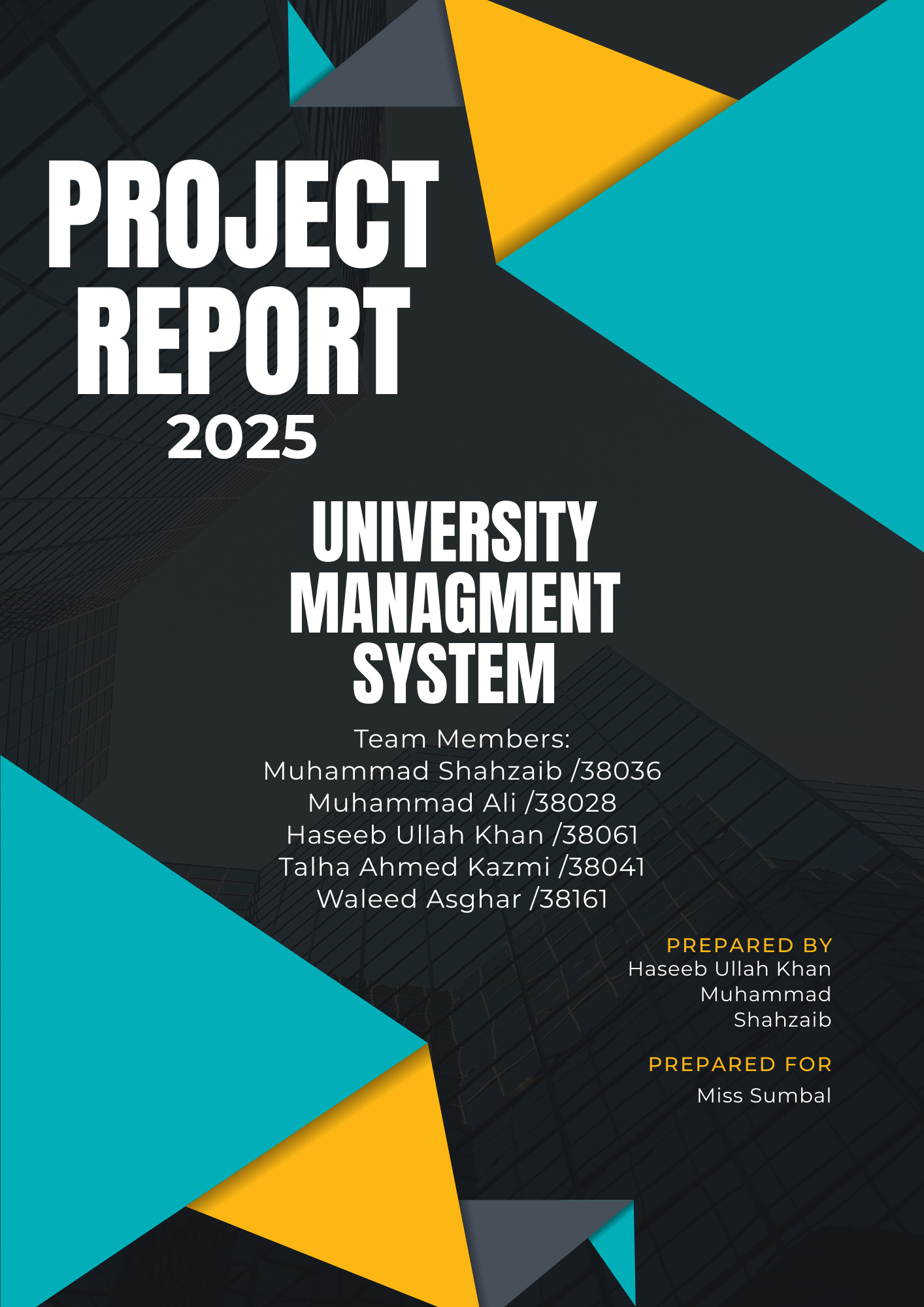
****

Contents

[Abstract 2](#_Toc198646501)

[1. Introduction 2](#_Toc198646502)

[2. Objectives 2](#_Toc198646503)

[3. System Overview 3](#_Toc198646504)

[4. Original Schema Design 3](#_Toc198646505)

[5. Modifications and Justification 3](#_Toc198646506)

[6. Entity-Relationship Diagram 4](#_Toc198646507)

[7. Implementation Details 4](#_Toc198646508)

[8. Testing and Validation 4](#_Toc198646509)

[12. Conclusion and Future Work 4](#_Toc198646510)

[13. References 5](#_Toc198646511)

[14. Appendix: Graphviz DOT Code 5](#_Toc198646512)

# 

# Abstract

This project presents a comprehensive database solution for managing university facilities, including room bookings, asset tracking, event organization, and service requests. The system is designed using a normalized relational schema with consistent naming conventions and enforced referential integrity. Major enhancements include the addition of timestamp fields, clear entity relationships, and a structured ERD using Graphviz. The database supports modular functionality for tenant companies, inventory, user roles, and petty cash handling. Extensive testing ensured data accuracy and performance. This foundation enables future integration of features such as role-based access control (RBAC), reporting dashboards, and audit trail capabilities for greater system transparency.

# 1. Introduction

Facility management in a university environment is a critical function that ensures the smooth operation of campus infrastructure, including classrooms, meeting spaces, maintenance services, and resource allocation. As universities grow in size and complexity, the need for a structured and efficient facility management system becomes increasingly important. Traditionally, many of these tasks are handled manually or through disconnected systems, leading to inefficiencies and data inconsistencies.

One of the primary challenges faced by institutions is the lack of centralized tracking for assets, inconsistent room reservation processes, and difficulty in managing service requests. These issues can result in resource conflicts, maintenance delays, and poor accountability.

This project addresses these challenges by developing a campus-wide facility management system using a relational database approach. The scope includes comprehensive modules for managing rooms, assets, parking tags, cleaning services, event permits, and petty cash usage. The system is designed for scalability, accuracy, and integration with future administrative tools.

# 2. Objectives

* Design a normalized, scalable relational database.
* Enforce referential integrity and naming conventions.
* Provide clear ER modeling and documentation.
* Implement and demonstrate core modules: rooms, assets, services, events, cash management.

# 3. System Overview

The system includes modules for tenant companies, workers (with card types), meeting rooms, cleaning services, event requests, inventory management, petty cash, news & parking, and user roles. Key entities consist of tenants, workers, rooms, events, inventory items, cash records, news posts, parking slots, and users. Their relationships are summarized in a table showing main tables and cardinalities, detailing how entities like tenants link to workers and rooms, events relate to cleaning and inventory, and user roles govern access.

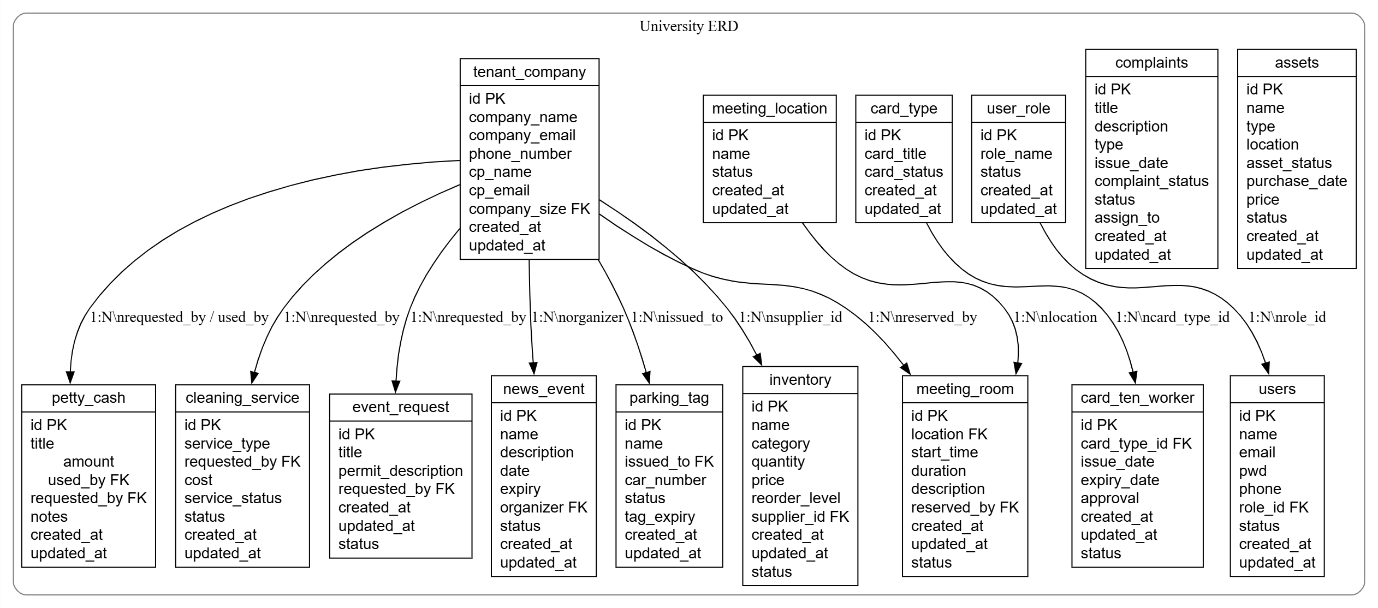
# 4. Original Schema Design

The original schema design included tables for tenants, workers, rooms, events, inventory, petty cash, news, parking, and user roles, each with relevant attributes. However, several issues were identified, such as missing foreign keys in event and inventory tables, inconsistent naming conventions (e.g., mixing snake\_case and camelCase), and unclear or undocumented status fields in modules like cleaning services and event requests. These gaps affected data integrity and clarity, highlighting the need for refinement and standardization across the schema.

# 5. Modifications and Justification

|  |  |  |
| --- | --- | --- |
| Modification | Justification | Impact |
| Renamed card\_ten\_worker.title → card\_type\_id | Clarify relation to card\_type; enforce FK | Improved referential integrity |
| Added FKs on inventory.supplier\_id | Link suppliers to tenant companies | Ensures valid supplier references |
| Introduced status lookup/enums | Standardize status across multiple tables | Consistency and easier maintenance |
| Added created\_at / updated\_at timestamps | Automate record tracking | Better audit and history |
| Unified naming to snake\_case & singular tables | Align with best practices | Consistent, readable schema |

# 6. Entity-Relationship Diagram

****

# 7. Implementation Details

The implementation involved DDL scripts for creating tables with appropriate constraints, indexes for performance, and enums for standardized status values (e.g., request\_status, room\_type). Foreign keys were added to maintain relational integrity. Sample queries included inserting a meeting room reservation with tenant and room IDs, date, and time details, and generating a monthly petty cash report by aggregating expenses grouped by category and filtered by date. These queries support core functionality and ensure reliable data management within the system.

# 8. Testing and Validation

* Data Integrity Tests: FK violations, null constraints.
* Performance Checks: Index usage, query EXPLAIN plans.

# 12. Conclusion and Future Work

The project successfully delivered a robust and well-structured database schema with clear documentation and a comprehensive ERD design, ensuring data consistency and logical relationships. Key modules were implemented with functional queries and optimized constraints. For future enhancements, introducing role-based access controls (RBAC) will improve security, while interactive report dashboards and audit trails can enhance system transparency, monitoring, and user accountability. These additions will further strengthen the system’s scalability and usability for real-world facility management scenarios.

# 13. References

1. Elmasri & Navathe, *Fundamentals of Database Systems*.
2. Graphviz documentation: https://graphviz.org.
3. SQL style guides and best practices.

# 14. Appendix: Graphviz DOT Code

digraph UniversityDB {

graph [rankdir=TB, ranksep=1.0];

node [shape=record, fontname=Helvetica];

subgraph cluster\_university {

label = "University ERD";

style = rounded;

color = gray;

tenant\_company [label="{tenant\_company|id PK\lcompany\_name\lcompany\_email\lphone\_number\lcp\_name\lcp\_email\lcompany\_size FK\lcreated\_at\lupdated\_at\l}"];

card\_type [label="{card\_type|id PK\lcard\_title\lcard\_status\lcreated\_at\lupdated\_at\l}"];

card\_ten\_worker[label="{card\_ten\_worker|id PK\lcard\_type\_id FK\lissue\_date\lexpiry\_date\lapproval\lcreated\_at\lupdated\_at\lstatus\l}"];

cleaning\_service [label="{cleaning\_service|id PK\lservice\_type\lrequested\_by FK\lcost\lservice\_status\lstatus\lcreated\_at\lupdated\_at\l}"];

event\_request [label="{event\_request|id PK\ltitle\lpermit\_description\lrequested\_by FK\lcreated\_at\lupdated\_at\lstatus\l}"];

meeting\_location [label="{meeting\_location|id PK\lname\lstatus\lcreated\_at\lupdated\_at\l}"];

meeting\_room [label="{meeting\_room|id PK\llocation FK\lstart\_time\lduration\ldescription\lreserved\_by FK\lcreated\_at\lupdated\_at\lstatus\l}"];

news\_event [label="{news\_event|id PK\lname\ldescription\ldate\lexpiry\lorganizer FK\lstatus\lcreated\_at\lupdated\_at\l}"];

parking\_tag [label="{parking\_tag|id PK\lname\lissued\_to FK\lcar\_number\lstatus\ltag\_expiry\lcreated\_at\lupdated\_at\l}"];

inventory [label="{inventory|id PK\lname\lcategory\lquantity\lprice\lreorder\_level\lsupplier\_id FK\lcreated\_at\lupdated\_at\lstatus\l}"];

user\_role [label="{user\_role|id PK\lrole\_name\lstatus\lcreated\_at\lupdated\_at\l}"];

users [label="{users|id PK\lname\lemail\lpwd\lphone\lrole\_id FK\lstatus\lcreated\_at\lupdated\_at\l}"];

complaints [label="{complaints|id PK\ltitle\ldescription\ltype\lissue\_date\lcomplaint\_status\lstatus\lassign\_to\lcreated\_at\lupdated\_at\l}"];

assets [label="{assets|id PK\lname\ltype\llocation\lasset\_status\lpurchase\_date\lprice\lstatus\lcreated\_at\lupdated\_at\l}"];

petty\_cash [label="{petty\_cash|id PK\ltitle\lamount\nused\_by FK\nrequested\_by FK\lnotes\lcreated\_at\lupdated\_at\l}"];

/\* Relationships with FK and cardinality annotations \*/

card\_type -> card\_ten\_worker [label="1:N\\ncard\_type\_id"];

tenant\_company -> cleaning\_service [label="1:N\\nrequested\_by"];

tenant\_company -> event\_request [label="1:N\\nrequested\_by"];

meeting\_location-> meeting\_room [label="1:N\\nlocation"];

tenant\_company -> meeting\_room [label="1:N\\nreserved\_by"];

tenant\_company -> news\_event [label="1:N\\norganizer"];

tenant\_company -> parking\_tag [label="1:N\\nissued\_to"];

tenant\_company -> inventory [label="1:N\\nsupplier\_id"];

user\_role -> users [label="1:N\\nrole\_id"];

tenant\_company -> petty\_cash [label="1:N\\nrequested\_by / used\_by"];

}

}