Remote Tech Support Service System

1. Introduction

This project models a Remote Tech Support Service system using database design principles. The purpose of the system is to manage customers, their devices, service tickets, technicians, and service logs. By creating a structured relational database, the system ensures that technical support requests are tracked efficiently, and all data is properly maintained.

2. Requirements Gathering Process

2.1 Methods Used

To gather system requirements, I used:

- **Research:** Looked at real-world IT support platforms such as Best Buy, Geek Squad, and Apple Support.
- **Interviews:** Conducted a short interview with a friend who works as a freelance IT support technician.
- **Brainstorming:** Considered common service needs like password resets, device troubleshooting, hardware replacements, and software installations.

2.2 Interview Notes

Interviewee: “Ali Bakayoko” – Freelance IT Support Technician.

- Biggest challenge: customers do not provide enough details about their devices.
- Needs: a clear way to track customer devices by serial numbers.
- Tickets must show the description of the issue, the assigned technician, and the resolution.
- Service logs should be kept for each ticket to avoid repeated issues.

3. System Requirements

Functional Requirements

1. The system must store **customer information** (name, email, phone, address).
2. Each customer can register **multiple devices** (device type, brand, model, serial number).
3. **Technicians** must be tracked (name, email, specialization, phone).
4. Customers submit **tickets** describing issues with devices.
5. Tickets are assigned to a **technician**.
6. Each ticket must have a **status** (Open, In Progress, Closed).
7. **Service logs** record the actions taken to resolve tickets.

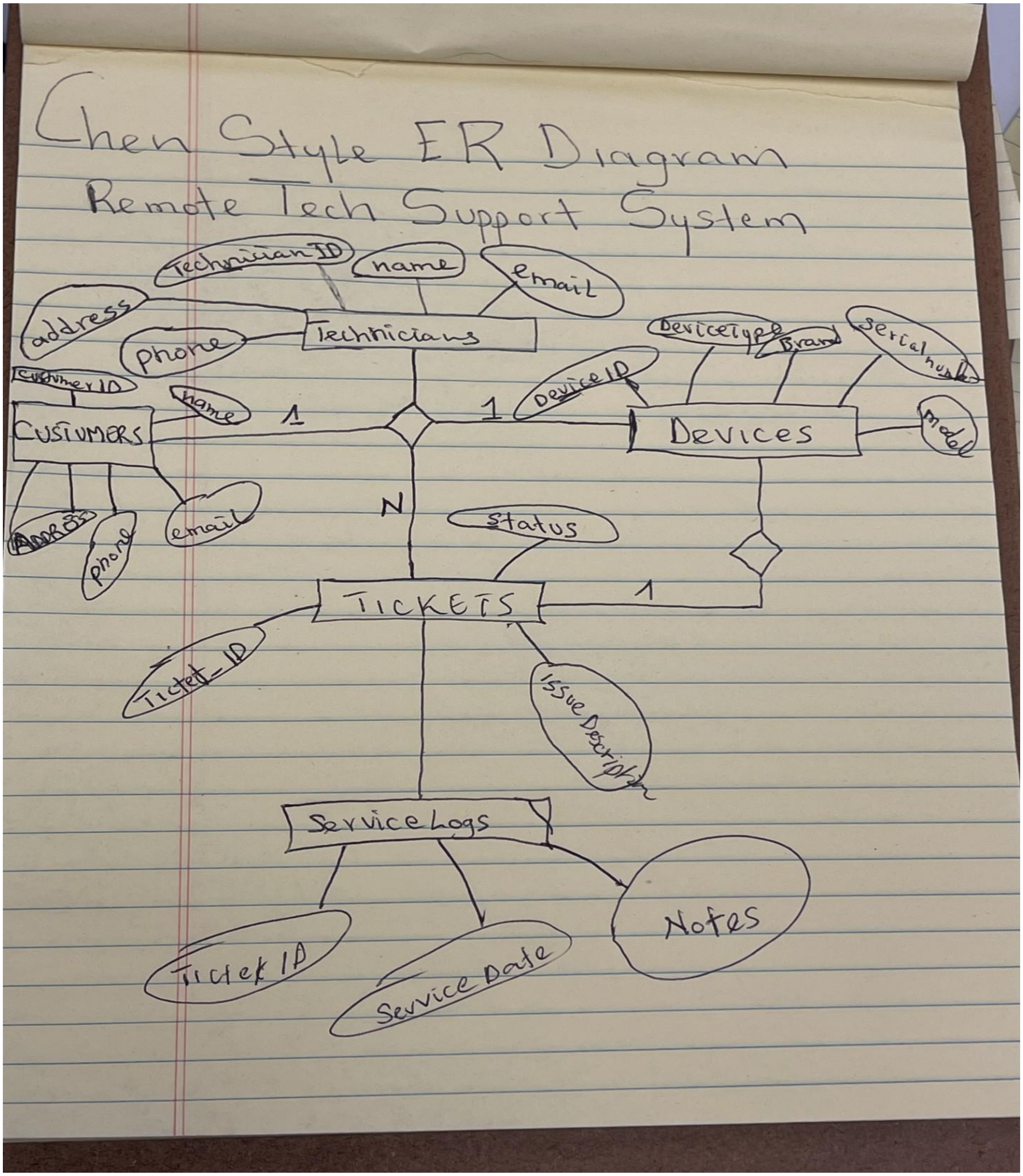
Non-Functional Requirements

- The database must enforce **data integrity** (unique emails, unique device serial numbers).
- Must support **foreign keys** for relationships.
- Must allow **cascading deletes** (deleting a customer should remove related devices and tickets).

4. Entity-Relationship (ER) Diagrams

4.1 Chen-Style ER Diagram (Hand-Drawn)

View the Image below ！



5. Database Design and Implementation

5.1 Database Creation (SQL) – View my SQL Workbench

The system includes 5 main tables:

- **Customers**
- **Technicians**
- **Devices**
- **Tickets**
- **Service Logs**

7. Conclusion

The Remote Tech Support Service database system successfully demonstrates how relational database design can streamline customer support processes. By carefully gathering requirements, designing ER diagrams, implementing SQL scripts, and testing queries, the system meets the needs of both customers and technicians. This project provided practical experience in translating real-world service needs into a functional database application.