DELIVERABLE -4

# 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. Requirements for phase 2 ………………………………………………………3
   1. User Interface for Admin Dashboard ……………………………………….3
   2. User Interface for Doctor Dashboard …………………………………….… 3
2. UML Diagrams …………………………………………………………………4
   1. Class Diagram ………………………………………………………………4
   2. Sequence Diagram ………………………………………………………….5
   3. Use Case Diagram ………………………………………………………….6
3. Test Cases for phase 2 ………………………………………………………….7
4. User Manual …………………………………………………………………….18
   1. Installations …………………………………………………………………18
   2. Screenshots ………………………………………………………………....19
5. Instructions to run the program ………………………………………………… 20
6. Peer review feedback ……………………………………………………………29
7. Accomplishments ……………………………………………………………….30
8. Member Contribution table ………………………………………………………32
9. **Requirements for phase 2**

Implementation plan for Hospital Management System (HMS) Phase 2 includes developing user interfaces for the admin dashboard and doctor dashboard.

* 1. **The user interface for Admin Dashboard**
* Admin is a critical role in the hospital management system, as he manages the hospital's overall master data.
* When the user logs in as an Admin with administrative credentials, the User can view the various types of tiles and responsibilities that he can perform in HMS.
* Adding/deleting hospital branch details and listing the number of branches for that hospital is one of the main functions or responsibilities of the Admin.
* Admin also enters the doctor's information and specializations into the database and maintains their information such as skills and experience, work schedules, treatments provided by the doctors, and so on.
* Also, when a new patient is admitted to the hospital or visits the hospital, the admin manages the patient details and enters them into the system.
* Admin is also in charge of the medication and treatment details. The admin user manages the medicine details, such as adding available medications and any new medications recommended by doctors.
* In addition to the above duties, he manages the hospital's room information and provides details about available rooms and operative rooms, which can assist hospital management in effectively treating patients.

**1.2. User interface for Doctor Dashboard**

* The doctor is the most important component of the hospital management system. When a user logs in as a doctor and enters doctor credentials. In HMS, the doctor can see the various types of tiles and responsibilities that he can perform.
* A doctor dashboard is created with various tiles so that the doctor can easily access and treat patients.
* In the Doctor dashboard, the doctor can see a list of patients under him as well as the treatment provided to that specific patient.
* The doctor can schedule an appointment with the patient and administer the medication.
* The doctor can also view the pending appointments for the patient's treatment.
* The doctor can schedule the appointment for different medicines.

1. **UML Diagram**
   1. **Class Diagram**

Diagram

Description automatically generated

*Fig 1 Class Diagram*

**2.2 Sequence Diagram**

The sequence diagram for the Hospital Management System is shown below. Patients, doctors, admin, and hospital staff can involve the Hospital Management System by logging in.

* If the role is doctor, he can view the list of patients, prescribe medications for the patient, provide a second opinion on surgeries, and keeps tracking patient information like treatment, diagnosis reports, and medication.
* If the role is admin, he can manage the master details of the patient, and doctor. Admin can add and update doctor details, medicine details, room availability details, and new and existing branch details.

Diagram

Description automatically generated

*Fig 2 Sequence Diagram*

* 1. **Use Case diagram**

Diagram

Description automatically generated

*Fig 3 Use case Diagram*

1. **Test Cases for Phase 2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.**  **No** | | **Requirement** | | **Test Priority** | **Test Steps & pre-**  **conditions** | **Expected Result** | **Obtained Result** | **Success/ Failure** | **Name of Tester** |
| 1 | | View Doctor Dashboard | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen | The system will show the Doctor Dashboard with data from various sections (Patient, Doctors, Appointments, Medicines) | Doctor Dashboard is displayed | Success | Abhay |
| 2 | | Side Menu in Doctor Dashboard | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen | The system will show the following sections on Side Nav Menu  Appointment, Treatment, Appointment Medicine | The side Menu section is displayed with Appointment, Treatment, Appointment Medicine Sections. | Success | Dheeraj |
| 3 | | List Appointment Screen | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen 3. Click on the Appointment menu from Side Nav 4. The system will navigate to List Appointment Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Appointment Screen | Success | Abhay |
| 4 | | Add Appointment Screen | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen 3. Click on the Appointment menu from Side Nav 4. The system will navigate to List Appointment Screen 5. Click on Add Appointment Button | The system will display a set of form fields with the button Create Appointment to submit the data | The system showed Form Fields to create a new appointment | Success | Meghana |
| 5 | | Edit Appointment Screen | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen 3. Click on the Appointment menu from Side Nav 4. The system will navigate to List Appointment Screen 5. Click on the Edit Appointment Button from the grid | The system will display a set of form fields with the button Update Appointment to update the appointment data. | The system showed Form Fields to update existing appointment | Success | Abhay |
| 6 | | Delete Appointment | | High | 1. User (Doctor) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen 3. Click on the Appointment menu from Side Nav 4. The system will navigate to List Appointment Screen 5. Click on Delete Appointment Button from the grid | The system will ask for delete record confirmation. On Yes, the record will be deleted from the grid.  On No, the system remains in the same state | On clicking the delete button system asked for confirmation and on clicking yes, records get deleted and the grid refreshes with updated data from the server | Success | Abhay |
| 7 | | User will be notified of successfully saved appointment record | |  |  | Alert Box/Toast will be shown to the user with a success message after creating or updating the Appointment action | Toast messages appear on the screen with appropriate success messages after creating and updating the action |  | Abhay |
| 8 | | The user will be notified of the successful deletion of the Appointment Record | |  |  | Alert Box/Toast will be shown to the user with the success message. | Toast messages appear on the screen with an appropriate success message after the delete action |  | Abhay |
| 9 | | List of Doctors | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Doctor in the menu from Side Nav 4. The system will navigate to List of Doctors Screen | Previously added records will be shown in the grid | The system is shown Doctor Branch records in a grid on the List of Doctors Screen | Success | Gopi | |
| 10 | | Add Doctor Screen | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Doctor menu from Side Nav 4. The system will navigate to List of Doctors Screen 5. Click on Add Doctor Button | The system will display a set of form fields with the button Create Doctor to submit the data. | The system showed Form Fields to create a new Doctor. | Success | Gopi | |
| 11 | | Edit Doctor details | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Doctor menu from Side Nav 4. The system will navigate to List of Doctor Screen 5. Click on the Edit Doctor icon from the grid | The system will display a set of form fields with the button Update Doctor to update the Doctor data. | The system showed Form Fields to update existing Doctors in the List of Doctors. | Success | Gopi | |
| 12 | | Delete doctor details | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Doctor Dashboard Screen 3. Click on the Doctor menu from Side Nav 4. The system will navigate to List of Doctor Screen 5. Click on Delete Doctor Button from the grid | The system will ask for delete record confirmation. On Yes, the record will be deleted from the grid.  On No, the system remains in the same state | On clicking the delete button system asked for confirmation and on clicking yes, records get deleted and the grid refreshes with updated data from the server | Success | Gopi | |
| 13 | | User will be notified of successfully saved Doctor record |  |  | Alert Box/Toast will be shown to the user with a success message after creating or updating the Doctor action | Toast messages appear on the screen with appropriate success messages after creating and updating the action |  | Gopi | |
| 14 | | The user will be notified of the successful deletion of the Doctor Record |  |  | Alert Box/Toast will be shown to the user with the success message. | Toast messages appear on the screen with an appropriate success message after the delete action |  | Gopi | |
| 15 | | View Admin Dashboard | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen | The system will show the Admin Dashboard with data from various sections (Patient, Doctors, Specializations, Medicines) | Admin Dashboard is displayed | Success | Gopi | |
| 16 | | Side Menu in Admin Dashboard | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen | The system will show the following sections on Side Nav Menu  Doctor, Patient, Specializations, Medicine | The side Menu section is displayed with Doctor, Patient, Specialization, Medicine, Treatment Sections. | Success | Gopi | |
| 17 | | List of Hospital Branch | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Hospital Branch in the menu from Side Nav 4. The system will navigate to List of Hospital Branch Screen | Previously added records will be shown in the grid | The system is showed Hospital Branch records in a grid on the List of Hospital Branch Screen | Success | Gopi | |
| 18 | | Add Hospital Branch Screen | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Hospital Branch menu from Side Nav 4. The system will navigate to List of Hospital Branch Screen 5. Click on Add Hospital Branch Button | The system will display a set of form fields with the button Create Hospital Branch to submit the data. | The system showed Form Fields to create a new hospital branch | Success | Gopi | |
| 19 | | Edit Hospital Branch Screen | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Hospital Branch menu from Side Nav 4. The system will navigate to List of Hospital Branch Screen 5. Click on the Edit Hospital Branch Button from the grid | The system will display a set of form fields with the button Update Hospital Branch to update the Hospital Branch data. | The system showed Form Fields to update the existing Hospital Branch | Success | Gopi | |
| 20 | | Delete Hospital Branch | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Hospital Branch Dashboard Screen 3. Click on the Hospital Branch menu from Side Nav 4. The system will navigate to List of Hospital Branch Screen 5. Click on Delete Hospital Branch Button from the grid | The system will ask for delete record confirmation. On Yes, the record will be deleted from the grid.  On No, the system remains in the same state | On clicking the delete button system asked for confirmation and on clicking yes, records get deleted and the grid refreshes with updated data from the server | Success | Gopi | |
|  | |  |  |  |  |  |  |  | |
| 21 | | List of Patient | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Patient in the menu from Side Nav 4. The system will navigate to List of Patients Screen | Previously added records will be shown in the grid | The system is showed Patient records in a grid on the List of Patients Screen | Success | Gopi | |
| 22 | | Add Patient Screen | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Patient menu from Side Nav 4. The system will navigate to List of Patients Screen 5. Click on Add Patient Button | The system will display a set of form fields with the button Create Patient to submit the data. | The system showed Form Fields to create a new patient. | Success | Gopi | |
| 23 | | Edit Patient Screen | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Patient menu from Side Nav 4. The system will navigate to List of Patients Screen 5. Click on the Edit Patient Button from the grid | The system will display a set of form fields with the button Update Patient to update the Patient data. | The system showed Form Fields to update existing Patient | Success | Gopi | |
| 24 | | Delete Patient | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Hospital Branch Dashboard Screen 3. Click on the Patient menu from Side Nav 4. The system will navigate to the List of Patient Screen 5. Click on Delete Patient Button from the grid | The system will ask for delete record confirmation. On Yes, the record will be deleted from the grid.  On No, the system remains in the same state | On clicking the delete button system asked for confirmation and on clicking yes, records get deleted and the grid refreshes with updated data from the server | Success | Gopi | |
| 25 | | User will be notified of successfully saved Patient record |  |  | Alert Box/Toast will be shown to the user with a success message after creating or updating the Patient action | Toast messages appear on the screen with appropriate success messages after creating and updating the action |  | Gopi | |
| 26 | | The user will be notified of the successful deletion of the Patient Record |  |  | Alert Box/Toast will be shown to the user with the success message. | Toast messages appear on the screen with an appropriate success message after the delete action |  | Gopi | |
| 27 | | List of Room Type | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Room Type in the menu from Side Nav 4. The system will navigate to List of Room Type Screen | Previously added records will be shown in the grid | The system is showed Room Type records in a grid on the List of Room Types Screen | Success | Gopi | |
| 28 | | List of Specializations | High | 1. User (Admin) logins to the system 2. On successful login, the system will navigate to Admin Dashboard Screen 3. Click on the Specializations in the menu from Side Nav 4. The system will navigate to the List of Specializations Screen | Previously added records will be shown in the grid | The system is showed Doctor’s specializations records in a grid on the List of Specializations Screen | Success | Gopi | |

1. **User Manual**

**4.1 Installation of Required software**

To run the Hospital management system, we require a certain set of software and IDEs to run the code and provide an output.

Graphical user interface, text, application

Description automatically generated

*Fig 4*

* Please install the “Node.js” software to run the script, from the following link
* [**https://nodejs.org/en/download/**](https://nodejs.org/en/download/)

**Step 2: Install the MySQL Workbench**

Graphical user interface, text

Description automatically generated

*Fig 5*

* Please install the “MySql” workbench to connect the database to the UI through WebAPI**,** [**https://www.mysql.com/downloads/**](https://www.mysql.com/downloads/)

**Step 3: Install Visual studio**

Timeline

Description automatically generated with medium confidence

*Fig 6*

* Please install Visual Studio to connect the database and WebAPI using the below link [**https://visualstudio.microsoft.com/downloads/**](https://visualstudio.microsoft.com/downloads/)

**Screen Usage**

a. **How to access application**: Access the application using browser URL <IP ADDRESS>:<PORT>/index.HTML

b. **How to Register**: Admin, Doctor, and Patient can register by clicking on Register as a new user icon in the navigation menu to Register to the application. If the user is already registered, then they can proceed to log in. If any of the end-user doesn’t have an account, they must enter the mandatory fields of Username, Email, First Name, Last Name, User Role, Password and can add Phone Number as an optional field and can click on the Register button. Customers will be redirected to the confirmation page screen on successful registration.

c. **How to login**: Admin, Doctor, and Patient can click on the login icon on the navigation menu to login into the application. The end-user can log in using the username and password they registered with the application and access the modules attached to their user role.

1. **Instructions to compile and run the program**

**Step 1**

* Download the code repository from GitHub. To download follow the below steps
* Create a new folder HMS\_code\_base and open the folder in “CMD” or Terminal.
* Run command: **git clone** [**https://github.com/abhayarora23UNT/UntSeProjects2022.git**](https://github.com/abhayarora23UNT/UntSeProjects2022.git)
* This will clone all the project code into the created folder.

**Step 2**

* To compile the program in a local machine you should have node and angular installed.
* Download and install node js from <https://nodejs.org/en/download/>
* Open Cmd/terminal and run the command npm install –g @angular/cli
* User node -v and ng –version commands to check whether the above software is installed.
* Go to path “HMS\_code\_base /UntSeProjects2022/HMS/Source Code/Front\_End/HmsApp/” open Cmd/Terminal.
* And run the command npm install. This will install all the required dependencies to run the application.
* After successful installation, run command ng serve to run the application in the local machine. This will run your application in localhost:4203 (port can be changed from angular. json)

Text

Description automatically generated

*Fig 7*

Text

Description automatically generated

*Fig 8*

**Step 3**

Please open the browser and search as **“localhost:4203”** to access the application

Graphical user interface, application

Description automatically generated

*Fig 9*

Users are required to enter their credentials to log in to the website. To login into the system. Users are required to register first in the system by providing the basic information given in the below screenshot

Graphical user interface, application

Description automatically generated

*Fig 10*

Users are required to register into the system by providing personal details such as username, name, last name, user role, phone no, and password.

**Doctor Dashboard**

In this phase, we, have developed the doctor dashboard and admin dashboard.

* If the user of Type (Doctor), logins into the system
  + On Success, System will show Doctor Dashboard Screen

A screenshot of a computer

Description automatically generated with medium confidence

*Fig 11*

* The user clicked on Appointment Menu from Side Nav Section.
  + The system will navigate to List Appointment Screen

A picture containing graphical user interface

Description automatically generated

*Fig 12*

* User clicks on Add Appointment Button
  + The system will navigate to Add Appointment Screen

Graphical user interface, application, Teams

Description automatically generated

*Fig 13*

* The user clicks on Delete Button
  + The system will show the confirmation to delete the record

Graphical user interface

Description automatically generated with medium confidence

*Fig 14*

**Admin Dashboard:**

When the user logs in as an administrator, it will redirect to admin dashboard. Where he can see number of tiles which are accessible for the admin

Table

Description automatically generated

*Fig 15*

* Admin can navigate to different tasks by clicking on the side nav bar where we can see the different type of navigations.
* Here, admin can view the details of the doctor by clicking on the doctor on the left side dashboard.

Graphical user interface, text

Description automatically generated

*Fig 16*

Admin can also add/edit/list the doctor details and add the doctor details by providing the basic details.

Graphical user interface, application

Description automatically generated

*Fig 17*

Similarly, admin can list/add/edit the details of the hospital branch and modify the details of the hospital branch

Graphical user interface, application

Description automatically generated

*Fig 18*

1. **Peer Review Feedback**

Meeting on Friday, 04/1 12:30 PM – 1:30 PM

**Purpose**: Peer Review with Eagle verse

**Participants**: Arun Sai Kumar Gutala, Pallavi Tulluri, Achyuth Dondapati, Pavani Mangugari, Neha Gummalla, Somasekar Bathina Suresh, ViswaTeja Ravipati and Milind Cherukuri and UntIgnitors Team Members.

We discussed the code inspection document and the phase 2 development

**Suggestions were given by the partner group:**

* + To add the comments for the source code about the functionalities
  + The length of the overall code inspection document can be reduced (as per the submission guidelines)
  + Comments should be added in angular (ts code) as it’s not present in some places
  + SQL code snippets can be avoided in inspection documents.

**Accepted:**

* + Accepted to include the comments in the source code and to reduce the overall document length

**Rejected:**

* + To remove the SQL code snippets from the document. Rejected this point, as the Database schema and DB Setup were already included in the phase 1 development plan

**Discussion with Eagle verse:**

* + Proper comments should be present in the spring-boot java API code
  + Code snippet of style/CSS files should be included in Code Inspection Document

1. **Accomplishments/Challenges**

As per the proposed phase-2 project plan, we accomplished the development and testing of the following screens

1. Doctor Module
   1. Dashboard, Appointments, Treatment.
2. Admin
   1. Hospital Branches, Doctors, Patients, Medicines, Operative rooms.
3. SQL Scripts of stored procedures were run to manage overall user data.
4. Web API was implemented to perform crud operations on the database.
5. Design/Guidelines Achievement
   * We followed the Trello Kanban Template to manage our development and other requirements/design tasks.
   * From the development point of view, we adhere to coding guidelines required for development in Angular, C# Framework/Technologies.
   * The initial architecture for Front End and Backend was built in .net web API and angular code. To follow various design patterns, reusable components are created to avoid code redundancy, which in turn improves the overall performance of the application.

Challenges

1. Few team members were not well skilled in front-end development in Angular Framework.
   1. To overcome that, some members do peer code reviews and provide internal training to other team members.
2. Deployment of Application on IIS (Internet Information Services)

Things to Improve

1. For future phases, more focus will be given to the integration of individual sub-modules.

**8. Member Contribution Table**

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall Contribution (%)** |
| DHEERAJ REDDY AGUTHU | * Deliverable 4 Documentation   + - 1. UML (Class Diagram)       2. Test Cases       3. User Manual * Backend Database creation * Testing | 12.5 |
| ABHAY ARORA | * Updated the Minutes of Meeting in Repo * Deliverable 4 Documentation  1. Code Inspection Doc 2. Test Cases 3. User Manual  * Front End Development (Initial Architecture, Doctor Dashboard, Appointment (List, Edit, Delete) * Deployment and Unit Testing * Managing Git and Trello Board * Deliverable 4 Task Management | 12.5 |
| RAVI TEJA BALAJI | * Development phases and Member Contribution tables. * Deliverable 4 Documentation  1. UML (Class Diagram) 2. Peer Review Section Feedback 3. Test Cases | 12.5 |
| PRAVEEN NAKKA | * Deliverable 4 Documentation  1. Requirements 2. Peer Review Section Feedback  * Front End Development (Admin Module) | 12.5 |
| CHARISHMA NAGA SAI SARADA BALUSU | * Deliverable 4 Documentation  1. Sequence Diagrams  * Testing | 12.5 |
| GEETHA KRISHNA DODDA | * Deliverable 4 Documentation  1. Sequence Diagrams  * Testing | 12.5 |
| SRIKANTH GOPI | * Deliverable 4 Documentation  1. Requirements 2. UML (Use Case Diagram) 3. Test Cases | 12.5 |
| MEGHANA JUNNUTULA | * Deliverable 4 Documentation  1. Requirements 2. UML (Use Case Diagram) 3. User Manual 4. Code Inspection  * API Initial Structure + API implementation (Admin and Doctor Module) * Front End Development (Doctor-Treatment Screen) * DB Schema and Records changes | 12.5 |

**9. Minutes of Meeting**

Minutes of the meeting are updated below path in the project repository

<https://github.com/abhayarora23UNT/UntSeProjects2022/tree/main/HMS/MOM>