NATIONAL TEACHERS COLLEGE

Barangay Health Management System

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

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.	**DDL/DML:**
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.	**Core Concepts**
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.	**Finals Concept:**
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.	**DQL/Reporting**

Non-Functional Requirements (NFR)

ID	Requirement	Metric
NFR1	**Robustness:** 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	**Maintainability:** 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	**Performance:** 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.

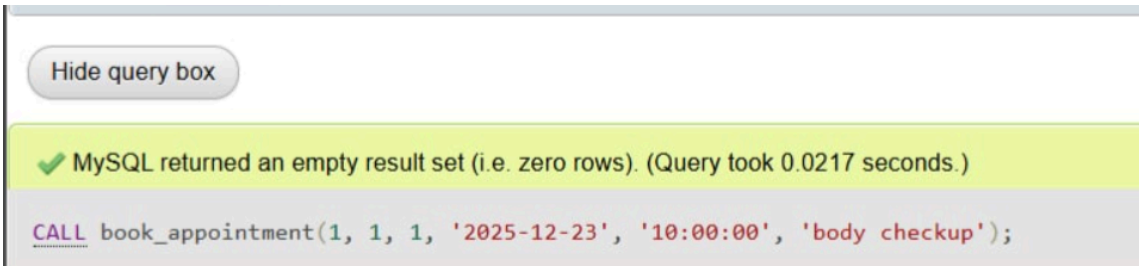
III. Design Specification

3.1 Core DBMS Concepts Used

Concept 1: Stored Procedure

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

```
1 CALL book_appointment(1, 1, 1, '2025-12-17', '09:00:00', 'General Checkup');
2
```



Concept 2: Triggers

A trigger is an automatic action that runs when something changes in the database.

When an event happens, such as adding a new appointment, the trigger automatically checks important rules. It ensures the doctor is available and prevents double booking. If a rule is violated, the trigger stops the action, protecting the system from invalid data.\

Screen shot

```
1 CALL book_appointment(21, 1, 1, '2025-12-15', '07:00:00', 'trigger test');
2
```

MySQL said:

#1644 - Appointment date cannot be in the past.

example 2

```
1 CALL book_appointment(21, 1, 1, '2025-12-20', '07:00:00', 'Date test');
2
```

MySQL said:

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

Example 3

```
CALL book_appointment(21, 1, 3, '2025-12-20', '07:00:00', 'wrong service');
```

MySQL said:

```
#1644 - GP can only do General Checkup.
```

(Concept 3: Foreign Key (Referential Integrity)

A foreign key is a database rule that links data from one table to another.

Foreign keys ensure that related data exists and is valid. For example, an appointment must reference an existing patient and doctor. This prevents invalid or orphan records in the database.

Example Query:

```
INSERT INTO appointments
(patient_id, doctor_id, service_id, date, time, reason)
VALUES
(54, 1, 1, '2025-12-19', '10:00:00', 'test');
```

MySQL said:

```
#1644 - Patient does not exist.
```

(Concept 4: View)

A view is a virtual table created from a query that displays selected data.

Views are used to simplify reporting and data viewing. They present useful information without exposing all the details from the database tables, making reports easier to understand and safer to access.

Example Query:

```
1 SELECT * FROM elderly_appointments;
```

		name	age	date	status	specialization
☐	Edit Copy Delete	Isabel Flores	60	2025-12-15	Completed	General Practitioner
☐	Edit Copy Delete	Pedro Reyes	65	2025-12-16	Completed	General Practitioner

(Concept 5: constraints)

Constraints are rules that restrict what kind of data can be stored in the database.

Constraints prevent invalid data such as missing required fields or duplicate entries. They help maintain data accuracy and consistency throughout the system.

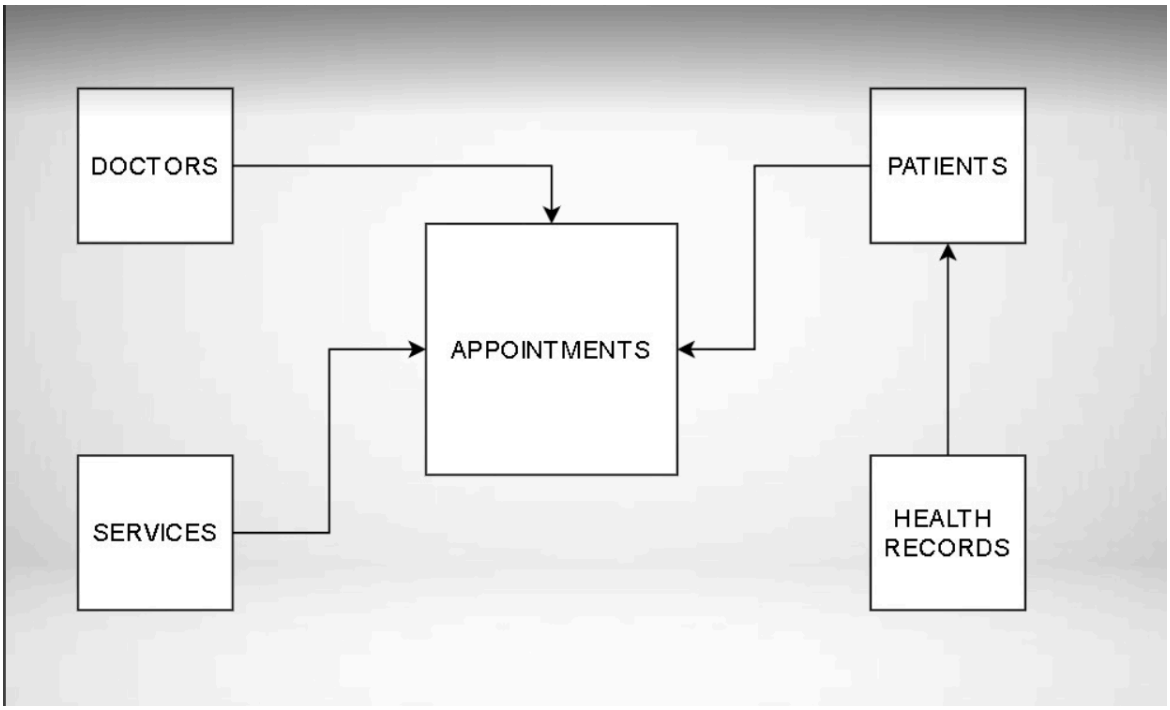
```
1 INSERT INTO patients
2 (name, age, address, contact_no) VALUES
3 ("test", "45", "Barangay 1", "09123456789");
```

MySQL said:

```
#1062 - Duplicate entry '09123456789' for key 'contact_no'
```

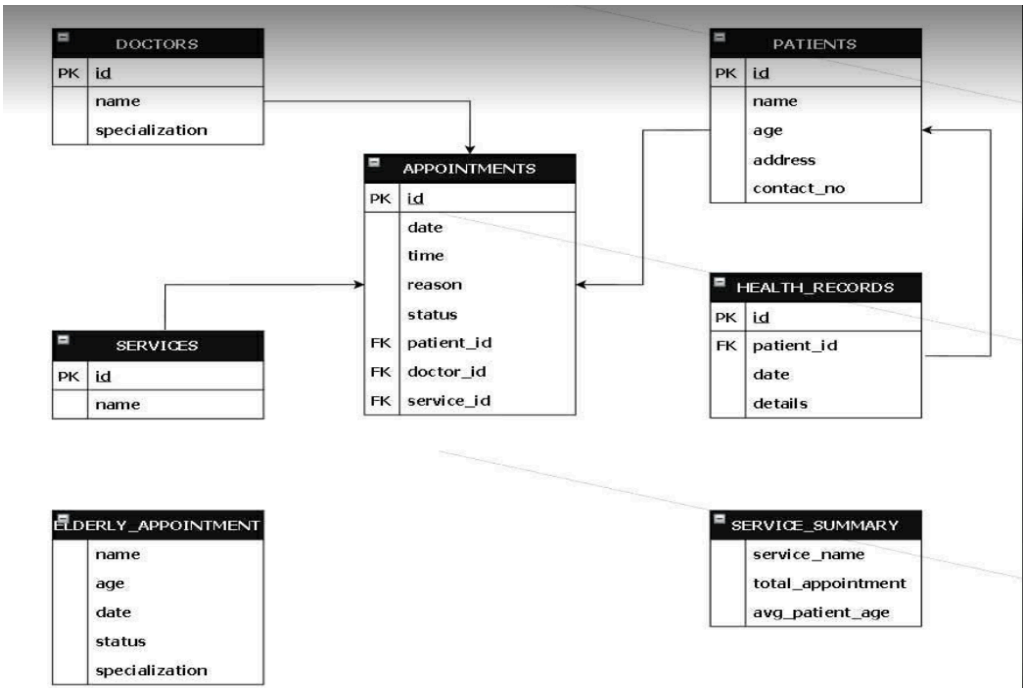
3.2 ER Diagram Guidance (For your visual ERD)

Conceptual erdConceptual ERD presents a high-level view of the system by identifying the main entities, such as patients, doctors, services, and appointments, and how they are related. It focuses on understanding the overall structure of the clinic system without technical details.



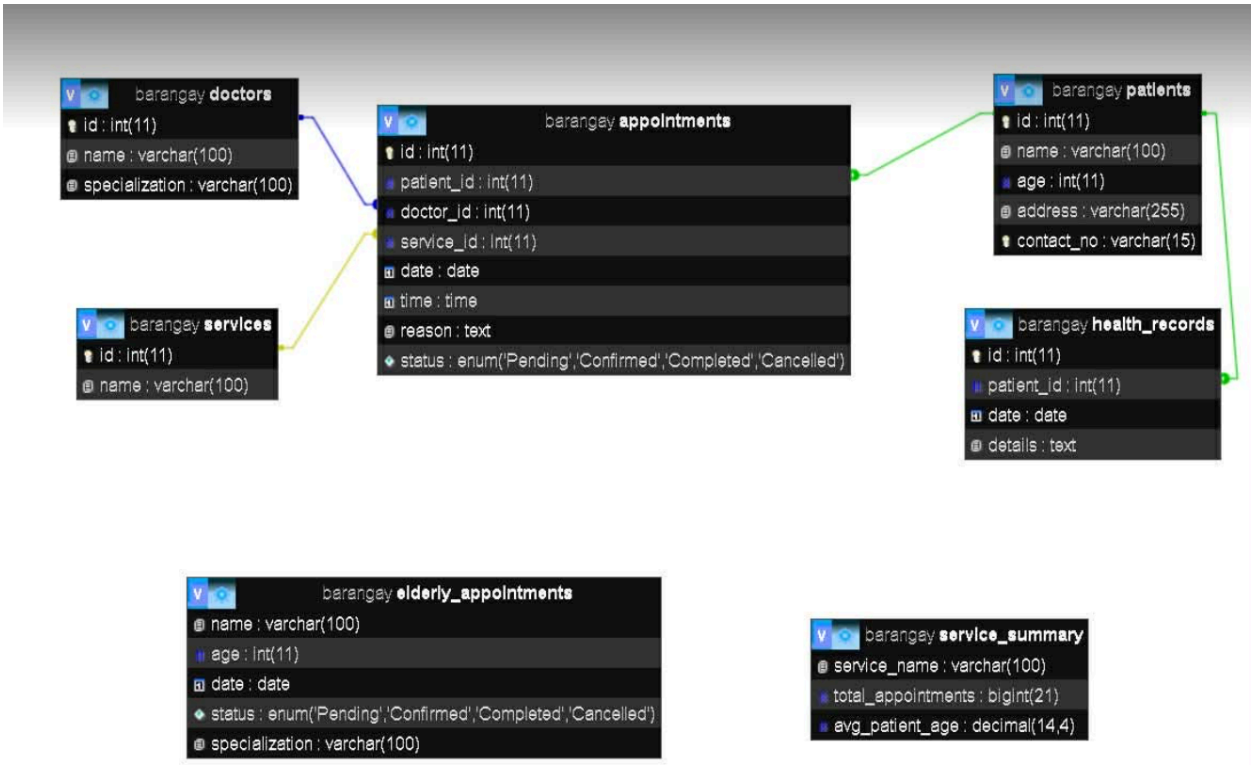
Logical erd

Logical ERD adds more detail to the conceptual design by defining the attributes of each entity and their relationships. It shows how data is logically organized and connected, but it remains independent of any specific database system.



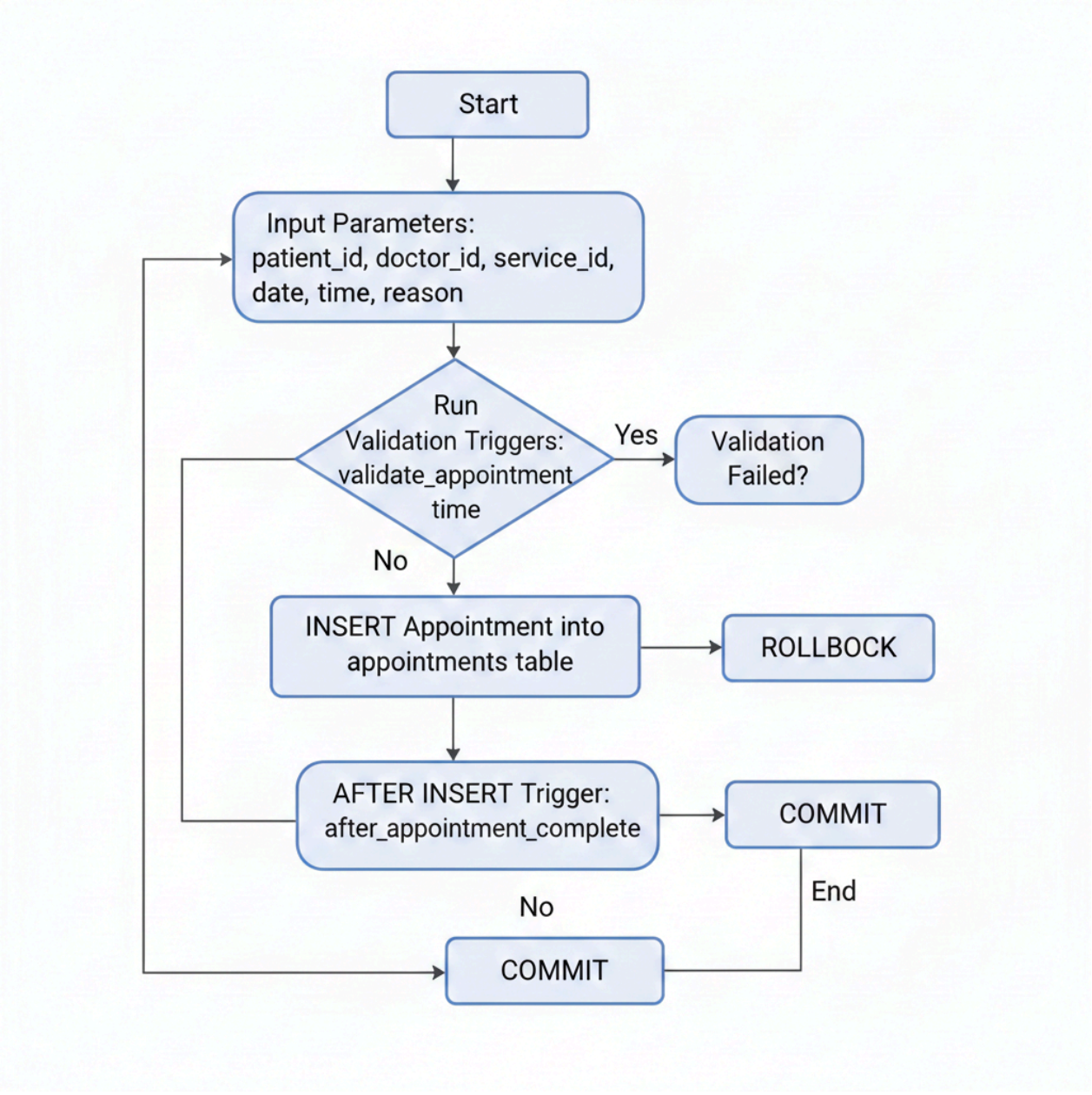
Physical erd

The Physical ERD shows how the database is actually built and implemented in MySQL. It includes the exact tables, column names, data types, primary keys, foreign keys, and constraints used in the system. This ERD ensures that data is stored correctly and that relationships between tables are enforced, such as linking appointments to valid patients, doctors, and services. It also helps prevent errors like duplicate records and invalid bookings by applying rules directly at the database level. Overall, the Physical ERD serves as the blueprint for creating a reliable, accurate, and efficient clinic database.



3.3 Transaction Flowchart Guidance (For book_appointment)

The flowchart shows how the system safely books an appointment in the barangay clinic. The process starts when the user provides the required details, such as the patient, doctor, service, date, time, and reason for the visit. After receiving the information, the system automatically checks if the appointment time is valid and if there are no schedule conflicts. If the validation fails, the system cancels the process and does not save anything. If the validation passes, the appointment is temporarily added to the system. Afterward, another automatic check confirms that the appointment was successfully recorded. If everything is correct, the system saves the appointment permanently and ends the process. This flow ensures that only valid and conflict-free appointments are recorded, preventing errors like double booking.



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)