**KAMALA NEHRU INSTITUTE OF TECHNOLOGY**

SULTANPUR-228118

PROJECT REPORT ON

**“Aayush (Online Doctor Management System)”**

****

**Dept. Of COMPUTER SCIENCE & ENGG.**

Prepared under the encouraging guidance of

**Mr. Sarvesh Kumar Soni**

**Guided By: Submitted By: Mr. Avadhesh kumar**

**Harshit Kumar Gupta (0910410025)**

**Gaurav Dubey (0910410023)**

**KaranDeep Singh (0910410027)**

**CERTIFICATE**



**To Whomsoever It May Concern**

This is to certify that **Harshit Kumar Gupta ,Gaurav Dubey** and **KaranDeep Singh** students of Computer Science And Engg, B.tech Final Year have carried out the project titled as **“Aayush(Online Doctor Management System)”** under my supervision. This project has been completed under the ordinance governing the course leading to the bachelor degree in “Computer Science And Engg” at Kamla Nehru Institute of Technology, affiliated to Gautam Buddha Technical University.

Their work is original and genuine. The report embodies the work carried out by them in the specified period.

**Mr. Sarvesh Soni Dr. R.K.Singh Mr. Awadesh Kumar**

(Project Guide) (Head of Department) (Project Incharge)

**ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the expression of gratitude to the people who made it possible. So, We take the opportunity to acknowledge all those graceful people who always stood people before us providing the priceless guidance and encouragement to understand the existing project schemas and helped us enthusiastically to head on towards the new development methodologies and research in the rapid growing field of Web Technologies.

We are grateful to our project guide Mr. Sarvesh Soni Sir for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project.

We also thank our colleagues who have helped in successful completion of the project.

Harshit Kumar Gupta (0910410025)

Gaurav Dubey (0910410023)

KaranDeep Singh (0910410027)

**Table of Contents**

|  |  |
| --- | --- |
| 1. **Introduction**    1. Introduction to the Project.    2. Purpose of the Project.    3. Problems in existing System    4. Solutions to these problems | 5  6  6 |
| 1. **System Analysis**   **2.1** Introduction  **2.2** Analysis Model  **2.3**  Study of the system  **2.4** Requirements Analysis  **2.5** Tools and Technologies used  **2.6** Input and Output | 7  7  8  10  10  10 |
| 1. **Feasibility Report** | 11 |
| 1. **Software Requirements Specifications**   **4.1** Functional Requirements  **4.2** Performance Requirements | 13  15 |
| 1. **Technologies** | 17 |
| 1. **Diagrams**   **6.1** User Case Diagram  **6.2** ER Diagram  **6.2.a** Entities  **6.2.b** Working Entities  **6.3** Class Diagram  **6.5** Database Connectivity | 19  20  20  21  22  23 |
| 1. **Coding** | 24 |
| 1. **Testing** | 41 |
| 1. **Snapshots** | 43 |

**1.Introduction:-**

**1.1 Purpose :**

IT enabled health world aims at the computerization of interaction process of doctor and users, to divert the attention from the trivial details of a medical field to better aspects of interaction, leading to a better healthy world. This project is aimed at developing an online Discussion Forum For Doctors and Users for the World . This project is an Internet based application that can be accessed throughout the world. This system can be used to search for Doctors with speciality and experianceor countrywide. This is one integrated system that contains both the user component and the doctor component. There are features like forum, searching doctor, downloading profile as pdf etc in this system. System maintains two level of users.

**1.2 Scope :**

* Secure online registration and authentication of all users and Doctor.
* Profile management facilities with customizable skins.
* Profile download feature for doctors and users.
* Central repository in the form of a database, which is only accessible by the Data Manager.
* FAQ section for helping the users.
* One can download one’s profile as pdf at one’s homepage.
* A Person should be able to login to the system through the first page of the application and change the password after login into the system.
* See the message and profiles of the Doctors .
* Search for a particular Doctor based on the name of the Doctor or country or experience and specialty.
* User can add doctors in their follow list and the number of follow doctors are shown on their profiles.
* There is a validation checking on login and registration page.
* Doctors can see the following users in their following list. The number of followers is shown on Doctors profiles.
* User can give feedback to a particular Doctor.

**Goals of proposed system**

1. **Planned approach towards working: -** The working in the organization will be well planned and organized. The data will be stored properly in data stores, which will help in retrieval of information as well as its storage.

2. **Accuracy: -** The level of accuracy in the proposed system will be higher. All operation would be done correctly and it ensures that whatever information is coming from the center is accurate.

3. **Reliability:** - The reliability of the proposed system will be high due to the above stated reasons. The reason for the increased reliability of the system is that now there would be proper storage of information.

4. **No Redundancy: -** In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

5. **Immediate retrieval of information: -** The main objective of proposed system is to provide for a quick and efficient retrieval of information. Any type of information would be available whenever the user requires.

6. **Immediate storage of information: -** In manual system there are many problems to store the largest amount of information.

7. **Easy to Operate: -** The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

**Definitions of problems**

Problems with conventional system

1. **Lack of immediate retrievals: -**The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient’s history, the user has to go through various registers. This results in inconvenience and wastage of time.

2. **Lack of immediate information storage: -** The information generated by various transactions takes time and efforts to be stored at right place.

3. **Lack of prompt updating: -** Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

4. **Error prone manual calculation: -** Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient’s bill based on various treatments.

5. **Preparation of accurate and prompt reports: -** This becomes a difficult

task as information is difficult to collect from various registers.

2-System analysis

**2.1 INTRODUCTION**

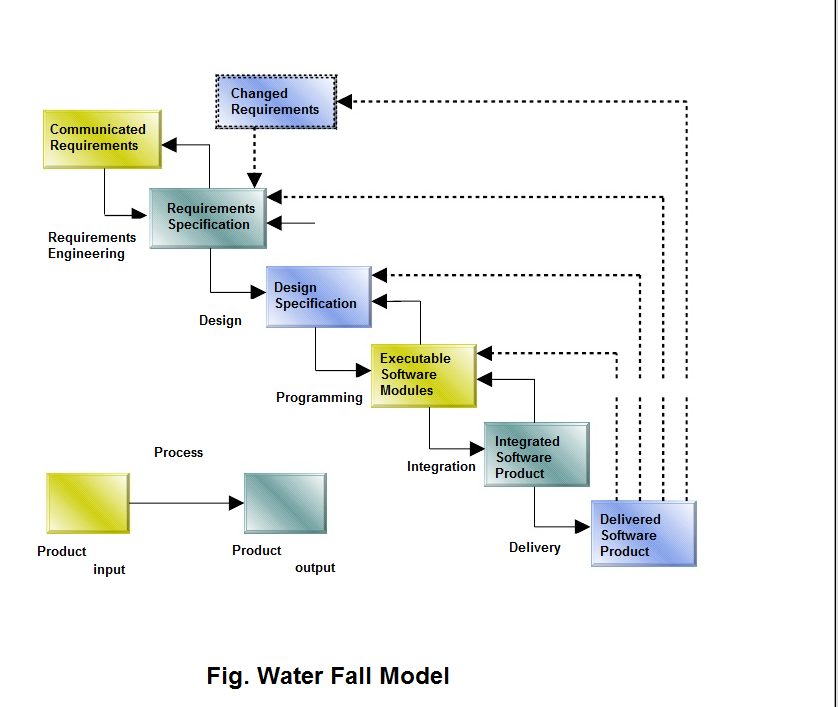
After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

**2.2 ANALYSIS MODEL**

The model that is basically being followed is the **WATER FALL MODEL**, which states that the phases are organized in a linear order. First of all, the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model. The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

* Requirement Analysis
* Project Planning
* System Design
* Detail Design
* Coding
* Unit Testing
* System Integration and Testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done. WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual system.



3-Feasibility Report

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time.

There are aspects in the feasibility study portion of the preliminary investigation:

• Technical Feasibility

• Operation Feasibility

• Economical Feasibility

**3.1 TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

• Does the necessary technology exist to do what is suggested?

• Do the proposed equipment’s have the technical capacity to hold the data required to use the new system?

• Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?

• Can the system be upgraded if developed?

• Are there technical guarantees of accuracy, reliability, ease of access and data security? Earlier no system existed to cater to the needs of ‘Secure InfrastructureImplementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to the users. The database’s purpose is to create, establish and maintain a work flow among various entities in order to facilitate all concerned users in their various capacities or roles.

Permission to the users would be granted based on the roles specified.Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

**3.2 OPERATIONAL FEASIBILITY**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

• Is there sufficient support for the management from the users?

• Will the system be used and work properly if it is being developed andimplemented?

• Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

**3.3 ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

**1.3 Definitions, Acronyms and Abbreviations :**

* **JSP :** Java Server Pages – A leading server side technology.
* **JVM :** Java Virtual Machine – needed for running java programs.
* **HTML -** Hypertext Markup Language used for creation of static web pages.
* **J2EE -** Java 2 Enterprise Edition – Platform for development of multitier java applications
* **XML -** Extensible Markup Language.
* **HTTP -** Hypertext Transfer Protocol –transaction oriented client- server protocol
* **HTTPS -** Secure Hypertext Transfer Protocol – Uses Secure Socket Layer for secure transmission of data.
* **AJAX –** Asynchronous Java Script and Extensible Markup Language.
* **RSS –** Really Simple Syndication

**1.4 References :**

* IEEE SRS Format
* Object Oriented Systems Development –Ali Bahrami (McGraw-Hill)
* Software Engineering – A Practitioner’s Approach – Roger S. Pressman – (McGraw-Hill)
* Project specification requirement (provided by IBM)
* Database System Concepts – Abraham Silberschatz
* JAVA –Complete Reference(Herbert Schildt)
* Servlet & JSP –Head First

**1.5 Tools to be used :**

**Design Tools :**

* Altova UML tool
* Smart Draw
* Star UML tool
* Rational Rose plugin for Eclipse

**Programming Tools :**

* Netbeans
* Macromedia Dreamweaver

**Technologies :**

**User Interface :**

* **JSP/Servlets** – Basic text files with Java code in HTML page or vice versa.
* **AJAX –** Provides server interaction without page reloading.
* **XML-** Use of XML as a data type in the database for RFC Module.

**Middleware :**

* **MySql –** A high-end database server for enterprise purposes.
* **Tomcat –** A user-friendly application server which supports all the modern web development practices.

**Other essential technologies and Tools:**

* **Eclipse SDK -6.0**
* **RSS –** Providesdynamic updates to the clients in an easy fashion**.**
* **Tivoli -** It provides an intelligent infrastructure management, to manage and enhance the business value of the client’s IT system**.**

**Functional components of the project**

Following is a list of functionalities of the system. More functionalities that you find suitable can be added to this list. And, in places where the explanation of functionality is not adequate, you can make proper assumptions and proceed.

There are registered people in the system (students, faculty, librarian et al). Each one of them may have some exclusive privileges .

* A users should be able to
* login to the system through the first page of the application
* change the password after logging into the system
* register himself and then login as user.
* Upload his/her picture and update his/her profile.
* see the profile of the Doctors and follow a particular Doctor by adding in his following list.
* search for a particular Doctor based on the name of the Doctors or country or specialty and experience
* see the number of Doctors whom he/she follows.
* See the message posted by Doctors.
* Download the profile of Doctors and users as pdf.
* Give feedback to a particular doctor.
* The Doctors should be able to
* Register and login on home page.
* change the password after logging into the system
* Upload his picture and update his profile.
* See the user who follow her in his follow list.
* See the number of followers .
* Post the message.
* Download profile of Doctors/Users as pdf.
* Search for a Doctors by his name or country or specialty and experience.
* Participate in discussion.

**1.6 Hardware / Software Requirements :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Client Side** | **Operating System** | **Processor** | **Disk Space** | **RAM** |
| Browser (Internet Explorer, Mozilla Firefox, Opera) | ANY | Pentium II onwards, AMD Opteron onwards | 1 GB | 64 MB |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Server Side** | **Operating System** | **Processor** | **Disk Space** | **RAM** |
| Tomcat Application Server v 7.0 | Windows(Development End), ANY other can also be used | Pentium IV @500 MHz onwards, AMD Opteron onwards | 2 GB | 512 MB (minimum),  1 GB (recommended) |
| MySQL Database Server v 5.3 | Windows (Development End), ANY other can also be used | Pentium IV @ 500 MHz onwards, AMD Opteron onwards | 2 GB + data storage | 512 MB (minimum),  1 GB (recommended) |
| NetBeans IDE 6.5 | Windows  (Development End), ANY other can also be used | Pentium !V @ 500 MHz onwards, AMD Opteron onwards | 10 GB+  Data storage | 1 GB (minimum)  2 GB  (recommended) |

**3.2** **Supplementary Requirements**:

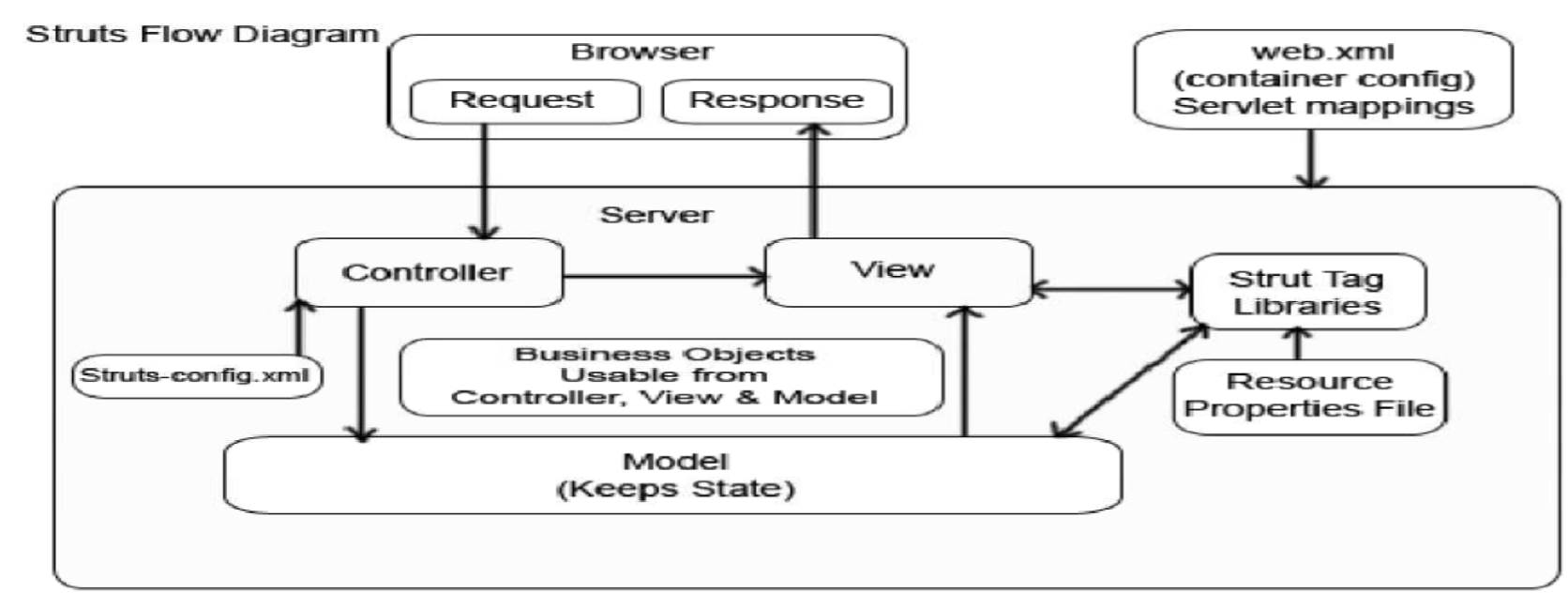
* Have hours of operation that are 24x7 : Being an automated process, this system can work non-stop for all the time. This requires a UPS (Uninterruptible Power Supply) backup for atleast 8 hours
* Server Performance – The modern server is needed with advance features like load balancing, clustering, database connection pooling and caching.
* Session Management – The server should be inherently capable of handling sessions, so that the developer hs to be least worried about such trivial details.

**User Characteristsics :**

* Should have a basic knowledge of English.
* Should be familiar with basic concepts of computer including web browsing and file uploading.

5-Technology

**Struts Framework**



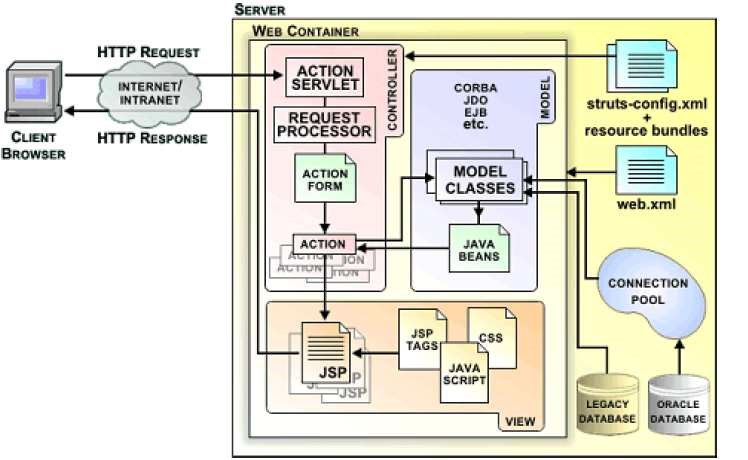
**Working of Struts Application**

The Struts framework is composed of approximately 300 classes and interfaces organized in about 12 top-level packages. Besides the utility and helper classes, much of the framework consists of classes and interfaces for working with either the controller functionality or the presentation through custom tag libraries. The choice of a model layer is a decision that's left up to you. Some argue that this is a deficiency in the framework, while seasoned Struts developers would say that this freedom is warranted and welcomed. A high-level view of the Struts architecture is shown in Figure

The Struts Framework offers a number of benefits to developers:

* Based on a Model 2 MVC implementation.
* Supports Java's "Write Once, Run Anywhere" philosophy.
* Supports different model implementations (JavaBeans, EJB, etc.).
* Support for internationalization (I18N).
* Includes a rich set of JSP tag libraries.
* Contains many extension points for customization.

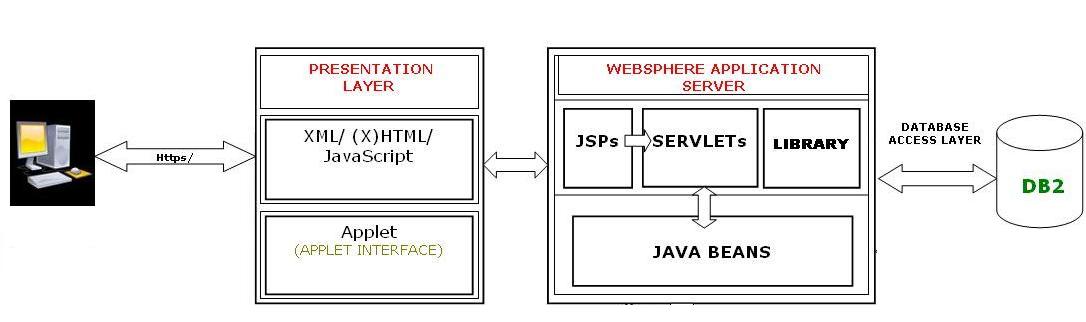
Supports different presentation implementations (JSP, XML/XSLT, JavaServer Faces).



***High Level View of Struts Architecture***

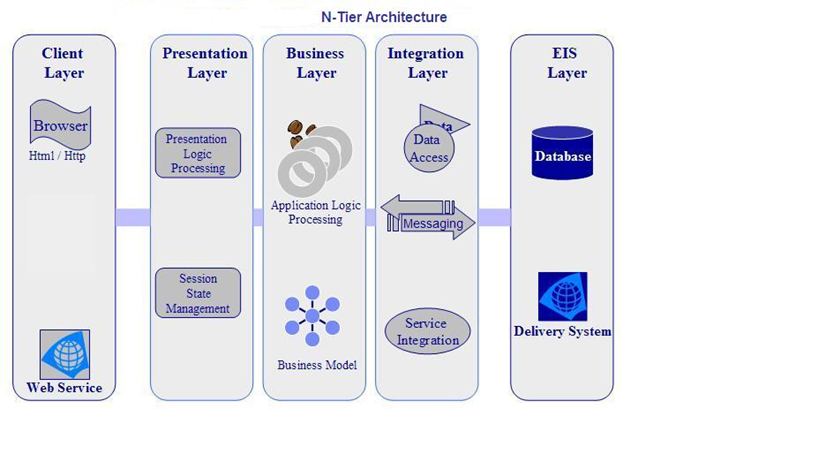
**Overall description(Diagrams)**

**2.1 Architecture Diagram :**



The above diagram basically depicts what is the architecture in the whole processing of our project that we have used. It clearly shows all the layers and what all layers are composed of. Each components are self ecxplanatory and hence are not explained further.

The architecture of the whole project is based on the 4+1 view of the rational approach:

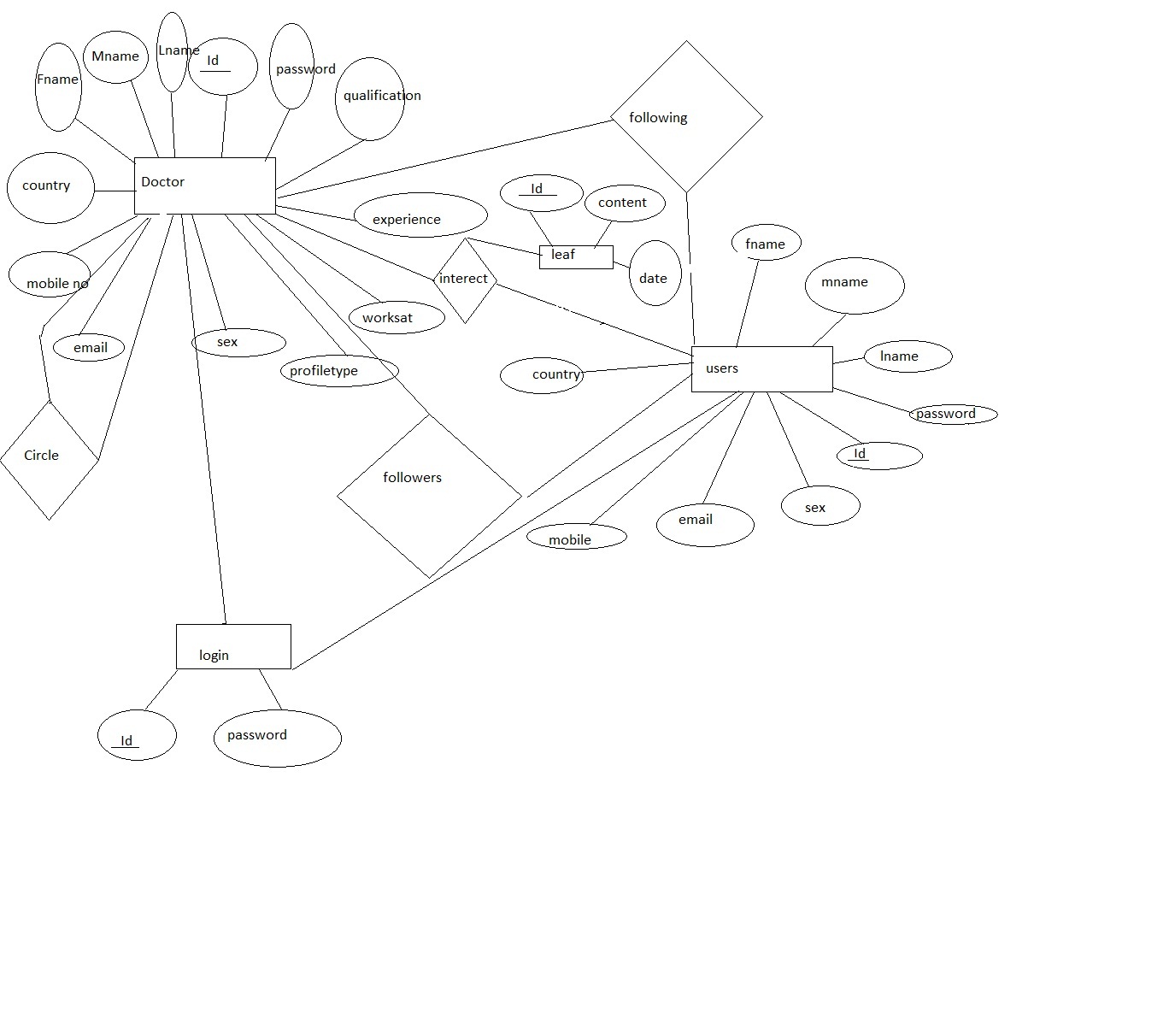


The layering model is based on a responsibility layering strategy that associates each layer with a particular responsibility.

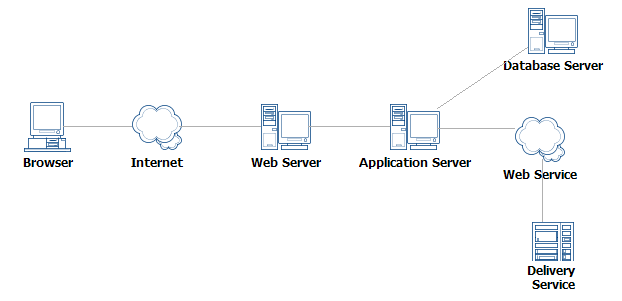
This strategy has been chosen because it isolates various system responsibilities from one another, so that it improves both system development and maintenance.

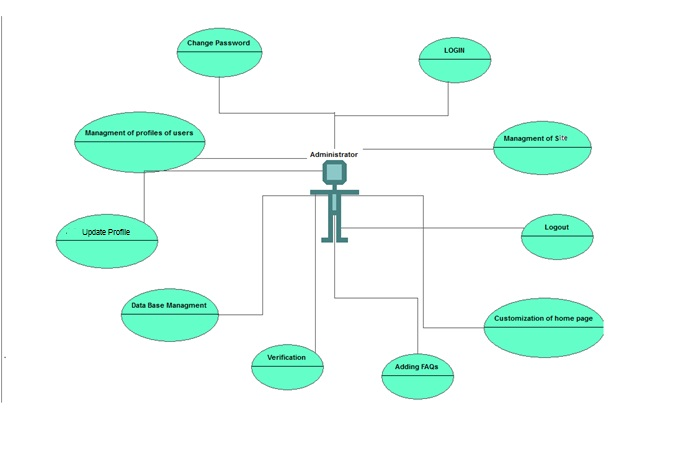
**E-R ( Entity Relationship )Diagram**:-

The E-R Diagram shows all the basic entities required in the project and the relationships existing between them. The E-R model has covered all the dependencies and the sharing of the data attributes between the entities.

****

**Deployment View Architecture of the Project:**

**3.1 Use Case Modelling**

**Admin:**

**Name of use case:** Registration

**Description:** Lets new users that is doctor and users to register themselves with the aayush website.

**Preconditions:**

* The user has connected to the omicron web interface.
* He has his id with himself.

**Normal flow of events:**

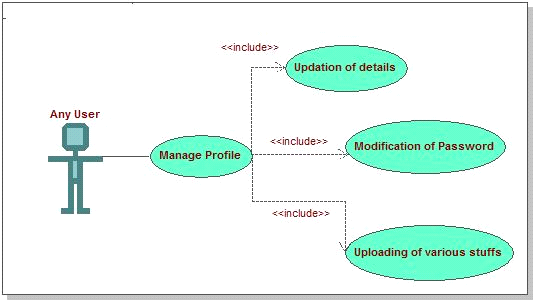
* The user logs in.
* The user fills his details.
* The account has been created.

**Alternate flow of events:** None.

**Post Condition:** None.

**Name of use case:**  Manage Profile

**Description:** Lets registered users to manage their profiles to update changes and upload image.



**Preconditions:**

* The user is logged in.
* The user provides valid information.

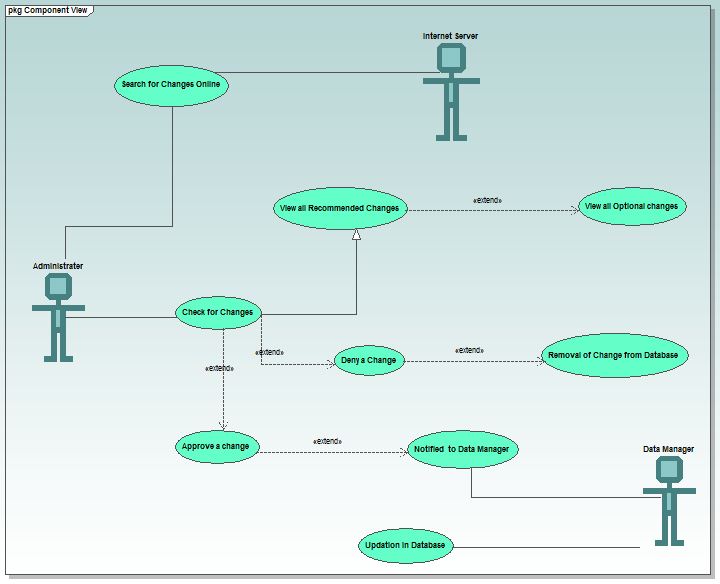
**Normal flow of events:**

* The user logs in.
* The user fills his details.
* Updates and saves the changes.

**Alternate flow of events:** None.

**Post Condition:** None.

**Name of use case:** Approval of changes



**Description:** Lets admin to approve the recommended changes by the faculty and the data manager.

**Preconditions:** The admin is logged in.

There have been some changes recommended for the admin to approve.

**Normal flow of events:**

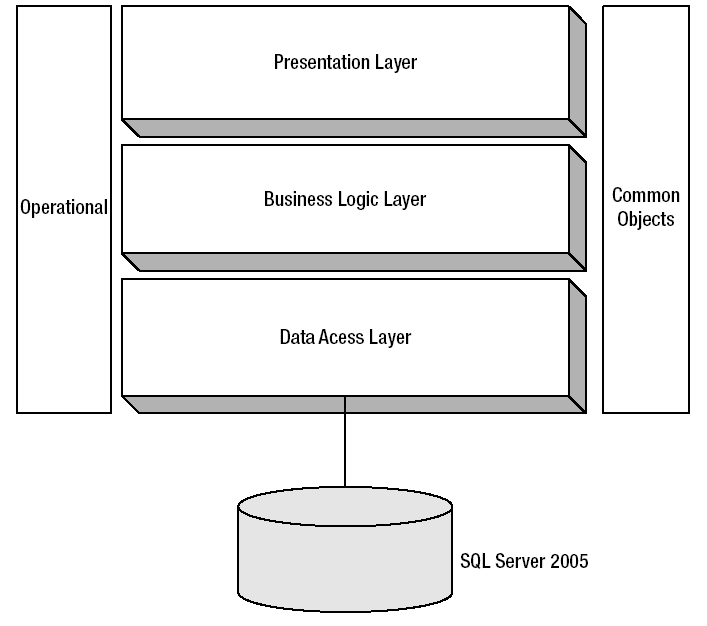
* The admin logs in.
* He checks changes details.
* Approves the appropriate one’s.

**Alternate flow of events:** None.

**Post Condition:**

* The data manager logs in.
* He views the approved changes.
* He modifies them in the most appropriate way.

**6.5 Database Connectivity:**

****

**4-Software Requirements specifications**

The software, Site Explorer is designed for management of web sites from a remote location.

**Users:**

* **Doctor:-**
* **Download Profile as pdf:-** A Doctor can download the profile of others and which are updated on regular basis.
* **Post Query:-**A Doctor can post feedback on any discussion.
* **View Feedback:-**A doctor can also view the feedbacks posted by other doctors.
* **Search for any Doctors:-** A doctor can search doctors on the basis of Country , Specialization and Work Experience .
* **View profile of Doctors:-** A doctor can view the profile.
* **Registration:-**A doctor would have to register for joining this system.
* **Edit Profile:-**A doctor has also the rights to update the profile
* **Manage Friend Circle –**A doctor can manage his friend list.
* **View Followers –**A doctors can view his followers.
* **Upload profile picture:-**A doctor can upload his picture.
* **Non-Doctor:-** 
  + - * **Download Profile as pdf:-** A user can download the profile of doctors and which are updated on regular basis.
      * **Post Query:-**A user can post feedback on any discussion.
      * **View Feedback:-**A user can also view the feedbacks posted by other doctors.
      * **Search for any Doctors:-** A user can search any doctor on the bassis of Country , Specialization and Work Experience .
      * **View profile of Doctors:-** A user can view the profile.
      * **Registration:-**A user would have to register for joining this system.
      * **Edit Profile:-**A user has also the rights to update the profile.
      * **View Following Doctor List-** A user can see following doctor list.
      * **Upload profile picture:-**A user can upload his picture.
* **Data Manager:-**
* **Manage Profile:-** Data manager can manage his profile and update changes on it.
* **View all details:-**The data manager has rights to view any details on the website.
* **Manage approved changes:-**The data manager would have to modify the database according to the changes approved by the dean.
* **Guest:-**
* **View FAQ:-** Guest can view Frequently asked Questions.
* **View Users and Doctors increasing count-** Guest can view number of registered users and doctors.

8-Testing

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors and error-prone areas in a system. The implementation is the final and important phase. The user tests the system and changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testing, errors are noted and correctness is the mode.

**OBJECTIVES OF TESTING**:

The objectives of testing are:

● Testing is a process of executing a program with the intent of finding errors.

● A Successful test case is one that uncovers an as- yet-undiscovered error.

System testing is a stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as per the user need, before the live operation commences. As stated before, testing is vital to the success of a system. System testing makes a logical assumption that if all parts of the as system are correct, the goal will be successfully achieved. A series of tests are performed before the system is ready for the user acceptance test.

The Testing Steps are:

● **Unit Testing**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

● **Integration Testing**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

● **User Acceptance Testing**

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

**FUTURE SCOPE OF APPLICATION :**

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

**SOFTWARE SCOPE:**

**Extensibility**: This software is extendable in ways that its original developers may not expect. The following principles enhances extensibility like hide data structure, avoid traversing multiple links or methods, avoid case statements on object type and distinguish public and private operations.

**Reusability**: Reusability is possible as and when require in this application. We can update it next version. Reusable software reduces design, coding and testing cost by amortizing effort over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability: Sharing of newly written code within a project and reuse of previously written code on new projects.

**Understandability:** A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method,

which small and coherent helps to accomplish this.

**Cost-effectiveness:** Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the

entire requirement. Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

**web.xml**

<?xml version="1.0" encoding="UTF-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

<init-param>

<param-name>config</param-name>

<param-value>/WEB-INF/struts-config.xml</param-value>

</init-param>

<init-param>

<param-name>debug</param-name>

<param-value>2</param-value>

</init-param>

<init-param>

<param-name>detail</param-name>

<param-value>2</param-value>

</init-param>

<load-on-startup>2</load-on-startup>

</servlet>

<servlet>

<servlet-name>leaf</servlet-name>

<servlet-class>hp.leaf</servlet-class>

</servlet>

<servlet>

<servlet-name>logout</servlet-name>

<servlet-class>hp.logout</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>leaf</servlet-name>

<url-pattern>/leaf</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>logout</servlet-name>

<url-pattern>/logout</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout>

30

</session-timeout>

</session-config>

<welcome-file-list>

<welcome-file>index.jsp</welcome-file>

</welcome-file-list>

<jsp-config>

<taglib>

<taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri> <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-html.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-logic.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-nested.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-nested.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-tiles.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-tiles.tld</taglib-location>

</taglib>

</jsp-config>

</web-app>

**struts-config.xml**

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE struts-config PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 1.3//EN” "http://jakarta.apache.org/struts/dtds/struts-config\_1\_3.dtd">

<struts-config>

<form-beans>

<form-bean name="pdf1Bean" type="hp.pdf1Bean"/>

<form-bean name="searchBean" type="hp.searchBean"/>

<form-bean name="pdfBean" type="hp.pdfBean"/>

<form-bean name="AddDoctorBean" type="hp.AddDoctorBean"/>

<form-bean name="CircleAddBean" type="hp.CircleAddBean"/>

<form-bean name="leafBean" type="hp.leafBean"/>

<form-bean name="updateUserBean" type="hp.updateUserBean"/>

<form-bean name="updateDoctBean" type="hp.updateDoctBean"/>

<form-bean name="UserRegBean" type="hp.UserRegBean"/>

<form-bean name="DoctorRegBean" type="hp.DoctorRegBean"/>

<form-bean name="loginBean" type="hp.loginBean"/>

</form-beans>

<global-exceptions>

</global-exceptions>

<global-forwards>

<forward name="welcome" path="/Welcome.do"/>

</global-forwards>

<action-mappings>

<action path="/Home" forward="def"/>

<action path="/doctor" forward="def1"/>

<action path="/user" forward="def2"/>

<action path="/register" forward="def3"/>

<action path="/login" forward="def4"/>

<action path="/updateDoct" forward="def6"/>

<action path="/updateUser" forward="def7"/>

<action path="/about" forward="def8"/>

<action path="/faq" forward="def9"/>

<action path="/following" forward="def10"/>

<action path="/followers" forward="def12"/>

<action path="/circle" forward="def13"/>

<action input="/login.jsp" name="loginBean" path="/abc" scope="request" type="hp.loginAction" validate="true">

<forward name="doctor" path="def1"/>

<forward name="user" path="def2"/>

<forward name="admin" path="def0"/>

</action>

<action input="/profileUpdateDoct.jsp" name="updateDoctBean" path="/pqr" scope="request" type="hp.updateDoctAction">

<forward name="profileDoct" path="def11"/>

</action>

<action input="/RegUser.jsp" name="UserRegBean" path="/wxyz" scope="request" type="hp.UserRegAction" validate="true">

<forward name="user" path="def2"/>

</action>

<action input="/RegDoctor.jsp" name="DoctorRegBean" path="/xyz" scope="request" type="hp.DoctorRegAction" validate="true">

<forward name="doctor" path="def1"/>

</action>

<action input="/profileUpdateUser.jsp" name="updateUserBean" path="/pqrs" scope="request" type="hp.updateUserAction">

<forward name="profileUser" path="def21"/>

</action>

<action input="/leafAyush.jsp" name="leafBean" path="/lmn" scope="request" type="hp.leafAction">

<forward name="doctor" path="def1"/>

</action>

<action input="/circle.jsp" name="CircleAddBean" path="/ert" scope="request" type="hp.CircleAddAction">

<forward name="circle" path="def13"/>

</action>

<action input="/following.jsp" name="AddDoctorBean" path="/rty" scope="request" type="hp.AddDoctorAction">

<forward name="following" path="def10"/>

</action>

<action input="/following.jsp" name="pdfBean" path="/pdf" scope="request" type="hp.pdfAction"/>

<action input="/searchAyush.jsp" name="searchBean" path="/srch" scope="request" type="hp.searchAction">

<forward name="searchResult" path="def14"/>

</action>

<action input="/circle.jsp" name="pdf1Bean" path="/pdf1" scope="request" type="hp.pdf1Action"/>

</action-mappings>

<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>

<message-resources parameter="com/myapp/struts/ApplicationResource"/>

<!-- ========================= Tiles plugin ===============================-->

<plug-in className="org.apache.struts.tiles.TilesPlugin" >

<set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />

<set-property property="moduleAware" value="true" />

</plug-in>

<!-- ========================= Validator plugin ================================= -->

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">

<set-property

property="pathnames"

value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>

</plug-in>

</struts-config>

**tiles-defs.xml**

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE tiles-definitions PUBLIC

"-//Apache Software Foundation//DTD Tiles Configuration 1.1//EN"

"http://jakarta.apache.org/struts/dtds/tiles-config\_1\_1.dtd">

<tiles-definitions>

<definition name="def" path="/layoutHome.jsp">

<put name="header" value="/headerHome.jsp" />

<put name="footer" value="/footer.jsp" />

<put name="ads" value="/Add.jsp" />

<put name="news" value="/News.jsp" />

<put name="body" value="/homeLogin.jsp" />

</definition>

<definition name="def3" extends="def">

<put name="body" value="/register.jsp" />

</definition>

<definition name="def4" extends="def">

<put name="body" value="/login.jsp" />

</definition>

<definition name="def5" extends="def">

<put name="body" value="/LogOut.jsp" />

</definition>

<definition name="def8" extends="def">

<put name="body" value="/Aboutus.jsp" />

</definition>

<definition name="def9" extends="def">

<put name="body" value="/Faq.jsp" />

</definition>

<definition name="def1" path="/layoutDoctor.jsp">

<put name="header" value="/headerDoct.jsp" />

<put name="footer" value="/footer.jsp" />

<put name="ads" value="/Add.jsp" />

<put name="news" value="/News.jsp" />

<put name="search" value="/searchAyush.jsp" />

<put name="leaf" value="/leafAyush.jsp" />

<put name="update" value="/updateAyush.jsp" />

</definition>

<definition name="def2" path="/layoutUser.jsp">

<put name="header" value="/headerUser.jsp" />

<put name="footer" value="/footer.jsp" />

<put name="ads" value="/Add.jsp" />

<put name="news" value="/News.jsp" />

<put name="search" value="/searchAyush.jsp" />

<put name="update" value="/updateAyush.jsp" />

</definition>

<definition name="def6" path="/layoutProfile.jsp">

<put name="header" value="/headerDoct.jsp" />

<put name="footer" value="/footer.jsp" />

<put name="ads" value="/Add.jsp" />

<put name="news" value="/News.jsp" />

<put name="profile" value="/profileUpdateDoct.jsp" />

</definition>

<definition name="def12" extends="def6">

<put name="profile" value="followers.jsp"/>

</definition>

<definition name="def13" extends="def6">

<put name="profile" value="circle.jsp"/>

</definition>

<definition name="def7" extends="def6">

<put name="header" value="/headerUser.jsp" />

<put name="profile" value="/profileUpdateUser.jsp" />

</definition>

<definition name="def10" extends="def6">

<put name="header" value="/headerUser.jsp" />

<put name="profile" value="/following.jsp" />

</definition>

<definition name="def11" extends="def6">

<put name="profile" value="/profileDoct.jsp"></put>

</definition>

<definition name="def21" extends="def7">

<put name="profile" value="/profileUser.jsp"></put>

</definition>

<definition name="def14" extends="def6">

<put name="profile" value="/searchResult.jsp"></put>

</definition>

</tiles-definitions>

**validation.xml**

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE form-validation PUBLIC

"-//Apache Software Foundation//DTD Commons Validator Rules Configuration 1.1.3//EN"

"http://jakarta.apache.org/commons/dtds/validator\_1\_1\_3.dtd">

<form-validation>

<global>

</global>

<formset>

<form name="loginBean">

<field

property="id"

depends="required">

<arg key="UserId" resource="false"/>

</field>

<field

property="pass"

depends="required,mask">

<arg key="Password" resource="false"/>

<var>

<var-name>mask</var-name>

<var-value>^[0-9a-zA-Z]\*$</var-value>

</var>

</field>

</form>

<form name="UserRegBean">

<field

property="fname"

depends="required">

<arg key="First Name" resource="false"/>

</field>

<field

property="id"

depends="required">

<arg key="UserId" resource="false"/>

</field>

<field

property="pass"

depends="required,mask">

<arg key="Password" resource="false"/>

<var>

<var-name>mask</var-name>

<var-value>^[0-9a-zA-Z]\*$</var-value>

</var>

</field>

<field

property="sex"

depends="required">

<arg key="Gender" resource="false"/>

</field>

<field

property="country"

depends="required">

<arg key="Country" resource="false"/>

</field>

</form>

<form name="DoctorRegBean">

<field

property="fname"

depends="required">

<arg key="First Name" resource="false"/>

</field>

<field

property="id"

depends="required">

<arg key="UserId" resource="false"/>

</field>

<field

property="pass"

depends="required,mask">

<arg key="Password" resource="false"/>

<var>

<var-name>mask</var-name>

<var-value>^[0-9a-zA-Z]\*$</var-value>

</var>

</field>

<field

property="sex"

depends="required">

<arg key="Gender" resource="false"/>

</field>

<field

property="qual"

depends="required">

<arg key="Qualification" resource="false"/>

</field>

<field

property="special"

depends="required">

<arg key="Speciality" resource="false"/>

</field>

<field

property="exp"

depends="required">

<arg key="Experience" resource="false"/>

</field>

<field

property="country"

depends="required">

<arg key="Country" resource="false"/>

</field>

</form>

</formset>

<formset language="fr">

<constant>

<constant-name>postalCode</constant-name>

<constant-value>^[0-9a-zA-Z]\*$</constant-value>

</constant>

<form name="logonForm">

<field

property="username"

depends="required">

<arg key="logonForm.username"/>

</field>

<field

property="password"

depends="required,mask">

<arg key="logonForm.password"/>

<var>

<var-name>mask</var-name>

<var-value>^[0-9a-zA-Z]\*$</var-value>

</var>

</field>

</form>

</formset>

</form-validation>

**layoutHome.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="t" uri="/WEB-INF/struts-tiles.tld" %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<table border="1" width="1000" height="850" align="center">

<tr>

<td colspan="2" height="170" width="1000"><t:insert attribute="header"/></td>

</tr>

<tr height="630" width="1000">

<td width="200"><t:insert attribute="ads"/> </td>

<td width="800"><t:insert attribute="body"/></td>

</tr>

<tr height="50">

<td colspan="2"><t:insert attribute="footer"/></td>

</tr>

</table>

</body>

</html>

**headerHome.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body >

<table border="0" width="1000" background="images/header21.jpg" align="center" height="150">

<tr height="100">

<td colspan="4"></td>

</tr></table>

<table align="center" width="1000" background="bg.jpg" height="20" cellpadding="0" cellspacing="0" border="0">

<tr>

<td align="center"><a href="about.do"><input type="image" src="images/aboutus.jpg" ></a></td>

<td align="center"><a href="register.do"><input type="image" src="images/register.jpg" ></a></td>

<td align="center"><a href="login.do"><input type="image" src="images/login.jpg" ></a></td>

<td align="center"><a href="faq.do"><input type="image" src="images/faqs.jpg" name="faqs"></a></td>

</tr>

</table>

</body>

</html>

**Footer.jsp**

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<table ><tr><td height="50" width="1000"><img src="images/footer1.jpg"/></td></tr></table>

</body>

</html>

**HomeLogin.jsp**

<%@page import="java.sql.\*"%>

<%@page import="hp.myConnection"%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

<%

Connection con=myConnection.dbConnection();

Statement st=con.createStatement();

String sql="select count(\*) from login where role='user'";

ResultSet rs=st.executeQuery(sql);

%>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<table border="1" background="images/bgAyush.jpg" height="630" width="800">

<tr style="vertical-align: bottom">

<td >

<%

while(rs.next()){

int cuser=rs.getInt(1);

%>

<center> <h1> Users on website</h1> <input type="text" value="<%= cuser %>"/></center>

<% } %>

</td>

</tr>

<tr style="vertical-align: bottom">

<td>

<%

Connection con1=hp.myConnection.dbConnection();

Statement st1=con1.createStatement();

String sql1="select count(\*) from login where role='doctor'";

ResultSet rs1=st1.executeQuery(sql1);

if(rs1.next()){

int duser=rs1.getInt(1);

%>

<center><h1> Doctors on website</h1> <input type="text" value="<%= duser %>"/></center>

<% } %>

</td>

</tr> </table> </body>

</html>

**myConnection.java**

package hp;

import java.sql.\*;

public class myConnection {

public static Connection c;

public static Connection dbConnection()

{

try

{

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

c=DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/aayush", "root", "123");

}

catch(Exception f)

{

System.out.println(f);

}

finally

{return (c);}

}

}

**loginBean.java**

package hp;

public class loginBean extends org.apache.struts.action.ActionForm {

private String id,pass,ch;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getPass() {

return pass;

}

public void setPass(String pass) {

this.pass = pass;

}

public String getCh() {

return ch;

}

public void setCh(String ch) {

this.ch = ch;

}

}

**loginAction.java**

package hp;

import java.sql.\*;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.struts.action.ActionForm;

import org.apache.struts.action.ActionForward;

import org.apache.struts.action.ActionMapping;

public class loginAction extends org.apache.struts.action.Action {

@Override

public ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response)

throws Exception {

loginBean ob=(loginBean)form;

Connection c=myConnection.dbConnection();

String id=ob.getId();

String pass=ob.getPass();

String ch=ob.getCh();

Statement st=c.createStatement();

ResultSet rs=st.executeQuery("select role from login where id='"+id+"' and pass='"+pass+"' ");

if(rs.next())

{ HttpSession ses=request.getSession(true);

ses.setAttribute("id",id);

if(ch!=null)

{

if(ch.equals("true"))

{ String t=id+":"+pass;

Cookie k=new Cookie("LoginCookie",t);

k.setMaxAge(60\*60\*24\*7);

response.addCookie(k);

}

}

String r=rs.getString(1);

if(r.equalsIgnoreCase("admin"))

{

return mapping.findForward("admin");

}

else if(r.equalsIgnoreCase("doctor"))

{

return mapping.findForward("doctor");

}

else if(r.equalsIgnoreCase("user"))

{

return mapping.findForward("user");

}

}

String s="Invalid username or password";

request.setAttribute("msg", s);

return mapping.getInputForward();

}

}

**pdfAction.java**

package hp;

import org.apache.struts.action.ActionForm;

import org.apache.struts.action.ActionForward;

import org.apache.struts.action.ActionMapping;

import javax.servlet.http.\*;

import com.itextpdf.text.pdf.PdfWriter;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Element;

import com.itextpdf.text.Paragraph;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class pdfAction extends org.apache.struts.action.Action {

@Override

public ActionForward execute(ActionMapping mapping, ActionForm form,

HttpServletRequest request, HttpServletResponse response)

throws Exception {

try {

response.setContentType("application/pdf");

response.setHeader("Content-Disposition", "attachment; filename=\"profile.pdf\"");

Document document = new Document();

pdfBean ob=(pdfBean)form;

Connection con=myConnection.dbConnection();

String did=ob.getDid();

HttpSession ses=request.getSession(true);

String uid=(String)ses.getAttribute("id");

PdfWriter.getInstance(document,response.getOutputStream());

Statement st = con.createStatement();

ResultSet rs;

rs = st.executeQuery("select \* from doctor where id='" + did + "'");

if (rs.next()) {

String fname=rs.getString(1);

String mname=rs.getString(2);

String lname=rs.getString(3);

String sex=rs.getString(6);

String qual=rs.getString(7);

String special=rs.getString(8);

String exp=rs.getString(9);

String workat=rs.getString(10);

String email=rs.getString(11);

String mobile=rs.getString(12);

String country =rs.getString(13);

Paragraph p=new Paragraph("Aayush Doctor Profile");

Paragraph p1 = new Paragraph("First Name : " + fname);

Paragraph p2 = new Paragraph("Middle Name : " + mname);

Paragraph p3 = new Paragraph("Last Name : " + lname);

Paragraph p4 = new Paragraph("Gender: " + sex);

Paragraph p5 = new Paragraph("Qualification : " + qual);

Paragraph p6 = new Paragraph("Specialization : " + special);

Paragraph p7 = new Paragraph("Experience : " + exp);

Paragraph p8 = new Paragraph("WorksAt: " + workat);

Paragraph p9 = new Paragraph("Email ID : " + email);

Paragraph p10 = new Paragraph("Mobile No: " + mobile );

Paragraph p11 = new Paragraph("Country: " + country + "\n");

p.setAlignment(Element.ALIGN\_CENTER);

p1.setAlignment(Element.ALIGN\_LEFT);

p2.setAlignment(Element.ALIGN\_LEFT);

p3.setAlignment(Element.ALIGN\_LEFT);

p4.setAlignment(Element.ALIGN\_LEFT);

p5.setAlignment(Element.ALIGN\_LEFT);

p6.setAlignment(Element.ALIGN\_LEFT);

p7.setAlignment(Element.ALIGN\_LEFT);

p8.setAlignment(Element.ALIGN\_LEFT);

p9.setAlignment(Element.ALIGN\_LEFT);

p10.setAlignment(Element.ALIGN\_LEFT);

p11.setAlignment(Element.ALIGN\_LEFT);

document.open();

document.add(p);

document.add(p1);

document.add(p2);

document.add(p3);

document.add(p4);

document.add(p5);

document.add(p6);

document.add(p7);

document.add(p8);

document.add(p9);

document.add(p10);

document.add(p11);

}

document.close();

} catch (SQLException ex) {

} catch (DocumentException ex) {

}

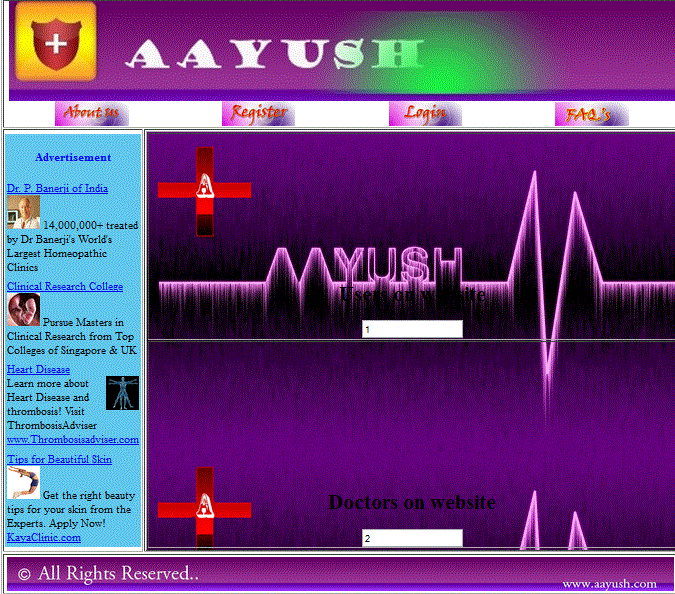
return mapping.findForward("following");

}

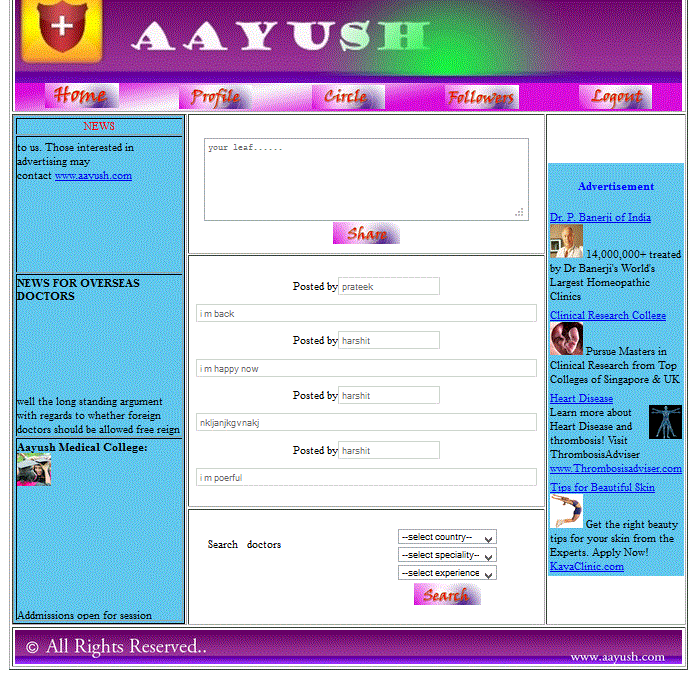
}

**Snapshots**

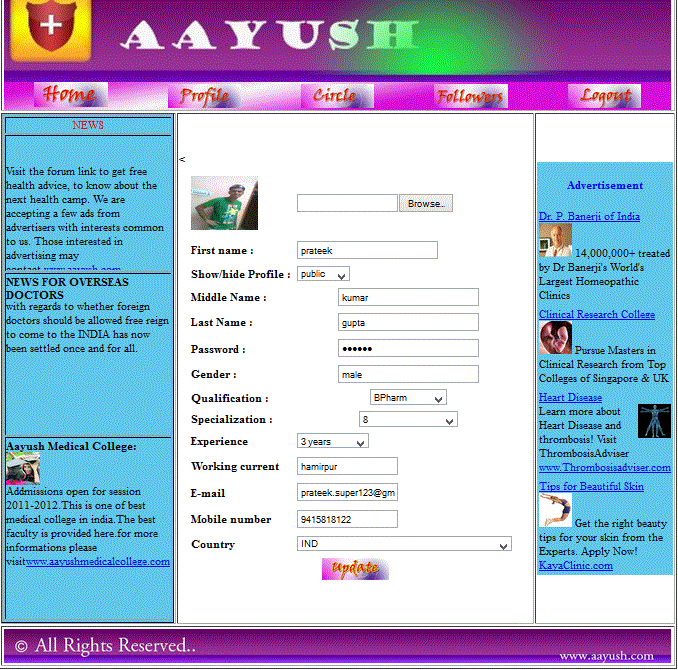
1- **HomePage**



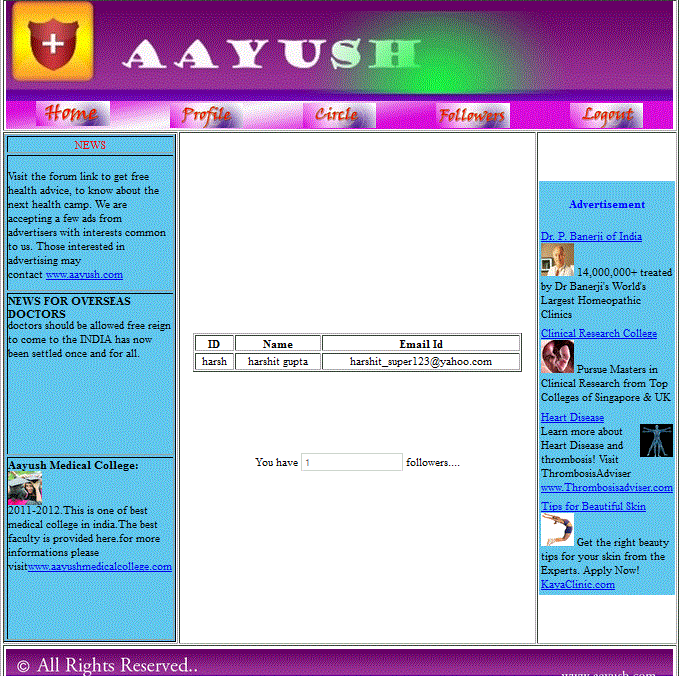
**2- Home Page of Doctor**

****

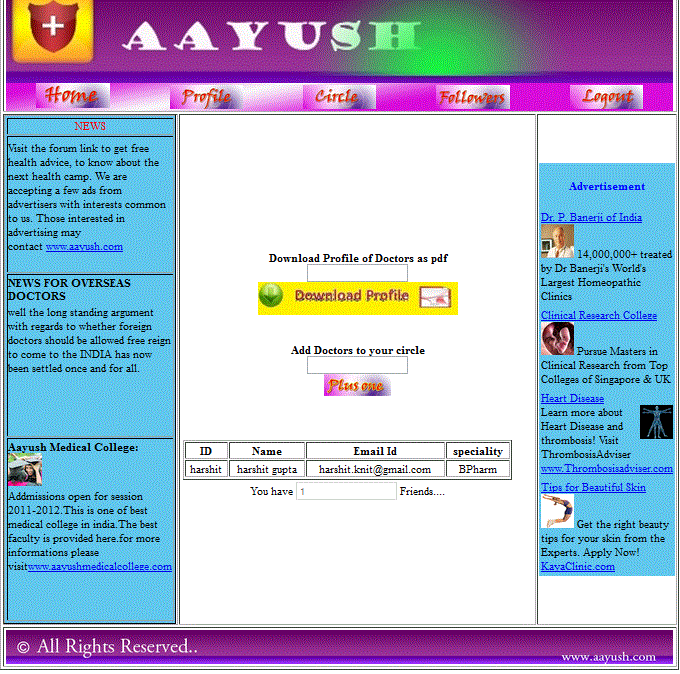
**3-Profile Page of Doctor**



**4- Followers Page of Doctor**

****

**5-Circle Page of Doctors**

****

.