Planning and Requirements Analysis

Functional & Security Requirements

Stakeholder Roles

Stakeholder	Role	Data access	Data type
Doctor	Reviews patient data to generate medical insights and issues prescriptions to pharmacists	Personal details (name, password, email, phone number) Patient data	Confidential (personal data) Public (patient data)
Nurse	Reviews patient data and doctor reports to provide informed medical care	Personal details (name, password, email, phone number) Doctor reports	Confidential (personal details, doctor reports)
System Administrator	Managers application databases, servers, user account security	Encrypted user credentials, system logs, access control lists, database backups	Confidential (user credentials, system logs) Internal (access control lists, database backups)
Developer	Builds, maintains, and tests the application	Application code, test data, application and bug reports	Internal (system data), Public (general application information)
Data Scientist	Analyses patient data to support medical insights and predict the likelihood of a stroke	Patient data	Public

User Stories

User Login

Element	Description
User Story	“As a registered health professional user, I want to log in securely so I access patient data and insights.”
Use Case	Title: User Login Actor: Registered health professional Goal: Gain secure access to account

	Main Flow: 1. User enters username & password 2. System validates credentials 3. System prompts for MFA code 4. User enters code 5. System verifies and grants access Alternative Flow: Invalid password → error message → retry (maximum of 5 attempts)
Acceptance Criteria	• Login must only succeed with valid credentials • After 5 failed attempts, account locked for 10 minutes • All login communication uses HTTPS • MFA required for login validation • Successful login generates an audit log

User Registration

Element	Description
User Story	"As a health professional, I want to register on the application to access patient data and insights."
Use Case	Title: Register on system using registration form Actor: Health professional Goal: Securely manage all users of the system" Main Flow: 1. User navigates to registration form on system 2. User fills in the registration form with personal details (name, email, password) 3. User submits registration form 4. System validates entered credentials 5. System prompts MFA set up 6. User sets up MFA 7. System stores user details in the database so the user is registered. 8. System informs the user that registration is successful. Alternative Flow: 1. User navigates to registration form → User fills in registration form → User submits registration form → Personal details entered fail input validation → System informs the user that their details must pass the required input formats → User rewrites their details in the form → User resubmits form → System validates entered credentials → System prompts MFA set up → User sets up MFA→ System stores user details in the database so the user is registered → System informs the user that registration is successful.
Acceptance Criteria	• User inputs in registration form must pass input validation • User password must be at least 8 characters, contain at least one capital letter, and at least one special character

	• User email must be in the correct format • System must prompt MFA set up • User must set up MFA to be fully registered

Sharing Medical Reports

Element	Description
User Story	"As a doctor, I want to share medical reports with other healthcare professionals within the hospital on the system."
Use Case	Title: Sharing medical reports Actor: Doctor Goal: Securely share medical reports Main Flow: 1. Doctor requests to share medical reports to intended healthcare professional recipients 2. System prompts warning message that the data they are sharing is sensitive and asks if they wish to proceed 3. Doctor proceeds with request 4. System securely encrypts medical report before sending 5. System transmits medical report to recipients 6. System notifies recipient of sent medical reports Alternative Flow: Doctor clicks not to proceed send → doctor retries sending request
Acceptance Criteria	• System must encrypt medical reports before sending • System must always prompt warning message before sending

Patient and Stroke Prediction Dashboard

Element	Description
User Story	"As a healthcare professional, I want to view a dashboard of all patients and their stroke predictions."
Use Case	Title: View Patient Dashboard Actor: Healthcare Professional (doctor, nurse) Goal: View a dashboard of all patients and stroke predictions Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User clicks on Patient Dashboard on account options 4. System presents patient dashboard to the user Alternative Flow: User clicks on side panel of account → User locates and selects Patient Dashboard page → System presents Patient Dashboard
Acceptance Criteria	• System must take user to Patient Dashboard when selected • Patient Dashboard must present all patients and stroke predictions

Add a New Patient Record

Element	Description
User Story	"As a healthcare professional, I want to add a new patient record to the hospital database."
Use Case	Title: Add new patient record Actor: Healthcare Professional (doctor, nurse, pharmacist) Goal: Create and add a new patient record Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User clicks on 'Add New Patient' on account options 4. System presents page to add a new patient record 5. User fills in details about a patient and clicks option to add 6. System saves the new patient record to the database 7. System notifies user that the new patient record has been added. Alternative Flow: User is on patient dashboard page → User selects option to add a new patient on the Patient Dashboard page → System presents page to add a new patient
Acceptance Criteria	• System must take user to page to add a new patient when selected • System must record new patient record to the database • System must notify the user that the new patient record has been added

Edit or Delete Patient Records

Element	Description
User Story	"As a healthcare professional, I want to edit or delete patient records to follow GDPR policies."
Use Case	Title: Edit or delete patient records Actor: Healthcare Professional (doctor, nurse, pharmacist) Goal: Edit or delete patient record Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User navigates to patient dashboard page 4. User searches for the patient record they want to edit/delete 5. User clicks the option to edit or delete 6. If user selects edit, the system allows user to edit details of the patient record, otherwise the system prompts a message asking if they are sure they want to delete the record. 7. User edits or deletes the patient record accordingly. 8. If User edits a record they click save when they're done

	9. System saves the user's edits to the patient record to the database. 10. System alerts user that the changes to the record have been saved or that a record has been deleted successfully. Alternative Flow: User creates a new patient record → System notifies the user that the record has been saved → User clicks the option to edit or delete the record
Acceptance Criteria	• System must allow a user to edit a patient record • System must present a message asking if they are sure they want to proceed before deleting a record • System must notify the user that edits to a user patient record are saved when user clicks save • System must delete the patient record from the database and any backups within 30 days • System must save changes to a record in the database

Upload Medical Records

Element	Description
User Story	"As a doctor, I want to upload medical reports or test results for a patient."
Use Case	Title: Upload medical reports or test results to the system for a patient Actor: Doctor Goal: Securely upload medical reports or test results for a patient Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User selects option to upload a medical report 4. System presents page to upload a medical report 5. User uploads a medical report from their files 6. System scans file for malware 7. System confirms that the file is save 8. User searches for the patient associated to the medical report 9. User selects the patient to link them to the medical report 10. User selects option to save 11. System saves the medical report to the database and the link between the medical report and associated patient 12. System alerts user that the medical report has been saved successfully Alternative Flow: User is on patient dashboard page → User searches for a patient → User selects the patient → System presents option to add a medical report → User selects option to add a medical report → User uploads a medical report → User selects option to save → System alerts user that the medical report has been saved successfully

Acceptance Criteria	• System must allow the user to upload a medical report in appropriate file format (eg: PDF, CSV) • System must save the medical report to the database • System must save the link between the medical report and associated patient record • System must alert the user that the medical report has been uploaded successfully.

Manage User Accounts

Element	Description
User Story	"As a system administrator I want to view, edit, and delete user credentials on the system to manage all registered health professionals that use the application."
Use Case	Title: View, edit, add and delete user credentials on user database Actor: System Administrator Goal: Securely manage all users of the system" Main Flow: 2. System administrator securely logs into the system 3. System administrator views all users in user database 4. System administrator makes changes to users if required (editing user details or deleting user account) 5. System records the changes/removes user record from database if deleted 6. System notifies system administrator of changed saved successfully. Alternative Flow: System adds → User searches for a patient → User selects the patient → System presents option to add a medical report → User selects option to add a medical report → User uploads a medical report → User selects option to save → System alerts user that the medical report has been saved successfully
Acceptance Criteria	• System must validate the system administrator login with MFA • System must record all changes made by the system administrator

Doctor User Stories that were implemented in the application

Create Medical Records

Element	Description
User Story	"As a doctor, I want to create medical reports for patients to log their health status, medical test results, and stroke prediction score to prescribe preventative treatment."
Use Case	Title: Create medical reports for a patient Actor: Doctor

	Goal: Securely create medical reports assigning a specific patient to them. Main Flow: 1. User securely logs in to the system 2. System presents dashboard to the user 3. User selects option to create a medical report 4. System presents page to create a medical report 5. User fills out the fields to create a medical report and clicks 'Create Report' button to submit details 6. System validates the inputs 7. System saves the medical report to the database 8. System redirects the user to the dashboard 9. System confirms that the report has been saved via a flash message Alternative Flow: System presents page to create a medical report → User fills out the fields to create a medical report and clicks 'Create Report' button to submit details → System finds bad data entry → System presents an error message → User re-enters valid data → System validates the inputs → System saves the medical report to the database → System redirects the user to the dashboard → System confirms that the report has been saved via a flash message
Acceptance Criteria	• System must allow the user to enter data into fields of report form • System must save the medical report to the database • System must save the link between the medical report and associated patient record • System must alert the user that the medical report has been uploaded successfully.

View, Edit and delete medical records

Element	Description
User Story	"As a doctor, I want to manage my medical reports on the application, edit report details if necessary, and delete medical reports if necessary due to error or otherwise."
Use Case	Title: View, edit and delete medical reports on the system that the logged in doctor has created Actor: Doctor Goal: Securely manage medical reports on the system Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User selects option to view medical reports 4. System presents page to view medical reports 5. User can strictly view the reports they have created and select options to either edit or delete medical reports

	6. If the user chooses to edit a medical report, the system redirects them to the edit medical report page 7. User edits the necessary fields and clicks save changes button 8. System securely saves the changes to the medical report to the database 9. System redirects them to 'view reports' page 10. System confirms that the changes were saved with a flash message Alternative Flow: User is on 'view reports' page → User selects delete on a medical report → System presents a prompt asking if they're sure they want to delete the report → User confirms yes → System securely deletes the medical report from the database → System presents a flash message to confirm the report has been deleted
Acceptance Criteria	• System must allow the user to edit and delete medical report only the logged in doctor has created • System must save changes or perform deletions of medical reports to the database • System must only allow logged in doctors to view and make changes to medical reports. • System must alert the user that the medical report edits or deletions have been successful.

View and Search Patient records

Element	Description
User Story	"As a doctor I want to view patient records for medical reports and other hospital duties."
Use Case	Title: View and search patient records by patient ID Actor: Doctor Goal: Securely view and search patient records by patient ID as a logged in doctor Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User selects option to view patient records 4. System presents page to view patient records 5. User can navigate through patient records via pagination feature 6. User uses search box to search for patients with patient ID containing their input 7. System presents patient records with ID containing the digits in search

	8. User can navigate through the resulting patient records via pagination feature Alternative Flow: User is on 'view patients' page → User uses search box to search for patients with patient ID containing their input → System presents patient records with ID containing the digits in search → User clears search → System reloads original view patients page with no filter by patient ID
Acceptance Criteria	• System must only allow logged in doctors to view patient records • System must allow user to navigate through patients records via pagination feature • System must show patient records based on Patient ID search if user makes use of it

Administrators User Stories that were implemented in the application

View, Edit and delete doctor credentials

Element	Description
User Story	"As an administrator, I want to manage registered doctors on the application, edit their credentials if necessary, and delete redundant doctor records."
Use Case	Title: View, edit and delete doctor credentials on the system Actor: Administrator (admin) Goal: Securely manage doctor credentials on the system Main Flow: 1. User securely logs in to the system 2. System presents account to the user 3. User selects option to view doctor records 4. System presents page to view doctor records 5. User can view the records and select options to either edit or delete doctor records 6. If the user chooses to edit a doctor record, the system redirects them to the edit doctor credentials page 7. User edits the necessary fields and clicks save changes button 8. System securely saves the changes to the doctor record to the database 9. System redirects them to 'view doctors' page 10. System confirms that the changes were saved with a flash message Alternative Flow: User is on 'view doctors' page → User selects delete on a doctor record → System presents a prompt asking if they're sure they want

	to delete the doctor → User confirms yes → System securely deletes the doctor record from the database → System presents a flash message to confirm the doctor has been deleted
Acceptance Criteria	• System must allow the user to edit and delete doctor records • System must save changes or perform deletions of doctor records to the database • System must only allow logged in admins to view and make changes to doctor records. • System must alert the user that the doctor record edits or deletion have been successful.