SOFTWARE ENGINEERING

ITCI4

DEPARTMENT OF INFORMATION TECHNOLOGY

NETAJI SUBHAS UNIVERSITY OF TECHNOLOGY

HOTEL MANAGEMENT SYSTEM

SHIVANI GUPTA
2018UIT2586

AVANI SINGH
2018UIT2580

IT-2

CERTIFICATE

This is to certify that Shivani Gupta (2018UIT2586) and Avani Singh (2018UIT2580), students of IVth semester Bachelor of Engineering (B.E.) of Netaji Subhas University of Technology (NSUT), Delhi have successfully completed the project **HOSTEL MANAGEMENT SYSTEM** in course Software Engineering (ITC14) under the guidance of Mr.Deepak Kumar Sharma.

(Signature)

Mr.Deepak Kumar Sharma

ACKNOWLEDGEMENT

We express our sincere gratitude to all those who helped us in various capacities undertaking this project and making this report. We express our sense of gratitude to our respected teachers Mrs.Priti Bansal and Mr.Deepak Kumar Sharma.

INDEX

PROBLEM STATEMENT

This document will propose all features and procedures to develop the system.
The document especially contains details about objectives, scope limitation, process model **software requirements specification (SRS**), **Unified Modelling Language(UML),** Diagrams and **Data Dictionary.**

As the name specifies "**HOSTEL MANAGEMENT SYSTEM**" is a software developed for managing various activities in the hostel.

Development of a Software System which accurately and efficiently stores information about various entities and describes their relationship between the entities involved in management of the hostels.The database should contain information relating to all the students, the rooms, their guardians, the mess menu and staff members working in the hostel.The SRS and various diagrams used should clearly define relationships between the various entities of the Hostel.

Currently there is no automated system for management of Hostels, this documentation will help in the stages Of software development and further testing and maintenance of the system.
This will help build a user friendly GUI for the users and guide for each step of development.

Activities like registering or registering to leave the Hostel for any reason is done manually by entering the details in the register and it is not reliable since it is very hectic task to find any entry and it might get lost Even.
In this system all the details whether small or large are stored in a database and can be accessed any time.

A hostel management system is designed to provide a computerized process that is stress free, reliable and quick to both the students and the staff in charge of the registration and hostel management processes.

Warden and student's affairs officers will also be able to access and create student records with ease and regular updates of student profiles are enhanced when adopted. The usual practice of filing and procedures
involved in students' registration and hostel accommodation in universities were appraised for efficiency, economics and time management. The existing procedure was formed to be manually carried out. This is associated with inadequacies as more personnel are often required and a lot of times are wasted while the semester is on with lectures and other activities.

SOFTWARE REQUIREMENT SPECIFICATION

(1) Preface:

This document has been written according to the new version of SRS Software Requirements Specification (IEEE standard).

This document is the basic intended for any individual user, developer, tester, project manager or documentation writer that needs to understand the basic system architecture and its Specifications.

(2) Introduction:

The purpose of this SRS document is to write the functional and non-functional user or system requirements that represent the characteristics of **HOSTEL MANAGEMENT SYSTEM.**

The scope and limitation of this system is:

- Hostel Management System is designed for Hostel(like Schools, Universities, Colleges).
- There will be a pre-defined criteria for the Reserve to the hostels.
- If the students are found eligible then they are alloted to the hostel room.
- Can check the attested application forms of the students obtained from the internet and verify it with the student database.

(3) User Requirements Definition:

The user requirement for this system is to make the system fast, flexible, less prone to error, reduce expenses and save the time.

- ❏ Less human error
- ❏ Strength and strain of manual labor can be reduced
- ❏ High security
- ❏ Data redundancy can be avoided to some extent
- ❏ Data Consistency
- ❏ Easy to handle
- ❏ Easy data updating
- ❏ Easy record keeping
- ❏ Backup data can be easily generated

(4) System Models:

In this system we are using the Waterfall model to apply these ideas.Which will help us to separate each step and when we finish one phase the output of it is the input to the next phase. Also, we can go backwards if there is a new requirement or to apply any update.

(5) System Requirement Specification:

(5.1) Functional System Requirements:

This section gives a functional requirement that is applicable to the Hostel Management System.
There are sub modules in this phase:

- ★ Administrator Module
- ★ User Module
- ★ Hostel Module
- ★ Registration Module

The functionality of each module is as follows:

- **Administrator Module:**

The Administrator can :

1. Allot different students to the different hostels.
2. Vacate the students for the hostels.
3. Control the status of the fee payment.
4. Edit the details of the students and modify the student records.

- **User Module:**
    1. It allows the different users to access their profiles.
    2. He can view the student administration division of the different hostels and also view the notice boards.

- **Hostel Module:**

As the student's course is over they will vacate their rooms. So it is required for the administrator to remove their records from the database tables.This section includes the option for the room vacation and the deletion of the particular record from the database.

- **Registration Module:**

This section provides a form to the students which can be filled by them, and a copy of the filled page can be taken in the printed form.
This is later submitted to the Hostel authorities can be verified by them before allowing them to the respective hostels rooms.

(5.2) Non-Functional System Requirements:

(5.2.1) Non-Functional System Requirements:

Some Performance requirements identified is listed below:

- The database shall be able to accommodate a thousand records to store.
- The software shall support use of multiple users at a time.
- There are no other specific performance requirements that will affect development.

(5.2.2) Safety Requirements:

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.

(5.2.3) Security Requirements:

Some of the factors that are identical to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below.Keep specific log or history data sets

- ❖ Assign certain functions to different modules
- ❖ Restrict communications between some areas of the program
- ❖ Check data integrity for critical variables
- ❖ Later versions of the software will incorporate encryption techniques in the user/license authentication process.

(5.2.4) Hardware Requirements:

- Processor: Pentium
- RAM: 512MB
- Hard Disk: 1GB
- Keyboard
- Monitor

(5.2.5) Software Requirements:

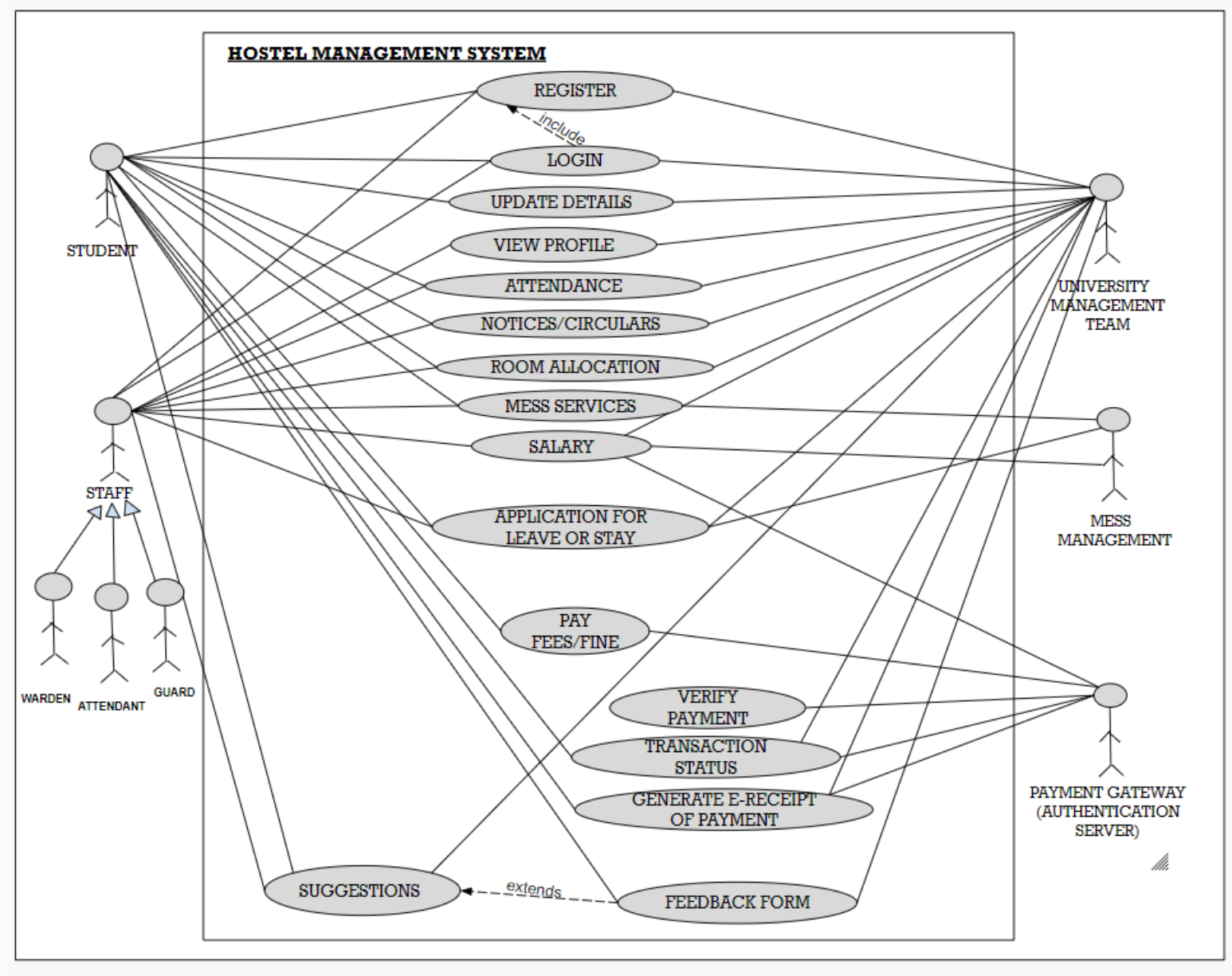
- Tool: Command Prompt
- Database: SQL
- Operating System: Microsoft Windows

(6) Definitions, Acronyms, Abbreviations

- ❖ IEEE : Institute of Electrical and Electronics Engineers
- ❖ UML : Unified Modelling Language
- ❖ DFD : Data Flow Diagram

USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



ACTORS IN THE DIAGRAM:

Students
Staff comprises of Wardens, Attendants and Guards
Admin of the System
Mess Employees
Payment Gateway

GENERALISATION:

Generalization is the process of extracting shared characteristics from two or more classes, and combining them into a generalized superclass. Shared characteristics can be attributes, associations, or methods.

This relationship is shown amongst Wardens, Attendants and Guards as Staff

BOUNDARY OF THE SYSTEM:

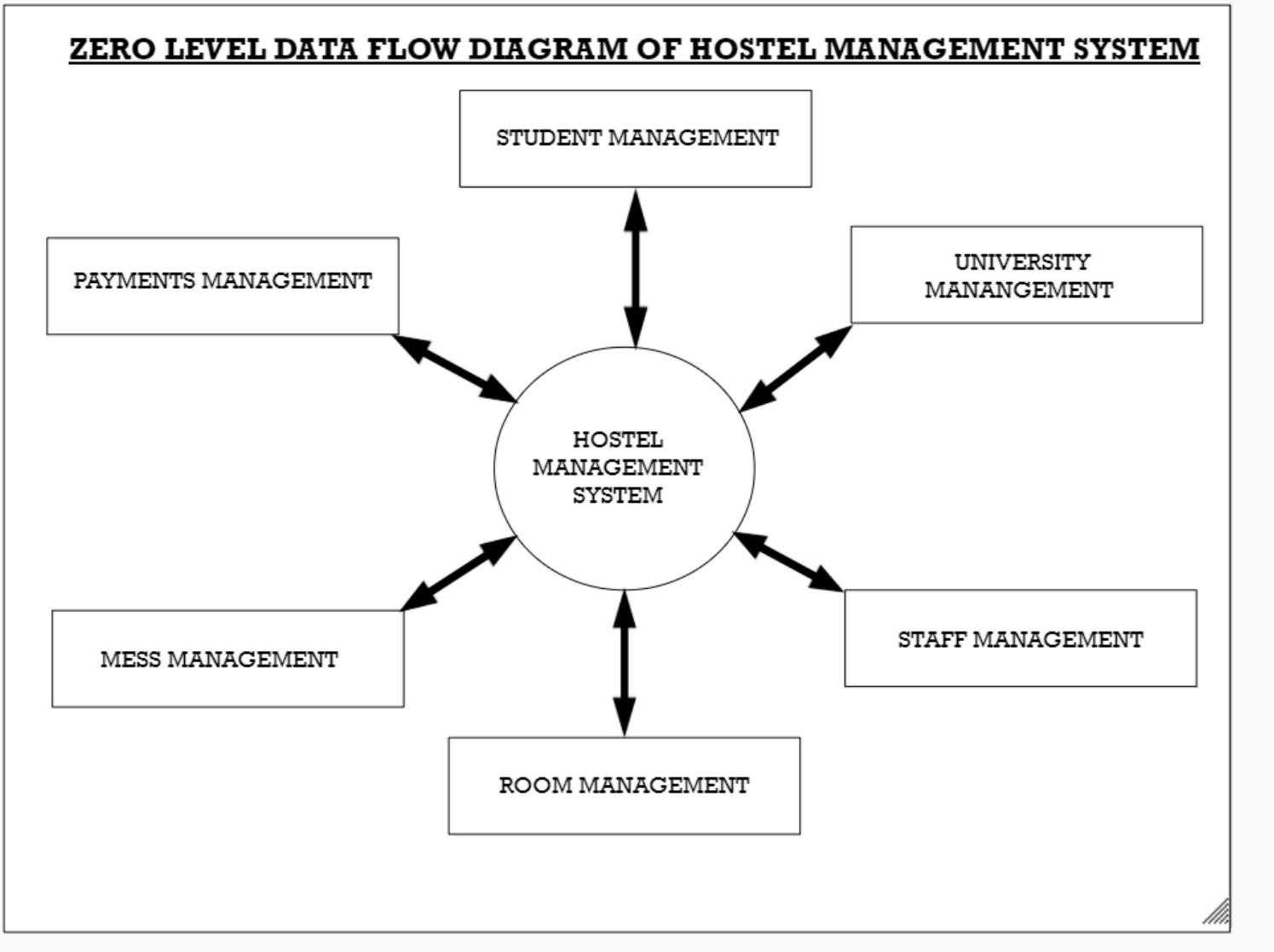
The system boundary is potentially the entire system as defined in the requirements documents.

COMMUNICATION LINK:

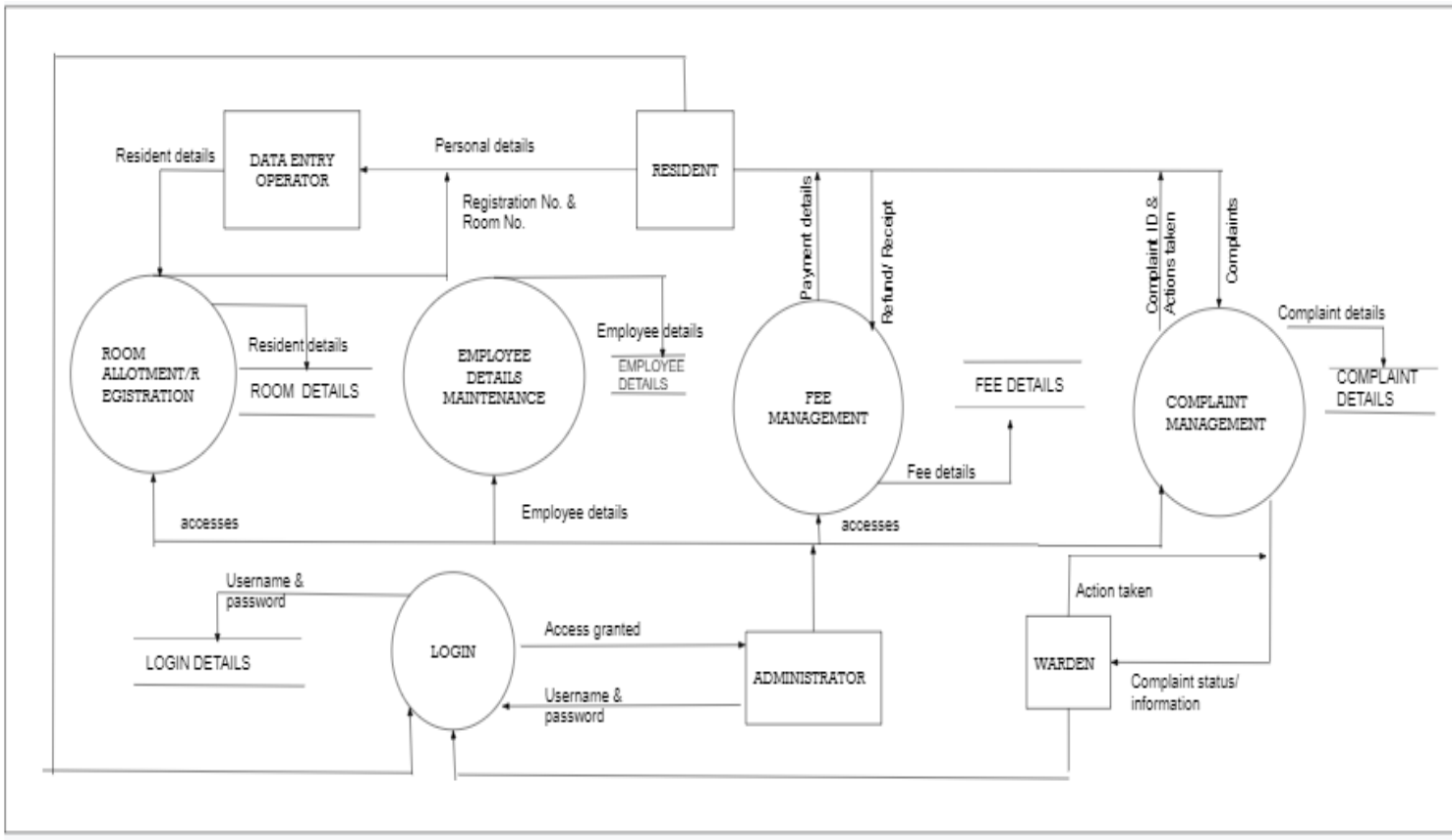
The participation of an actor is shown by connecting an actor to a use case by solid link.

CONTEXT DIAGRAM

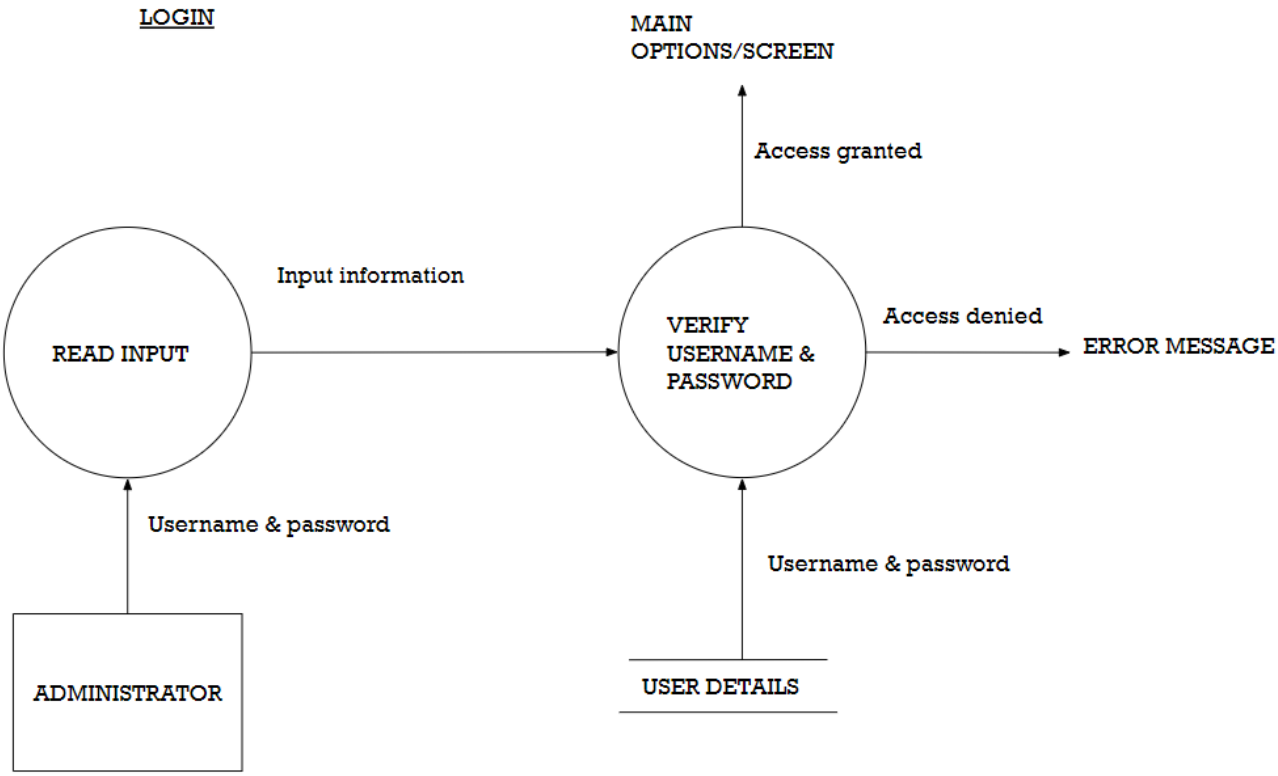
A **context diagram**, sometimes called a level 0 data-flow **diagram**, is drawn in order to define and clarify the boundaries of the software system. It identifies the flows of information between the system and external entities. The entire software system is shown as a single process.

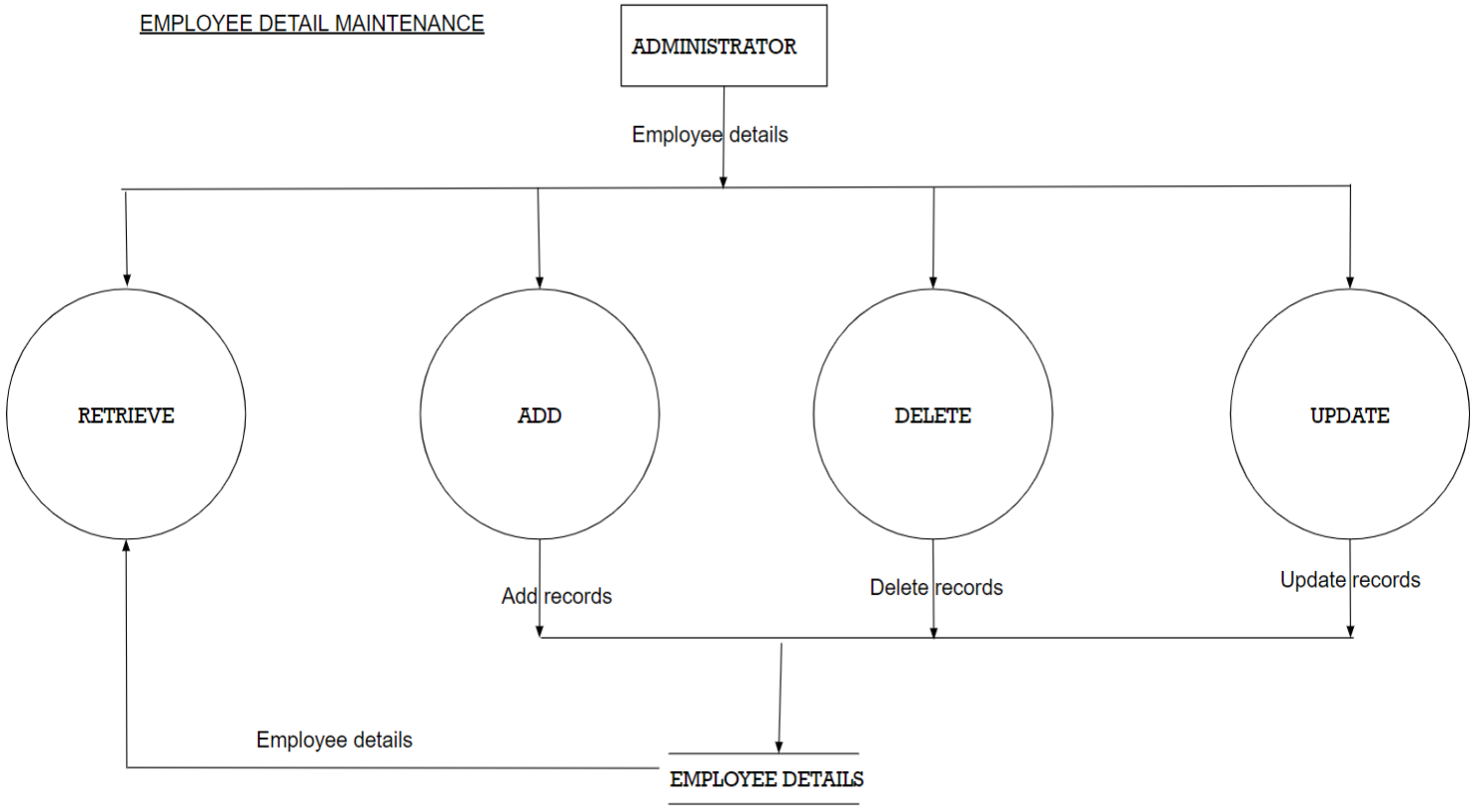


LEVEL 1 DATA FLOW DIAGRAM

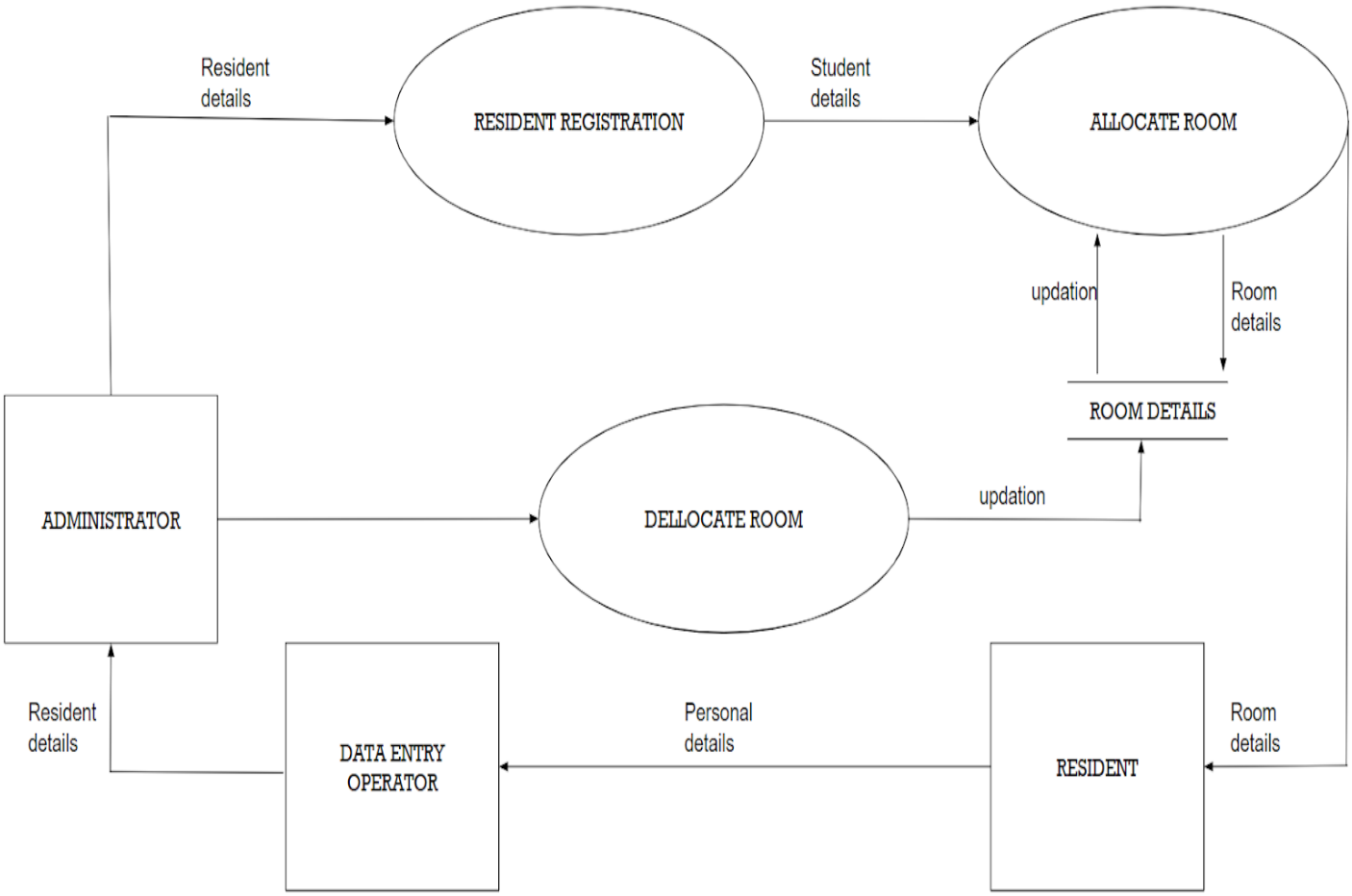


LEVEL 2 DATA FLOW DIAGRAM

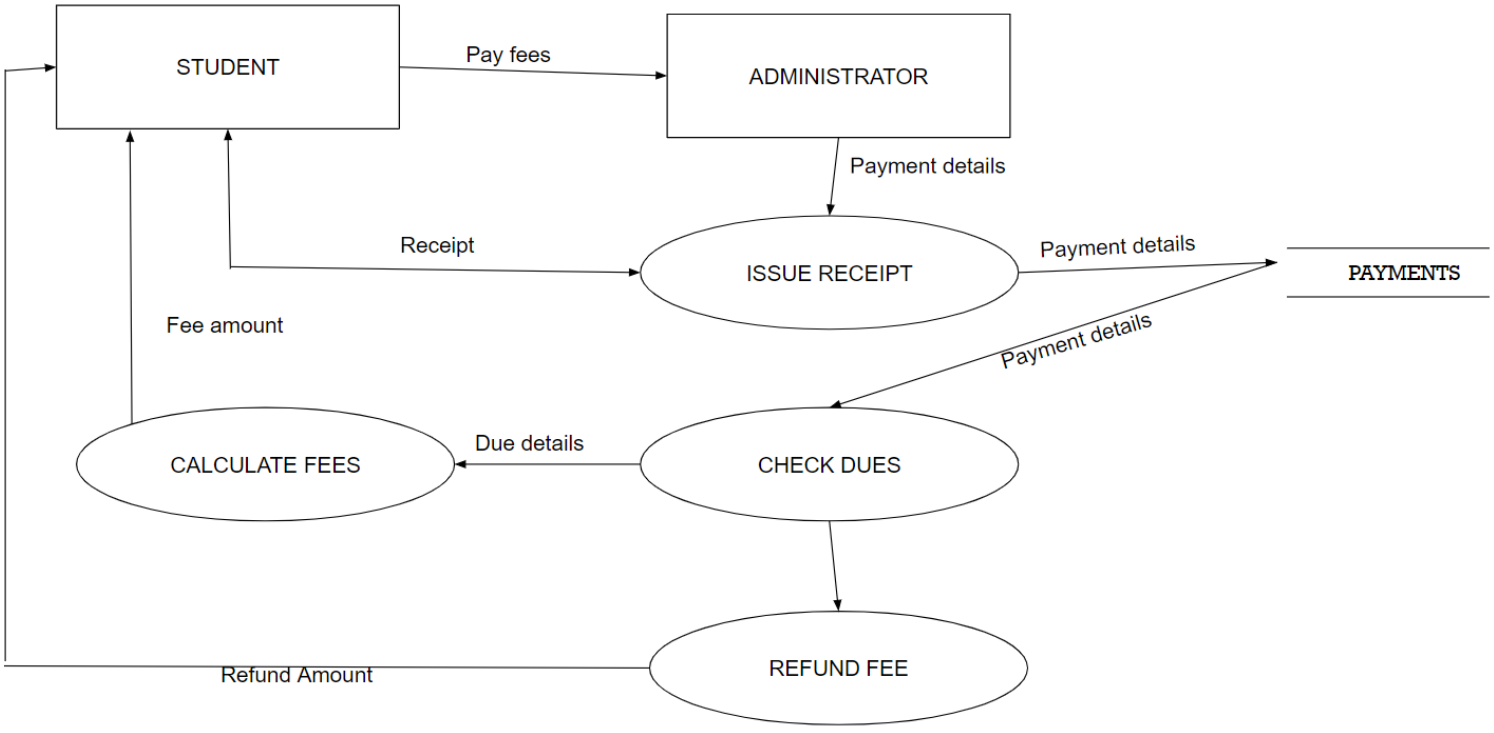




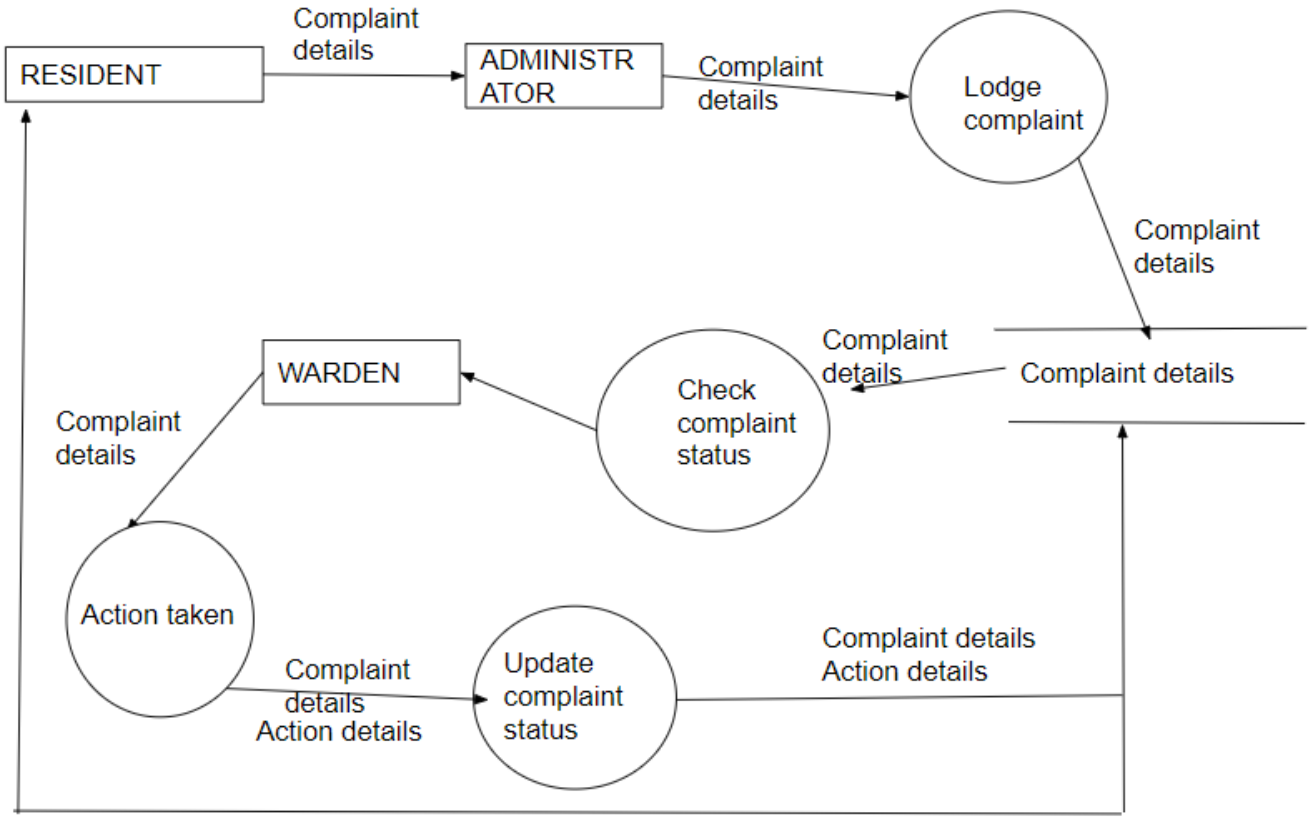
ROOM ALLOCATION-TERMINATION & RESIDENT REGISTRATION



FEE MANAGEMENT



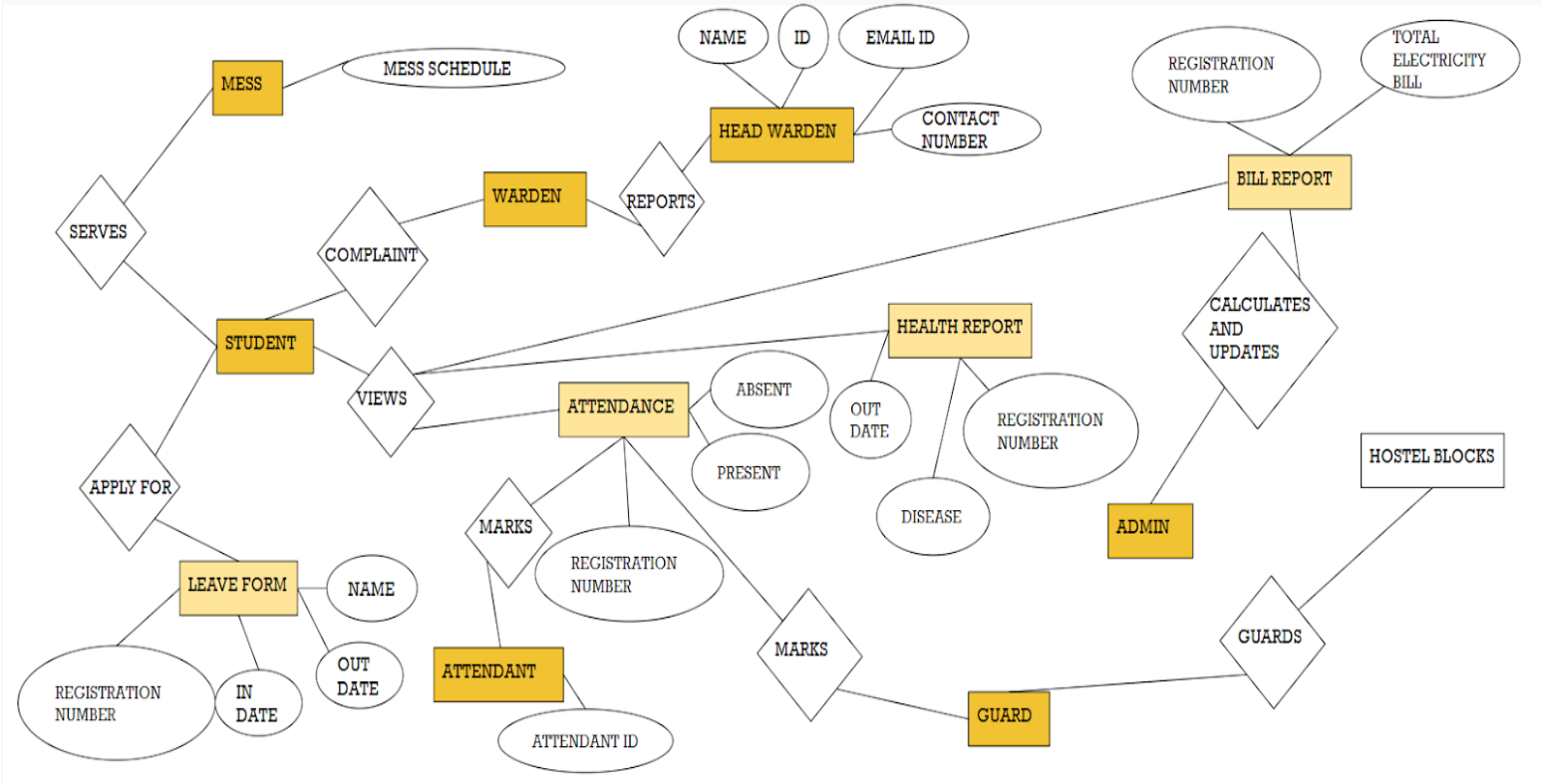
COMPLAINT MANAGEMENT



ENTITY-RELATIONSHIP DIAGRAM

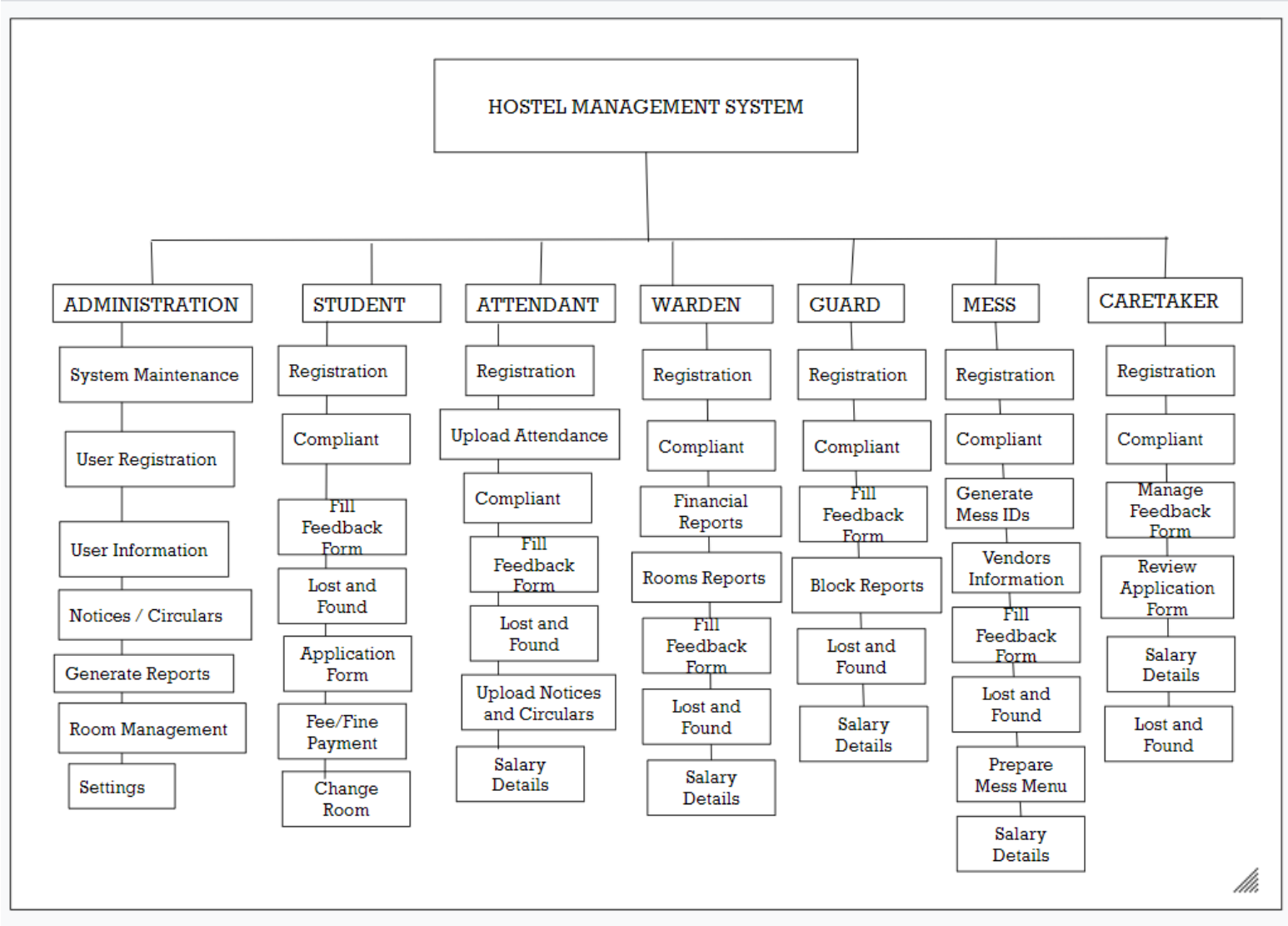
Entities in the System:

1. Student
2. Warden
3. Attendant
4. Guard
5. Mess employees
6. Admin of the system

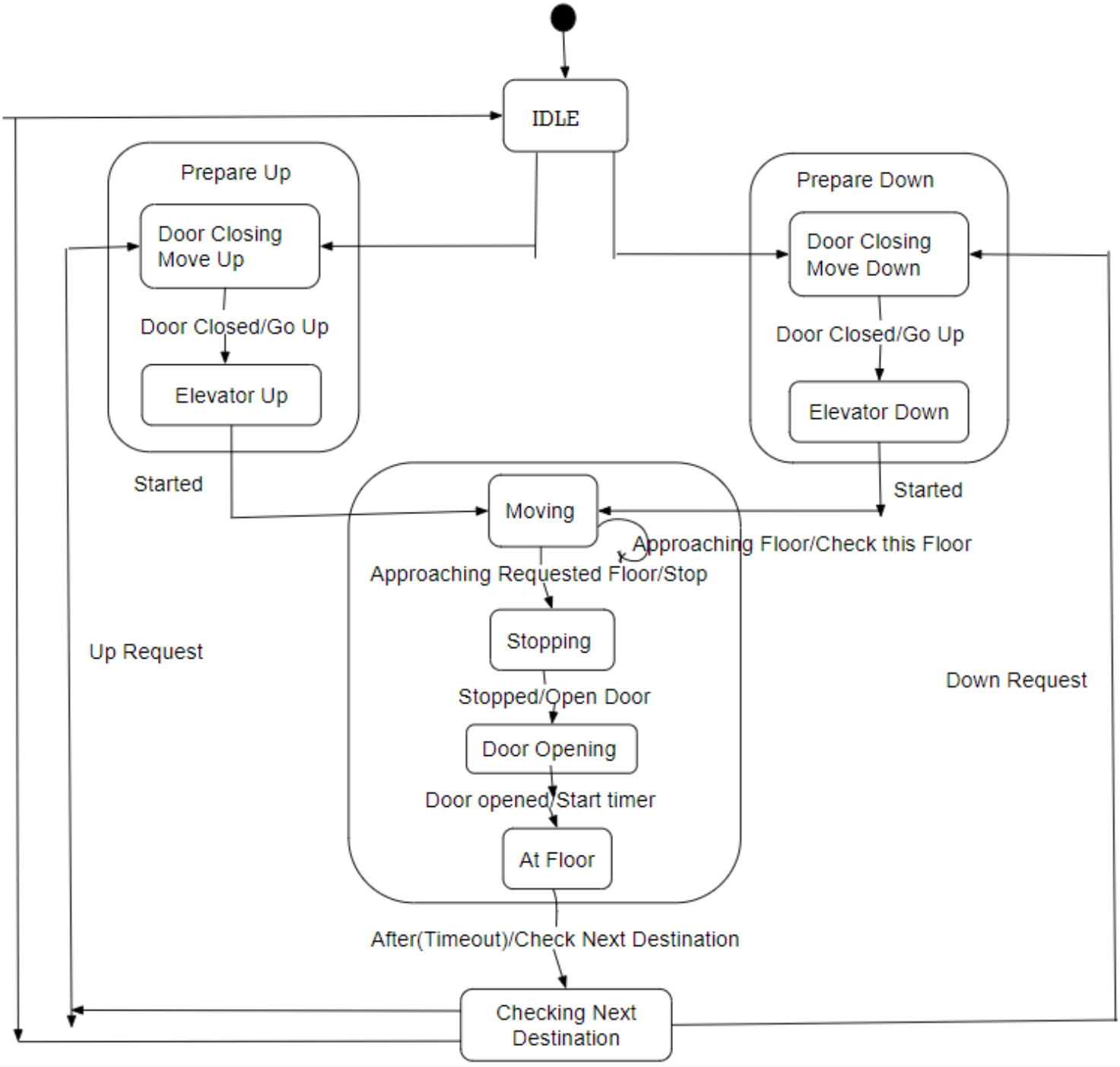


FUNCTIONAL DECOMPOSITION DIAGRAM

Functional decomposition corresponds to the various functional relationships as how the original complex business function was developed. It mainly focuses on how the overall functionality is developed and its interaction between various components.



STATE TRANSITION DIAGRAM for LIFT



DATA DICTIONARY

Data Dictionary is a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

1.
Name : Login
Aliases : None
Where used/ How used : Administrator wants to login
Description : Stores the login ID and Password

S.No.	Field Name	Data type	Description
1.	Login ID	Characters(30)	Login Name of the Administrator
2.	Password	Characters(20)	Password of the Administrator

2.
Name : Payments
Aliases : None
Where used/ How used :To store the details of Payments
Description : Stores the payment details

S.No.	Field Name	Data type	Description
1.	Registration Number	Integer(6)	Gives the registration number
2.	Type	Characters(15)	Type of the facility
3.	Transaction ID	Integer(10)	Stores the payment details of the resident
4.	Due Date	both characters and numbers(10)	Date on which the payment is due
5.	Receipt Number	Integer(40)	Receipt number given to the resident
6.	Date of payment	both characters and numbers(10)	Date on which the payment is made

3.
Name : Students
Aliases : None
Where used/ How used :Students registering for the semester
Description : Stores the details of all students of the hostel including past students

S.No.	Field Name	Data type	Description
1.	First Name	Characters(20)	First name of the resident
2.	Last Name	Characters(20)	Last name of the resident
3.	Address	Both characters and numbers(40)	Residential address of the resident
4.	Phone no.	Integer(10)	Phone number of the resident
5.	Email ID	Characters, numbers and symbols(30)	E-mail of the resident
6.	Gender	One character long(1)	Sex of the resident
7.	Occupation	Characters(20)	Occupation of the resident
8.	Registration Number	Integer(12)	Registration number of resident
9.	Date of Birth	Both characters and numbers(10)	Date of Birth of resident
10.	State	Characters(10)	State of the resident
11.	Pin code	Integer(10)	Pin code of state
12.	Country	Characters(20)	Country of the resident

5.
Name : Room
Aliases : None
Where used/ How used :Room allotted to residents
Description : Keeps record of the rooms that have been allotted and those which are vacant

S.No.	Field Name	Data type	Description
1.	Room No.	Integer(3)	Identifies a unique room no.
2.	Room Type	Characters(15)	Category of rooms
3.	Vacancy	Integer(3)	The number of people that Can be accommodated in the room apart from those already staying
4.	Phone No.	Integer(10)	Phone number of resident

6.
Name : Employee
Aliases : None
Where used/ How used :to store the details of an employee.
Description : Stores the details of all employees in the hostel.

S.No.	Field Name	Data type	Description
1.	First Name	Characters(20)	First name of the resident
2.	Last Name	Characters(20)	Last name of the resident
3.	Address	Both characters and numbers(40)	Residential address of the resident
4.	Phone no.	Integer(10)	Phone number of the resident
5.	Email ID	Characters, numbers and symbols(30)	E-mail of the resident
6.	Designation	Characters(15)	Designation of the employee
9.	Employee ID	Integer(6)	Id of the employee
10.	Date of birth	Both characters and numbers(40)	Date of birth of the employee
11.	Salary	Integer(5)	Salary of the employee

7.
Name : Fees
Aliases : None
Where used/ How used :To calculate the fees of a room
Description : Stores the amount of the fees for different fee heads like Security, Mess Charges etc.

S.No.	Field Name	Data type	Description
1.	Fees type	Characters(15)	Type of the fees
2.	Charges	Integer(5)	Monthly amount to be paid
3.	Frequency	Characters(15)	The frequency at which the fees is collected

8.
Name : Complaint
Aliases : None
Where used/ How used :To store the complaint details
Description : Stores the details of all complaints in the hostel

S.No.	Field Name	Data type	Description
1.	Complaint date	Both characters and numbers(40)	Date of issue of complaint
2.	Particulars	Characters(60)	Details of the complaint
3.	Status	Characters(10)	Current status of the compliant
4.	Registration Number	Integer(6)	Registration number of the resident who made the complaint
5.	Compliant ID	Integer(5)	Complaint identification number
6.	Room No.	Integer(3)	The room number associated with the complaint

9.

Name : Local Guardian
Aliases : None
Where used/ How used :To store the details of local guardian
Description : Stores the record of the local guardians associated with each resident

S.No.	Field Name	Data type	Description
1.	Registration number	Integer(6)	Registration number of the student
2.	First Name	Characters(20)	First name of the local guardian of the student
3.	Last Name	Characters(20)	Last name of the local guardian of the student
4.	Email ID	Both characters and numbers(40)	Email id of the local guardian of the student
5.	Address	Both characters and numbers(40)	Address of the local guardian of the student
6.	State	Characters(10)	State as a part of address
7.	Telephone No.	Integer(10)	Phone no. of the local guardian of the student

10.
Name : Organization
Aliases : None
Where used/ How used :To store the details of Organization.
Description : Stores the names, contact details of the organisation

S.No.	Field Name	Data type	Description
1.	Registration no.	Integer(6)	Gives the registration number
2.	Name	Characters(20)	Name of the organisation
3.	Email ID	Both characters and numbers(40)	Email ID of the organisation
4.	Address	Both characters and numbers(40)	Address of the organisation
5.	Telephone No.	Integer(10)	Phone Number of the organisation

LINKS AND REFERENCES OF DIAGRAMS MADE

Use Case Diagram

Entity-Relationship Diagram

Level 0 Data Flow Diagram(Context Diagram)

Level 1 Data Flow Diagram

Functional Decomposition Diagram

State Transition Diagram for Lift

Level 2 Data Flow Diagrams:

Complaint Management

Fee Management

Employee Management

Login Management

Room Management