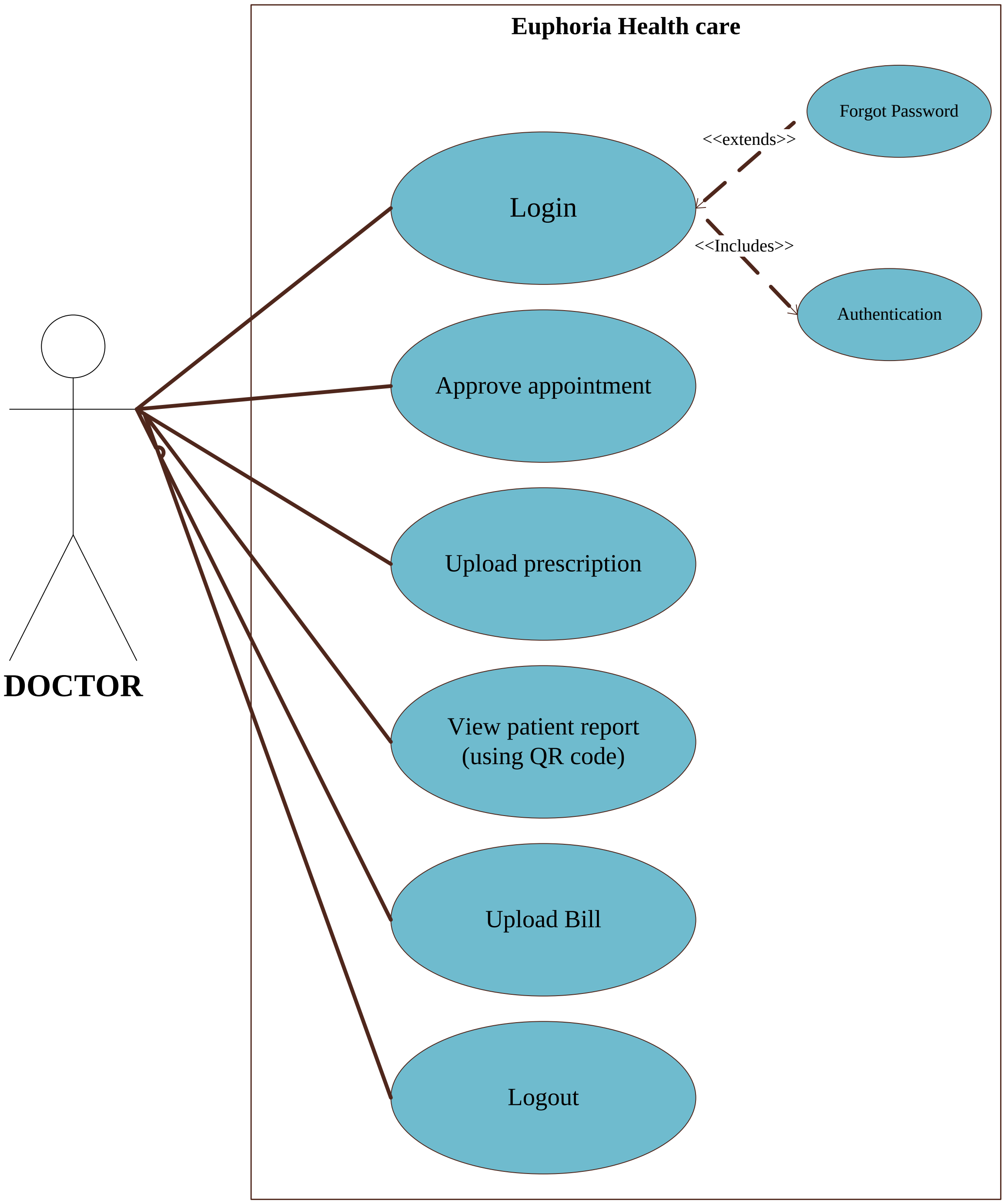

Fig 3.11 use case for doctor

- **Use case for Medical Representative.**

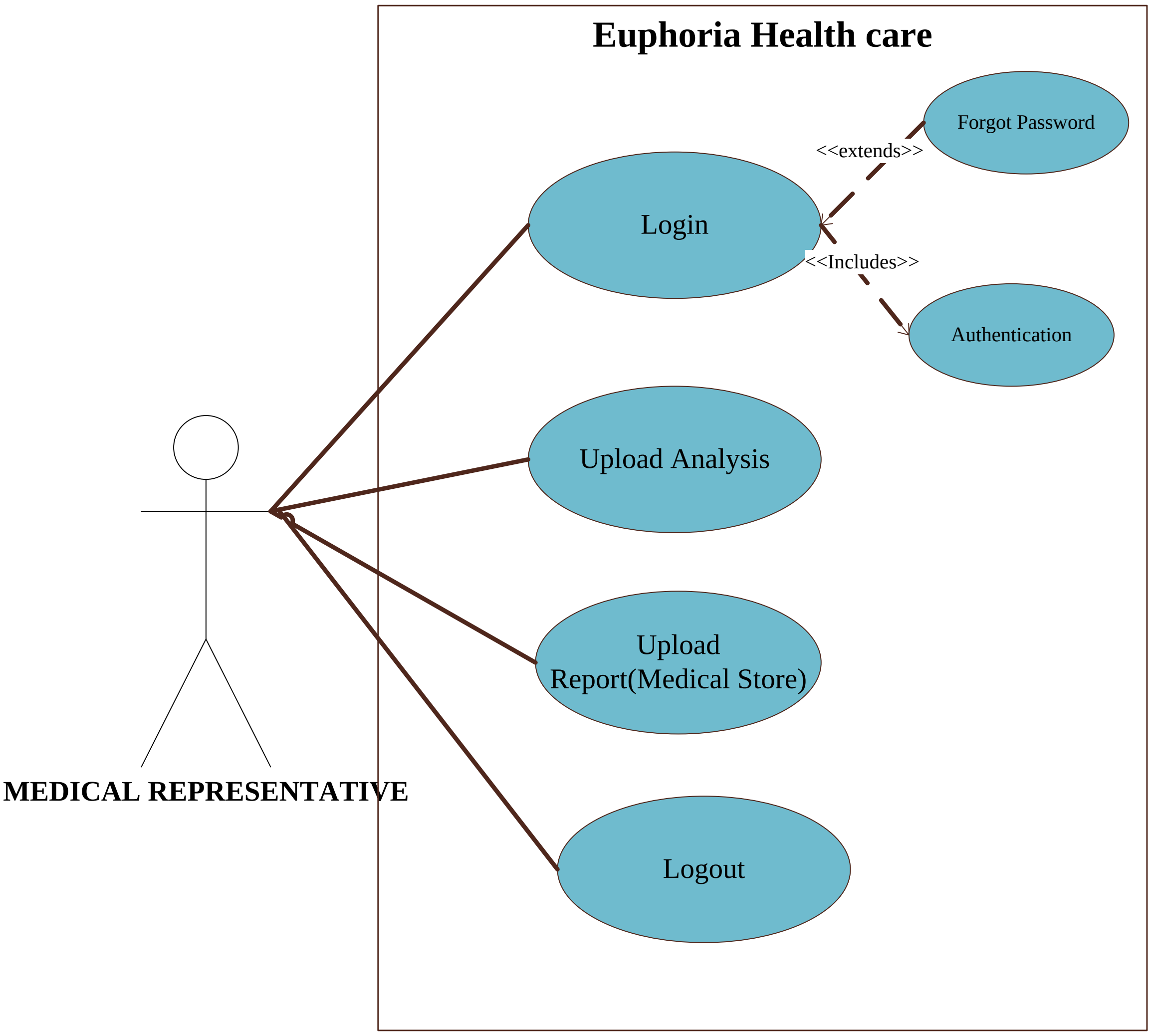

Fig 3.12 use case for medical representative

- **Use case for Patient.**

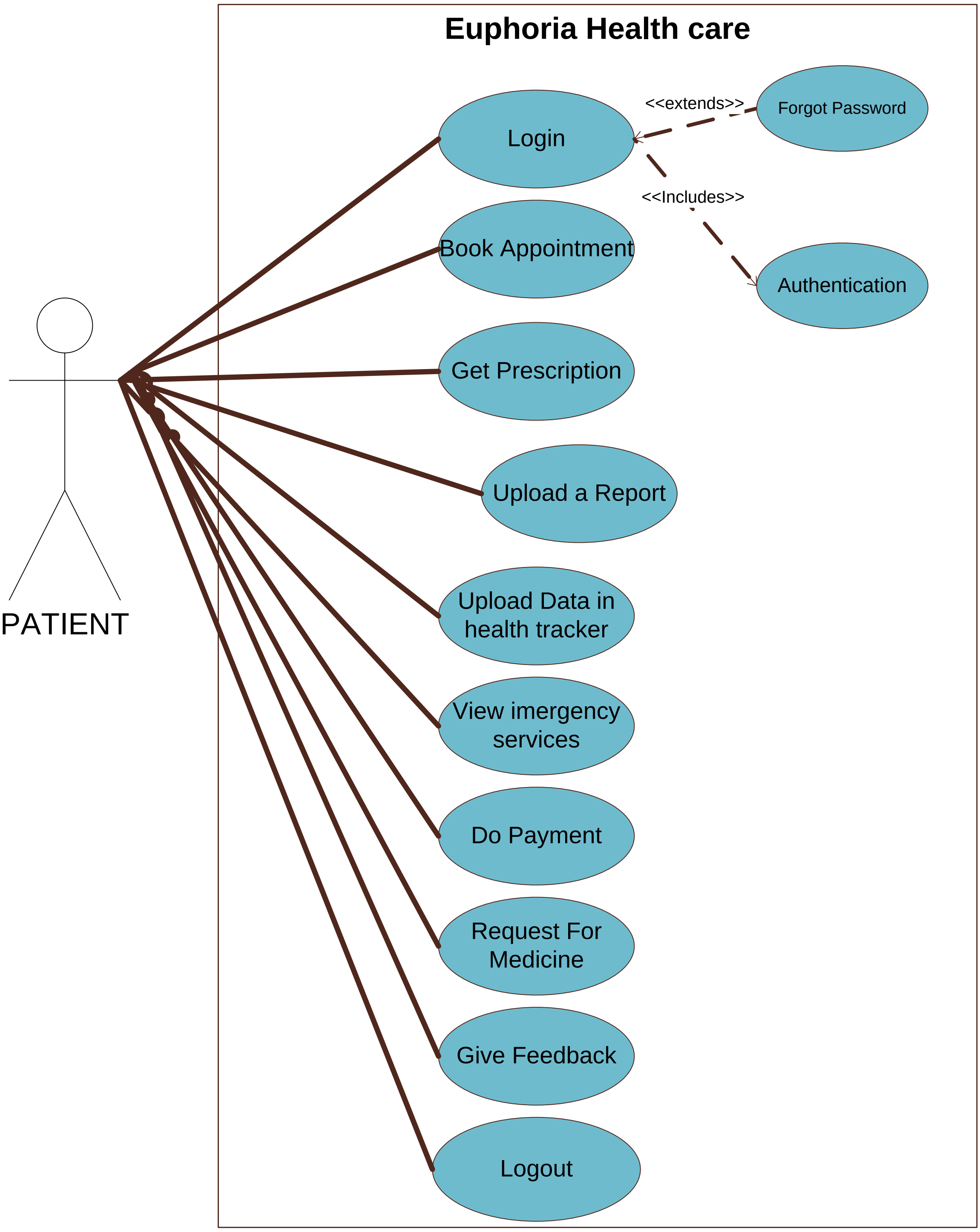

Fig 3.13 use case for patient

- **Use case for Pharmacist.**

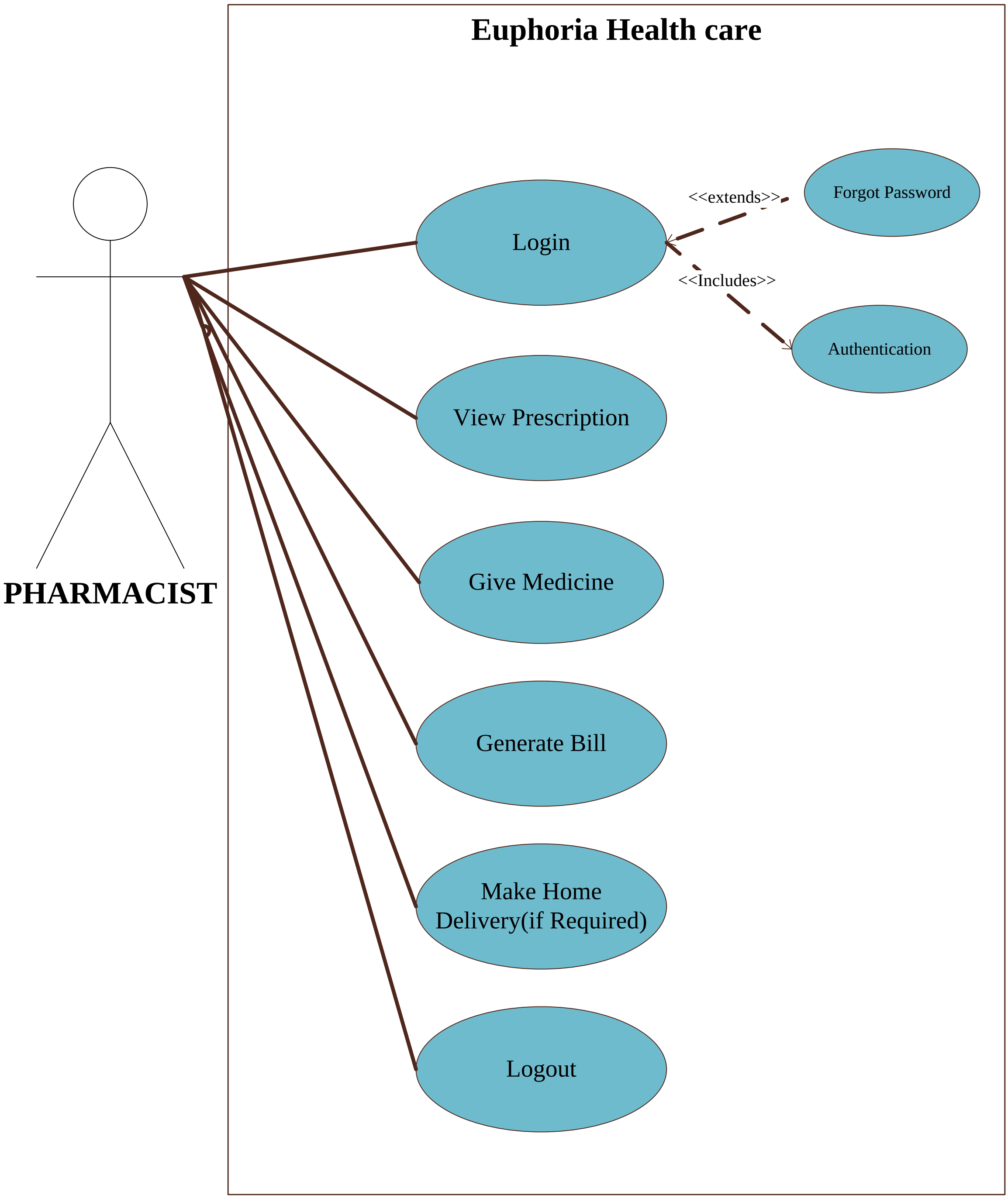

Fig 3.14 use case for pharmacist

Chapter :4

IMPLEMENTATION

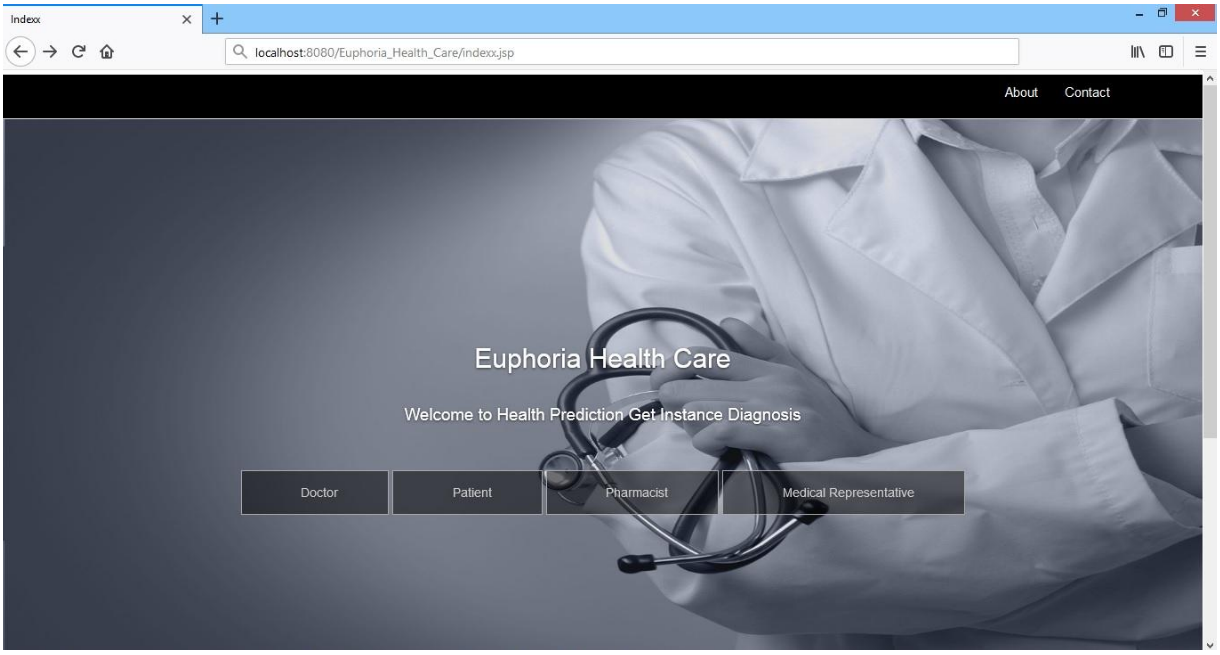
Indexx
localhost:8080/Euphoria_Health_Care/indexx.jsp
About
Contact
Euphoria Health Care
Welcome to Health Prediction Get Instance Diagnosis
Doctor
Patient
Pharmacist
Medical Representative

Fig 4.1 homepage

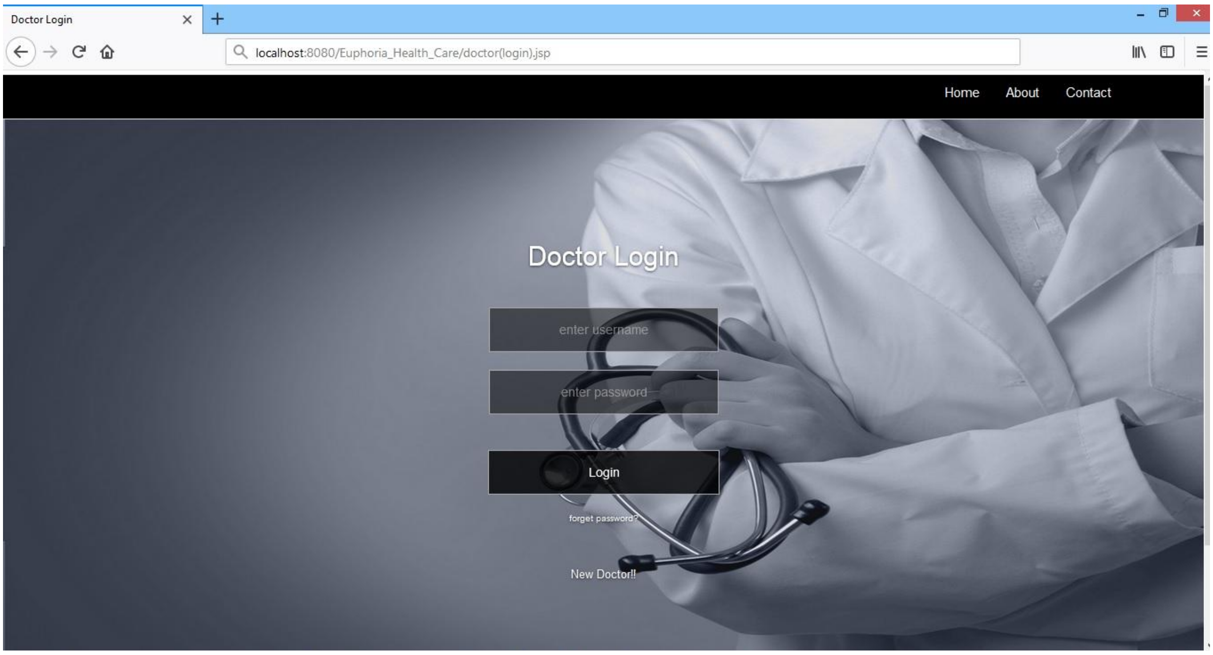

Fig 4.2 login page for doctor

Chapter : 5

CONCLUSION AND FUTURE WORK

➢ Conclusion

This is to conclude that the project assigned to us was worked upon with a sincere effort. Most of the requirements have been fulfilled up to the mark and the requirements which have been remaining, can be completed within a short extension. The application may be used by the company for the further development into a content management system.

We had a very good experience of an extremely professional environment to work under and a good project to work under the guidance of experienced and skilled employees. We were able to learn through the company, management of a project, how to analyze the given task and handle it swiftly.

Also I able to learn java Technology which would help us a lot when I put our first step in the industry. As per the further discussion team and we might be working on further application for which we may have to go under induction training.

➢ Future Enhancement

- Today everything is done online and there is need for improvement in medical field.
- Dynamic application will in future specifically for patients to find doctors and remedy.
- This application will promotes doctor to prescribe patients also this application helps medical stores to expand their business while delivering medicines to end-patients.
- Users will get the information of the doctors with user ratings and check availability.
- User can get daily tips and tricks regarding medical field.
- Mobile app will be developed in near future.

➢ References:

- List of Web References:-
  - www.javatpoint.com
  - www.tutorialspoint.com
  - www.geeksforgeeks.com
  - www.w3school.com
  - www.stackoverflow.com
  - www.eclipse.org
  - www.mysql.com
  - www.learnjavaonline.org
  - www.c4learn.com
  - www.learn-java-tutorial.com

- List of Text References.:-
  - Programming with Java by E Balagurusamy
  - Java: The Complete Reference, Seventh Edition by Herbert Schildt
  - Complete Reference Java 2 by Herbert Schildt
  - Java Programming by Sachin Malhotra, Saurabh Choudhary
  - Core Java Volume-I Fundamentals by Cay S. Horstmann Gary Cornell