**SOFTWARE REQURIMENT SPECIFICATION (SRS) FOR**

**E HEALTH CARE ADVISOR**

**Introduction**

**1.1 Purpose:**

The main objective of this project is to implement a computer based Healthcare Information System. This system will help the users to identify certain diseases by answering certain questions asked by the system. Based on the diagnose received the user will be getting some suggestion of medicines that are available at the local chemist without prescription with an advice to visit the doctor. The system once ready should be able to train itself with the feedback given to it (Artificial Intelligence). The database will be developed with open source software.

E – Healthcare Advisor is a web application which provides online medical services to everyone at their doorstep. The users living in metro or remote village can connect through internet or approach nearby kiosk to get these services. This web application is more effective, quick in providing medical help especially to people in villages where very few doctors are present. This helps the patients to maintain a neat health record and to lead a healthy life.

* 1. **Scope: The Scope of the Project includes:**

There are four basic users - Patient, Admin, Doctors .

1.All users have their own profiles in VMH.

2.The web-cam interaction between doctor and patient.

3.Patients can search for doctor and make online appointments. They also can view their health record, lab reports, doctor’s prescription and medical expenses. Patient can also register complaint on any doctor.

4.Doctor’s can give appointments, e-prescription and can view patient’s health record.

5.Admin has the authority to add/delete users, grant permission to doctors and kiosk manager, to generate and view reports. He also views the complaints of patients and takes necessary actions.

**1.3 Definitions, Acronyms and Abbreviations**

**EHA**

E – Healthcare Advisor. It’s a web application that provides online medical services for people.

**Admin**

Administrator. He has the authority to add/delete users, grant permission to doctors and kiosk manager.

* 1. **References**

**1.**Object Oriented Modeling and Design with UML-Michael Blaha, James Rambaugh.

2.Software Engineering, Seventh Edition, Ian Sommerville.

3.IBM Red Books.

4.IBM TGMC Website

5.IBM – [www.ibm.in/developerworks](http://www.ibm.in/developerworks)

**1.5 Existing System:**

**1.**Registration for users

**2.**Discussion forum

**2.Over All Description**

**2.1.Product Perspective**

**Software Requirements**

Front end client on Internet

FF web Browser , Debian Os

Web server

Data Base Server

Development End

**Hardware requirements:**

1.**Client Side**

processor with 800MHZ

RAM 128MHz

Disk space 900MB

Mozila firefox6

**2.Server Side**

processor Intel Pentium

Ram 1500MB

**3. Communication Interface**

using HTTP protocol

using HTTPS protocol

**2.2 Product Functions:**

1.To conduct a diagnose in order to identify the disease

2.To design a health care management system

3.To maintain patient history and system keep self learning to update the database

|  |  |  |
| --- | --- | --- |
| Class of use cases | Use cases | Description of use cases |
| Use case related to  Installation | Installation |  |
| Use case related to  System Authorization | Login |  |
| Change password |
| Use case related to  Register | Registration | Register new patients |
| Use case related to  Search | Search | Search the patient is already present in the database or not |
| Use case related to create profile | View profile | Containing personal details |
| Update profile | To update profile |
| Use case related to E health record | Create health record | Containing health record |
| Update health record | Updating health record |
| Use case related to securities | Create security |  |
| Delete security |
| Use case related to make an appointment | Seek an appointment |  |
| Use case related to delete  data |  |  |
| Use case related to disease database |  |  |
| Use case related to prescription |  |  |

**2.3 User Characteristics:**

**Users**

1.Patients

2.Nonmembers

3.Administrator

**1. Patient**:

search for doctors

get online appointments

view their record, lab reports

**2.Doctors:**

Give appointment

e-prescription

Updated and view patients health record

**3.Admin:**

Add and delete the users

Grant permissions to generate and view reports

View the complaints of patients and take necessary actions

**2.4 Principal Actors:**

The two principal actors in EHM are “user” and “system”.

**2.5 General Constraints:**

1. For full working EHM requires Internet connection.

2. EHM is single-user software.

**3 Specific Requirements:**

**3.1 Functional Requirements:**

We describe the functional requirements by giving various use cases.

Use case related to installation:

**Use Case 1:** Installation

Primary Actor: Administrator

Pre-Condition: Internet connection available.

Main Scenario:

1. User initiates EHM installation program.

2. System asks the user for the home directory in which all the working

Files will be created. User is also asked for the initial login and password.

3. User specifies the home directory and login/password.

4. System creates the working files in the specified home directory.

Working files contain:

a. Information of Patient.

b. Current Location of Doctor.

c. Backup & Restore of data.

Alternate Scenario:

a. Network failure.

b. Installation aborted.

Use cases related to system authorization:

**Use Case 2**: Login

Primary Actor: Doctor ,Patient, Administrator Pre-Condition: Internet connection available

Main Scenario:

1. Start the application. User prompted for login and password.

2. User gives the login and password.

3. System does authentication.

4. Main screen is displayed.

Alternate Scenario:

1. Authorization fails.

2. Prompt the user that he typed the wrong password.

3. Allow him to re-enter the password. Give him 3 chances.

**Use Case 3:** Change Password

Primary Actor: Doctor , Patient, Administrator

Pre-Condition: Certain User logged in

Main Scenario:

1. User initiates the password change command.

2. User is prompted for old password, new password and confirm new

Password.

3. User gives the old password, new password and confirm new

Password.

4. System does authentication.

5. New password is registered with the system.

6. User gets a message to his mail-id and mobile that password was changed.

Alternate Scenario:

1. Authorization fails.

2. Prompt the user that he typed the wrong password.

3. Allow him to re-enter the password. Give him 3 chances.

4. New password and confirm new password do not match.

4.1 Allow him to re-enter the attributes. Give 3 chances.

**Use Case 4:** Search

Primary Actor: Doctor , Patient, Administrator.

Pre-Condition: Certain User logged in.

Main Scenario:

1. User clicks on search box

2. System asks the user for the name of Patient.

3. User enters name/blood group/Image.

4. Click on the search button

Alternate Scenario:

1. Patient name does not exists.

2. Patient name exists but finger print /blood group does not match.

**Use Case 5:** Profile

Primary Actor: Patient

Pre-Condition: Patient logged in

Main Scenario:

1. Maintain the database

2. Grant role to other user

3. Revoke role from other users.

4. Back and restore the data

Alternate Scenario:

1. Data missing of Patient

2. Backup and restore is done every day

**Use Case 6:** Record

Primary Actor: Patient

Pre-Condition: Patient logged in

Main Scenario:

1. Maintain the database

2. Grant role to other user

3. Revoke role from other users.

4. Back and restore the data

Alternate Scenario:

1. Data missing of Certain Patient

2. Backup and restore is done every day

**Use Case 7:** Delete data.

Primary Actor: Patient

Pre-Condition: Patient logged in.

Main Scenario:

1.Patient initiates the ”delete data” functionality.

2. System asks for the name of the data.

3. The data is deleted.

Alternate Scenario:

1. Data does not exist.

2. Deletion fails, error message is displayed.

**Use Case 8:** Create a security.

Primary Actor: Administrator.

Pre-Condition: User logged in.

Main Scenario:

1. User selects the data in which the security is to be created.

2. User initiates the “create security” functionality.

3. System asks the user to enter the attributes of the security.

4. User specifies the following fields:

a. Name of Patient or place

b. health record

c prescription

5. An empty security of specified attributes is created.

Alternate Scenario:

1. A security with the given name already exists.

2. Security creation fails, error message is displayed.

**Use Case 9:** Delete security.

Primary Actor: Administrator

Pre-Condition: User logged in.

Main Scenario:

1. User selects the portfolio.

2. User initiates the “delete security” functionality.

3. System asks for the security name.

4. Security is deleted.

Alternate Scenario:

1. Security does not exist.

2. Deletion fails, error message is displayed.

.

**Use Case 10:** Seek an Appointment

Primary actor: Administrator

Pre-Condition: User logged in.

Main Scenario:

1 .user clicks into appointment page.

2 .System asks the patient name ,age ,place, and other details.

3 . User enter the details.

4.Click on submit button.

Alternate Scenario:

Appointment is not given.

**Use Case 11:** Disease

Primary actor: Doctor.

Pre-Condition: User logged in.

1. Patients have to fill the questions given by the system.

2. Based on the answers given by patient is matched with any one of the symptoms .

3. Then particular disease is displayed.

4. The doctor prescribes the medicines to the patient.

Alternate Scenario:

1. If the answers given by the patient is not understood, there is no prescription of medicines.

**3.2 Functional Requirements**:

1.May be placed on both the product and the process.

2.Derived from the environment in which the system is developed.

3.Application domain information.

4.Organisational considerations.

* 1. **Non-functional Requirements:**

1.Secure access of confidential data.

2.User friendly UI for easy interface, so that illiterate patients can also use it.

3.Execution qualities such as security which are observable at run time

4.Execution qualities like such as testability ,maintainability and scalability which are embedded in the architecture of the software system

**Optional Features:**

1. .Smart way for choosing generic and non generic medicine where available

2.It also include optiton to read dignose of a similar disease treated by the doctor using same medicine

**User interface:**

1.Professional look and feel

2.Use of XFORMS with all registration forms

3.Browser testing and support for IE, NW, Mozila firefox

4.Use of graphical tool like JASPER to show strategic data to admin

5.Reports exportable in XLS, PDF or any other any desirable formats

**Reports:**

1.Search the name, place, disease, periodic base reports .

2.Search the nearest government approved Clinic/Hospital and suggest it to the patient.

3.Website would be highly customizable and and flexible enough to easily deploy .

**Technologies to be used**

**J2EE** / Java Platform, Enterprise Edition or Java EE is a widely used platform for server programming in the Java programming language. The Java platform (Enterprise Edition) differs from the JavaStandard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

**JAVA** is an object-oriented programming language developed by Sun Microsystems a company best known for its high end UNIX workstations. Java language was designed to be small, simple, and portable across platforms, operating systems, both at the source and at the binary level, which means that Java programs (applet and application) can run on any machine that has the Java virtual machine (JVM) installed. Java technology is both a high-level object-oriented programming language and a platform. Java technology is based on the concept of a single Java virtual machine (JVM) -- a translator between the language and the underlying software and hardware. All implementations of the programming language must emulate the JVM, enabling Java programs to run on any system that has a version of the JVM.

**JSP**

Java Server Pages. It is used to create dynamic web content.

**UML**

Unified Modeling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document

**XML**

Extensible Markup Language is a text based format that let developers describe, deliver and exchange structured data between a range of applications to client for display and manipulation. XML provides you with a great deal of functionality and power. To top it off, it is a fairly simple and straightforward technology. Once you learn the basics, you can pick and choose what you want to learn. If you are a programmer, you can concentrate on programming and the various APIs. If not, you can concentrate on the various uses that do not involve programming. Either way, you are sure to find a place for XML in your work in today's world.

**HTTP**

Hypertext Transfer Protocol. It’s a service protocol.

**Tools to be used**

**Web server – WASCE**

Web Sphere Application Server Community Edition (from now on WASCE) is a free, certified Java EE 5 server for building and managing Java applications. It is IBM's supported distribution of Apache Geronimo that uses Tomcat for servlet container and Axis 2 for web services. Over 15 WASCE developers are committers in the Apache Geronimo project.

Web Sphere Application Server Community Edition. It is an application server that runs and supports J2EE and web service applications. IBM Web Sphere Application Server Community Edition is a free lightweight Java 2 Platform, Enterprise Edition (J2EE) application server built on Apache Geronimo, the open source application server project of the Apache Software Foundation. It harnesses the latest innovations from the open source community and provides a readily accessible and flexible foundation for building Java applications. Web Sphere Application Server CE helps you accelerate your development and deployment by offering technology that is quick to download and easy to use. The Community Edition includes Apache Tomcat and IBM Cloudscape. It also offers an enhanced Eclipse plug-in.

**Development tool –RAD**

IBM Rational Application Developer for WebSphere Software (RAD) is an integrated development environment (IDE), made by IBM's Rational Software division, for visually designing, constructing, testing, and deploying Web services, portals, and Java (J2EE) applications.

Rational Application Developer is a development tool that helps to design web pages and also helps to design the diagrams like ER, Database schema diagrams and to generate DDL. This powerful and extensible platform also makes it easy to develop large-scale applications. IBM Rational Application Developer for Web Sphere helps developers to quickly design, develop, analyze, test, profile, and deploy high-quality web, SOA, Java, J2EE, and portal applications.

**Database platform – DB2**

DB2 Database is the database management system that delivers a flexible and cost effective database platform to build robust on demand business applications and supports the J2EE and web services standards.

Database\_2 .A database management system that provides a flexible and efficient database platform to maintain records of students, teachers, admin and dm. DB2 offers information-leveraging solutions that are built on a portfolio of data management tools. There are more than 60 million DB2 users from 425,000 companies worldwide relying on IBM DB2 information management solutions. IBM is the only data management software provider with integrated solutions for database management, tools, content management, enterprise information integration, and business intelligence.

**DB2 Everyplace**® features a small-footprint relational database and high-performance data synchronization solution that enables enterprise applications and data to be securely extended to mobile devices, such as personal digital assistants (PDAs), smartphones, and other embedded mobile devices.

**IBM Cloudscape -- A complete RDBMS written in the Java language** provides a full-featured, robust, small-footprint database server that is simple to deploy and reduces the cost of embedded and web-based applications.

**DB2 Express-C** is a version of DB2 Universal Database Express Edition (DB2 Express) for the community. It is a no-charge data server used in development and deployment. Providing the same core data server features, development interfaces, and system limits as DB2 Express in a smaller package, DB2 Express-C offers a solid base to build and deploy all applications. DB2 Express-C is available for Linux® and Windows® running on 32- or 64-bit hardware.

**Design tool – Rational Software Modeler**

IBM Rational Software Modeler, (RSM) made by IBM's Rational Software division, is a Unified Modeling Language UML 2.0-based visual modeling and design tool. Rational Software Modelers built on the Eclipse open-source software framework and includes capabilities focused on visual modeling and model driven development (MDD) with the UML for creating resilient, thought-out applications and web services.

**Final deliverable includes:**

1. Online or offline help to above said users, Application development executive and developer
2. Application archive (.war/.ear) with source code
3. Database backup and DDL Script
4. Complete Source code