DELIVERABLE – 5

# 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 3 ………………………………………………………3
   1. User Interface for Patient Dashboard ……………………………………….3
   2. User Interface for Staff Dashboard …………………………………….… 3
2. UML Diagrams …………………………………………………………………4
   1. Class Diagram ………………………………………………………………4
   2. Sequence Diagram ………………………………………………………….5
   3. Use Case Diagram ………………………………………………………….6
3. Testing ………………………………………………………………………….7
   1. Unit Testing for phase 3…………………………………………………….7
   2. System Testing…………………………………………………………… 15
4. User Manual …………………………………………………………………….18
   1. Installations …………………………………………………………………18
   2. Screenshots ………………………………………………………………....19
5. Instructions to run the program ………………………………………………… 20
6. Features, limitations, and feedback from peer review sessions.………………… 29
7. Accomplishments ……………………………………………………………….30
8. Member Contribution table ………………………………………………………32
9. Minutes of Meeting ………………………………………………………………36
10. **Requirements for phase 3**

Implementation plan for Hospital Management System (HMS) Phase 3 includes developing user interfaces for the Patient dashboard and Hospital staff dashboard.

* 1. **The user interface for Patient Dashboard**
* The patient is a critical role in the hospital management system.
* When the user logs in as a Patient with patient credentials, the User can view the various types of tiles and responsibilities that he can perform/View in HMS.
* The patient has the privilege to select the doctor and schedule an appointment with them by checking the dates of availability.
* Patients can view the list of doctors and their contact details so that they can contact them in case of emergency.
* Patients can view the list of medicines available and select the medicines which are prescribed by the respective doctor.

**1.2. User interface for Hospital Staff Dashboard**

* The Hospital staff is the most important component of the hospital management system.
* When the patient admits the hospital with an injury, or any emergency, staff will have access to update the patient information in the records
* A hospital staff dashboard is created with various tiles so that the staff can easily access and treat patients.
* The doctor can schedule an appointment with the patient and administer the medication.
* The Hospital Staff can also view the pending appointments for the patient's treatment.
* The Hospital Staff can also view the list of doctors available in the hospital.

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 Patient, the patient logs in to the system and he has different tiles in the dashboard, where he can add the details and book an appointment with the doctor.
* If the role is hospital staff, he can add the details in the DB and can view the list of doctors and their availability

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 3**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S.**  **No** | **Requirement** | **Test Priority** | **Test Steps & pre-**  **conditions** | **Expected Result** | **Obtained Result** | **Success/ Failure** | **Name of Tester** |
| 1 | View Patient Dashboard | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient Dashboard Screen | The system will show the Patient Dashboard with data from various sections (Doctors, Appointments, Medicines) | The patient Dashboard is displayed | Success | Abhay |
| 2 | Side Menu in Patient Dashboard | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient Dashboard Screen | The system will show the following sections on Side Nav Menu  Doctors, Treatment Medicine, Appointment Medicine | The side Menu section is displayed with Doctors, Treatment Medicine, Appointment Medicine Sections. | Success | Dheeraj |
| 3 | List Appointment Screen | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient 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 (Patient) logins to the system 2. On successful login, the system will navigate to Patient 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 (Patient) logins to the system 2. On successful login, the system will navigate to Patient 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 (Patient) logins to the system 2. On successful login, the system will navigate to Patient 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 Treatment Medicine Screen | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient Dashboard Screen 3. Click on the Treatment Medicine menu from Side Nav 4. The system will navigate to List Treatment Medicine Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Treatment Medicine Screen | Success | Abhay |
| 10 | List Appointment Medicine Screen | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient Dashboard Screen 3. Click on the Appointment Medicine menu from Side Nav 4. The system will navigate to List Treatment Medicine Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Appointment Medicine Screen | Success | Abhay |
| 11 | List Doctors Screen | High | 1. User (Patient) logins to the system 2. On successful login, the system will navigate to Patient Dashboard Screen 3. Click on the Doctors menu from Side Nav 4. The system will navigate to List Doctors Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Doctors Screen | Success | Abhay |
| 12 | View Hospital staff Dashboard | High | User (Staff) logins into the system  On successful login, the system will navigate to the staff Dashboard Screen | The system will show the staff Dashboard with data from various sections (Doctors, Appointments, Patients) | Staff  Dashboard is displayed | Success | Dheeraj |
| 13 | Side Menu in staff Dashboard | High | User (Patient) logins to the system  On successful login, the system will navigate to Patient Dashboard Screen | The system will show the following sections on Side Nav Menu  Doctors, Treatment Medicine,  Patient details | The side Menu section is displayed with Doctors, Appointment, Patient details | Success | Ravi |
| 14 | List Appointment Screen | High | User (Patient) logins to the system  On successful login, the system will navigate to Staff Dashboard Screen  Click on the Appointment menu from Side Nav  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 | Gopi |
| 15 | 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 |
| 16 | List Medicine Screen | High | User (Staff) logins into the system  On successful login, the system will navigate to Staff  Dashboard Screen  Click on the Medicine menu from Side Nav  The system will navigate to List Medicine Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Medicine Screen | Success | Dheeraj |
| 17 | List Doctors Screen | High | User (Staff) logins into the system  On successful login, the system will navigate to Staff Dashboard Screen  Click on the Doctors menu from Side Nav  The system will navigate to List Doctors Screen | Previously added records will be shown in the grid | The system is showed appointment records in a grid on the List Doctors Screen | Success | Meghana |
| 18 | List Patient Screen | High | User (Staff) logins into the system  On successful login, the system will navigate to the staff Dashboard Screen  Click on the Patient menu from Side Nav  The system will navigate to the list of patients | Previously added records will be shown in the grid | The system is showed Patients records in a grid on the List Patients Screen | Success | Geetha |
| 19 | Add Patient details Screen | High | User (Staff) logins into the system  On successful login, the system will navigate to the staff Dashboard Screen  Click on the Patient menu from Side Nav  The system will navigate to the list of patients | The system will display a set of form fields with the button Create Patient | The system showed Form Fields to create a new Patient | Success | Meghana |
| 20 | Edit Patient | High | User (Staff) logins into the system  On successful login, the system will navigate to the staff Dashboard Screen  Click on the Patient menu from Side Nav  The system will navigate to the list of patients | The system will display a set of form fields with the button Update Patient to update the Patient details | The system showed Form Fields to update existing patient details | Success | Gopi |
| 21 | Delete Patient | High | User (Staff) logins into the system  On successful login, the system will navigate to the staff Dashboard Screen  Click on the Patient menu from Side Nav  The system will navigate to the list of patients | 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 | Ravi |

* 1. **System Test Cases**

**Test case 1:**

**Functionality: Registration**

Step 1: Enter all the required fields. Username, Email, First Name, Last Name, User Role, Password, Confirm Password.

* Username: admin1
* Email: xyz@abc.com
* First Name: mark
* Last Name: Zuckerberg
* User Role: Patient
* Password: 123
* Confirm Password: 1234

Here the password is intentionally entered wrong as 1234 instead of 123

Step 2:

Click on Create Account

Expected result: Error message “New Password and Confirm New Password should be same”

Actual result: Error message “New Password and Confirm New Password should be same”

Success /Failure: Success

**Test Case 2:**

**Functionality: Password**

Step 1: Enter all the required fields. UserName, Email, First Name, Last Name, User Role, Password, Confirm Password.

* UserName: admin1
* Email: xyz@abc.com
* First Name: mark
* Last Name: Zuckerberg
* User Role: Patient
* Password: 123
* Confirm Password: 123

Step 2:

Click on Create Account

Expected result: Display message “Registered Successfully ”

Actual result: Display message “Registered Successfully ”

Success /Failure: Success

**Functionality Use case - Login:**

Test Case 1:

Step 1: Enter Username, Password

Username:

Password:

Expected result: Enters the homepage

Actual result: Enters the homepage

Success /Failure: Success

**Functionality Use case - Homepage**

**Test Case 1:**

Functionality: The homepage has various buttons on it like the dashboard, Appointment Medicine, Treatment Medicine, Appointment. We are testing if at the click of each button we are moving to the target page.

Step 1: On clicking the dashboard item

Expected result: Stays on the homepage

Actual result: Stays on the homepage

Success /Failure: Success

**Test Case 2:**

Functionality: The homepage has various buttons on it like the dashboard, Appointment Medicine, Treatment Medicine, Appointment. We are testing if at the click of each button we are moving to the target page.

Step 1: On clicking the Hospital Branch item

Expected result: Must move to the Hospital Branch page

Actual result: Must move to the Hospital Branch page

Success /Failure: Success

**Test Case 3:**

Functionality: The homepage has various buttons on it like the dashboard, Appointment Medicine, Treatment Medicine, Appointment. We are testing if at the click of each button we are moving to the target page.

Step 1: On clicking the Doctor item

Expected result: Must move to the Doctor page

Actual result: Must move to the Doctor page

Success /Failure: Success

**Test Case 4:**

Functionality: The homepage has various buttons on it like the dashboard, Appointment Medicine, Treatment Medicine, Appointment. We are testing if at the click of each button we are moving to the target page.

Step 1: On clicking the Patient item

Expected result: Must move to patient page

Actual result: Must move to patient page

Success /Failure: Success

**Test Case 5:**

Functionality: The homepage has various buttons on it like the dashboard, Appointment Medicine, Treatment Medicine, Appointment. We are testing if at the click of each button we are moving to the target page.

Step 1: On clicking the Specialization item

Expected result: Must move to the Specialization page

Actual result: Must move to the Specialization page

Success /Failure: Success

**Functionality Use Case: Admin**

Test Case 1:

Functionality: Add Hospital Branch

Step 1: Enter all required fields. Name, Address, City, Phone 1, Email.

Name: health care

Address: Spain

City: Barca

Phone 1: 2374384982

Email: x@y.com

Click on create button

Expected Result: Adds the hospital branch records successfully.

Actual Result: Adds the hospital branch records successfully.

Success /Failure: Success

**Functionality Use Case: Doctor**

Test Case:

Functionality: Add Appointment Medicine

Step 1: Enter all required fields. Appointment Id, Medicine Id.

Appointment Id: PAY1002

Medicine Id: Crocin

Click on create button

Expected Result: Adds to the Appointment Medicine records successfully.

Actual Result: Adds to the Appointment Medicine records successfully.

Success /Failure: Success

**Functionality Use Case: Patient**

Test Case:

Functionality: Add Appointment

Step 1: Enter all required fields. Hospital Id, Doctor Id, Patient Id, Appointment Date.

Hospital Id: Saroja specialist

Doctor Id: Ryan

Patient Id: Louis

Click on create button

Expected Result: Adds to the Appointment records successfully.

Actual Result: Adds to the Appointment records successfully.

Success /Failure: Success

**Functionality Use Case: Staff**

Test Case:

Functionality: List of Medicines

Expected Result: Displays the List of Medicines available on the screen.

Actual Result: Displays the List of Medicines available on the screen.

Success /Failure: Success

**End2End Test Cases**

<https://github.com/abhayarora23UNT/UntSeProjects2022/tree/main/HMS/Testing/HMS_e2e_Test_Cases>

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, Staff, 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, Staff, 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.

**Patient Dashboard**

In this phase 3, we, have developed the Patient dashboard and Hospital staff dashboard.

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

Graphical user interface, website

Description automatically generated

*Fig 11*

* If the user of Type (Patient), logins into the system he can view the modules on the left side of the dashboard where we can see the different functionalities. Patients can view the doctor's list by clicking on the doctors’ button.

Table

Description automatically generated

*Fig 12*

* On the left side of the dashboard, where we can see the different functionalities. Patients can view/edit/add the appointments with the doctor by clicking on the appointments list button.

Table

Description automatically generated

*Fig 13*

* On the left side of the dashboard, where we can see the different functionalities. Patients can add the appointments with the doctor by clicking on the add appointments button.

Graphical user interface, application

Description automatically generated

*Fig 14*

* Patients can view the medicines prescribed by the doctor for that treatment by clicking on the treatment medicine button

Graphical user interface, application, table

Description automatically generated

*Fig 15*

* Patients can view the medicines prescribed by the doctor on that particular appointment by clicking on the treatment medicine button.

Table

Description automatically generated

*Fig 16*

**Staff Dashboard**

In this phase 3, we, have developed the Hospital staff dashboard.

Graphical user interface, website

Description automatically generated

*Fig 17*

* Table

  Description automatically generatedStaff can view the doctor's available list on the dashboard

*Fig 18*

* Staff can view the patient's list and add the patient details or edit the details

Table

Description automatically generated

*Fig 19*

* Staff can view the appointments list for that doctor

Table

Description automatically generated

*Fig 20*

* Staff can view the list of available medicines in the hospital

Table

Description automatically generated

1. **Features, limitations, and feedback from peer reviews**.

**Features**

1. **Authentication and Authorization**
   1. **User Registration**
      * The user of type Doctor, Patient, Staff, and Admin can register to Hospital Management System.
   2. **User Login**:
      * Only those users who are registered and active can log in to this system.
   3. **Role-Based Authorization**:
      * The system will show the relevant section of the screen to the specific type of user on basis of the role permissions matrix.
2. **Patient Module**
   1. Registered patients can schedule new appointments with a specific doctor on basis of the doctor’s availability.
   2. The system also provides the feature to reschedule or cancel existing appointments.
   3. Patients can view the details of doctors working in a specific department.
   4. The system showcases medications linked with specific patient appointments and generic against the disease treatment.
3. **Staff Module**
   1. Staff members of the hospital can modify the details of patients, who visited for consultation or treatment.
   2. Staff can view the details of doctors working in a specialized department.
   3. The system also provides the feature to staff members to view all the patient appointments.
   4. Staff can provide the details of different medicines available in the hospital pharmacy.
4. **Doctor Module**
   1. Registered doctors can add the details of prescribed medicines against specific patient appointments
   2. Doctors can request specific medicines which are required for some common types of treatments
   3. The system also provides the feature to doctors to view and modify patient appointments.
5. **Admin Module**
   1. Admin is the primary user of this system.
   2. Admin has complete access and privileges to perform any actions.
   3. The system provides various rights to the admin to modify hospital branches, doctors, and patient details in the system.
   4. Admin can maintain inventory of items required in managing the whole hospital system like room types, specializations, treatment, and medications.

**Feedback received during usability testing session**

* Tooltip text can be added to some buttons/icons.
* The use of colors and font size can be improved.
* Icons inside the navbar should be like the action they perform.
* User data can be secured using some additional framework at the backend level.

Pointers Accepted

* Tooltip text can be added to some buttons/icons.
* Icons inside the navbar should be like the action they perform.

Pointers Rejected

* The use of colors and font size can be improved.
* User data can be secured using some additional framework at the backend level.

**Feedback received during requirement/proposal discussion**

* Adding Hospital Staff to collect data in the event of an emergency.
* Nurse login

Accepted**:**

* Hospital Staff

Rejected**:**

* We have rejected login for nurses instead we have added hospital staff.

**Feedback received during code inspections**

* 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

**Future Scope and Plans**

1. Integration of Insurance module
   * We will consult regional Insurance providers and TPAs to get registered with our hospital system to provide insurance services to the patient and staff members.
2. Integration with external Blood Bank Society
   * The hospital will provide blood bank facilities to patients and their families.
3. Notification Messaging
   * We will try to incorporate Firebase Push Notifications to send timely reminders to patients about their appointments.
4. Support Chat
   * A chatbot feature will be added to the website to assist users to navigate easily within the system and scheduling appointments with minimal user actions.
5. Mobile Support
   * We will try to release the beta version of the mobile application of the system
6. **Accomplishments/Challenges**

As per the proposed phase-3 project plan, we accomplished the development and testing of the following screens

1. Patient Module
   1. Dashboard, Appointments, Doctors, Medicines.
2. Staff Module
   1. List of Appointments, Doctors, Medicines, Add/edit Patient details
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 with development in Angular Framework and .Net Framework.
   1. To overcome that, some members do peer code reviews and provide internal training to other team members.
2. Deployment and Hosting on Microsoft Azure Server
   1. Faces some issues related to licensing and server cost

**Things to Improve**

1. For future phases, more focus will be given to the integration of individual sub-modules.
2. Unit testing and system testing.
   1. We will use some automated test cases framework to find and fix the bugs at the initial level of development.
   2. More regression usability testing should be done to avoid any issues which can be faced by end-users

**8. Member Contribution Table**

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall Contribution (%)** |
| DHEERAJ REDDY AGUTHU | * Deliverable 5 Documentation   + - 1. UML (Class Diagram)       2. Test Cases       3. User Manual * Backend Database creation * Front End Development(Staff Module) * Testing | 12.5 |
| ABHAY ARORA | * Updated the Minutes of Meeting in Repo * Deliverable 5 Documentation  1. Code Inspection Doc 2. Test Cases 3. User Manual  * Front End Development (Initial Architecture, Patient Dashboard, Appointment (List, Edit, Delete) * Deployment and Unit Testing on Azure * Deliverable 5 Task Management | 12.5 |
| RAVI TEJA BALAJI | * Development phases and Member Contribution tables. * Front End Development(Staff Module) * Deliverable 5 Documentation * UML (Class Diagram) * Peer Review Section Feedback * Test Cases | 12.5 |
| PRAVEEN NAKKA | * Deliverable 5 Documentation  1. Requirements 2. Peer Review Section Feedback  * Front End Development (Staff Module) and admin module | 12.5 |
| CHARISHMA NAGA SAI SARADA BALUSU | * Deliverable 5 Documentation  1. Sequence Diagrams  * Testing | 12.5 |
| GEETHA KRISHNA DODDA | * Deliverable 5 Documentation  1. Sequence Diagrams  * Testing | 12.5 |
| SRIKANTH GOPI | * Deliverable 5 Documentation  1. Requirements 2. UML (Use Case Diagram) 3. Test Cases | 12.5 |
| MEGHANA JUNNUTULA | * Deliverable 5 Documentation  1. Requirements 2. UML (Use Case Diagram) 3. User Manual 4. Code Inspection  * API Initial Structure + API implementation (Patient and Staff Module) * 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>