



**AMERICAN INTERNATIONAL UNIVERSITY–
BANGLADESH (AIUB)**
Dept. of Computer Science
Faculty of Science and Technology

CSC3112: SOFTWARE ENGINEERING
Spring: 2024-2025
Section: [T]
Group No: 01

Project Proposal On
Project Name
[MediNexis]

Supervised By
TONNY SHEKHA KAR

Submitted By:

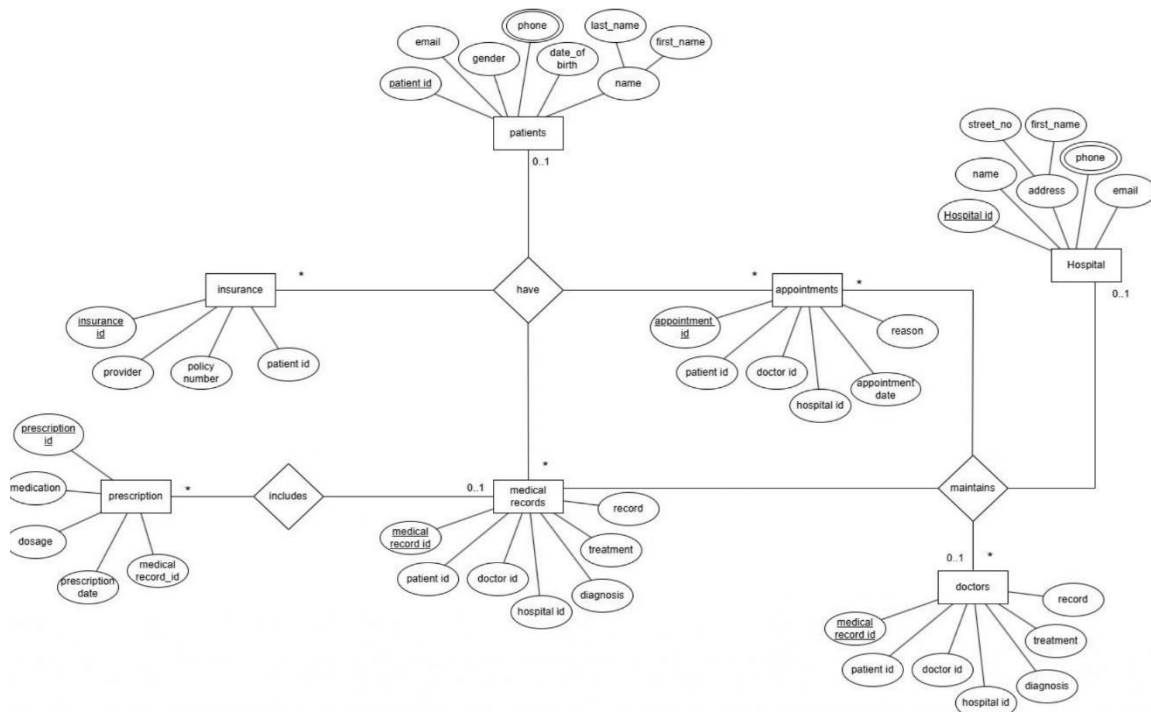
Name	ID
<i>1. Mozahidul Islam</i>	<i>23-51293-1</i>
<i>2. Sohik Wahid</i>	<i>23-51440-1</i>
<i>3. Md. Iftakhar Awal Chowdhury</i>	<i>23-51103-1</i>
<i>4. MD. Shamsul Alam</i>	<i>23-50735-1</i>
<i>5. Md Raquibur Rahman</i>	<i>22-47220-1</i>

MediNexis: A Comprehensive Medical Database

ER DIAGRAM DIAGRAM :

The diagram models a healthcare system with entities like **Patients**, **Doctors**, **Hospitals**, **Appointments**, **Medical Records**, **Prescriptions**, and **Insurance**.

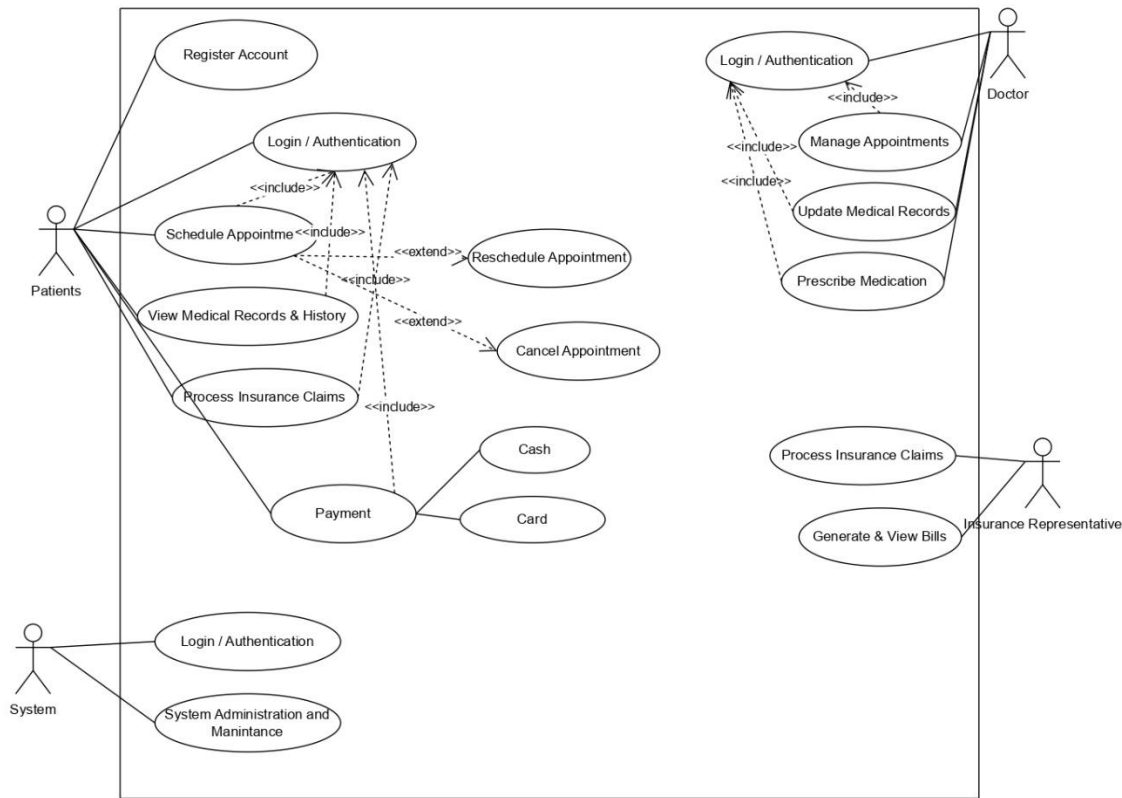
- ❖ **Patients** can have multiple **Appointments**, **Medical Records**, and **Insurance** entries.
- ❖ **Doctors** are associated with **Medical Records** and **Appointments** and are linked to **Hospitals**.
- ❖ **Medical Records** include **Diagnoses**, **Treatments**, and **Prescriptions**.
- ❖ **Prescriptions** list **Medications** and their **Dosage**.
- ❖ **Appointments** record the **date**, **reason**, and involved **Doctor** and **Hospital**.
- ❖ Each **Insurance** is linked to a **Patient** with provider and policy details.



USE CASE DIAGRAM :

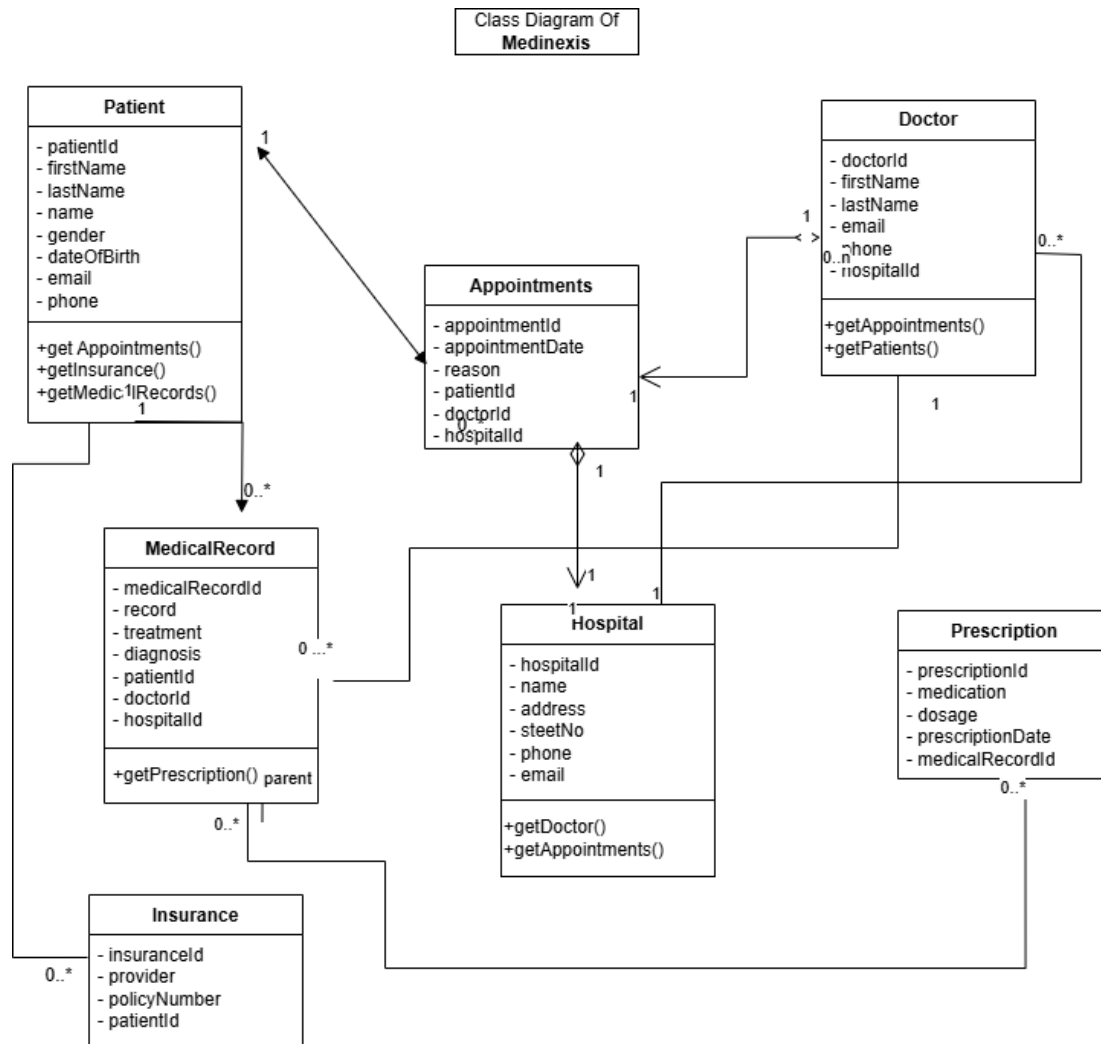
The diagram outlines interactions between users (Patients, Doctors, Insurance Representatives, and System Admins) and the system.

- ❖ **Patients** can register, log in, schedule/reschedule/cancel appointments, view medical records, make payments (cash/card), and process insurance claims.
- ❖ **Doctors** can log in, manage appointments, update medical records, and prescribe medications.
- ❖ **Insurance Representatives** handle insurance claims and generate/view bills.
- ❖ The **System** handles authentication and system maintenance.



CLASS DIAGRAM :

- ❖ **Patient:** Has personal details and can have multiple appointments, insurances, and medical records.
- ❖ **Doctor:** Linked to a hospital and can have multiple appointments and patients.
- ❖ **Appointments:** Connects patients, doctors, and hospitals with appointment details.
- ❖ **MedicalRecord:** Linked to a patient, doctor, and hospital; can have multiple prescriptions.
- ❖ **Prescription:** Contains medication details and is linked to a medical record.
- ❖ **Insurance:** Stores insurance details associated with a patient.
- ❖ **Hospital:** Hosts doctors, appointments, and medical records.



SEQUENCE DIAGRAM :

Registration & Authentication:

- ❖ Patient registers and logs in.
- ❖ System verifies insurance and stores patient data.

Appointment Scheduling:

- ❖ Patient schedules, reschedules, or cancels appointments.
- ❖ System communicates with the doctor and confirms actions.

Medical Consultation & Care:

- ❖ Patient visits doctor.
- ❖ Doctor updates medical records and stores them in the database.

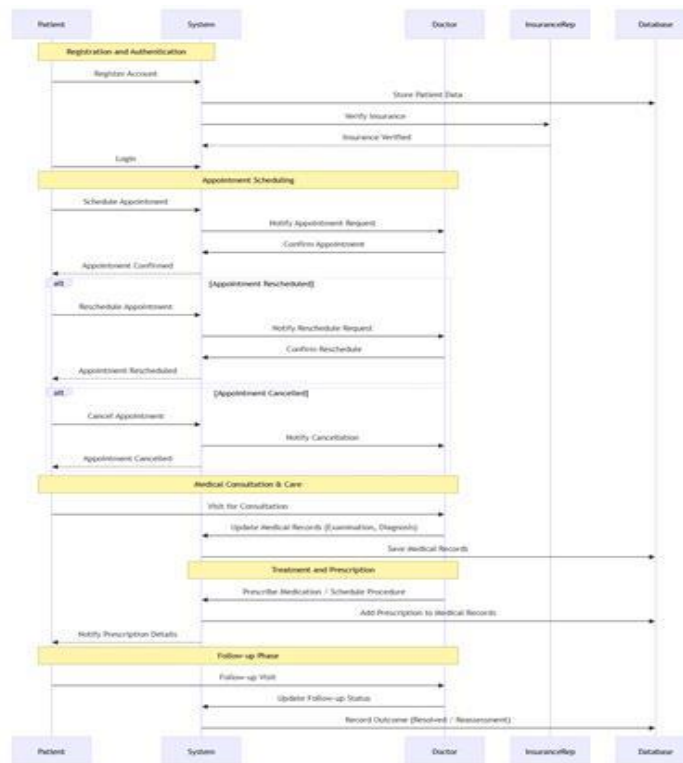
Treatment & Prescription:

- ❖ Doctor provides treatment and adds prescriptions to medical records.

Follow-up Plan:

- ❖ Patient follows up.

- ❖ System records outcomes and updates records.



STATE DIAGRAM :

Registration:

- ❖ A new patient signs up, creating data and verifying insurance.

Appointment Phase:

- ❖ The patient books an appointment (Scheduled → Confirmed → Completed).
- ❖ Appointments can be rescheduled or canceled.

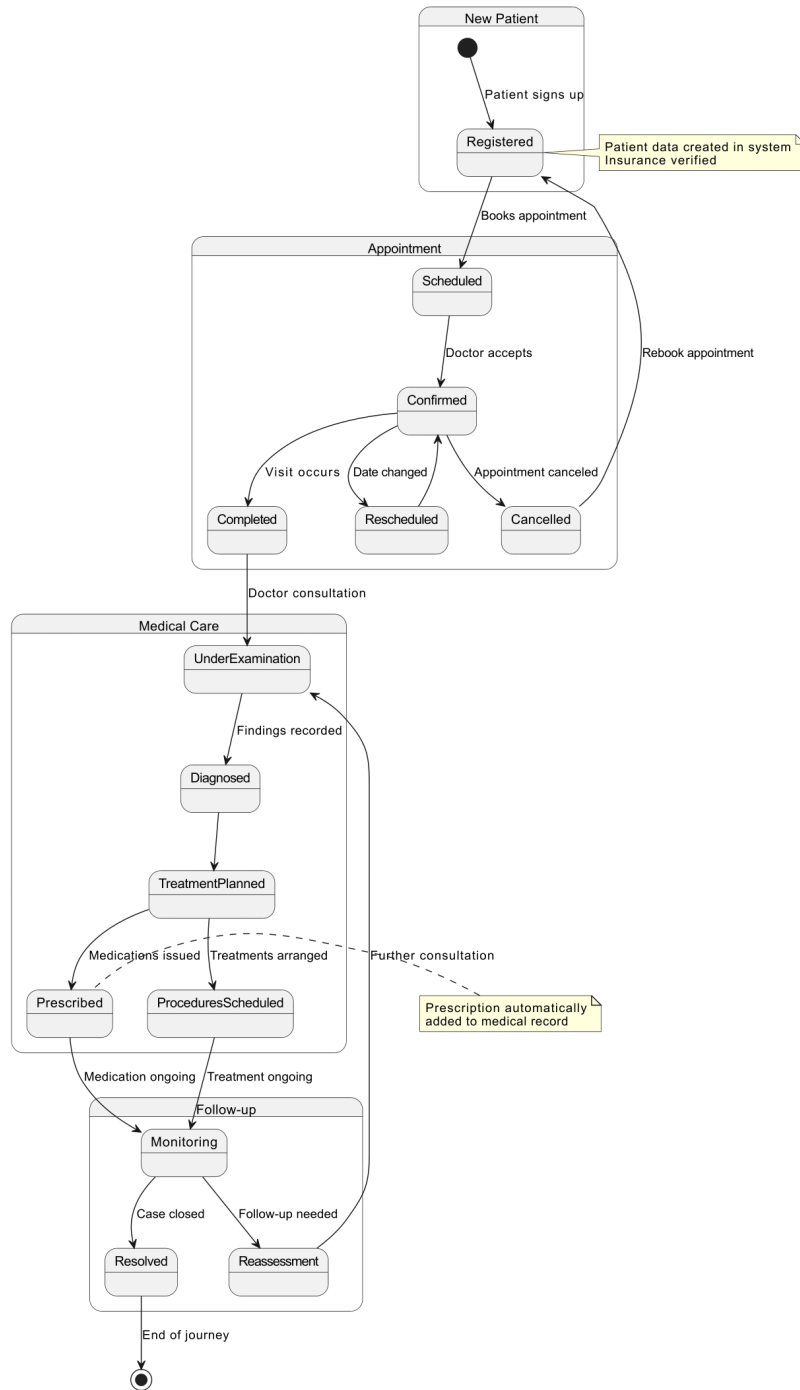
Medical Care:

- ❖ Patient undergoes examination, diagnosis, and treatment planning.
- ❖ Treatment can lead to medication (Prescribed) or procedures (Scheduled).

Follow-up and Monitoring:

- ❖ Treatment progress is monitored.
- ❖ Cases are either resolved or require reassessment.

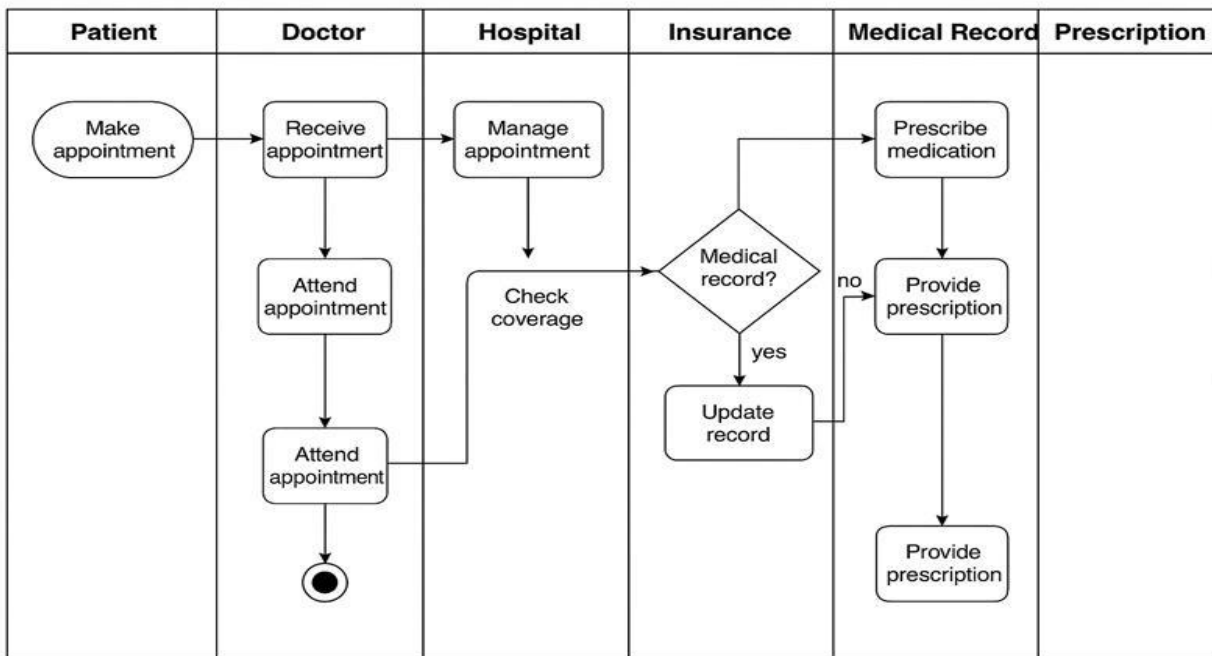
MediNexis Patient Journey State Diagram



ACTIVITY DIAGRAM :

The process starts with the patient making an appointment, which the doctor receives and later attends. The hospital manages the appointment and checks insurance coverage. The insurance provider checks if a medical record exists. If it does, they update the record; if not, it proceeds to prescription handling. The doctor prescribes medication, and the medical record system provides the prescription, which is then finalized in the Prescription lane.

The flow ensures collaboration between all parties involved in a healthcare visit, from appointment scheduling to prescription issuance, while verifying insurance and updating medical records accordingly.



Prepared by group 01.



American International University-Bangladesh

SOFTWARE ENGINEERING

Section: T

Project Title: MediNexis

Group No: 01

**Supervised By:
TONNY SHEKHA KAR**

Sl	Name	Id
1	Mozahidul Islam	23-51293-1
2	Sohih Wahid	23-51440-1
3	Md. Iftakhar Awal Chowdhury	23-51103-1
4	MD. Shamsul Alam	23-50735-1
5	MD RAQUIBUR RAHMAN	22-47220-1

Extreme Programming (XP) Process Model for MediNexis

1. Introduction

MediNexis is a centralized medical database designed to provide secure, real-time access to patient data across multiple healthcare institutions. Given its dynamic nature, rapid development needs, and emphasis on high-quality code, the Extreme Programming (XP) Model is the best choice for managing the software development lifecycle efficiently.

2. Why Choose XP for MediNexis?

XP is highly suitable for MediNexis due to:

- **Frequent Requirement Changes:** The healthcare industry evolves continuously, necessitating an agile and adaptive development process.
- **Security and Accuracy:** XP's test-driven development (TDD) ensures bug-free, secure code.
- **Collaborative Development:** XP promotes teamwork, making it ideal for a large-scale, multi-team project.
- **Continuous Integration:** Ensures real-time data synchronization across various medical institutions.
- **Customer Involvement:** Encourages direct feedback from stakeholders to refine features.

3. XP Process Model for MediNexis

3.1 Planning Phase

- **User Stories:** Define clear and concise user stories for each feature, such as secure patient records, AI-powered predictions, and real-time updates.
- **Release Planning:** Break down the project into multiple releases with iterative improvements.

3.2 Design Phase

- **Simple Design:** Focus on the most essential functionalities first to ensure scalability.
- **Refactoring:** Continuously improve code structure for maintainability.

3.3 Development Phase

- **Pair Programming:** Developers work in pairs to reduce errors and improve code quality.
- **Test-Driven Development (TDD):** Write tests before writing the actual code to ensure high reliability.
- **Continuous Integration:** Regularly integrate and test new features in the system.

3.4 Testing Phase

- **Unit Testing:** Each module is tested before integration.
- **Automated Testing:** AI-powered health predictions and patient data retrieval are verified using automated test suites.
- **User Acceptance Testing:** Healthcare professionals test the system to ensure usability.

3.5 Deployment & Maintenance

- **Frequent Releases:** Deploy incremental updates to the MediNexis system.
- **Customer Feedback:** Gather real-world feedback from hospitals and refine features accordingly.
- **Bug Fixes and Enhancements:** Continuously improve the system based on real-time user interactions.

4. How XP Benefits MediNexis

- **Ensures Rapid Development:** Quick iterations allow new features to be implemented efficiently.
- **Reduces Risk:** Frequent testing prevents major system failures.
- **Enhances Security:** Continuous feedback and code reviews ensure data integrity.
- **Encourages Collaboration:** Improves coordination among developers, testers, and medical professionals.

5. Conclusion

By implementing the XP model, MediNexis will achieve a scalable, secure, and adaptable healthcare database system. Its iterative approach will ensure that the software evolves efficiently to meet the needs of healthcare providers, ensuring high-quality service for patients and medical professionals alike.



American International University-Bangladesh

SOFTWARE ENGINEERING

Section: T

Project Title: MediNexis

Group No: 01

**Supervised By:
TONNY SHEKHA KAR**

Submitted by : Mozahidul Islam

Sl	Name	Id
1	Mozahidul Islam	23-51293-1
2	Mirza Wahidur Rahman	23-51440-1
3	Md. Iftakhar Awal Chowdhury	23-51103-1
4	MD. Shamsul Alam	23-50735-1
5	Md Raquibur Rahman	22-47220-1

RISK ANALYSIS :

Risks	Category	Probability	Impact	RMMM
Data Synchronization Failure	TE	10%	1	Strong data sync systems, add backup and recovery options, and set up alerts to quickly catch and fix any data issues.
Authentication Weaknesses	TE	20%	2	Strong encryption, multi-factor authentication. Conduct regular security reviews.
Limited UI Accessibility	DE	40%	3	Follow accessibility guidelines, conduct usability tests with users and get feedback to improve designs.
Test Case Gap	DE	20%	4	Design compact test cases, regularly review test coverage.
Time Constraint Risk	PS	60%	2	Break tasks into Small parts, Make priority list and monitor timelines sincerely. Adaptive schedules will manage delays efficiently.
Insufficient quality or quantity of data for AI model training	TE	50%	2	Early data sourcing and quality assessment, define clear data requirements, pilot AI features with sample data, explore data augmentation
Delays or difficulties in coordinating testing with real healthcare institutions	BU	60%	2	Establish clear MOUs and communication channels early, dedicate liaison personnel, plan

				buffer time for external dependencies.
Difficulty achieving real-time data synchronization across institutions	TE	20%	1	Implement defense-in-depth, regular security audits & penetration testing, strict key management, incident response plan.
Scalability Bottlenecks	TE	50%	3	It runs early hardware and software testing, use scalable architecture model and monitor performance to detect issues early. Occurrence plans are made for handling growth.
Not Enough funding problem	BU	40%	2	It manages multiple funding sources, calculate expenses and phase project rollouts. Financial tracking and emergency plans like debt, ensure stability.
Staff inexperienced	ST	30%	2	Provide prerequisite training, assign mentors and appreciate continuous learning. Performance tracking and essential support plans will help to gain skills.
Staff turnover will be high	ST	60%	2	upgrade work conditions, provide increment and ensure knowledge sharing.

Impact Values:

1 — Catastrophic

2 — Critical

3 — Marginal

4 — Negligible

Categories:

PS — Product Size

BU — Business Impact

CU — Customer Characteristics

PR — Process Definition

DE — Development Environment

TE — Technology to be Built

ST — Staff Size and Experience



**AMERICAN INTERNATIONAL UNIVERSITY–
BANGLADESH (AIUB)**
Dept. of Computer Science
Faculty of Science and Technology

CSC3112: SOFTWARE ENGINEERING
Spring: 2024-2025
Section: [T]
Group No: 01

Project Proposal On
Project Name
[MediNexis]

Supervised By
TONNY SHEKHA KAR

Submitted By:

Name	ID
<i>1. Mozahidul Islam</i>	<i>23-51293-1</i>
<i>2. Sohik Wahid</i>	<i>23-51440-1</i>
<i>3. Md. Iftakhar Awal Chowdhury</i>	<i>23-51103-1</i>
<i>4. MD. Shamsul Alam</i>	<i>23-50735-1</i>

MediNexis: A Comprehensive Medical Database

1. Introduction

1.1 Purpose

MediNexis is designed to provide a secure, real-time, centralized medical database to improve healthcare data management. It enables efficient patient record retrieval, real-time synchronization, and AI-powered analytics.

1.2 Scope

The system will facilitate:

- Centralized patient record management.
- Real-time data synchronization across healthcare institutions.
- AI-powered data analysis for predictive insights.
- Secure access control for sensitive data.
- Cross-platform integration with hospital management systems.

1.3 Definitions, Acronyms, and Abbreviations

- RBAC: Role-Based Access Control
- HIPAA: Health Insurance Portability and Accountability Act
- MFA: Multi-Factor Authentication

1.4 References

- Healthcare Data Regulations (HIPAA, GDPR)
 - MediNexis Project Proposal Document
-

2. Overall Description

2.1 Product Perspective

MediNexis will act as a cloud-based, secure database accessible by hospitals, clinics, and insurance companies, ensuring a single source of truth for patient records.

2.2 User Characteristics

- Doctors & Medical Staff: Access and update patient records.
- Hospitals & Clinics: Manage patient data securely.
- Patients: View medical history and prescriptions.

- **Insurance Companies:** Validate claims using patient records.

2.3 Assumptions and Dependencies

- A stable internet connection is required for real-time synchronization.
- Compliance with data privacy regulations.
- Hospitals must integrate with MediNexis API.

2.4 Software Development Methodology

MediNexis will follow the Agile Development Model, which ensures flexibility, iterative improvements, and continuous feedback from stakeholders. Agile allows:

- Rapid prototyping and incremental development.
 - Frequent releases to incorporate real-time user feedback.
 - Improved adaptability to evolving healthcare requirements.
-

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication and Access Control

- The system shall allow users to log in using a username and password.
- The system shall implement Multi-Factor Authentication (MFA) for secure login.
- The system shall use Role-Based Access Control (RBAC) to restrict access.

3.1.2 Patient Record Management

- The system shall allow authorized users to add, update, and delete patient records.
- The system shall maintain a complete history of patient interactions.

3.1.3 Real-Time Data Synchronization

- The system shall synchronize records across healthcare providers in real-time.
- The system shall prevent duplicate entries using unique patient identifiers.

3.1.4 AI-Powered Data Analysis

- The system shall use machine learning to provide predictive healthcare insights.
- The system shall generate early warning alerts for high-risk patients.

3.1.5 Secure Data Encryption

- The system shall encrypt patient data both in transit and at rest.
- The system shall comply with HIPAA and GDPR security standards.

3.1.6 System Notifications

- The system shall send email notifications for new prescriptions and test results.
- The system shall alert users when unauthorized access is detected.

Priority Levels:

- **High:** Authentication, Record Management, Security
 - **Medium:** AI Insights, Notifications
 - **Low:** Optional features like dark mode UI
-

3.2 Non-Functional Requirements

3.2.1 Performance Requirements

- The system shall retrieve patient records within 3 seconds on average.
- The system shall support at least 100,000 concurrent users.

3.2.2 Usability Requirements

- A trained user shall be able to submit a patient update request within 5 minutes.
- The UI shall be accessible for visually impaired users.

3.2.3 Security Requirements

- The system shall log out inactive users after 10 minutes.
- The system shall allow password reset only through multi-step verification.

3.2.4 Availability Requirements

- The system shall maintain 99.9% uptime.
 - A backup mechanism shall run every 6 hours.
-

3.3 Project Constraints

- The system must be developed within 6 months.
- Development must comply with regulatory healthcare data standards.
- The system must be tested with real healthcare institutions before launch.

4. Appendices

- **System Architecture Diagram**
- **Database Schema Overview**
- **Security Compliance Checklist**

Prepared by: Group 01, AIUB



American International University-Bangladesh

SOFTWARE ENGINEERING

Section: T

Project Title: MediNexis

Group No: 01

Supervised By:
TONNY SHEKHA KAR

Sl	Name	Id
1	Mozahidul Islam	23-51293-1
2	Sohih Wahid	23-51440-1
3	Md. Iftakhar Awal Chowdhury	23-51103-1
4	MD. Shamsul Alam	23-50735-1

Project Title :

MediNexis: A Comprehensive Medical Database

Background to the Problem

The healthcare sector generates a vast amount of data, including patient records, medical histories, test results, prescriptions, and treatment plans. However, most of this data remains fragmented across different hospitals, clinics, and medical institutions. This leads to:

- **Inefficiencies:** Medical staff often struggle to retrieve accurate patient data in a timely manner.
- **Misdiagnoses:** Lack of access to complete patient history can lead to incorrect treatments.
- **Duplicate Testing:** Incomplete data sharing results in unnecessary and costly medical tests.
- **Data Loss:** Poor data management systems can lead to data corruption or loss, jeopardizing patient care.

Existing medical databases are either outdated, poorly integrated, or lack real-time synchronization. Moreover, many healthcare institutions still rely on paper-based systems, further complicating data access and management.

The root cause of these issues is the absence of a unified, intelligent medical database capable of real-time synchronization, secure access, and efficient data analysis.

Why is This Problem Important ?

Healthcare data is one of the most critical and sensitive types of information. Efficient data management directly impacts the quality of care, patient safety, and operational efficiency. Without an integrated medical database, healthcare providers face several challenges:

- **Delayed Treatment:** Slow access to patient data increases response time during medical emergencies.
- **Data Breaches:** Poorly managed systems are vulnerable to unauthorized access and data leaks.
- **High Operational Costs:** Manual data entry, correction, and retrieval increase administrative burden and costs.
- **Legal and Compliance Issues:** Mismanagement of sensitive patient data can lead to legal penalties and loss of trust.

A modern medical database solution is essential to ensure data accuracy, enhance treatment efficiency, and improve patient outcomes.

Project Objective & Solution

MediNexis is designed to be a centralized medical database that ensures secure, real-time access to patient data across multiple healthcare institutions. It will provide healthcare professionals with a single source of truth for patient records, treatment plans, prescriptions, and medical history.

Key Features

1. Centralized Patient Record Management

- Technology: A secure, cloud-based database that consolidates patient data from multiple healthcare providers.
 - Impact: Doctors and medical staff can access complete patient histories, ensuring accurate diagnosis and treatment.
-

2. Real-Time Data Synchronization

- Technology: Real-time synchronization using secure APIs and cloud infrastructure.
 - Impact: Any updates made to a patient's record will instantly reflect across all connected platforms.
-

3. AI-Powered Data Analysis

- Technology: Machine learning models to analyze patient data for predictive insights and risk assessment.
 - Impact: Improved early diagnosis, better treatment recommendations, and reduced medical errors.
-

4. Secure Access Control & Encryption

- Technology: Role-based access control (RBAC), multi-factor authentication (MFA), and data encryption.
 - Impact: Protects sensitive medical data from unauthorized access and cyberattacks.
-

5. Efficient Data Retrieval

- Technology: Fast search algorithms and indexing for quick retrieval of patient records.
 - Impact: Reduces time spent on data searches, improving overall efficiency.
-

6. Cross-Platform Integration

- Technology: Integration with hospital management systems, diagnostic labs, and insurance platforms.
 - Impact: Ensures that all healthcare providers have consistent and updated patient data.
-

Project Weakness & Improvement

1. Data Privacy and Regulatory Compliance

- Weakness: Handling sensitive patient data requires strict compliance with healthcare regulations (e.g., HIPAA).
 - Improvement: Implement regular security audits and comply with global healthcare data standards.
-

2. Scalability and Performance

- Weakness: High data volume may affect system performance during peak usage.
 - Improvement: Use cloud-based infrastructure with automatic scaling to handle increased demand.
-

3. System Complexity for Small Clinics

- Weakness: Smaller healthcare providers may face difficulties in adopting complex systems.
 - Improvement: Develop a simplified interface and provide training sessions for small healthcare facilities.
-

4. Dependence on Network Connectivity

- Weakness: Internet downtime can disrupt access to real-time data.
 - Improvement: Introduce offline data caching and synchronization upon reconnection.
-

Target Users & Benefits

1. Hospitals & Clinics

- Benefit: Instant access to complete patient records, improving treatment accuracy.
 - Value: Faster diagnosis, reduced medical errors, and improved operational efficiency.
-

2. Doctors & Medical Staff

- Benefit: AI-based insights and predictive analysis improve clinical decision-making.
 - Value: Enhanced medical outcomes and reduced workload.
-

3. Patients

- Benefit: Faster and more accurate diagnosis and treatment.
 - Value: Better healthcare experience and increased trust in the system.
-

4. Laboratories & Diagnostic Centers

- Benefit: Direct access to patient history for better interpretation of test results.
 - Value: Reduced errors and faster reporting.
-

5. Insurance Companies

- Benefit: Accurate medical records for processing claims and preventing fraud.
 - Value: Faster claim settlement and improved customer satisfaction.
-

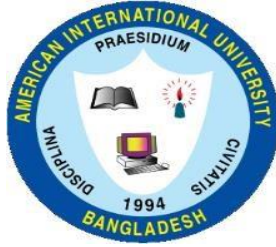
Impact of MediNexis

By implementing MediNexis, the healthcare industry will benefit from:

- ✓ Improved Patient Care: Centralized records enable faster and more accurate treatment.
 - ✓ Reduced Administrative Burden: Automated data entry and retrieval improve operational efficiency.
 - ✓ Enhanced Data Security: Advanced encryption and access control prevent unauthorized access.
 - ✓ Predictive Healthcare: AI-powered insights will help identify potential health risks early.
 - ✓ Scalable System: Cloud-based infrastructure ensures that the system can handle growing demand.
-

Conclusion

MediNexis aims to revolutionize medical data management by providing a unified, secure, and real-time database solution for healthcare providers. By centralizing patient records, enhancing data security, and leveraging AI for predictive analysis, MediNexis will improve healthcare delivery, reduce operational costs, and strengthen patient trust. This comprehensive medical database will serve as the backbone for a modern, data-driven healthcare system, ensuring better health outcomes for all.



American International University-Bangladesh

SOFTWARE ENGINEERING

Section: T

Project Title: MediNexis

Group No: 01

**Supervised By:
TONNY SHEKHA KAR**

Submitted by : Mozahidul Islam

Sl	Name	Id
1	Mozahidul Islam	23-51293-1
2	Mirza Wahidur Rahman	23-51440-1
3	Md. Iftakhar Awal Chowdhury	23-51103-1
4	MD. Shamsul Alam	23-50735-1
5	Md Raquibur Rahman	22-47220-1

Test Plan for MediNexus

Version: 1.0

Prepared by: QA Team (MD. Mozahidul Islam , Md. Iftakhar Awal Chowdhury, Md. Shamsul Alam, MD Raquibur Rahman, Mirza Wahidur Rahman)

Date: 14-May-2025

1. Introduction

This document outlines the testing strategy for MediNexus, a healthcare management system. The goal is to ensure functionality, security, and usability across key modules like User Authentication, Patient Record Management, Appointments, Notifications, Payments, and Security and the total test case number is 38.

2. Test Objectives

- Validate login, session management, and security features.
- Ensure patient records can be added, updated, and deleted securely.
- Verify appointment booking, rescheduling, and cancellation.
- Test payment processing, notifications, and two-factor authentication (2FA).
- Confirm data encryption and role-based access control.

3. Scope

In-Scope Modules:

- User Authentication (Login, Logout, 2FA, Password Change)
- Patient Record Management (CRUD operations)
- Appointments & Scheduling
- Notifications (Alerts, SMS)
- Payments (Transactions, Methods)
- Security (Encryption, Session Timeout)

Out-of-Scope:

- Third-party API integrations (e.g., insurance providers).
 - Hardware compatibility testing.
-

4. Test Approach

Testing Levels:

1. Unit Testing: Developers test individual components (e.g., login function).
2. Integration Testing: Verify interactions between modules (e.g., appointment → notification).
3. System Testing: End-to-end validation (e.g., patient record creation → payment).
4. Acceptance Testing: UAT by stakeholders.

Techniques:

- Black-Box: Functional testing (e.g., login with valid/invalid credentials).
- White-Box: Code review for security (e.g., data encryption).

5. Test Environment

- OS: Windows 11, macOS
- Browsers: Chrome, Firefox, Edge
- Mobile: Android, iOS
- Database: MySQL (encrypted)

Test Case 1:

Project Name: MediNexus			Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-1.1			Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): High			Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: User Authentication			Test Execution date: 11-May-2025		
Test Title: Verify login with valid username and password					
Description: Test successful login to the system using correct credentials.					
Precondition (If any): User has a valid user ID and password.					
Test Steps		Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Navigate to the login page 2. Enter username 3. Enter password 4. Click the “Sign In” button	Username: atif Password: 123atif	User is successfully logged into the application	As expected,	Pass
Post Condition: User is logged into the system. Session is active.				

Test Case 2:

Project Name: MediNixus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-1.2		Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: User Authentication		Test Execution date: 11-May-2025		
Test Title: Verify login attempt with invalid username or password				
Description: Test system response to an invalid username or password.				
Precondition (If any): User has a valid user ID and password.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the login page 2. Enter invalid username 3. Enter invalid password 4. Click the “Sign In” button	Username: xyz Password: x123x	An appropriate error message (e.g., "Invalid username or password") is displayed. User is not logged in.	As expected,	Pass
Post Condition: User remains on the login page. No session is created.				

Test Case 3:

Project Name: MediNixus	Test Designed by: Md. Iftakhar Awal Chowdhury
Test Case ID: FR-2.1	Test Designed date: 11-May-2025
Test Priority (Low, Medium, High): High	Test Executed by: Md. Iftakhar Awal Chowdhury
Module Name: Patient Record Management	Test Execution date: 11-May-2025
Test Title: Verify adding a new patient record	
Description: Test the functionality to add a new patient record by an authorized user.	

Precondition (If any): User is authenticated and has "Add Patient" access rights (e.g., Doctor role).				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an authorized user. 2. Navigate to the 3. "Add Patient Record" page. 4. Fill in all mandatory patient details. 5. Click "Save".	<ul style="list-style-type: none"> Patient Name: Asif Nazrul DOB: 1990-01-01 Patient ID: P123456789 	New patient records are successfully created and saved in the system. A confirmation message is displayed. The record is searchable.	As expected,	Pass
Post Condition: New patient record exists in the database.				

Test Case 4:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR-3		Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Md. Shamsul Alam		
Module Name: Notifications		Test Execution date: 11-May-2025		
Test Title: Verify user can view and interact with notifications				
Description: Test the functionality to get a new notification.				
Precondition (If any): User is logged in and system-generated notifications exist.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Login as a user 2.Trigger an event (e.g., book an appointment) 3.Open the notification panel	<ul style="list-style-type: none">Username: atifPassword: 123atif	Notification appears with correct message and information	As expected,	Pass
Post Condition: New notification appears in the notification panel.				

Test Case 5:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR-4.1		Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Shamsul Alam		
Module Name: Doctor's Appointment / Booking		Test Execution date: 11-May-2025		
Test Title: Verify booking appointment with available doctor				
Description: Test the functionality to add an appointment for a doctor				
Precondition (If any): User filled up all information and created the appointment card				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<ul style="list-style-type: none">Navigate to appointment pageSelect DoctorSelect dateSelect timeClick “Book Schedule”	<ul style="list-style-type: none">Doctor name: Dr. David H. Browndate: "2025-05-15"time: "5:00 PM"Click “Book Schedule”	Appointment confirmed message shown and also get a notification	As expected,	Pass
Post Condition: Appointment record created				

Test Case 6:

Project Name: MediNexus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR- 4.2		Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Shamsul Alam		
Module Name: Two-Factor Verification		Test Execution date: 11-May-2025		
Test Title: Verify two-factor verification process				
Description: User has entered valid login credentials and the two-factor authentication (2FA) feature is enabled for their account.				
Precondition (If any): User has entered valid login credentials and the two-factor authentication (2FA) feature is enabled for their account.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
	<ul style="list-style-type: none">• Username: atif• Password: 123atif	User is authenticated and taken to dashboard	As expected,	Pass

<ul style="list-style-type: none"> • Enter correct credentials • Receive verification code • Enter code • Click continue 	<ul style="list-style-type: none"> • Verification Method: In-app code entry (OTP) • Sent OTP Code: 482 • Entered OTP Code: 482 • OTP Expiry Time: 3 minutes 			
Post Condition: Login session active				

Test Case 7:

Project Name: MediNexus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR- 4.3		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Md. Shamsul Alam		
Module Name: View option		Test Execution date: 13-May-2025		
Test Title: OPD – View Option				
Description: Ensure that authorized users can successfully view detailed information of a selected patient from the patient list.				
Precondition (If any): User is logged in with appropriate privileges (Doctor/Admin) and patient records exist in the system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in with authorized credentials 2. Navigate to “Patient List” 3. Locate and click the “View” icon/button next to a patient record 4. Click “Close” or navigate back	Username: drbrown Password: test123 Patient ID: P123456789 Name, DOB, Gender, Address, Medical History, etc.	All information is correctly shown	As expected,	Pass
Post Condition: User has successfully viewed a patient’s detailed information.				

Test Case 8:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR- 5		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Shamsul Alam		
Module Name: back button navigates		Test Execution date: 13-May-2025		
Test Title: Verify the back button navigates				
Description: Ensure that clicking the "Back" button from the patient detail view successfully redirects the user to the patient list page without refreshing or losing state.				
Precondition (If any): User is logged in and has accessed a patient's detail view from the patient list.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in with authorized credentials 2. Go to “Patient List” 3. Click the “View” button next to a patient 4. Click the “Back” button on the detail view page 5. Ensure patient list is still displayed correctly	Username: drbrown Password: test123 Patient ID: P123456789	User is returned to the Patient List page after clicking back button	As expected,	Pass
Post Condition: User is back on the Patient List view with no data loss or page reload.				

Test Case 9:

Project Name: MediNixus	Test Designed by: Md. Shamsul Alam
Test Case ID: FR-1.1	Test Designed date: 13-May-2025
Test Priority (Low, Medium, High): High	Test Executed by: Md. Shamsul Alam
Module Name: Change Password	Test Execution date: 13-May-2025
Test Title: Verify user can change password successfully	

Description: Ensure that users can securely update their password by entering their current password and confirming the new password.				
Precondition (If any): User is logged in and navigated to the “Change Password” section in Account Settings.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to Account Settings > Change Password 2. Enter current password 3. Enter new password 4. Confirm new password 5. Click “Update Password”	1.CurrentPass:123 2. NewPass456	Confirmation matches and Confirmation message: “Password changed successfully”	As expected,	Pass
Post Condition: User’s password is updated in the system. Future logins require the new password.				

Test Case 10:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR- 4.4		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Shamsul Alam		
Module Name: Delete Account		Test Execution date: 13-May-2025		
Test Title: Verify user account can be deleted securely				
Description: Ensure that a user can delete their account after confirmation and that the system removes associated data as per policy.				
Precondition (If any): User is logged in and navigated to the "Delete Account" section.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to Account Settings > Delete Account 2. Click “Delete Account” button	1. Click "Yes, delete my account" 2. Use deleted credentials	Account is deleted and user logged out and when tried to login that account system shows	As expected,	Pass

3. Confirm deletion		Login fails with “Account does not exist”		
4. Try to log in again				
Post Condition: User account is permanently deleted from the system.				

Test Case 11:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR-4.5		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Md. Shamsul Alam		
Module Name: User Authentication – Logout		Test Execution date: 13-May-2025		
Test Title: Verify user can log out from the system				
Description: Ensure that clicking the logout button ends the session and redirects the user to the login screen.				
Precondition (If any): User is logged in.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Click the profile/user menu 2. Click “Logout” 3. Click browser back button	1. Username: atif 2. Password: 123atif	User is logged out and redirected to login page	As expected,	Pass
Post Condition: User session is terminated. Login is required to access any protected feature.				

Test Case 12:

Project Name: MediNixus		Test Designed by: Md. Shamsul Alam		
Test Case ID: FR- 5.1		Test Designed date: 11-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Shamsul Alam		
Module Name: Lab Work Management		Test Execution date: 11-May-2025		
Test Title: Verify lab work details can be added and viewed correctly				
Description: Ensure that authorized medical staff (doctor/lab technician) can successfully add lab reports and view them under a specific patient record.				
Precondition (If any): User is logged in with proper role permissions (e.g., Lab Technician, Doctor). A patient record exists in the system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in with authorized user account 2. Navigate to "Lab Work" or “Lab Reports” section	1. Username: labtech1 Password: lab@123 2. Patient ID: P001	Newly added report is listed with correct details	As expected,	Pass

3. Click “Add Lab Report” 4. Click “Submit”	Test Name: CBC Date: 2025-05-10 Result: Normal Remarks: No infection found			
Post Condition: Lab report is saved in the system and viewable under the associated patient’s profile.				

Test Case 13:

Project Name: MediNixus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-2.2		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: Patient Record Management		Test Execution date: 13-May-2025		
Test Title: Verify updating an existing patient record				
Description: Test the functionality to update an existing patient record by an authorized user.				
Precondition (If any): User is authenticated and has "Update Patient" access rights. An existing patient record is available.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as an authorized user. 2. Search for and open an existing patient record. 3. Modify one or more fields (e.g., address). 4. Click "Save" or "Update Patient".	Existing Patient ID: P000000001 Field to update: Address New Address: 123 New Street	The patient record is successfully updated with the new information. A confirmation message is displayed.	As expected,	Pass
Post Condition: Patient record reflects the updated information.				

Test Case 14:

Project Name: MediNexus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-2.3		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: Patient Record Management		Test Execution date: 13-May-2025		

Test Title: Verify deleting an existing patient record				
Description: Test the functionality to delete an existing patient record by an authorized user.				
Precondition (If any): User is authenticated and has "Delete Patient" access rights. An existing patient record is available.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<ol style="list-style-type: none"> 1. Log in as an authorized user. 2. Search for and open an existing patient record to be deleted. 3. Click the "Delete Patient" button/option. 4. Confirm deletion if a prompt appears. 	Existing Patient ID: P000000001	The patient record is successfully deleted from the system (or marked as inactive, depending on soft/hard delete policy). A confirmation message is displayed. The record is no longer searchable as active.	As expected,	Pass
Post Condition: Patient record is no longer active/available in the system.				

Test Case 15:

Project Name: MediNexus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-5.1		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: Security / Data Storage		Test Execution date: 13-May-2025		
Test Title: Verify encryption of patient data				
Description: Test that patient data stored on the server/database is encrypted.				
Precondition (If any): Patient data is stored in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Create/update a patient record. 2. (Backend step) Access the database directly (with appropriate authorization and tools). 3. Attempt to view the stored patient data fields (e.g., PII, health conditions).	Patient ID: P_EncryptionTest_01	Sensitive patient data fields in the database are in an encrypted format and not human-readable directly from the database tables without decryption keys.	As expected,	Pass

Post Condition: Data remains encrypted in the database.

Test Case 16:

Project Name: MediNixus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-5		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: Session Management		Test Execution date: 13-May-2025		
Test Title: Verify automatic logout after inactivity				
Description: Test that the system automatically logs out an inactive user after 10 minutes.				
Precondition (If any): User is logged into the system.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in to the system. 2. Leave the session idle for more than 10 minutes. 3. Attempt to perform an action that requires an active session.	N/A	The user is automatically logged out. Attempting an action redirects to the login page or shows a "session expired" message.	As expected,	Pass
Post Condition: User is logged out. Session is terminated.				

Test Case 17:

Project Name: MediNixus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-6		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: System Notifications / Security		Test Execution date: 13-May-2025		
Test Title: Verify admin alert for unauthorized access attempts				
Description: Test that administrators are alerted when multiple failed login attempts (suspicious activity) are detected for an account.				
Precondition (If any): Administrator account is configured to receive security alerts. Threshold for failed attempts is set (e.g., 5 attempts).				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Attempt to log in to a valid user account	Username: existing user	An alert is generated and sent to the system	As expected,	Pass

with an incorrect password multiple times (exceeding the threshold).				
2. Check the administrator's alert dashboard or designated notification channel.	Password: incorrect_password (repeatedly)	administrator(s) regarding the suspicious login activity, including details like username, IP address (if logged), and timestamp.		
Post Condition: Suspicious activity is logged. Account may be temporarily locked based on security policy.				

Test Case 18:

Project Name: MediNexus		Test Designed by: Md. Iftakhar Awal Chowdhury		
Test Case ID: FR-4		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Md. Iftakhar Awal Chowdhury		
Module Name: AI-Powered Data Analysis / System Notifications		Test Execution date: 13-May-2025		
Test Title: Verify generation of early warning alerts for high-risk patients				
Description: Test that the system generates alerts when AI analysis identifies a high-risk patient based on predefined factors.				
Precondition (If any): AI analysis successfully executed. A patient's data matches high-risk criteria. Alerting mechanism is configured.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. A patient's data is updated to meet high-risk criteria. 2. Allow AI analysis to process the data. 3. Check the system's alert dashboard or designated notification channel for clinicians.	Patient ID: P_HighRisk_01 Risk Factors: (e.g., specific lab values, recent diagnosis combination)	An early warning alert is generated and made visible to appropriate personnel (e.g., on a dashboard, via secure message) for the identified high-risk patient.	As expected,	Pass
Post Condition: Alert is logged and visible.				

Test Case 19 :

Project Name: MediNexus	Test Designed by: MD. Mozahidul Islam
Text Case ID: FR- 5.1	Test Designed date: 13-May-2025

Text Priority (Low, Medium, High): High		Text Executed by: MD. Mozahidul Islam		
Module Name: Payment System		Test Execution date: 13-May-2025		
Test Title: Verify user can add a new payment				
Description: Description: Test adding a new payment entry for a patient.				
Precondition (If any): User is logged in and on the patient’s record page with a pending bill.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the payment page. 2. Enter payment details. 3. Click Submit.	Amount: 500 Payment Method: Credit Card	Payment is added successfully.	As expected.	Pass
Post Condition: Payment is added successfully.				

Test Case 20 :

Project Name: MediNexus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 5.2		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD. Mozahidul Islam		
Module Name: Payment System		Test Execution date: 13-May-2025		
Test Title: Verify payment status updates correctly.				
Description: Test the update of payment status after adding a new payment.				
Precondition (If any): User has just made a payment.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Check payment status. 2. Verify updated total.	Previous Total: 1000 Paid: 500	Updated balance should reflect the new payment..	As expected.	Pass
Post Condition: Updated balance should reflect the new payment.				

Test Case 21 :

Project Name: MediNexus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 5.3		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD. Mozahidul Islam		
Module Name: Payment System		Test Execution date: 13-May-2025		
Test Title: Verify user can add a new payment method				
Description: Test adding a new payment method (e.g., credit card) in account settings.				
Precondition (If any): User is logged in and on the Account Settings page.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Navigate to Account Settings. 2. Select "Payment Methods". 3. Click "Add New Payment Method". 4. Enter card information and confirm.	Card Number: 4111 1111 1111 1111 Expiry: 12/26 CVV: 123	New payment method is added to the user's account and is visible in the list.	As expected.	Pass
Post Condition: New payment method is saved and listed in Payment Methods.				

Test Case 22:

Project Name: MediNixus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 6.1		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD. Mozahidul Islam		
Module Name: Payment System		Test Execution date: 13-May-2025		
Test Title: Appointment Scheduling				
Description : Verify booking appointment with available doctor				
Precondition (If any): User is logged in and has selected a doctor with available slots.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Appointments page. 2. Select a doctor from the available list. 3. Choose a date and time slot. 4. Click “Book Schedule” or “Confirm Appointment”.	Doctor name: Dr. David H. Brown Date: 15-May-2025 Time: 5:00 PM	Appointment is booked successfully and a confirmation message is displayed.	As expected,	Pass
Post Condition: Appointment record is created and confirmed.				

Test Case 23 :

Project Name: MediNixus	Test Designed by: MD. Mozahidul Islam
Text Case ID: FR- 6.2	Test Designed date: 13-May-2025
Text Priority (Low, Medium, High): High	Text Executed by: MD. Mozahidul Islam
Module Name: Schedule Management	Test Execution date: 13-May-2025
Test Title: Verify rescheduling an existing appointment	
Description: Test changing the date/time of an already booked appointment.	
Precondition (If any): User is logged in and has at least one upcoming appointment scheduled.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Appointments list and find the existing appointment. 2. Click on “Reschedule” next to the appointment. 3. Select a new date and/or time. 4. Confirm the changes.	Original Date: 15-May-2025 at 5:00 PM New Date: 18-May-2025 at 6:00 PM	Appointment changes are saved and a reschedule confirmation is shown.	As expected.	Pass
Post Condition: Appointment is updated to the new date/time.				

Test Case 24 :

Project Name: MediNexus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 6.3		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): Medium		Text Executed by: MD. Mozahidul Islam		
Module Name: Schedule Management		Test Execution date: 13-May-2025		
Test Title: Verify cancelling an appointment				
Description: Test cancelling an existing upcoming appointment.				
Precondition (If any): User is logged in and has an upcoming appointment scheduled.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Appointments list. 2. Click on “Cancel” next to the relevant appointment. 3. Confirm the cancellation prompt.	Appointment to cancel: Dr. David H. Brown on 15-May-2025 at 5:00 PM	Appointment is cancelled and removed from the upcoming list (or moved to Cancelled).	As expected,	Pass
Post Condition: Appointment status is marked as Cancelled.				

Test Case 25 :

Project Name: MediNixus	Test Designed by: MD. Mozahidul Islam
Text Case ID: FR- 7.1	Test Designed date: 13-May-2025
Text Priority (Low, Medium, High): Medium	Text Executed by: MD. Mozahidul Islam
Module Name: Doctor Profile	Test Execution date: 13-May-2025
Test Title: Verify user can view doctor's profile details	

Description: Test that a user can view detailed information about a doctor.				
Precondition (If any): User is logged in and on the Doctors page or can access doctor profiles.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Doctors list or search for a doctor. 2. Select a doctor (e.g., Dr. David H. Brown) to view profile. 3. Verify the profile page shows name, specialization, experience, hours, and rates.	Doctor name: Dr. David H. Brown	Doctor's profile displays correct details such as name, specialty (Psychologist), experience, hours (e.g., 10:30am - 5:30pm), and hourly rate..	As expected,	Pass
Post Condition: N/A				

Test Case 26 :

Project Name: MediNexus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 7.2		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): Medium		Text Executed by: MD. Mozahidul Islam		
Module Name: Doctor Interaction		Test Execution date: 13-May-2025		
Test Title: Verify user can initiate chat with a doctor				
Description: Test sending a message to a doctor via the chat feature.				
Precondition (If any): User is logged in and has an active chat with a doctor or can start a new chat.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Chat section. 2. Select a doctor from the chat contacts (e.g., Dr. David H. Brown). 3. Type a message into the chat input field. 4. Send the message.	Message: Hello Doctor, I would like to ask about my treatment.	Message appears in the chat window and is delivered to the doctor.	As expected,	Pass
Post Condition: The message is sent and appears in the chat conversation.				

Test Case 27 :

Project Name: MediNixus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 7.3		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): Medium		Text Executed by: MD. Mozahidul Islam		
Module Name: Doctor Interaction		Test Execution date: 13-May-2025		
Test Title: Verify user can initiate an audio call with a doctor				
Description: Test starting an audio call with a doctor from the chat or call interface.				
Precondition (If any): User is logged in and the doctor’s chat profile is open.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. In the Chat or Doctor Profile page, click on the audio call icon for Dr. David H. Brown. 2. Wait for the call connection to establish.	Doctor: Dr. David H. Brown.	Audio call window opens showing call duration and doctor’s name (e.g., David H. Brown).	As expected,	Pass
Post Condition: The audio call is initiated and connected.				

Test Case 28 :

Project Name: MediNixus		Test Designed by: MD. Mozahidul Islam		
Text Case ID: FR- 7.2		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): Medium		Text Executed by: MD. Mozahidul Islam		
Module Name: Patient Record Management		Test Execution date: 13-May-2025		
Test Title: Verify user can search and view a patient's record				
Description: Test searching for a patient and viewing their details from the patient list.				
Precondition (If any): Test searching for a patient and viewing their details from the patient list.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Patient List or Dashboard. 2. Enter the patient name (e.g., Akash Mozahid) in the search box and search. 3. Click “View” or select the patient from the results to open the record.	Patient Name: Mozahid	Patient record page is displayed showing personal and medical details.	As expected,	Pass

Post Condition: N/A

Test Case 29 :

Project Name: MediNixus		Test Designed by: Md Raquibur Rahman		
Text Case ID: FR 5.1		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD Raquibur Rahman		
Module Name: Appointment Booking		Test Execution date: 13-May-2025		
Test Title: Verify appointment booking for available slot				
Description: Verify appointment booking for available slot				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Date, time, doctor	Appointment confirmed	pending	Not Executed
Post Condition: As per functionality -user logged in, record saved.				

Test Case 30 :

Project Name: MediNixus	Test Designed by: Md Raquibur Rahman
Text Case ID: FR 5.1	Test Designed date: 13-May-2025
Text Priority (Low, Medium, High): High	Text Executed by: MD Raquibur Rahman
Module Name: Appointment Booking	Test Execution date: 13-May-2025
Test Title: Verify appointment booking for unavailable slot	

Description: Verify appointment booking for unavailable slot				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Fully booked date/time	Slot unavailable message	Pending	Not Executed
Post Condition: As per functionality -user logged in, record saved.				

Test Case 31 :

Project Name: MediNixus		Test Designed by: Md Raquibur Rahman		
Text Case ID: FR 5.2		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD Raquibur Rahman		
Module Name: EMR Viewer		Test Execution date: 13-May-2025		
Test Title: Verify access denial for unauthorized user				
Description: Verify access denial for unauthorized user				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Doctor login and patient ID	Display patient EMR	Pending	Not Executed
Post Condition: As per functionality - user logged in, record saved				

Test Case 32:

Project Name: MediNixus		Test Designed by: Mirza Wahidur Rahman		
Text Case ID: FR-4.2		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Mirza Wahidur Rahman		
Module Name: Chat and Audio Call features		Test Execution date: 13-May-2025		
Test Title: Initiate an audio call				
Description: Verify that tapping the phone icon next to a contact initiates an audio call screen.				
Precondition (If any): Chat list is visible with contacts and call icons.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to doctor's info and select an existing doctor 2. Tap the phone (call) icon next to the doctor's name. 3. Observe that the screen transitions to the Audio Call.	Contact: David H. Brown (Psychologist). Contact: Robert Johnson (Neurologist).	The Audio Call screen appears showing the contact's name and call controls (mic, speaker, end call).	As expected,	Pass
Post Condition: Audio Call screen remains until call ends.				

Test Case 33:

Project Name: MediNixus		Test Designed by: MD Raquibur Rahman		
Text Case ID: FR-5.3		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD Raquibur Rahman		
Module Name: EMR Viewer		Test Execution date: 13-May-2025		
Test Title: Verify access denial for unauthorized user				
Description: Verify access denial for unauthorized user				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Receptionist login and patient ID	Access denied message	Pending	Not executed
Post Condition: As per functionality user logged in, record saved.				

Test Case 34:

Project Name: MediNexus		Test Designed by: MD Raquibur Rahman		
Text Case ID: FR-5.4		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Test Executed by: MD Raquibur Rahman		
Module Name: Chatbot		Test Execution date: 13-May-2025		
Test Title: Verify chatbot response for common query				
Description: Verify chatbot response for common query				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Question about clinic hours	Chatbot displays clinic hours	Pending	Not executed
Post Condition: As per functionality - user logged in, record saved.				

Test Case 35 :

Project Name: MediNixus		Test Designed by: Mirza Wahidur Rahman		
Text Case ID: FR-2.3		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): High		Test Executed by: Mirza Wahidur Rahman		
Module Name: OPD Module		Test Execution date: 13-May-2025		
Test Title: Add new OPD event with valid details				
Description: Ensure the users can add a new event in OPD by clicking 'Add Event' and filling in valid details.				
Precondition (If any): User is on OPD screen and clicks 'Add Event'.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Click the '+ Add Event' button on the OPD calendar screen. 2. Enter a valid event title, date, and description. 2. Save the event.	Event: 'Temperature Check', Date: '07-May-2025', Description: 'Patient fever monitoring.'	The new event appears on the calendar for the selected date.	As expected,	Pass
Post Condition: Event is added and visible on OPD calendar.				

Test Case 36:

Project Name: MediNixus		Test Designed by: MD Raquibur Rahman		
Text Case ID: FR-5.4		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD Raquibur Rahman		
Module Name: E-Prescription		Test Execution date: 13-May-2025		
Test Title: Verify generation of e-prescription by doctor				
Description: Verify generation of e-prescription by doctor				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Doctor selects medication and dosage	e-Prescription generated and stored	Pending	Not executed
Post Condition: As per functionality - user logged in, record saved.				

Test Case 37 :

Project Name: MediNixus		Test Designed by: MD Raquibur Rahman		
Text Case ID: FR-5.5		Test Designed date: 13-May-2025		
Text Priority (Low, Medium, High): High		Text Executed by: MD Raquibur Rahman		
Module Name: Notifications		Test Execution date: 13-May-2025		
Test Title: Verify SMS reminder for appointment				
Description: Verify SMS reminder for appointment				
Precondition (If any): User has valid access.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Login / Register 2. Enter test input 3. Trigger action	Scheduled appointment with patient phone number	SMS sent 24 hours before appointment	Pending	Not executed
Post Condition: As per functionality - user logged in, record saved.				

Test Case 38 :

Project Name: MediNixus		Test Designed by: Mirza Wahidur Rahman		
Text Case ID: FR-2.4		Test Designed date: 13-May-2025		
Test Priority (Low, Medium, High): Medium		Test Executed by: Mirza Wahidur Rahman		
Module Name: OPD Module		Test Execution date: 13-May-2025		
Test Title: Add OPD event with missing information (negative)				
Description: Verify that the system does not allow adding an OPD event if required fields are missing.				
Precondition (If any): User is on the OPD 'Add Event' form with incomplete data.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<div>1. Click '+ Add Event' on the OPD calendar.</div> <div>2. Leave the event title blank and enter date and description.</div> <div>3. Attempt to save the event.</div> <div>•</div>	Event: '', Date: '07-May-2025', Description: 'Follow-up visit.'	An error message appears indicating the title is required and the event is not saved.	As expected,	Pass
Post Condition: No incomplete event is added.				



American International University-Bangladesh

SOFTWARE ENGINEERING

Section: T

Project Title: MediNexis

Group No: 01

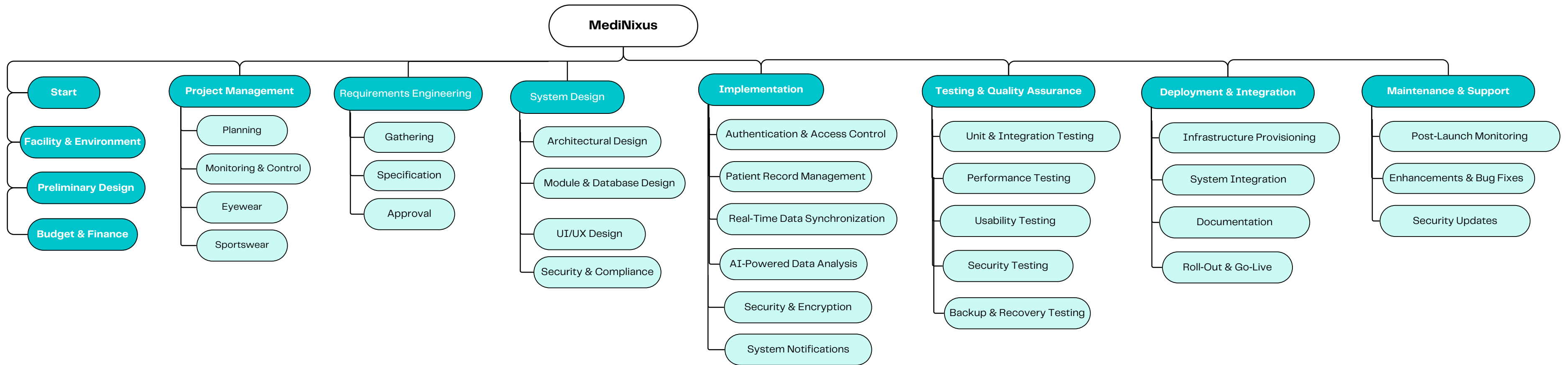
**Supervised By:
TONNY SHEKHA KAR**

Sl	Name	Id
1	Mozahidul Islam (Effort Estimation, Baseline & Scheduling, Revised)	23-51293-1
2	Mirza Wahidur Rahman (Effort Estimation, Baseline & Scheduling)	23-51440-1
3	Md. Iftakhar Awal Chowdhury (Effort Estimation)	23-51103-1
4	MD. Shamsul Alam (Scheduling)	23-50735-1
5	Md Raquibur Rahman	22-47220-1

Submitted by : Mozahidul Islam

Work Breakdown Structure (WBS)

In the *MediNexis* project, we used a Work Breakdown Structure (WBS) to divide the whole project into smaller parts. We split the work into steps like starting the project, collecting requirements, designing the system, coding, testing, and finally launching it. This helped us stay organized and made it easier to plan, assign work, and track progress. Here is this...



Effort Estimation and Scheduling Report

Project Name: Medinexius

Estimation Model: Basic COCOMO (Organic Type)

Project Type: Organic

Assumed Size: 14,000 Source Lines of Code (SLOC)

COCOMO Constants Used (Organic Model):

- **a** = 2.4
- **b** = 1.05
- **c** = 2.5
- **d** = 0.38

Step 1: Effort Estimation (Person-Months)

Formula:

$$PM = a \times (SLOC / 1000)^b$$

$$PM = 2.4 \times (14,000 / 1000)^{1.05}$$

$$PM = 2.4 \times 14^{1.05}$$

$$PM \approx 2.4 \times 15.08$$

$$PM \approx 36.19 \text{ person-months}$$

Step 2: Development Time (Months)

Formula:

$$TDEV = c \times (PM)^d$$

$$TDEV = 2.5 \times (36.19)^{0.38}$$

$$\text{TDEV} \approx 2.5 \times 3.71$$

$$\text{TDEV} \approx 9.28 \text{ months}$$

Step 3: Average Team Size

Formula:

$$\text{Team Size} = \text{PM} / \text{TDEV}$$

$$\text{Team Size} = 36.19 / 9.28$$

$$\text{Team Size} \approx 3.9 \approx 4 \text{ developers}$$

Final Summary Table

Metric	Value
Total Effort (PM)	36.19 person-months
Original Time (TDEV)	9.28 months
Original Team Size	4 developers

Work Breakdown Structure (WBS) and Timeline

Task	Task Name	Predecessors	Duration (weeks)
1	Project Management	-	6
2	Requirements Engineering	1	5
3	System Design	2	5
4	Implementation	3	14
5	Testing & Quality Assurance	4	6
6	Deployment & Integration	5	2
7	Maintenance & Support	6	2

Scheduling Overview

Based on the effort estimation, the estimated total project duration is approximately **9.28 months**. A detailed task schedule aligned with the WBS ensures smooth execution and timely delivery. With an average of 2–4 team members, the project can be efficiently managed over the proposed timeline.

