**E-HEALTHCARE**

**Abstract:**

The objective behind this project is to develop a web application, which helps patients to get prescriptions and suggestions from their preferred choice of doctor, without actually step in to the hospital or make an appointment with doctor.

**Description:**

The users using this ‘E-Healthcare’ system need not go to any clinic to get the prescriptions or advice from the doctor. They simply login to our application as a registered member to raise a request for prescription from their preferred choice of doctor. In order to become registered member of our application they have to sign up with registration number provided by the hospital at the time of their visit. Along with registered members guest users can also look out for information like health tips, information regarding common medical conditions and doctors availability. Our application also allows individual specialists and hospitals to register. As an admin of this application one can generate various statistical reports. Any user who signed in to our application able to securely signed out.

**List of Features / Modules:**

1. **Administrator Module:**
2. **Hospital Module**
3. **Doctor Module**
4. **Users Module**
   1. **Member user**
   2. **Registered user**
   3. **Guest user**

**Features description:**

**1. Administrator Module:** Administrator will manage the entire application and his features ranging from accepting new doctors and hospitals requests to add to our application. He can also modify, delete the profiles of doctors andhospitals. Broadcasting news and upcoming events to the subscribed users. Admin also has access to delete users.

**2. Hospital Module:** Hospitals showing interest to register our application can raise a request to sign up by providing some mandatory details like name, address and contact information and etc.

**3. Doctor Module:** Doctor who is ready to sign up our application must fill the form provided on the website and after approved by admin they become our beloved member of our ‘E-healthcare’ family.

**4. User Module:** We categorize the users in to three types.

1. **Guest user:**  They have access to browse information like doctors availability, regarding common medical conditions and health tips.
2. **Registered user:** Users who use signup form without registration number will become our registered users. As a registered user they have privilege to access some services like asking for prescriptions for common medical conditions by providing their bodily symptoms after paying the consultation fee.
3. **Member user:** Users who registered with health card number along with registration number given by our member hospitals have access to premium services like getting prescriptions from their desired doctors by providing bodily symptoms. They have any time access to old prescriptions. They can also shortlist doctors based on the reviews and ratings given by fellow members. They can opt for home delivery of medication from their choice of pharmacies.

**Technical Specifications:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Software Specifications** | | **Hardware basic requirements** | |
| **Language** | Java | **Processor** | > Pentium enabled PC’s |
| **Web technologies** | Servlets, JSP | **DRAM** | > 1 GB |
| **Front End language** | HTML, CSS | **HDD / SDD** | > 40 GB |
| **Scripting** | Java script |  |  |
| **Database** | Oracle 10g |  |  |
| **Browser** | Chrome, Mozilla, IE, Safari |  |  |
| **Operating System** | Windows, Mac, Linux, Unix |  |  |

**Future feature enhancements:**

1. Make appointments online
2. Web cam consultation.
3. Make medication and health care products online.
4. Home delivery of medications.

**Timeline & Milestones:**

|  |  |
| --- | --- |
| **Activity** | **Time Estimate (in Days)** |
| **Step1: Planning** |  |
| Activity1.1: Objective | **1** |
| Activity1.2: Functionality | **3** |
| Activity1.3: Risk Management | **3** |
| Activity1.4: Effort estimates | **2** |
| Activity1.5: Charts and documentation | **2** |
| **Step2: Requirement and Analysis** |  |
| Activity2.1: Functional & Non Functional requirements gathering. | **2** |
| Activity2.2: Finalizing the requirements | **2** |
| **Step3: Design** |  |
| Activity3.1: Architecture of the system | **3** |
| Activity3.2: UML design |  |
| Activity3.2.1: Class Diagram | **1** |
| Activity3.2.2: Sequence Diagram | **2** |
| Activity3.2.3: Use Case Diagram | **1** |
| **Step4: UI design and development** |  |
| Activity4.1: Home, About us, contact us pages. | **1** |
| Activity4.2: Sign up forms for various users, doctors and hospitals. | **1** |
| Activity4.3: Get prescription page. | **1** |
| Activity4.4: Doctors activity page. | **1** |
| Activity4.5: Access restrictions. | **1** |
| Activity4.6: Admin home page and activity pages. | **1** |
| Activity4.7: Payment pages. | **1** |
| Activity4.8: Feedback page. | **1** |
| Activity4.9: Other UI requirements if missed any | **2** |
| **Step5: Interface Integration** |  |
| Activity5.1: Developing servlets to handle requests. | **2** |
| Activity5.2: Database Connectivity. | **2** |
| Activity5.3: Bandwidth left to complete necessary integration work. | **2** |
| **Step6: Database Design** |  |
| Activity6.1: Database tables design and development. | **3** |
| Activity6.2: Querying DB | **2** |
| **Step7: Integration and Unit testing** |  |
| Activity7.1: Test cases writing. | **3** |
| Activity7.2: Executing those test cases. | **3** |
| **Step8: Deployment and testing** |  |
| Activity8.1: Hosting the project and testing | **1** |

**Pert Chart:**

Requirement Analysis

Planning

Integration

UI Design and Development

Testing

Design

Start

Finish

Deployment

Database Design

**PERT CHART**

**Note: Critical Path highlighted in ‘Blue’**

**Risk Management:**

**A. Risks identified:**

1. Technical limitations in implementing some requirements will force to change those requirements.
2. Unrealistic schedule, as we don’t have prior knowledge of working on similar type of projects.
3. Unfamiliar areas of project take more time than expected to design and implement.
4. The features requested might be beyond what the development team can deliver in the time available.
5. Specific areas of the product may consume more time than expected.
6. Quality may get affected because of the upcoming deadlines.
7. If documentation consumes considerable time, project may cross the deadline.
8. Conflicts among team members result in poor communication and may end up critical development work being performed by few developers.
9. Everyone doesn’t have same level of expertise across all the technologies.
10. Lack of industry level experience in the team.

**B. Plan for monitoring risks:**

1. Plan a head and do enough research regarding the stream we are working on.

2. Arrange frequent meetings and reviews and make use of Mom’s.

3. A) Hire new resources.

B) Push the requirement to the next build / iteration to develop.

C) Over time – This may effect quality of the product.

Note: As options A & B won’t be possible in our team. So, we left with only option C.

4. Plan ahead and build expertise in technical as well as functional knowledge.

5. Proper scheduling helps us avoiding pitfalls and makes the product development easy without bottlenecks.

6. Do documentation in parallel to development.

7. Equal distribution of work might reduce conflicts. In addition to that, arranging team hangouts, frequent get to together and stress relieving activities are some of the helpful factors in reducing team conflicts.

8. Knowledge transfers and helping each other in their tasks will bring everyone on the same pace.

9. Upgrade you self and put more man hours to get sufficient expertise.

10. Good knowledge on various software technologies and tools and their limitations ahead of starting the project will help overcome requirement changes due to technical difficulties.

**Note:** Risks identified in section A and plan to handle those in B are mapped in order.

**C. Contingency plans for these risks:**

1. Considering and implementing modularity in the product development will guarantee Minimum viable product.
2. Maintaining less dependencies among interfaces will avoid delay in milestone will not steep in to other milestones.
3. Implementing best modeling and documentation techniques like ‘Agile’ model to make a project best fit in a fast changing environment.

These are some of the principles in software life cycle to handle emergencies.

**About Team:**

**Progress on this deliverable:**

Functional requirement, Risk Management, documentation: Surya & Manasa

Project planning, Timeline, PERT chart, Mom: Divya & Sriram

**Team structure:** As a ‘Mixed control team’ one-person act as a authority or team lead. But, members along with team lead work democratically for discussions.

**Team members & Roles:**

|  |  |  |
| --- | --- | --- |
| **Member** | **Role** | **Work Planning** |
| **Divya Sharavani Boddu** | **Member** | **Doctor Module** |
| **Manasa Channagiri** | **Member** | **Users Module** |
| **Sairam** | **Member** | **Hospital Module and part of user module** |
| **Surya Pasumarthi** | **Team Lead** | **Admin Module** |

**Project repository checkout and update policies:**

We are using GIT as a version control system. A public URL shared across the team members and Instructor to fetch the project documents. Only the members of the team can commit to GIT public folder and the merge activity will be monitored by team lead to avoid any conflicts.