

**MALAWI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**MALAWI INSTITUTE OF TECHNOLOGY**

**APPLIED STUDIES DEPARTMENT**

**2023/2024 SYSTEM ANALYSIS AND DESIGN**

**CODE:** SYAD 2200

**FINAL DOCUMENTATION FOR MOBILE HEALTH AWARENESS & CONSULTATION SYSTEM**

**TO:** ARTHUR MWANG’ONDA

**FROM:** GROUP 9

MUSSA MANYAMBA (BIT)

ULEMU MKWAILA (BIT)

DICKSON DZINJALAMALA (BIT)

ULLANDA MASUKU (CSS)

SYLVESTER MAWETCHA (CSS)

KHALIDWE CHIDZANJA (BIT)

LEWIS MANDALA (CSS)

Table of Contents

[Introduction 3](#_Toc181107140)

[Project Overview 4](#_Toc181107141)

[Current System Problems 5](#_Toc181107142)

[For Students: 5](#_Toc181107143)

[In the Industry: 5](#_Toc181107144)

[For the Hospital: 5](#_Toc181107145)

[Proposed New System 6](#_Toc181107146)

[System Features (Main Features) 6](#_Toc181107147)

[Extra Features (Optional if time allows) 6](#_Toc181107148)

[Non-Functional Requirements 6](#_Toc181107149)

[Tools and Technologies 7](#_Toc181107150)

[Benefits of the New System 8](#_Toc181107151)

[Plans & Development Methodology 9](#_Toc181107152)

[Database Design 10](#_Toc181107153)

[Database Schema 10](#_Toc181107154)

[Entity Relationship Diagram (ERD) 14](#_Toc181107155)

[Description of Relationships 15](#_Toc181107156)

[Use Case Diagram 16](#_Toc181107157)

[System Briefing 17](#_Toc181107158)

[Functionalities 17](#_Toc181107159)

[1. Role Based Authentication 17](#_Toc181107160)

[2. Symptom Checker 18](#_Toc181107161)

[3. Telemedicine 18](#_Toc181107162)

[4. Notifications 19](#_Toc181107163)

[5. User Management (especially for Doctors) 19](#_Toc181107164)

[6. Making appointments 20](#_Toc181107165)

[Conclusion 21](#_Toc181107166)

# Introduction

The following is the final documentation of the Mobile Health Awareness & Consultation System. It will contain designs and diagrams representing parts and processes in this system i.e. Database schema, ERD, Flow Charts, Use-Case diagram, etc.

# Project Overview

With the rapid evolution of technology, accessing reliable health information and professional consultations i.e. services, has become more efficient. The new system proposed will leverage web services to allow remote consultations and self-diagnosis without requiring physical hospital visits as some of the main functions of this system.

The system is based on **MUST Clinic** as the heath services provider. Most scenarios will be drawn from the observations made on the clinic.

# Current System Problems

For Students:

* **Inconvenient** – When a student doesn’t feel well, they have to visit the clinic which is slightly far from their hostels where once they arrive they would basically just be given paracetamol or any other pain killer depending on what problem they have, which makes it inconvenient in such a way that if that was the basic treatment needed, it would be possible to just not visit the hospital either ways especially if the problem isn’t big.
* **Time-consuming** – Students usually have much to do hence sometimes to get diagnosed you either have to wait in line or come be rescheduled for meeting multiple times in cases where the clinician/doctor is unavailable.

In the Industry:

* **Lack of innovation** – With technology advancing, it is such a waste of potential to not implement new and much more efficient ways of servicing patients i.e. minimize inconveniences as much as possible while maximizing quality treatment for the patients.

## For the Hospital:

* **Poor service delivery** – There times where students find the clinician or doctor is unavailable temporarily, which is highly inappropriate i.e. shouldn’t happen and also worse still the treatment being a few pain killers or something minimal with very shallow diagnosis compared to what one would expect.
* **Insufficient staff** – There are not many staff in the clinic making it difficult to service large groups of patients at once or on time.
* **Inefficiency and time-consuming processes** – This is something that has already been explained and emphasized in the above points.

# Proposed New System

The new system will be a web-based platform enabling users to:

* Look up symptoms and perform self-diagnosis.
* Engage in telemedicine consultations with healthcare providers.
* Enable students to make appointments prior to visiting the clinic.
* Easier patient and doctor management.
* Enable easy making of medical reports for the School administration to see and be easily briefed.

# System Features (Main Features)

* Symptom Checker: Basic diagnostic tool for common health conditions with advice and recommended treatments though with a disclaimer that it is not 100% accurate but only gives a quick service to the user.
* Telemedicine Consultations: Enables virtual meetings with healthcare professionals.
* Notifications: for appointments part of the system.
* Creation of student and Doctor user account with a role-based authentication with an additional role of admin.
* Making appointments

## Extra Features (Optional if time allows)

* + Can generate a pdf of a patient’s medical record
  + Print out reports and medical records
  + Upload files and messaging between users (Doctor and Patient)
  + Recording results of symptom checker
  + Profile Management
  + Admin ability to grant permissions for specific actions (depending on situation)

## Non-Functional Requirements

* Security: Protect user data with the latest encryption techniques.
* Scalability: Adaptable to increased usage as demand grows.
* Reliability: Consistently available and dependable.
* Online Access: Host the system and also have it on local server.
* Offline Access: For users in areas with limited internet connectivity.
* User-friendly Interface: Accessible for both rural and urban users.

Tools and Technologies

* Laptops: For system development.
* Communication Tool: WhatsApp for project communication.
* UML Tools: StarUML or Draw.io for designing system architecture.
* IDE: Visual Studio Code for writing and testing the code.
* Database Management Tools: MySQL Workbench or XAMPP for database management.
* Languages: HTML, CSS, JavaScript, PHP, and SQL.

# Benefits of the New System

* Innovative: Utilizes cutting-edge technologies for healthcare.
* Efficient: Saves time for healthcare professionals and patients alike.
* Cost-saving: Reduces travel and hospital expenses for patients.

# Plans & Development Methodology

The Methodology for developing this system will most likely be using Agile Methodology. This will help to be quicker and more function focused i.e. error handling is much better with this methodology.

The following is a Gantt chart showing how long the period for covering each task will take:

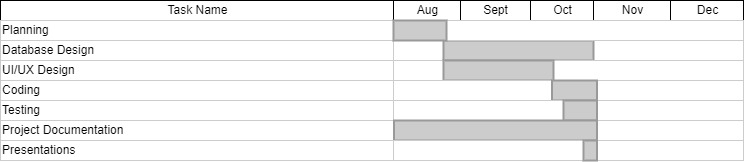


Figure 1:The diagram shows a gantt chart for this systems development period

# Database Design

This systems database is made out of currently 11 tables, with others being removed due to time that it would take to implement the functionalities that come with them.

## Database Schema

The following is the database schema for MHAC system:

1. Admins Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| admin\_id | int(11) | YES | NO |  |
| name | varchar(100) |  | NO |  |
| email | varchar(100) |  | NO |  |
| password | varchar(255) |  | NO |  |
| role\_id | int(11) |  | NO |  |
| created\_at | timestamp |  | NO | current\_timestamp() |

1. Appointments Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| appointment\_id | int(11) | YES | NO |  |
| student\_id | int(11) |  | NO |  |
| doctor\_id | int(11) |  | NO |  |
| appointment\_date | datetime |  | NO |  |
| status | enum('Pending','Confirmed','Completed','Cancelled') |  | YES | 'Pending' |
| created\_at | timestamp |  | NO | current\_timestamp() |

1. Conditions Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| condition\_id | int(11) | YES | NO |  |
| name | varchar(100) |  | NO |  |
| description | text |  | YES | NULL |
| health\_tips | text |  | YES | NULL |

1. Condition Symptoms Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| condition\_id | int(11) | YES | NO |  |
| symptom\_id | int(11) | YES | NO |  |
| severity\_weight | int(11) |  | YES | 1 |

1. Doctors Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| doctor\_id | int(11) | YES | NO |  |
| name | varchar(100) |  | NO |  |
| specialization | varchar(100) |  | NO |  |
| bio | text |  | YES | NULL |
| phone\_number | varchar(20) |  | YES | NULL |
| profile\_picture | varchar(255) |  | YES | NULL |
| years\_of\_experience | int(11) |  | YES | NULL |
| email | varchar(30) |  | YES | NULL |

1. Medical Records Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| record\_id | int(11) | YES | NO |  |
| student\_id | int(11) |  | NO |  |
| weight | decimal(5,2) |  | YES | NULL |
| height | decimal(5,2) |  | YES | NULL |
| blood\_group | varchar(5) |  | YES | NULL |
| chronic\_conditions | text |  | YES | NULL |
| allergies | text |  | YES | NULL |
| created\_at | timestamp |  | NO | current\_timestamp() |

1. Notifications Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| notification\_id | int(11) | YES | NO |  |
| user\_id | int(11) |  | NO |  |
| message | text |  | NO |  |
| type | enum('Appointment','Prescription','Message','System') |  | NO |  |
| sent\_at | timestamp |  | NO | current\_timestamp() |
| is\_read | tinyint(1) |  | YES | 0 |

1. Roles Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| role\_id | int(11) | YES | NO |  |
| role\_name | enum('Student','Doctor','Admin') |  | NO |  |

1. Students Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| student\_id | int(11) | YES | NO |  |
| reg\_number | varchar(20) |  | NO |  |
| email | varchar(100) |  | NO |  |
| name | varchar(100) |  | NO |  |
| program\_of\_study | varchar(50) |  | NO |  |
| year\_of\_admission | int(11) |  | NO |  |
| expected\_graduation\_year | int(11) |  | NO |  |
| status | enum('Active','Graduated','Withdrawn') |  | NO |  |
| date\_of\_birth | date |  | NO |  |
| gender | enum('Male','Female','Other') |  | NO |  |
| phone\_number | varchar(20) |  | YES | NULL |
| profile\_picture | varchar(255) |  | YES | NULL |

1. symptoms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| symptom\_id | int(11) | YES | NO |  |
| name | varchar(100) |  | NO |  |
| description | text |  | YES | NULL |

1. Users Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **PK** | **Nullable** | **Default** |
| user\_id | int(11) | YES | NO |  |
| student\_id | int(11) |  | YES | NULL |
| doctor\_id | int(11) |  | YES | NULL |
| email | varchar(100) |  | NO |  |
| password | varchar(255) |  | NO |  |
| role\_id | int(11) |  | NO |  |
| created\_at | timestamp |  | NO | current\_timestamp() |

## Entity Relationship Diagram (ERD)

Using the current database schema, the following is the entity relationship diagram mainly showing the relationships between the entities we have for this system:

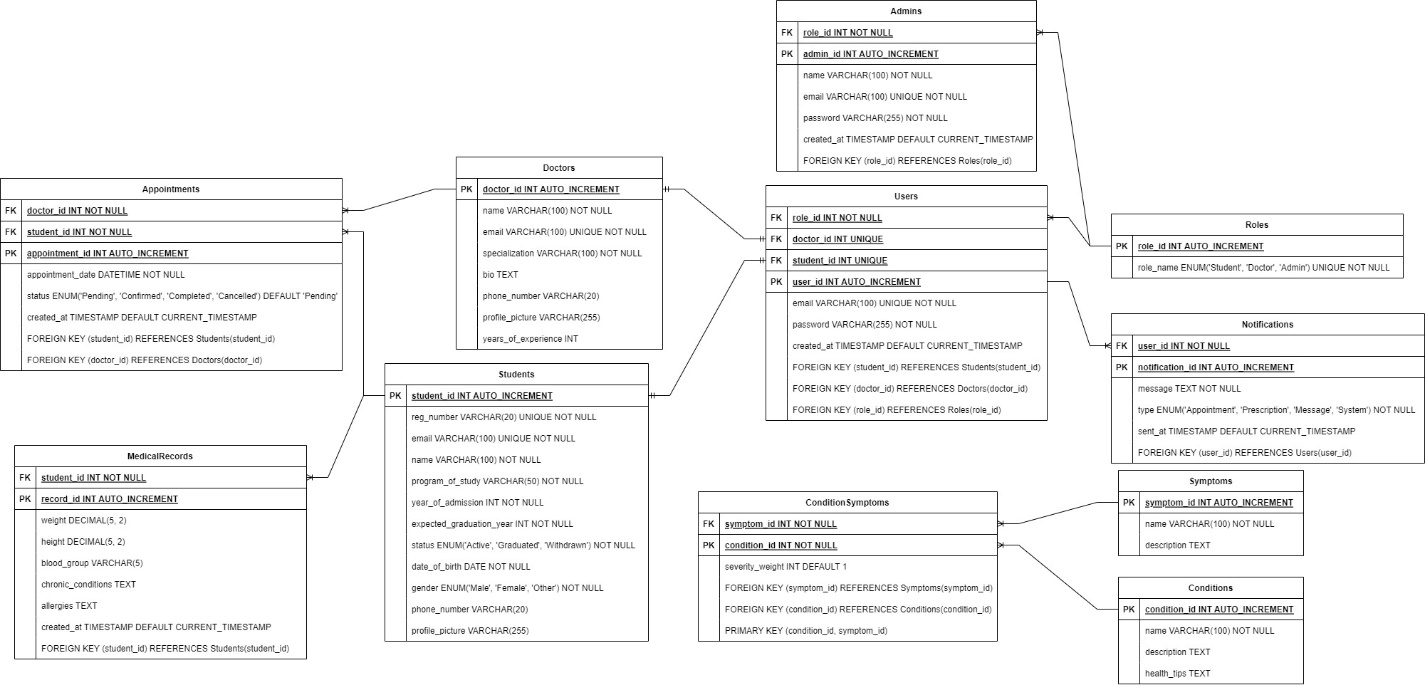


Figure 2: The following is the ERD for the MHAC- database

### Description of Relationships

For the above relationships to make more sense, the following is a quick description of how they are used:

1. **Conditions and Symptoms via ConditionSymptoms**:

* When a user enters symptoms, the system references ConditionSymptoms to assess potential conditions based on severity. This approach supports differential diagnosis by weighting symptoms more heavily that are crucial to a specific condition.

1. **Students and MedicalRecords**:

* When a student logs in, the system pulls data from MedicalRecords to display the student’s personal health details (e.g., weight, height, chronic conditions) when they are on profile page. This relationship also helps doctors quickly access and review medical details relevant to diagnosis and treatment.

1. **Appointments linking Students and Doctors**:

* Through the Appointments table, students can schedule the virtual consultations. Doctors can review and accept appointment requests, while students receive update notification, hence efficient communication.

1. **Notifications for Appointments and Prescriptions**:

* Notifications ensure users (students or doctors) receive timely alerts, such as appointment confirmations.

1. **Users, Roles, and Admins**:

* The Users and Roles structure allows for role-based access control, where admin users manage the system’s Doctors, doctors handle appointments, and students interact with the symptom checker and make appointments.

## Use Case Diagram

A use case diagram shows the interactions between the users and the system. The following will be the use-case diagram for MHAC system:

// Make a use-case guys…

# System Briefing

The following will be a breakdown of how the system is working, i.e. the functions it currently is able to do.

This may include flow charts to explain the processes better.

## Functionalities

### Role Based Authentication

This Function is where as a user is authenticated based on their role. The system first assumes that a student is in the student database (the students table in the DB represents that). It first checks if student is there, when you are registering that is, since if you are there, it means that you have a student email already etc., all your credentials as a student hence in this system you just have to verify email and create a password to be added to the user table hence become a user of the system with role assigned as student. The following is the flow chart to give you a picture of what has just been explained:

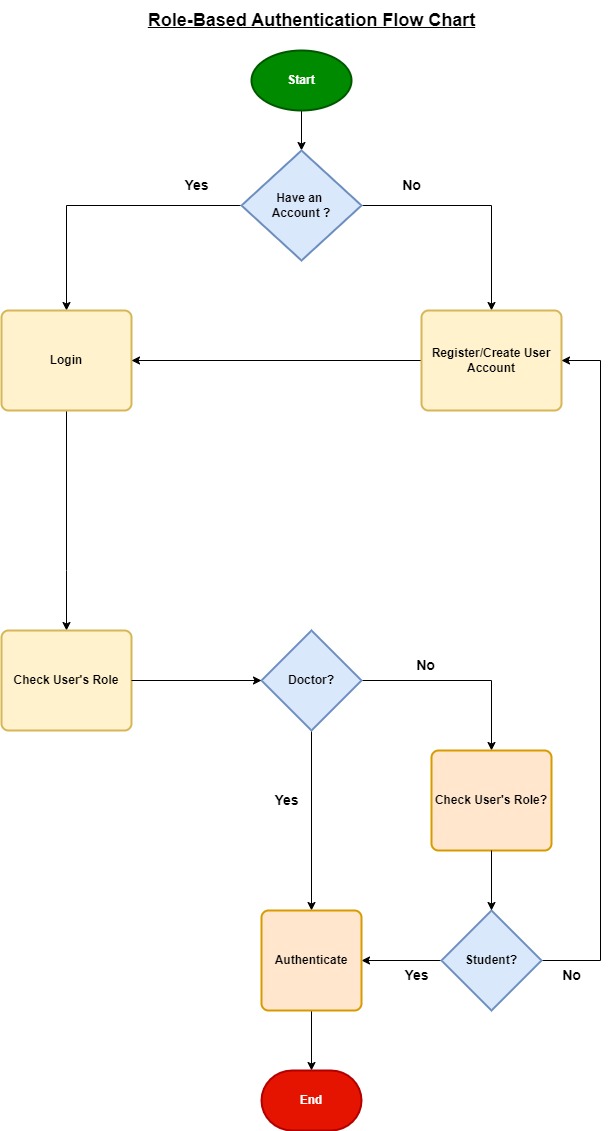


Figure 3: Showing Role based authentication flow chart

### Symptom Checker

This feature allows the processing of given symptoms, and giving the most likely condition associated with them. It also recommends treatment but does not guarantee 100% accurate conclusions, but can at least give suggestions for how to proceed to the patient. The following is a flow chart of how it works:

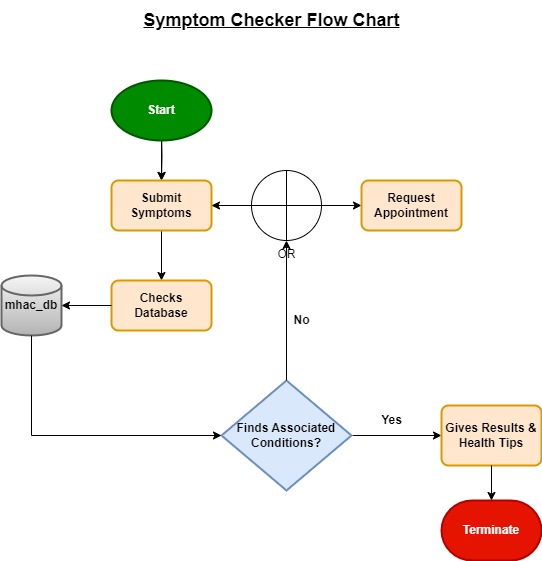


Figure 4: Shows Symptom Checkers basic logical process

### Telemedicine

This feature allows for the completion of the appointment process which is having a virtual interaction with the doctor via video conferencing where u join a chatroom initiated by the doctor.

### Notifications

This is a mechanism set to alert doctors and patients/students alike within the appointment mechanism. When a student requests an appointment, a doctor gets notification they are able to see on their dashboard. When the request is accepted, student will also get a popup notification. The following is the flow hart for this mechanism:

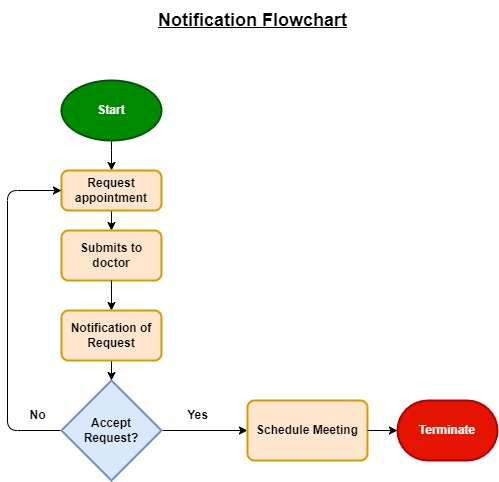


Figure 5: Shows notification process

### User Management (especially for Doctors)

This basically allows the creation of a doctor account which is then finalized by having to register their desired password then added to user list while being allocated the doctor role. It also enables admin to update and delete doctor information.

Besides this, this mechanism has also encompassed patient record monitoring, i.e. doctor can view a student patients medical record and view their basic student details like gender, program of study, year of admission, etc. It also has a filtering (by gender, status, etc.) and a search feature where u can search users by name.

### Making appointments

A student after registering is able to request an appointment. They send a request to the doctors available, and the doctor is notified of the request and can accept and then schedule it or leave it on pending, or even cancel. When it is accepted, a notification with the join button is made hence a virtual meeting can then be held upon clicking join.

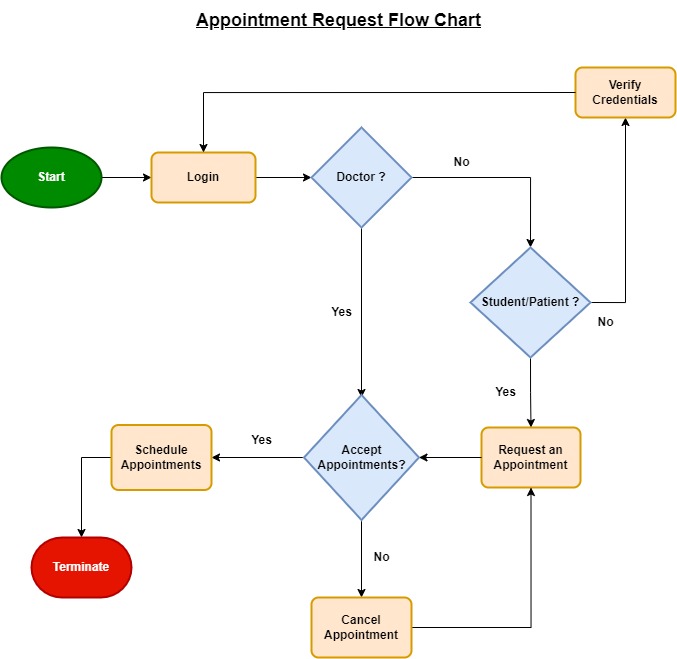


Figure 6: Shows flow chart of appointment request i.e. booking process

# Conclusion

The Mobile Health Awareness & Consultation System for MUST Clinic aims to improve healthcare at the institute and has a possibility to be scaled even further. By providing timely health advice and virtual consultations, it reduces the need for physical hospital visits, especially for those who are not able to go to the hospital, thereby increasing public health awareness and empowering users to make informed decisions and avoid wasted time visiting the hospital/clinic.