OOAD -
MINI PROJECT DOCUMENTATION

for

HOSTEL MANAGEMENT PORTAL

Team Members

Hemanent S - 195001038

Jayakrishnan S V – 195001040

Jayaraja S K – 195001041

Karthik A – 195001046

SECTION 1

PROBLEM STATEMENT

DESCRIPTION

In today's modern world, although a lot of things have been modernized/digitalised, still most of the activities carried out by the students at hostel and mess involves a lot of paperwork (or) they have to spend a lot of their time waiting.

PROBLEMS FACED CURRENTLY

1. **MESS BILL PAYMENT AND CANCELLATION**
   In order to pay the mess bill, the student has to first take a look at the bill amount that has been posted on the notice board and later has to pay the bill in the form of cash in person. Also, to cancel the mess bill, students have to wait in a very long queue and different notebooks are maintained for students of different years.

2. **OUTPASS / LEAVE FORM**
   If a student needs an outpass to leave the campus, then he should meet the Hostel Warden in person and receive the leave form, where the date and time are filled and signed by the Warden after verifying the student's identity.

   When a hosteller wishes to leave the campus on holidays, he should get an out pass from the Hostel Warden by meeting him in-person, where the Warden would verify the student's identity and grant him the permission to leave by filling his details in a leave form, which could be used by the student to leave the campus.

3. **COMPLAINT REGISTRATION**
   In order to register a complaint on any available amenities at the hostel, the student has to meet the supervisor in person to explain the problem and only then will be allowed to file a complaint in the log book.

PROPOSED SOLUTION

A web application is to be designed and developed, which integrates the Hostel and Mess activities at one place, to address the above-mentioned issues.

1. The web app will have features for the students to pay and cancel their bill from any place. The mess in-charge would be able to upload the duellist as an Excel file and view the list of students who have paid their dues and the students who haven't. Also, the mess in charge would be able to view the list of students who have cancelled their mess and cross check it with their attendance system.
2. Students would be able to request for an outpass by entering their ID Number, date of leave and the reason, where on receiving confirmation from the Warden, the leave form will be automatically generated and a copy would be sent to the security personnel's.
3. Students could register their complaints regarding various issues like Electrical, Carpentry and Plumbing. This data can be viewed by the Warden and respective authorities which facilitates them to take actions effectively.
4. A general announcement page regarding the Hostel and Mess is to be added where the Hostel Warden or Mess In-charge could post any important information.
5. A Forum can be added for students to connect among themselves.

SECTION 2

SOFTWARE REQUIREMENTS SPECIFICATION

1. INTRODUCTION

Paying mess bills, cancelling mess, Filing Complaints or Receiving leave form at Hostel and Mess is currently done manually. In order to pay the mess bill, the student has to first take a look at the bill amount that has been posted on the notice board and later pay the bill in the form of cash in person. Also, to cancel the mess bill, students have to wait in a very long queue and different notebooks are maintained for students of different years. If a student needs an out pass to leave the campus, then he should meet the Hostel Warden in person and receive the leave form, where the warden fills in the date and time and acknowledges the same after verifying the student's identity. In order to register a complaint on any available amenities at the hostel, the student has to meet the supervisor in person to explain the problem and only then will be allowed to file a complaint in the log book.

To overcome all these issues, a web application is to be developed. This system would be used by the students to pay and cancel mess bills, whereas the mess incharge would be able to upload the duellist at the start of every month and view the payment status of each student. The system would also be used by the students to file a complaint on any available amenities and also request for a leave form where the Warden would verify and accept the same. The purpose of the document is to analyse and elaborate the needs and features of the Hostel Management System. (The details of what all are the needs of the Hostel Management System and if it fulfils these needs are detailed in the use-case and supplementary specifications.)

1.1 Purpose

The purpose of the Software Requirements Specification (SRS) document is to describe the behaviour of the Hostel Management System. Specification defines and describes the operations, interfaces, performance, and quality assurance requirements of the Hostel Management System. The document also describes the non-functional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed, and other factors necessary to provide a complete and comprehensive description of the requirements for the software. The Software Requirements Specification (SRS) captures the complete software requirements for the system, or a portion of the system.

1.2 Scope

The Software Requirements Specification captures all the requirements in a single document. The Hostel Management System provides a way for the students to pay or cancel the mess bills, register any complaint, request for an e-leave form and also accommodates many other facilities. The Hostel Management System is supposed to have the following features.
- The system provides a logon facility to the users.
- The system allows the members to see their account details and change the same, if required.
- The system allows the students to pay their mess dues with the help of an external payment gateway
- The system allows the students to cancel their mess for the requested number of days.
- The system allows the students to register any complaint.

- The system allows the students to request for an e-leave form.
- The system allows the mess incharge to upload an excel file, containing the bill amount.
- The system allows the mess incharge to view the list of students who have paid their dues and the students who haven't separately. Also, the incharge would be able to view the list of students who have sent requests to cancel their mess bill.
- The system allows the Hostel Warden to accept or reject the leave form requests received from students
- The system has a general announcements page for the students where any important announcements posted by the Hostel Warden could be viewed here.

The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system is based on the level up to which the features described in this document are implemented in the system.

1.5   Overview

The SRS will provide a detailed description of the Hostel Management System. This document will provide the outline of the requirements, overview of the characteristics and constraints of the system.

1.5.1 Section - 2: This section of the SRS will provide the general factors that affect the product and its requirements. It provides the background for those requirements. The items such as product perspective, product function, user characteristics, constraints, assumptions and dependencies and requirements subsets are described in this section.

1.5.2 Section - 3: This section of SRS contains all the software requirements mentioned in section 2 in detail sufficient enough to enable designers to design the system to satisfy the requirements and testers to test if the system satisfies those requirements.

2. OVERALL DESCRIPTION

- **Product Perspective**
  The Hostel Management System to be developed, benefits greatly the hostel students, mess incharge and warden. The system provides a platform to pay or cancel mess bills, file complaints regarding hostel amenities and request for leave forms.

  The complete overview of the system is as shown in the overview diagram below:

  The product to be developed has interactions with the users: Hostel Students, Mess Incharge and Hostel Warden.
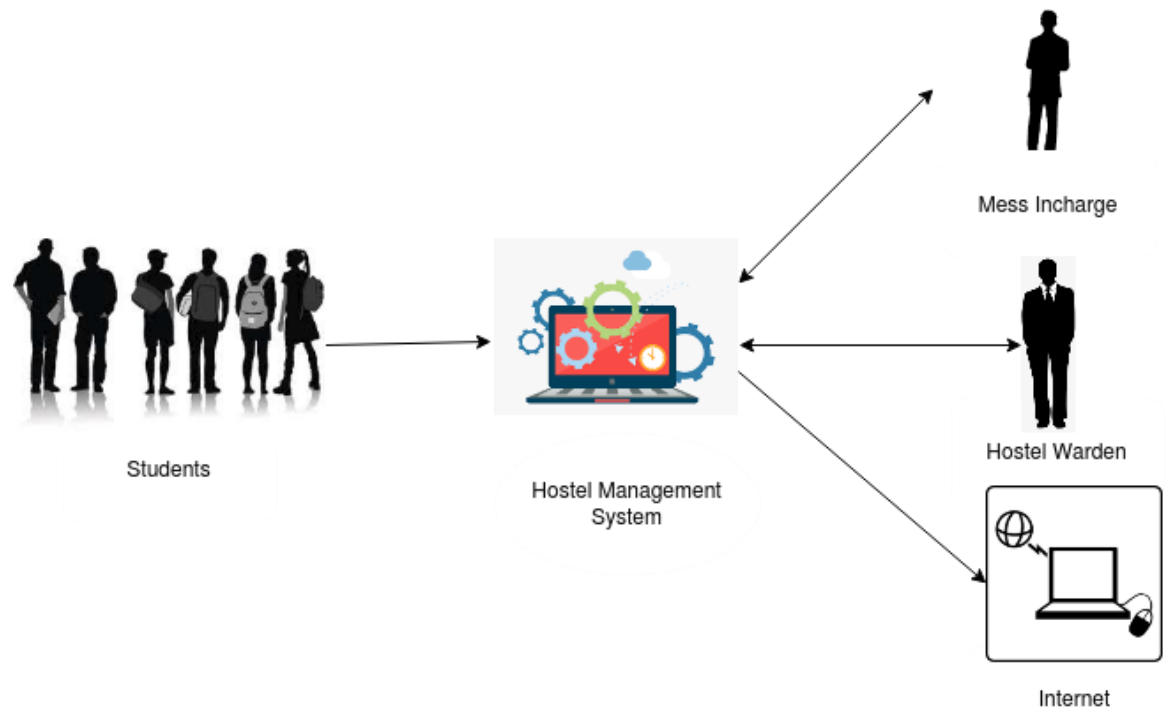
(Overview of the proposed system)

- **Product Functions**
  The Product functions are more or less the same as described in the product perspective. The functions of the system include the system providing different types of services based on the types of users.
    o The students should be able to view their updated mess dues, leave requests and complaints filed.
    o The students are given a provision to check their account information and change the same at any time.
    o The mess incharge should be able to view the mess bill status of students. Also, he should be able to view the list of students who have canceled their mess bills.
    o The Warden should be able to view the list of requests for leave forms and the complaints filed.
    o Special Admin access is given to the warden to add, remove or modify the student database.

- **User Characteristics**
  The users of the system are students, mess incharge and Hostel Warden. Special administrator access is given to the Hostel Warden to maintain the system. The users are assumed to have basic knowledge of the computers and Internet browsing. The administrators of the system are to have more knowledge of the internals of the system and are able to rectify the small problems that may arise due to disk crashes, power failures and other catastrophes to maintain the system. The proper user interface, users manual

and online help must be sufficient to educate the users on how to use the system without any problems.

- **Constraints**
  - The information of all the users must be stored in a database that is accessible by the Hostel Management System.
  - The Hostel Management System is connected to the computer in which the system will be deployed and is running at all active hours.
  - The users access the Hostel Management System from any device that has Internet browsing capabilities and an Internet connection.
  - The users must have their correct usernames and passwords to enter into the Hostel Management System.
  - The warden must accept or reject the e-leave forms within 24 hours.

- **Assumptions and dependencies**
  - The users have sufficient knowledge of computers.
  - The computer in which the system is to be deployed should have Internet connection and Internet server capabilities.
  - The users know the English language, as the user interface will be provided in English.
  - The product can access the hostel student database.

3. SPECIFIC REQUIREMENTS

3.1 Functionality

3.1.1 Logon Capabilities
The system shall provide the users with logon capabilities.

3.1.2 Mobile Devices
The Hostel Management System is also supported on mobile devices such as cell phones.

3.2 Usability

- The system shall allow the users to access the system from the Internet using HTML or its derivative technologies. The system uses a web browser as an interface.
- Since all users are familiar with the general usage of browsers, no specific training is required.
- The system is user friendly and self-explanatory.

3.3 Reliability

The system has to be very reliable due to the importance of data

3.3.1 Availability

The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.

3.3.2 Mean Time Between Failures (MTBF)

The system will be developed in such a way that it may fail once in a year.

3.3.3 Meantime to Repair (MTTR)

Even if the system fails, the system will be recovered back up within an hour or less.

3.3.4 Accuracy

The accuracy of the system is limited by the accuracy of the speed at which the users use the system.

3.3.5 Maximum Bugs or Defect Rate

Not specified.

3.3.6 Access Reliability

The system shall provide 100% access reliability.

3.4 Performance

3.4.1 Response Time

The information is refreshed every two minutes. The access time for a mobile device should be less than a minute. The system shall respond to the member in not less than two seconds from the time of the request submission. The system shall be allowed to take more time when doing large processing jobs.

3.4.2 Administrator Response

The system shall take as little time as possible to provide service to the administrator.

3.4.3 Throughput

The number of transactions is directly dependent on the number of users, the users may be students, mess incharge and the hostel warden.

3.4.4 Capacity

The system is capable of handling 250 users at a time.

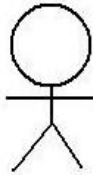### 3.4.5 Resource Utilization

The resources are modified according to the students who wish to make the payments or request for a leave form.
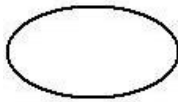
SECTION 3

USE CASE MODEL

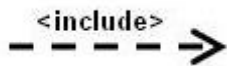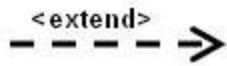**NOTATION**

1. Actors

2. Use Cases

3. Generalization


4. Includes

5. Extends

IDENTIFICATION OF ACTORS

- **Primary Actors**
  - ❖ Student
  - ❖ Warden
  - ❖ Mess In-Charge

IDENTIFICATION OF SCENARIOS

1. Login
2. Verify credentials
3. Display error message on invalid login
4. Warden
   - Add new user
   - View list of complaints filed
   - View processed leave requests
   - View current leave request
5. Students
   - Pay mess dues
   - File a complaint
   - Cancel mess bill
   - Request for gate pass
6. Mess In charge
   - View payment status of students
   - View bill cancellations
   - Upload student's attendance biometric data

FULLY DRESSED USE CASE DIAGRAM
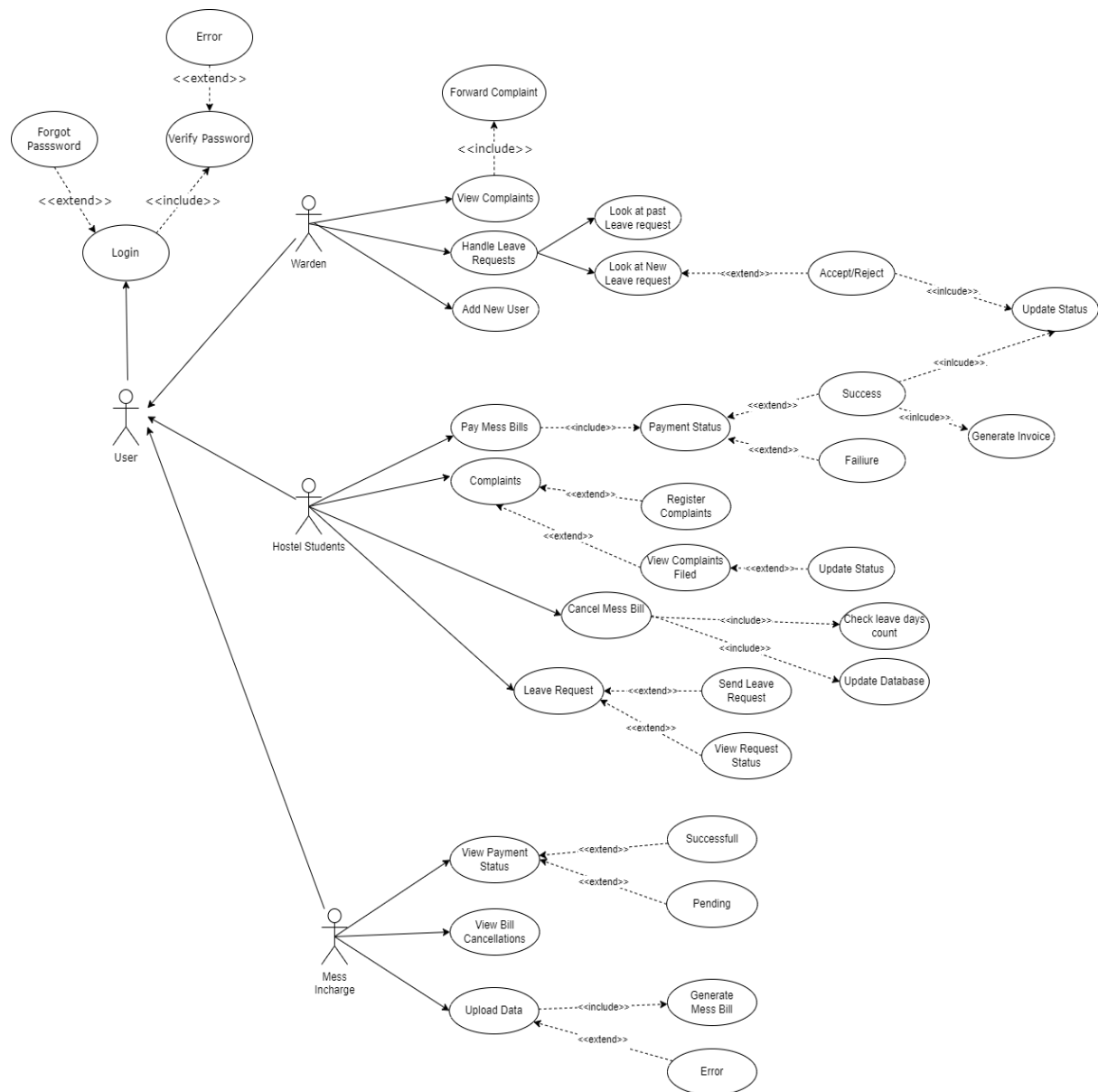
MAIN SUCCESS SCENARIOS

1. **Login**
   ● To perform any of kind of action in any of the below listed success scenarios, the user has to first log into the portal

2. **Mess Bill Payment**
   ● Mess in charge uploads the student's biometric attendance data at the start of every month
   ● Students can view the list of pending mess dues
   ● Students can pay the mess due of each month by clicking on the payment button
   ● After each payment, the status would be recorded and updated in the database
   ● Mess in charge can view the list of students who have made their dues and the ones who haven't

3. **Complaint Registration**
   ● Students clicks on the Complaint Registration to file a complaint on basic amenities by filling the form and submitting the same
   ● Hostel Warden can view the list of complaints and forward the same to an external person
   ● Students can view the complaints filed by them and can update the status of each one if it has been resolved

4. **Leave Request (Gate pass)**
   ● Students can send a leave request to the Hostel Warden to leave the campus by entering the required details
   ● Hostel Warden can view the list of leave requests and can accept the same by verifying the details
   ● The status of the leave request would be updated in the student's side
   ● Hostel Warden can also view the list of processed leave requests

5. **Mess Bill Cancellation**
   ● Students can cancel the mess bill by entering the from and to date
   ● Mess in charge can view the list of students who have canceled their mess bills in a particular month

FREQUENT ALTERNATE SCENARIOS

1. **Leave Request can be Rejected**
   - When a student requests for a gate pass during College Hours or on Working days, the request would be rejected by the Hostel Warden unless he has uploaded the permission letter from Department HoD

2. **Uploading Biometric Data**
   - When the Mess in charge uploads the biometric data more than once in a month, he would be notified that it has already been uploaded and can the same cannot be performed again

3. **Mess Bill Cancellation**
   - When a student enters the start and end date that is less than 2 days, then he would not be allowed to cancel the mess bill
   - Also, when the total no. Of days on which the mess bill has been canceled exceeds more than 25 days in a month, then he/she would not be allowed to do any further cancellations in that month

SUB FUNCTIONS

1. **Verify Password**
   - When a user enters invalid credentials for logging in to the portal, an error message would be displayed
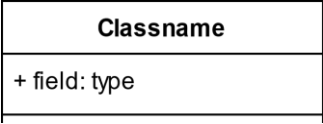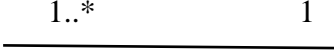
2. **Forgot Password**
   - User could generate a new password for his account when he doesn't remember the credentials

SECTION 4

UML DIAGRAMS
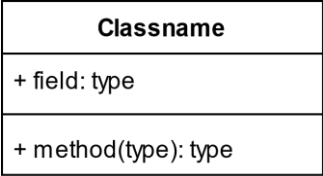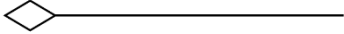
NOTATIONS FOR DOMAIN MODEL

Notation	Type	Explanation
Classname + field: type	Conceptual Class	An object in the problem domain
———————	Association	Relationship between objects or the classes
1..*　　　　　1	Multiplicity of the Role	Defines how many instances of a class A can be associated with one instance of a class B
———————→	Generalization	The child class inherits the characteristics of base class

NOTATIONS FOR CLASS DIAGRAM

Notation	Type	Explanation
Classname + field: type + method(type): type	Conceptual Class	An object in the problem domain along with its attributes
◇———————	Aggregation	It is a relationship between two classes where one class is a part of another where one class is meaningful even without the aggregate
◆———————	Composition	It is a relationship between two classes where one class is a part of another where one class is not meaningful without the other
——————→	Generalization	The child class inherits the characteristics of base class
———————	Association	Relationship between objects or the classes
1..* 1 ———————	Multiplicity of the Role	Defines how many instances of a class A can be associated with one instance of a class B

IDENTIFICATION OF CLASSES

a) **Identification of Conceptual Classes:**

Conceptual class category	Conceptual Class
transaction	Payment
roles of people	User ● Hostel Students ● Warden ● Mess In-charge
events	Bill Payment and Cancellations, Request Leave Form, Complaint Registration
records of finance, work, contracts, legal matters	Receipt, Leave Form
computer system	Management System, Mess System, Hostel System

b) **Identification of Noun Phrases:**
   Initial List of Noun Phrases:
   - Pay Bills
   - Cancel Bills
   - Mess in-charge
   - Upload due list
   - Students
   - Attendance System
   - ID Number
   - Hostel Warden
   - Out pass Request
   - Leave Form
   - Register Complaints
   - Electrical
   - Carpentry
   - Plumbing
   - General Announcement
   - Hostel
   - Mess
   - Forum

Final List of Noun Phrases:
- Pay Bills
- Cancel Bills
- Mess in-charge
- Students
- Hostel Warden
- Out pass Request
- Register Complaints

IDENTIFICATION OF ASSOCIATIONS

a) **Association Identification:**

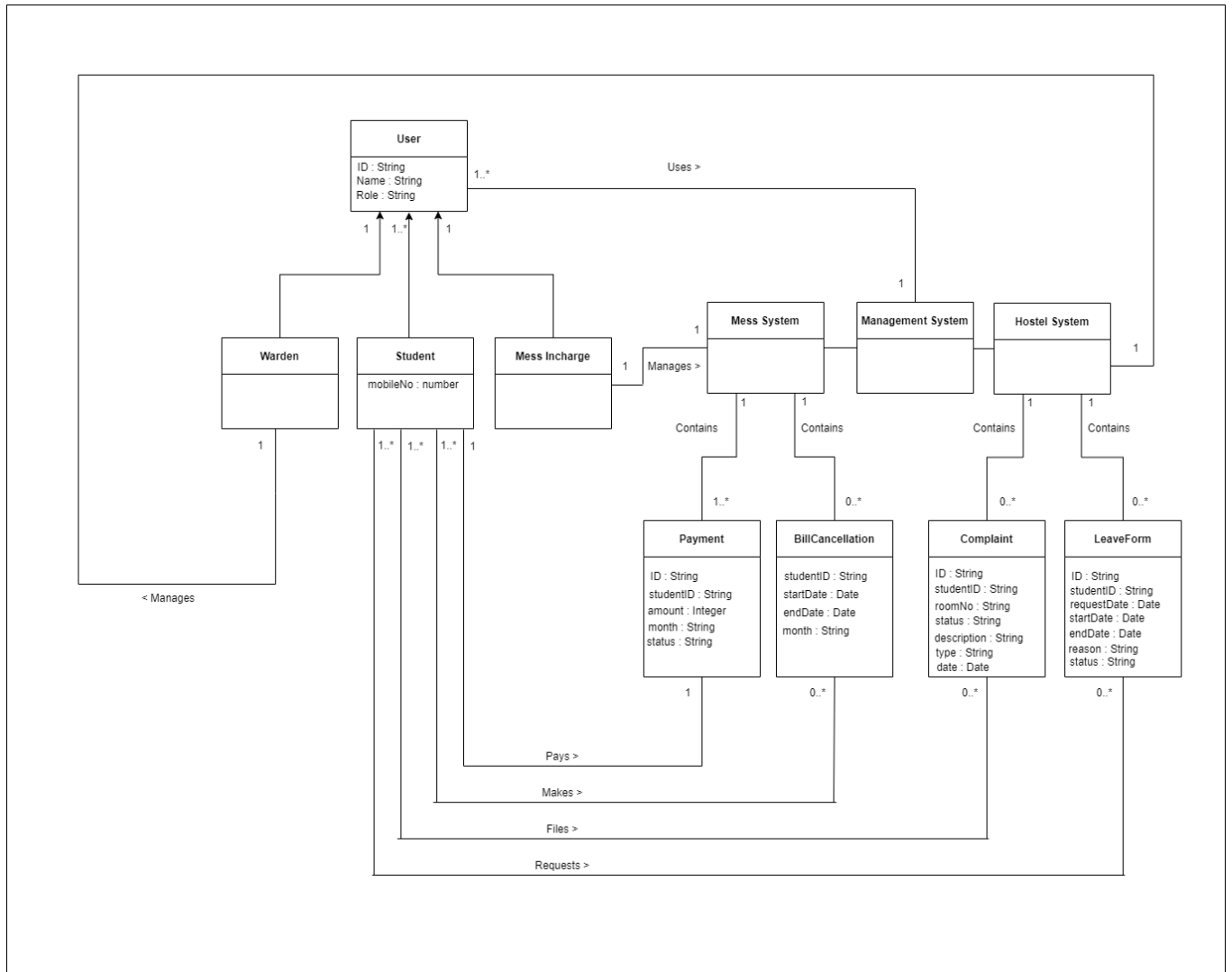| Category	Association	Between
A is a logically contained in *B*	Contains	Mess System, Hostel System and Management System
A is a logically contained in *B*	Contains	Bill Payment, Cancellation and Mess System
A is a logically contained in *B*	Contains	Complaint Registration, Leave Requests and HostelSystem
A manages *B*	Manages	Mess in-charge and Mess System
A manages *B*	Manages	Hostel Warden and Hostel System
A is a transaction related to *B*	Pays	Student and Bill Payment
A is related to *B*	Makes	Students and Bill Cancellation
A communicates with *B*	Requests	Student and Leave Requests
A is related to *B*	Files	Student and Complaint Registration

b) Definition of Association:

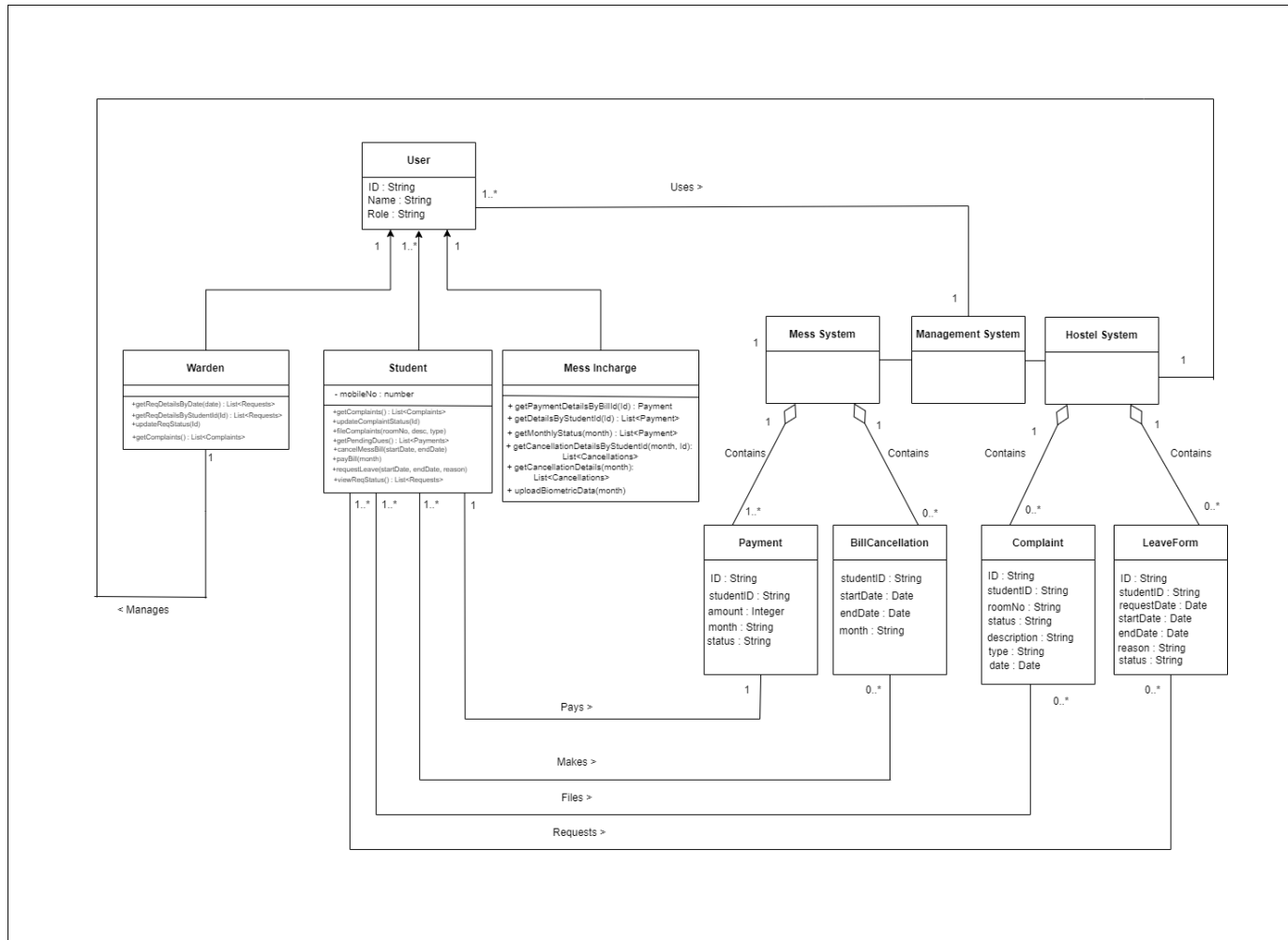| Association	Explanation
Generalization	The Generalization association ("is a") is the relationship between the base class that is named as "superclass" or "parent" and the specific class that is named as "subclass" or "child"
Aggregation	An aggregation is a relationship between classes which says one class 'is a part of' another class. In aggregation, the part (constituent) is meaningful without the whole (aggregate)
Composition	An aggregation is a relationship between classes which says one class 'is a part of' another class. In composition, the part (component) is not meaningful without the whole (container)

c) Multiplicity:

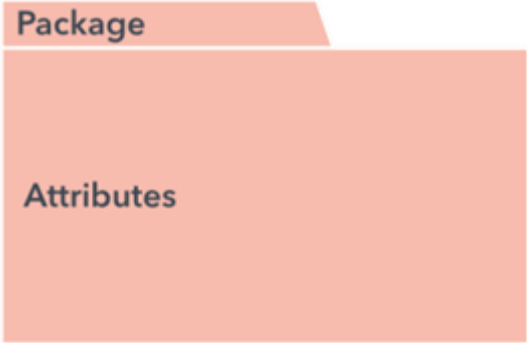| Relationship	Multiplicity
Mess Incharge - Mess System	One Mess Incharge manages One Mess System
Warden - Hostel System	One Warden manages One Hostel System
Student - Bill Payment	One Student pays One Bill
Student - Bill Cancellation	One or many Students Makes Zero or more Bill Cancellations
Student - Complaint Registration	One or many Students Files Zero or more Complaints
Student - Leave Requests	One or many Students requests Zero or more Leave Requests
User - Management System	One or many users uses One Management System
Mess System - Bill Payment	One Mess System contains one or many Bill payments
Mess System - Bill Cancellation	One Mess System contains zero or more Bill cancellation
Hostel System - Complaint Registration	One Hostel System contains zero or more Complaint Registration
Hostel System - Leave Requests	One Hostel System contains zero or more Leave Requests

DOMAIN MODEL

User

ID : String
Name : String
Role : String

Uses >

1..*

1   1..*   1

Warden

Student

mobileNo : number

Mess Incharge

Mess System

Management System

Hostel System

Manages >

1   1

1

Contains   Contains   Contains   Contains

1   1   1   1

< Manages

1   1..*   1..*   1..*   1

1..*   0..*   0..*   0..*

Payment

ID : String
studentID : String
amount : Integer
month : String
status : String

BillCancellation

studentID : String
startDate : Date
endDate : Date
month : String

Complaint

ID : String
studentID : String
roomNo : String
status : String
description : String
type : String
date : Date

LeaveForm

ID : String
studentID : String
requestDate : Date
startDate : Date
endDate : Date
reason : String
status : String

1   0..*   0..*   0..*

Pays >

Makes >

Files >

Requests >

CLASS DIAGRAM

User

ID : String
Name : String
Role : String

Uses >

1..*

1

1..*

1

1

Mess System

Management System

Hostel System

1

1

1

1

1

1

Warden

+getReqDetailsByDate(date) : List<Requests>
+getReqDetailsByStudentId(id) : List<Requests>
+updateReqStatus(id)
+getComplaints() : List<Complaints>

1

Student

- mobileNo : number

+getComplaints() : List<Complaints>
+updateComplaintStatus(id)
+fileComplaints(roomNo, desc, type)
+getPendingDues() : List<Payments>
+cancelMessBill(startDate, endDate)
+payBill(month)
+requestLeave(startDate, endDate, reason)
+viewReqStatus() : List<Requests>

Mess Incharge

+ getPaymentDetailsByBillId(Id) : Payment
+ getDetailsByStudentId(id) : List<Payment>
+ getMonthlyStatus(month) : List<Payment>
+ getCancellationDetailsByStudentId(month, Id): List<Cancellations>
+ getCancellationDetails(month): List<Cancellations>
+ uploadBiometricData(month)

Contains

Contains

Contains

Contains

1..*

0..*

0..*

0..*

< Manages

1..*   1..*   1..*   1

Payment

ID : String
studentID : String
amount : Integer
month : String
status : String

BillCancellation

studentID : String
startDate : Date
endDate : Date
month : String

Complaint

ID : String
studentID : String
roomNo : String
status : String
description : String
type : String
date : Date

LeaveForm

ID : String
studentID : String
requestDate : Date
startDate : Date
endDate : Date
reason : String
status : String

1

0..*

0..*

0..*

Pays >

Makes >

Files >

Requests >

SECTION 5

SEQUENCE DIAGRAM

UML NOTATIONS

S.No	Name	Symbol	Description
1	Object Symbol		Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape.
2	Activation box		Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes.
3	Actor symbol		Shows entities that interact with or are external to the system.

4	Package Symbol		Used in UML 2.0 notation to contain interactive elements of the diagram. Also known as a frame, this rectangular shape has a small inner rectangle for labeling the diagram.
5	Lifeline Symbol		Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol
6	Option loop symbol		Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions.

7	Alternative Symbol	Alternative [Condition] - - - - - - - - [Else]	Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside.

LOGIN
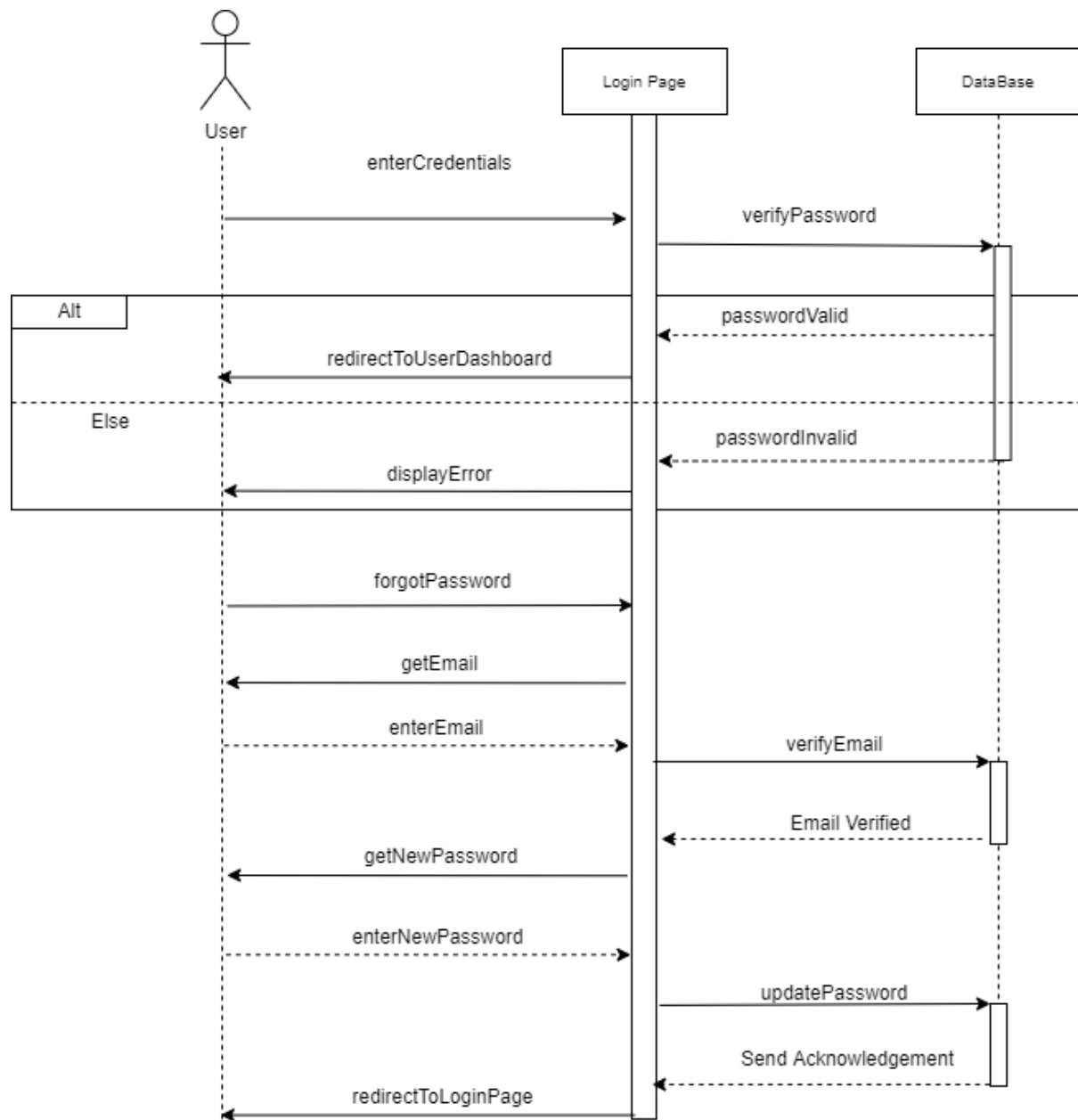
MESS SYSTEM
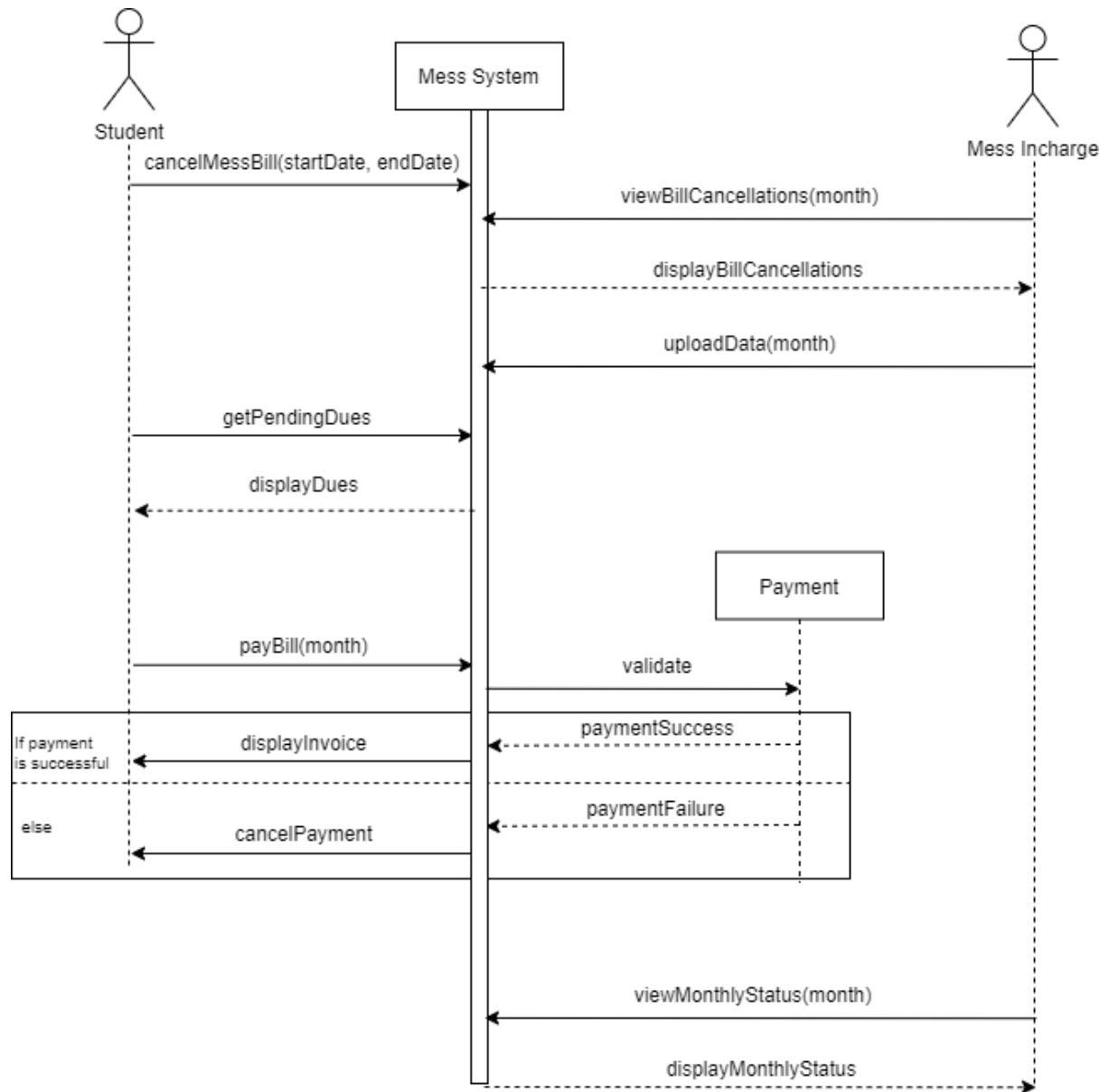
HOSTEL SYSTEM
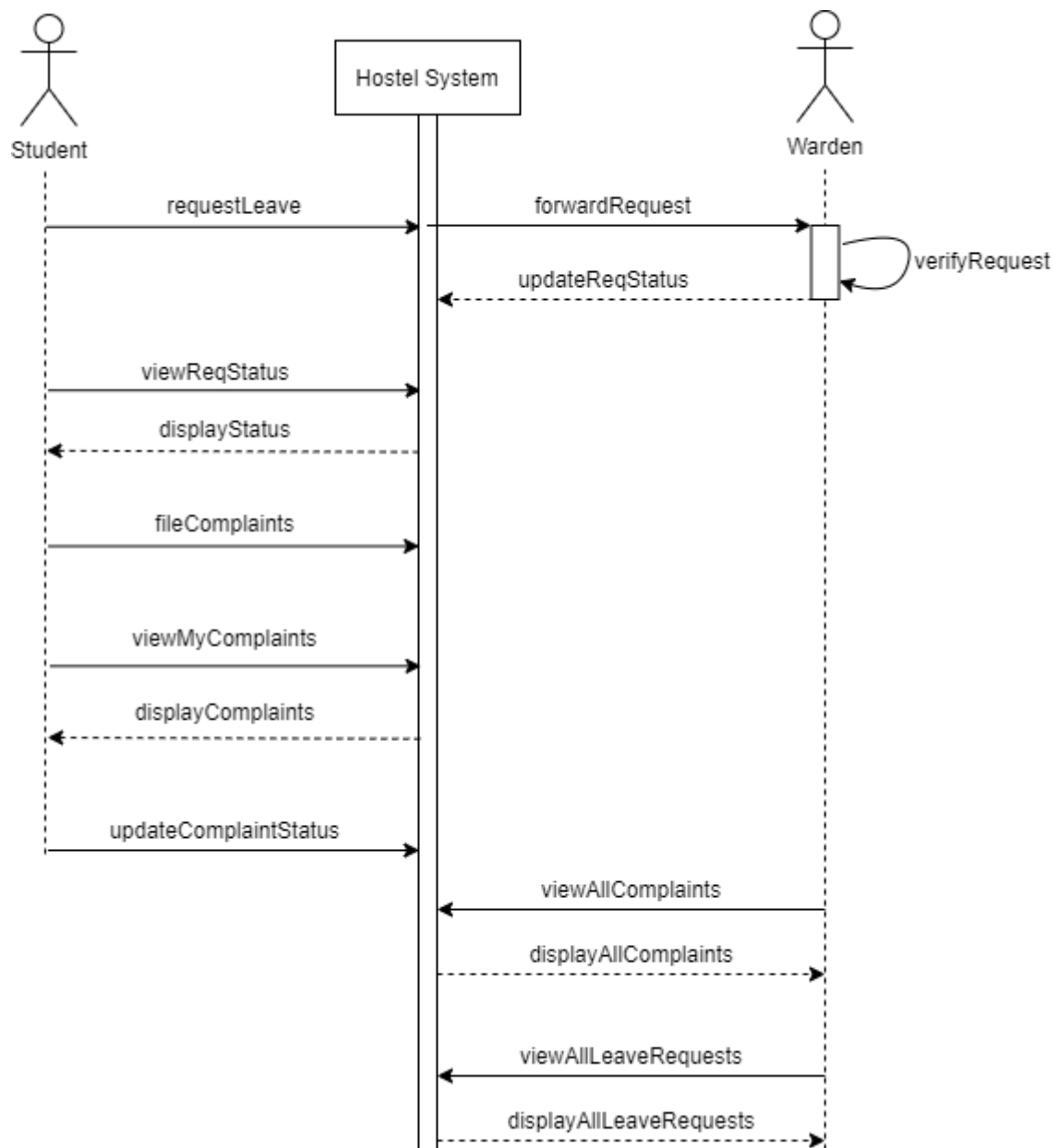
SECTION 6

COLLOBORATION DIAGRAM

COLLABORATION DIAGRAM

requestLeave
viewReqStatus
fileComplaints
viewMyComplaints
updateComplaintStatus

forwardRequest
displayAllComplaints
displayAllLeaveRequests

Hostel System

Student

displayStatus
displayComplaints

updateReqStatus
viewAllComplaints
viewAllLeaveRequests

Warden
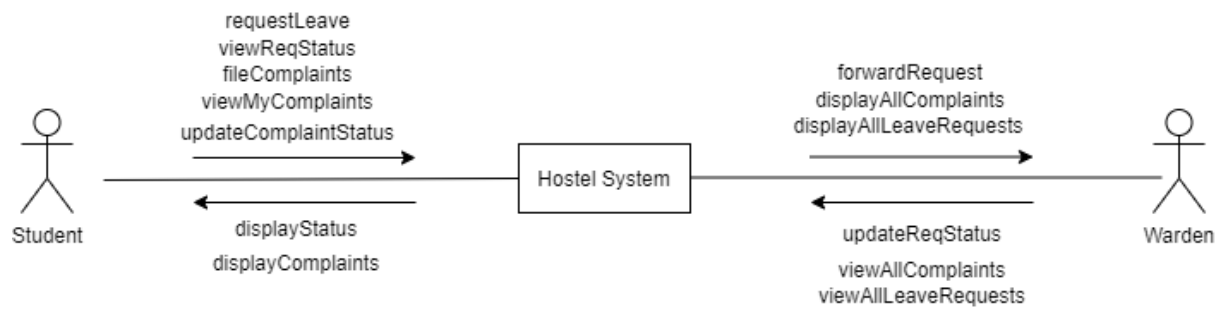
SECTION 7

STATE MACHINE
AND
ACTIVITY DIAGRAM

UML NOTATIONS

Notations for State Machine Diagram:

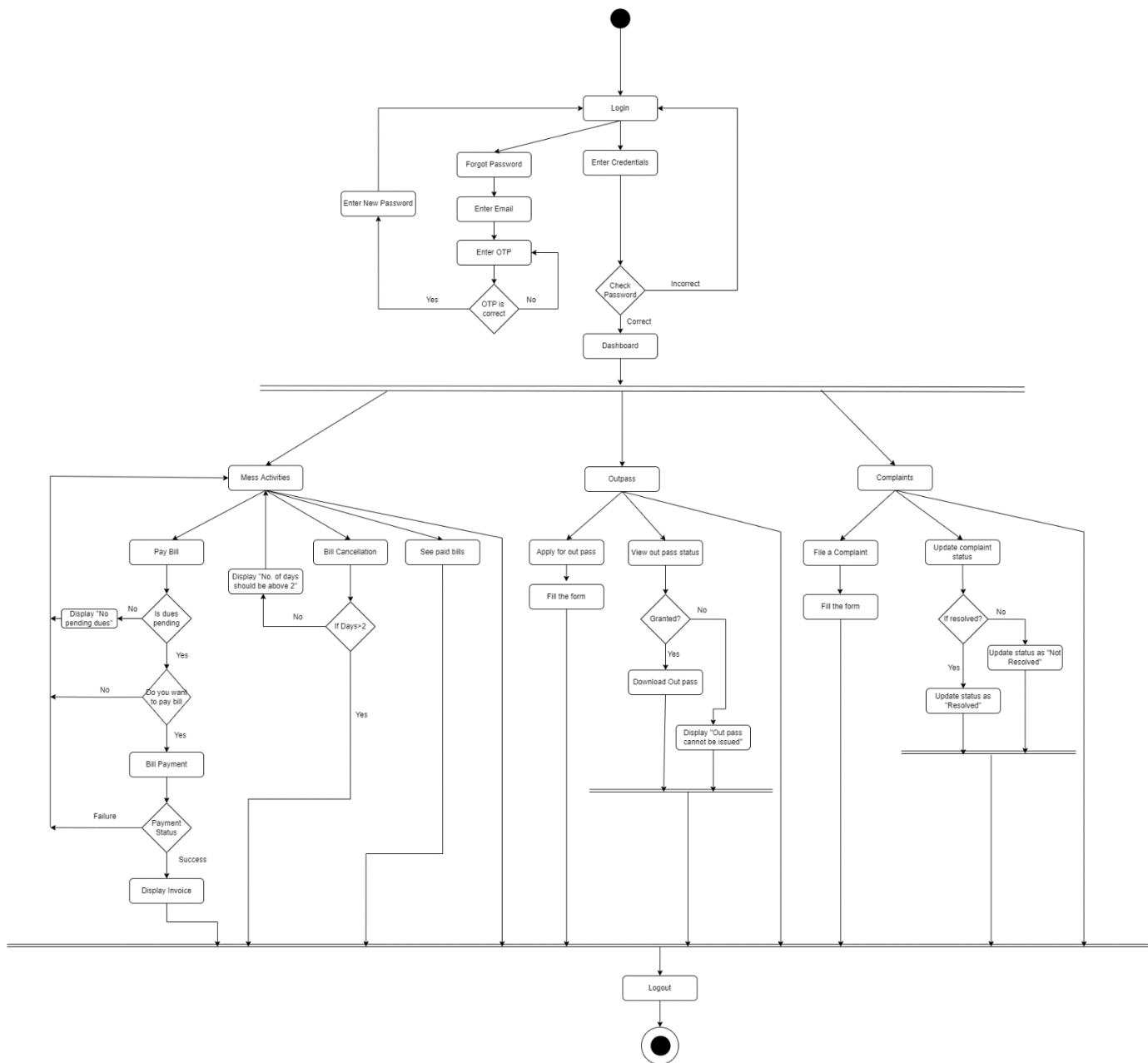| Notation	Type	Explanation
	Start	It initiates the transition
	End	It marks the end of the transitions
State	State	Each state in the state machine is represented using this rounded rectangle
[condition]	Transition based on condition	It represents transition based on condition
\Action	Transition based on action	It represents the transition based on the user action
_____	Transition	It represents transition without any action or condition

Notations for Activity Diagram:

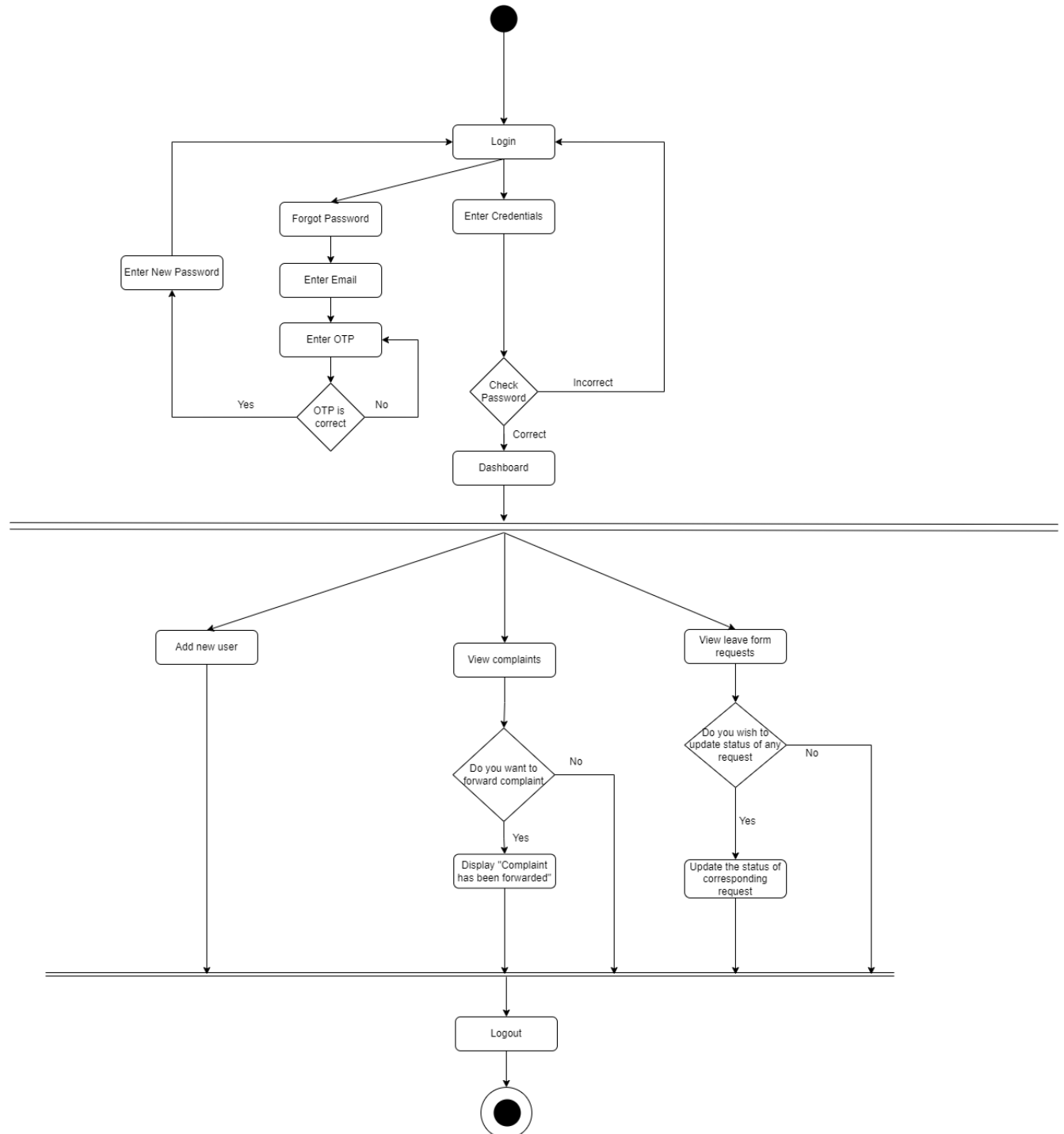| Notation	Type	Explanation
⬤ →	Start	It initiates the transition
◯	End	It marks the end of the transitions
State	State	Each state in the state machine is represented using this rounded rectangle
Decision	Decision	It represents the transitions based on the conditions
▬▬▬	Fork	It is used for concurrent executions

STATE MACHINE DIAGRAM
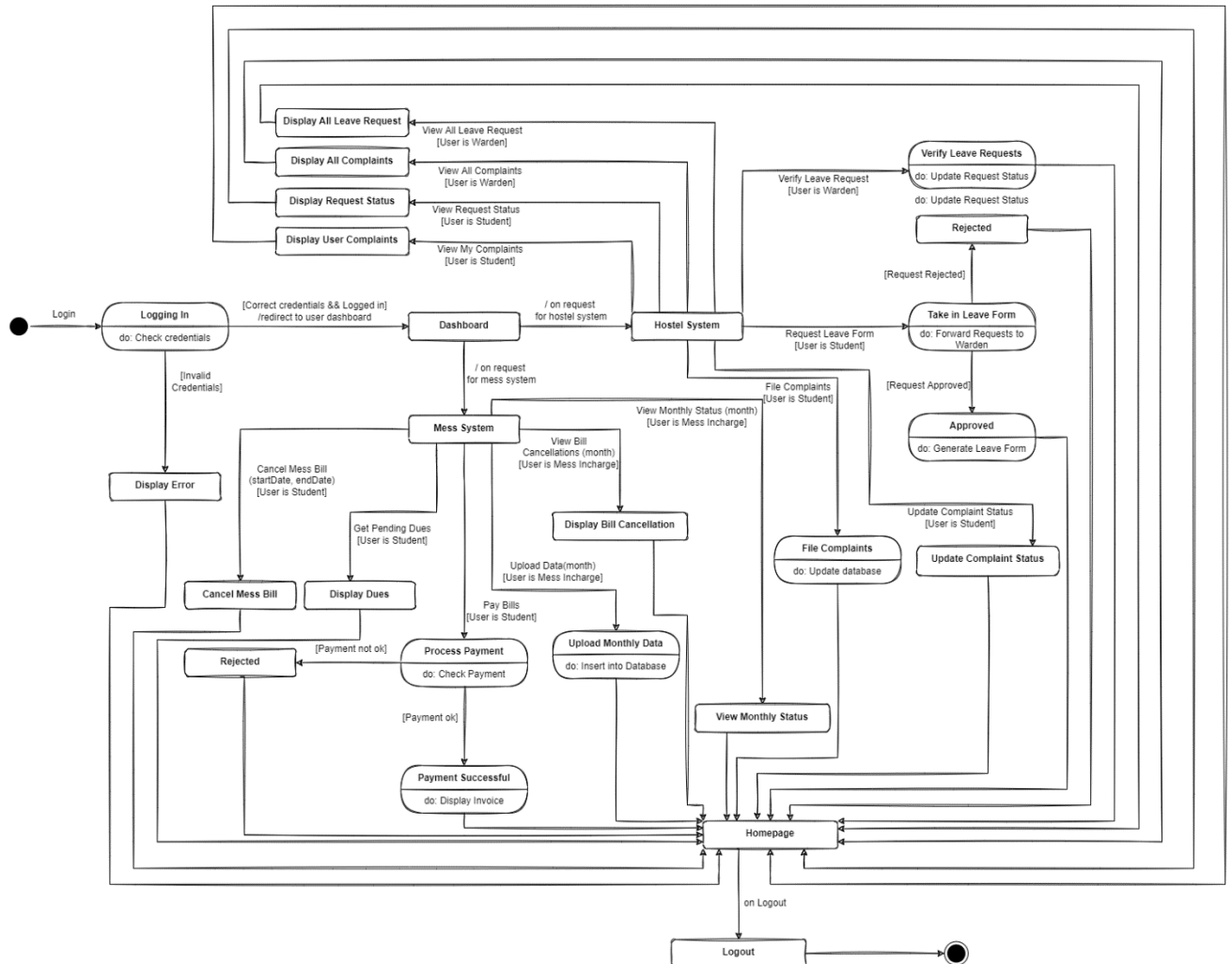
ACTIVITY DIAGRAM

STUDENT

WARDEN

MESS INCHARGE