



**NATIONAL TEACHERS COLLEGE**

**Process Management and Foundation Day**  
**Connection**

**Submitted By:**

Olea, Simon Anakin  
Sanchez, Rafael V.  
Sildo, Vhone Gabrielle M.  
Tumlos, Marth Dominique P.  
Trigusa, Sam Lawrence A.

**Submitted To:**

Justin Louise R. Neypes

# Barangay Health Management System

## I. Introduction

### 1.1 Project Overview & UN SDG Target

The Barangay Clinic Appointment Management System is a MySQL-based database application designed to efficiently manage patient records, doctor schedules, healthcare services, and appointment bookings. The system supports organized data storage, reliable transaction handling, and accurate reporting for clinic operations. This project aligns with the United Nations Sustainable Development Goal (SDG) 3: Good Health and Well-Being, by improving access to primary healthcare services through better data management and scheduling efficiency at the barangay level.

### 1.2 Problem Statement

Many barangay clinics rely on manual record-keeping or unstructured data systems, which often result in scheduling conflicts, duplicate records, and inaccurate patient information. These issues lead to inefficient clinic operations, longer waiting times, and increased risk of data errors. The system addresses this real-world data problem by providing a centralized MySQL database that enforces data integrity, prevents invalid appointments, and ensures accurate, consistent, and timely management of clinic information.

- **Double Booking:** Patients scheduled for the same doctor at the same time.
- **Schedule Conflicts:** Appointments made when doctors are unavailable.
- **Data Redundancy:** Patient details repeated across multiple logbooks.
- **Reporting Delays:** Difficulty in generating summary reports for local government requirements.

## II. Requirements & Analysis

### 2.1 Functional & Non-Functional Requirements

#### Functional Requirements (FR)

ID	Requirement	Alignment
FR1	The system successfully loads and stores input data into a fully designed relational database using MySQL. This is achieved through properly structured tables such as patients, doctors, services, appointments, and health_records, all defined using DDL.	<b>DDL/DML:</b>
FR2	DBMS concepts using MySQL, including referential integrity through foreign keys, data validation using constraints and triggers, and controlled data operations through stored procedures. These ensure data consistency, accuracy, and reliability.	<b>Core Concepts</b>
FR3	The system performs transactional operations using a MySQL stored procedure that enforces ACID properties. Appointment bookings are executed atomically and validated through triggers, ensuring consistency, isolation from conflicts such as double booking, and durability of committed data.	<b>Finals Concept:</b>
FR4	The system generates reports using MySQL views and DQL (SELECT) statements, allowing efficient retrieval and presentation of summarized data such as elderly appointments and service-based statistics.	<b>DQL/Reporting</b>

### Non-Functional Requirements (NFR)

ID	Requirement	Metric
NFR1	<b>Robustness:</b> The system handles invalid inputs and operations gracefully without violating database integrity. This is enforced through MySQL triggers that validate appointment dates, doctor schedules, service compatibility, and duplicate bookings. Informative error messages are returned when violations occur.	Error messages are clear and informative
NFR2	<b>Maintainability:</b> The database schema and stored logic are modular, well-structured, and follow SQL best practices. The use of views, triggers, and stored procedures improves readability and simplifies future updates or modifications to the system.	Use of comments and logical VIEW definitions.
NFR3	<b>Performance:</b> The system ensures efficient execution of required reports by using indexed columns, optimized joins, and pre-defined views. This allows report queries to return results quickly, even as data volume increases.	Reports should return results in under 1 second on standard hardware with test data.

## 2.2 Data Requirements

The system is pre-loaded with over 50 records, including:

- **Patients:** 53 registered residents with contact info and addresses.
- **Doctors:** 3 specialists (General checkup, Obstetrician, Pediatrician).
- **Services:** 3 core services (Checkup, Prenatal, Pediatric).

## 2.3 Schema Normalization Analysis

The database is designed in **Third Normal Form (3NF)**:

- **1NF:** All columns contain atomic values.
- **2NF:** All non-key attributes are fully dependent on the primary key (e.g., Doctor specialization depends only on doctor\_id).
- **3NF:** Transitive dependencies are removed. For instance, appointment details reference patient\_id rather than duplicating patient names, ensuring that updating a patient's address in the patients table reflects everywhere.

## III. Design Specification

### 3.1 Core DBMS Concepts Used

#### Concept 1: Stored Procedure

The screenshot displays a MySQL query editor window. At the top, a text area contains the SQL query: `CALL book_appointment(1, 1, 1, '2025-12-23', '10:00:00', 'body checkup');`. Below the text area are three buttons: "Clear", "Format", and "Get auto-saved query". Underneath these buttons is a checkbox labeled "Bind parameters" with a help icon. Below the checkbox is a text input field with the placeholder "Bookmark this SQL query:". At the bottom of the editor, there is a row of controls including a "Delimiter" dropdown set to ";", and three checkboxes: "Show this query here again", "Retain query box", and "Rollback v". Below these controls is a "Hide query box" button. At the very bottom, a green status bar displays a checkmark icon and the message: "MySQL returned an empty result set (i.e. zero rows). (Query took 0.0217 seconds.)". Below the status bar, the executed query is shown again: `CALL book_appointment(1, 1, 1, '2025-12-23', '10:00:00', 'body checkup');`.


## Concept 2: Triggers

### Error

---

SQL query: [Copy](#)

```
INSERT INTO appointments
(patient_id, doctor_id, service_id, date, time, reason)
VALUES
(21, 1, 1, '2025-01-01', '10:00:00', 'test for trigger');
```

MySQL said: 


#1644 - Appointment date cannot be in the past.

### Error

---

SQL query: [Copy](#)

```
INSERT INTO appointments
(patient_id, doctor_id, service_id, date, time, reason)
VALUES
(21, 1, 2, '2025-12-24', '10:00:00', 'Wrong service test');
```


MySQL said: 

#1644 - GP can only do General Checkup.

## Error

SQL query: [Copy](#)

```
INSERT INTO appointments
(patient_id, doctor_id, service_id, date, time, reason)
VALUES
(21, 1, 1, '2025-12-15', '7:00:00', ' date test');
```

MySQL said: 


#1644 - GP available Mon-Fri, 9AM-5PM only.

(Concept 3: Foreign Key (Referential Integrity))

## Error

SQL query: [Copy](#)

```
INSERT INTO appointments (patient_id, doctor_id, service_id, date, time, reason)
VALUES (999, 1, 1, '2025-12-24', '10:00:00', 'Test foreign key');
```

MySQL said: 

#1644 - Patient does not exist.

#### (Concept 4: View)

✓ Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT * FROM elderly_appointments;
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows:  Filter rows:

Extra options

				name	age	date	status	specialization
<input type="checkbox"/>	Edit	Copy	Delete	Isabel Flores	60	2025-12-15	Completed	General Practitioner
<input type="checkbox"/>	Edit	Copy	Delete	Pedro Reyes	65	2025-12-16	Completed	General Practitioner

#### (Concept 5: constraints)

### Error

SQL query: [Copy](#)

```
INSERT INTO patients (name, age, address, contact_no)
VALUES ('Test', 30, 'Barangay 1', '09123456789');
```

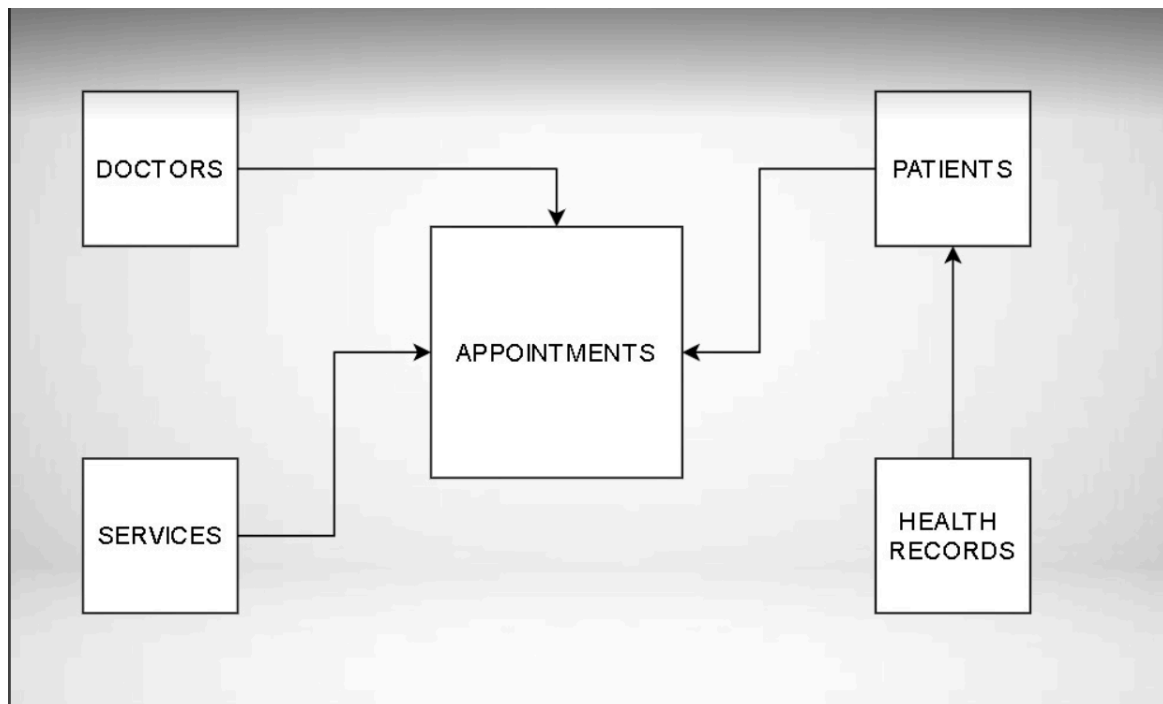
MySQL said:

#1062 - Duplicate entry '09123456789' for key 'contact\_no'

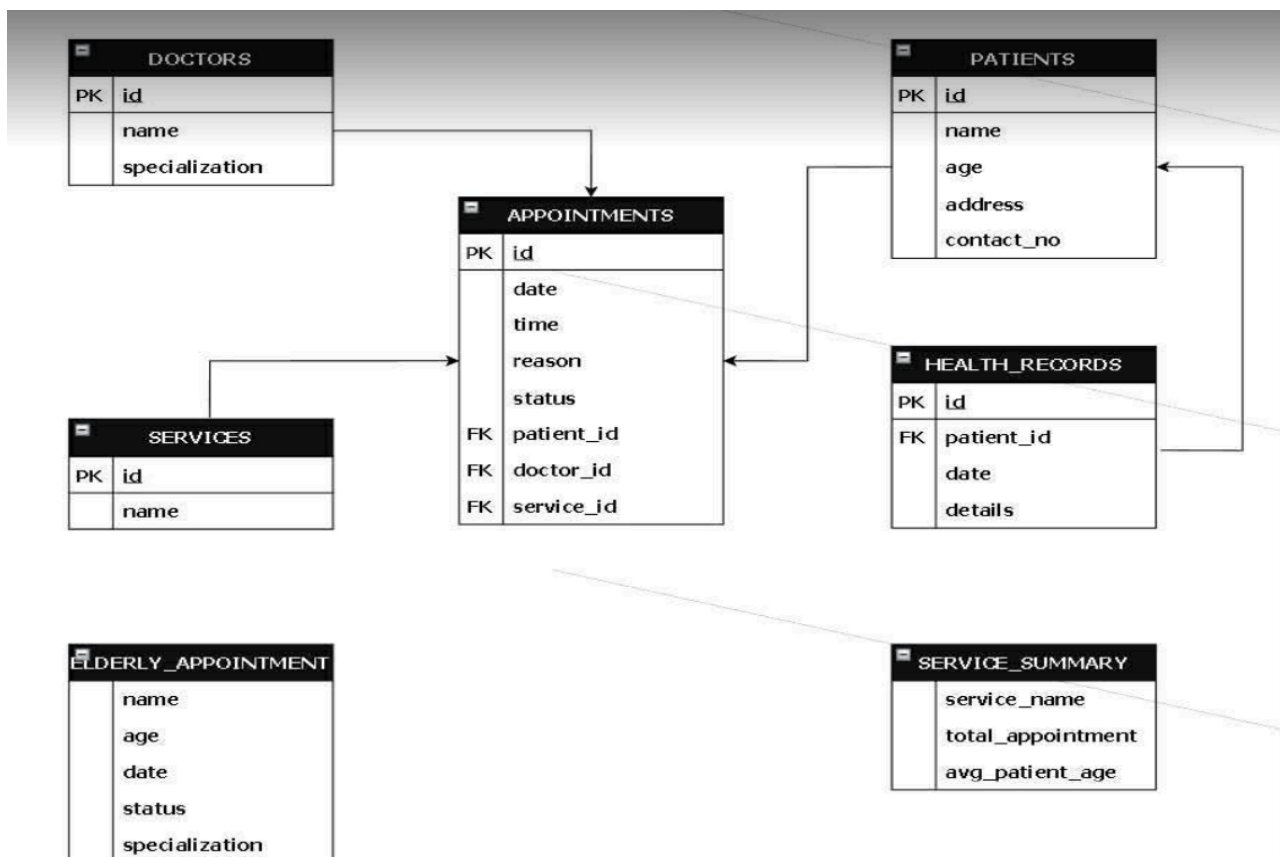


### 3.2 ER Diagram Guidance (For your visual ERD)

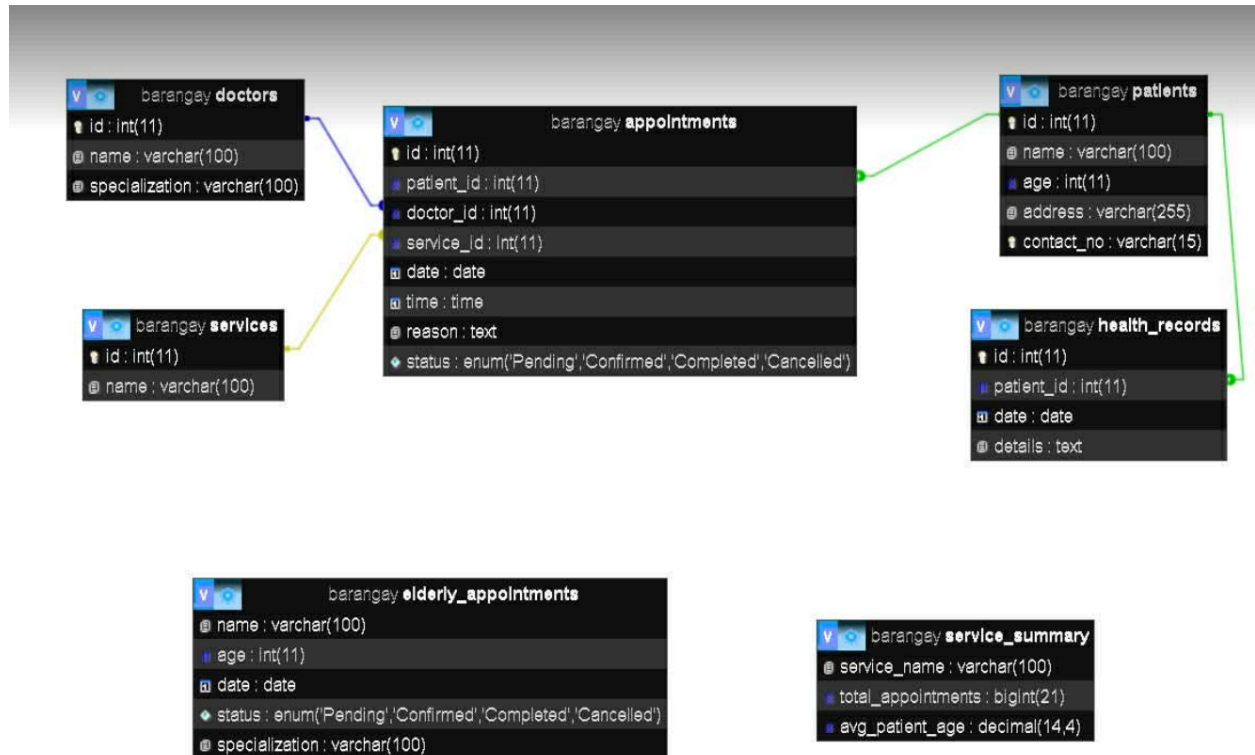
#### Conceptual erd



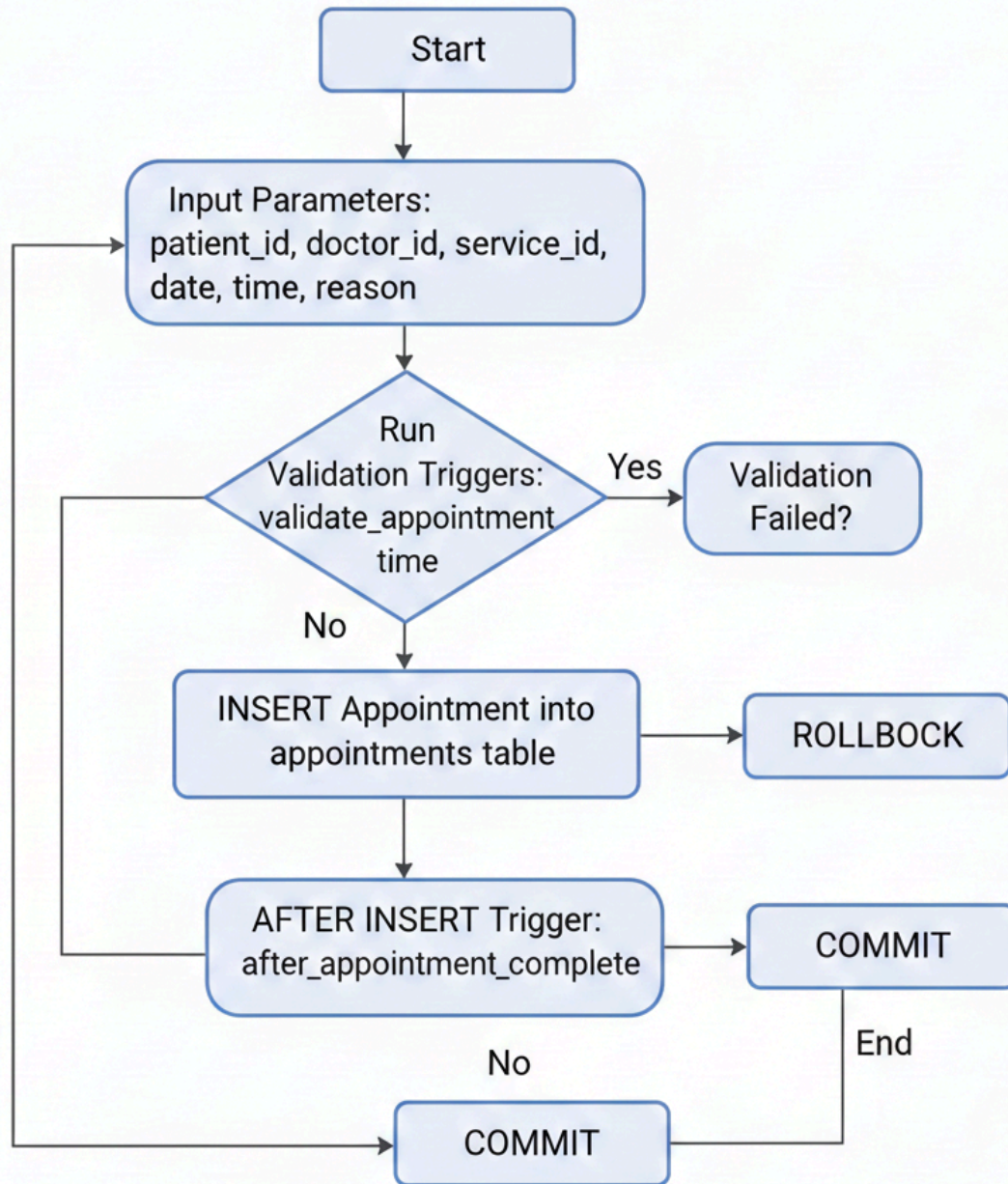
#### Logical erd



## Physical erd



### 3.3 Transaction Flowchart Guidance (For `book_appointment`)



## V. Conclusion

The Barangay Clinic Appointment Management System successfully applies core DBMS concepts using MySQL. The system efficiently manages patient, doctor, service, and appointment data through a well-designed relational schema. Stored procedures and triggers ensure secure and consistent transactions by enforcing business rules and maintaining data integrity. Foreign keys, constraints, and views further support data accuracy and simplify reporting. Overall, the system demonstrates a reliable, organized, and practical database solution suitable for clinic operations.

## Contributions

SAM LAWRENCE A. TRIGUSA	DDL (PRE LIM)
RAFAEL V. SANCHEZ	ERD (MIDTERM)
SIMON ANAKIN OLEA	JOINTS(MIDTERM)
MARTH DOMINIQUE P. TUMLOS	STORED PROCEDURE (FINALS)
VHONE GABRIELLE M. SILDO	TRIGGER(FINALS)