**SYSTEM AS A DOCTOR-A TOPIC MODELING APPROACH FOR TRADITIONAL CHINESE MEDICINE PRESCRIPTIONS**

AN INDUSTRIAL INTERNSHIP REPORT

*Submitted in partial fulfilment for the award of the degree of*

**MTech**

***in***

**Software Engineering**

*by*

**M SAI DARSHINI (16MIS0240)**





**School of Information Technology and Engineering**

**Department of Software and Systems Engineering**

**NOVEMBER 2019**



**School of Information Technology and Engineering**

**Department of Software and Systems Engineering**

**DECLARATION BY THE CANDIDATE**

I hereby declare that the Industrial Internship report entitled

**“SYSTEM AS A DOCTOR- A TOPIC MODELING APPROACH**

**FOR TRADITIONAL CHINESE MEDICINE PRESCRIPTIONS"**

submitted by me to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **MTech (Software Engineering)**  is a record of bonafide **Industrial Internship -SWE3099** carried out by me under the guidance of **Rajan, Head Developer>**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree in this institute or any other institute or university.

**Place**: Vellore

**Date**: **Signature of the Candidate**



**School of Information Technology and Engineering**

**Department of Software and Systems Engineering**

**BONAFIDE CERTIFICATE**

This is to certify that the Industrial Internship report entitled

"**SYSTEM AS A DOCTOR-A TOPIC MODELING APPROACH FOR TRADITIONAL CHINESE MEDICINE PRESCRIPTIONS"**

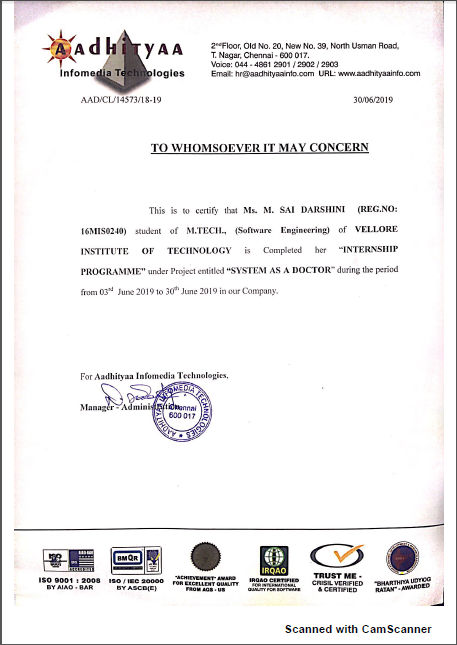
by **M SAI DARSHINI (16MIS0240)**  to VIT, Vellore, in partial fulfillment of the requirement for the award of the degree of **MTech (Software Engineering)** is a record of bonafide work carried out by him /her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

**PROF PRABHU J**

**Signature of Internal Guide**

**Examiner(s) Signature**



****

**ABSTRACT**

Our goal is to provide a tool to assist professionals and consumers in finding and choosing drugs. To achieve this goal, we develop an approach that allows a user to query for drugs that satisfy a set of conditions based on drug properties, such as drug indications, side effects, and drug interactions, and also takes into account patient profiles, We also analyze the disease and best drug advised to that specific patient through Big Data analysis we are implementing an application to know about disease based on our symptoms. Based on the symptoms system will predict the type of disease and suggest the specialist doctor based on the rank method. After that best drug will suggested by the system.

This system will facilitate the patients to interact with doctors without making any physical appointments, but the patients are restricted to only one message per day before receiving a reply. If the patient gets a response from the corresponding doctor, the patient is allowed to send another message. Similarly, this is beneficial to doctors by providing the following functionalities: patient interaction through messaging, sending prescription to pharmacies, confirming appointments, information sharing with other doctors, and patient referrals.

**Acknowledgement**

I wish to express our heartfelt gratitude to **Dr.G.Viswanathan**, Chancellor, VIT, Vellore, for providing facilities for the Industrial Internship. I am highly grateful to our Vice President, **Dr.G. Sekar Viswanathan**, Vice chancellor **Dr. Anand A. Samuel,** and Pro-Vice Chancellor **Dr.S.Narayanan,** for providing the necessary resources.

My sincere gratitude to **Dr. Balakrushna Tripathy**, Dean, School of Information Technology and Engineering, for giving me the opportunity to undertake the project.

I wish to express my sincere gratitude to **Dr. S. Sree Dharinya**, Head of the Department, Software and Systems Engineering, **Prof. P.Ushapreethi & Prof. Ramaprabha KP**, Industrial Internship Coordinators, M.Tech (Software Engineering), School of Information Technology and Engineering for providing me continuous support to do my project work**.**

I would like to express my special gratitude and thanks to my external guide **Mr Rajan, Head Developer**, **Aadhitya infomedia Technologies** and internal guide **Prof. Prabhu J, Assistant Professor, SITE** for their esteemed guidance, immense support and encouragement to complete the internship successfully.

I thank the management of VIT, Vellore for permitting me to use the library resources. I also thank all the faculty members of VIT, Vellore for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work.

Place: Vellore

Date: **M SAI DARSHINI**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | v |
|  | **LIST OF TABLES** | ix |
|  | **LIST OF FIGURES** | x |
|  | **LIST OF SYMBOLS** | xi |
|  |  |  |
| **1.** | **INTRODUCTION** | xii |
|  | 1.1.Problem Statement | xii |
|  | 1.2.Motivation | xiii |
|  | 1.3.Objective | xiii |
|  | 1.3.1.Proposed System | xiv |
|  | 1.3.2.Advantages of Proposed system | xiv |
| **2.** | **TECHNOLOGIES LEARNT** | xv-xviii |
|  |  |  |
| **3.** | **SYSTEM DESIGN** | xix |
|  | 3.1 System Architecture | xix |
|  | 3.2 Module description | xix |
|  | 3.3 System Specification |  |
|  | 3.3.1 Software Requirements | xxii |
|  | 3..3.2 Hardware Requirements | xxii |
|  | 3.4 Detailed Design |  |
|  | 3.4.1 Use case Diagram | xxiii |
|  | 3.4.2 Sequence Diagram | xxiv |
|  | 3.4.3 Class Diagram | xxv |
|  | 3.4.4 Dataflow diagram | xxvi |
|  | 3.4.5 Activity diagram | xxvii |
|  |  |  |
| **4.** | **IMPLEMENTATION** | xxviii |
|  | 4.1 Implementation details | xxviii |
|  |  |  |
| **5.** | **TEST RESULTS** | xxxix |
|  | 5.1 Test cases | xxxix |
|  |  |  |
| **6.** | **RESULTS AND DISCUSSIONS** | xli |
|  |  |  |
| **7.** | **CONCLUSION AND FUTURE WORK** | xliv |
|  | 7.1 Conclusion | xliv |
|  | 7.2 Future Work | xliv |
|  |  |  |
| **8.** | **REFERENCES** | xlv |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Title** | **Page No** |
| 5.1 | **TEST CASE FOR LOG IN FORM** | xxxix |
| 5.2 | **TEST CASE FOR REGISTRATION PAGE** | xl |
| 5.3 | **TEST CASE FOR MAIL GENERATION** | xl |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Title** | **Page No** |
| 2.1 | JAVA AND API | xvii |
| 3.1 | SYSTEM ARCHITECTURE | xix |
| 3.2.1 | SERVER DEPLOYMENT | xx |
| 3.2.2 | CONSTRUCTION OF DISEASE TRAINING SET | xx |
| 3.2.3 | BIG DATA BASED ANALYSIS | xxi |
| 3.2.4 | PRE STORED DATA COMPARISON | xxi |
| 3.2.5 | BEST DOCTOR RECOMMENDATION | xxi |
| 3.2.6 | DRUG AND TRAINING SET CONSTRUCTION | xxii |
| 3.4.1 | USE CASE DIAGRAM | xxiii |
| 3.4.2 | SEQUENCE DIAGRAM | xxiv |
| 3.4.3 | CLASS DIAGRAM | xxv |
| 3.4.4 | DATAFLOW DIAGRAM | xxvi |
| 3.4.5 | ACTIVITY DIAGRAM | xxvii |

**LIST OF Abbreviations**

|  |  |
| --- | --- |
| **ACRONYM** | **EXPANSION** |
| **TCM** | Traditional Chinese medicine |
| **SQL** | Structured Query Language |
| **URL** | Uniform Resource Locator |
| **HDFS** | Hadoop Distributed File System |
| **JVM** | Java Virtual Machine |

**CHAPTER 1**

**INTRODUCTION**

Traditional Chinese medicine (TCM) is an important example of how ancient and accumulated knowledge is applied in a holistic approach in present day health care. In TCM, a prescription is a group of herbal medicines (mineral medicines and animal medicines are also used, we will use the word “herb” to refer to medicinal materials in prescriptions), which is the main way to cure diseases for thousands of years. In the long Chinese history, a lot of prescriptions have been invented to treat diseases and more than 100,000 have been recorded . It has a source book, composition herbs, usage and indication symptoms. Regularities on the herbs composition in prescriptions and corresponding symptoms play a significant role for clinical treatment and novel prescription development.

Previous works proposed many methods that could discover regularities in prescriptions but they failed to comprehensively describe how a prescription is generated using TCM theories or utilize TCM domain knowledge well. The therapeutic process in traditional Chinese medicine can be called as li-fa-fang-yao which is of critical importance in clinical practices ,which means principles, methods, prescriptions and Chinese herbs respectively. It indicates the four basic steps of diagnosis and treatment: determining the cause, mechanism (syndromes) of the disease according to symptoms, then deciding the treatment methods based on the mechanism, and finally selecting a prescription as well as proper herbs.

**1.1.PROBLEM STATEMENT**

In the current marketplace there are several apps for communication between doctors and patients but most of their functionalities are commercial, and not really helpful to the users. There are no apps which actually meet the patient’s basic needs and no such apps that fulfill the patient’s expectations from the doctors. This project aims at changing this kind of misleading and unnecessary interaction between doctors and patients, with a new initiative which focuses more on patient benefits relating to his/her disease treatment and facilitating the doctors with an easy and optimal way to treat the patients.

In the present system, more number of people were getting serious and even died because of not aware of disease. Even more number of applications are available to know about disease but they need login and patients have to give symptoms manually. Regularities in the prescriptions are for both clinical practice and the novel prescription development. This issued a common problem in the everyday life and helps in the development of a new software.

**1.2.MOTIVATION**

We do not usually visit hospitals unless we need to but this visit does not always gives us good experience. We face various problems there. There are automated hospital management system but they do not provide any functionality for us. Even the doctors do not have access to their patient’s data while they are at home. It will be great for the patients to have an application that will keep them close to their personal data like: prescriptions, test reports and other important materials. Patients will be able to browse through all the doctors available and ask for appointment for their desired doctor. A doctor will also experience the similar benefits. This will reduce a lot of paper works and make things easy for everyone associated with it. I have some personal experience visiting the hospital and I also have gathered experience from other people by asking them and observing them for some time. People experience their worst nightmare while they need to visit the hospital. Nobody visits the hospital unless it is extremely necessary. People get mad but they do not have anything to do then. This gave me the perfect motivation to build something for them and offer them some help at their most needed time. I am confident that this product can and will help them.

**1.3.OBJECTIVE**

•The objective of the project is to identify the disease based on the patient symptoms and suggest the best drug by English and ayurvedic medicine.

•To suggest doctor based on ranking.

**1.3.1 PROPOSED SYSTEM:**

We implement a web application to know about their disease themselves using Big data. People were easily gives their symptoms by selecting symptoms form drop down box. we develop an approach that allows a user to query for drugs that satisfy a set of conditions based on drug properties, such as drug indications, side effects, and drug interactions, and also takes into account patient profiles, We also analyze the disease and best drug advised to that specific patient through Big Data analysis .we are implementing an application to know about disease based on our symptoms. Based on the symptoms system will predict the type of disease and suggest the specialist doctor based on the rank method. After that best drug will suggested by the system. Based on disease we suggest the medicine in allopathic and ayurvedhic.

**1.3.2.ADVANTAGES OF PROPOSED SYSTEM**

* Predict the disease based on the symptoms
* Best drugs are suggested
* Best doctors were suggested based on ranking system
* Both ayurvedic and English medicine are recommended
* People can easily communicate with doctors
* Both people can gain through this application
* Time will consume.

**CHAPTER 2**

**TECHNOLOGIES LEARNT**

**BIGDATA:**

Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, duration, search, sharing, storage, transfer, visualization, and privacy violations. The trend to larger data sets is due to the additional information derivable from analysis of a single large set of related data, as compared to separate smaller sets with the same total amount of data, allowing correlations to be found to "spot business trends, prevent diseases, combat crime and so on. So we can implement big data in our project because every employ has instructed information so we can make analysis on this data.

**HADOOP BACKGROUND:** Hadoop is an open-source Apache Software Foundation project written in Java that enables the distributed processing of large datasets across clusters of commodity. Hadoop has two primary components, namely, HDFS and Map Reduce programming framework. The most significant feature of Hadoop is that HDFS and Map Reduce are closely related to each other; each are co-deployed such that a single cluster is produced. Therefore, the storage system is not physically separated from the processing system.

### MAP REDUCE IN CLOUDS: Map Reduce accelerates the processing of large amounts of data in a cloud; thus, Map Reduce, is the preferred computation model of cloud providers. Map Reduce is a popular cloud computing framework that robotically performs scalable distributed applications and provides an interface that allows for parallelization and distributed computing in a cluster of servers. The approach is to apply scientific computing problems to the Map Reduce framework where scientists can efficiently utilize existing resources in the cloud to solve computationally large-scale scientific data.

**MYSQL:**

MySQL is a database management system. A database is a structured collection of data. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server.

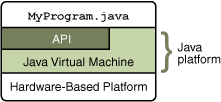
MySQL is a relational database management system. A relational database stores data in separate tables rather than putting all the data in one big storeroom. This adds speed and flexibility. The SQL part of “MySQL” stands for “Structured Query Language.” MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. MySQL Server works in client/server or embedded systems. The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). A large amount of contributed MySQL software is available. MySQL is a widely supported database, and very likely that most of your applications supports it.

**JAVA:**

The Java platform is the ideal platform for network computing. Running across all platforms -- from servers to cell phones to smart cards -- Java technology unifies business infrastructure to create a seamless, secure, networked platform for your business. The Java platform benefits from a massive community of developers and supporters that actively work on delivering Java technology-based products and services as well as evolving the platform through an open, community-based, standards organization known as the Java Community Process program.

**java virtual machine:** A **Java virtual machine** (**JVM**) is a virtual machine that can execute Java byte code. It is the code execution component of the Java software platform. A Java virtual machine is a program which executes certain other programs, namely those containing Java byte code instructions. JVMs are most often implemented to run on an existing operating system, but can also be implemented to run directly on hardware. A JVM provides an environment in which Java byte code can be executed, enabling such features as automated exception handling, which provides root*-*cause debugging information for every software error (exception), independent of the source code.

**java API:** The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages.



**2.1 JAVA AND API**

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

**JAVA WEB APPLICATIONS:** The main function of the Tomcat server is to act as a container for Java web applications. Therefore, before we can begin our Tomcat-specific discussion, a brief introduction as to exactly what web applications are is in order. The concept of a web application was introduced with the release of the Java servlets specification.

* Java Server Pages (JSPs)
* Utility classes
* Static documents, including HTML, images, JavaScript libraries, cascading style sheets, and so on
* Client-side classes
* Meta-information describing the web application

One of the main characteristics of a web application is its relationship to the Servlets Context. Each web application has one and only one Servlets Context. This relationship is controlled by the servlets container and guarantees that no two web applications will clash when accessing objects in the Servlets Context.

**HTML :**

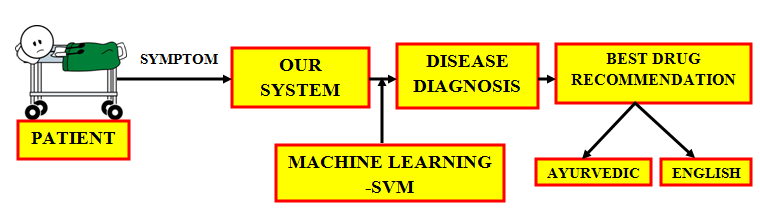
HTML or Hyper Text Markup Language is the standard markup language used to create Web pages. HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like).

The HTML and CSS tools allow for the inspection and editing of HTML and CSS elements on a web page. Later versions of Firebug allow users to see live changes to the CSS. Visualization of CSS elements is shown while inspecting HTML elements.

**CHAPTER 3**

**SYSTEM DESIGN**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. The process or art of defining the hardware and software architecture, components, modules, interfaces, and data for a computer system to satisfy specified requirements.

**3.1 SYSTEM ARCHITECTURE**

**3.2 MODULE DESCRIPTION**

**MODULES:**

* A modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different part of system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity that is software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements.
* Modularity is the single attribute of software that allows a program to be intellectually manageable. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are: Modular decomposability, Modular Comps ability, Modular Understandability, Modular continuity, Modular Protection.
* The following are the modules of the project, which is planned in aid to complete the project with respect to the proposed system, while overcoming existing system and also providing the support for the future enhancement.

1. **SERVER DEPLOYMENT**
2. **CONSTRUCTION OF DISEASE TRAINING SET**
3. **BIG DATA BASED ANALYSIS**
4. **PRE STORED DATA COMPARISON**
5. **PREDICTIVE DISEASE ANALYSIS**
6. **DRUG AND TRAINING SET CONSTRUCTION**

**3.2.1 SERVER DEPLOYMENT**

Data Service Provider will contain the large amount of data in their Data Storage. Also the. The drugs and disease information will be stored in the Database . Also the Data Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job.

Admin

Data Base

**3.2.2 CONSTRUCTION OF DISEASE TRAINING SET**

In this module we can design and implementation of train the disease to system. Server will store a set of trained dataset and its relevant diagnosis pattern. Multiple number diseases and symptoms will be collected and stored in a training set. Different type diseases and symptoms will be stored in a dataset.

Admin

Dataset

Training set

**BIG DATA BASED**

**ANALYSIS**

In this module we implement big data, in this big data we will have lot or vast amount of data that may wanted or unwanted information in simple the information in the big data are unstructured. So in this module the patient is going allow permission to access the server by the big data analyst .The big data analyst get the all the disease and drugs information which mention above and extract the information by the technique of map reducing formation to get useful information which is useful for patient.

Big data

Result on DB

**3.2.3 PRE STORED DATA COMPARISON**

In this module doctor will import all the details about the medicine i.e. what are the symptoms, dosages and drug .And hw will store more about of the medicine so that we can make some use of it for example we can give awareness to the society .we store the all the data in the clustering format so that data can spitted and stored in the different clusters. So that it will easily to classify the data for the research.

**3.2.4 PREDICTIVE DISEASE ANALYSIS**

In this module we implement predictive disease analysis system in which the data will be analysis so that we can predictive the disease based on the symptoms .This module interact with server to analysis, the analysiation is done by the researchers. So they get the data from the server to make analysis to find the disease based on the symptoms

**3.2.5 BEST DOCTOR RECOMMENDATION**

In this module we suggest the best doctor based on ranking. These ranking system is based on patient review. Patients will give review for the treatment and based on review doctors will come upon top most place.

**3.2.6 DRUG AND TRAINING SET CONSTRUCTION**

In this module we will train the drugs for every disease and also train the side effects of the drugs. User will be giving their Symptoms & Diagnostic reports to the system for the diagnosis of the disease. If any side effects came by using the prescribed drug user can give the side effects on the same website and those side effects will be match the drugs and suggest the alternate drug for that disease. In this we will suggest the medicine Based upon patients willing like whether they need ayurvedic or English medicine.

**3.3 SYSTEM SPECIFICATION**

**3.3.1 SOFTWARE REQUIREMENTS**

The software requirements are the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team’s and tracking the team’s progress throughout the development activity.

* Operating system : Windows 7/ 8
* Languages : Java JDK 8.1
* Data Base : Mysql
* IDE : Net Beans 8.2

**3.3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the systems do and not how it should be implemented.

* Hard disk : 120 GB
* Monitor : 15’ color with vgi card support
* Ram : Minimum 256 MB
* Processor : Pentium iv and above (or) equivalent
* Processor speed : Minimum 500 MHZ

**3.4 DETAILED DESIGN**

**3.4.1 USE CASE DIAGRAM**

Use case diagrams overview the usage requirement for system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe “the meant” of the actual requirements. A use case describes a sequence of action that provides something of measurable value to an action and is drawn as a horizontal ellipse.



**3.4.2 SEQUENCE DIAGRAM**

Sequence diagram model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and commonly used for both analysis and design purpose. Sequence diagram are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system.



**3.4.3 CLASS DIAGRAM**

****

**3.4.4 DATAFLOW DIAGRAM**

**LEVEL 0**

**User**

**System**

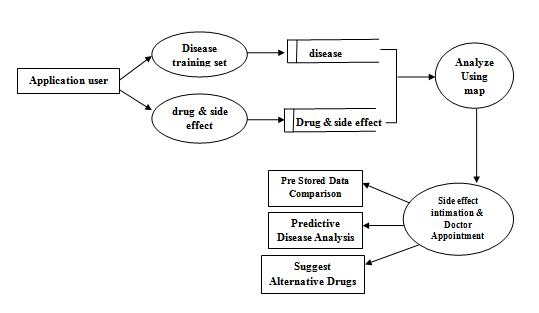
Give symptoms

Training set

Check with dataset

Get best drug

**LEVEL 1**

****

**3.4.5 ACTIVITY DIAGRAM**

Activity diagram are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. The activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. Activity diagram consist of Initial node, activity final node and activities in between.



**CHAPTER 4**

**IMPLEMENTATION**

**4.1 IMPLEMENTATION DETAILS**

**ADMIN :**

* Login
* Provide Password
* Login into the System

**DISEASE :**

* Add Disease Name
* Add Disease Description

**DISEASE SYMPTOMS:**

* Selection of the Disease
* Disease Symptoms

**REGISTRATION:**

* Username
* Password
* E-mail Address
* Selection of Disease
* Contact Number
* Gender and DOB

**APPOINTMENT: PATIENT INFO GATHERING:**

* Token Id and Date of Registration
* Username ,E-mail and DOB
* Selection of Symptoms

**PROCEDURES:**

1.Create login pages for the admin and enter the login details and enter the system with the valid credentials as admin.

2. The user logins into the system and provides the symptoms of the disease with its description.

3. The user then selects the doctor and registers the appointment with the doctor in-charge for the selected disease for the given symptoms.

4. The database provided uses clustering algorithms and ranks the Doctors according to the symptoms provided by the User and selects the suitable Doctor in-charge.

5.The doctor-in charge thus registers under the particular patient at a particular time provided with a token id .

6. The patient thus can anytime attend the Doctor registered at the desired time.

**SAMPLE CODES:**

**USER REGISTRATION:**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.nura.servlet;

import com.nura.dao.impl.UserDetailsDAOImpl;

import com.nura.entity.UserDetails;

import java.io.IOException;

import java.io.InputStream;

import java.io.PrintWriter;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.annotation.MultipartConfig;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.Part;

/\*\*

\*

\* @author ArunRamya

\*/

@MultipartConfig

@WebServlet(name = "UserRegistration", urlPatterns = {"/UserRegistration"})

public class UserRegistration extends HttpServlet {

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

\* methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

UserDetails \_userDtls = new UserDetails();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");

Date date = null;

try {

date = sdf.parse(request.getParameter("dob"));

} catch (ParseException ex) {

Logger.getLogger(UserRegistration.class.getName()).log(Level.SEVERE, null, ex);

}

\_userDtls.setDateOfBirth(date);

\_userDtls.setGender(request.getParameter("gender"));

// \_userDtls.setLivesIn(request.getParameter("lives\_in"));

\_userDtls.setMobileNumber(request.getParameter("mobile\_no"));

\_userDtls.setPassword(request.getParameter("password"));

// \_userDtls.setStudiedAt(request.getParameter("studied\_at"));

\_userDtls.setUserName(request.getParameter("username"));

\_userDtls.setSpecialist(request.getParameter("disease\_name"));

\_userDtls.setFirstName(request.getParameter("usr\_name"));

// \_userDtls.setWorkedAt(request.getParameter("worked\_at"));

//String[] userLikes = request.getParameterValues("user\_likes");

try {

for (Part parts : request.getParts()) {

System.out.println("Parts");

InputStream inputStream = parts.getInputStream();

byte[] getImgytes = new byte[inputStream.available()];

inputStream.read(getImgytes);

String fileName = getFileName(parts);

System.out.println("File name:-" + fileName);

if (fileName != null) {

\_userDtls.setUserImage(getImgytes);

}

}

} catch (Exception ex) {

System.out.println(ex.getLocalizedMessage());

}

if (new UserDetailsDAOImpl().saveUserDtls(\_userDtls)) {

response.sendRedirect("loginPage.jsp?msg=User Details saved");

} else {

response.sendRedirect("loginPage.jsp?msg=Alert!Unable to save.Contact Admin");

}

}

}

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

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

// Extract file name from content-disposition header of file part

private String getFileName(Part part) {

final String partHeader = part.getHeader("content-disposition");

System.out.println("\*\*\*\*\* partHeader: " + partHeader);

for (String content : part.getHeader("content-disposition").split(";")) {

if (content.trim().startsWith("filename")) {

return content.substring(content.indexOf('=') + 1).trim()

.replace("\"", "");

}

}

return null;

}

}

**PATIENT APPOINMENT:**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package com.nura.servlet;

import com.nura.dao.impl.UserDetailsDAOImpl;

import com.nura.entity.UserDetails;

import java.io.IOException;

import java.io.InputStream;

import java.io.PrintWriter;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.annotation.MultipartConfig;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.Part;

/\*\*

\*

\* @author ArunRamya

\*/

@MultipartConfig

@WebServlet(name = "UserRegistration", urlPatterns = {"/UserRegistration"})

public class UserRegistration extends HttpServlet {

/\*\*

\* Processes requests for both HTTP <code>GET</code> and <code>POST</code>

\* methods.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

UserDetails \_userDtls = new UserDetails();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-mm-dd");

Date date = null;

try {

date = sdf.parse(request.getParameter("dob"));

} catch (ParseException ex) {

Logger.getLogger(UserRegistration.class.getName()).log(Level.SEVERE, null, ex);

}

\_userDtls.setDateOfBirth(date);

\_userDtls.setGender(request.getParameter("gender"));

// \_userDtls.setLivesIn(request.getParameter("lives\_in"));

\_userDtls.setMobileNumber(request.getParameter("mobile\_no"));

\_userDtls.setPassword(request.getParameter("password"));

// \_userDtls.setStudiedAt(request.getParameter("studied\_at"));

\_userDtls.setUserName(request.getParameter("username"));

\_userDtls.setSpecialist(request.getParameter("disease\_name"));

\_userDtls.setFirstName(request.getParameter("usr\_name"));

// \_userDtls.setWorkedAt(request.getParameter("worked\_at"));

//String[] userLikes = request.getParameterValues("user\_likes");

try {

for (Part parts : request.getParts()) {

System.out.println("Parts");

InputStream inputStream = parts.getInputStream();

byte[] getImgytes = new byte[inputStream.available()];

inputStream.read(getImgytes);

String fileName = getFileName(parts);

System.out.println("File name:-" + fileName);

if (fileName != null) {

\_userDtls.setUserImage(getImgytes);

}

}

} catch (Exception ex) {

System.out.println(ex.getLocalizedMessage());

}

if (new UserDetailsDAOImpl().saveUserDtls(\_userDtls)) {

response.sendRedirect("loginPage.jsp?msg=User Details saved");

} else {

response.sendRedirect("loginPage.jsp?msg=Alert!Unable to save.Contact Admin");

}

}

}

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

/\*\*

\* Handles the HTTP <code>GET</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Handles the HTTP <code>POST</code> method.

\*

\* @param request servlet request

\* @param response servlet response

\* @throws ServletException if a servlet-specific error occurs

\* @throws IOException if an I/O error occurs

\*/

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

/\*\*

\* Returns a short description of the servlet.

\*

\* @return a String containing servlet description

\*/

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

// Extract file name from content-disposition header of file part

private String getFileName(Part part) {

final String partHeader = part.getHeader("content-disposition");

System.out.println("\*\*\*\*\* partHeader: " + partHeader);

for (String content : part.getHeader("content-disposition").split(";")) {

if (content.trim().startsWith("filename")) {

return content.substring(content.indexOf('=') + 1).trim()

.replace("\"", "");

}

}

return null;

} }

**CHAPTER 5**

**TEST RESULTS**

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

**Objectives of software testing -:**

– Ensures the quality of product

– Defect prevention and detection

– Verify and validate the user requirement

– Focus on accurate and reliable result

– Discuss on generating test cases test cases

– Provide information to take decision for next phase

– Gain confidence of work and Evaluates the capabilities and system performance.

**5.1 TEST CASES**

**TEST CASE FOR LOG IN FORM:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Case** | **Input Data** | **Expected Output** | **Actual Output** | **Status** |
| **1** | Using valid credentials | Login  Dashboard | Login  Dashboard | Pass |
| **2** | Using invalid credentials | No proper Authentication  Username or password invalid | No authentication username or password Fail | Pass |

**TEST CASE FOR REGISTRATION PAGE**

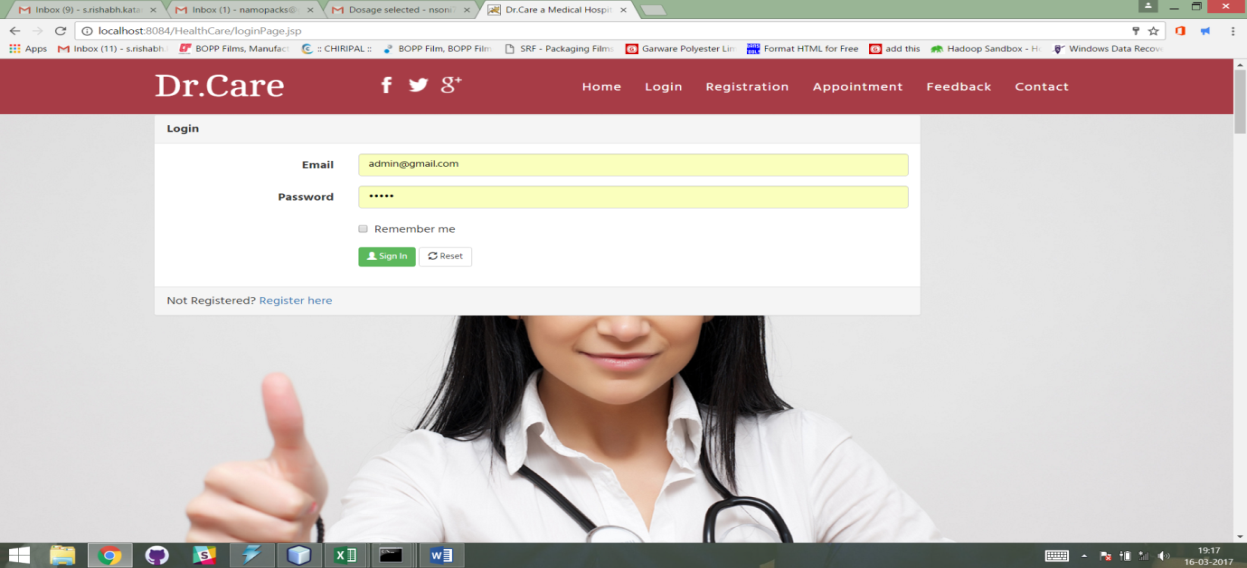
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Case** | **Input Data** | **Expected Output** | **Actual Output** | **Status** |
| **1** | Enter valid data | Success page and receive a mail | Success page and receive a mail | Pass |
| **2** | Enter invalid data | Error Message  Please enter valid details | Error Message  Please enter valid details | Pass |

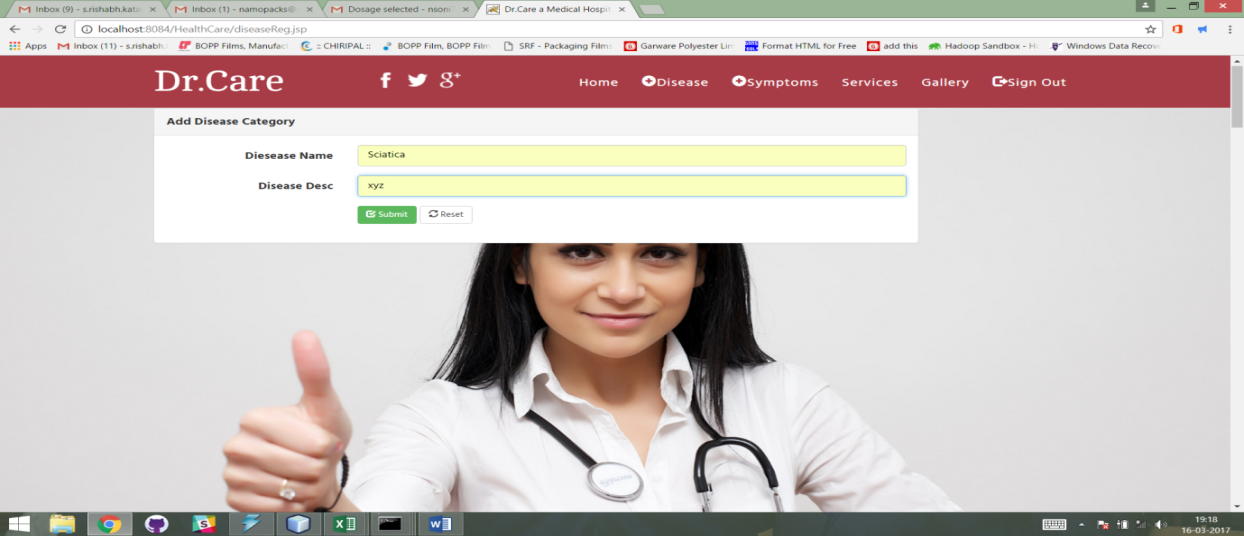
**TEST CASE FOR MAIL GENERATION:**

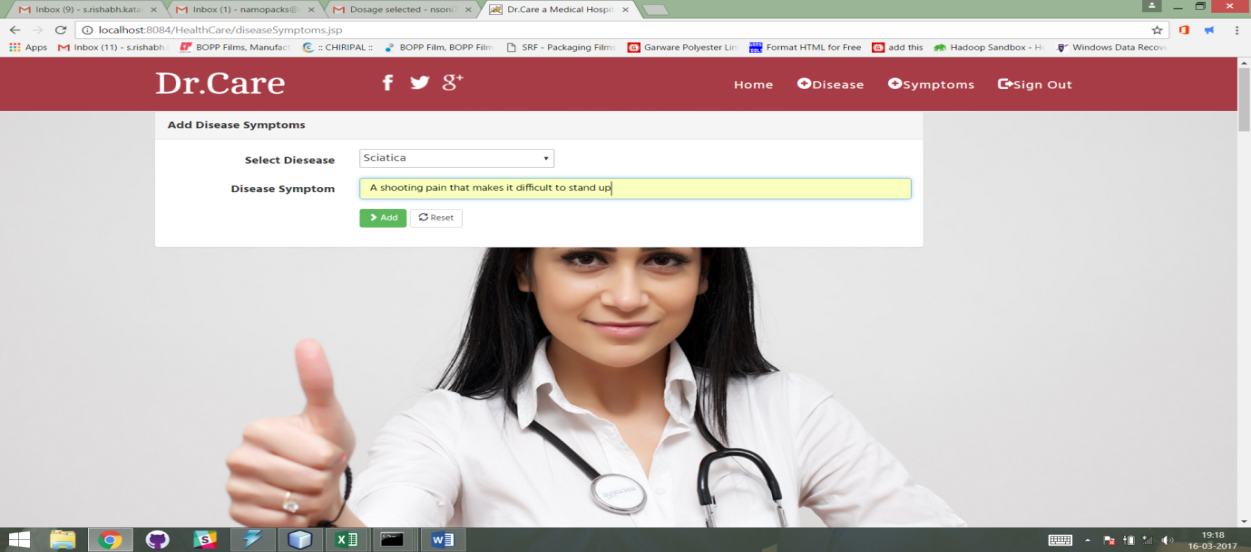
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Case** | **Input Data** | **Expected Output** | **Actual Output** | **Status** |
| **1.** | Enter valid mail and password | Success and receive a mail | Success and receive a mail | Pass |
| **2.** | Enter invalid mail and password | Invalid mail. Enter valid details | Invalid mail. Enter valid details | Pass |

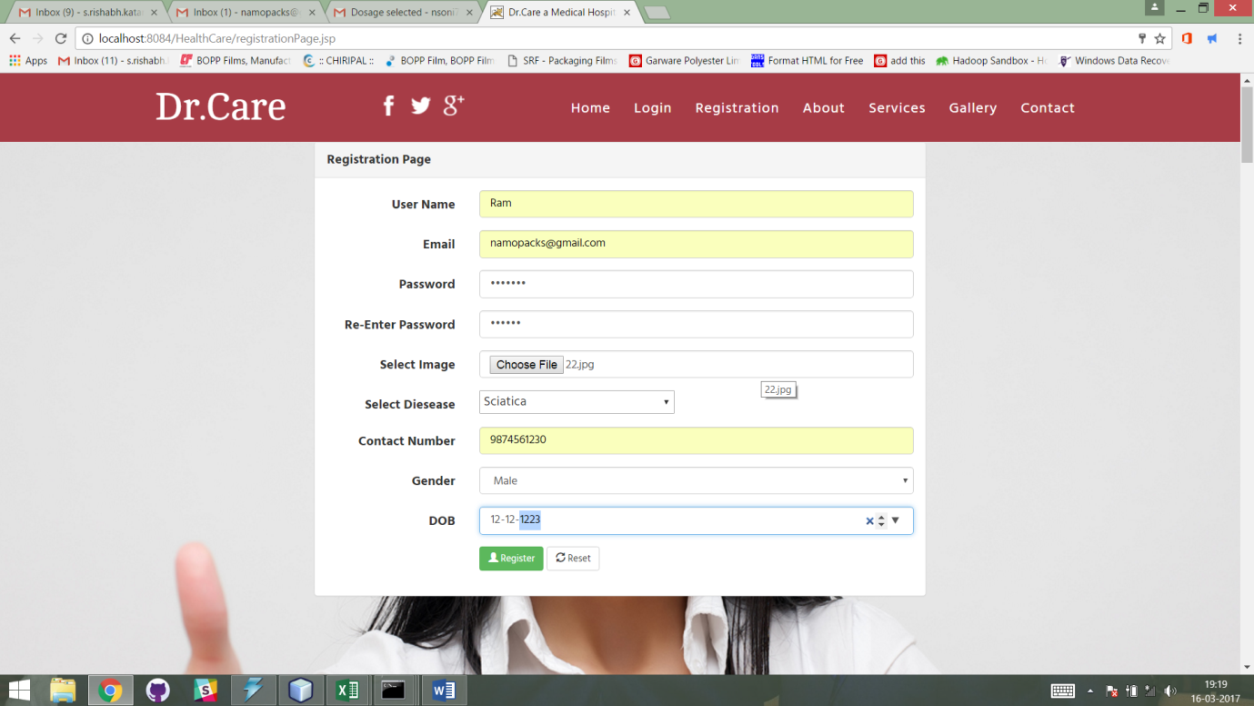
**CHAPTER 6**

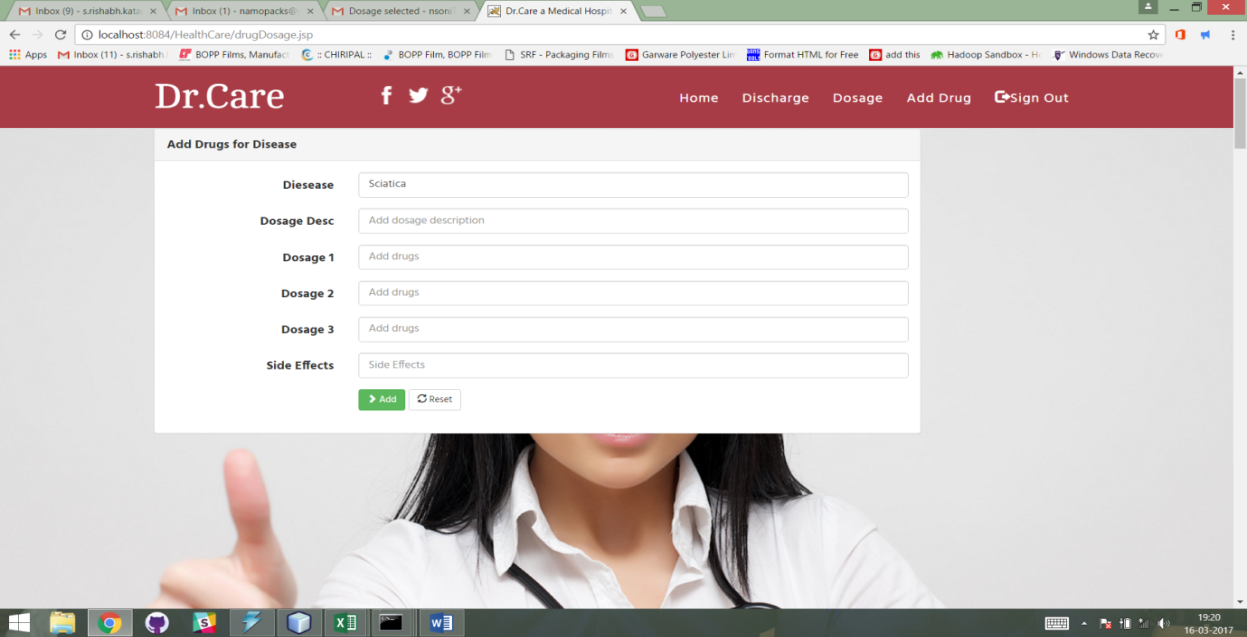
**RESULTS AND DISCUSSIONS**

****

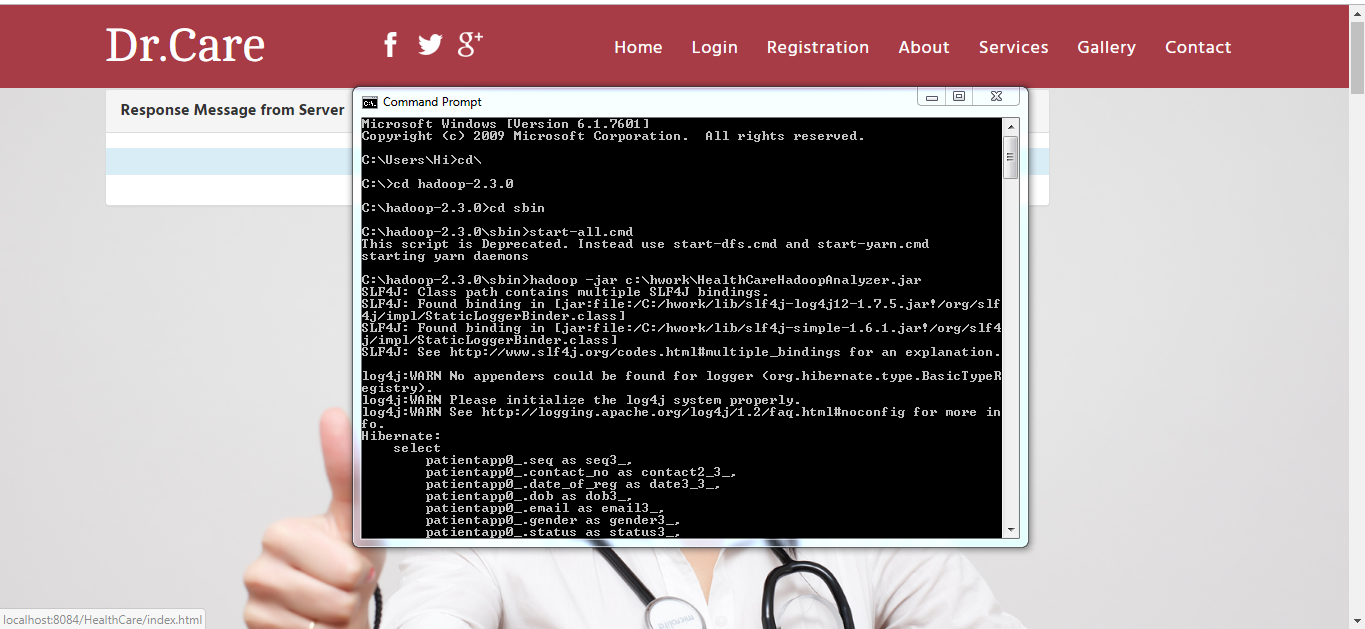
****

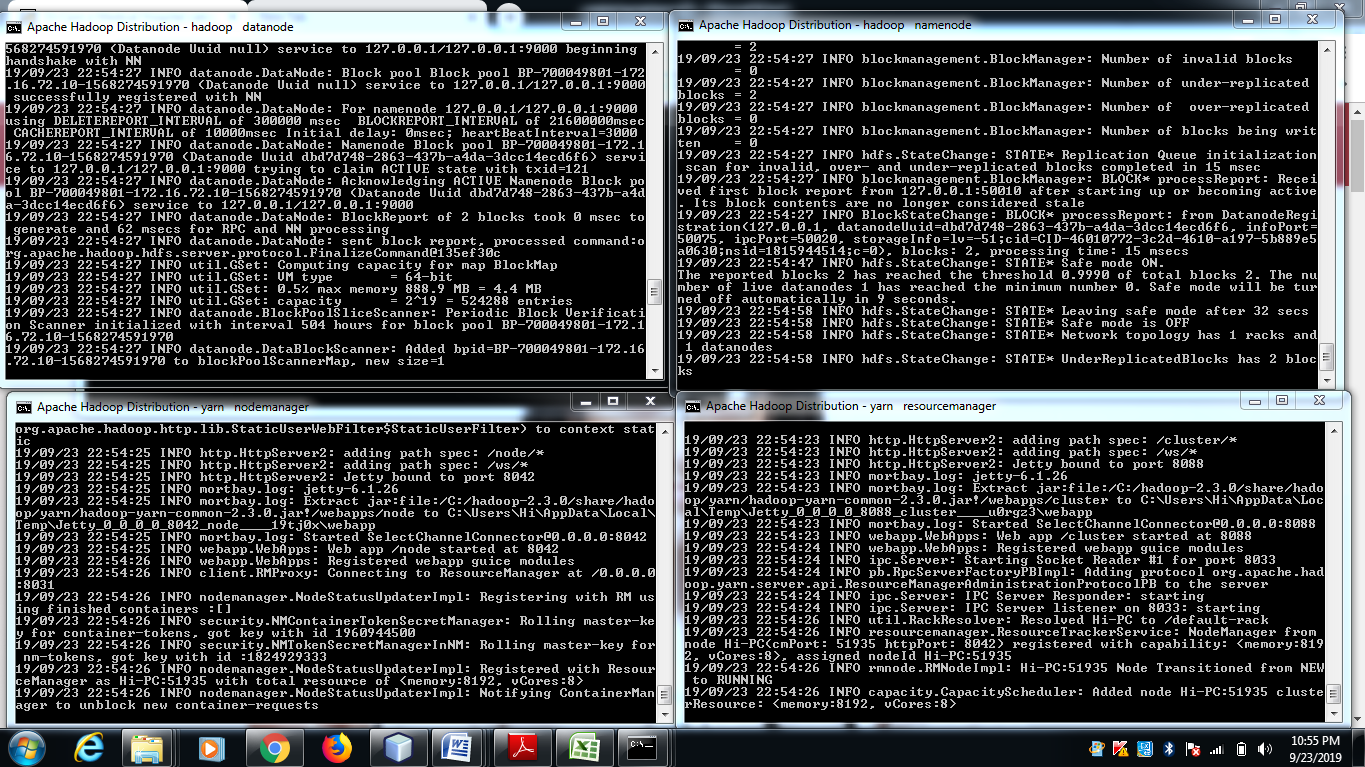
****

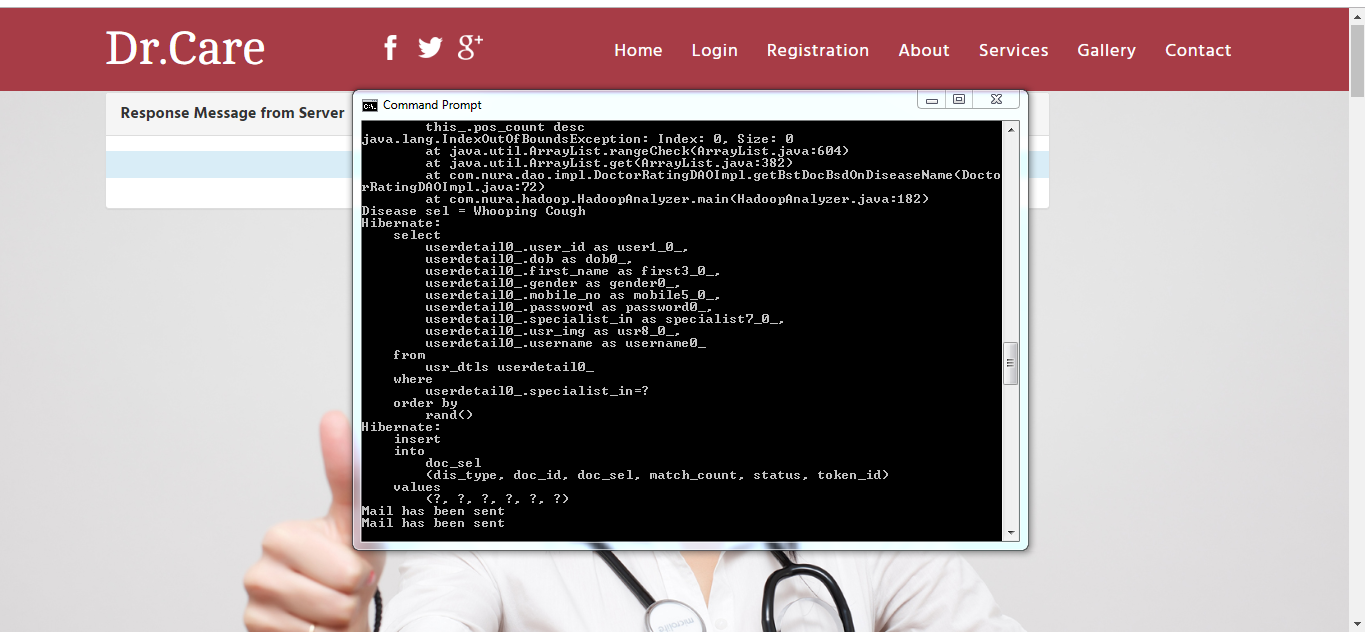
****

****

****

****

****

****

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

**7.1 CONCLUSION**

In this paper, we propose an approach for answering drug queries to support drug prescription. Our focus is on how to obtain and rank answers based on incomplete information and provide personalization. To cope with incomplete and noisy data, we allow both exact and close matches when answering queries. We also present an intuitive approach to display answers to users, which aims to help users to understand the ranked results and possibly refine their queries.

**7.2 FUTURE WORK**

It is very important that the big data research community does not repeat the same mistake. While there is clearly an important research space examining the fundamental methods and technologies for big data analytics, it is vital to acknowledge that it is also necessary to fund domain-targeted research that allows specialised solutions to be developed for specific applications. Healthcare in general and computational biomedicine in particular, seems a natural candidate for this.

**CHAPTER 8**

**REFERENCES**

[1] S. Khemmarat and L. Gao, “Supporting drug prescription via predictive and personalized query system,” in PervasiveHealth. IEEE, 2015.

[2] C. Knox et al., “Drugbank 3.0: a comprehensive resource for omics research on drugs,” Nucleic acids research, vol. 39, no. suppl 1, pp. D1035–D1041, 2011.

[3] M. Kuhn et al., “A side effect resource to capture phenotypic effects of drugs,” Molecular systems biology, vol. 6, no. 1, p. 343, 2010.

[4] M. Kanehisa and S. Goto, “Kegg: kyoto encyclopedia of genes and genomes,” Nucleic acids research, vol. 28, no. 1, pp. 27–30, 2000.

[5] T. Fawcett, “An introduction to roc analysis,” Pattern recognition letters,vol. 27, no. 8, pp. 861–874, 2006.

[6] K. Sangkuhl et al., “Pharmgkb: understanding the effects of individual genetic variants,” Drug Metab. Rev., vol. 40, no. 4, pp. 539–551, 2008.

[7] A. Langer et al., “A text based drug query system for mobile phones,” Int. J. Mob. Commun., vol. 12, no. 4, pp. 411–429, Jul. 2014.

[8] C. Doulaverakis et al., “Panacea, a semantic-enabled drug recommendations discovery framework,” J. Biomed. Semant., vol. 5, p. 13, 2014.

[9] M. Dumontier and N. Villanueva-Rosales, “Towards pharmacogenomics knowledge discovery with the semantic web,” Briefings in bioinformatics, vol. 10, no. 2, pp. 153–163, 2009.

[10] A. Ben Abacha and P. Zweigenbaum, “Medical question answering: translating medical questions into sparql queries,” in Proceedings of the 2nd ACM SIGHIT. ACM, 2012, pp. 41–50.

[11] J. Jin et al., “Querying web-scale information networks through bounding matching scores,” in WWW 2015, 2015, pp. 527–537.

[12] J. Jin et al., “A distributed approach for top-k star queries on massive information networks,” in ICPADS 2014, 2014, pp. 9–16.

[13] A. Khan et al., “Neighborhood based fast graph search in large networks,” in SIGMOD 2011. ACM, 2011, pp. 901–912.

[14] A. Khan et al., “Nema: Fast graph search with label similarity,” in Proc. of the VLDB Endowment, vol. 6, no. 3, 2013, pp. 181–192.