



University of
Staffordshire

A • P • I • I • T
Inspire love for learning

ASSIGNMENT

LEVEL 5

COMP50001 : Commercial Computing

Hetti Arachchige Lakindu Hesara Perera
CB015653

Abbas Ibrahim Mohamede Feizal (Stefan)
CB015841

Senithi Amasha Ranatunga
CB015671

Abdul Muheed
CB013114

Ayodhya Prabhasani Jayawardhna
CB017378

Table of Contents

Contents

Introduction	3
PM Document	4
Time Management	4
Critical Path Method (CPM)	4
Moscow prioritization	5
PM tool evidence – Jira	6
Standup meetings – sprint 2	8
Time Assigned for Each Feature Development	7
Sprint 2	10
Sprint goal	10
Sprint roles	10
Sprint 2 workshop output	11
Conceptual design	13
Upload patient records	Error! Bookmark not defined.
Patient History	13
Notification/Reminder System	16
Payment Integration	18
Implementation – Justification of tools used & how user pain points dealt with .	20
Evidence of testing	21
Sprint Retrospective	38

Introduction

This is second sprint report- including the Time management and justifications that were not mentioned in the first sprint.

And the sprint goal for the second sprint is to establish long term clinical continuity by digitizing patient history and reducing administrative burden on the receptionist through automated notifications and secure online payments. Which is a MVP.

In the report we have included the time management part from the previous PM document and the sprint 2 goal, roles of the members, the dashboard that was created in the workshop, conceptual design including the use case diagrams, activity diagrams, sequence diagrams and the test cases of the features that were developed and the team retrospectives.

PM Document

Time Management

Critical Path Method (CPM)

The **Critical Path Method (CPM)** was used to identify activities that directly impact the overall project timeline.

Critical Activities

Critical Path Sequence for MediFollow:

1. Authentication System (Sprint 1)
2. Appointment Core Module (Sprint 1)
3. Follow-up Workflow (Sprint 2)
4. Patient History System (Sprint 2)
5. Analytics Dashboard (Sprint 3)

These activities form the **critical path** because:

Authentication System – Required first ,secures user access.

Appointment Core Module – Builds on authentication, enables scheduling and clinical interactions.

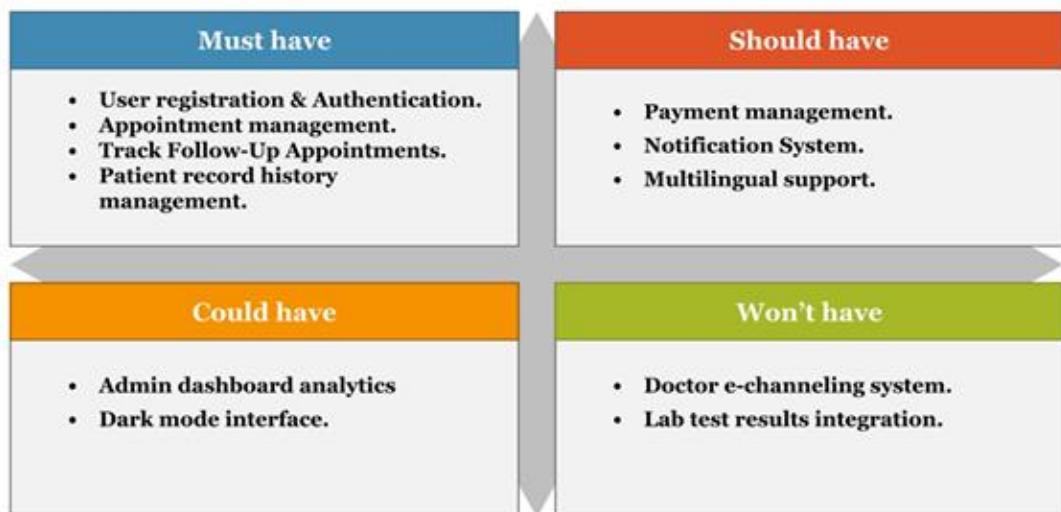
Follow up Workflow – Relies on completed appointments, ensures continuity of care.

Patient History System – Depends on consultations and follow ups, stores longitudinal patient data.

Analytics Dashboard – Uses aggregated data from appointments, follow-ups, and history, provides insights.

Moscow prioritization

MoSCow CHART



Must have

These features form the core of the MVP, ensuring clinical continuity by addressing security needs and supporting essential appointment and patient history workflows.

Should have

These features reduce receptionist workload through automation, while core consultations can still function without them in early testing.

Could have

These are nice to have features that enhance usability and insights but do not affect the core patient doctor follow up workflow.

Won't have

These items were excluded to control scope, as complex third party integrations are not required to address follow up appointment mismanagement.

PM tool evidence – Jira

The screenshot shows the Jira interface for the 'MediFollow CC' space. The left sidebar includes links for 'For you', 'Recent', 'Starred', 'Apps', 'Spaces', 'Recent', 'Teams', and 'Give feedback on the n...'. The main area displays the 'Backlog' tab under the 'Summary' section. A search bar at the top allows filtering by 'User', 'Status', and 'Priority'. Below the search bar, a summary for 'SCRUM Sprint 2' (11 Dec - 30 Dec) shows 7 work items. The backlog items are:

- SCRUM-10 As a Patient, I want to manage my appointments s... APPOINTMENT MAN... TESTING
- SCRUM-14 As a Doctor/Admin, I want to schedule a follow-up... TO DO
- SCRUM-12 As a Doctor, I need to consult the patient and reco... PATIENT HISTORY TO DO
- SCRUM-18 As a Patient, I want to make an online payment for... PAYMENT INTEGRATI... TO DO
- SCRUM-13 As a Doctor, I want to view incoming follow-up req... TO DO
- SCRUM-15 As a Receptionist/Admin, I want to upload patient r... TO DO

A 'Quickstart' button is visible on the right.

The screenshot shows the Jira 'Board' screen for the 'MediFollow CC' space. The board is divided into four columns: 'TO DO' (5 items), 'IN PROGRESS' (1 item), 'TESTING' (1 item), and 'DONE' (0 items). Each column contains cards representing work items. The 'TO DO' column has one card: 'SCRUM-14 As a Doctor/Admin, I want to schedule a follow-up appointment directly from an approved request so the request converts into a linked appointment.' The 'IN PROGRESS' column has one card: 'SCRUM-23 As a Patient/Doctor, I want to receive reminder notifications so I don't miss my appointments'. The 'TESTING' column has one card: 'SCRUM-10 As a Patient, I want to manage my appointments so that I may get care from the system.' The 'TESTING' card is associated with the 'APPOINTMENT MANAGEMENT' epic. A 'Quicks' button is visible on the right.

The screenshot shows the Jira 'Timeline' screen for the 'MediFollow CC' space. The timeline is displayed as a grid where rows represent sprints and columns represent time periods (December and January '26). The 'Work' column lists sprints: 'SCRUM Sprint 1' (green bar), 'SCRUM Sprint 2' (purple bar), 'SCRUM-6 Appointment Management' (blue bar), 'SCRUM-7 Patient History' (green bar), 'SCRUM-8 Payment Integration' (purple bar), and 'SCRUM-20 Doctor Scheduling' (green bar). The 'Epic' and 'Status category' dropdowns are visible at the top. A navigation bar at the bottom includes 'Today', 'Weeks', 'Months' (selected), 'Quarters', and 'Years'.

Time Assigned for Each Feature Development

Feature	Total days	dates	Justifications
Follow up System	16 days	Nov 27 - Dec 18	Main feature for managing follow ups, involves complex workflow, secure handling of patient data, and careful testing for safety.
Email notifications	8 days	Dec 1 - Dec 10	External Service Integration ,Backend focused integration with email providers, frontend is minimal
Payment Integration	12 days	Dec 8 - Dec 23	High risk feature, external API integration, and thorough testing for secure transactions.
Patient history	11 days	Dec 15 - Dec 29	Patient Timeline Frontend focused feature using appointment data, involves complex timeline logic and detailed medical record display for better UX.

Standup meetings – sprint 2

Date	Meeting type	Progress update
28/11/2025	WhatsApp group call	Role distribution.
30/11/2025	WhatsApp group call	Completing the 1 st sprint missed parts.(All the members)
01/12/2025	WhatsApp group call	Developer making API work, BA drawing diagrams and cleared some misunderstandings on notations.
02/12/2025	WhatsApp group call	Completing system API endpoints .BA working on diagrams.
03/12/2025	Physically at APIIT	Follow-up System backend 60% complete.BA s working on diagrams. QA working on previous sprint work.
05/12/2025	WhatsApp group call	Reviewed Follow-up System diagrams. Started Email Notifications planning. Security review for medical data. QA working on previous sprint missed development.
07/12/2025	Through messages	Developer started Email Notifications backend. BA s working on diagrams. QA preparing test cases for the completed diagrams.
08/12/2025	WhatsApp group call	One of the BA s sent all her diagrams. QA making test cases. Developer working on the notification feature.
10/12/2025	Physically at APIIT	Developer started Payment Integration feature.
12/12/2025	WhatsApp group call	Payment transactions working in test mode. Patient History database design started. Email notifications integrated with follow-up system.

15/12/2025	Through messages	Follow-up System ready for QA testing. Payment Integration frontend development started. Started making the document.
16/12/2025	Physically at APIIT	One of the BA's having issues with the diagrams. QA making test cases and testing the features.
17/12/2025	Physically at APIIT	Patient History backend development. BA's finalizing the diagrams
18/12/2025	Physically at APIIT	Final bug fixes for Payment system. Patient History frontend development started. Preparing for deployment.
20/12/2025	WhatsApp group call	QA working on the test cases
22/12/2025	Through messages	No updates
23/12/2025	Physically at APIIT	Developer working on the patient history feature. QA working on test cases and testing the features
24/12/2025	Through messages	No updates
27/12/2025	WhatsApp group call	Developer fixing bugs and missed parts. QA working on the testing of the features
28/12/2025	WhatsApp group call	Developer fixing bugs in the developed features. QA doing the test cases.
29/12/2025	WhatsApp group call	Sprint retrospective. Final documentation updates. Demo preparation for review.

Sprint 2

Sprint goal

To establish long term clinical continuity by digitizing patient history and reducing administrative burden on the receptionist through automated notifications and secure online payments.

Sprint 2 Backlog

- Notification/Reminder System
- Patient History
- Payment Integration

Sprint roles

Sprint 2

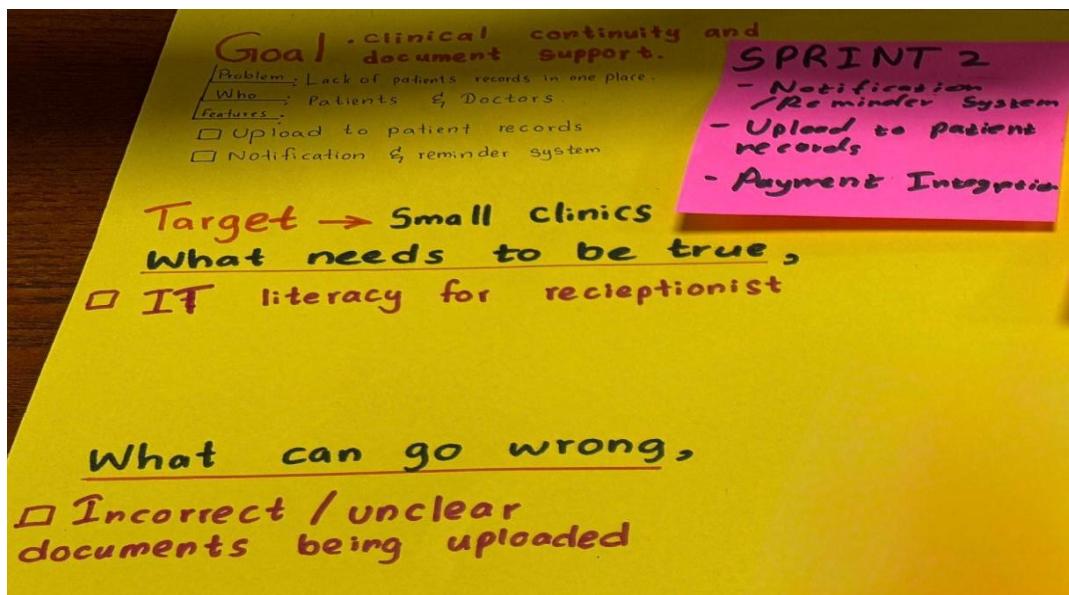
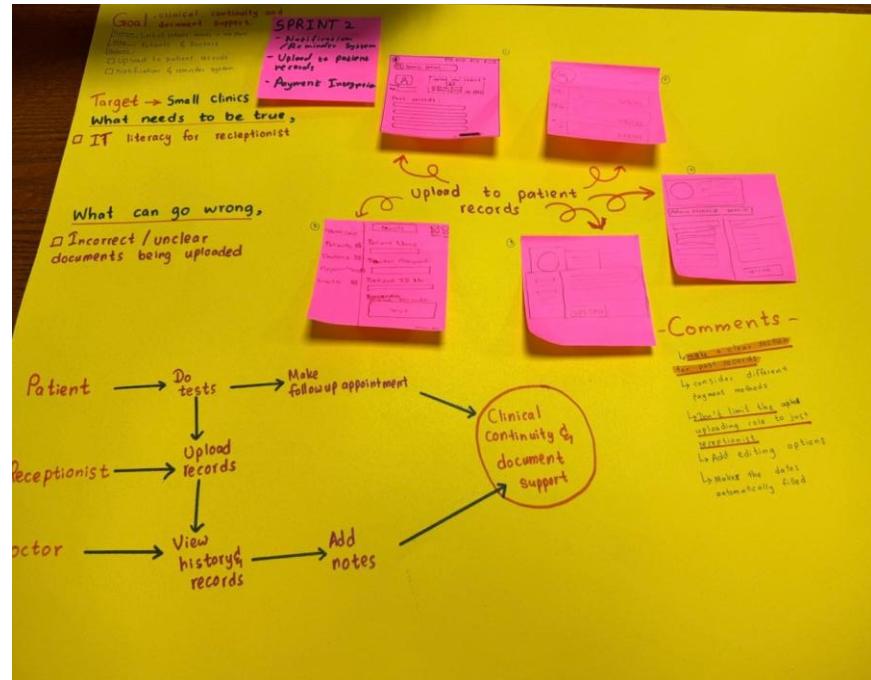
Scrum Master (SM) – Senithi Ranathunga

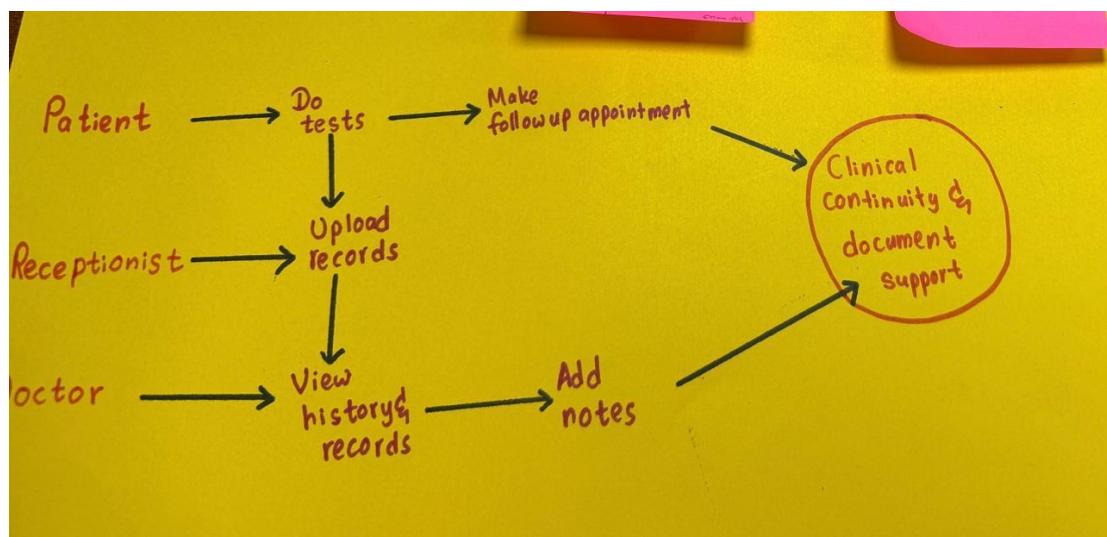
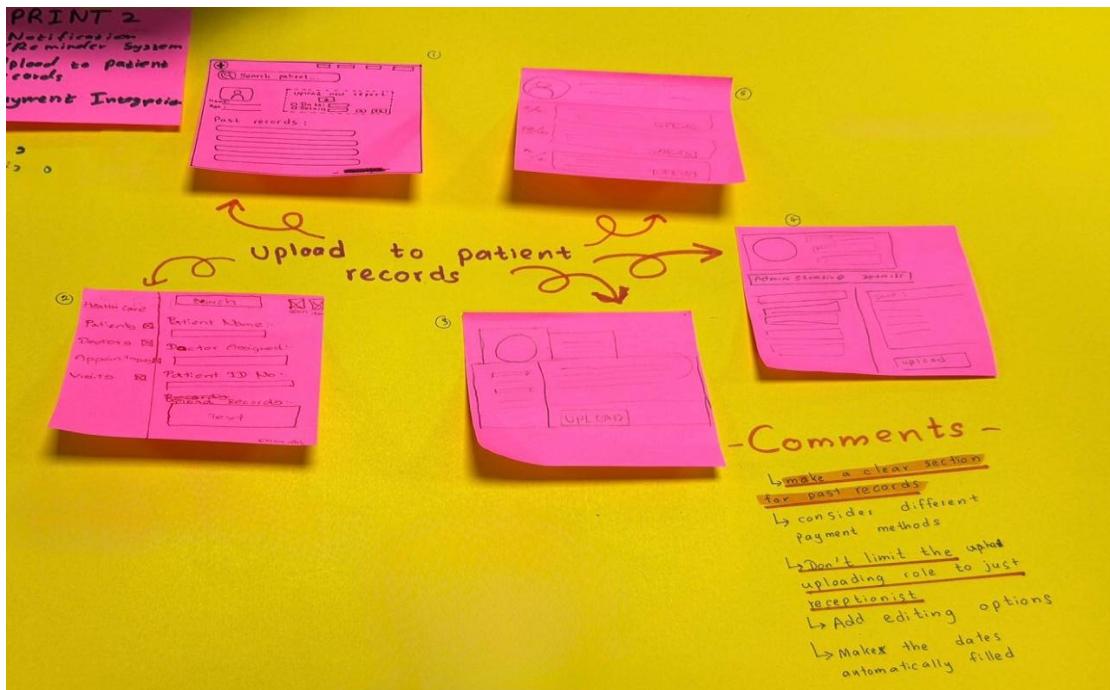
Business Analyst (BA) – Abdul Muheed , Ayodhya Prabashini

Developer –Lakindu Perera

Quality Assurance (QA) – Stefan Shabbir

Sprint 2 workshop output





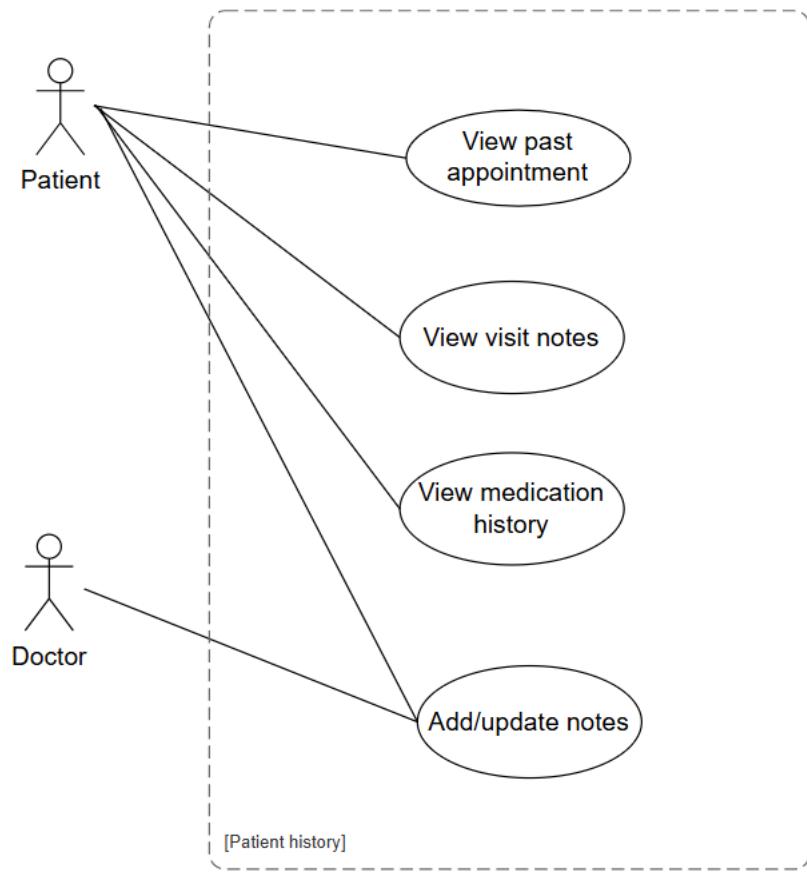
Conceptual design

Patient History

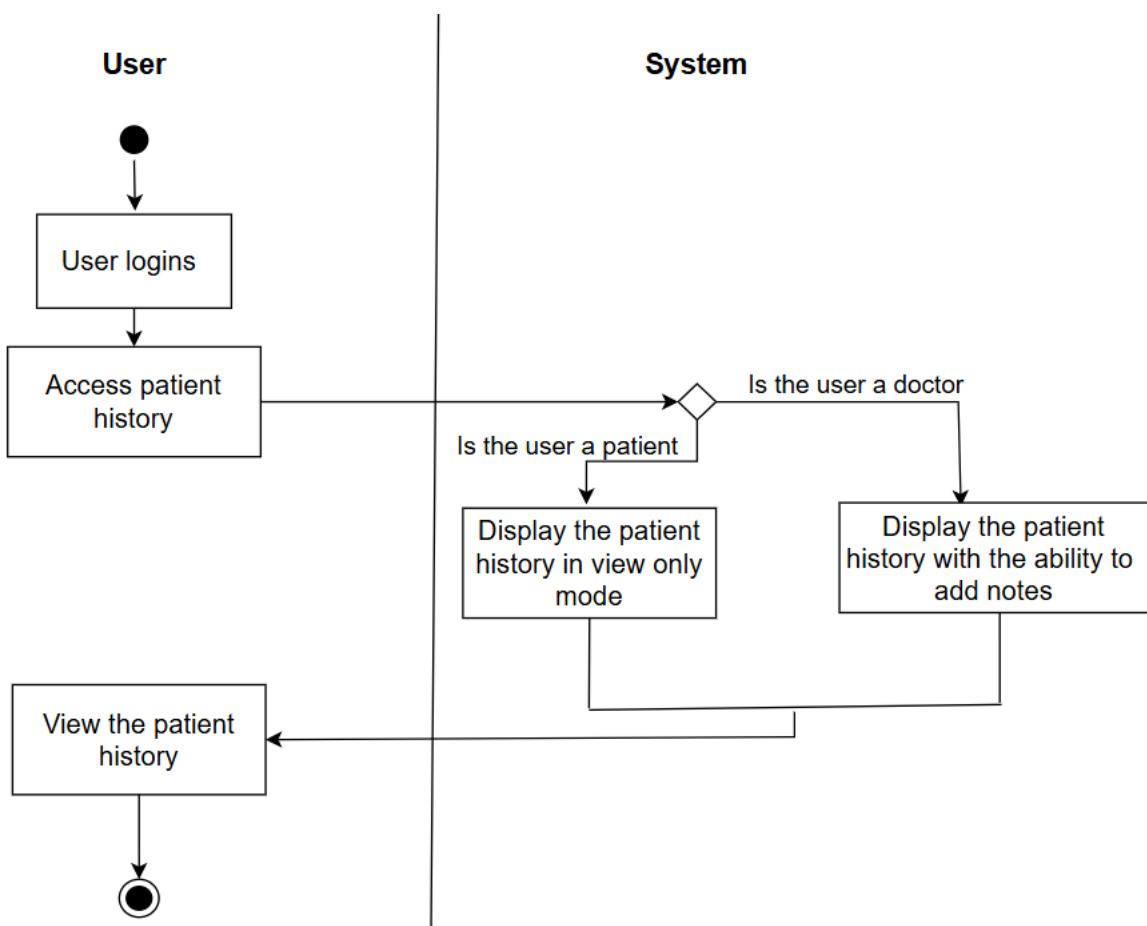
User stories

As a Doctor, I need to consult the patient and record their patient history so that they may get the appropriate care

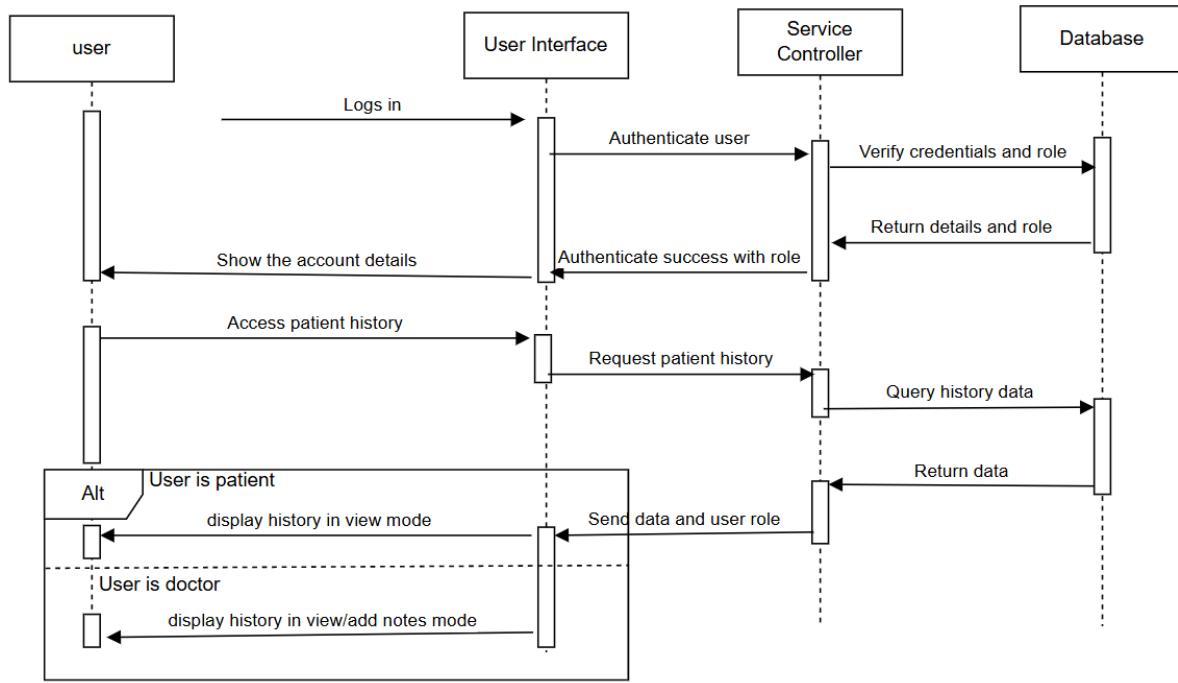
Use case diagram



Activity diagram



Sequence diagram

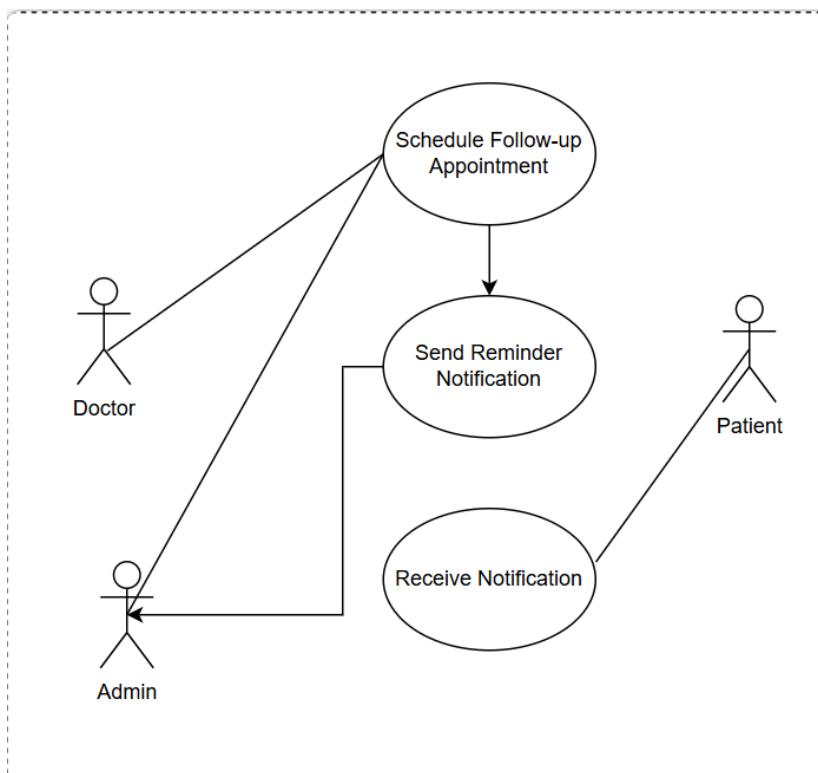


Notification/Reminder System

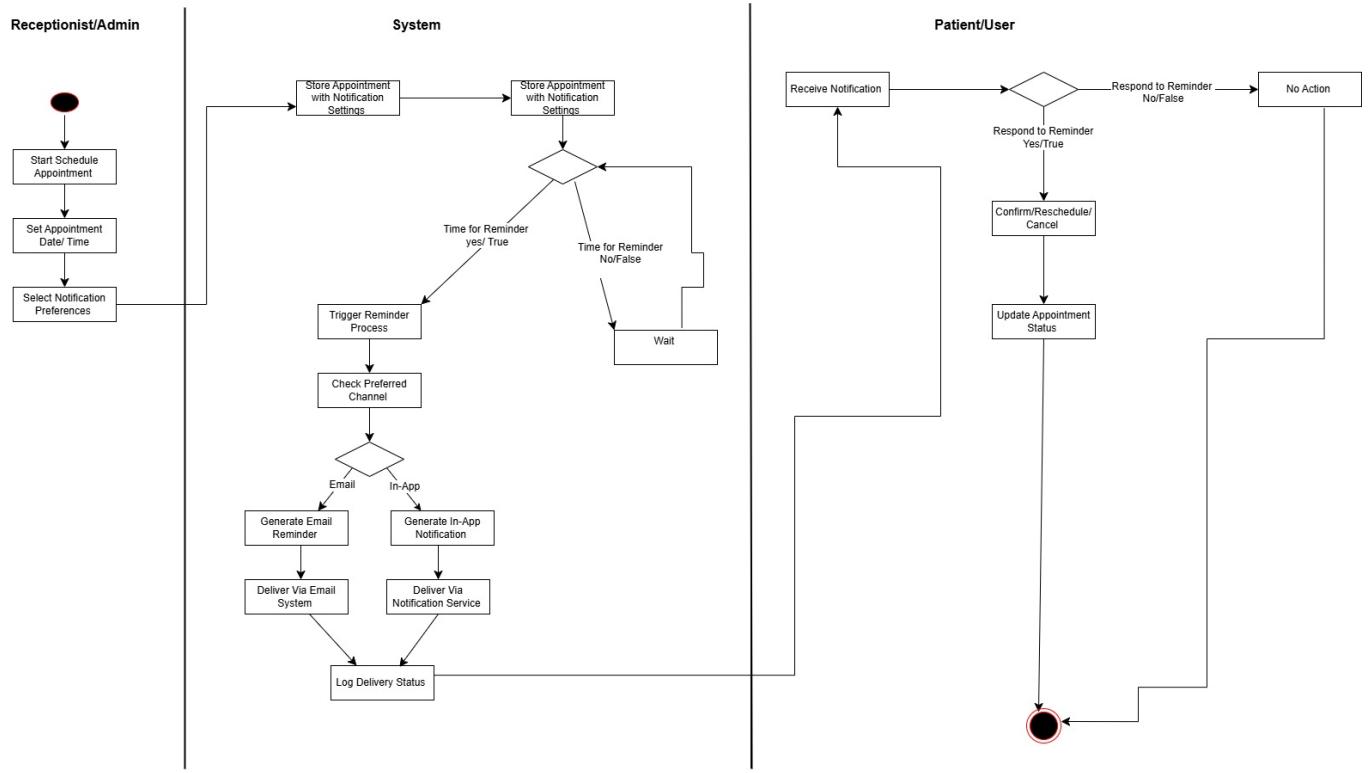
User stories

As a Doctor/Admin, I want to schedule a follow-up appointment directly from an approved request so the request converts into a linked appointment.

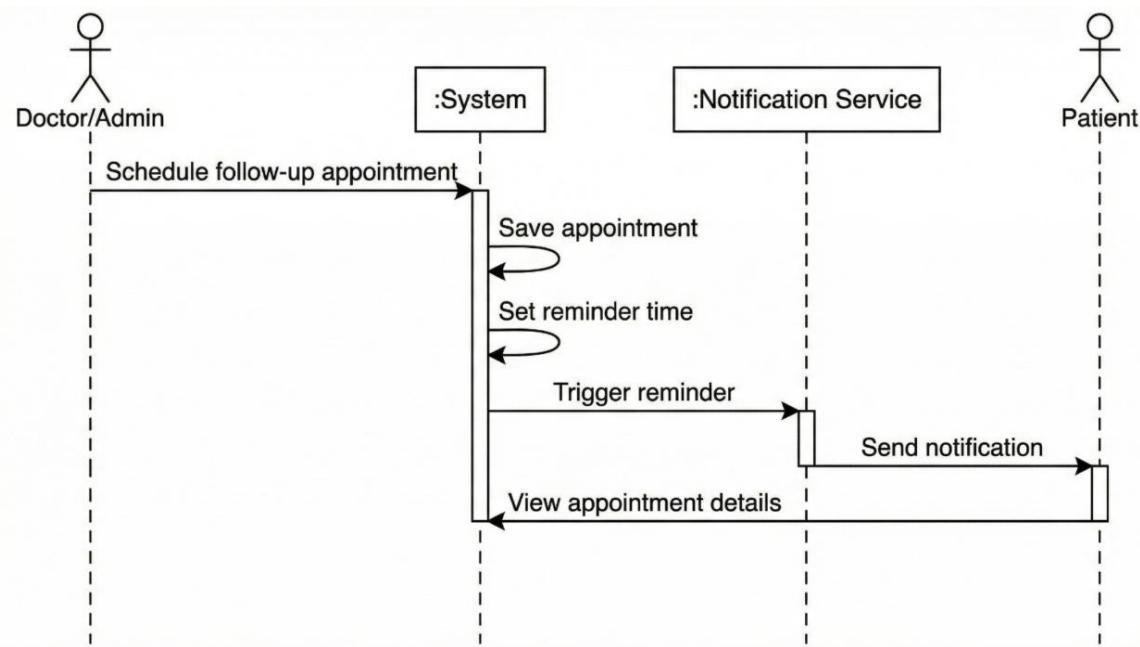
Use case diagram



Activity diagram



Sequence diagram

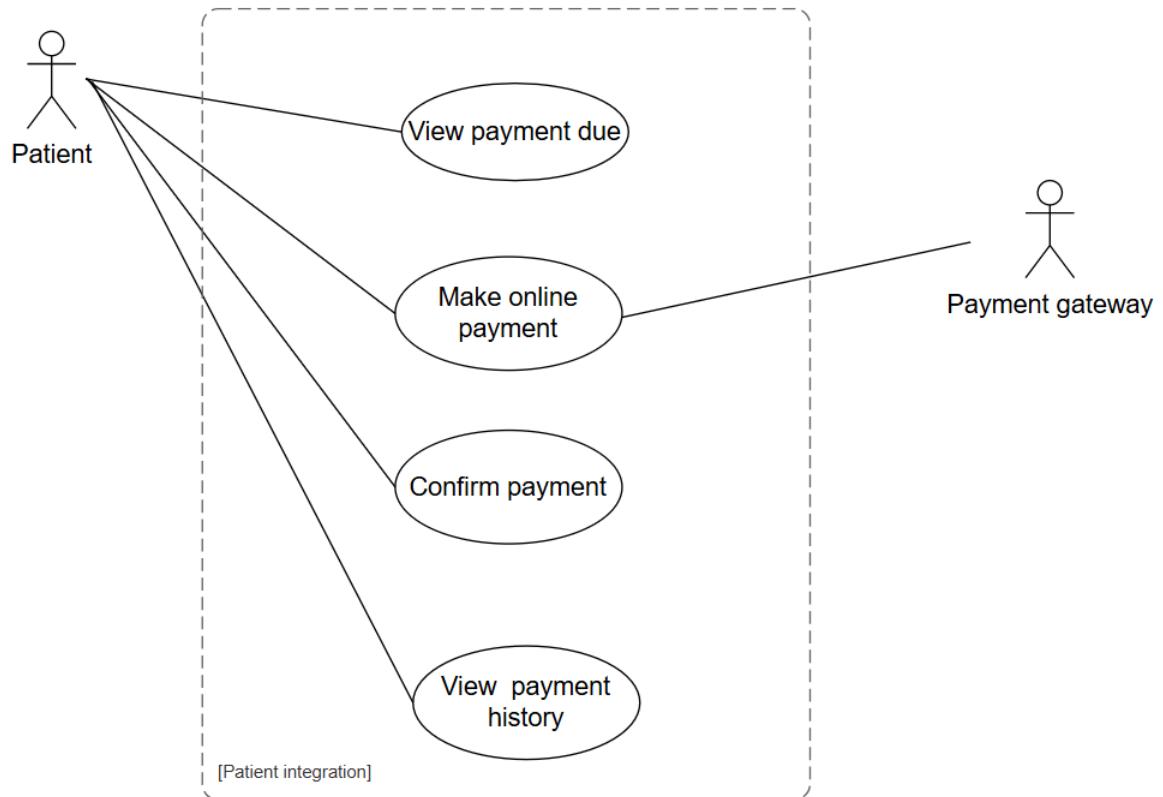


Payment Integration

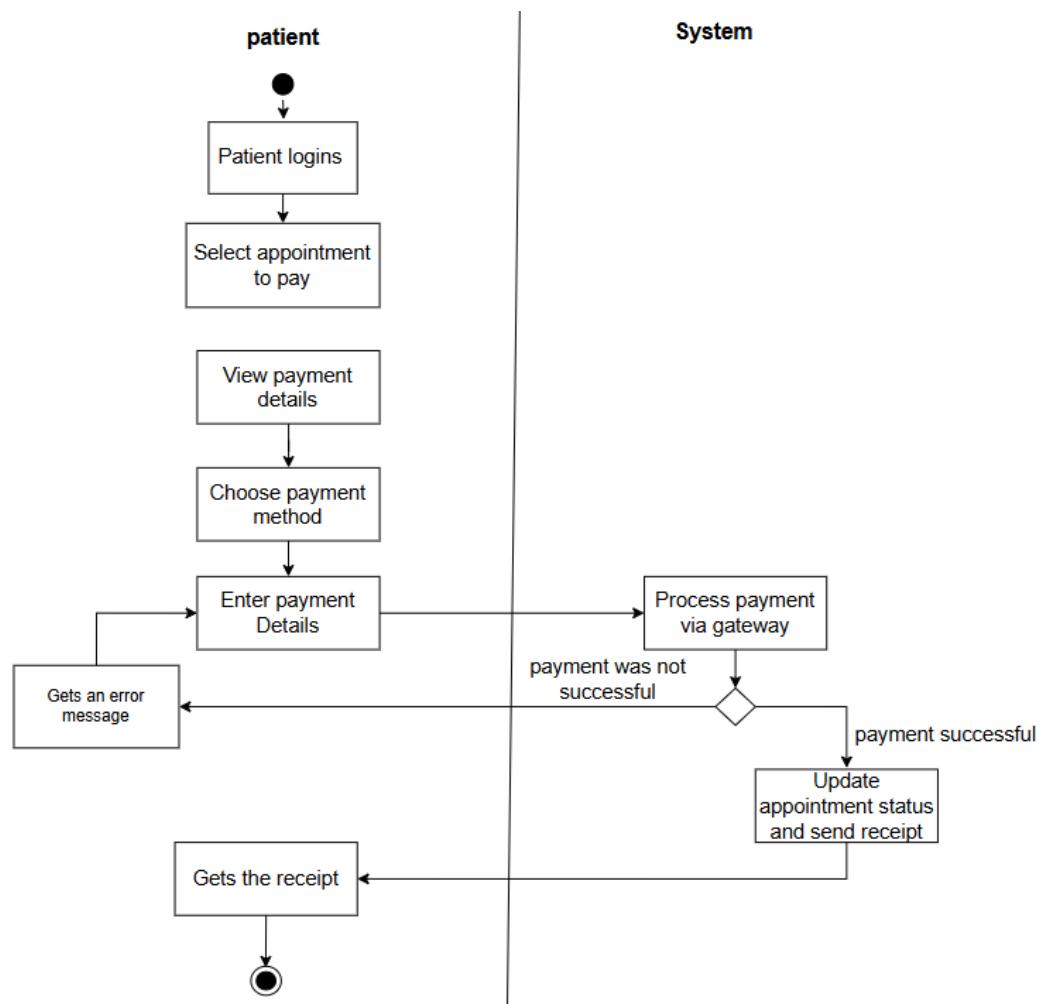
User stories

As a Patient, I want to make an online payment for my appointment.

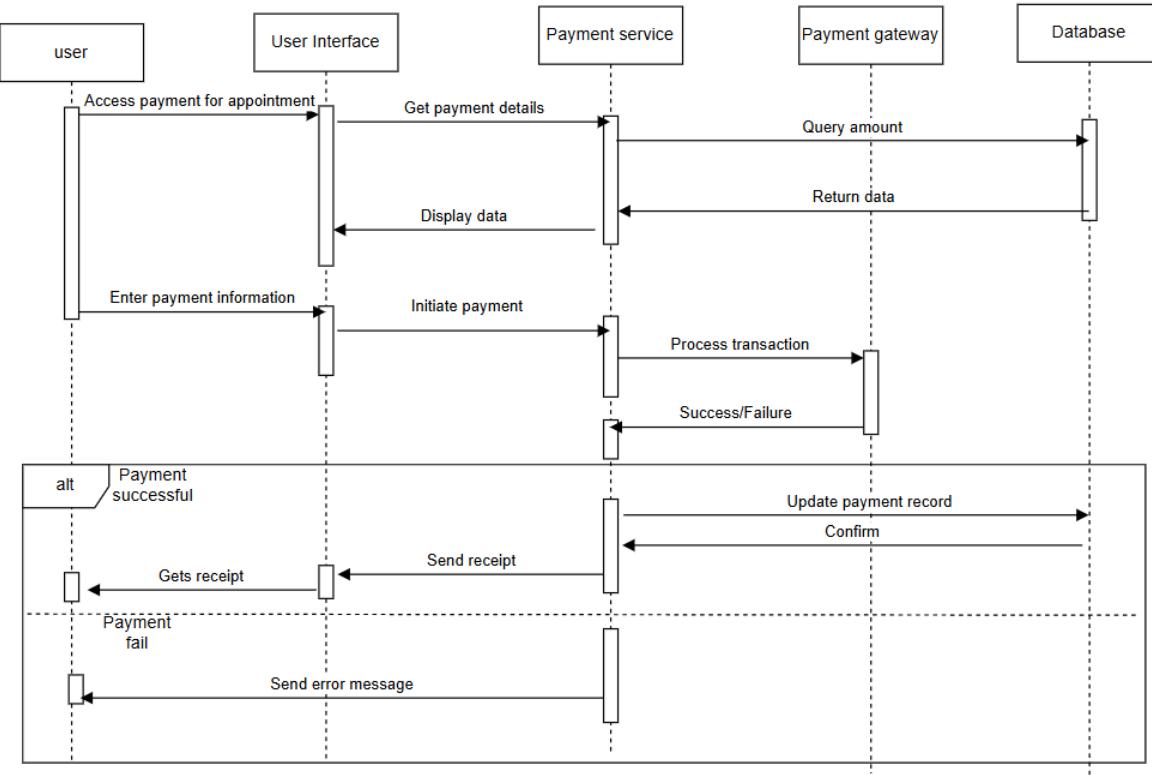
Use case diagram



Activity diagram



Sequence diagram



Implementation – Justification of tools used & how user pain points dealt with

Supabase Storage for Medical Records

We utilized **Supabase Storage**, which leverages **S3 Buckets**, to manage patient records and test results. This tool allows the system to store files securely and provides authorized users access via unique links. By digitizing these documents, we ensure **clinical continuity**, allowing doctors to review a patient's full medical history instantly rather than relying on physical paperwork that may be lost or delayed.

Google SMTP for Automated Notifications

Google SMTP was selected for our notification system because it provides a reliable and free service that is easy to integrate compared to other providers. This tool directly addresses the lecturer's feedback regarding the "**overloaded receptionist**" by automating appointment reminders and follow-up alerts. Automating these communications removes the need for manual outreach, significantly reducing the administrative burden on clinic staff.

Stripe for Secure Payment Integration

For the financial module, we integrated **Stripe**, as it is the only viable free-to-integrate option that offers a robust test environment. By offloading transaction processing to Stripe's secure infrastructure, we ensure that sensitive financial data is handled with industry-standard encryption. This directly addresses the instructor's requirement that "**Security needs to be thought of**" while simultaneously removing the billing bottleneck from the reception desk.

Evidence of testing

Test Case ID	Description	Test Procedure	Expected Result	Actual Result	Status	Comments
TC01	Create Patient Account (Valid)	1. (A/D) Logs in. 2. Navigates to "Create Patient Account". 3. Enters all valid patient details (e.g., Name, Email, Phone). 4. Submits the form.	System creates the account. A success message is shown. New patient record is stored in the database with a unique User ID.	System creates the account. A success message is shown. New patient record is stored in the database with a unique User ID.	Pass	

TC02	Create Doctor Account (Valid)	<p>1. (A) Logs in.</p> <p>2. Navigates to "Create Doctor Account".</p> <p>3. Enters all valid doctor details (e.g., Name, Specialty, Email).</p> <p>4. Submits the form.</p>	System creates the account. A success message is shown. New doctor record is stored in the database with a unique User ID.	System creates the account. A success message is shown. New doctor record is stored in the database with a unique User ID.	Pass	Doctor is created in DB but Doctor Cannot login
TC03	Create Account - Missing Required Fields	<p>1. (A/D) Logs in.</p> <p>2. Navigates to account creation.</p> <p>3. Leaves one or more required fields (e.g., Email, Password) blank.</p> <p>4. Submits the form.</p>	System does not create the account. A clear error message appears next to the missing field(s) prompting the user to fill them in. The form is not submitted	System does not create the account. The form is not submitted as the button is deactivated.	Pass	
TC04	Create Account - Invalid Email Format	<p>1. (A/D) Logs in.</p> <p>2. Navigates to account creation.</p> <p>3. Enters an invalid email (e.g., user@, userexample.com).</p> <p>4. Fills other valid details.</p> <p>5. Submits the form.</p>	System does not create the account. An error message indicates the email format is invalid.	System does not create the account. The form is not submitted as the button is deactivated.	Pass	

TC05	Create Account - Duplicate Username/Email	<p>1. (A/D) Logs in.</p> <p>2. Navigates to account creation.</p> <p>3. Enters a username or email that already exists in the database.</p> <p>4. Fills other valid details.</p> <p>5. Submits the form.</p>	System does not create the account. An error message indicates that the username/email is already taken.	System does not create the account. An error message indicates that the username/email is already taken.	Pass	
TC06	Create Account - Weak Password Policy	<p>1. (A/D) Logs in.</p> <p>2. Navigates to account creation.</p> <p>3. Enters a password that does not meet the policy (e.g., too short, no numbers).</p> <p>4. Fills other valid details.</p> <p>5. Submits the form.</p>	System does not create the account. An error message explains the password policy requirements (e.g., "Must be 8 characters, include a number and a symbol").	System does not create the account.	Pass	
TC07	Patient Login (Valid Credentials)	<p>1. (P) Navigates to the login page.</p> <p>2. Enters a valid username/email and password for a patient account.</p>	Credentials are valid. System authenticates the user and redirects to the Patient	Credentials are valid. System authenticates the user and redirects to the Patient	Pass	

		3. Clicks "Login".	Dashboard. The user's session is started.	Dashboard. The user's session is started.		
TC08	Doctor Login (Valid Credentials)	1. (D) Navigates to the login page. 2. Enters a valid username/email and password for a doctor account. 3. Clicks "Login".	Credentials are valid. System authenticates the user and redirects to the Doctor Dashboard. The user's session is started.	Credentials are valid. System authenticates the user and redirects to the Doctor Dashboard. The user's session is started.	Pass	
TC09	Admin Login (Valid Credentials)	1. (A) Navigates to the login page. 2. Enters a valid username/email and password for an admin account. 3. Clicks "Login".	Credentials are valid. System authenticates the user and redirects to the Admin Dashboard. The user's session is started.	Credentials are valid. System authenticates the user and redirects to the Admin Dashboard. The user's session is started.	Pass	
TC10	Login - Invalid Username	1. User navigates to the login page. 2. Enters an incorrect/non-existent username but a valid password. 3. Clicks "Login".	Credentials are not valid. System displays a generic message: "Wrong credentials" or "Invalid username or password."	Credentials are not valid. System displays a generic message: "Wrong credentials" or "Invalid username or password."	Pass	

			User is not logged in.	User is not logged in.		
TC11	Login - Invalid Password	1. User navigates to the login page. 2. Enters a valid username but an incorrect password. 3. Clicks "Login".	Credentials are not valid. System displays a generic message: "Wrong credentials" or "Invalid username or password." User is not logged in.	Credentials are not valid. System displays a generic message: "Wrong credentials" or "Invalid username or password." User is not logged in.	Pass	
TC12	Login - Both Fields Empty	1. User navigates to the login page. 2. Clicks "Login" without entering any data.	System displays validation errors for both fields, indicating they are required. The form is not submitted.	System displays validation errors for both fields, indicating they are required. The form is not submitted.	Pass	
TC13	Login - Password Field Empty	1. User navigates to the login page. 2. Enters a valid username but leaves the password field blank. 3. Clicks "Login".	System displays a validation error on the password field, indicating it is required.	System displays a validation error on the password field, indicating it is required.	Pass	
TC14	Login Case Sensitivity	1. User navigates to the login page. 2. Enters a valid username but with	System behavior should be consistent. Expected:	System behavior should be consistent. Expected:	Pass	

		<p>changed case (e.g., User instead of user).</p> <p>3. Enters the correct password.</p> <p>4. Clicks "Login".</p>	<p>Either login fails with "Wrong credentials" (if case-sensitive) or succeeds (if case-insensitive). The result should align with the system's design specification.</p>	<p>Either login fails with "Wrong credentials" (if case-sensitive) or succeeds (if case-insensitive). The result should align with the system's design specification.</p>		
TC15	Successful Login Redirects to Target Page	<p>1. User tries to access a protected page (e.g., /dashboard) while logged out.</p> <p>2. System redirects to login page.</p> <p>3. User enters valid credentials and logs in.</p>	<p>After successful authentication, the user is automatically redirected to the originally requested protected page (/dashboard), not just a generic home page.</p>	<p>After successful authentication, the user is automatically redirected to the originally requested protected page (/dashboard), not just a generic home page.</p>	Pass	
TC16	Session Management & Logout	<p>1. User logs in successfully.</p> <p>2. Navigates to a few different pages within the application.</p>	<p>User session is terminated. User is redirected to the login page or home page. If the user tries to use the</p>	<p>User session is terminated. User is redirected to the login page or home page. If the user tries to use the</p>	Pass	

		3. Clicks the "Logout" button.	browser's back button, they cannot access the protected pages.	the browser's back button, they cannot access the protected pages.		
TC17	Successful Appointment Booking	1. Patient logs in. 2. Navigates to "Book Appointment". 3. Selects a desired Doctor, Date, and available Time Slot. 4. Submits the booking request.	System confirms the appointment. A success message is shown (e.g., "Appointment Booked Successfully"). The appointment appears in the patient's and doctor's upcoming appointments.	Available Doctors are not displayed.	Fail	
TC18	Booking - No Slots Available	1. Patient logs in. 2. Navigates to "Book Appointment". 3. Selects a Doctor and Date where no slots are free.	The system displays a message: "No available slots for the selected date/doctor." The patient cannot proceed to book.	The system displays a message: "No available slots for the selected date/doctor." The patient cannot proceed to book.	Pass	
TC19	Booking - Mandatory Field Validation	1. Patient logs in. 2. Navigates to "Book Appointment".	The system does not submit the form. Clear error messages	Not Implemented	Fail	

		3. Attempts to submit the form without selecting a Doctor, Date, or Time.	prompt the user to fill in all required fields.			
TC20	Booking - Past Date Selection	1. Patient logs in. 2. Navigates to "Book Appointment". 3. Tries to select a date that is in the past.	The system validation prevents selection of past dates, either by greying them out or showing an error upon submission.	Not Implemented	Fail	
TC21	Booking Slot Becomes Unavailable	1. Patient selects an available slot. 2. Another patient books the same slot before the first patient submits. 3. First patient submits the booking.	The system detects the conflict and displays an error message: "This slot is no longer available. Please select another time."	Not Implemented	Fail	
TC22	Patient Views Upcoming Appointments	1. Patient logs in. 2. Navigates to "My Appointments" or Dashboard.	The system displays a list of the patient's upcoming appointments, showing details like Doctor, Date, Time, and Status.	Not Implemented	Fail	

TC23	Doctor Views Upcoming Appointments	1. Doctor logs in. 2. Navigates to "My Appointments" or Schedule.	The system displays a list of the doctor's upcoming appointments for the day/week, showing Patient Name, Time, and Type.	Not Implemented	Fail	
TC24	View Past Appointment History	1. Patient/Doctor logs in. 2. Navigates to Appointments. 3. Filters or navigates to "Past Appointments".	The system displays a list of successfully completed or past appointments.	Not Implemented	Fail	
TC25	View Appointment Details	1. User (Patient/Doctor) logs in and views their appointments list. 2. Clicks on a specific appointment.	The system displays detailed information about the appointment, including all relevant notes, patient history (for doctor), or doctor's bio (for patient).	Not Implemented	Fail	
TC26	View Appointments - No Appointments	1. A new patient/doctor with no bookings logs in. 2. Navigates to the appointments page.	The system displays a friendly message: "No upcoming appointments found."	Not Implemented	Fail	

TC27	Successful Rescheduling	<ol style="list-style-type: none"> 1. Patient logs in. 2. Views upcoming appointments. 3. Selects an appointment and clicks "Reschedule". 4. Selects a new, available Date and Time. 5. Confirms the change. 	System confirms the rescheduling. A success message is shown. The old slot is freed up, and the new appointment details are reflected in both the patient's and doctor's schedules.	Not Implemented	Fail	
TC28	Reschedule - No Slots Available	<ol style="list-style-type: none"> 1. Patient initiates rescheduling for an appointment. 2. On the booking page, finds no available slots for the desired new date/doctor. 	The system displays a message: "No available slots for the selected date/doctor." The patient can cancel the rescheduling or choose a different date.	Not Implemented	Fail	
TC29	Reschedule - Cancel Operation	<ol style="list-style-type: none"> 1. Patient initiates rescheduling. 2. On the slot selection page, clicks "Cancel" or the back button. 	The system aborts the rescheduling process and returns the user to the appointment details or list view without making any changes. The original	Not Implemented	Fail	

			appointment remains booked.			
TC30	Reschedule Past Appointment	1. Patient tries to reschedule an appointment that has already passed.	The system should not allow rescheduling. The "Reschedule" button is either hidden, disabled, or clicking it shows an error: "Cannot reschedule a past appointment."	Not Implemented	Fail	
TC31	Successful Cancellation	1. Patient logs in. 2. Views upcoming appointments. 3. Selects an appointment and clicks "Cancel". 4. Confirms cancellation in the	System confirms the cancellation. A success message is shown (e.g., "Appointment Cancelled"). The appointment	Not Implemented	Fail	

		confirmation dialog.	is removed from the patient's upcoming list and the slot is freed up in the doctor's schedule.			
TC32	Cancellation - Deny Confirmation	1. Patient initiates cancellation for an appointment. 2. A confirmation dialog appears ("Are you sure?"). 3. Patient clicks "No" or "Cancel".	The cancellation process is aborted. The confirmation dialog closes, and the appointment remains booked in the system.	Not Implemented	Fail	
TC33	Cancel Past Appointment	1. Patient tries to cancel an appointment that has already passed.	The system should not allow cancellation. The "Cancel" button is hidden, disabled, or shows an error: "Cannot cancel a past appointment."	Not Implemented	Fail	
TC34	Cancel Already Cancelled Appointment	1. Patient navigates to an appointment that is already in a "Cancelled" state. 2. Attempts to cancel it again.	The system does not allow the action. The "Cancel" button is not available for an appointment	Not Implemented	Fail	

			already marked as cancelled.			
TC35	Successful Login & Navigation	<p>1. Doctor goes to the login page.</p> <p>2. Enters valid credentials.</p> <p>3. Clicks "Login".</p> <p>4. Navigates to the "Scheduling" or "My Availability" section.</p>	Doctor is logged in and successfully lands on the scheduling/availability management page.	Doctor is logged in and successfully lands on the scheduling/availability management page.	Pass	
TC37	Set New Availability (Add Slots)	<p>1. Doctor logs in and navigates to the scheduling section.</p> <p>2. Initiates an edit (e.g., clicks "Edit Availability").</p> <p>3. Adds new available slots (e.g., selects a date and time range).</p> <p>4. Saves the changes</p>	System confirms the update. A success message like "Availability Updated" is shown. The new slots are added to the displayed list and are now bookable by patients.	System confirms the update. A success message like "Availability Updated" is shown. The new slots are added to the displayed list and are now bookable by patients.	Pass	
TC38	Update Existing Availability (Modify Slots)	<p>1. Doctor logs in and navigates to the scheduling section.</p> <p>2. Initiates an edit.</p> <p>3. Modifies an existing available</p>	System confirms the update. The modified slot is updated in the system. Any existing appointment in the old slot is handled	System confirms the update. The modified slot is updated in the system. Any existing appointment in the old slot is handled	Pass	

		<p>slot (e.g., changes the time).</p> <p>4. Saves the changes.</p>	<p>based on business rules (e.g., automatically cancelled or the doctor is warned).</p>	<p>based on business rules (e.g., automatically cancelled or the doctor is warned).</p>		
TC39	Delete Availability (Remove Slots)	<p>1. Doctor logs in and navigates to the scheduling section.</p> <p>2. Initiates an edit.</p> <p>3. Deletes one or more existing available slots.</p> <p>4. Saves the changes.</p>	<p>System confirms the update. The deleted slots are removed from the displayed list. They become unavailable for booking.</p>	<p>System confirms the update. The deleted slots are removed from the displayed list. They become unavailable for booking.</p>	Pass	
TC40	Cancel Edit Operation	<p>1. Doctor logs in and navigates to the scheduling section.</p> <p>2. Initiates an edit.</p> <p>3. Makes some changes (adds/modifies/deletes slots).</p> <p>4. Instead of saving, clicks "Cancel".</p>	<p>The system discards all unsaved changes. The availability display reverts to the state it was in before the edit was initiated. No changes are made to the availability.</p>	<p>The system discards all unsaved changes. The availability display reverts to the state it was in before the edit was initiated. No changes are made to the availability.</p>	Pass	
TC44	Save with No Changes	<p>1. Doctor initiates edit mode.</p>	<p>The system saves without error, displaying a</p>	<p>The system saves without error, displaying a</p>	Pass	

		<p>2. Makes no changes to the availability.</p> <p>3. Clicks "Save".</p>	<p>neutral message like "No changes to save" or simply returning to view mode. The availability remains unchanged.</p>	<p>neutral message like "No changes to save" or simply returning to view mode. The availability remains unchanged.</p>		
TC45	Successful Login & View Pending Requests	<p>1. Doctor navigates to the login page.</p> <p>2. Enters valid doctor credentials.</p> <p>3. Clicks "Login".</p> <p>4. Navigates to "Follow-up Requests" or similar section.</p>	<p>System authenticates the doctor and displays the dashboard. The "Follow-up Requests" page loads and shows a list of pending requests from patients, including patient name and relevant details.</p>	Not Implemented	Fail	
TC46	View Pending Requests - No Requests	<p>1. Doctor logs in.</p> <p>2. Navigates to the "Follow-up Requests" section.</p>	<p>The system displays a message such as "No pending follow-up requests." The list is empty.</p>	Not Implemented	Fail	
TC47	Reject Follow-up Request	<p>1. Doctor logs in and views pending follow-up requests.</p>	<p>The system removes the request from the "Pending"</p>	Not Implemented	Fail	

		<p>2. Selects a request and clicks a "Reject" button.</p> <p>3. Confirms the rejection in the dialog box.</p>	<p>list. A confirmation message is shown (e.g., "Follow-up request rejected."). The patient may be notified of the rejection.</p>		
TC48	Reject Request - Cancel Operation	<p>1. Doctor initiates rejecting a request.</p> <p>2. A confirmation dialog appears.</p> <p>3. Doctor clicks "Cancel" or closes the dialog.</p>	<p>The cancellation is successful. The dialog closes, and the follow-up request remains in the "Pending" list. No action is taken.</p>	Not Implemented	Fail
TC49	Action Persistence & List Update	<p>1. Doctor logs in and views pending requests.</p> <p>2. Performs an action (Accept, Reject, or Send Message) on a specific request.</p> <p>3. Refreshes the page or navigates back to the list.</p>	<p>The list is updated consistently. The request that was acted upon is no longer visible in the "Pending" list (for Accept/Reject) or is marked accordingly (for Message).</p>	Not Implemented	Fail
TC50	Unauthorized Access Block	<p>1. A user with a "Patient" role logs in.</p>	<p>The system blocks access, redirecting the user to their</p>	The system blocks access, redirecting the	Pass

		2. Attempts to access the doctor's follow-up requests page via URL (e.g., /doctor/follow-up-requests).	own dashboard or displaying a "Access Denied" error.	user to their own dashboard		
--	--	--	--	-----------------------------	--	--

A	B	C	D	E	F	G	H	I	J
Test Case	Descriptive Test Proce	Expected	Actual Res	Status	Comments				
[cite_start Upload Pa	1. Login as System ac	Pending Ex	Not Run						
TC_S2_02 Upload Pa	1. Login as System va	Pending Ex	Not Run		Verify loop back to file selection				
TC_S2_03 Upload Re	1. Comple Doctor rec	Pending Ex	Not Run		Integration check				
TC_S2_04 Upload Re	1. Login as Upload prc	Pending Ex	Not Run						
TC_S2_05 View Patie	1. [cite_st: System dis	Pending Ex	Not Run		Role-based access check				
TC_S2_06 View Patie	1. Login as System dis	Pending Ex	Not Run		Role-based access check				
TC_S2_07 Doctor Ad	1. Login as System sa	Pending Ex	Not Run						
TC_S2_08 Doctor Up	1. Login as System up	Pending Ex	Not Run						
TC_S2_09 Patient Vic	1. Login as List of prev	Pending Ex	Not Run						
TC_S2_10 Patient Vic	1. Login as List of pres	Pending Ex	Not Run						
TC_S2_11 Access Hi	1. Login as Access De	Pending Ex	Not Run		Security check				
TC_S2_12 View Paym	1. Login as System dis	Pending Ex	Not Run						
TC_S2_13 Make Onli	1. Select a Gateway p	Pending Ex	Not Run		Happy Path				
TC_S2_14 Make Onli	1. Initiate Gateway r	Pending Ex	Not Run		Unhappy Path				
TC_S2_15 Make Onli	1. Initiate System/G	Pending Ex	Not Run		Input validation				
TC_S2_16 Verify Rec	1. Comple Receipt is	Pending Ex	Not Run						
TC_S2_17 Payment S	1. Comple Status upc	Pending Ex	Not Run		Database integrity check				
TC_S2_18 Payment F	1. Perform All past tra	Pending Ex	Not Run						
TC_S2_19 Gateway T	1. Initiate System ha	Pending Ex	Not Run		Edge case				
TC_S2_20 Concurrer	1. Open pa System de	Pending Ex	Not Run		Concurrency check				

Sprint Retrospective

Developer –Lakindu Perera

What Went Well	What Went Wrong	What We Commit to Improve in Next Sprint
All the features from previous sprint implemented as well as a few from this sprint's.	One of the BA were not cooperative and did not deliver diagrams on time affecting development.	BA should work more closely with developer to ensure diagrams are delivered on time.
Agile methodology was better followed this sprint.	Communication among other members was poor as not all were able to attend daily standup at a time	Adjust meeting times, updates through text if not able to attend meeting.

Quality Assurance (QA) – Stefan Shabbir

What Went Well	What Went Wrong	What We Commit to Improve in Next Sprint
<ul style="list-style-type: none">• Standups were more consistent and informative• The SM was helpful and always reached out when meetings could not be held• Feature Relevant Diagrams were	<ul style="list-style-type: none">• Members were not as freely available compared to the last sprint• There was another rush to complete roles towards the sprint deadline	<ul style="list-style-type: none">• Better Clarity in role distribution and role requirements + deliverables• Try to plan out feature implementation in terms of complexity, starting with complex features being developed at

shared as they went		the start of the sprint
---------------------	--	-------------------------

Business Analyst (BA) – Ayodhya Prabashini

What Went Well	What Went Wrong	What We Commit to Improve in Next Sprint
The team already had experience working together from previous sprints, which helped maintain general coordination and understanding of the project workflow.	The other BA did not communicate difficulties early, despite being asked several times whether help was needed.	Encourage open and honest communication when team members face difficulties.
In Sprint 2, the Business Analyst (BA) role was shared between two members, including myself. I completed all diagrams and documentation assigned to me within the agreed timeline.	Poor attendance at daily stand-ups resulted in gaps in communication within the team.	Promote shared accountability while maintaining personal responsibility for assigned tasks.

Business Analyst (BA) – Abdul Muheed

What Went Well	What Went Wrong	What We Commit to Improve in Next Sprint
-----------------------	------------------------	---

Use case, sequence, and activity diagrams were completed clearly and helped the team understand system flow	Some tasks were started late, causing minor time pressure.	Start tasks earlier and set internal deadlines
Team collaboration was good during discussions and feedback on diagrams. Diagrams followed project requirements and were easy to present	Communication gaps at times slowed progress Limited time for reviewing and refining work	Improve regular communication and updates Allocate time for peer review and improvements

Scrum Master (SM) – Senithi Ranathunga

What Went Well	What Went Wrong	What We Commit to Improve in Next Sprint
Sprint 1 developer completed the missed paths at the start of Sprint 2	Communication can be improved. team members should inform if they fall behind.	Improve communication and transparency on progress.

We were able to finalize decisions on diagrams and feature processes before starting the sprint.
And meetings were more frequent compared to previous sprint

Having two BAs wasn't as effective, having two QAs might have been more helpful for this sprint.

Ensure roles/resources align better with sprint needs.