I. Introduction

1.1 Project Overview

The **NEU Clinic - School Health Management System (SHMS)** is a comprehensive digital healthcare platform designed exclusively for New Era University (NEU). This system revolutionizes student health services by transitioning from paper-based processes to a fully integrated digital solution that aligns with NEU's institutional workflows and commitment to student well-being.

The platform provides:

- **Centralized Digital Health Records**: Secure storage and management of medical histories for all NEU students.

- **Streamlined Appointment System**: Efficient scheduling for general check-ups and personal consultations with NEU medical staff.

- **QR-Based Technology**:

    - Virtual student IDs linked to NEU enrollment records for identity verification.
    - Unique QR codes for each appointment to automate attendance tracking.
- **Role-Specific Interfaces**: Customized portals for students, medical staff, and administrators to ensure smooth operation across all levels of NEU's healthcare services.

1.2 Project Goals and Objectives

The NEU Clinic SHMS is designed with three core objectives to enhance healthcare delivery at New Era University:

NEU-Centric Healthcare Transformation

- **Digital Transition**: Replace outdated paper-based health records and manual processes with a modern digital platform tailored to NEU's clinic workflows.

- **Policy Compliance**: Ensure strict adherence to NEU's data privacy and security policies, protecting sensitive student health information in compliance with institutional and national regulations.

- **Institutional Integration**: Seamlessly connect with existing NEU systems, including student enrollment databases and institutional email, to create a unified healthcare management ecosystem.

Operational Excellence and Efficiency

- **Automated Processes**: Implement smart automation for appointment scheduling, attendance tracking via QR codes, and health record management to reduce administrative workload.

- **Priority Case Management**: Develop specialized workflows to help NEU medical staff quickly identify and address urgent health cases (e.g., contagious illnesses, mental health emergencies).

- **Resource Optimization**: Use data-driven insights to improve clinic room usage, staff allocation, and medical supply management based on historical and predictive analytics.

Enhanced Student Healthcare Experience

- **24/7 Access**: Provide students with continuous access to their health profiles, appointment scheduling, and teleconsultation options through a secure, mobile-friendly portal.

- **Personalized Care**: Enable students to request one-on-one appointments, receive health reminders, and access tailored wellness recommendations.

- **Convenient Service**: Offer multiple access points, including web portals, mobile interfaces, and on-campus kiosks, to accommodate NEU's diverse student population.

1.3 Scope of the Project

The NEU Clinic SHMS will comprehensively cover the following aspects of the university's healthcare services:

User Coverage

- **Students**: All enrolled NEU students across undergraduate, graduate, and professional programs.

- **Medical Staff**: Physicians, nurses, dentists, mental health professionals, and other healthcare providers within NEU.

- **Administrators**: NEU health office personnel and authorized university staff responsible for overseeing healthcare operations.

- **Special Access**: Provisions for NEU athletic department staff to monitor student-athlete health and wellness.

Physical Infrastructure

- **Primary Location**: Full implementation at the NEU Main Health Clinic, covering all existing medical services.

- **Satellite Offices**: Extension to health offices in major NEU campus buildings for decentralized access.

- **Mobile Units**: Support for temporary health stations during NEU events (e.g., sports meets, university fairs).

- **Special Facilities**: Integration with NEU's quarantine and isolation rooms for emergency health situations.

Process Coverage

- **Health Records**: Management from student enrollment through graduation, including immunization tracking and disability accommodations.

- **Appointments**: End-to-end management, from booking to completion, for general and personal consultations.

- **Specialized Services**:

    - Health clearances for NEU athletes.
    - Compliance monitoring for international student health requirements.
    - Coordination with NEU's disability support services.
- **Seasonal Campaigns**: Synchronization with NEU's academic calendar for health initiatives like flu vaccination drives.

- **Emergency Protocols**: Integration with NEU's crisis management systems for rapid response to health emergencies.

II. System Requirements Analysis

2.1 Use Case Diagram

Detailed Use Case Descriptions

Actors

- **Students**: Primary users who access health services at NEU.
- **Doctor**: Medical professionals who provide healthcare services.
- **Nurse**: Healthcare staff who assist doctors and manage patient check-ins.
- **Admin**: System administrators who manage the application.

Student Use Cases

1. **Book Appointment**
    - **Description**: Students can schedule appointments with clinic staff.
    - **Precondition**: Student must be logged in.
    - **Main Flow**: Student selects available time slot and appointment type.
    - **Postcondition**: Appointment is booked and confirmation is provided.
2. **Check-in for Appointment**
    - **Description**: Students can digitally check in when arriving for appointments.
    - **Precondition**: Student must have a scheduled appointment.
    - **Main Flow**: Student indicates arrival and confirms appointment details.
    - **Postcondition**: System notifies relevant staff of student's arrival.
3. **View Health Records**
    - **Description**: Students can access their personal health records.
    - **Precondition**: Student must be authenticated.
    - **Main Flow**: Student navigates to health records section and views available data.
    - **Postcondition**: Student's health information is displayed securely.
4. **Submit Medical Documents**
    - **Description**: Students can upload required medical documentation.
    - **Precondition**: Student must be logged in.
    - **Main Flow**: Student uploads digital copies of medical documents.
    - **Postcondition**: Documents are stored in student's health profile.

Medical Staff Use Cases

1. **Send Health Reports**
    - **Description**: Doctors can create and send health reports.
    - **Precondition**: Doctor must be logged in and have examined the student.
    - **Main Flow**: Doctor creates report and sends it to the student.
    - **Postcondition**: Report is delivered to student and saved in records.

2. **View & Update Student Health Records**
    - **Description**: Medical staff can view and modify student health information.
    - **Precondition**: Staff must be authenticated with appropriate permissions.
    - **Main Flow**: Staff accesses student record and makes necessary updates.
    - **Postcondition**: Changes are saved to the database securely.
3. **Check-in Students (Scan QR Code & Verify Identity)**
    - **Description**: Staff can verify student identity using QR codes.
    - **Precondition**: Student must present appointment QR code.
    - **Main Flow**: Staff scans code and confirms student identity.
    - **Postcondition**: Appointment status is updated to "checked in."
4. **View Appointments**
    - **Description**: Medical staff can see scheduled appointments.
    - **Precondition**: Staff must be logged in.
    - **Main Flow**: Staff accesses appointments calendar.
    - **Postcondition**: Current and upcoming appointments are displayed.
5. **Schedule Appointments**
    - **Description**: Staff can create appointments for students.
    - **Precondition**: Staff must have scheduling permissions.
    - **Main Flow**: Staff selects available time slot and assigns to student.
    - **Postcondition**: Appointment is created and notification is sent to student.

Admin Use Cases

1. **Appointment Attendance Tracker**
    - **Description**: The system tracks and manages student attendance for scheduled clinic appointments.
    - **Precondition**: Appointments must exist in the system.
    - **Main Flow**:
        1. System automatically records attendance status when students check in.
        2. Medical staff can manually update attendance status (attended, no-show, cancelled, rescheduled).
        3. Admin can generate attendance reports by time period, provider, or appointment type.
        4. System flags repeated no-shows for follow-up.
    - **Postcondition**: Accurate attendance records are maintained and available for reporting.
    - **Related Use Cases**: This use case includes "Check-in Students" and is included by "Send Health Reports" as attendance information may be relevant to health reporting.
2. **Manage System Users**
    - **Description**: Admins can create, update, and deactivate user accounts for all user types.
    - **Precondition**: Admin must be logged in with appropriate permissions.
    - **Main Flow**: Admin accesses user management interface to add new users or modify existing accounts.

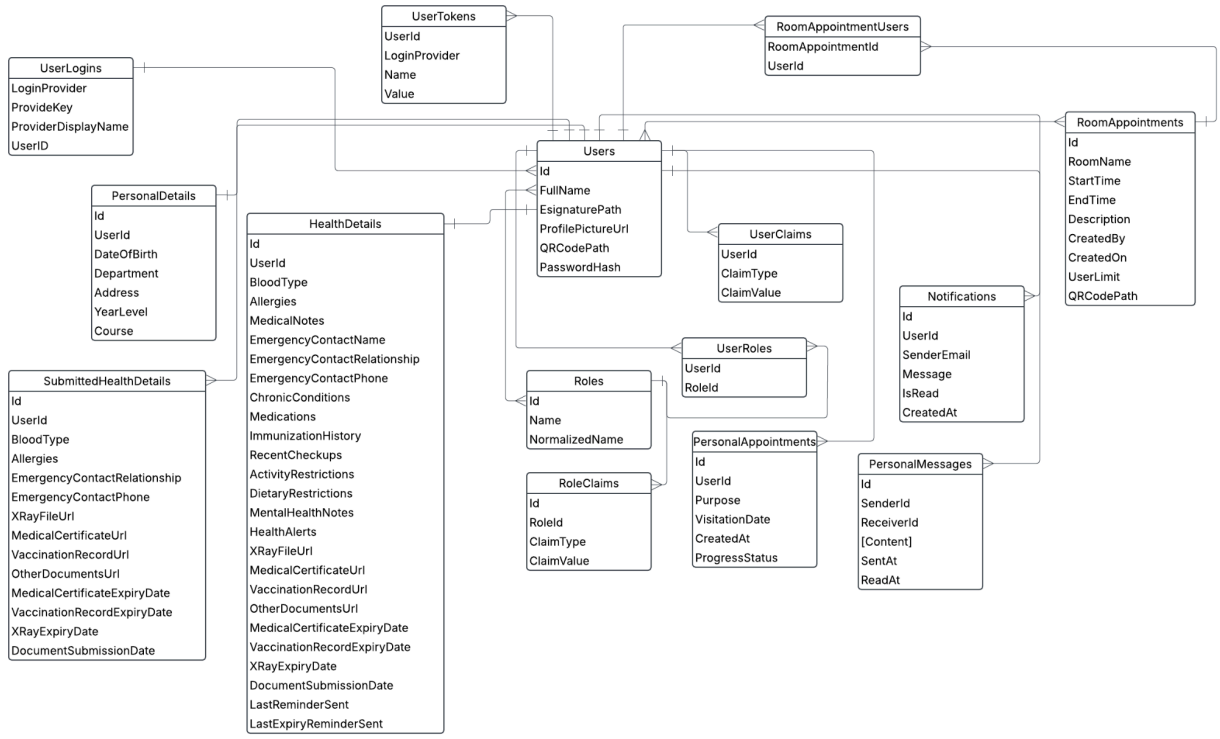
    - **Postcondition**: User accounts are updated in the system.
3. **Configure System Settings**
    - **Description**: Admins can adjust system-wide settings and parameters.
    - **Precondition**: Admin must be authenticated with system configuration privileges.
    - **Main Flow**: Admin navigates to settings panel and modifies configuration options.
    - **Postcondition**: System settings are updated and applied.
4. **Generate Reports**
    - **Description**: Admins can create various reports on clinic operations and usage.
    - **Precondition**: Admin must have reporting permissions.
    - **Main Flow**: Admin selects report type, parameters, and date range.
    - **Postcondition**: System generates and displays the requested report.

System-wide Use Cases

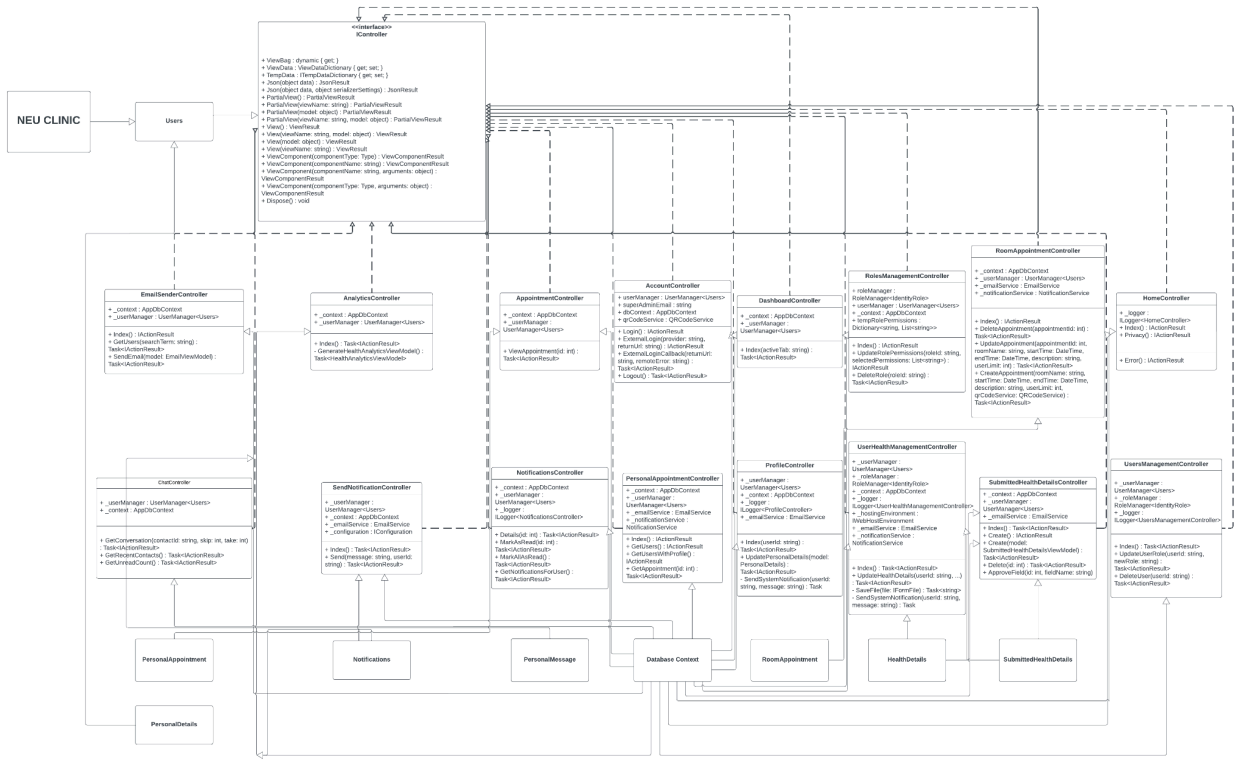
1. **Login**
    - **Description**: All users must authenticate to access the system.
    - **Precondition**: User must have valid credentials.
    - **Main Flow**: User enters username/password and submits.
    - **Postcondition**: User gains access to role-appropriate features.
2. **Logout**
    - **Description**: Users can securely end their session.
    - **Precondition**: User must be logged in.
    - **Main Flow**: User selects logout option.
    - **Postcondition**: Session is terminated securely.
3. **Generate QR Codes for Appointments**
    - **Description**: System creates unique QR codes for each appointment.
    - **Precondition**: Appointment must be confirmed in the system.
    - **Main Flow**: QR code is generated with encrypted appointment details.
    - **Postcondition**: QR code is available for student to use for check-in.

III. System Design

3.1 Entity-Relationship Diagram (ERD)



- 3.2 UML Class Diagram
    - Detailed Class, Attribute, and Method Descriptions
    - Explanation of Relationships



System Design Overview

The Health Management System is a comprehensive application designed to manage health-related information, appointments, and communications within an organization (likely an educational institution). The system facilitates health document submissions, appointment scheduling (both personal and group), notifications, and direct messaging between users.

Domain Models

This section describes the core data entities in the system and their relationships.

Users

Description: Represents an application user with authentication information.

Relationships:

- Has one-to-one relationship with PersonalDetails
- Has one-to-one relationship with HealthDetails
- Has one-to-many relationship with SubmittedHealthDetails
- Has one-to-many relationship with Notification

- Has many-to-many relationship with RoomAppointment (via RoomAppointmentUser join entity)
- Has one-to-many relationship with PersonalAppointment
- Has one-to-many relationship with PersonalMessage (as sender)
- Has one-to-many relationship with PersonalMessage (as receiver)

Key Attributes:

- User ID (primary key)
- Authentication credentials
- Role information

PersonalDetails

Description: Stores personal information for a user.

Relationships:

- Belongs to one User (via foreign key)

Key Attributes:

- Personal Details ID (primary key)
- User ID (foreign key)
- Name
- Contact information
- Address
- Other personal identifiers

HealthDetails

Description: Stores health-related information for a user.

Relationships:

- Belongs to one User (via foreign key)

Key Attributes:

- Health Details ID (primary key)
- User ID (foreign key)
- Medical history
- Current conditions
- Allergies
- Medications
- Approval status

SubmittedHealthDetails

Description: Represents a user's submission of health documents for approval by administrative or medical staff.

Relationships:

- Belongs to one User (via foreign key)

Key Attributes:

- Submitted Health Details ID (primary key)
- User ID (foreign key)
- Submission date
- Document references
- Approval status
- Notes

RoomAppointment

Description: Represents a group appointment (e.g., for a health seminar, group therapy, or other room-based event).

Relationships:

- Has many-to-many relationship with Users (via RoomAppointmentUser join entity)

Key Attributes:

- Room Appointment ID (primary key)
- Room information
- Date and time
- Duration
- Purpose/description
- Capacity
- Status

RoomAppointmentUser

Description: Join entity for the many-to-many relationship between RoomAppointment and Users.

Relationships:

- Belongs to one User (via foreign key)
- Belongs to one RoomAppointment (via foreign key)

Key Attributes:

- Room Appointment User ID (primary key)
- User ID (foreign key)
- Room Appointment ID (foreign key)
- Enrollment date
- Attendance status

PersonalAppointment

Description: Represents a personal appointment for a user (e.g., individual consultation).

Relationships:

- Belongs to one User (via foreign key)

Key Attributes:

- Personal Appointment ID (primary key)
- User ID (foreign key)
- Date and time
- Duration
- Purpose/description
- Status

Notification

Description: Represents a notification sent to a user.

Relationships:

- Belongs to one User (via foreign key)

Key Attributes:

- Notification ID (primary key)
- User ID (foreign key)
- Message content
- Date and time
- Read status
- Type/category

PersonalMessage

Description: Represents a direct message between two users.

Relationships:

- Belongs to one User as sender (via foreign key)
- Belongs to one User as receiver (via foreign key)

Key Attributes:

- Message ID (primary key)
- Sender User ID (foreign key)
- Receiver User ID (foreign key)
- Content
- Date and time
- Read status

ErrorViewModel

Description: Used for error display in the UI. Not a persisted entity but a view model.

Key Attributes:

- Error message
- Status code
- Additional details

Controllers

This section details the controllers responsible for handling specific functionality in the system.

AccountController

Responsibility: Handles user authentication and account creation.

Key Operations:

- User registration
- User login/logout
- Password management
- Ensures new users have associated PersonalDetails and HealthDetails

UsersManagementController

Responsibility: Allows SuperAdmins to manage users.

Key Operations:

- Create/read/update/delete users
- Assign and modify user roles
- View and manage user details

RolesManagementController

Responsibility: Allows SuperAdmins to manage roles.

Key Operations:

- Create/read/update/delete roles
- Assign permissions to roles (temporary functionality)
- Manage role hierarchies

UserHealthManagementController

Responsibility: Allows admins/medical staff to manage users' health information.

Key Operations:

- View users' health details
- Update health information
- Track health history

SubmittedHealthDetailsController

Responsibility: Handles the submission and review of health documents.

Key Operations:

- Submit health documents
- Review submitted documents
- Approve or reject submissions
- Update HealthDetails upon approval

PersonalAppointmentController

Responsibility: Manages personal appointments.

Key Operations:

- Create/read/update/delete personal appointments
- Send notifications about appointments
- Send email reminders
- Manage appointment status

RoomAppointmentController

Responsibility: Manages group appointments.

Key Operations:

- Create/read/update/delete room appointments
- Manage user enrollment in appointments
- Send notifications about room appointments
- Track attendance

NotificationsController

Responsibility: Handles user notifications.

Key Operations:

- Display notifications for a user
- Mark notifications as read
- Filter notifications by type

SendNotificationController

Responsibility: Sends notifications to users.

Key Operations:

- Create and send notifications
- Send emails with notifications
- Target notifications to specific user groups

EmailSenderController

Responsibility: Manages email communications.

Key Operations:

- Send emails to users
- Attach documents to emails
- Send reminder emails
- Schedule recurring emails

DashboardController

Responsibility: Displays dashboard information.

Key Operations:

- Show available appointments for students
- Display user-specific information
- Provide quick access to common functions

AnalyticsController

Responsibility: Provides analytical insights.

Key Operations:

- Aggregate health data
- Analyze document submission patterns
- Generate reports
- Visualize trends

ProfileController

Responsibility: Manages user profiles.

Key Operations:

- Display user profile
- Update personal details
- Update health details
- Manage user preferences

ChatController

Responsibility: Handles direct messaging between users.

Key Operations:

- Send messages
- Display conversation history
- Mark messages as read
- Manage user contacts

AppointmentController

Responsibility: Displays appointment details.

Key Operations:

- Show details of a specific room appointment

- Display participant list
- Provide appointment management functions

HomeController

Responsibility: Handles basic navigation.

Key Operations:

- Display home page
- Show error pages
- Provide system status information

System Interactions

Health Document Workflow

1. User submits health documents via SubmittedHealthDetailsController
2. Admin/medical staff reviews the submission
3. Upon approval, HealthDetails is updated accordingly
4. User receives notification about approval/rejection

Appointment Management Workflow

1. Admin creates appointments (personal or room) via respective controllers
2. Users view available appointments on their dashboard
3. Users enroll in room appointments or schedule personal appointments
4. Notifications and emails are sent as reminders
5. After the appointment, attendance/completion is recorded

User Communication Workflow

1. Users can send direct messages via ChatController
2. System notifications are sent via NotificationsController and SendNotificationController
3. Email communications are handled by EmailSenderController

Security and Access Control

The system implements role-based access control:

- SuperAdmins: Full system access, including user and role management
- Admins/Medical Staff: Access to health details, appointment management, and notifications

- Regular Users: Access to their own details, appointments, and messaging

Technical Considerations

- The system follows the MVC (Model-View-Controller) architectural pattern
- Error handling is standardized through the ErrorViewModel
- Authentication and authorization control access to sensitive health information
- Email and notification systems ensure timely communication

IV. User Manual (Jairus)

- 4.1 Introduction
    - Purpose and Scope
    - Target Audience
    - How to Use
- 4.2 Getting Started
    - System Requirements
    - Installation Instructions
    - Initial Setup
- 4.3 Core Functionalities
    - Step-by-step Instructions with Visual Aids
    - Examples of Use
- 4.4 Advanced Features
- 4.5 Frequently Asked Questions (FAQ)
- 4.6 Glossary
- 4.7 Index or Table of Contents

V. Implementation

- 5.1 Technology Stack (Kent)
- 5.2 System Architecture (Kent)
- 5.3 Key Implementation Details (Kent)