HealthEase System: Work Breakdown Structure (WBS)

1.1. Work Breakdown Structure

1.1.1. Project Management (Assigned to Aqsa Malik)

1.1.1.1. Project Planning & Scheduling

- Define the project scope, objectives, and deliverables.
- Develop a project timeline with milestones.
- Allocate tasks to team members.
- Conduct regular project status meetings.
- Track project progress and resolve any issues that arise.

1.1.1.2. Risk Management

- Identify potential risks to the project (technical, operational, etc.).
- Develop strategies for mitigating risks.
- Monitor and manage risks throughout the project lifecycle.

1.1.1.3. Progress Reporting

- Track the completion of tasks and user stories.
- Provide regular status updates to the team and stakeholders.
- Document progress and maintain transparency.

1.1.2. UI design (Assigned to Aqsa Malik)

1.1.2.1. Design Registration Page

- Layout creation for patient and doctor registration forms.
- Input fields: name, email, phone number, password, etc.

- Add validation for fields such as email format.

1.1.2.2. Design Login Page

- User-friendly login page for both patients and doctors.

1.1.2.3. Design Appointment Booking Page

- Develop UI for selecting a doctor, date, and time slots.
- Include real-time availability of doctors.
- Display confirmation of appointment once booked.

1.1.2.4. Design Appointment Cancellation Page

- Interface for patients to view and cancel upcoming appointments.
- Confirmation modal before finalizing the cancellation.

1.1.2.5. Design Doctor Availability Page

- Allow doctors to set their available/unavailable time slots.
- Provide an interface for updating availability with a calendar view.

1.1.2.6. Confirmation & Notifications of UI

- Design email confirmation templates for appointment booking and cancellations for both patients and doctors.
- Display appointment reminders and updates in the user interface.

1.1.2.7. User Testing of UI

- Conduct usability testing to ensure ease of navigation and functionality.
- Collect feedback from real users to identify areas for improvement.

1.1.3. Backend Development (Assigned to Tooba Ali and Sawab Akbar)

1.1.3.1. Set up an Authentication System

- Implement secure user registration and login functionality.
- Include role-based authentication for patients and doctors.

1.1.3.2. Develop Appointment Scheduling Logic

- Implement logic for booking, rescheduling, and canceling appointments.
- Ensure appointments can only be booked in available time slots.
- Develop logic for conflict resolution when overlapping appointments are attempted.

1.1.3.3. Implement Doctor Availability Management

- Enable doctors to set and update their availability.
- Ensure the system correctly reflects the doctor's availability in the appointment booking system.

1.1.3.4. Implement an Email Notification System

- Integrate an external email service (e.g., Gmail API) for sending confirmation and reminder emails.
- Set up automated emails for patient and doctor notifications upon booking, cancellation, and reminders.

1.1.3.5. Implement an Appointment Cancellation Feature

- Provide functionality for patients to cancel appointments.
- Notify doctors when an appointment is canceled.

1.1.3.6. Develop Patient Feedback System

- Enable patients to provide feedback after a consultation.

- Ensure feedback is associated with the corresponding doctor.

1.1.4. Database Setup (Assigned to Sawab Akbar)

1.1.4.1. Design Database Schema

- Design database schema for patients, doctors, appointments, and feedback.
- Define relationships between entities (e.g., patients and appointments, doctors and availability).

1.1.4.2. Create Tables for Patients & Doctors

- Implement tables to store patient and doctor information, including names, contact details, credentials, and roles.

1.1.4.3. Create Appointment Table

- Design and implement a table to track appointments (appointment time, status, patient, doctor, etc.).

1.1.4.4. Configure Email Service Integration

- Set up email service configuration for sending automated emails.
- Store logs of sent emails for audit purposes.

1.1.4.5. Database Optimization & Indexing

- Optimize database queries for faster data retrieval, especially for appointment scheduling.
- Create indexes on commonly queried columns (e.g., patient ID, doctor ID, appointment status).

1.1.5. Testing (Assigned to Aqsa Malik and Tooba Ali)

1.1.5.1. Test Patient Registration

- Validate patient registration functionality, including proper validation of inputs and email confirmation.

1.1.5.2. Test Doctor Registration

- Ensure doctor registration works correctly, allowing them to manage appointments and availability.

1.1.5.3. Test Login & Authentication

- Test the login functionality for both patients and doctors.
- Ensure the authentication process is secure and user-friendly.

1.1.5.4. Test Appointment Booking & Rescheduling

- Verify that patients can book appointments with available doctors.
- Test rescheduling functionality to ensure it updates correctly in the system.

1.1.5.5. Test Appointment Cancellation

- Test appointment cancellation process for both patients and doctors.
- Confirm email notifications are sent when cancellations are made.

1.1.5.6. Test Doctor Availability Management

- Verify that doctors can accurately update their availability and that the system reflects these changes in real-time.

1.1.5.7. Test Feedback System

- Ensure that patients can submit feedback after appointments and that feedback is stored correctly.

1.1.5.8. Performance Testing

- Test the system's performance under high loads (e.g., 100 concurrent users).
- Ensure there is no performance degradation during peak times.

1.1.6. Documentation (Assigned to Aqsa Malik, Sawab Akbar & Tooba Ali)

1.1.6.1. Prepare User Guide

- Create a user-friendly guide for both patients and doctors detailing how to use the system.

1.1.6.2. System Architecture Documentation

- Document the overall system architecture, including front-end, back-end, and database components.
- Describe integrations with external services like email API

1.1.6.3. Database Design Documentation

- Provide documentation for the database schema, tables, and relationships.
- Include instructions for managing and maintaining the database.

1.1.6.4. Testing Documentation

- Document all testing processes, including test cases, testing results, and fixes.
- Provide a final report of system tests to ensure functionality and performance.

1.2. Gantt Chart

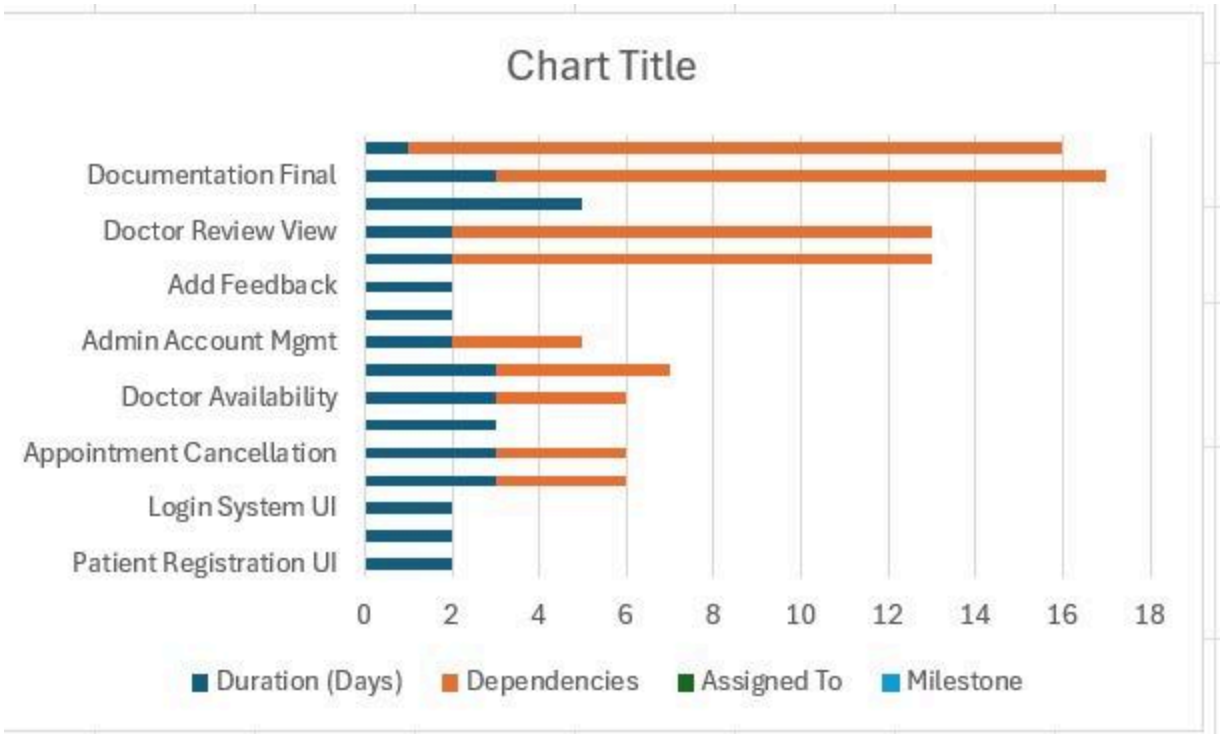
Chart Title
Documentation Final
Doctor Review View
Add Feedback
Admin Account Mgmt
Doctor Availability
Appointment Cancellation
Login System UI
Patient Registration UI
0
2
4
6
8
10
12
14
16
18
Duration (Days)
Dependencies
Assigned To
Milestone