SEVA SADAN'S

R. K. TALREJA COLLEGE

OF

ARTS, SCIENCE & COMMERCE ULHASNAGAR – 421003

CERTIFICATE

This is to certify that Mr./Ms. Mohammed Imran Khan of S.Y. Information Technology (SYIT) Roll No. 2542016 has satisfactorily completed the Open Source DataBase Management System Mini Project entitled
Office Asset Management System
during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for submission.

PROF. INCHARGE **HEAD OF DEPT**

_____ _____

INDEX

CHAPTER 1: INTRODUCTION

A database system is a structured way to store, manage, and retrieve data so that information can be used easily and safely. Instead of keeping records in paper files or scattered digital documents, a database system stores all information in organized tables. Each table contains rows and columns, where rows represent individual records and columns represent specific details. In the context of an Office Asset Management system, the database system is used to store information about office departments, employees, assets, and asset assignments. This structured storage helps the organization keep accurate records of assets such as laptops, printers, and furniture. A database system also allows multiple users to access the same data at the same time without causing confusion or data loss. It ensures that data remains consistent even when updates are made. Using a database system reduces manual work, saves time, and improves accuracy. In this project, the database system acts as the backbone that connects all asset-related information and allows smooth operations like adding new assets, assigning them to employees, and tracking their current status.

The Office Asset Management database system addresses the problem of handling office assets using manual or unorganized methods. In many offices, asset details are recorded in registers, Excel sheets, or separate files, which can easily lead to errors. These methods often cause data duplication, missing information, and difficulty in tracking asset usage. For example, it becomes hard to know which employee is using a specific asset or whether an asset is available or already assigned. Manual systems also lack proper security, meaning anyone can change records without permission. Searching for information is slow and timeconsuming, especially when records grow over time. The database system solves these problems by keeping all asset-related data in one central place. It allows quick searching, easy updating, and accurate reporting. By using relationships between tables, the system clearly shows the connection between employees and assets. This helps the office manage resources better, reduce losses, and maintain clear records for decision-making.

MySQL is used in this project because it is a reliable and open-source database management system that is easy to learn and use. Being open-source means it is free to use, which makes it suitable for educational projects and small to medium organizations. MySQL supports all essential database features such as table creation, data insertion, data retrieval, and security control. It also supports advanced concepts like transactions, locking, views, and user privileges, which are used in this Office Asset Management system. MySQL works well with structured query language (SQL), which allows users to interact with the database using simple and readable commands. Another reason for using MySQL is its strong data security and backup support, which helps protect important asset information. It runs smoothly on different operating systems like Windows and Linux. Due to its stability, speed, and wide community support, MySQL is a practical choice for building a database-driven asset management system.

CHAPTER 2: PROBLEM DEFINITION

In many offices, asset management is still handled using manual or unstructured systems such as paper registers, Excel sheets, or simple text files. In this type of system, details about office assets like laptops, printers, and furniture are written by hand or stored in separate digital files. Employee and department details are often maintained in different places without any proper connection between them. Because of this, tracking which asset is assigned to which employee becomes difficult. Updating records requires manual effort, and mistakes are common during data entry. If a file is lost, damaged, or deleted, important information may be gone permanently. There is also no standard format for maintaining records, which leads to confusion. As the number of assets and employees increases, managing records becomes more complex and timeconsuming. This manual approach is not suitable for offices that require accuracy, fast access, and reliable asset tracking.

The manual or unstructured system creates several serious problems in managing office assets. The most common issues are listed below:

- **Data Redundancy:**

  The same asset or employee information may be written multiple times in different files or registers.

  This wastes storage space and increases the chance of errors.

- **Data Inconsistency:**

  When the same data exists in multiple places, updates may not be done everywhere. This results in conflicting information, such as an asset shown as both available and assigned.

- **Lack of Security:**

  Manual records and simple files do not have proper access control. Anyone can view or change the data, which can lead to misuse or data loss.

- **Difficulty in Tracking:**

  Finding out asset status or assignment history takes a lot of time and effort.

These issues reduce efficiency and reliability in office asset management.

To overcome the problems of the manual system, a database-driven solution is required. A database system stores all asset-related information in an organized and structured way. It connects departments, employees, and assets using relationships, which makes tracking easy and accurate. With a database, data is stored only once, which removes redundancy and ensures consistency. Security features such as user access control prevent unauthorized users from modifying data. Queries can be used to quickly retrieve information like available assets or assigned assets. The database also supports transactions, which help maintain data accuracy

during critical operations. Backup and recovery features protect data from loss. Therefore, a database-driven Office Asset Management system is necessary to improve efficiency, accuracy, and security in managing office resources.

CHAPTER 3: OBJECTIVES OF THE PROJECT

The first main objective of this project is to design a structured and well-organized database system for managing office assets. A structured database system means that data is stored in properly defined tables with clear relationships between them. In this project, tables such as department, employee, asset, and asset_assignment are created to store different types of information in an organized manner. Each table serves a specific purpose and avoids mixing different data in one place. This structure helps in managing data easily and reduces confusion. By separating data into related tables, the system becomes easy to understand and maintain. A structured design also helps in future expansion, such as adding new asset types or departments. This objective focuses on building a logical database layout that reflects real office operations and supports smooth asset tracking and management.

Another important objective of this project is to implement SQL queries for managing data efficiently. SQL queries are used to insert, update, retrieve, and manage data stored in the database. In this project, SQL commands like INSERT are used to add new records, SELECT is used to view data, and UPDATE is used to modify asset status. JOIN queries help combine data from multiple tables, such as showing which employee is assigned to which asset. Advanced queries like GROUP BY, HAVING, subqueries, and views are also implemented to improve data analysis and reporting. By using SQL queries, the system allows quick and accurate data handling. This objective helps in understanding how SQL commands work together to manage real-world data in a database system.

Ensuring data integrity is a key objective of this project. Data integrity means that the data stored in the database is accurate, consistent, and reliable. This is achieved by using database constraints such as PRIMARY KEY, FOREIGN KEY, NOT NULL, and UNIQUE. Primary keys ensure that each record is unique, while foreign keys maintain proper relationships between tables. NOT NULL constraints prevent missing important data, and UNIQUE constraints avoid duplicate entries. These rules stop invalid data from entering the system. For example, an asset cannot be assigned to a non-existing employee. This objective ensures that the database follows strict rules, which helps maintain correct and trustworthy asset records at all times.

The final objective of this project is to gain practical, hands-on experience with MySQL as a database management system. Through this project, students learn how to create databases, design tables, write SQL queries, and manage users. Practical experience with transactions, locking, security, backup, and recovery helps in understanding how databases work in real office environments. MySQL provides a simple and userfriendly platform for learning database concepts. By working directly with MySQL, students develop confidence in handling database operations. This objective helps build strong database skills that are useful for academic learning as well as future professional work.

CHAPTER 4: SCOPE OF THE PROJECT

The Office Asset Management system can be used in many places where assets need to be tracked and managed properly. This system is suitable for offices that use assets such as computers, printers, furniture, and other equipment on a daily basis. It can be used in corporate offices to keep records of assets assigned to employees and departments. Educational institutions like colleges and training centers can also use this system to manage lab equipment and office resources. Small businesses and startups can benefit from this system as it helps them control assets without using complex software. The system is useful in any organization where asset tracking is required to avoid loss, misuse, or confusion. Since the database is structured and simple, it can be easily adapted to different working environments. The scope of this project is mainly focused on database-level management and does not depend on a specific user interface, making it flexible and reusable.

The Office Asset Management system can be used by different types of users based on their roles and responsibilities. The main users are listed below:

- **Administrator:**

  The admin has full control over the system. The admin can create and manage departments, employees, assets, and user accounts. The admin is also responsible for assigning assets and managing security, backup, and recovery.

- **Staff or Employees:**

  Staff members can view the assets assigned to them and check asset details. They may not have permission to modify data but can access information related to their usage.

- **Students:**

  In educational institutions, students can be users of the system to view assigned equipment such as lab devices or library assets.

- **Business Users:**

  Business users can use the system to monitor asset usage, availability, and planning future purchases.

These users access the system according to the permissions assigned to them.

This project is developed mainly for academic and learning purposes, so it has certain limitations. The system focuses only on database design and SQL operations. It does not include a graphical user interface or web-based access. The project handles basic asset management and does not cover advanced features like asset depreciation, maintenance tracking, or automated notifications. The database is designed for small-scale use and may require modifications for large organizations. Despite these limitations, the project is useful in the following subdomains:

- **Educational Use:**

Helps students understand database concepts and SQL queries.

- **Business Use:**

Suitable for basic asset tracking in small offices.

- **Administrative Use:**

Assists administrators in managing and maintaining asset records.

CHAPTER 5: REQUIREMENT SPECIFICATION

Hardware Requirements:

• Computer/Laptop

• Minimum 4GB RAM

• 20GB free storage

Software Requirements:

Software	**Purpose**
MySQL Server	Database management
MySQL Workbench	Query execution
OS (Windows/Linux)	Platform
SQL	Query language

CHAPTER 6: SYSTEM DESIGN

The system architecture of the Office Asset Management system explains how the user, database, and MySQL server work together. This system follows a simple database-driven approach where all operations are handled using SQL queries. The design focuses only on database interaction and processing, without involving hardware or user interface design. The architecture ensures that data is accessed, processed, and stored in a secure and organized manner.

- The user interacts with the database by using SQL commands through tools such as MySQL Workbench.
- The user performs operations like adding assets, assigning assets, and viewing records by executing SQL queries.
- Commands such as INSERT, SELECT, UPDATE, and JOIN are used to manage asset data.
- The user does not directly access database files; all interaction happens through SQL.
- User actions are controlled by access permissions to prevent unauthorized changes.
- The MySQL server stores all data related to departments, employees, assets, and assignments.
- It manages table relationships and applies constraints to maintain data accuracy.
- The server checks user privileges before allowing any operation.
- It handles transactions and locking to prevent data conflicts.
- MySQL also supports backup and recovery to protect important asset data.
- The process starts when the user submits an SQL query.
- The MySQL server checks the user’s permission for the requested operation.
- The query is validated for errors and correctness.
- If valid, the server executes the query on the database.
- Constraints and locks are applied during execution if needed.
- The result or confirmation is sent back to the user.

CHAPTER 7: DATABASE DESIGN

The Department table is used to store information about different departments in the office. Each department represents a functional unit such as IT, HR, or Finance.

The Employee table stores information related to office employees. It contains details such as employee name, email address, and department association. Each employee belongs to one department, which is maintained using a relationship with the Department table

The Asset table stores details of all office assets such as laptops, printers, and furniture. It includes information about asset name, type, purchase date, and current status. This table helps track whether an asset is available or assigned. It is the core table of the system, as all asset management operations depend on it.

The Asset_Assignment table manages the relationship between employees and assets. It records which asset is assigned to which employee along with assignment and return dates. This table helps track asset history and usage. It connects the Employee and Asset tables and ensures proper asset tracking.

Table: Department

- **Attributes:**
  - dept_id
  - dept_name
- **Data Types:**
  - dept_id – INT
  - dept_name – VARCHAR(50)

Table: Employee

- **Attributes:** ○ emp_id
  - emp_name
  - emp_email
  - dept_id

- **Data Types:** ○ emp_id – INT
    - emp_name – VARCHAR(50)
    - emp_email – VARCHAR(50)
    - dept_id – INT

Table: Asset

- **Attributes:**
    - asset_id ○ asset_name ○ asset_type ○ purchase_dat e ○ asset_status

- **Data Types:**
    - asset_id – INT
    - asset_name – VARCHAR(50)
    - asset_type – VARCHAR(30)
    - purchase_date – DATE
    - asset_status – VARCHAR(20)

Table: Asset_Assignment

- **Attributes:** ○ assign_id
    - asset_id
    - emp_id
    - assign_date
    - return_date

- **Data Types:**
    - assign_id – INT
    - asset_id – INT
    - emp_id – INT
    - assign_date – DATE
    - return_date – DATE

- **PRIMARY KEY:**

Primary keys are used to uniquely identify each record in a table. In this project, fields like dept_id, emp_id, asset_id, and assign_id are primary keys. They ensure that no two records have the same identity and help maintain data accuracy.

- **FOREIGN KEY:**

Foreign keys are used to create relationships between tables. The dept_id in the Employee table connects employees to departments. The asset_id and emp_id in the Asset_Assignment table connect assets to employees. This ensures valid relationships and prevents incorrect data entry.

- **NOT NULL:**

NOT NULL constraints ensure that important fields cannot be left empty. For example, names of departments, employees, and assets must always have values. This helps maintain complete and meaningful records.

- **UNIQUE:**

UNIQUE constraints ensure that certain values are not repeated. In this project, employee email addresses are unique so that no two employees can have the same email. This helps avoid confusion and duplication.

CHAPTER 8: UML DIAGRAMS

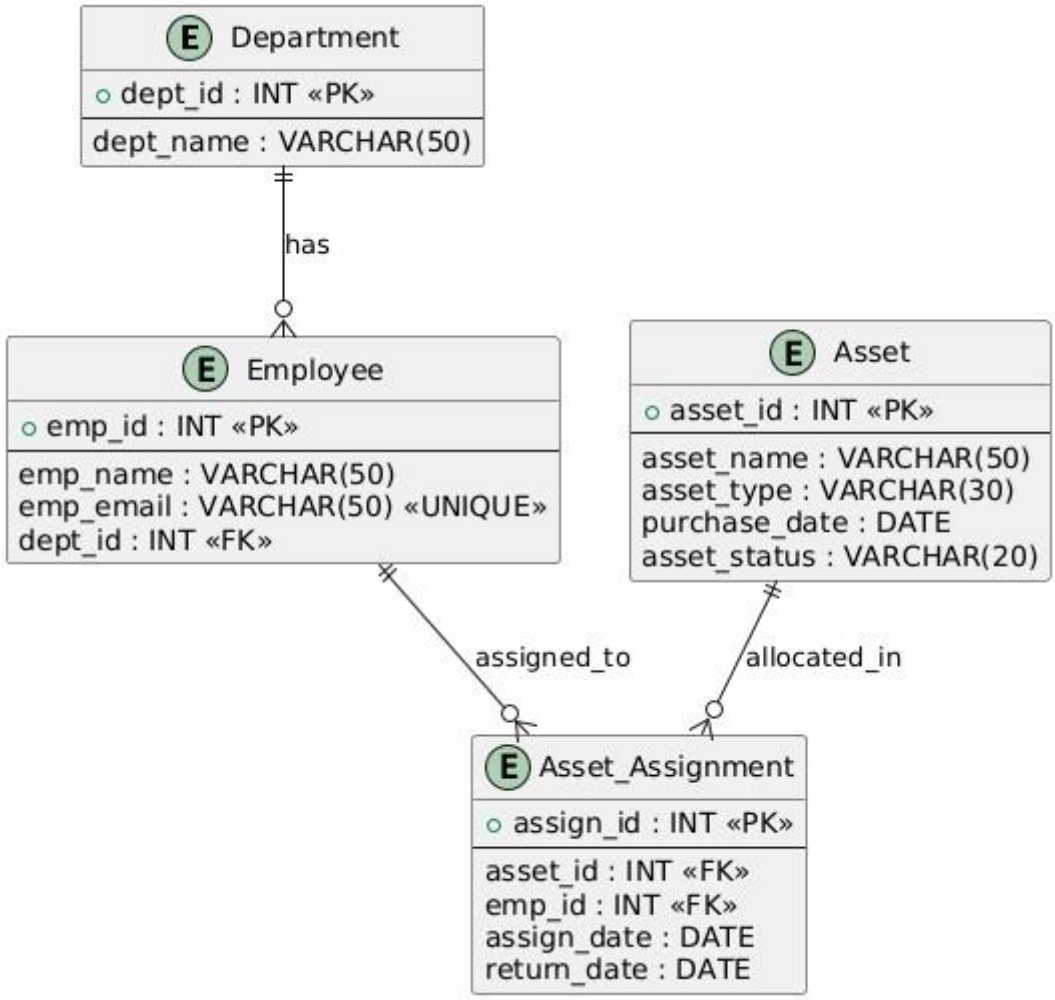

Figure 8.1: ER[Entity Relationship] Diagram

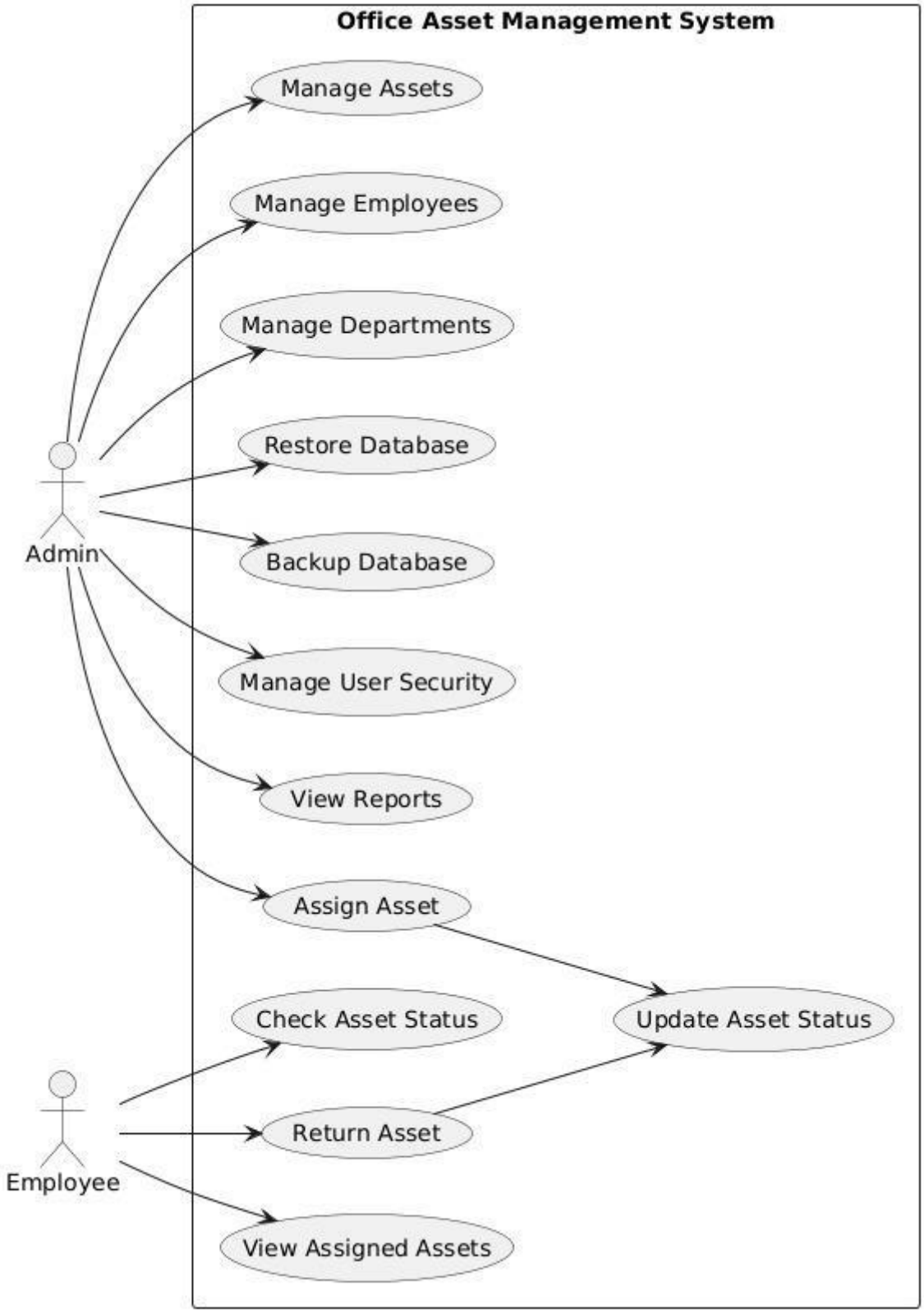

Figure 8.2: USE CASE Diagram

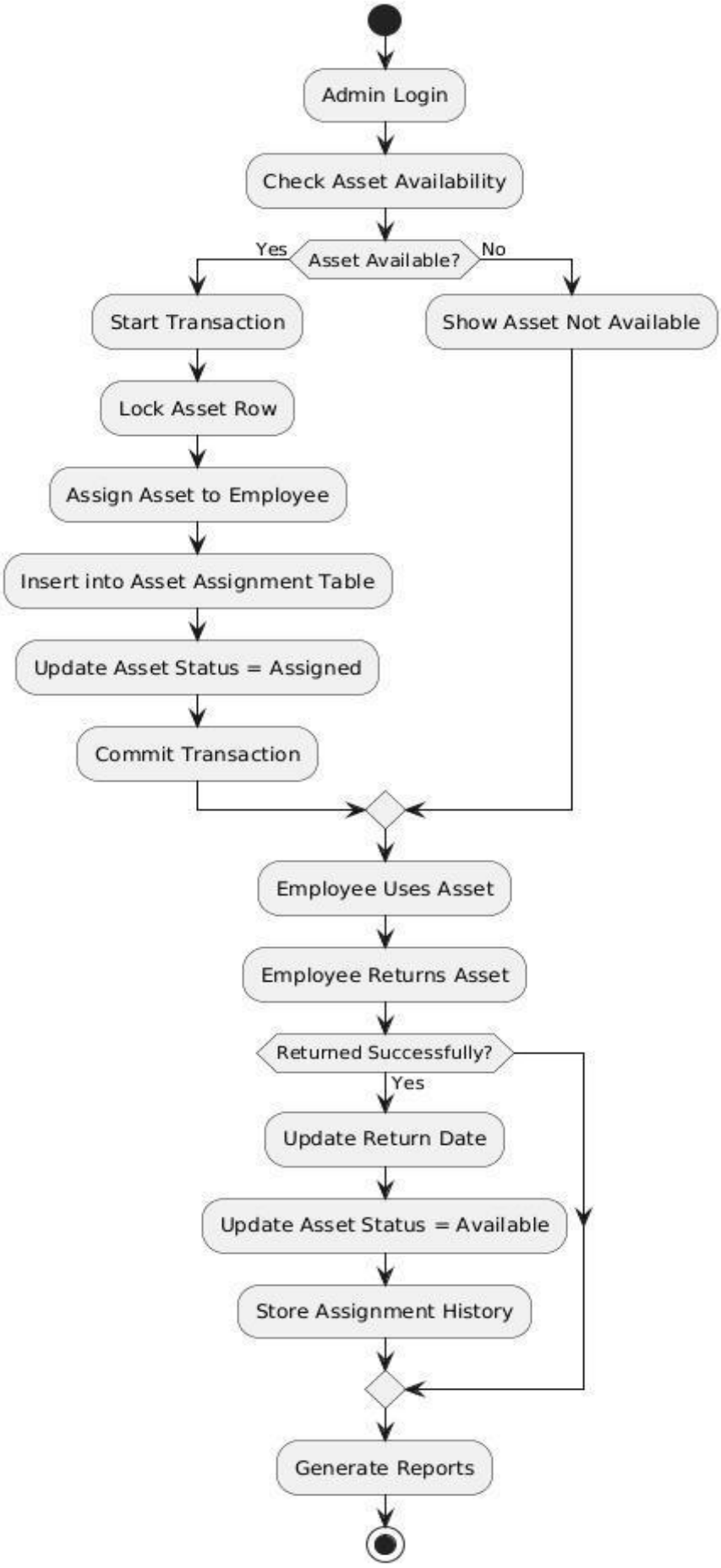

Figure 8.3: Activity Diagram

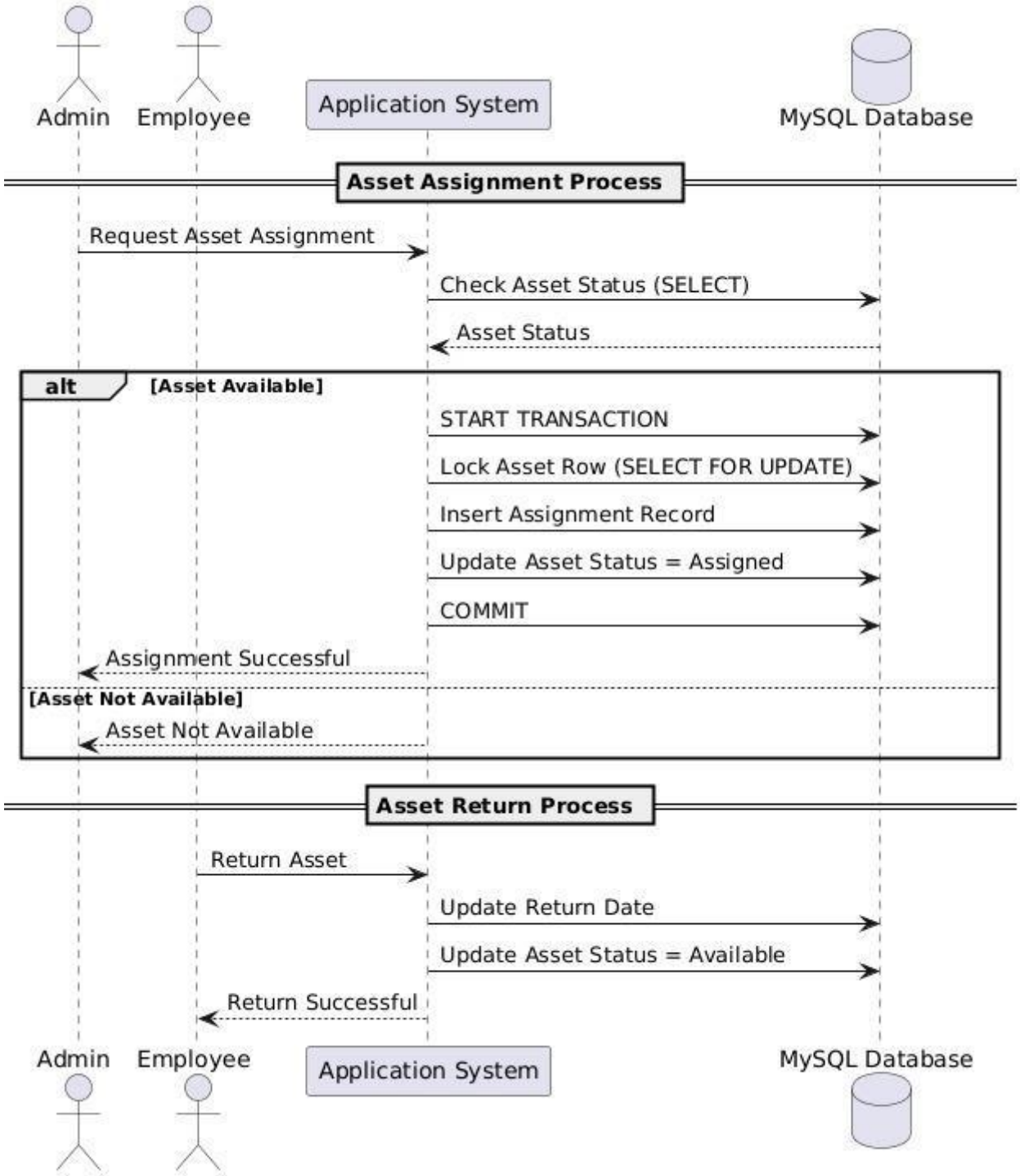

Figure 8.4: Sequence Diagram

CHAPTER 9: SQL IMPLEMENTATION

9.1 Database Creation

```
CREATE DATABASE office_asset_management;

USE office_asset_management;
```

9.2 Table Creation

Department Table:

```
CREATE TABLE department (
   dept_id INT PRIMARY KEY AUTO_INCREMENT,
   dept_name VARCHAR(50) NOT NULL UNIQUE
);
```

Employee Table:

```
CREATE TABLE employee (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_name VARCHAR(50) NOT NULL,
    emp_email VARCHAR(100) NOT NULL UNIQUE,
    dept_id INT NOT NULL,
    FOREIGN KEY (dept_id)
        REFERENCES department(dept_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

Asset Table:

```
CREATE TABLE asset (
    asset_id INT PRIMARY KEY AUTO_INCREMENT,
    asset_name VARCHAR(50) NOT NULL,
    asset_type VARCHAR(30) NOT NULL,
    purchase_date DATE NOT NULL,
    asset_status ENUM('Available','Assigned','Under Maintenance')
        DEFAULT 'Available'
);
```

Asset Assignment Table:

```
CREATE TABLE asset_assignment (
    assign_id INT PRIMARY KEY AUTO_INCREMENT,
    asset_id INT NOT NULL,
    emp_id INT NOT NULL,
    assign_date DATE NOT NULL,
    return_date DATE,
    FOREIGN KEY (asset_id)
        REFERENCES asset(asset_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE,
    FOREIGN KEY (emp_id)
        REFERENCES employee(emp_id)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);
```

9.3 Data Insertion

Insert Departments:

```
INSERT INTO department (dept_name) VALUES

('IT'),

('HR'),

('Finance');
```

Insert Employees:

```
INSERT INTO employee (emp_name, emp_email, dept_id)
VALUES

('Rahul Sharma', 'rahul@office.com', 1),

('Aisha Khan', 'aisha@office.com', 2),

('Vikram Patel', 'vikram@office.com', 1);
```

Insert Assets:

```
INSERT INTO asset (asset_name, asset_type,

purchase_date)

VALUES

('Dell Laptop', 'Electronics', '2023-01-10'),

('HP Printer', 'Electronics', '2022-08-15'),

('Office Chair', 'Furniture', '2021-05-20');
```

9.4 Asset Assignment

Assign Assets

```
INSERT INTO asset_assignment (asset_id, emp_id,
assign_date, return_date)

VALUES
```

```
(1, 1, '2024-01-01', NULL),
(2, 2, '2024-02-10', NULL);
```

Update Asset Status

```
UPDATE asset
SET asset_status = 'Assigned'
WHERE asset_id IN (1,2);
```

9.5 Data Retrieval Queries

View All Available Assets

```
SELECT * FROM asset
WHERE asset_status = 'Available';
```

Employee with Department

```
SELECT e.emp_name, e.emp_email, d.dept_name
FROM employee e
JOIN department d
ON e.dept_id = d.dept_id;
```

Asset Assignment Details

```
SELECT a.asset_name, e.emp_name, aa.assign_date, aa.return_date
FROM asset_assignment aa
JOIN asset a ON aa.asset_id = a.asset_id
JOIN employee e ON aa.emp_id = e.emp_id;
```

9.6 Advanced Queries

GROUP BY

```
SELECT asset_type, COUNT(*) AS total_assets
FROM asset
GROUP BY asset_type;
```

HAVING

```
SELECT asset_type, COUNT(*) AS total_assets
FROM asset
GROUP BY asset_type
HAVING COUNT(*) > 1;
```

Subquery – Employees Who Have Assets

```
SELECT emp_name
FROM employee
WHERE emp_id IN (
    SELECT emp_id
    FROM asset_assignment
);
```

9.7 VIEW CREATION

```
CREATE VIEW asset_assignment_view AS
SELECT e.emp_name, a.asset_name, aa.assign_date, aa.return_date
FROM asset_assignment aa
JOIN employee e ON aa.emp_id = e.emp_id
JOIN asset a ON aa.asset_id = a.asset_id;
SELECT * FROM asset_assignment_view;
```

CHAPTER 10: SYSTEM TESTING AND RESULT

```
mysql> SELECT * FROM asset
    -> WHERE asset_status = 'Available';
+----------+--------------+------------+---------------+--------------+
| asset_id | asset_name   | asset_type | purchase_date | asset_status |
+----------+--------------+------------+---------------+--------------+
|        2 | HP Printer   | Electronics | 2022-08-15    | Available    |
|        3 | Office Chair | Furniture   | 2021-05-20    | Available    |
|        4 | Dell Laptop  | Electronics | 2023-01-10    | Available    |
|        5 | HP Printer   | Electronics | 2022-08-15    | Available    |
|        6 | Office Chair | Furniture   | 2021-05-20    | Available    |
+----------+--------------+------------+---------------+--------------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT e.emp_name, d.dept_name
    -> FROM employee e
    -> JOIN department d
    -> ON e.dept_id = d.dept_id;
+--------------+-----------+
| emp_name     | dept_name |
+--------------+-----------+
| Rahul Sharma | IT        |
| Aisha Khan   | HR        |
| Vikram Patel | IT        |
+--------------+-----------+
3 rows in set (0.00 sec)
```

```
mysql> SELECT asset_type, COUNT(*) AS total_assets
    -> FROM asset
    -> GROUP BY asset_type;
+-------------+--------------+
| asset_type  | total_assets |
+-------------+--------------+
| Electronics |            4 |
| Furniture   |            2 |
+-------------+--------------+
2 rows in set (0.01 sec)

mysql>
mysql> SELECT asset_type, COUNT(*) AS total_assets
    -> FROM asset
    -> GROUP BY asset_type
    -> HAVING COUNT(*) > 1;
+-------------+--------------+
| asset_type  | total_assets |
+-------------+--------------+
| Electronics |            4 |
| Furniture   |            2 |
+-------------+--------------+
2 rows in set (0.00 sec)
```

CHAPTER 11: SECURITY, BACKUP AND RECOVERY

Security is an important part of the Office Asset Management system because the database stores important office information. Proper security ensures that data is protected from unauthorized access and misuse. In this project, security is mainly handled using user privileges and access control features provided by MySQL. These features help control who can view or modify the data. Only authorized users are allowed to perform specific operations on the database. This helps maintain data safety and reliability.

- **User Privileges:**

  User privileges define what actions a user can perform on the database. In this system, a separate database user is created and given limited privileges such as SELECT, INSERT, and UPDATE. This means the user can view and modify data but cannot delete tables or change the database structure. Privileges help protect the database from accidental or unauthorized changes.

- **Access Control:**

  Access control ensures that only approved users can access the database. Each user is given a username and password. MySQL checks these credentials before allowing access. Access control prevents unauthorized users from entering the system and ensures that sensitive asset data remains secure.

Backup is the process of creating a copy of the database to prevent data loss. In this project, backup is taken using the mysqldump tool provided by MySQL.

- **mysqldump Explanation:**

  The mysqldump command is used to export the entire database into a SQL file. This file contains all table structures and data. If the system fails or data is lost, this backup file can be used to restore the database. Regular backups help protect important asset information and ensure data safety.

Recovery is the process of restoring the database using a backup file. It helps bring the database back to a previous working state after failure or data corruption.

- **Restore Concept:**

  The restore process uses the backup SQL file created by mysqldump. By importing this file into MySQL, the database structure and data are recreated. This ensures that no important asset data is permanently lost. Recovery is essential for maintaining system reliability and business continuity.

CHAPTER 12: FUTURE SCOPE AND CONCLUSION

The Office Asset Management system developed in this project provides a basic and reliable database solution, but it can be further improved in the future. One possible enhancement is **web integration**. The database can be connected to a web application so that users can access asset information through a browser. This would make the system easier to use and allow multiple users to interact with the database remotely. Another improvement is advanced reporting. Reports such as asset usage history, department-wise asset allocation, and asset availability summaries can be generated automatically. These reports would help management in decision-making. The system can also be extended with analytics features. Analytics can be used to study asset usage patterns, identify frequently used assets, and plan future asset purchases. These improvements would make the system more powerful and suitable for real-world office environments.

This project successfully achieved its goal of developing an Office Asset Management database system using MySQL. The system was able to store, manage, and retrieve asset-related data in an organized manner. It helped track assets, employees, and assignments accurately. Through this project, important database concepts such as table creation, relationships, constraints, queries, transactions, security, backup, and recovery were learned and implemented. The project provided valuable learning outcomes by giving practical experience with SQL and database design. MySQL proved to be an important tool due to its simplicity, reliability, and open-source nature. Overall, the project demonstrated how a database system can improve data management and efficiency in an office environment.

CHAPTER 13: REFERENCES

- **MySQL Official Documentation:**

MySQL Documentation, Oracle Corporation.

Used as the primary reference for understanding MySQL commands, database creation, table design, queries, security, backup, and recovery concepts.

- **Prescribed Textbooks:**

Standard textbooks on Database Management Systems recommended in the academic syllabus. These books were referred to for understanding basic DBMS concepts such as relational databases, SQL queries, constraints, and normalization.

- **Online Learning Sources:**

Educational websites, tutorials, and online learning platforms related to SQL and MySQL.

These sources helped in learning practical query execution, examples, and best practices for database implementation.

CHAPTER 14: GLOSSARY

The glossary section provides simple meanings of important terms used throughout the Office Asset Management project. These terms are related to database systems and SQL concepts that are essential to understand how the project works. The definitions are written in easy language so that readers with basic or no prior knowledge of databases can understand them clearly. This section helps remove confusion and acts as a quick reference for technical words used in the documentation.

- **DBMS (Database Management System):**

A DBMS is software that is used to store, manage, and retrieve data in an organized way. It allows users to handle large amounts of data efficiently and securely.

- **SQL (Structured Query Language):**

SQL is a standard language used to communicate with a database. It is used to create tables, insert data, retrieve records, and manage database operations.

- **Primary Key:**

A primary key is a field that uniquely identifies each record in a table. It ensures that no two records have the same value.

- **Foreign Key:**

A foreign key is a field that connects one table to another table. It helps maintain relationships and data consistency between tables.

- **MySQL:**

MySQL is an open-source relational database management system used to create and manage databases. It is widely used because it is reliable, easy to use, and supports all basic database features.