DELIVERABLE -2

# GROUP DETAILS:

**Project Title**: Hospital Management System

**Team Name**: Unt Ignitors

**LIST OF TEAM MEMBERS:**

|  |  |
| --- | --- |
| **Members** | **Student ID** |
| DHEERAJ REDDY AGUTHU | 11555619 |
| ABHAY ARORA | 11549231 |
| RAVI TEJA BALAJI | 11514148 |
| PRAVEEN NAKKA | 11532917 |
| CHARISHMA NAGA SAI SARADA BALUSU | 11525468 |
| GEETHA KRISHNA DODDA | 11558520 |
| SRIKANTH GOPI | 11514330 |
| MEGHANA JUNNUTULA | 11539646 |

Table of Contents

1. Overall Specifications
   1. Purpose
   2. Project Description
   3. System Structure and Components
   4. Diagrams
   5. Operating Environment
2. Requirement Specification
   1. Functional Requirements
   2. Non-Functional Requirements
3. External Interfaces
   1. User Interfaces
   2. Hardware Interfaces
   3. Software Interfaces
4. Development Phases
   1. Development phase 1
   2. Development phase 2
   3. Development phase 3
5. Member Contributions
6. Minutes of Meeting
7. **Overall Specification**
   1. **Purpose:**

The main purpose of our system is to allow patients to book appointments with doctors over the internet, and we keep track of their customized data in the device, In addition to the above modules, our project comprises administration, tracking, reporting, alerting, and training programs. Only authorized users can view the system, and only authorized users are permitted to update data and manage bookings.

* 1. **Project Description:**

The project's objective is to develop a website that allows patients to schedule online appointments with doctors based on their availability and specialties. While there are a few existing systems that provide similar functionality, but we intend to create a system that is much more advanced. This can be done by using an easy-to-use interface and scheduling the appointments. We've added several new features to our site, including OP, which allows staff members to enroll patient information in an emergency

* The proposed application will be used by the four types of users (i.e., Admin, Doctors, Patients, Office Staff).
* Admin is the root user, who has the access to perform different kind of operations that includes managing master data of doctors, patients, and other medical workers, all these information is stored in the database using Web API database connectivity.
* Admin can also perform actions like addition of doctors along with their specializations, managing branches which are newly constructed or existing branches and update the details about the room availability and medicine availability details in the database
* Doctor has the access to view the list of patients, manage appointments, provide second opinion on surgeries/operations and keeping track of patient information such as treatment and diagnosis reports, as well as medications.
* Patients search for and contact doctors by entering their office hours and location into the system and schedule an appointment.
* Office staff plays a major role in the HMS, as they are the first person in the hospital who takes the basic information about the patient who joins as an emergency case and store the data in the system database.
  1. **System Structure and Components**

**System Structure**

Diagram

Description automatically generated

**Components**

**1. Admin:**

Admin will have full control over the system and will be able to make any modifications that are required.

**2. Doctor:**

Doctor has the access to view the patients, Medicines availability, he can schedule an appointment and give prescription for the patient who is taking Treatment and suggest if any surgery is required.

**3. Patient:**

Patients have access to view the Pending treatments which has to be taken by him, he can able to see the how many medications or appointments he need to take to complete his course.

**4. Office Staff:**

Office Staff have access to record the details of the patients, who have admitted in the hospital as an emergency. They also provide a mandatory first aid when the patient arrives at the hospital

* 1. **Diagram**

Diagram

Description automatically generated

Fig 1.1

* 1. **Operating Environment**

|  |  |
| --- | --- |
| Frontend | HTML, CSS, JavaScript, Angular |
| Backend | C#, .Net and ADO.Net |
| Database | MySQL |
| Run time environment | Visual Studio 2019, MS VS Code |
| Web server | WebAPI |

**2. Requirement Specification**

**2.1 Functional Requirements**

**Authentication Requirements:**

* The system allows admin to register users based on role specifications of each user.
* The system must allow users to log in to their accounts by using their username and password.

**Authorization Requirements:**

* Only Admin has the authority to create, edit and delete for every option provided in UI.

**Home Page:**

* In the Home page user can be able to see the registration and login screens
* In this page we can see the set of tabs like Home, Services, About, Contact, Register and login tabs.
* Each tab will perform different functionalities.

**Registration:**

* If the user is the new one user should be able to register with set of details like

First name, Last name, Email, Dob, password, Confirm Password, Username and User Role.

* User role contains set of roles in the drop down like Admin, Doctor, Patient, Office Staff.
* If it is the existing user, he can directly redirect to login page by clicking on the hyperlink provided in the registration Page.

**Login:**

* Login Page contains Mandatory fields like Username and Password which are used to login in their respective roles.
* If the user is new one, he can be redirected to the registration page by clicking on the hyper link provided in the login page.

**Admin Dashboard:**

* The admin user, often known as the root user, can execute a variety of tasks, including controlling master data for doctors, patients, and other medical personnel.
* Admins can also undertake tasks such as adding doctors and their specialty, administering newly constructed or existing branches, and updating information on room and medicine availability.
* Admin has access to all information in the system and can create, edit, and delete fields in the user interface. We have the flexibility to add new items to each list.
* No unauthorized individual can handle any operation without a username and password.

**Doctor:**

* The doctor can see the patients, see what medicines are available, schedule an appointment, and write a prescription for the patient who is undergoing treatment.
* He can see a list of scheduled appointments for that day and add medicine IDs to the interface.
* The doctor has access to the patient list, manages treatments, and offers opinions on surgeries and operations.

**Patient:**

* The patient can see the pending therapies he has had, as well as how many drugs, pending appointments, and treatments he has had. The patient has access to all this information.
* Patients have the option of scheduling appointments with a variety of specialists. They can also add or amend their personal health data.

**OP:**

* We've added new feature to our site by including OP, which allows staff members to enroll patient information in an emergency

**2.2 Non-Functional Requirements**

**Performance**

Performance is the rate, volume or number of transactions or units of work a system/application is capable of, measured over time. An understanding of the immediate and projected performance needs for a business process flow is critical to ensure that they are taken into consideration in the design.

|  |  |  |
| --- | --- | --- |
| NFR No | Functional Area | Information |
| NFR1 | Approximate total number of users of current/proposed system. | 800-1000 |
| NFR2 | Page Response Time | 3 to 7 Seconds |

**Usability / Supportability**

|  |  |
| --- | --- |
| NFR No | Information |
| NFR3 | New system will support latest version of Chrome browser. |
| NFR4 | Appropriate error message will be displayed when the application is not functioning. |
| NFR5 | All dates would be presented in the format MM/DD/YYYY |
| NFR6 | All drop down lists will be searchable as user types |

**Security**

As part of the Data Security Policy, it is required that all data used within a project is assigned a classification for confidentiality, integrity, and availability.

|  |  |  |
| --- | --- | --- |
| NFR No | Functional Area | Information |
| NFR7 | API | API will be secure using JWT Token |
| NFR8 | Authorization | Only authenticated user can access the system’s feature/pages/menu based on the authorization mapped in the new system. |
| NFR9 | Environment | This will be a web-based system and can be accessed through given URL and valid authentication |
| NFR10 | Data Breach | The developer's inner circle should be given administrator login information.  Other security methods include encrypting databases to prevent SQL injection attacks, with hashing algorithms playing a key role in this situation |

**Accessibility**

|  |  |  |
| --- | --- | --- |
| NFR No | Functional Area | Information |
| NFR11 | Usability Requirements | Screen resolution will support desktop only. Grids will have search bar, filter option (wherever it will be required) |
| NFR12 | Optimized for Screens Size | Optimized for responsive desktop layout from resolution 1920x1200 to 800x600. |

**Integration**

|  |  |  |
| --- | --- | --- |
| NFR No | Functional Area | Information |
| NFR13 | Email/Push Notification | All email notifications will be sent using SMTP |

**Maintainability:**

NFR14 - The system offers efficiency for data backup and check for errors in the patient's database.

**3.External Interface Requirements**

**3.1 User Interfaces**

We're endeavoring to make a website with the best UI features and the most easy and convenient Way to the user. So, a client/User with no earlier information may essentially move around and complete the necessary activities.

1. Here, the first page is the homepage which has the fields of signup, login and other features.

2. Once a user clicks on signup button, the page will redirect to the registration page where user can register themselves.

3. After signing up, the user will first move back to the homepage and in homepage user can see many options like register, login. These options are embedded as buttons, on clicking on every option user, is redirected to respective pages.

4. Users can log in to their page and undertake operations by clicking the "Login" button.

5. By selecting the profile button, the user can edit his or her profile information. We offered an edit option to carry out this activity.

6. On the appointments page, the user can see a list of appointments.

1. Make appointments with specialists as needed, as well as add and update personal health information.

8. This application, which can be accessed by Admin, deals with the specialization information of doctors and other staff members, and admin has the authority to create, delete, and edit options for every field in the UI.

9. Admin has access to add doctor specializations, update room availability details, new and existing hospital branch details, and medicine availability details.

1. Doctors will be able to view patients record details and manage appointments, and medications.

**3.2 Hardware Interfaces**

**Processor:** Intel i3 and above

**RAM:** 8gb ram

**Space on disk:** 150 GB

**3.3 Software Interfaces**

**Web Server:** Web API

**Front End:** HTML, CSS, Java Script.

**Database:** Sql Server

**UI:** Mvc

**Backend:** C#.Net and ADO.Net

**4. Development Phases**

**4.1 Development Phase 1:**

* + - Home Page
    - Login Page Creation
    - Registration Page
    - Services
    - Testing

**4.2 Development Phase 2:**

* + - Admin Dashboard creation.
    - Admin can create account for new users.
    - Doctor Dashboard creation.
    - If another user has already reserved a time slot for the patient, he or she will be unable to schedule an appointment.
    - Another user, such as a doctor, can view the list of patients, manage appointments, and provide a second opinion on surgeries/operations.
    - Testing

**4.3 Development Phase 3:**

* + - Patient page creation
    - Patients can make appointments with various specialists. They can also add or update personal health information.
    - Page creation for Staff members.
    - Implementing new features to our site, including OP, which allows staff members to enroll patient information in an emergency.
    - Bug Fixing
    - Testing

**5. Member Contributions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall Contribution (%)** | **Contribution for Document** |
| DHEERAJ REDDY AGUTHU | Backend Developer, Database | 12.5 | Overall Project Description |
| ABHAY ARORA | Front End Developer,  SCRUM Master,  Project Coordinator | 12.5 | Non-Functional Requirements and updating the documents in GitHub. |
| RAVI TEJA BALAJI | UI Developer, Database | 12.5 | Development phases and Member Contribution tables. |
| PRAVEEN NAKKA | UI Developer, Database | 12.5 | Functional Requirements |
| CHARISHMA NAGA SAI SARADA BALUSU | Backend Developer, Documentation | 12.5 | External Interfaces |
| GEETHA KRISHNA DODDA | UI Developer, Database | 12.5 | External Interfaces |
| SRIKANTH GOPI | UI Developer, Database | 12.5 | Functional Requirements |
| MEGHANA JUNNUTULA | Backend Developer,  UI Developer | 12.5 | Product Perspective |

**6. Minutes of Meeting**

Deliverable 2 Discussion on 24th February 2022

* + - **Attendees:**

|  |  |
| --- | --- |
| **Members** | **Student ID** |
| DHEERAJ REDDY AGUTHU | 11555619 |
| ABHAY ARORA | 11549231 |
| RAVI TEJA BALAJI | 11514148 |
| PRAVEEN NAKKA | 11532917 |
| CHARISHMA NAGA SAI SARADA BALUSU | 11525468 |
| GEETHA KRISHNA DODDA | 11558520 |
| SRIKANTH GOPI | 11514330 |