



**UNIVERSITY TECHNOLOGY MARA (UiTM), CAWANGAN NEGERI SELANGOR,
KAMPUS PUNCAK PERDANA
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
BACHELOR OF INFORMATION SCIENCE (HONORS)
INFORMATION CONTENT MANAGEMENT - CDIM263**

ADVANCED DATABASE MANAGEMENT SYSTEM (IMS560)

INDIVIDUAL ASSIGNMENT: HOSTEL MANAGEMENT SYSTEM

GROUP: IMD2634E

PREPARED BY:

RAZ ADELYNA DALILAH BINTI RAZIF (2025179429)

PREPARED FOR:

AMZARI BIN ABU BAKAR

SUBMISSION DATE: 30 DISEMBER 2025

TABLE OF CONTENTS

| | |
|---|-----------|
| 1.0 INTRODUCTION | 3 |
| 2.0 BUSINESS RULES..... | 4 |
| 3.0 ENTITIES AND RELATIONSHIP..... | 5 |
| 4.0 ENTITY RELATIONSHIP DIAGRAM (ERD)..... | 6 |
| 5.0 DATA DICTIONARY..... | 7 |
| 6.0 CONCLUSIONS..... | 11 |

1.0 INTRODUCTION

A Hostel Management System (HMS) is a computerized system designed to manage and organize the daily administrative and operational activities of a hostel. The system helps replace manual, paper-based record keeping with a centralized digital platform that is more efficient, accurate, and secure.

The Hostel Management System manages various hostel operations such as student registration, room allocation, fee and fine management, complaint handling, outing records, visitor monitoring, and asset tracking. By using this system, hostel management can easily monitor hostel activities, improve financial transparency, and enhance student safety. At the same time, students benefit from convenient access to their personal information, payment records, and complaint submissions.

This report presents the analysis and design of a Hostel Management System by identifying business rules, entities and their relationships, an Entity Relationship (ER) diagram, and a data dictionary.

2. BUSINESS RULES

The following business rules describe the relationships and constraints between entities in the Hostel Management System:

1. A student can occupy only one room in one semester.
2. A room can be occupied by one or more students, depending on the room capacity.
3. A hostel block can be occupied by only one gender, either male or female.
4. A hostel block cannot have mixed gender occupancy.
5. A student must be registered before being assigned to a room.
6. A student must make hostel fee payments based on the room occupied.
7. A student may receive one or more fines for violating hostel rules.
8. A student can submit one or more complaints regarding hostel facilities.
9. A complaint is monitored and managed by a warden.
10. A warden can manage multiple complaints at any given time.
11. A student must record check-in and check-out times for every outing.
12. A student may submit a request for irregular outing approval.
13. A visitor must be registered before entering the hostel premises.
14. A visitor is associated with only one student per visit.
15. A room can contain one or more hostel assets.
16. A hostel asset is assigned to only one room at a time.

3.0 ENTITIES AND RELATIONSHIP

3.1 IDENTIFIED ENTITIES

Based on the Hostel Management System requirements, the following main entities are identified:

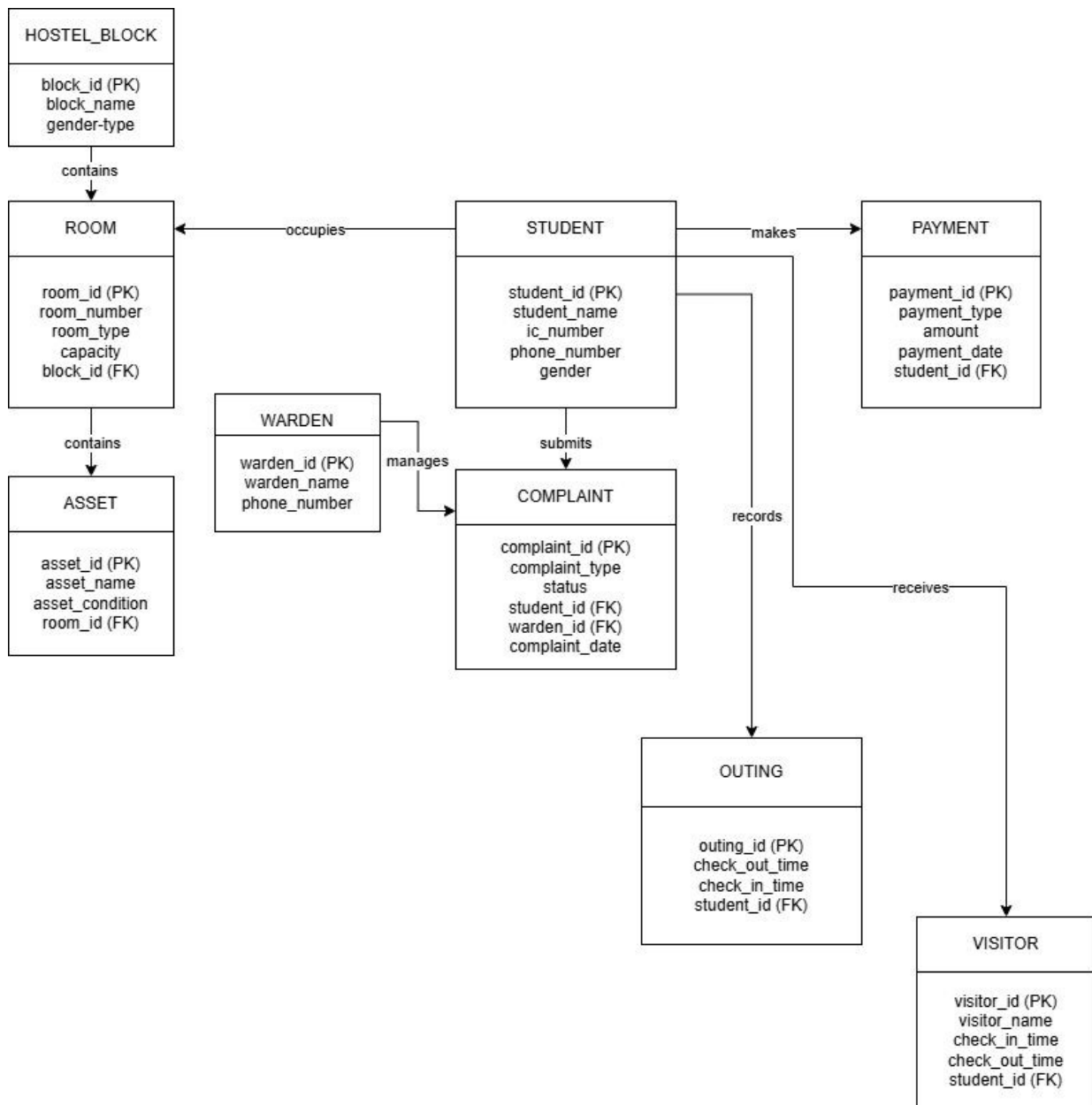
1. **Student** – Represents students who stay in the hostel.
2. **Room** – Represents hostel rooms assigned to students.
3. **Hostel Block** – Represents blocks in the hostel that separate students by gender.
4. **Payment** – Represents hostel fee and fine payments made by students.
5. **Complaint** – Represents maintenance or facility issues reported by students.
6. **Warden** – Represents hostel staff responsible for managing hostel operations.
7. **Outing** – Represents student check-in and check-out records.
8. **Visitor** – Represents visitors entering the hostel premises.
9. **Asset** – Represents hostel assets such as beds, tables, and chairs.

3.2 RELATIONSHIP BETWEEN ENTITIES

The relationships between the entities are described as follows:

1. A **student occupies a room** during a semester.
2. A **room is located in a hostel block**.
3. A **hostel block accommodates students of only one gender**.
4. A **student makes one or more payments** for hostel fees and fines.
5. A **student submits one or more complaints**.
6. A **warden manages multiple complaints**.
7. A **student records outing check-in and check-out times**.
8. A **student receives visitors** during approved visiting hours.
9. A **visitor visits one student per visit**.
10. A **room contains one or more assets**.

4.0 ENTITY RELATIONSHIP DIAGRAM ERD



The Entity Relationship Diagram (ERD) illustrates the relationships between entities in the Hostel Management System using Crow's Foot notation. Each relationship is represented with a verb to describe the interaction between entities. The diagram shows how students are assigned to rooms, how payments and complaints are managed, and how hostel operations are organized.

5.0 DATA DICTIONARY

5.1 STUDENT

| Attribute | Data Type | Length | Description |
|--------------|-----------|--------|---|
| student_id | INT | 10 | Unique identifier for student. Example: 1001 |
| Student_name | VARCHAR | 100 | Full name of student. Example: “Nur Aisyah Binti Ahmad” |
| Ic_number | VARCHAR | 12 | Student identity card number. Example: “010203045678” |
| Phone_number | VARCHAR | 15 | Student contact number. Example: “0123456789” |
| gender | VARCHAR | 10 | Student gender. Example: “Female” |

5.2 ROOM

| Attribute | Data Type | Length | Description |
|-------------|-----------|--------|--|
| room_id | INT | 10 | Unique identifier for room. Example: 201 |
| room_number | VARCHAR | 10 | Room number. Example: “A-203” |
| room_type | VARCHAR | 20 | Type of room. Example: “Double” |
| capacity | INT | 2 | Maximum number of students in room. Example: 2 |
| block_id | INT | 10 | Hostel block identifier (FK). Example: 1 |

5.3 HOSTEL_BLOCK

| Attribute | Data Type | Length | Description |
|-------------|-----------|--------|--|
| block_id | INT | 10 | Unique identifier for hostel block. Example: 1 |
| block_name | VARCHAR | 50 | Name of hostel block. Example: "Block A" |
| gender_type | VARCHAR | 10 | Gender assigned to block. Example: "Male" |

5.4 PAYMENT

| Attribute | Data Type | Length | Description |
|--------------|-----------|--------|--|
| payment_id | INT | 10 | Unique identifier for payment. Example: 5001 |
| payment_type | VARCHAR | 20 | Type of payment. Example: "Fee" |
| amount | DECIMAL | 8.2 | Amount paid. Example: 450.00 |
| payment_date | DATE | - | Date of payment. Example: "2025-06-01" |
| student_id | INT | 10 | Student identifier (FK). Example: 1001 |

5.5 COMPLAINT

| Attribute | Data Type | Length | Description |
|----------------|-----------|--------|--|
| complaint_id | INT | 10 | Unique identifier for complaint. Example: 3001 |
| complaint_type | VARCHAR | 50 | Type of complaint. Example: "Fan not working" |
| status | VARCHAR | 20 | Complaint status. Example: "Pending" |
| student_id | INT | 10 | Student identifier (FK). Example: 1001 |
| warden_id | INT | 10 | Warden identifier (FK). Example: 10 |

5.6 WARDEN

| Attribute | Data Type | Length | Description |
|--------------|-----------|--------|--|
| warden_id | INT | 10 | Unique identifier for warden. Example: 10 |
| warden_name | VARCHAR | 100 | Name of warden. Example: “Mr Ahmad” |
| phone_number | VARCHAR | 15 | Warden contact number. Example: “0134567890” |

5.7 ASSET

| Attribute | Data Type | Length | Description |
|-----------------|-----------|--------|--|
| asset_id | INT | 10 | Unique identifier for asset. Example: 9001 |
| asset_name | VARCHAR | 50 | Name of asset. Example: “Bed” |
| asset_condition | VARCHAR | 20 | Condition of asset. Example: “Good” |
| room_id | INT | 10 | Room identifier (FK). Example: 201 |

5.8 OUTING

| Attribute | Data Type | Length | Description |
|----------------|-----------|--------|--|
| outing_id | INT | 10 | Unique identifier for outing. Example: 7001 |
| check_out_time | DATETIME | - | Time student checks out. Example: “2025-06-10 18:00” |
| check_in_time | DATETIME | - | Time student checks in. Example: “2025-06-10 22:00” |
| student_id | INT | 10 | Student identifier (FK). Example: 1001 |

5.9 VISITOR

| Attribute | Data Type | Length | Description |
|----------------|-----------|--------|---|
| visitor_id | INT | 10 | Unique identifier for visitor. Example: 8001 |
| visitor_name | VARCHAR | 100 | Name of visitor. Example: “Siti Aminah” |
| check_in_time | DATETIME | - | Visitor check-in time. Example: “2025-06-10 14:00” |
| check_out_time | DATETIME | - | Visitor check-out time. Example: “2025-06-10 16:00” |
| student_id | INT | 10 | Student identifier (FK). Example: 1001 |

6.0 CONCLUSIONS

In conclusion, this report has presented the analysis and design of a Hostel Management System by identifying the business rules, entities, and relationships involved in hostel operations. The system was analyzed to understand how students, rooms, payments, complaints, outings, visitors, and hostel staff interact with one another.

An Entity Relationship Diagram (ERD) was developed to visually represent these relationships and ensure clear understanding of data flow within the system. In addition, a data dictionary was produced to define the attributes, data types, and constraints for each entity. The design of this Hostel Management System aims to improve efficiency, data accuracy, and transparency in managing hostel activities compared to manual methods.

Overall, the proposed system provides a structured and organized approach to hostel management and can serve as a foundation for future system development and implementation.