Software Requirements Specification

for

University Health Services System with Electronic Medical Records (EMR) Integration

Version 1.0

System Name: University Health Services System (UHSS)
Tutorial Section: TT2L
Date of Submission: 15th September

Name	ID
Bin Afeef, Abdullah Omar Hamad	**1211306604**
Youssef Fathy Fathy Mahrous Elsakkar	**1221302092**
KAWSAR	**1211310827**
AL REFAI, AL BARAA	**1221301987**

Table of Contents

1 Introduction

1.1 Overview

The Software Requirements Specification (SRS) document for the University Health Services System (UHSS) with Electronic Medical Records (EMR) Integration provides a comprehensive description of the software system's functional and non-functional requirements. The SRS aims to detail the system's purpose, scope, and overall product perspective, along with user and system interfaces, performance criteria, and design constraints. This document serves as a guide for developers, stakeholders, and testers, ensuring a shared understanding of the system's requirements and objectives.

1.2 Purpose

The purpose of the University Health Services System with EMR Integration is to create a centralized digital platform for managing student health records, scheduling appointments, and processing health services billing. The system aims to enhance healthcare delivery by integrating with existing Electronic Medical Records (EMR) systems, enabling seamless data sharing between the university's health services and local healthcare providers. This integration is intended to improve the continuity of care, provide comprehensive health records access, and streamline administrative processes within the university health services.

1.3 Scope

The **University Health Services System (UHSS)** is a software product designed to manage student health records, schedule healthcare appointments, and handle billing and insurance for health services. This system will integrate with external Electronic Medical Records (EMR) systems used by local healthcare providers, allowing for seamless sharing and updating of health data. The goal of developing this software is to enhance the efficiency and accuracy of university health services by digitalizing records and streamlining administrative processes. The software aims to provide improved access to comprehensive medical information, faster appointment scheduling, and efficient billing and insurance management. Additionally, it facilitates better communication and coordination between the university and local healthcare providers, ensuring students receive consistent and high-quality care.

1.4 Intended Audience

This document is intended for audiences of the following groups:

• **Project Stakeholders**: This group may include project managers, developers etc.
• **End Users:** Primarily university community which includes students, lecturers and staff at the universities.
• **System Administrators**: System administrators who are responsible for UHSS maintenance and management.
• **Administration Staff**: University administrators or other decision-makers required to approve and control the implementation of UHSS.

1.5 Product Overview

1.5.1 Product Perspective

The University Health Services System is designed as a comprehensive, integrated software solution tailored specifically for managing health services within a university setting. The system will provide a centralized platform for administration, administration staff, and the university community (students and university staff) to manage health-related activities efficiently. Additionally, the system will interface with external healthcare providers and a local Electronic Medical Records (EMR) system to ensure continuity of care and comprehensive health records management.

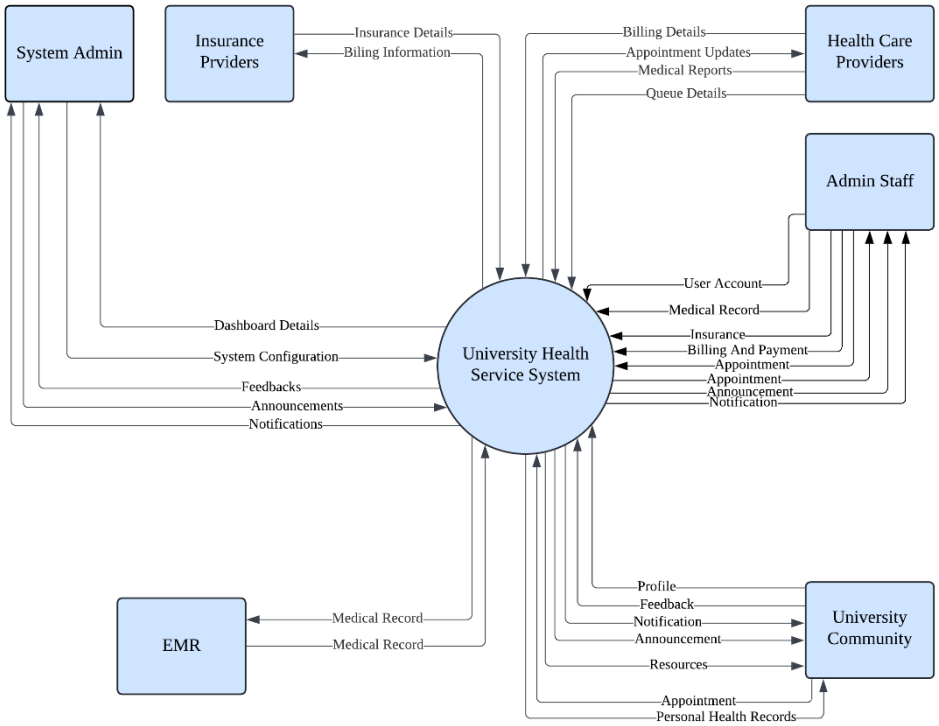


Figure 1: Context Diagram

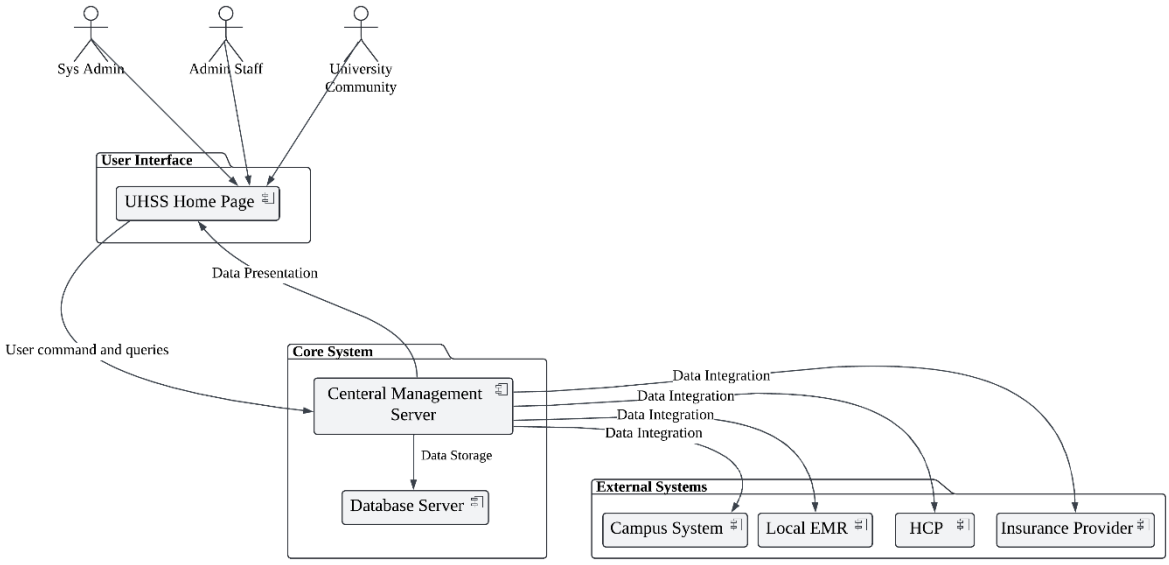


Figure 2: System Overview Diagram

1.5.1.1 System Interfaces

Requirement ID	Description	Priority	Author
REQ_SI001	The University Health Services System shall interface with the local Electronic Medical Records (EMR) system to retrieve and update comprehensive student health records. This interface will use secure APIs to ensure data privacy and integrity.	High	Abdulllah
REQ_SI002	The system shall interface with Health Care Providers' systems to facilitate the scheduling of appointments and sharing of relevant medical reports. This connection will be encrypted to maintain confidentiality and data security.	Medium	Abdulllah
REQ_SI003	The system shall integrate with university insurance provider systems to verify student and university staff insurance coverage and process claims efficiently. The interface must ensure secure data transfer and compliance with financial regulations.	Medium	Abdulllah

1.5.1.2 User Interfaces

Requirement ID	Main Function	Description	Style Elements	Priority	Author
REQ_UI001	Dashboard Interface	Provides an overview and access to frequently used features such as appointments, health records, billing, and notifications. Displays widgets and real-time notifications.	- Typography: Headings 24px bold, body text 16px regular. - Color Palette: Background (#f0f4f8), text (#333333), alert colors (#dc3545, #28a745). - Layout: Use a 12-column grid, 16px margins, and 12px padding.	High	Abdullah
REQ_UI002	Appointment Scheduling Interface	Allows service receivers to schedule, reschedule, or cancel health service appointments. Features a calendar view, search functionality, and confirmation dialogs.	- Typography: Headings 20px bold, body text 16px regular. - Color Palette: Calendar view in light blue (#f0f4f8), action buttons in blue (#007bff). - Controls: Confirm dialogs with hover effect, active states.	High	Abdullah
REQ_UI003	Health Records Interface	Enables authorised users to view and manage health records, including medical history and test results. Offers tabbed navigation, search options, and export functions.	- Typography: Headings 20px, body text 16px, labels 14px. - Color Palette: Background (#f8f9fa), text dark gray (#333333). - Icons: Use line-style icons next to text; export buttons in green (#28a745).	Medium	Abdullah
REQ_UI004	Billing and Insurance Interface	Manages billing information, insurance claims, and	- Typography: Bold for key figures (18px), regular for labels (14px).	Medium	Abdullah

		payments. Includes billing summaries, claim submission features, and a secure payment portal.	- Color Palette: Billing summary background light gray (#e9ecef), text dark gray (#333333), error text red (#dc3545). - Buttons: Rounded corners, hover state.		
REQ_UI005	User Profile Interface	Allows users to manage personal information, preferences, and account settings. Features editable fields, and notification settings.	- Typography: Standard for form fields (16px). - Color Palette: Editable fields background (#f0f4f8), save button blue (#007bff). - Form Fields: Light gray background with focus state border change to blue (#007bff).	Low	Abdullah

1.5.1.3 Hardware Interfaces

Requirement ID	Main Function	Description	Configuration Characteristics	Priority	Author
REQ_HI001	Server Hardware Interface	Connects the University Health Services System to the server hardware that hosts the software and stores the database.	- Configuration: Requires a server with at least 16 CPU cores, 64GB RAM, and 1TB SSD storage. - Ports: Supports Ethernet ports for network connectivity.	High	Abdullah
REQ_HI002	Client Workstation Hardware Interface	Provides support for a client device such as desktop, laptop, and tablet used by System administration	- Configuration: Compatible with Windows, MacOS, and Linux operating systems. - Devices Supported: Desktops, laptops,	Medium	Abdullah

		to change configuration in the server.	tablets with at least 8GB RAM and dual-core processors. - Protocols: Supports full-screen and touch input support.		
REQ_HI003	Printer and Scanner Hardware Interface	Allows integration with printers and scanners for printing medical records, billing documents, and scanning documents into the system.	- Configuration: Supports USB and network-connected printers and scanners. - Devices Supported: Laser printers, inkjet printers, and flatbed scanners. - Protocols: Supports standard print and scan drivers (e.g., TWAIN, WIA).	Medium	Abdullah
REQ_HI004	Backup Hardware Interface	Interfaces with external storage devices or network-attached storage (NAS) for data backup and recovery purposes.	- Configuration: Compatible with USB 3.0 or higher and network-attached storage. - Devices Supported: External hard drives, NAS devices with RAID configuration support. - Protocols: Supports standard backup protocols (e.g., NFS, SMB).	High	Abdullah

1.5.1.4 Software Interfaces

Category	Software Name	Version Number	Purpose	Reference
Database	MySQL	8.0.23	Used to create, store, modify, and retrieve data from user interactions.	MySQL Official Page
Operating System	macOS 11 and later		Support for UHSS applications on Apple hardware.	Apple Official Page
Operating System	Linux Ubuntu 20.04 LTS		Serve as a stable and secure platform for servers running the UHSS backend.	Ubuntu Official Page
Web Browser	Google Chrome	Latest	Access the UHSS web-based dashboard for real-time monitoring and management.	Chrome Browser Page
Web Browser	Mozilla Firefox	Latest	Ensure compatibility and optimal performance for users accessing the dashboard via Firefox.	Mozilla Official Page
Web Browser	Safari	Latest	Ensure compatibility and optimal performance for users accessing the dashboard via Safari.	Safari Official Page
Middleware	Django	5.1.1	Handle requests, execute backend logic, and manage communication between frontend, backend and external system interfaces.	Django Official Page
Security Software	OpenSSL	Latest	Encrypt data transmissions within the UHSS to secure user data and sensitive information.	OpenSSL Official Page
APIs	RESTful API Services	Latest	Facilitate communication between UHSS components and external integrations like EMR, HCP, Insurance Provider and campus systems.	-

1.5.1.5 Communications interfaces

External System Integration:

- ⇒ **Protocol**: HTTPS (for secure web services communication)
- ⇒ **Description**: Integration with external systems such as EMR systems, insurance providers, and third-party healthcare providers will use secure web service APIs with HTTPS to encrypt data in transit.
- ⇒ **Purpose**: Ensures secure communication with external healthcare providers and insurance systems, enabling seamless and secure sharing of medical records and billing information.

Local Area Network (LAN):

- ⇒ **Protocol**: Ethernet (IEEE 802.3)
- ⇒ **Description**: The system will use the university's internal wired LAN for communication between the server and client machines namely system admin and admin staff.
- ⇒ **Purpose**: Facilitates secure, high-speed data transfer between system components located within the university.

1.5.1.6 Memory constraints

1. **Primary Memory (RAM)**:

Server-Side:

- ⇒ Minimum: 16 GB
- ⇒ Recommended: 32 GB or higher
- ⇒ Purpose: To support multiple concurrent users, process real-time data, and manage integration with external systems such as EMRs. High performance is necessary during peak usage times like appointment bookings and accessing medical records.

Client-Side:

- ⇒ Minimum: 2 GB
- ⇒ Recommended: 4 GB or higher
- ⇒ Purpose: Ensures that the administrative and healthcare staff's workstations can run the system's user interface smoothly and handle patient data operations without delays.

2. **Secondary Memory (Storage)**:

Server-Side:

- ⇒ Minimum: 500 GB SSD
- ⇒ Recommended: 1 TB SSD or more
- ⇒ Purpose: To provide sufficient local storage for application-related data, temporary caching, and system logs. This storage will also be used for storing temporary files, backup copies of logs, and caching frequently accessed data to improve system performance.

Client-Side:

- ⇒ Minimum: 256 GB HDD/SSD
- ⇒ Recommended: 512 GB SSD
- ⇒ Purpose: Allows the local storage of essential application files and cache for administrative and healthcare staff. Ensures quick access to session data and smooth operation of the system.

3. **Database Storage Constraints**:

The system's database shall accommodate increasing patient data over time. The system shall handle archiving old records efficiently to maintain performance while retaining access to historical data when necessary.

4. **Cloud Storage**:

The system shall utilize cloud-based storage solutions for flexible, scalable data storage, particularly for handling large volumes of patient records or backups. This shall allow for system expansion without physical hardware limitations.

1.5.1.7 Operations

Requirement ID	Name	Description	Users Involved	Frequency	Priority
REQ_OP001	System Configuration	System Admin configures and manages system settings such announcements, and dashboard displays.	System Admin	As needed	High
REQ_OP002	Appointment Management	Users schedule, modify, or cancel appointments with health care providers. Admin staff ensure availability, and the system sends notifications about appointment updates.	University Community, Admin Staff, Health Care Providers	Daily	High
REQ_OP003	Billing and Payment Processing	Handles insurance claims and payments for services, with records automatically updated.	University Community, Admin Staff	As needed (during billing)	Medium
REQ_OP004	Medical Records Management	Health care providers and Admin Staff update medical	University Community, Health Care	During consultations and Registration	High

		records, and users access their personal health records securely through the system.	Providers, Admin Staff		
REQ_OP005	Queue and Workflow Management	Provides real-time updates on patient queues to streamline workflow for health care providers and university community.	University Community, Health Care Providers	Real-time	Medium
REQ_OP006	Feedback and Announcements	Users submit feedback and system sends announcements about health services to the University Community.	University Community, Admin Staff, System Admin	Weekly / As needed	Medium
REQ_OP007	Notifications	System sends notifications related to appointments, medical reports, billing, and updates automatically to the respective users.	University Community, Admin Staff, Health Care Providers	Real-time	High
REQ_OP008	User Profile Management	Users can manage their personal profile information, including contact details and health records. Admin staff verify and update user information as needed.	University Community, Admin Staff	As needed	Medium

1.5.1.8 Site adaptation requirements

Not relevant

1.5.1.9 Interfaces with services

1. **Electronic Medical Records (EMR) Integration**:
The system will interface with external EMR platforms for retrieving and updating. The communication will be facilitated through secure APIs using industry standards such as **FHIR** and **HL7**.
⇒ **Type of Service**: SaaS or cloud-based service

2. **Cloud Storage Services**:
The system may use cloud-based storage solutions for storing user records, appointment data, and backups. It will securely upload and retrieve data from the cloud to ensure redundancy and scalable storage.

⇒ **Type of Service**: Cloud storage AWS S3

3. **Insurance Verification System**:
The system will connect to third-party insurance providers' systems for real-time verification of patient insurance information. This integration ensures that billing and claims processing are handled efficiently.

⇒ **Type of Service**: SaaS

4. **Billing and Payment Services**:
The system will integrate with external payment gateway namely **Stripe** for handling patient billing and online payment processing.

⇒ **Type of Service**: SaaS (third-party payment gateway)

5. **HCP Services:**
The system will integrate with external healthcare providers interfaces such as local healthcare providers system to update appointments, retrieve billing information and retrieve real-time queue updates.

1.5.2 Product Functions

This section outlines the product functions of UHSS. These functions include functionalities such as user login, student or lecture registration through staff, booking, reporting, viewing announcements and more. Admins have unique capabilities for managing users and appointments, announcements, and responding to user reports. The aim of this system is to enhance campus health service operations and user experience.

Feature ID	Function	Description	Actor
REQ_F001	Login	Enables users to enter their credentials and safely log into the UHSS	University Community, Admin, Admin Staff
REQ_F002	Registration	Allows system registration of new students by admin staff	Admin Staff
REQ_F003	View list of Health care providers	Allow users to view detailed information about the health care service providers	University Community, Admin staff
REQ_F004	Book appointment	Enables users to book an appointment with a health care provider	University Community
REQ_F005	Cancel booking	Allows users to cancel their existing appointment bookings.	University Community
REQ_F006	View Announcements	Allows users to view announcements published by the health care providers and admin staff	University Community, System Admin, Admin Staff
REQ_F007	Manage Announcements	Allows the admin staff to add, delete, or update announcement details.	System Admin
REQ_F008	Manage Health Care providers	Allows the user to add, delete or update healthcare provider or their details and availability.	Admin Staff
REQ_F009	Symptom Checker maintenance	Allows the user to add, delete or update questions in the symptom checker.	Admin Staff

REQ_F010	Feedback	Allows user to provide feedback on their health service experience, including ratings and comments.	University Community
REQ_F011	View Queue	Provides real-time updates on the user's position in the appointment queue.	University Community
REQ_F012	Symptom Checker	A tool that allows users to input symptoms and receive recommendations on specialist to see based on the severity.	University Community
REQ_F013	View Medical Records	Allows user to view their medical records, test results.	University Community
REQ_F014	Billing and Payment	Allows the user to process billing for health services rendered and manage payments and claim insurance.	Admin Staff
REQ_F015	Update EMR	Facilitates secure sharing of medical records between the university system and external healthcare providers using local EMR integration.	System
REQ_F016	View Analytics	Provides an analytics dashboard to monitor appointment statistics and feedback for continuous improvement.	System Admin
REQ_F017	View Feedbacks	Allows system administrators to view the feedback submitted by users (students, lecturers, staff) to improve	System Admin

		services and user experience.	
REQ_F019	Manage Appointment	Allows both admin staff and users from the university community to manage appointments, including scheduling, modifying, rescheduling and viewing history. Admin staff can manage appointments on behalf of users when necessary.	University Community, Admin Staff
REQ_F020	View Appointment	Enables both users and admin staff to view detailed information about upcoming or past appointments, including appointment date, time, and healthcare provider.	University Community, Admin Staff
REQ_F021	User Profile	Allows users to view and update their personal information, such as contact details, preferences, and notification settings.	University Community

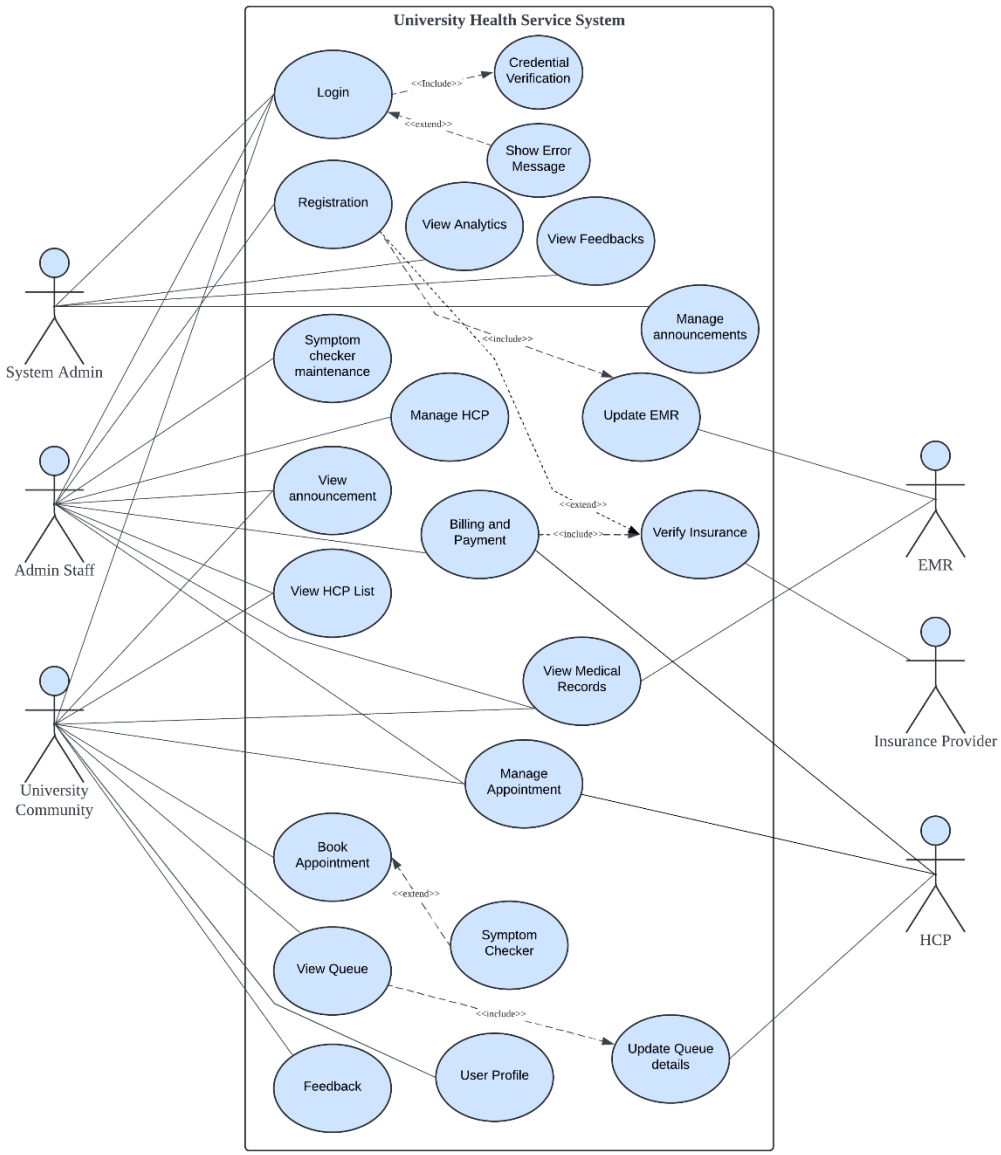


Figure 3: Use Case Diagram

1.5.3 User Characteristics

Role	Description	Expected Knowledge
University Community (Students, Lecturers, Staff)	These users interact with the system for health-related activities such as managing their health appointments, viewing announcements, providing feedback, accessing medical records, and using the symptom checker as part of the appointment-making process.	Basic understanding of the system's user-friendly interface for booking appointments, viewing records, submitting feedback, and using the symptom checker to determine appointment needs. Knowledge of notifications.
System Administrators	They maintain the system, including updating system configurations, overseeing data security, dealing with feedback from the University Community, and ensuring that the system integrates properly with the EMR and external healthcare providers. The system administrator also has access to analytics via the dashboard for monitoring system performance.	Advanced knowledge of system configuration, user management, and data security protocols. Troubleshooting, system updates, and backups. Familiar with dashboard analytics for monitoring performance and feedback management.
Admin Staff	Admin staff are responsible for patient registration, managing appointments, processing billing and insurance claims, updating healthcare provider details, and handling announcements and most importantly updating symptom checker contents in collaboration with healthcare providers.	Proficient in using the system for managing patient records, billing processes, insurance claims, appointments, and updating healthcare provider information. Competent in managing announcements.

1.5.4 Limitations

a) Regulatory Requirements and Policies:

⇒ The system handles, stores, and shares personal and sensitive data in compliance with **healthcare data protection regulations** like **HIPAA** and **GDPR**. These regulations require data encryption and anonymization, which can reduce performance during data processing.

⇒ The system retains patient data for long periods in accordance with **data retention policies** (e.g., 5-10 years), which increases storage demands and adds complexity to managing historical data.

b) Hardware Limitations:

⇒ The system relies on the **university's network infrastructure**, and during peak usage, limited network bandwidth may slow down data access and synchronization with external services such as EMR systems.

⇒ Users who access the system from **older computers or low-memory devices** may experience slower performance and degraded user experience.

⇒ The system may not function optimally on **older mobile devices or outdated browsers**, limiting accessibility for some users.

c) Higher-Order Language Requirements:

⇒ The development team uses programming languages such as **Java**, **Python**, and **JavaScript** to build the system. Integrating with older software versions or legacy systems may create limitations due to outdated technology.

⇒ The system follows industry standards like **FHIR** and **HL7** when interacting with external APIs, such as EMR systems and insurance verification services. This dependency limits flexibility when adding new features or integrating additional systems.

d) Safety and Security Considerations:

⇒ The system encrypts data both in transit and at rest to comply with security regulations. This encryption process impacts the system's performance, especially when processing large amounts of sensitive information.

⇒ The system requires **multi-factor authentication (MFA)** and **role-based access control (RBAC)** for user login and data access. These security measures introduce slight delays in the login and data access process.

⇒ The system depends on **AWS cloud services** to store data. Any performance issues or downtime in AWS may reduce the system's availability and limit access to data stored in the cloud.

Others:

⇒ If the user base or data volume grows beyond initial projections, the system may experience performance degradation unless the infrastructure scales accordingly.

⇒ The system must regularly update to maintain compatibility with **external APIs** (e.g., healthcare services, insurance providers, telehealth platforms). If these updates lag, the system may face compatibility issues.

1.5.5 Definition

User	A user is a student, lecturer, university staff, or staff member who can interact with the system independently.
HCP	Healthcare Providers
EMR	Electronic Medical Record
UHSS	University Health Service System

2 Requirements

2.1 Specified Requirements

2.1.1 **App**ointment Management System

Requirement: The system shall provide an online scheduling interface to book and cancel appointments with healthcare providers.

Details from Elicitation: Observations from existing systems (Harvard, FSU, Michigan) indicate the need for an intuitive booking interface with automated clear cancellation policies.

Kano Model Category: Satisfier.

Rationale: Efficient appointment management is expected in modern healthcare systems, directly enhancing user satisfaction.

Fulfilling Function: REQ_F019 - Manage Appointment

Author: ABDULLAH

2.1.2 Integration with Electronic Medical Records (EMR)

Requirement: The system shall securely integrate with the local EMR for seamless access and sharing of patient medical records.

Details from Elicitation: EMR integration is crucial for providing accurate healthcare services. Healthcare providers highlighted the need for easy access to patient records, indicating EMR as a fundamental requirement.

Kano Model Category: Dissatisfier.

Rationale: Failure to provide reliable EMR integration would lead to significant dissatisfaction.

Fulfilling Function: REQ_F015 - Update EMR

Author: ABDULLAH

2.1.3 Insurance Services Integration

Requirement: The system shall interface with insurance providers for direct insurance claims processing.

Details from Elicitation: Effective billing and insurance processes are vital to avoid user frustration, indicating the necessity of integrating with insurance services.

Kano Model Category: Dissatisfier.

Rationale: Seamless billing and insurance are fundamental needs, and their absence could result in dissatisfaction.

Fulfilling Function: REQ_F019 – Billing and Payment

Author: KAWSAR

2.1.4 Integrated Symptom Checker

Requirement: The system shall include a symptom checker that guides patients to the appropriate specialist based on their symptoms before booking an appointment.

Details from Elicitation: Interviews with healthcare providers highlighted the importance of guiding patients to the correct specialist. A decision-tree-style questionnaire was proposed to assist in symptom assessment.

Kano Model Category: Delighter.

Rationale: A symptom checker exceeds basic expectations by providing proactive health recommendations, improving the user experience.

Fulfilling Function: REQ_F012 - Symptom Checker

Author: KAWSAR

2.1.5 Real-Time Queue Monitoring System

Requirement: The system shall provide real-time updates on appointment wait times and queue status to patients.

Details from Elicitation: Brainstorming sessions suggested this feature to manage patient flow and enhance system efficiency.

Kano Model Category: Delighter.

Rationale: While not always expected, this feature significantly improves user satisfaction if implemented.

Fulfilling Function: REQ_F011 – View Queue

Author: YOUSSEF FATHY

2.1.6 Feedback Mechanism

Requirement: The system shall allow users to provide feedback on the healthcare services and their overall experience.

Details from Elicitation: Feedback is considered essential in modern systems to foster a sense of participation and improve services.

Kano Model Category: Satisfier.

Rationale: Allowing user feedback enhances satisfaction by showing that their input is valued.

Fulfilling Function: REQ_F010 – Feedback

Author: YOUSSEF FATHY

2.1.7 User-Friendly Patient Portal

Requirement: The system shall include a comprehensive patient portal that allows users to manage appointments, view medical records, and access health information.

Details from Elicitation: Observations showed the need for a centralized portal that facilitates healthcare management, including appointment scheduling and record access.

Kano Model Category: Satisfier.

Rationale: A multi-functional portal meets the expectations of modern users for centralized health management.

Fulfilling Function: REQ_F004 – Book Appointment, REQ_F005 – Cancel Booking, REQ_F010 – Feedback, REQ_F011 – View Queue

Author: AL BARAA

2.1.8 Notification System

Requirement: The system shall provide multi-channel notifications (e.g., email, SMS, WhatsApp) for appointment reminders and health services notifications.

Details from Elicitation: Students preferred varied notification methods like email and WhatsApp. Implementing flexible notification methods ensures timely and personalized notifications.

Kano Model Category: Satisfier.

Rationale: Timely notifications are expected for maintaining engagement and meeting user expectations for communication.

Fulfilling Function: REQ_F006 – View Announcements

Author: AL BARAA

2.2 Assumptions and dependencies

The development and operation of the **University Health Service System (UHSS)** are based on several key assumptions and dependencies, which must be considered to ensure the system functions as intended. These factors may impact the software requirements, design, and implementation.

Assumptions:

Stable Internet Connectivity:
We assume that all users (students, staff, healthcare providers) and the university's systems will have stable and high-speed internet access to ensure smooth real-time operations, such as appointment booking, medical record updates, and insurance verification.

User Access to Technology:
We assume that users will have access to personal devices such as smartphones, tablets, or computers to interact with the UHSS. We also assume users are familiar with basic technology, such as web browsers and mobile apps.

Integration with External Systems:
We assume that external systems, such as **Electronic Medical Records (EMR)** and **Insurance Provider Systems**, will have well-documented APIs that are secure, reliable, and available for integration with UHSS. These external systems are expected to remain compliant with international standards like **HL7** and **FHIR**.

Timely Data Updates:
It is assumed that healthcare providers and administrative staff will update medical records, billing information, and appointment details in a timely manner, ensuring that the system provides accurate and up-to-date information to users.

Regulatory Compliance:
We assume that the university and healthcare providers will comply with local and international regulations (e.g., HIPAA, GDPR) regarding data protection and privacy. This ensures that the system operates legally when handling sensitive health information.

System Scalability:
We assume that the UHSS will handle an increasing number of users over time, as the university grows. The system is designed to scale, but it depends on infrastructure that can expand accordingly, including cloud services and server resources.

Dependencies:

External EMR System:
The UHSS depends on the external EMR system for storing and retrieving medical records. If the EMR system is unavailable, certain features (e.g., viewing medical history) will not function as expected, although core features like appointment scheduling will remain operational.

Insurance Provider Systems:
The system depends on integration with **Insurance Providers** for verifying coverage and processing claims. Delays or downtime in these systems may affect insurance-related functionality, such as billing or payment verification.

User Authentication Systems:
The system depends on the university's **Identity Management System** for user authentication and role-based access control (RBAC). If the identity management system encounters issues, user login and access control may be compromised.

Regulatory Changes:
The system's design is dependent on current healthcare regulations (e.g., HIPAA, GDPR). Any significant changes in legal requirements would necessitate updates to the system's data handling, security, and privacy protocols to remain compliant.

2.3 Apportioning of requirements

The table below outlines the **apportioning of requirements** for the **University Health Service System**. Each module represents a distinct functional area within the system, with specific requirements associated with the respective users or actors. The key requirements are distributed across the system's modules based on user roles, such as the **University Community**, **Healthcare Providers**, and **System Admin**, ensuring that the system operates efficiently and meets all user needs.

Module	**Key Requirements**	**Author**
University Community	Profile management, appointment scheduling, symptom checker, medical record access, feedback collection	Kawsar
Healthcare Providers	EMR access, symptom reports, patient data updates	Abdullah
System Admin	System configuration, monitoring, permissions, security, announcements, feedback handling	Abdullah
Admin Staff	User account management, insurance verification, billing, appointment scheduling and management	Youssef
Symptom Checker	Input symptoms, process data, suggest healthcare providers, ensure protocol compliance	Kawsar
EMR	Secure storage of medical records, update records, healthcare provider access	Youssef
Billing and Insurance	Insurance claim verification, billing processing, payment tracking	Al Baraa
Feedback	Collect and process feedback, report to system admins for improvement	Al Baraa

2.4 External interfaces

The **University Health Service System (UHSS)** interacts with several external systems and entities to handle user data, medical records, appointments, insurance verification, and notifications. These interfaces facilitate the exchange of key data inputs and outputs with other systems to support the operational functionality of the UHSS.

Input/Output	Description	Priority	Author
Input: Health Records Data	The system receives comprehensive health records from the External EMR System whenever a user or administrator requests access to medical history.	High	Youssef
Input: Insurance Coverage Data	The system receives real-time Insurance Coverage Information from insurance providers, ensuring insurance validity before healthcare services.	Medium	Al Baraa
Input: Appointment Updates	Healthcare Providers send updated appointment details, cancellations, and rescheduling information to the system, which stores it for user access.	High	Youssef
Output: Appointment Confirmations	The system sends Appointment Confirmations and reminders to users via the external notification systems (e.g., SMS and email gateways).	Medium	Al Baraa
Output: Updated Health Records	The system sends updated medical information (such as diagnoses and treatments) to the External EMR System following healthcare appointments.	High	Abdullah
Output: Insurance Claims	The system submits insurance claims and relevant billing data to Insurance Providers for processing after healthcare services are provided.	Medium	Kawsar

Output: Billing Information	The system generates and sends Billing Information to users and Insurance Providers for payment processing.	Medium	Kawsar
Output: Notifications	The system generates notifications for appointments, reminders, and updates, sending them to users through external notification channels.	Medium	Abdullah

Description of External Inputs and Outputs:

1. **Inputs**:

⇒ **Health Records**: The system retrieves user health records from the **External EMR System** in response to user requests to view medical history or when medical updates are needed after appointments.
⇒ **Insurance Coverage Data**: The system pulls real-time **insurance coverage information** from insurance providers before healthcare services are provided to ensure valid coverage.
⇒ **Appointment Updates**: The system receives updates to scheduled appointments (e.g., cancellations or rescheduling) from **Healthcare Providers** and adjusts the system's calendar accordingly.

2. **Outputs**:

⇒ **Updated Medical Records**: The system sends newly updated medical information (e.g., diagnosis, treatment plans) to the **External EMR System** after healthcare appointments are completed.
⇒ **Insurance Claims and Billing Information**: The system sends **insurance claims** and related **billing information** to insurance providers and users after medical services are rendered.
⇒ **Appointment Confirmations** and **Notifications**: The system generates and sends appointment confirmations, reminders, and other notifications to users through external notification systems (such as SMS and email).

Conclusion

The **UHSS** handles several key inputs and outputs as part of its interaction with external systems. By retrieving, processing, and transmitting data such as medical records, insurance claims, billing, and appointment information, the system ensures a smooth and secure exchange of information between users, healthcare providers, and external systems.

2.5 Functions: Activity and Sequence Diagrams

2.5.1 Login

Table: REQ_F001

Requirement ID	REQ_F001	**Version**	1.0
Description	Enables users to enter their credentials and safely log into the UHSS.		
Actor	University Community, Admin Staff		
Author	Youssef		

Table: Login Use Case

Feature	Login	**Version**	1.0
Purpose	To allow users to securely log in to the system.		
Actor	University Community, Admin Staff		
Precondition	User is on the login page.		
Postcondition	User gains access to the system or receives an error message for invalid credentials.		
Main flow	1. The user navigates to the login page. 2. The user enters credentials. 3. The system checks the entered credentials. 4. If valid, the system grants access. 5. If invalid, the system displays an error message.		
Alternate Scenario	3.a. If credentials are invalid, show an "Error Message" and prompt the user to re-enter credentials.		
Author	Youssef		

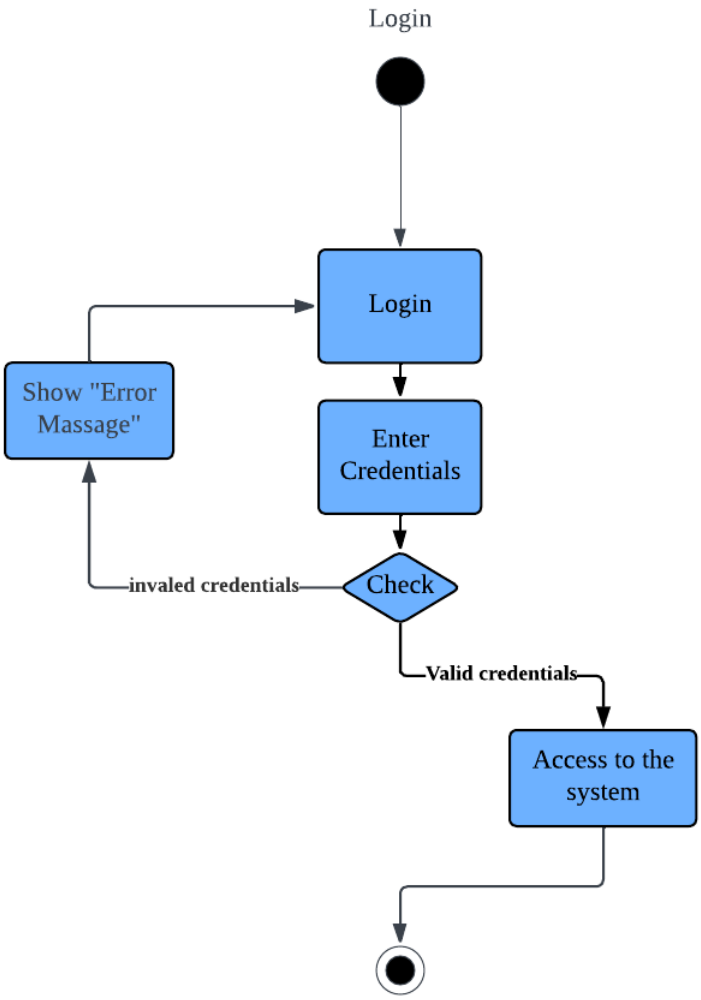


Figure 4: Login activity Diagram

2.5.2 View announcement

Table: REQ_F006

Requirement ID	REQ_F006	**Version**	1.0
Description	Allows users to view announcements published by the health care providers and admin staff.		
Actor	University Community, Admin Staff		
Author	Abdullah		

Table: View Announcement Use Case

Feature	View announcement	**Version**	1.0
Purpose	To allow users to view announcements published by healthcare providers and admin staff.		
Actor	University Community, Admin Staff		
Precondition	User is logged in and navigates to the announcements section.		
Postcondition	User views the list of announcements successfully.		
Main flow	1. The user clicks "View Announcement". 2. The user interface sends a request to the system to retrieve the announcement list. 3. The system requests the announcement data from the database. 4. The database retrieves the announcement data and sends it to the system. 5. The system displays the announcement list to the user.		
Alternate Scenario	None specified.		
Author	Abdullah		

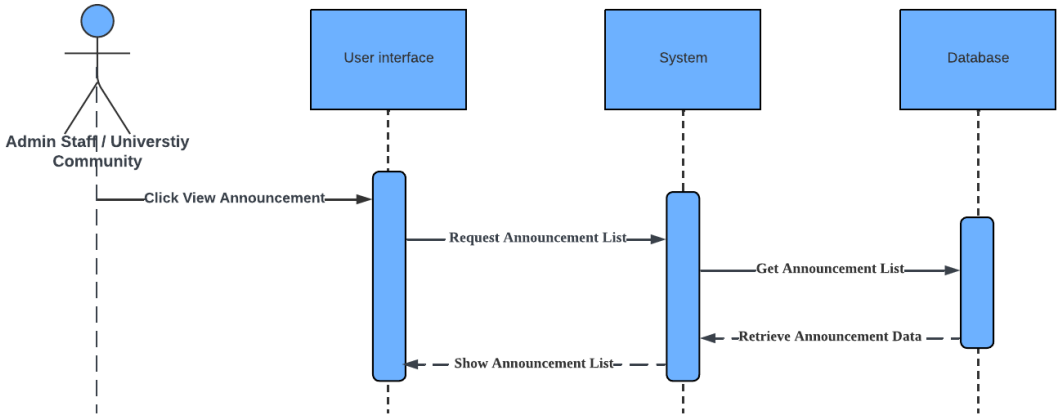


Figure 5: View Announcement Sequence Diagram

2.5.3 View HCP list

Table: REQ_F003

Requirement ID	REQ_F003	**Version**	1.0
Description	Allows users to view detailed information about the health care service providers.		
Actor	University Community, Admin Staff		
Author	Kawsar		

Table: View HCP list Use Case

Feature	View HCP list	**Version**	1.0
Purpose	To enable users to view the list of health care providers with detailed information.		
Actor	University Community, Admin Staff		
Precondition	User is logged in and navigates to the health care providers section.		
Postcondition	User views the list of health care providers successfully.		
Main flow	1. The user clicks "View HCP List". 2. The user interface sends a request to the system to retrieve the list of health care providers. 3. The system queries the database for the HCP list. 4. The database returns the HCP list to the system. 5. The system displays the HCP list to the user.		
Alternate Scenario	None specified.		
Author	Kawsar		

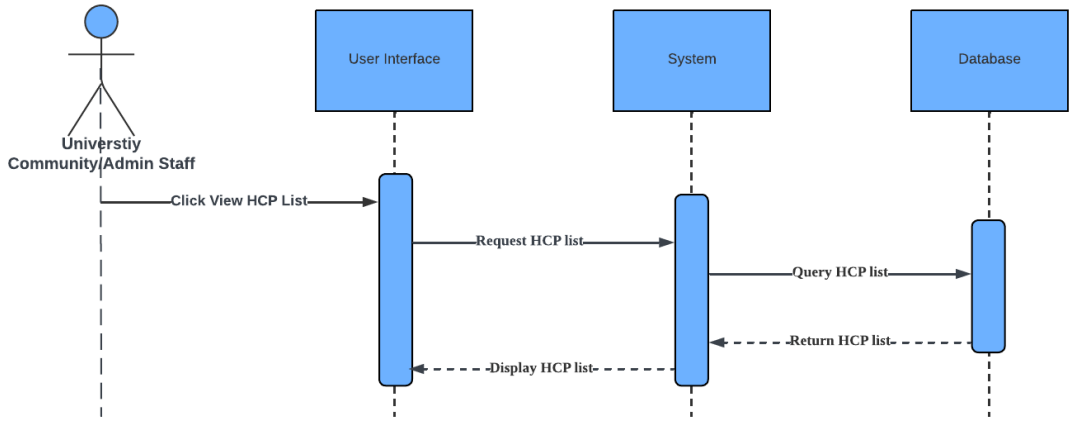


Figure 6: View HCP List Sequence Diagram

2.5.4 Book appointment

Table: REQ_F004

Requirement ID	REQ_F004	**Version**	1.0
Description	Enables users to book an appointment with a health care provider.		
Actor	University Community		
Author	AL BARAA		

Table: Book Appointment Use Case

Feature	Book Appointment	**Version**	1.0
Purpose	To allow users to book an appointment with a health care provider based on their symptoms and availability.		
Actor	University Community		
Precondition	User is logged in and has navigated to the booking section.		
Postcondition	User successfully books an appointment with the chosen health care provider.		
Main flow	1. The user clicks "Book Appointment". 2. The user interface requests the symptom checklist. 3. The system queries the database for the symptom checklist. 4. The database retrieves and sends the symptom checklist to the system. 5. The system displays the symptom checklist to the user. 6. The user submits their symptoms. 7. The system queries the specialist list based on the symptoms. 8. The database retrieves and sends the specialist list. 9. The system displays the specialist list. 10. The user selects a specialist. 11. The system acknowledges the selection and shows the list of health care providers (HCPs). 12. The user selects an HCP. 13. The system acknowledges the selection and shows the booking calendar. 14. The user selects a date. 15. The system requests available slots from the HCP. 16. The HCP returns the available slots. 17. The user selects an available slot. 18. The system posts the booking form. 19. The system saves the booking details and confirms the save. 20. The system displays the confirmed booking details to the user.		
Alternate Scenario	None specified.		
Author	AL BARAA		

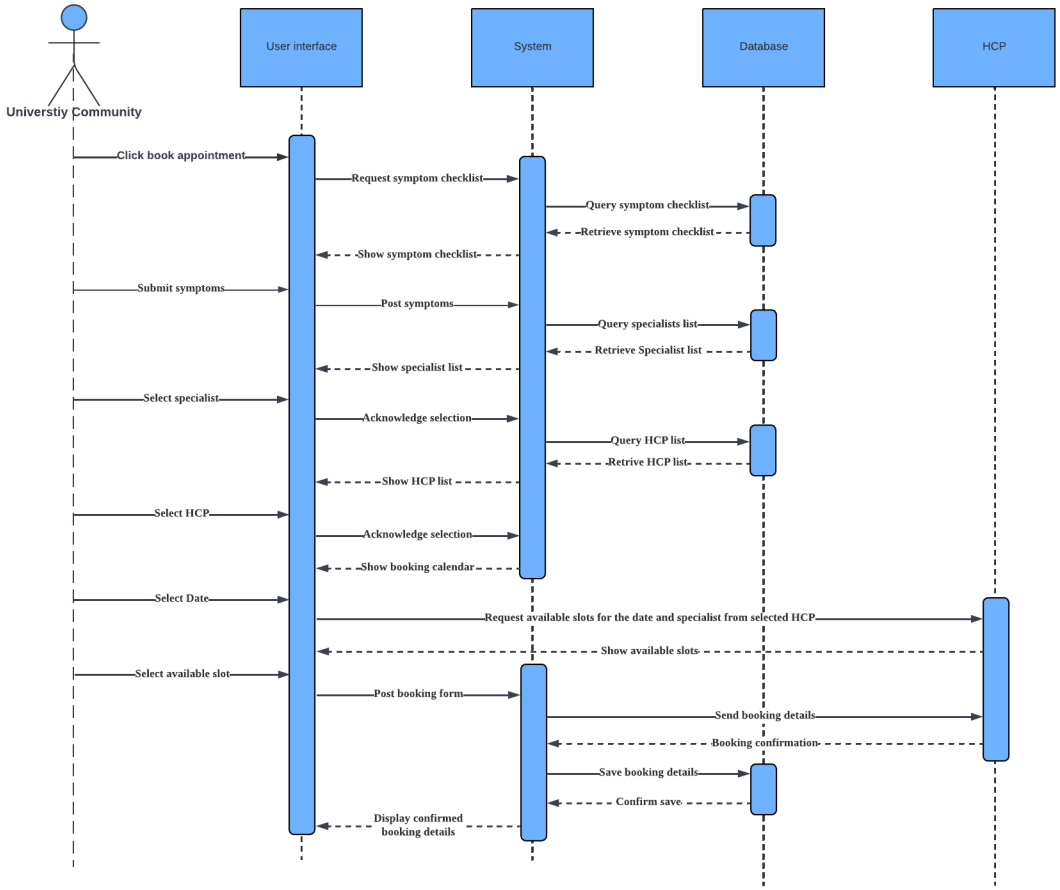


Figure 7: Book Appointment Sequence Diagram

2.5.5 View Queue

Table: REQ_F0011

Requirement ID	REQ_F011	**Version**	1.0
Description	Provides real-time updates on the user's position in the appointment queue.		
Actor	University Community		
Author	Kawsar		

Table: View Queue Use Case

Feature	View Queue	**Version**	1.0
Purpose	To allow users to view the current queue status for their appointments.		
Actor	University Community		
Precondition	User is logged in and navigates to the queue section.		
Postcondition	User views the current status of the appointment queue successfully.		
Main flow	1. The user clicks "View Queue". 2. The user interface sends a request to the system to retrieve the current queue stat 3. The system requests queue details from the health care provider (HCP). 4. The HCP returns the queue list to the system. 5. The system displays the current queue to the user.		
Alternate Scenario	None specified.		
Author	Kawsar		

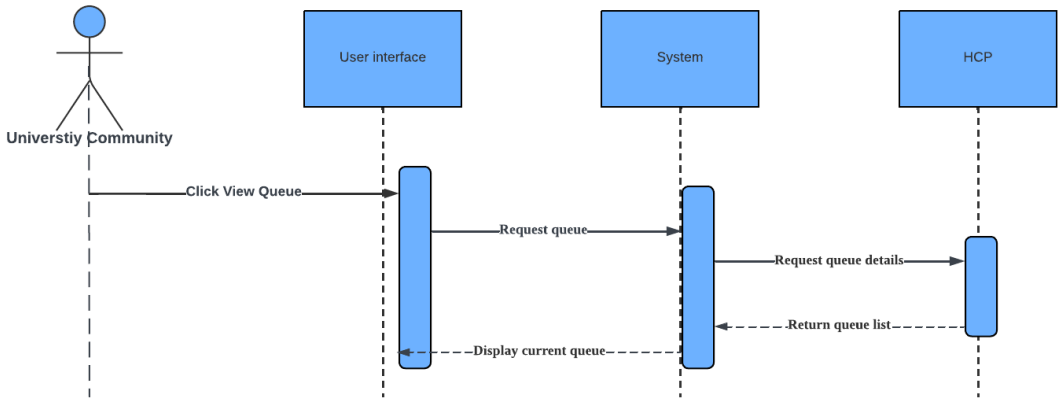


Figure 8: View Queue Sequence Diagram

2.5.6 Feedback

Table: REQ_F0010

Requirement ID	REQ_F010	**Version**	1.0
Description	Allows users to provide feedback on their health service experience, including rat comments.		
Actor	University Community		
Author	Kawsar		

Table: Feedback Use Case

Feature	Feedback	**Version**	1.0
Purpose	To enable users to provide feedback on their health service experience.		
Actor	University Community		
Precondition	User is logged in and navigates to the feedback section.		
Postcondition	User successfully submits feedback, which is stored in the database.		
Main flow	1. The user clicks "Provide feedback". 2. The user interface requests the feedback form. 3. The system displays the feedback form to the user. 4. The user submits the feedback form. 5. The system posts the feedback data to the database. 6. The database stores the feedback and confirms the save. 7. The system shows a success message to the user.		
Alternate Scenario	None specified.		
Author	Kawsar		

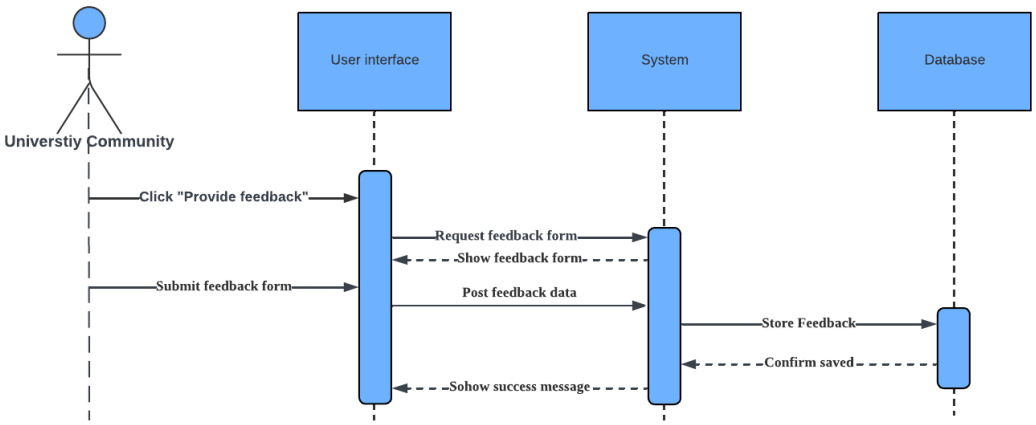


Figure 9: Feedback Sequence Diagram

2.5.7 View medical records

Table: REQ_F0013

Requirement ID	REQ_F013	**Version**	1.0
Description	Allows users to view their medical records and test results.		
Actor	University Community		
Author	AL BARAA		

Table: View Medical Records Use Case

Feature	View Medical Records	**Version**	1.0
Purpose	To enable users to view their personal medical records and test results.		
Actor	University Community		
Precondition	User is logged in and navigates to the medical records section.		
Postcondition	User views their medical records successfully.		
Main flow	1. The user clicks "View Medical Records". 2. The user interface sends a request to the system to retrieve the user's medical rec 3. The system queries the Electronic Medical Record (EMR) system for the user's medical records. 4. The EMR returns the user's medical records to the system. 5. The system displays the medical records to the user.		
Alternate Scenario	None specified.		
Author	AL BARAA		

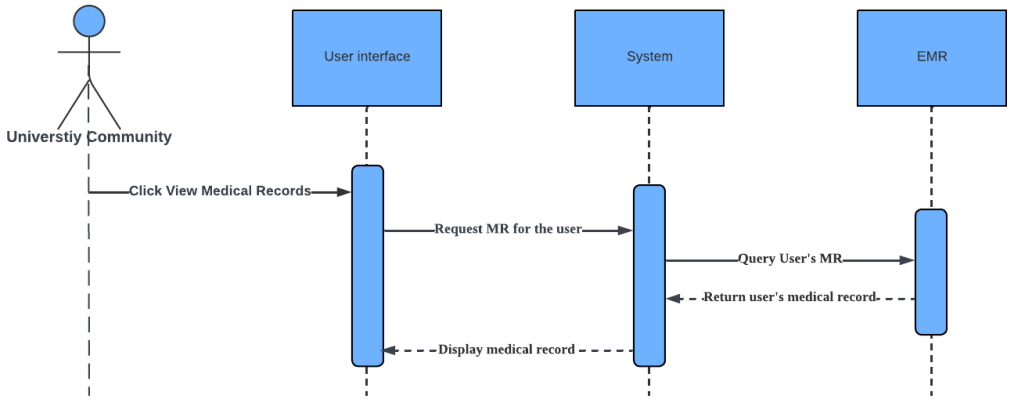


Figure 10: View Medical Records Sequence Diagram

2.5.8 Manage appointments

Table: REQ_F0019

Requirement ID	REQ_F019	**Version**	1.0
Description	Allows users from the university community to manage appointments, including sc modifying, rescheduling, and viewing history.		
Actor	University Community		
Author	AL BARAA		

Table: Manage Appointment Use Case

Feature	Manage Appointment	**Version**	1.0
Purpose	To allow users to manage their appointments, including canceling existing appointments.		
Actor	University Community		
Precondition	User is logged in and navigates to the manage appointments section.		
Postcondition	Appointment is successfully managed (e.g., canceled) and reflected in the system.		
Main flow	1. The user clicks "Manage Appointment". 2. The user interface sends a request to the system to retrieve the appointment list.		

	3. The system queries the database for the user's appointment list. 4. The database retrieves and sends the appointment list to the system. 5. The system displays the appointment list to the user. 6. The user selects an appointment to cancel. 7. The system acknowledges the selection and prompts for confirmation. 8. The user confirms the cancellation. 9. The system requests the cancellation from the HCP. 10. The HCP confirms the cancellation. 11. The system updates the appointment status to canceled in the database. 12. The system displays a cancellation success message to the user.
Alternate Scenario	None specified.
Author	AL BARAA

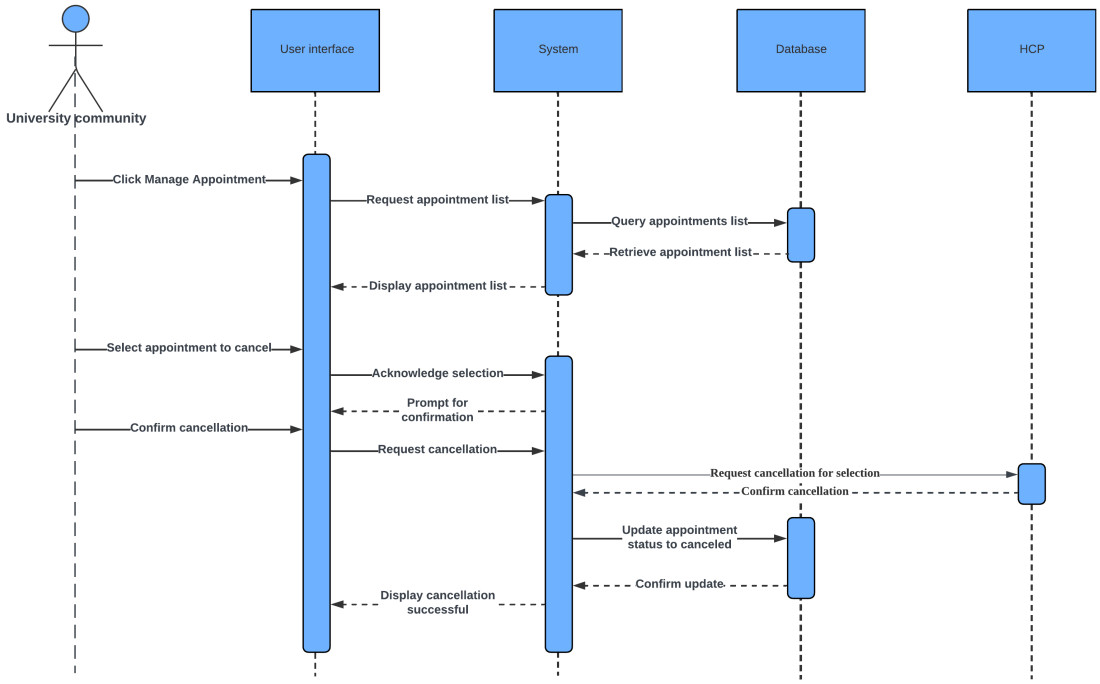


Figure 11: Manage Appointment Sequence Diagram

2.5.9 User profile

Table: REQ_F0021

Requirement ID	REQ_F021	**Version**	1.0
Description	Allows users to view and update their personal information, such as contact details and notification settings.		
Actor	University Community		
Author	Youssef		

Table: User Profile Use Case

Feature	User Profile	**Version**	1.0
Purpose	To enable users to view and update their personal profile information.		
Actor	University Community		
Precondition	User is logged in and navigates to the user profile section.		
Postcondition	User profile is successfully updated in the system.		
Main flow	1. The user clicks "User Profile". 2. The user interface retrieves profile data from the system. 3. The system queries the database for the user's profile data. 4. The database returns the profile data to the system. 5. The system displays the profile data to the user. 6. The user edits their profile information. 7. The user interface sends the updated profile data to the system. 8. The system updates the user profile in the database. 9. The database confirms the update. 10. The system displays a success message to the user.		
Alternate Scenario	None specified.		
Author	Youssef		

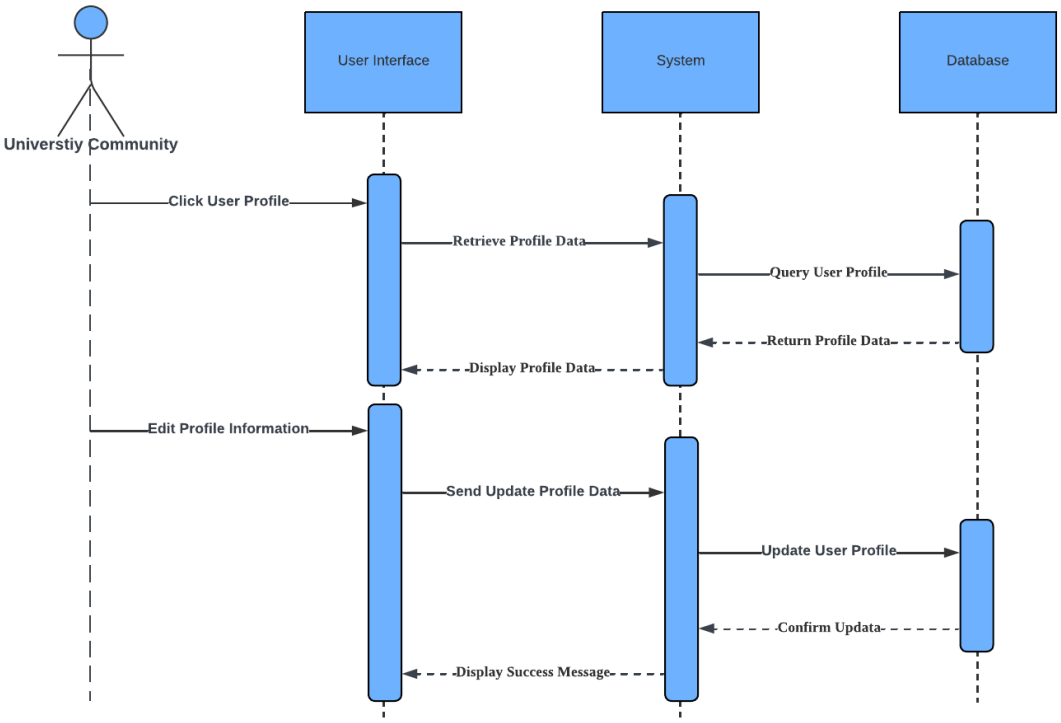


Figure 12: User Profile Sequence Diagram

2.5.10 Registration

Table: REQ_F0021

Requirement ID	REQ_F002	**Version**	1.0
Description	Allows system registration of new students by admin staff.		
Actor	Admin Staff		
Author	Youssef		

Table: Registration Use Case

Feature	Registration	**Version**	1.0
Purpose	To enable the admin staff to register new students into the system.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the registration section.		

Postcondition	The student is successfully registered in the system and their medical records are updated.
Main flow	1. Admin staff clicks "Registration". 2. The user interface requests the registration form from the system. 3. The system displays the registration form to the admin staff. 4. The admin staff inserts the registration details. 5. The system checks for user existence in the database. 6. If the user already exists: a. The system shows an error message. b. The registration process terminates. 7. If the user does not exist: a. The system sends medical records to the EMR. b. The EMR updates successfully. c. The system saves the user in the database. d. The system displays a successful registration message.
Alternate Scenario	If the user already exists, show an error message and terminate the registration process.
Author	Youssef

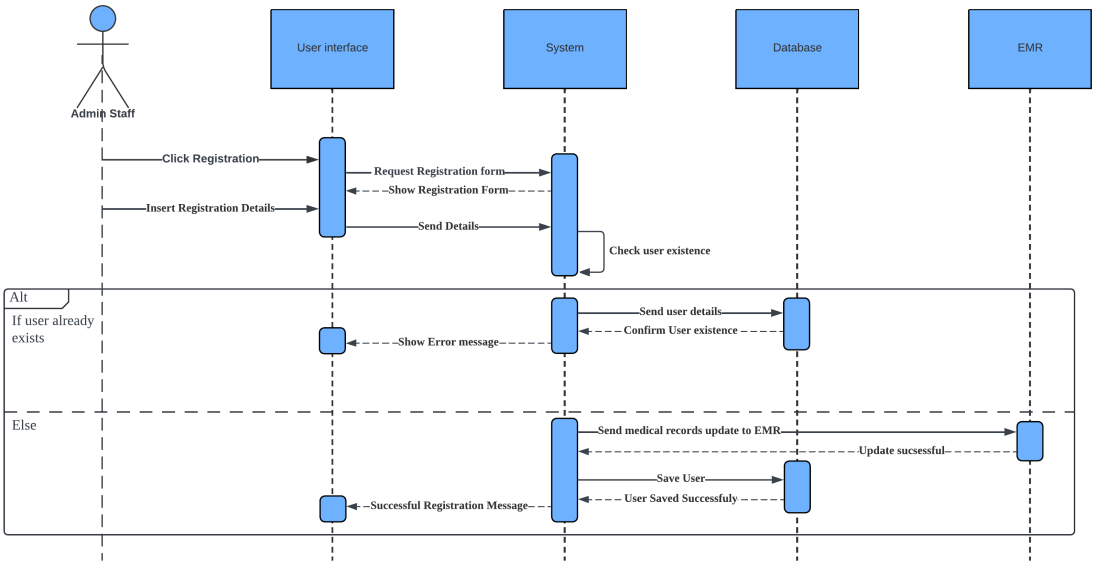


Figure 13: Registration Sequence Diagram

2.5.11 Manage symptom checker

Table: REQ_F009

Requirement ID	REQ_F009	**Version**	1.0
Description	Allows the admin staff to add, delete, or update questions in the symptom checker.		
Actor	Admin Staff		
Author	Abdullah		

Table: Manage Symptom Checker Use Case

Feature	Manage Symptom Checker	**Version**	1.0
Purpose	To enable admin staff to manage the symptom checker, including adding, deleting, and updating symptoms.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the symptom checker management section.		
Postcondition	Symptoms in the checker are successfully managed (added, updated, or deleted) and reflected in the system.		
Main flow	1. Admin staff clicks "Manage Symptom Checker". 2. The user interface requests the symptoms list from the system. 3. The system queries the database for the symptom list. 4. The database returns the symptom list to the system. 5. The system displays the symptom list to the admin staff. 6. Admin staff selects a symptom to edit. 7. The system requests the symptom edit form. 8. The admin staff submits updated details. 9. The system updates the symptom in the database. 10. The system shows a success message. **Adding a Symptom:** 1. Admin staff clicks "Add Symptom". 2. The system shows the add symptom form. 3. Admin staff submits the form. 4. The system validates input. 5. If the symptom exists, show error message; else, save symptom and show succes **Deleting a Symptom:** 1. Admin staff selects a symptom. 2. The system shows options for selection. 3. Admin staff clicks "Delete". 4. The system prompts for confirmation.		

	5. Admin staff confirms deletion. 6. The system deletes the symptom entry and shows a delete success message.
Alternate Scenario	If the symptom already exists during addition, show an error message and halt the process.
Author	Abdullah

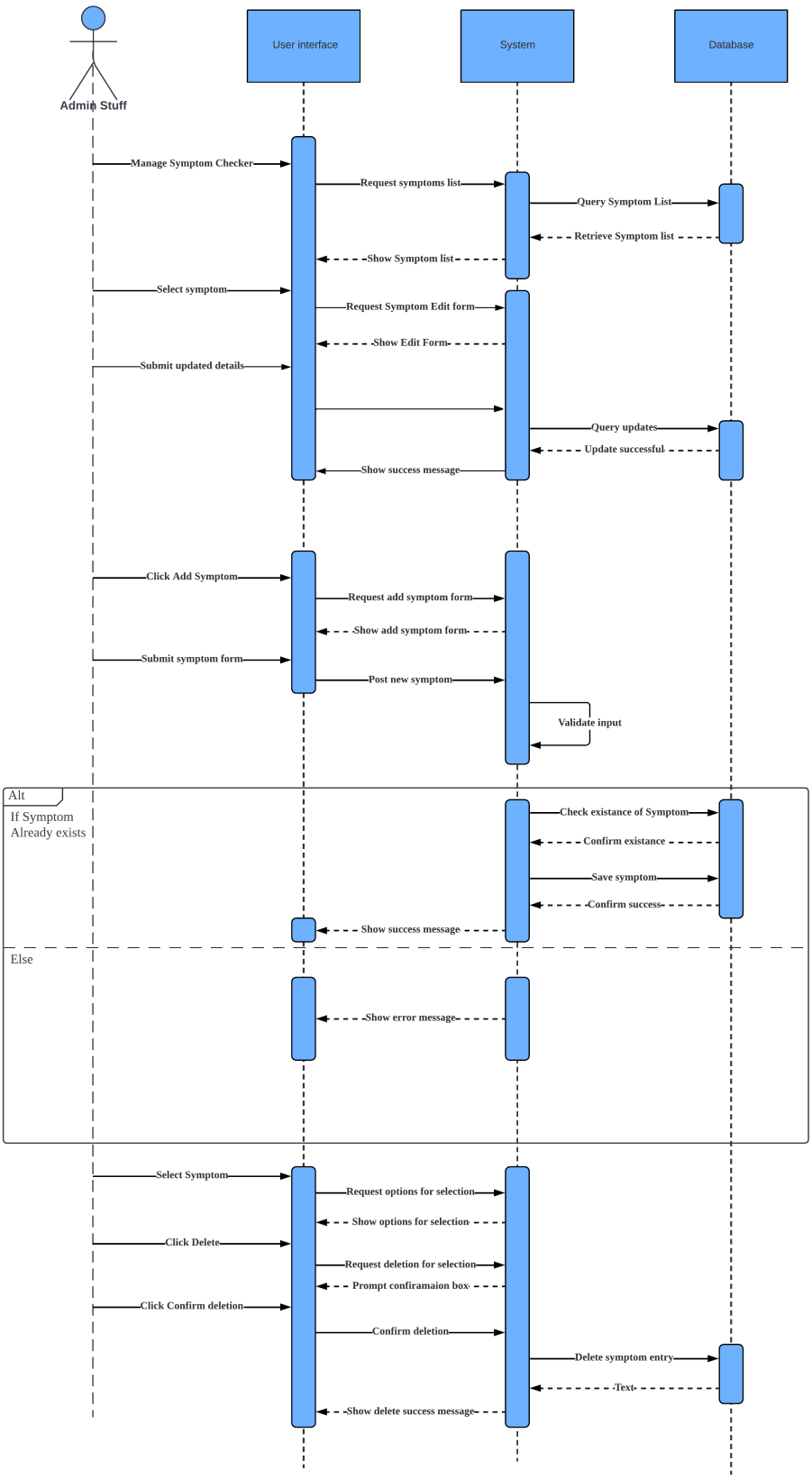


Figure 14: Manage Symptom Checker Sequence Diagram

2.5.12 Manage HCP

Table: REQ_F008

Requirement ID	REQ_F008	**Version**	1.0
Description	Allows the admin staff to add, delete, or update healthcare providers or their detail availability.		
Actor	Admin Staff		
Author	Abdullah		

Table: Manage Health Care Providers List Use Case

Feature	Manage Health Care Providers List	**Version**	1.0
Purpose	To enable admin staff to manage health care providers, including adding, deleting, and updating HCP details.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the HCP management section.		
Postcondition	Health care providers are successfully managed (added, updated, or deleted) and reflected in the system.		
Main flow	**Adding HCP:** 1. Admin staff requests to add a new HCP. 2. The system shows the "Add HCP" form. 3. Admin staff fills up the form and clicks submit. 4. The system validates input and checks for HCP existence in the database. 5. If HCP doesn't exist: a. The system confirms insertion and displays a success message. 6. If HCP exists, an error message is displayed. **Deleting HCP:** 1. Admin staff clicks "Remove HCP". 2. The system requests removal with HCP ID. 3. The system deletes the HCP entry in the database. 4. The system shows a deletion success message. **Updating HCP:** 1. Admin staff clicks "Update HCP". 2. The system displays the list of HCPs. 3. Admin staff selects an HCP and requests the update form.		

	4. The system shows the update form. 5. Admin staff submits updated details. 6. The system updates the HCP details in the database. 7. The system shows a success message.
Alternate Scenario	If the HCP already exists during addition, show an error message and halt the process.
Author	Abdullah

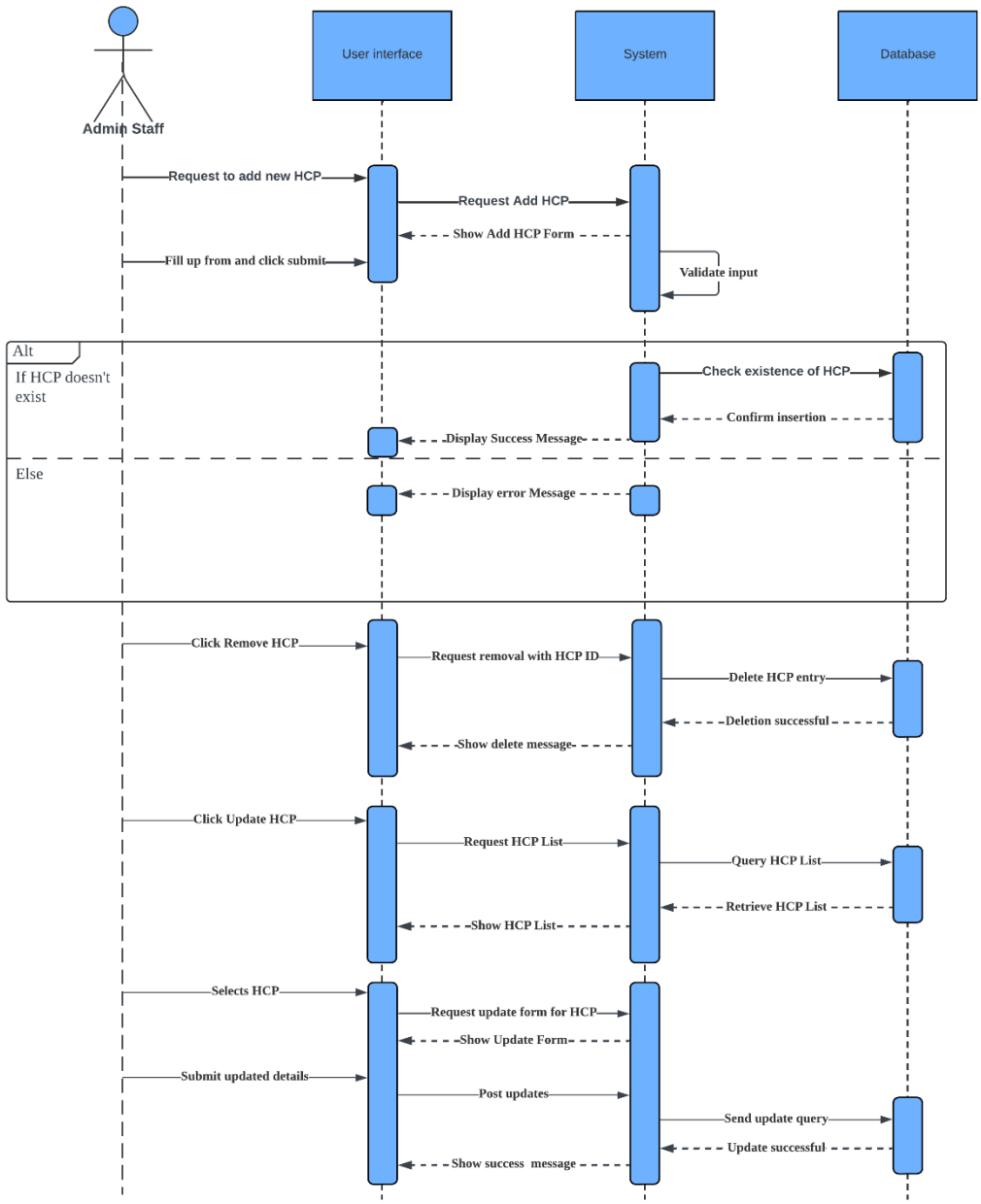


Figure 15: Manage HCP List Sequence Diagram

2.5.13 Manage appointments

Table: REQ_F019

Requirement ID	REQ_F019	**Version**	1.0
Description	Allows admin staff to manage appointments, including scheduling, modifying, res and viewing history on behalf of users.		
Actor	Admin Staff		
Author	AL BARAA		

Table: Manage Appointments Use Case

Feature	Manage Appointment	**Version**	1.0
Purpose	To enable admin staff to manage appointments for users, including cancelling existing appointments.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the appointment management section.		
Postcondition	The appointment is successfully managed (e.g., canceled) and reflected in the system.		
Main flow	1. Admin staff clicks "Manage Appointment". 2. The user interface requests the list of university community (UC) members. 3. The system queries the database for the UC list. 4. The database retrieves and returns the UC list to the system. 5. The system displays the UC list to the admin staff. 6. Admin staff selects a UC member. 7. The system requests the appointment list for the selected UC member. 8. The system queries the database for the appointments. 9. The database returns the appointment list to the system. 10. The system displays the UC member's appointment list. 11. Admin staff selects an appointment to cancel. 12. The system acknowledges the selection and prompts for confirmation. 13. Admin staff confirms the cancellation. 14. The system requests cancellation from the HCP. 15. The HCP confirms the cancellation. 16. The system updates the appointment status to "canceled" in the database.		

	17. The system displays a cancellation success message to the admin staff.
Alternate Scenario	None specified.
Author	AL BARAA

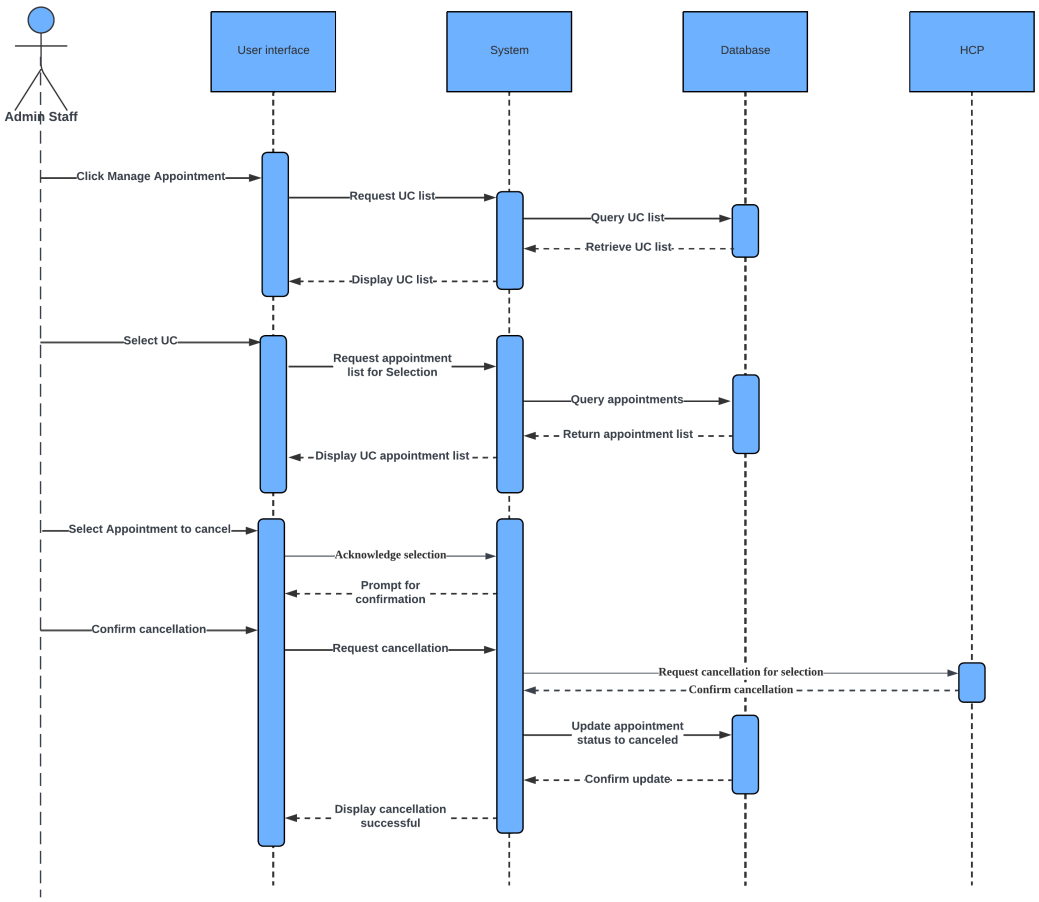


Figure 16: Manage Appointments Sequence Diagram

2.5.14 View medical records

Table: REQ_F013

Requirement ID	REQ_F013	**Version**	1.0
Description	Allows admin staff to view medical records of university community members.		
Actor	Admin Staff		
Author	Kawsar		

Table: View Medical Records Use Case

Feature	View Medical Records	**Version**	1.0
Purpose	To enable admin staff to view medical records of university community members.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the medical records section.		
Postcondition	Admin staff successfully views the medical records of the selected university community member.		
Main flow	1. Admin staff clicks "View Medical Records". 2. The user interface sends a request to the system to retrieve the list of university community (UC) members. 3. The system retrieves and displays the UC list. 4. Admin staff selects the intended UC member. 5. The system requests the medical records (MR) for the selected UC member. 6. The system queries the Electronic Medical Record (EMR) system with the appropriate attributes. 7. The EMR retrieves and sends the medical records to the system. 8. The system displays the medical records to the admin staff.		
Alternate Scenario	None specified.		
Author	Kawsar		

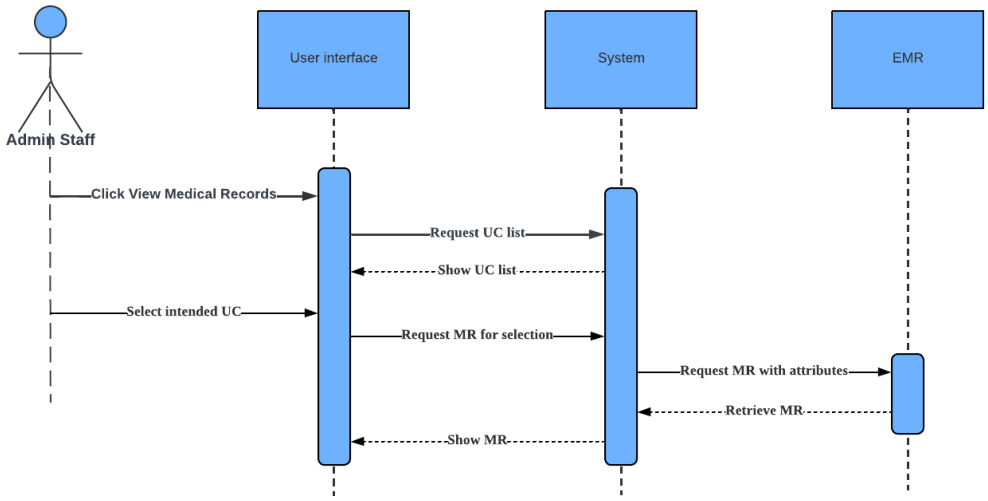


Figure 17: View Medical Records Sequence Diagram

2.5.15 Billing and Payment

Table: REQ_F014

Requirement ID	REQ_F014	**Version**	1.0
Description	Allows the user to process billing for health services rendered and manage payme insurance.		
Actor	Admin Staff		
Author	Kawsar		

Table: Billing and Payment Use Case

Feature	Billing and Payment	**Version**	1.0
Purpose	To enable the admin staff to manage billing for health services, check insurance st process payments or claims.		
Actor	Admin Staff		
Precondition	Admin staff is logged in and navigates to the billing section.		
Postcondition	Billing is successfully managed, and insurance claims are processed if applicable.		
Main flow	1. Admin staff clicks "View Billing". 2. The user interface requests the appointment list from the system. 3. The system retrieves and displays the appointment list. 4. Admin staff selects an appointment. 5. The system requests billing details from the HCP. **Alternate Flow:** - **If Billing is Pending:** a. The system shows a pending message. - **If Billing Exists:** a. The admin staff clicks "Check Insurance". b. The system checks insurance status. c. If insurance doesn't exist, show a "No Insurance" message. d. If insurance exists, show insurance details. e. Admin staff clicks "Claim Insurance". f. The system initiates the insurance claim process. **Insurance Claim Processing:** g. If the insurance is disapproved, show a disapproved message. h. If approved, allow admin staff to submit the bill.		

	i. The system posts bill submission and confirms. j. Display receipt to the admin staff.
Alternate Scenario	1. If billing is pending, display a pending message. 2. If insurance doesn't exist, display a "No Insurance" message. 3. If insurance is disapproved, show a disapproved message.
Author	Kawsar

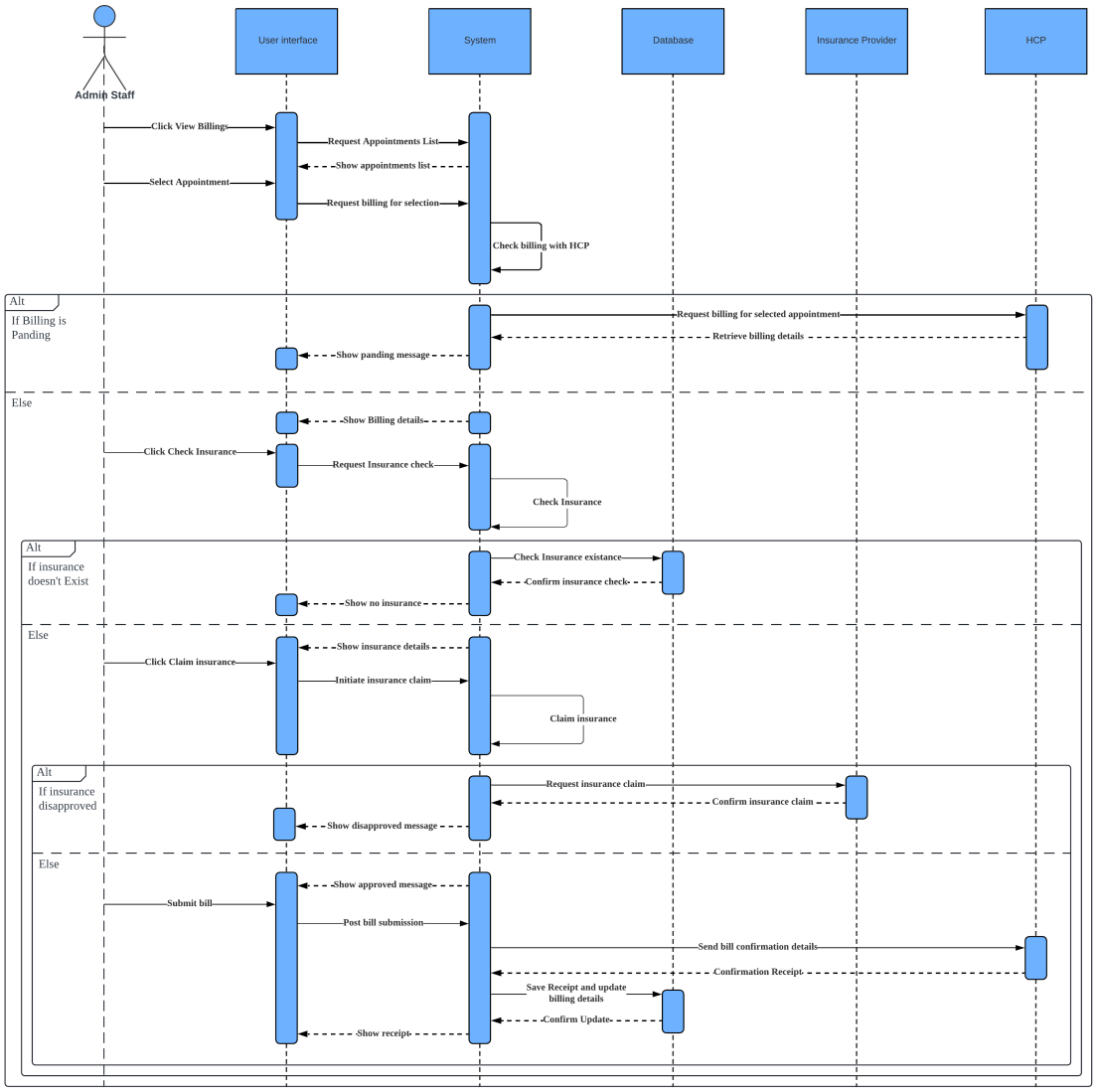


Figure 18: Billing and Payment Sequence Diagram

2.5.16 View Analytics

Table: REQ_F016

Requirement ID	REQ_F016	**Version**	1.0
Description	Provides an analytics dashboard to monitor appointment statistics and feedback for continuous improvement.		
Actor	System Admin		
Author	Youssef		

Table: View Analytics Use Case

Feature	View Analytics	**Version**	1.0
Purpose	To allow the system admin to view analytics data related to appointment statistics and user feedback for monitoring and continuous improvement.		
Actor	System Admin		
Precondition	System admin is logged in and navigates to the analytics section.		
Postcondition	System admin successfully views the analytics data.		
Main flow	1. The system admin clicks "View Analytics". 2. The user interface sends a request to the system for analytics data. 3. The system queries the database to retrieve the relevant analytics data. 4. The database returns the analytics data to the system. 5. The system displays the analytics data to the system admin.		
Alternate Scenario	None specified.		
Author	Youssef		

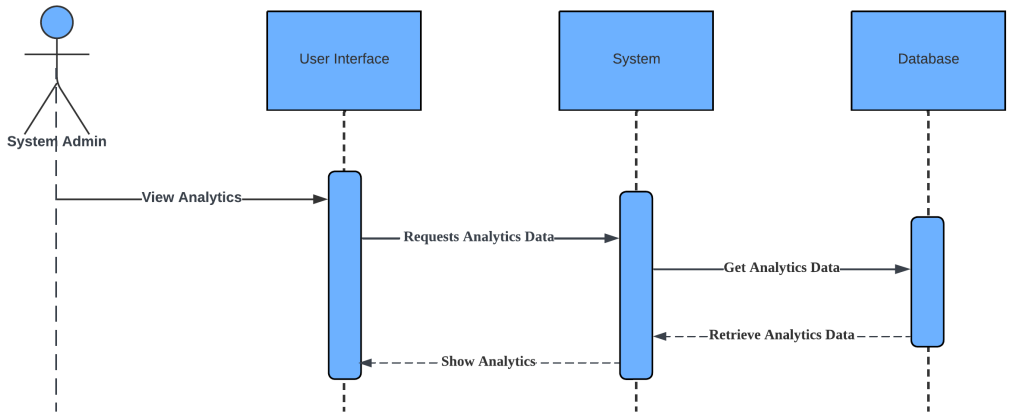


Figure 19: View Analytics Sequence Diagram

2.5.17 View feedback

Table: REQ_F017

Requirement ID	REQ_F017	**Version**	1.0
Description	Allows system administrators to view feedback submitted by users (students, lectu improve services and user experience.		
Actor	System Admin		
Author	Abdullah		

Table: View Feedback Use Case

Feature	View Feedback	**Version**	1.0
Purpose	To enable the system admin to view feedback submitted by users for the purpose of improving services and user experience.		
Actor	System Admin		
Precondition	System admin is logged in and navigates to the feedback section.		
Postcondition	System admin successfully views the feedback list.		
Main flow	1. The system admin clicks "View Feedback". 2. The user interface sends a request to the system to get feedback data. 3. The system requests feedback data from the database. 4. The database retrieves and sends the feedback data to the system. 5. The system displays the feedback list to the system admin.		
Alternate Scenario	None specified.		
Author	Abdullah		

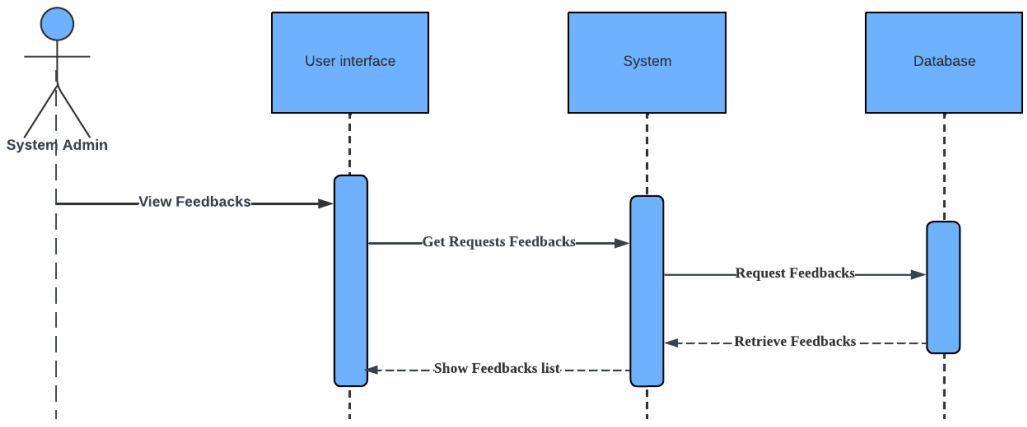


Figure 20: View Feedback Sequence Diagram

2.5.18 Manage announcements

Table: REQ_F007

Requirement ID	REQ_F007	**Version**	1.0
Description	Allows the system admin to add, delete, or update announcement details.		
Actor	System Admin		
Author	AL BARAA		

Table: Manage Announcement Use Case

Feature	Manage Announcements	**Version**	1.0
Purpose	To enable the system admin to manage announcements, including adding and deleting announcements.		
Actor	System Admin		
Precondition	System admin is logged in and navigates to the announcements management section.		
Postcondition	Announcements are successfully managed (added or deleted) and reflected in the system.		
Main flow	**Adding an Announcement:** 1. System admin clicks "Manage Announcements". 2. The user interface requests the announcement list from the system. 3. The system retrieves and displays the announcement list. 4. The system admin clicks "Add Announcement". 5. The system displays the announcement form. 6. The system admin inserts announcement details. 7. The system posts the announcement details to the database. 8. The database saves the announcement details and confirms the success. 9. The system displays a success message. **Deleting an Announcement:** 1. The system admin clicks "Delete Announcement". 2. The system sends a delete request with the announcement ID. 3. The database deletes the announcement.		

	4. The database confirms the deletion. 5. The system displays a delete success message.
Alternate Scenario	None specified.
Author	AL BARAA

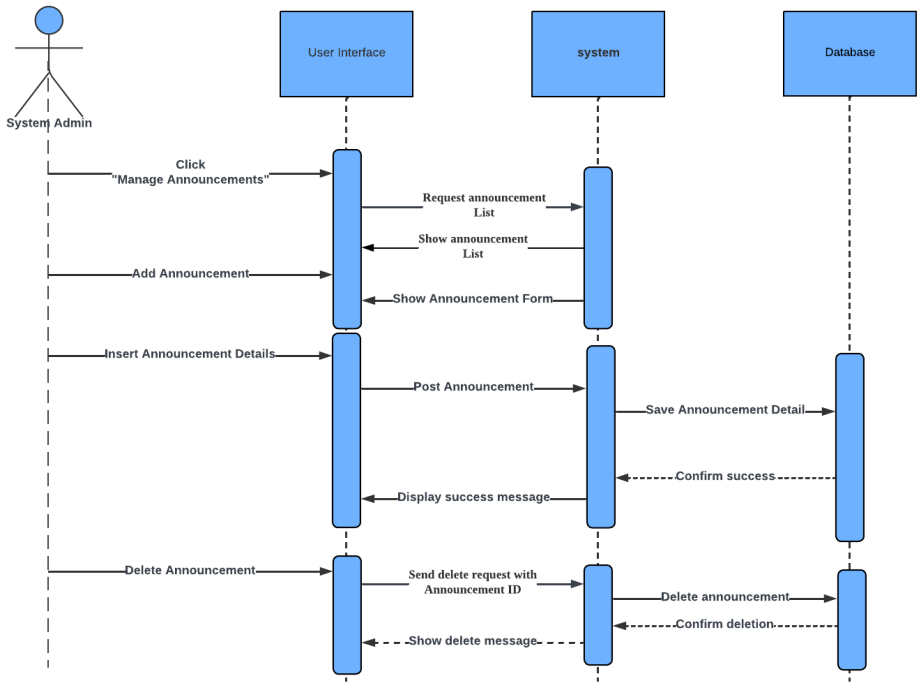


Figure 21: Manage Announcements Sequence Diagram

2.6 Usability requirements

The **UHSS** is designed with specific usability requirements to ensure a user-friendly experience for students, healthcare providers, and administrative staff. The system's usability objectives focus on measurable aspects of **effectiveness**, **efficiency**, **user satisfaction**, and **avoiding harm**.

Description	Priority	Author
Users must easily achieve their goals (e.g., scheduling appointments, accessing medical records, receiving healthcare provider suggestions) with minimal errors and successfully complete tasks.	High	Kawsar
95% of users should successfully schedule appointments without assistance.	High	Kawsar
The symptom checker should provide accurate healthcare provider suggestions for 90% of the reported symptoms based on medical protocols.	High	Kawsar
Users should retrieve and view their medical records with a 98% success rate.	High	Abdullah
Users should complete the appointment scheduling process in less than 3 minutes on average.	Medium	Abdullah
The system should provide symptom-based healthcare provider suggestions within 5 seconds.	Medium	Kawsar
Healthcare providers and users should access medical records in less than 2 seconds.	High	Al Baraa
User feedback should result in an 85% or higher satisfaction rating based on surveys.	Medium	Al Baraa
The system should score at least 80 on the System Usability Scale (SUS).	Medium	Youssef
At least 70% of users should submit feedback on their experience with the system.	Low	Youssef
The system must flag incorrect or incomplete data entries immediately and notify users without causing delays.	High	Al Baraa
The system must comply with all relevant healthcare data protection laws to protect sensitive information.	High	Youssef
The system must ensure that users are matched with relevant healthcare providers with a less than 1% mismatch rate.	High	Youssef
The system should support screen readers for visually impaired users.	Medium	Al Baraa
All major system functions should be fully navigable using keyboard shortcuts.	Medium	Al Baraa

2.7 Performance requirements

The table below outlines the **performance requirements** for the **University Health Service System**. These requirements include both static and dynamic aspects, ensuring that the system can efficiently support users while maintaining scalability, data security, and fast response times for all critical functions. Each requirement is associated with specific performance metrics to guide the system's design and implementation.

Requirement Type	Description	Author
Scalability	The system must support 10,000 concurrent users without significant performance loss. The database will handle 100,000 new records annually while maintaining efficiency.	Kawsar
Storage	The system will maintain 1 TB of storage capacity and expand up to 5 TB as the database grows. The system will store medical records, appointment logs, and user data for 10 years.	Abdullah
Data Backup	The system will perform daily incremental backups and weekly full backups to prevent data loss.	Al Baraa
System Response Time	Users will receive responses for queries (appointment scheduling, medical record access) within 2 seconds. For complex operations like insurance verification, the system will respond within 5 seconds.	Youssef
Symptom Checker	The system will process symptoms and suggest healthcare providers within 5 seconds.	Kawsar
Appointment Booking	Users should complete the appointment booking process, including selecting a provider and confirming the time slot, within 3 minutes.	Kawsar

Medical Record Access	Users and healthcare providers will access medical records in less than 2 seconds, regardless of record size.	Al Baraa
Billing and Insurance	The system will complete insurance verification within 5 seconds and generate billing information within 3 seconds following an appointment.	Abdullah
Concurrent Users	The system will support 5,000 concurrent users accessing records and scheduling appointments without performance issues.	Al Baraa
Database Queries	The system will complete simple queries (e.g., retrieving appointment history) in 1 second and complex queries (e.g., cross-referencing records and billing) in 3 seconds.	Youssef
Notification Delivery	The system will deliver notifications (appointment reminders, updates) to users via email within 1 minute of the event trigger.	Al Baraa

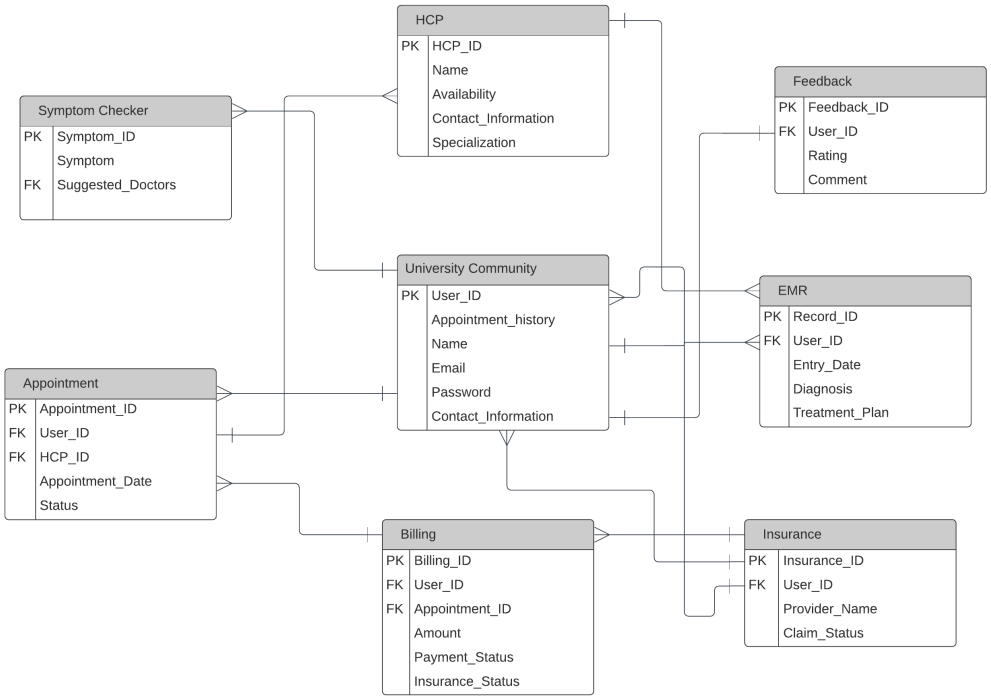
2.8 Logical database requirements

2.8.1 Entity-Relationship Diagram (ERD):

The **University Health Service System** ERD illustrates the relationships between the core entities involved in managing healthcare services for the university community. The **University Community** entity represents the users, including students, faculty, and staff, who interact with the system. It stores user-specific data such as personal information, medical history, and contact details. The **Healthcare Providers (HCP)** entity manages information about the doctors and medical professionals, including their specialization, availability, and contact information. Users can interact with the **Symptom Checker**, which allows them to input symptoms and receive doctor suggestions. The **Symptom Checker** connects with both the user data and healthcare providers to make relevant recommendations.

Appointments are managed through the **Appointments** entity, which links the users with healthcare providers and tracks the date, time, and status of each appointment. Following appointments, medical information such as diagnoses and treatment plans are stored in the **Electronic Medical Records (EMR)** entity, which maintains users' health data securely.

The system also handles financial aspects through the **Billing** entity, which records payment amounts, statuses, and insurance information for each appointment. Insurance details are further managed by the **Insurance Verification** entity, ensuring that claims are processed accurately. Lastly, users can provide feedback through the **Feedback** entity, which captures ratings and comments about their healthcare experiences. All entities are interconnected, ensuring seamless data flow and efficient management of healthcare services within the university.



2.8.2 Data Flow Diagram (Level 0)

Figure 1 refers to the **Context Diagram** (also known as **DFD Level 0**) for the **University Health Service System (UHSS)**. This diagram provides an overview of the entire system by showing the main interactions between external entities and the system itself. The system serves as a central process, interacting with multiple external entities, such as **University Community**, **Admin Staff**, **Healthcare Providers**, **Insurance Providers**, **System Administrator**, and the **EMR (Electronic Medical Records) System**.

The **University Community** can access several services within the system, including managing their profiles, scheduling appointments, viewing medical records, and submitting feedback. The **Admin Staff** plays a key role in managing user accounts, updating medical records, and handling appointments and billing. The **Healthcare Providers** receive appointment details and update medical records, which are stored in the **External EMR System** and later retrieved by the **University Community**. The system also interacts with **Insurance Providers** to process insurance claims and manage billing information.

The **System Administrator** is responsible for overseeing the system's configuration, handling feedback reports, and ensuring that the system operates smoothly. The **EMR System** interacts with the UHSS to provide updated medical records, ensuring that users have access to their most recent medical history.

The arrows in the diagram represent the data flow between these entities and the UHSS. For example, **Appointment Requests** flow from the **University Community** to the UHSS, while **Appointment Confirmations** flow back from the system to the users. Similarly, **Billing Information** is sent to the **Insurance Providers**, and **Medical Records** are updated and retrieved through the **EMR System**.

2.8.3 Data Flow Diagram (Level 1)

The **University Health Service System (UHSS) Level 1 Data Flow Diagram (DFD)** illustrates the various interactions between processes, external entities, and data stores within the system. The **User Management** process is handled primarily by the **Admin Staff**, who register new users and update their profiles. User credentials are stored in the **User Data** store, allowing the **University Community** to log in and manage their profiles. Upon login or profile updates, the system confirms the action by sending back a **Login Confirmation** or **Profile Update Confirmation**.

In the **Symptom Checking** process, the **University Community** submits their symptoms, which are stored in the **Symptom Data** store. Based on these symptoms, the system suggests relevant healthcare providers and sends **Doctor Suggestions** back to the user. The system manages and stores this interaction between reported symptoms and suggested providers efficiently in the **Symptom Data** store.

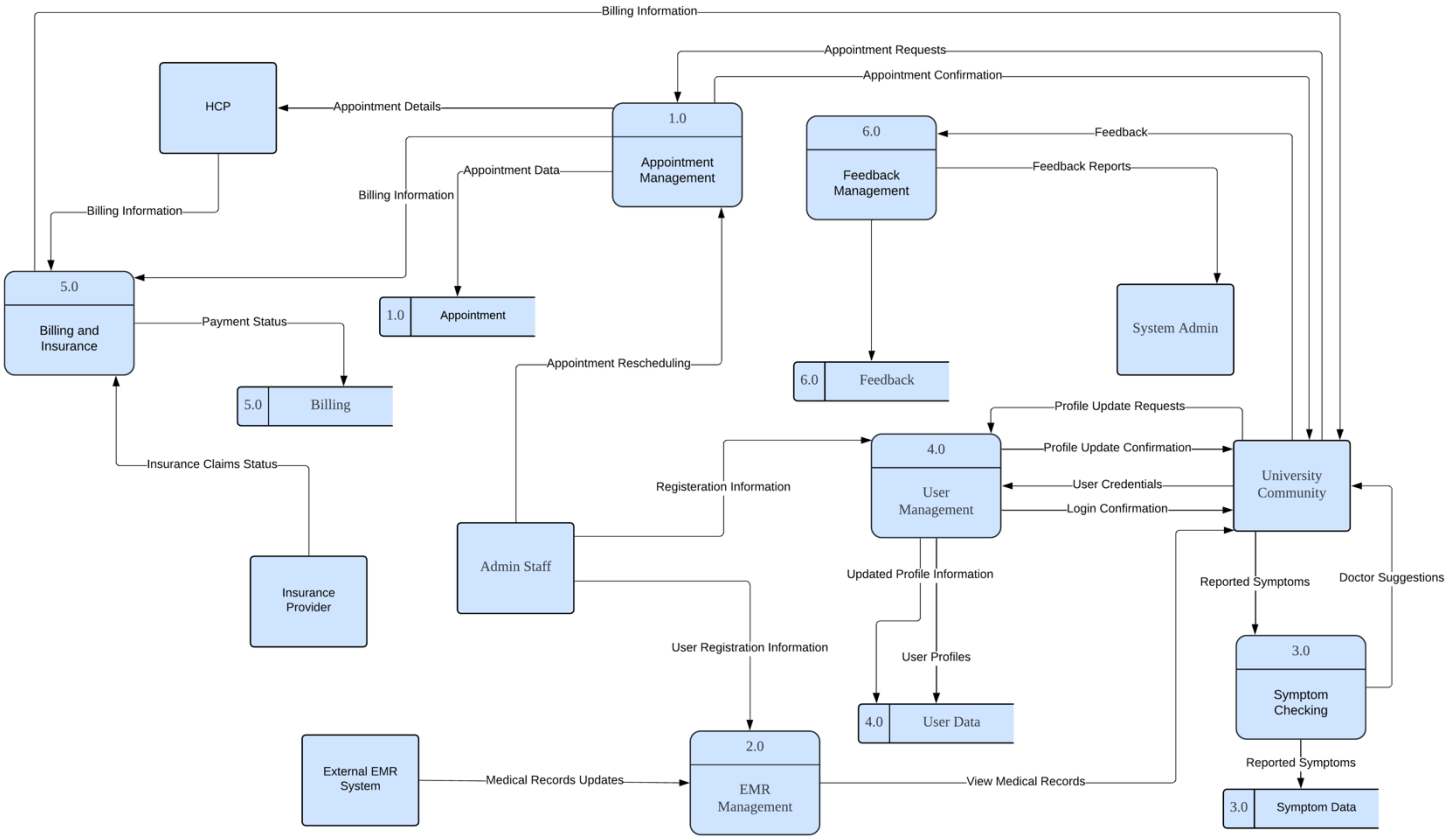
The **Appointment Management** process is central to the system, enabling users to request, cancel, or modify appointments. **University Community** members can send **Appointment Requests**, which are processed by the system and stored in the **Appointment** data store. **Admin Staff** also have the ability to reschedule or manage appointments on behalf of users. After an appointment is booked or cancelled, the **Healthcare Providers** receive **Appointment Details**, ensuring they stay updated on their schedules.

The **EMR Management** process interacts with an external EMR system. Once **Admin Staff** registers a new user, their information is sent to the **External EMR System**, which manages all medical records. Users can view their medical records, which are retrieved from this external system, ensuring that medical records are up to date without being stored locally within the UHSS. The system automatically pulls the latest medical information for users when they request to view their records.

Billing and Insurance is another critical component of the UHSS, involving interactions between **Healthcare Providers**, **University Community**, and **Insurance Providers**. **Healthcare Providers** send **Billing Information** to the system, which processes it and stores it in the **Billing** data store. **Insurance Providers** verify insurance claims and communicate the **Insurance Claims Status** back to the system, enabling smooth payment processing and updates to the user's payment status.

Finally, the **Feedback Management** process allows the **University Community** to provide feedback on their healthcare experiences. This feedback is stored in the **Feedback** data store, where it can be reviewed by the **System Administrator** for future improvements. **Feedback Reports** help the system adapt and improve based on user input, ensuring continuous enhancement of services.

Overall, this Level 1 DFD presents a clear view of how data flows between the various processes, external entities, and data stores within the **University Health Service System**. The diagram captures key interactions related to user registration, symptom checking, appointment management, medical record access, billing, and feedback, ensuring that all elements of the system are accounted for and operate efficiently.



2.9 Design constraint

This table outlines the key design constraints imposed on the **University Health Service System (UHSS)**. These constraints ensure compliance with regulatory requirements, data security standards, system performance expectations, and project limitations. Each constraint is designed to guide the system's development while adhering to industry standards and project-specific restrictions.

Constraint Category	Description	Author
Regulatory Compliance	The system complies with HIPAA regulations by encrypting all user data, especially medical records, and restricting access only to authorized personnel. We also ensure that users from the EU can exercise their rights under GDPR, including accessing, correcting, and deleting their personal data.	Abdullah
Data Security	We use AES-256 encryption to protect sensitive data, such as user profiles, medical records, and billing information. We also secure data transmission using TLS. Multi-factor authentication (MFA) and Role-Based Access Control (RBAC) ensure only authorized users can access certain data. Additionally, the system maintains audit trails to track critical actions, including access to medical records and appointment scheduling.	Kawsar
Interoperability	The system supports HL7 standards to ensure seamless integration with the External EMR System, enabling healthcare providers to access and update medical records. We also support FHIR standards for fast and efficient data exchange	Youssef

	with external entities such as Insurance Providers.	
Performance Constraints	The system processes user actions, such as appointment booking, within 2 seconds, even with 10,000 concurrent users. We ensure the system is scalable and capable of handling future growth in user base and data.	Youssef
Project Limitations	We develop the system within the project's budget constraints by integrating with existing systems, such as the External EMR System, to avoid unnecessary development costs. Additionally, we prioritize essential features, such as user registration and appointment management, to meet the 6-month project timeline.	Al Baraa
Compatibility and Integration	We ensure the system is compatible across various devices, including desktop and mobile, by using responsive web design. The system integrates with the university's Student Information System (SIS) and identity management system to streamline user registration and authentication.	Kawsar

2.10 Software system attributes

The **University Health Service System (UHSS)** has several essential attributes that ensure its functionality, security, and adaptability across different environments. These attributes guarantee the system's reliability, availability, security, maintainability, and portability, ensuring a smooth experience for all users, including the **University Community**, **Admin Staff**, and external entities.

a) Reliability

We design the system to ensure that it functions reliably under normal operating conditions. At the time of delivery, we guarantee that the system will meet a reliability target of **99.9% uptime**. This ensures that users can book appointments, access medical records, and interact with the system without unexpected crashes or failures. To establish reliability, we conduct extensive testing, including load testing, to simulate real-world scenarios and ensure that the system handles concurrent users effectively. Our **error-handling mechanisms** detect and resolve any unexpected issues promptly to maintain continuous service.

b) Availability

We ensure high system availability by implementing robust **checkpointing**, **recovery**, and **restart mechanisms**. The system maintains regular backups of all critical data, including appointment information, medical records, and billing data. In the event of a system failure, our recovery mechanisms allow the system to restart from the last checkpoint with minimal data loss. We guarantee an **availability level of 99.9%**, ensuring that users and healthcare providers can access the system when needed, even in case of minor disruptions. This level of availability supports both everyday healthcare interactions and critical emergency access.

c) Security

We prioritize security in every aspect of the **UHSS**. To protect the system from accidental or malicious access, we implement **multi-factor authentication (MFA)** for sensitive user accounts, particularly for **Admin Staff** and **Healthcare Providers**. We enforce **role-based access control (RBAC)**, ensuring that only authorized personnel access sensitive data, such as **medical records** and **billing information**. We use **end-to-end encryption** (AES-256) for data at rest and **TLS encryption** for data in transit to prevent unauthorized data interception or tampering. Additionally, the system logs all user activities and maintains **audit trails** to monitor and detect any unusual or malicious activity. Regular security audits help us identify potential vulnerabilities, ensuring continuous protection against evolving threats.

d) Maintainability

We design the **UHSS** to be highly maintainable by following a modular architecture. Each process, such as **Appointment Management**, **EMR Management**, and **Billing**, operates as an independent module with well-defined interfaces. This approach simplifies the maintenance process, as updates or fixes to one module can be made without disrupting the entire system. We ensure that each module adheres to low complexity standards, making the system easier to debug and improve over time. Additionally, our use of standardized communication protocols and interfaces, such as **HL7** and **FHIR**, allows easy integration with external systems like the **External EMR System** and **Insurance Providers**. This design makes it easier for developers to update and extend the system, ensuring long-term maintainability.

e) Portability

We ensure that the **UHSS** can be easily ported to different environments, including various **operating systems** and **cloud infrastructures**. The system is developed using **cross-platform technologies** and standardized APIs, which allow it to run on multiple platforms,

such as Windows, Linux, and cloud-based environments like **AWS**. Additionally, we ensure that the system can be adapted to different hardware configurations without significant rework. The system's modular design allows it to be easily integrated into other university systems or healthcare infrastructures, ensuring flexibility and adaptability as the needs of the institution grow.

Conclusion

The **UHSS** is designed with reliability, availability, security, maintainability, and portability in mind to ensure it delivers a high-quality experience to all users. By adhering to these attributes, we ensure the system meets the highest standards and continues to operate smoothly and securely over time.

2.11 Verification

To ensure the correctness and quality of the **University Health Service System (UHSS)**, we plan to use multiple verification approaches and methods. These verification actions will prove that the system meets its requirements, and that the software has been built correctly. Below is a breakdown of the verification methods, who will perform them, when they will occur, and the environment in which they will take place.

Verification Item	**Method**	**Who**	**When**	**Where**	**Criteria**
System Interfaces	Review/Inspection	Product Team	After initial development of each interface	Development Environment	Ensure proper data exchange between the UHSS and external systems (EMR, Insurance, HCP) through secure APIs.
Communication Interfaces	Integration Testing	Product Team, Vendor	After completing interface integration	University and Vendor Systems	Verify secure, encrypted data communication (e.g., HTTPS) between UHSS and external providers.

Memory Constraints	Load and Performance Testing	Product Team	After system integration	Development Environment	Ensure system handles concurrent users efficiently and meets RAM and storage requirements.
External Interfaces	System Testing	Product Team, Vendor	After integration with external systems	Development Environment	Confirm data is exchanged correctly between UHSS and external services without errors or data loss.
Reliability and Availability	Stress Testing	Product Team	Before system goes live	Staging Environment	Verify system uptime, response times, and failover mechanisms meet reliability standards (99.9%).
Security	Security Audit	Security Team, External Auditors	After completing system security implementation	Staging Environment	Ensure system complies with security requirements, including encryption, authentication, and access controls.
Maintainability	Code Review/Inspection	Product Team	After each development cycle	Development Environment	Confirm system architecture follows modular design principles and allows easy updates

					and maintenance.
Portability	Cross-platform Testing	Product Team	After development completion	Various Platforms (Windows, Linux, Cloud)	Verify that the system operates smoothly on all target platforms without reconfiguration issues.
User Interface (UI) Verification	Usability Testing	Product Team, End-Users	Before final release	Development and Testing Environments	Ensure the UI is intuitive, responsive, and meets user requirements for usability and accessibility.
Billing and Insurance Integration	System Testing	Product Team, Vendor	After integration with insurance providers	Vendor and University Systems	Validate that billing and insurance claims are processed correctly and securely through external interfaces.
Performance Requirements	Load Testing	Product Team, Vendor	Before final deployment	Staging Environment	Ensure the system meets performance requirements under peak usage conditions (e.g., during peak appointment bookings).

2.12 Verification Plan Overview

Our verification process for the **UHSS** includes a comprehensive set of methods to validate every major aspect of the system, from interface communication and memory constraints to security and usability. By involving different teams and leveraging multiple testing environments, we ensure that the system meets all reliability, performance, and security requirements before going live. Each step in the verification process is designed to systematically assess the system's correctness, ensuring that the software is built to meet its specifications.

2.13 Supporting information

2.13.1 Observation

2.13.1.1 Existing Systems Analyzed:

- Harvard University Health Services
- Florida State University Health Services
- University of Michigan Health System
- Facebook's Notification System

2.13.1.2 Key Features Observed:

- **Appointment Scheduling:** These systems offer various appointment scheduling methods, such as patient portals for online booking, call-based appointments, and in-person visits.
- **User Navigation:** Clear navigation structures were noted, featuring sections like "Get Care" and "Make an Appointment."
- **Notification Systems:** Secure messaging and notifications were used in these systems to inform patients of appointments and health services, similar to Facebook's multi-channel notification system.
- **Virtual Care:** Systems like the University of Michigan Health System offer e-visits and video visits, providing patients with remote care options.

2.13.1.3 Application in University Health Service System:

- The proposed system will integrate an online scheduling system, a comprehensive patient portal, and multi-channel notification systems. The inclusion of real-time booking availability and automated reminders are derived from the observed practices.

2.13.2 Interview

2.13.2.1 Interview Targets:

- **Students on Campus:**
  - Students expressed a preference for an integrated online booking system that includes features like calendar syncing and instant approval of appointments.
  - Preferred notification methods included email and WhatsApp, emphasizing the need for timely and personalized notifications.
- **Healthcare Providers:**

- Highlighted the necessity for a specialist guidance feature to avoid incorrect specialist bookings.
- Suggested implementing a pre-appointment questionnaire to assess symptoms accurately and direct patients to the appropriate care.

2.13.2.2 Key Insights:

- **Dissatisfiers:** Lack of a unified booking system and incorrect specialist selection.
- **Satisfiers:** Integration of a symptom-based questionnaire, calendar syncing, and instant appointment approval.
- **Delighters:** Personalized notifications via platforms like WhatsApp and an integrated symptom checker for preliminary assessments.

2.13.2.3 Integration in the System:

- These interviews informed the need for a user-friendly booking system, a symptom checker to guide patients to the correct specialists, and flexible notification methods. A decision-tree-style questionnaire will be incorporated to streamline specialist selection.

2.13.3 Brainstorming

2.13.3.1 Innovative Ideas Proposed:

Enhancing User Experience: Real-time booking availability with HCP integration.

- **Improving System Efficiency:** Real-time queue monitoring to provide users with updates on wait times.
- **Introducing Delighters:** A symptom checker that helps patients select the appropriate specialist based on their symptoms.

2.13.3.2 Categorization Based on Kano Model:

- **Delighters:** Chatbot for appointment scheduling, virtual health diary, and a loyalty program. These features were proposed to move beyond basic needs and focus on enhancing user satisfaction.

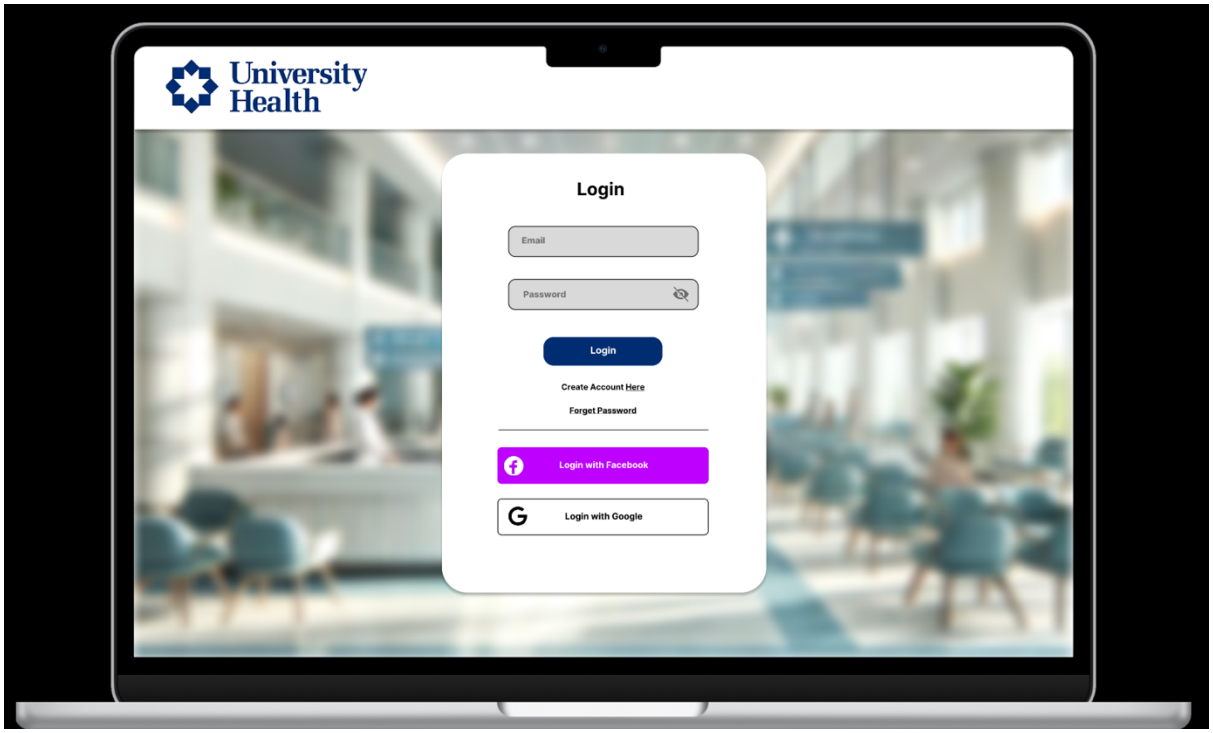
2.13.3.3 Application in the System:

- Brainstorming sessions led to the integration of innovative features such as the symptom checker and real-time queue monitoring to offer a forward-thinking health service system. The aim is to not only meet user expectations but also introduce elements of delight and surprise.

2.13.4 Prototype

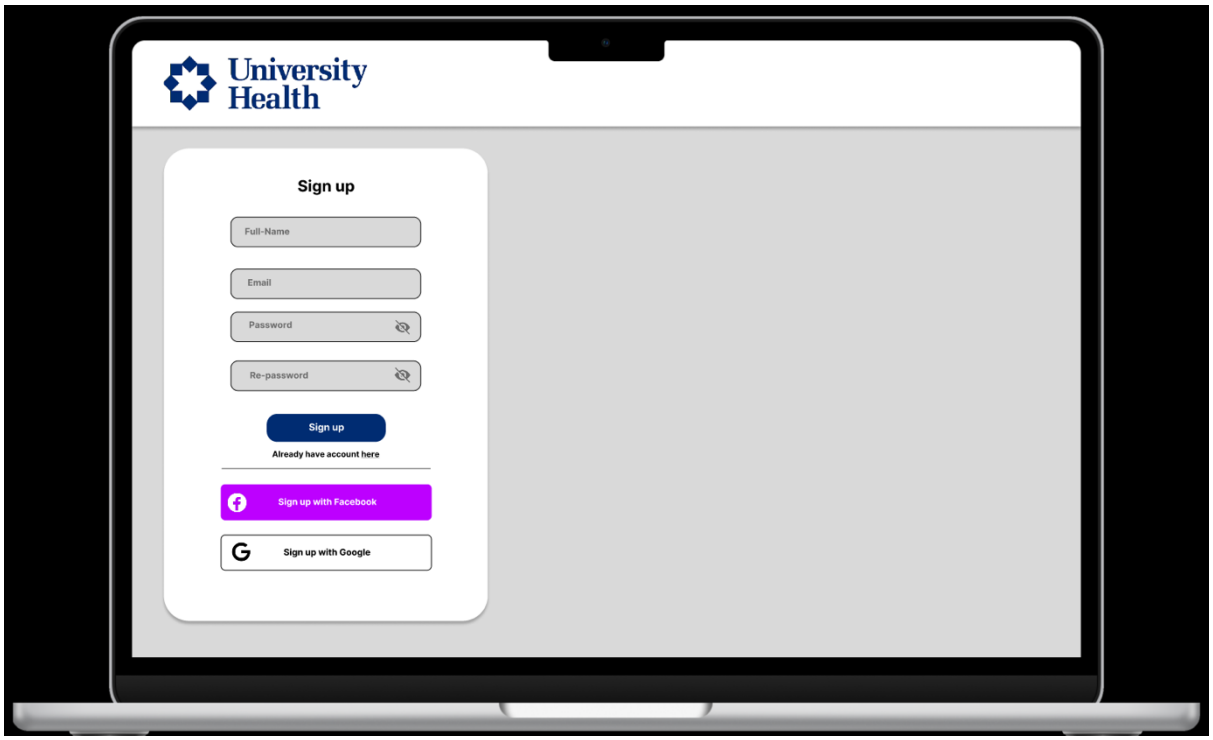
2.13.4.1 Login Page

This page allows users to log in using their email and password, or via social media (Facebook or Google). Users also have options to recover their password.



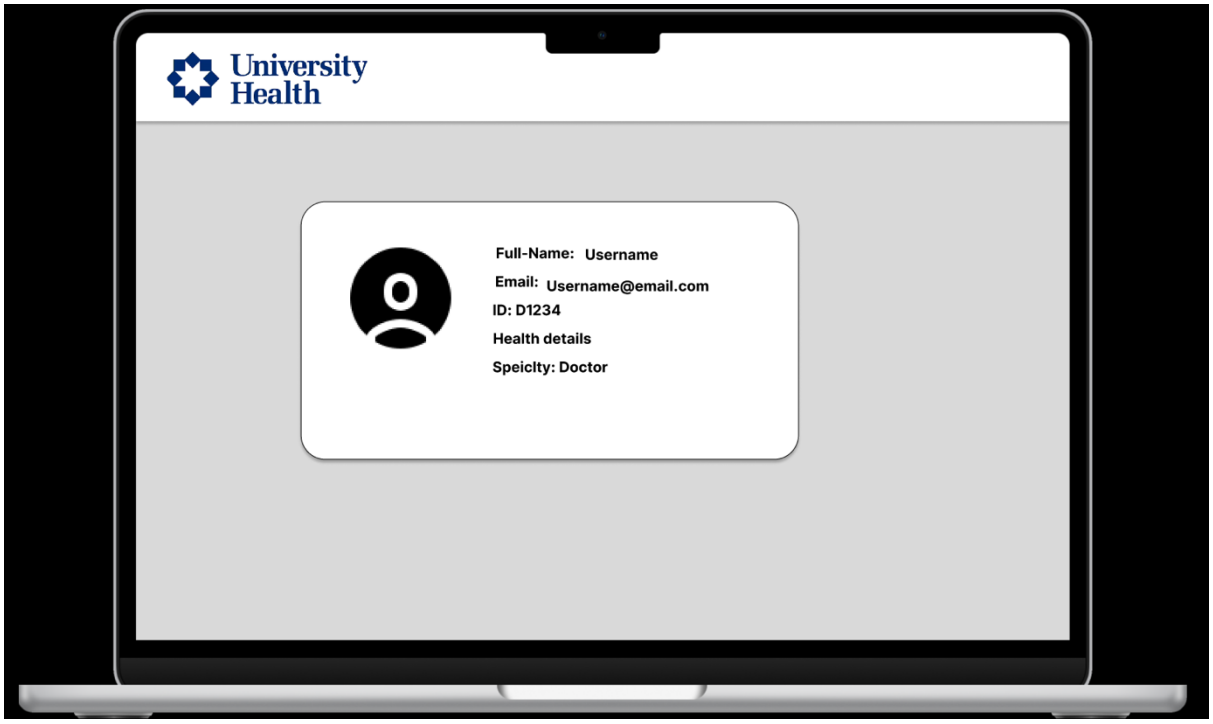
2.13.4.2 Registration Page

The registration page enables admin staff to create an account for new users by entering their full name, email, and password.



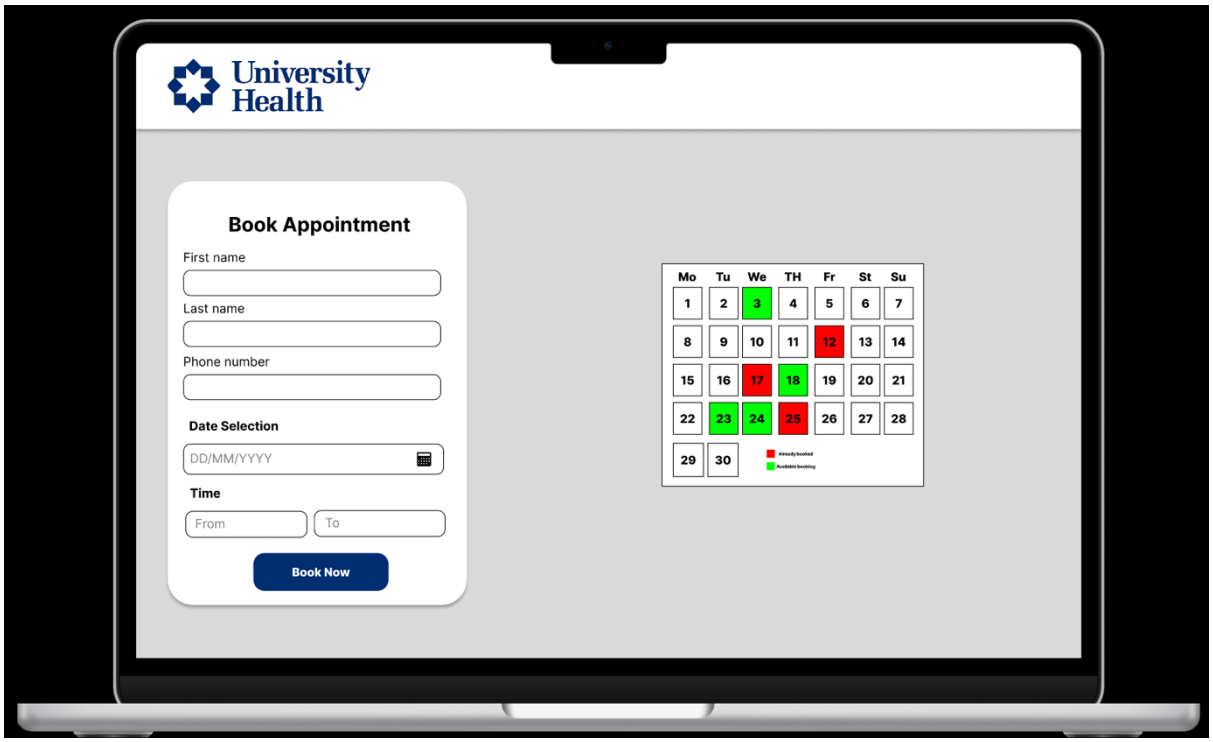
2.13.4.3 Profile Page

This page displays the user's personal information, including full name, email, user ID, and health details. It provides an overview of the user's healthcare data



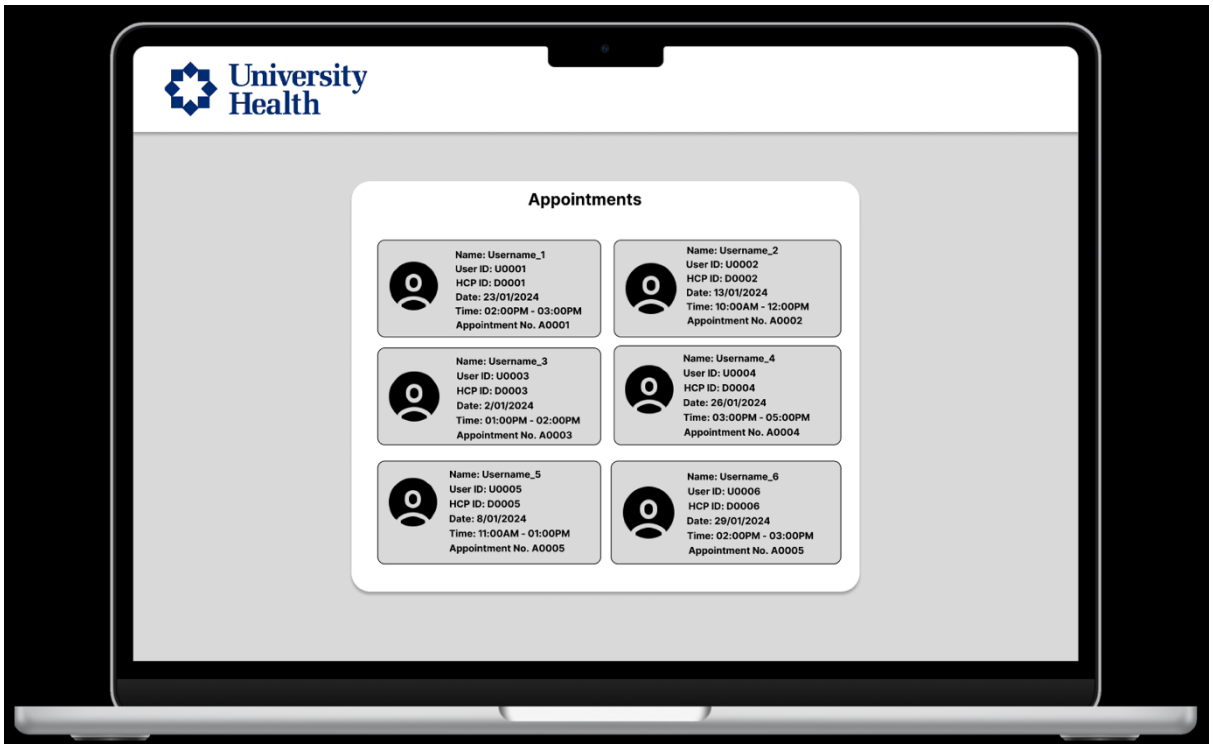
2.13.4.4 Manage Appointment Page for university community

Users can book appointments by entering their personal details and selecting a date and time. A calendar shows available booking slots (green) and unavailable or booked slots (red).



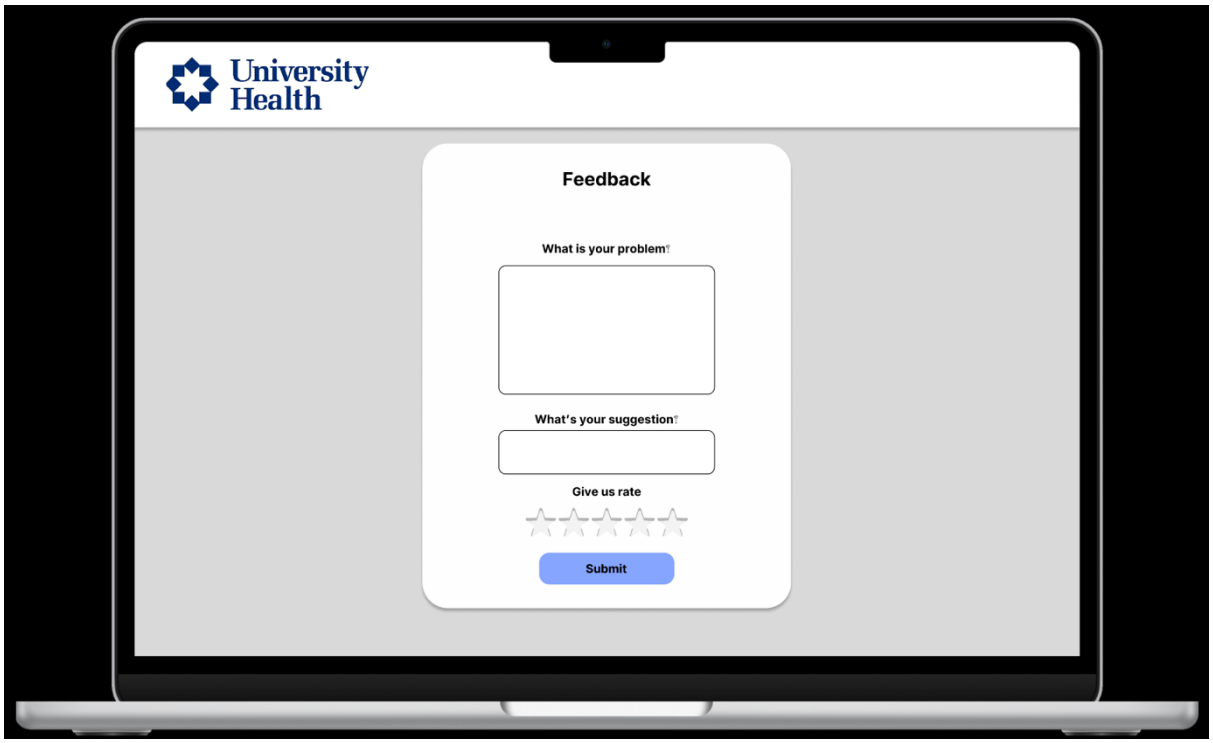
2.13.4.5 View Appointment

This page provides a list of appointments for the user



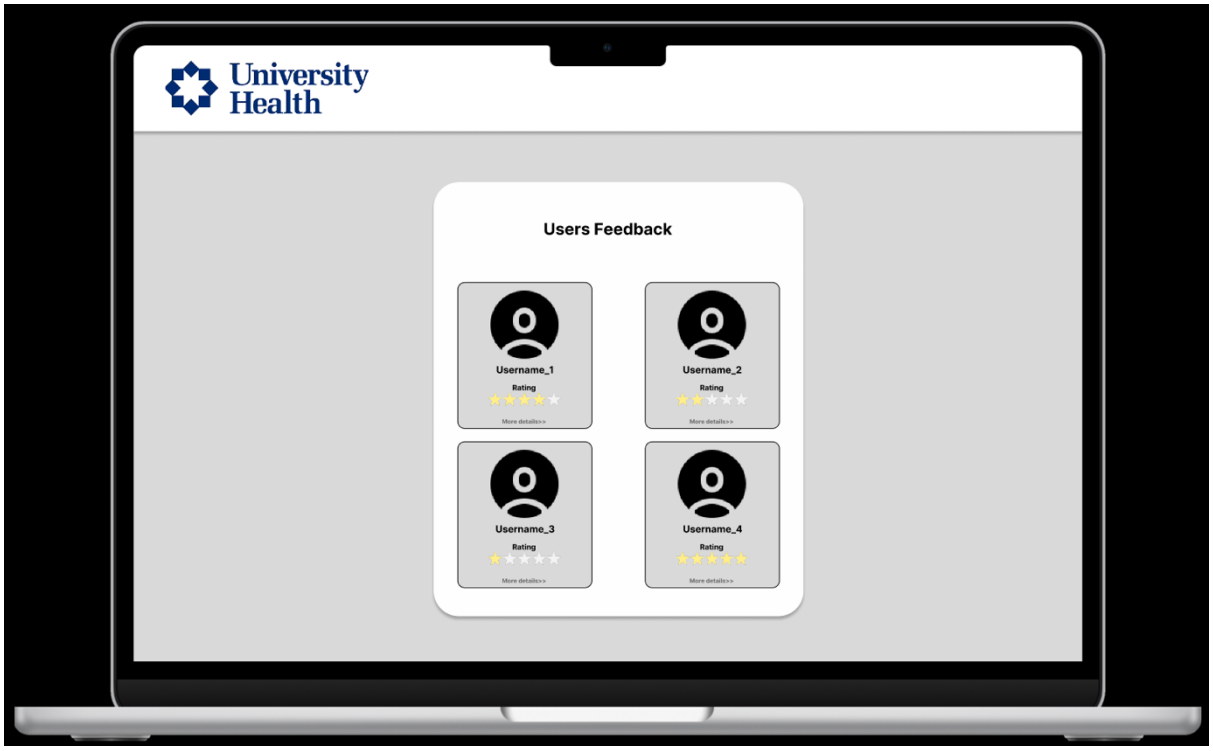
2.13.4.6 Feedback Form Page

Users can provide feedback about their experience, rate the services, and submit suggestions or concerns regarding the health services they received.



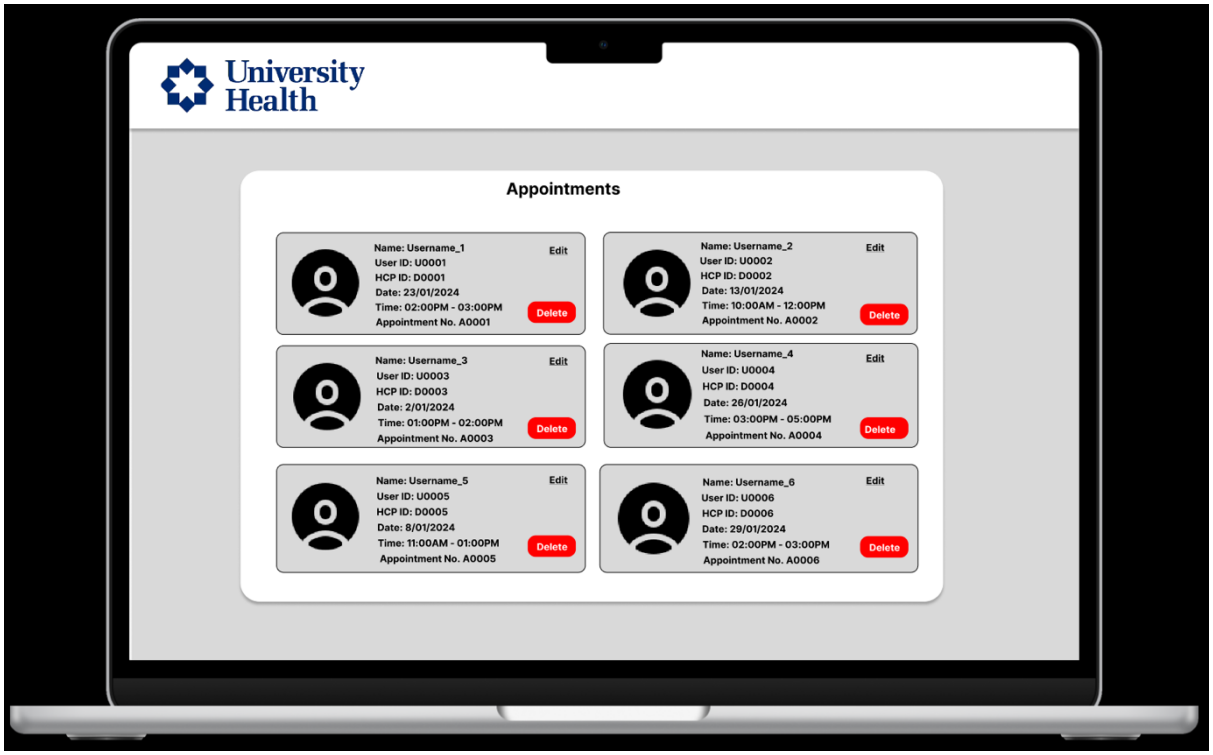
2.13.4.7 Feedback Overview Page

Displays feedback provided by other users, showing their rating and comments. System Admin can view more details about each feedback entry.



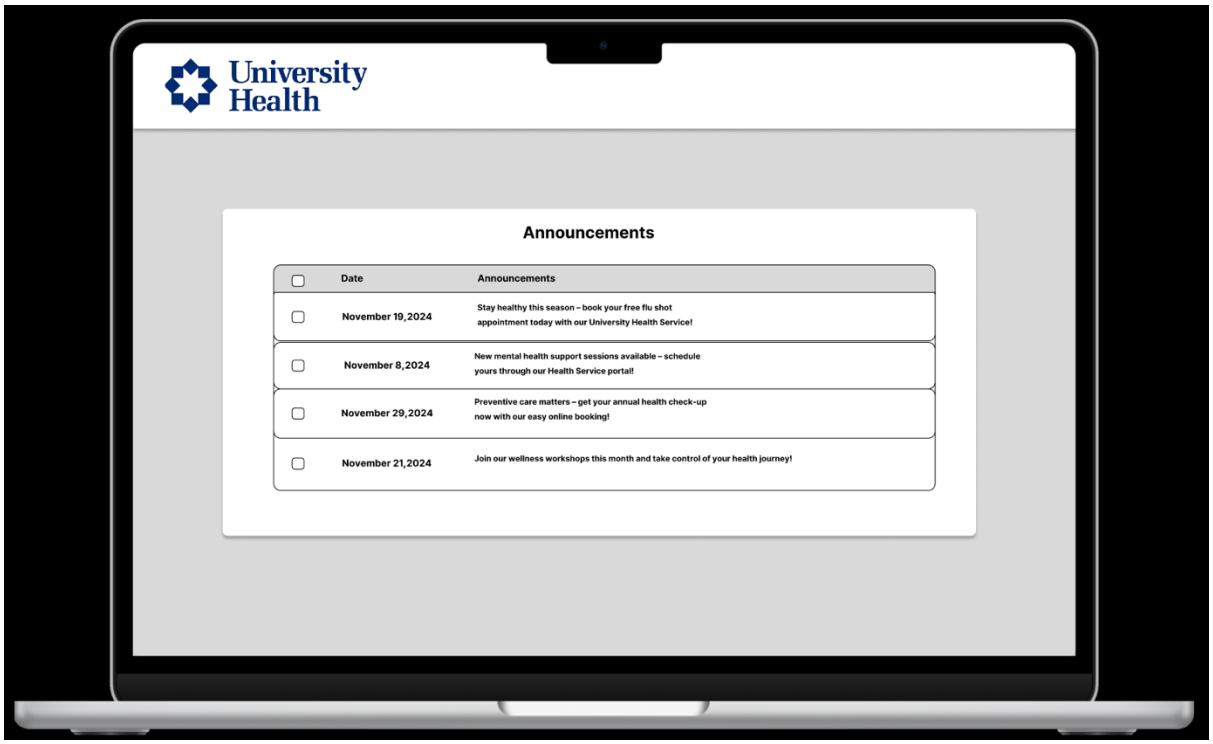
2.13.4.8 Appointment Management Page

This page shows a detailed view of all appointments, allowing users to edit or delete them. It is designed for administrators staff.



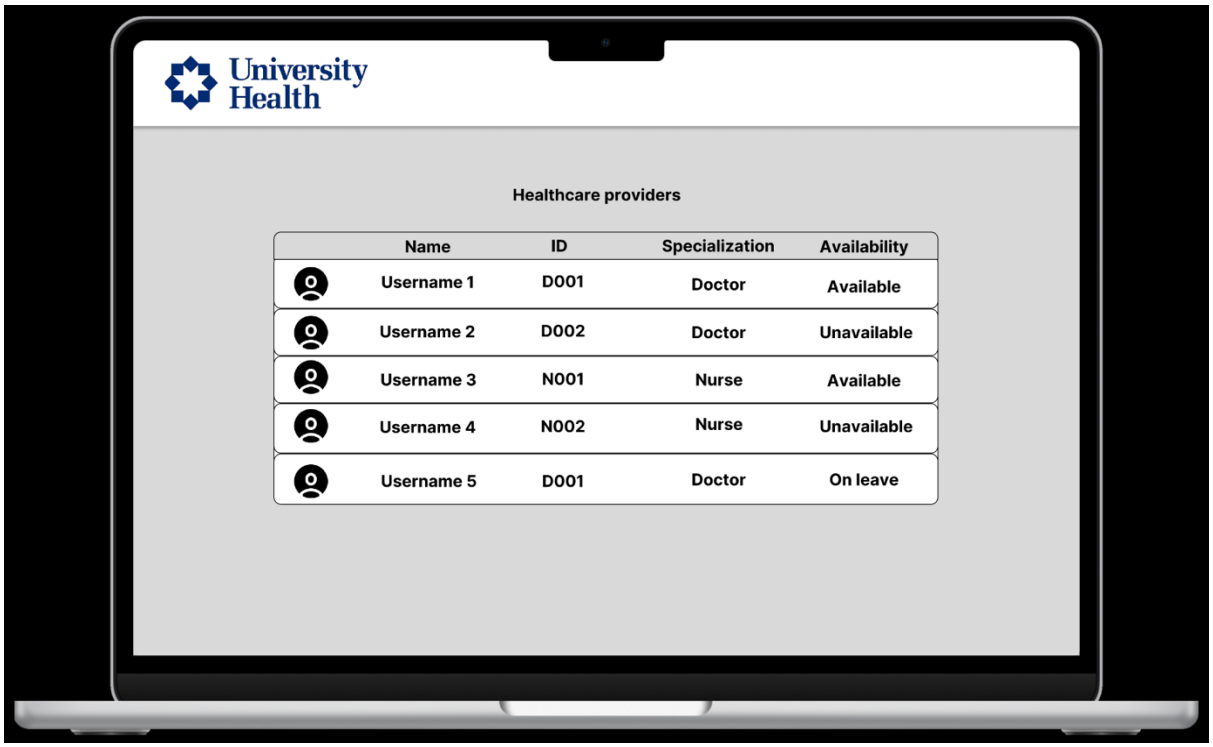
2.13.4.9 Announcement Page

Displays a list of announcements related to health services, such as upcoming flu shot campaigns or mental health support sessions. Users can stay informed about important updates.



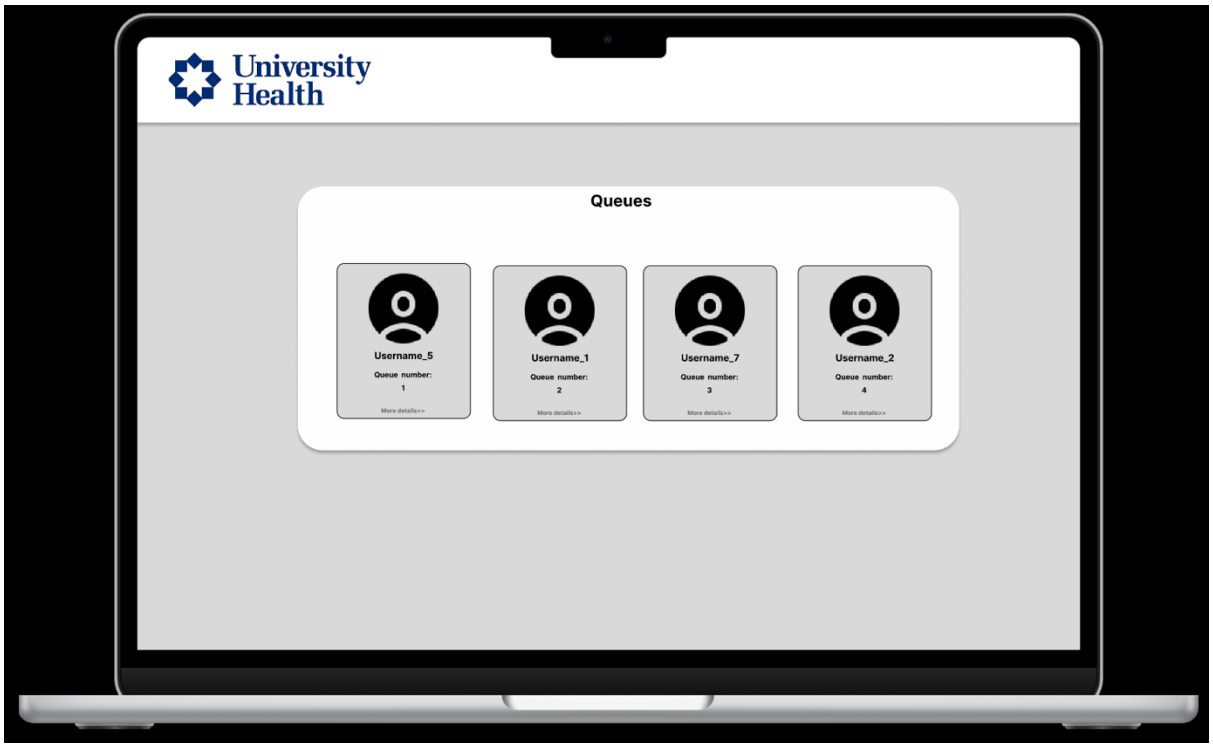
2.13.4.10 HCP List

This page shows a list of healthcare providers, including their names, ID numbers, specialization, and availability status, helping users choose the right provider for their appointments.



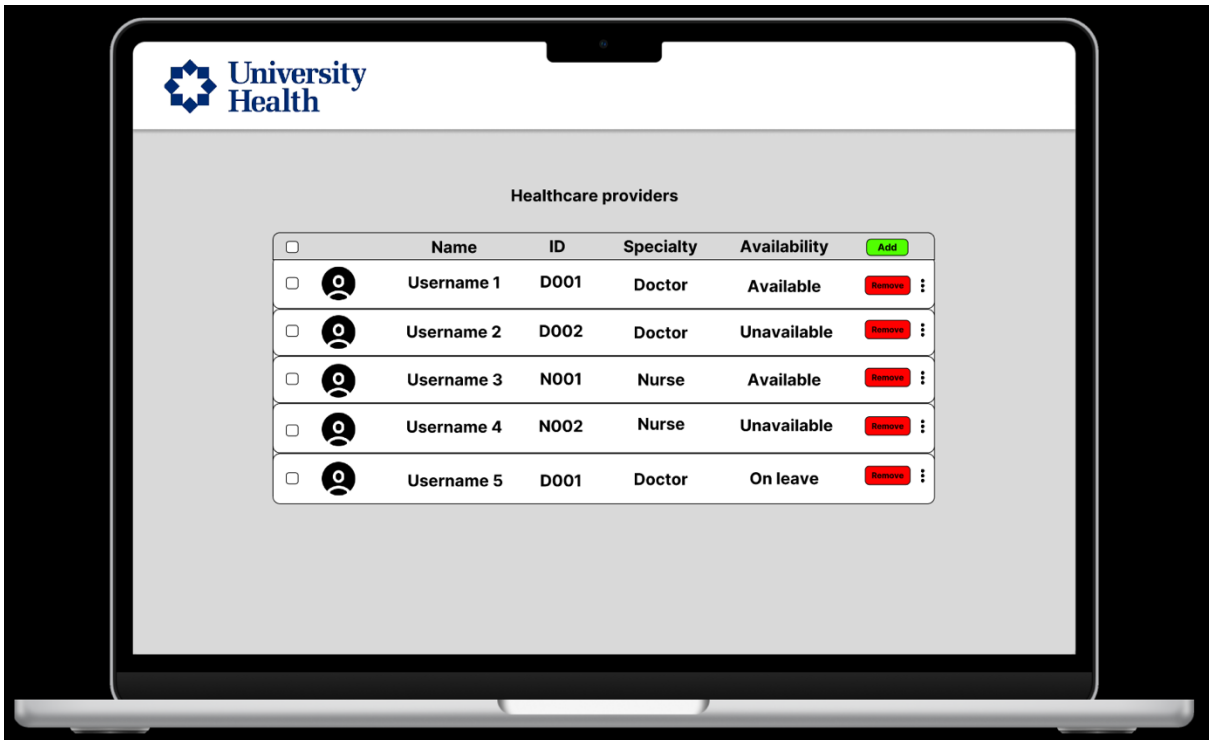
2.13.4.11 Queue Management Page

his page displays the current queue for healthcare services, listing users with their queue numbers in order of service. It helps users track their position in the queue, allowing them to know when their turn is coming up.



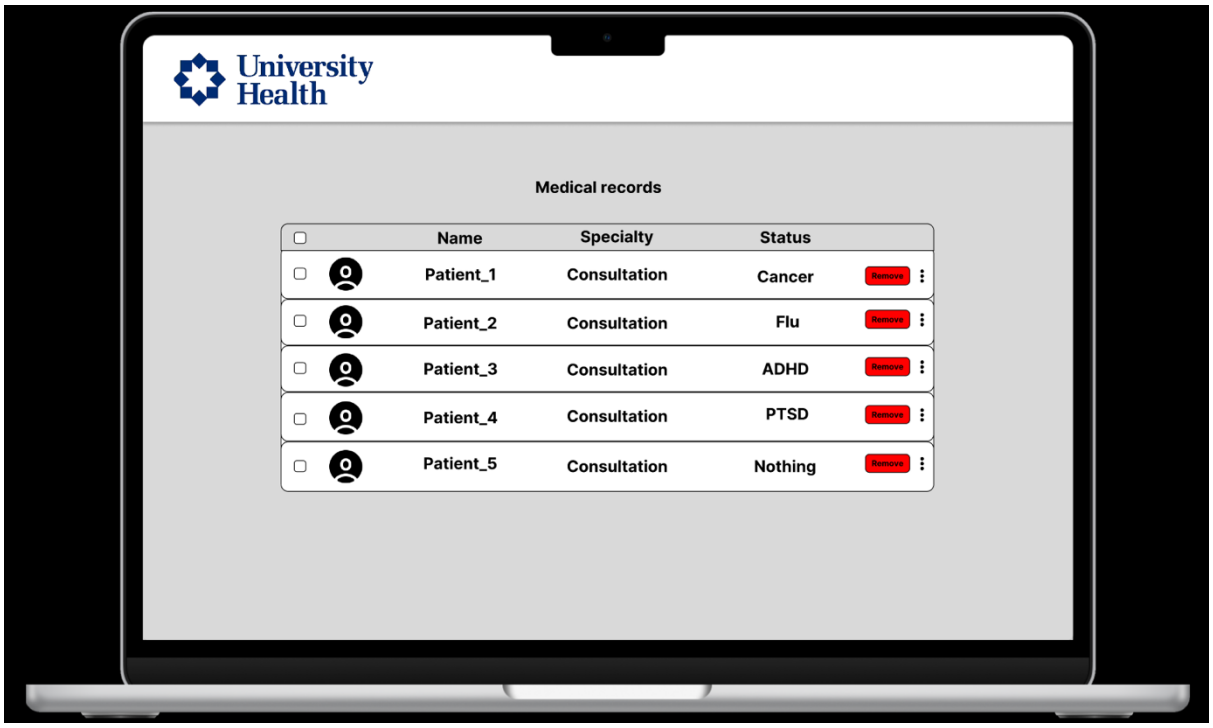
2.13.4.12 Healthcare Providers Management Page

This page displays a list of healthcare providers, including their names, ID numbers, specialties, and availability status (available, unavailable, on leave). Admin staff can add or remove providers from the system, as well as update their availability status. This interface helps manage the healthcare professionals that users can book appointments with.



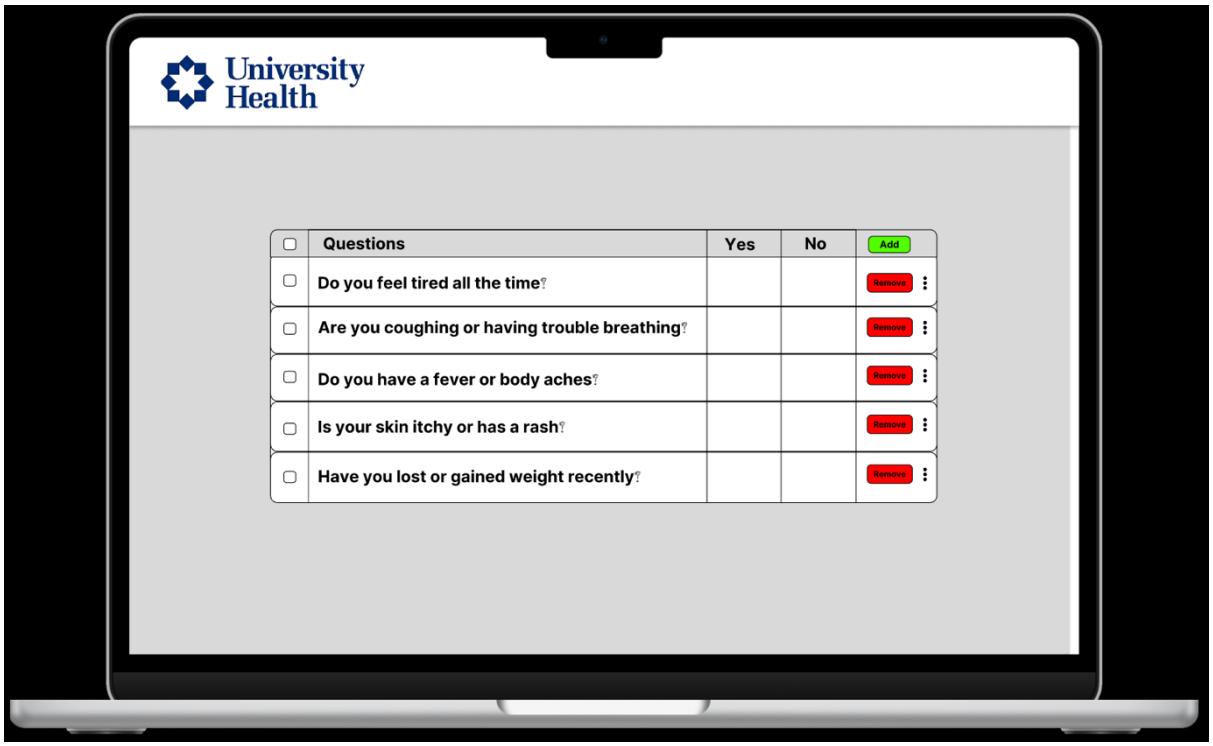
2.13.4.13 Medical Records Page

This page displays a list of patients along with their consultation details, medical specialty, and current health status. Healthcare providers can manage patient records, with options to remove patients from the list if needed.



2.13.4.14 Symptom Checker Page

This page presents a set of health-related questions for users to answer (Yes/No) as part of the symptom-checking process. The responses help the system suggest relevant healthcare actions or professionals. Healthcare providers can add or remove questions from the list.



2.13.4.15 Payments Page

This page displays a list of healthcare providers, including their names, ID numbers, specialties, and availability status (available, unavailable, on leave). Admin staff can add or remove providers from the system, as well as update their availability status. This interface helps manage the healthcare professionals that users can book appointments with.