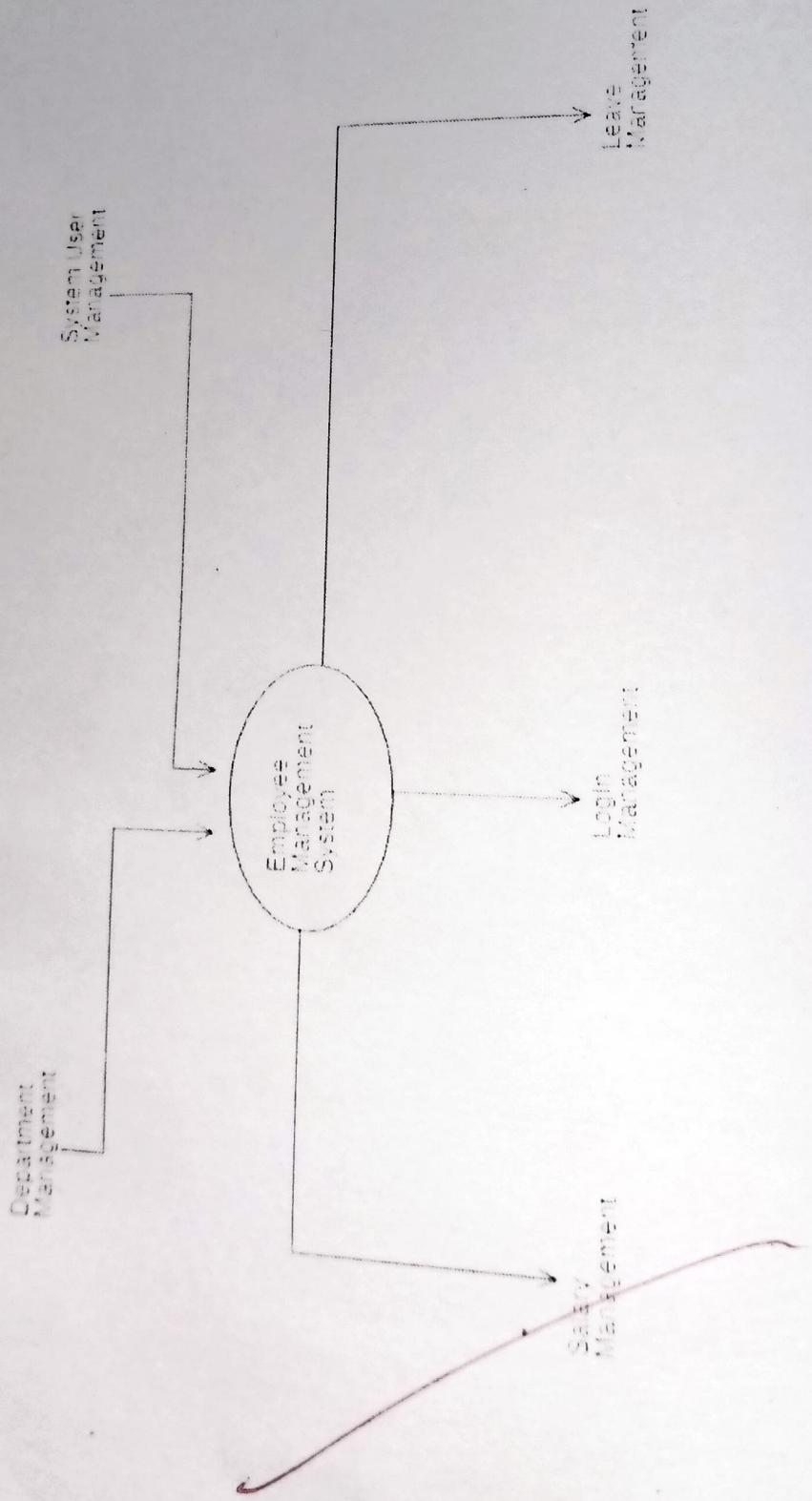


ZERO LEVEL DFD



PROJECT: EMPLOYEE MANAGEMENT SYSTEM

Zero-level DFD

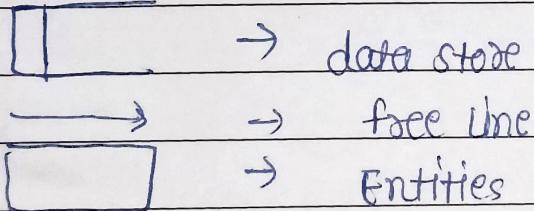
Data Flow Diagram :

DFD is the flow of data of a system or a process. It also gives insight into the inputs and output of each entity and the process itself.

Zero level DFD : It is a basic overview of the whole system or process being analyzed or modeled. It is also called a context diagram.

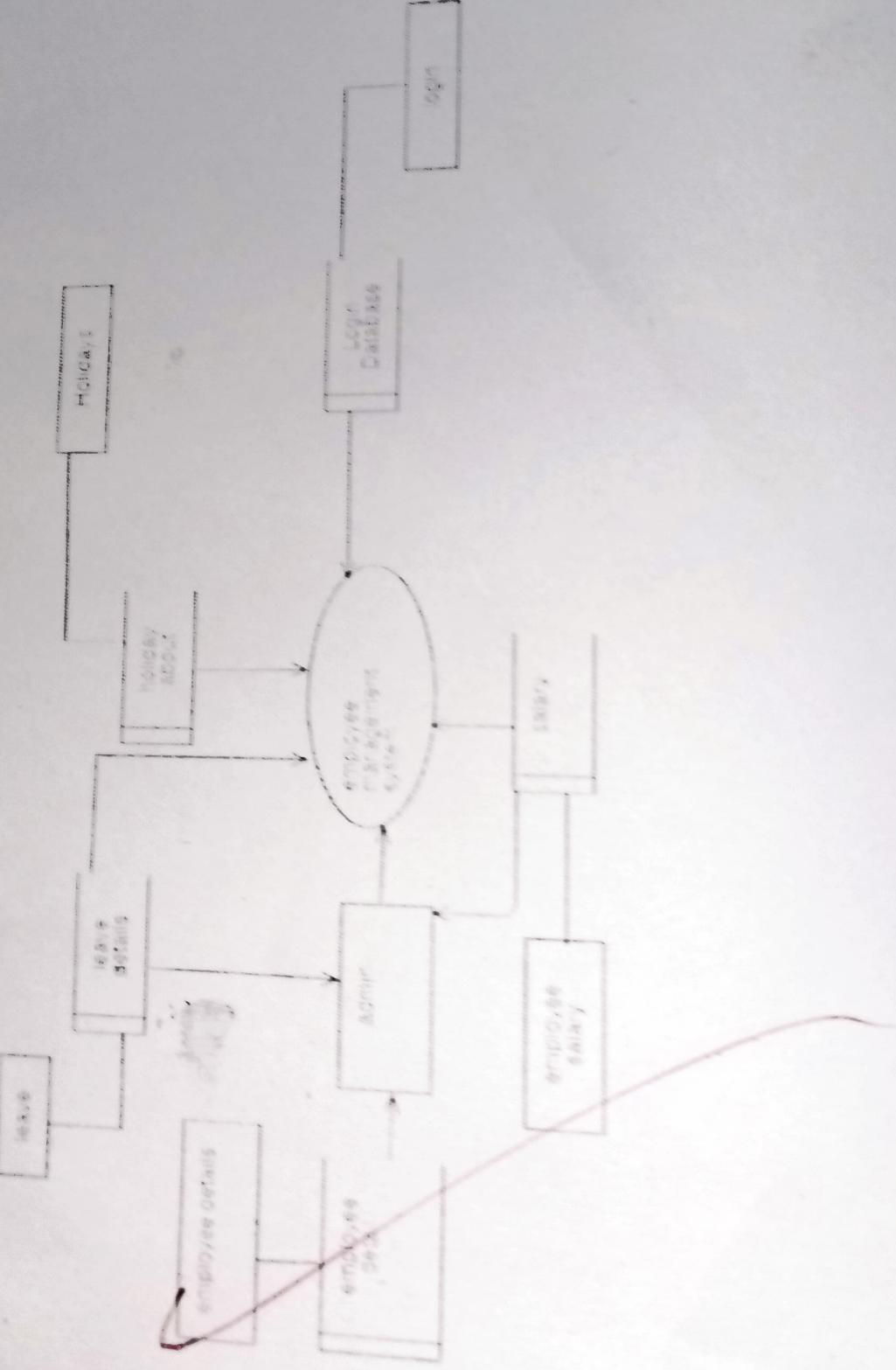
Project : Employee Management System

→ Notation :



- In this diagram, we use the Login management, Leave management, System User management, Department management and Salary management Entities.
- the Login Management is used for the Login of the user and Admin
- the leave management will be used for the leave Page No.....

1 Level DFD



application of the employee

- the department management is used for the provided the department of employee
- the salary management is used for the manage salary of employee.

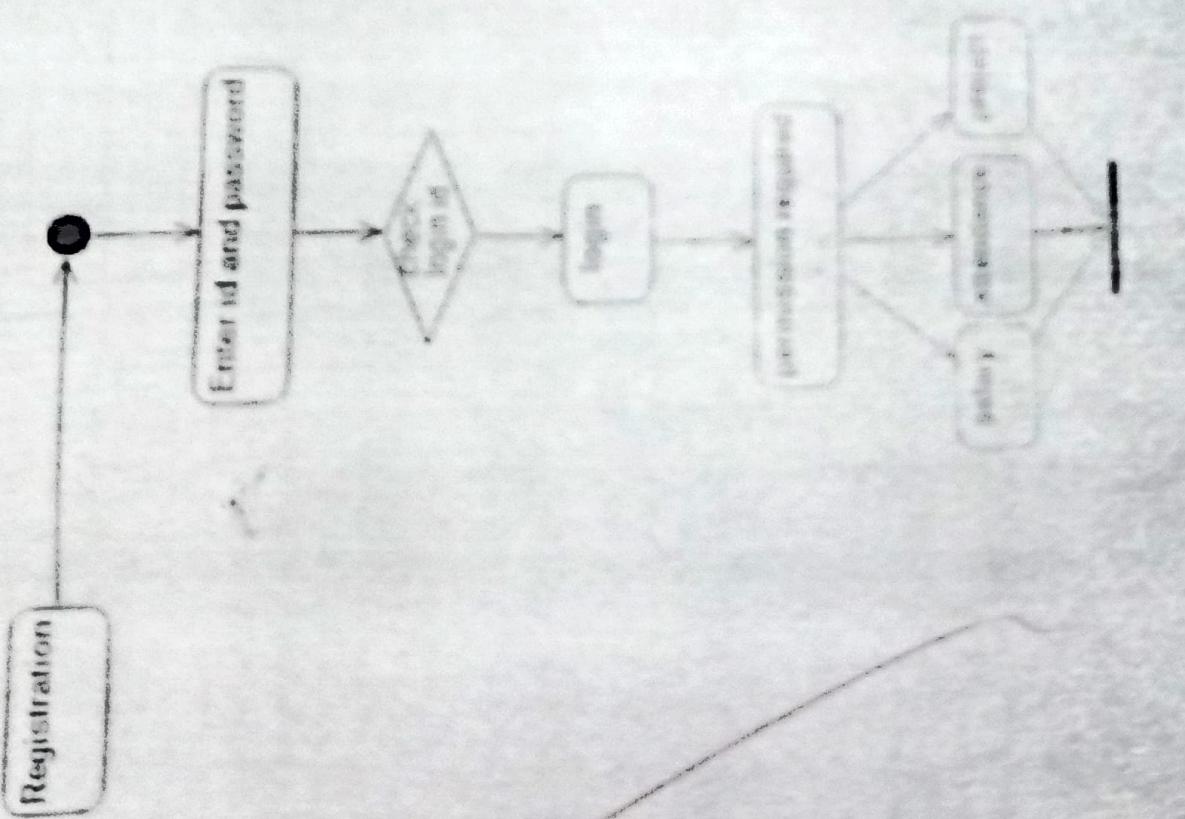
1 level DFD :

In 1-level DFD, the single process mode from the context diagram is broken down into sub-process.

- ⇒ In the 1-level DFD we use the Holidays, login, salary, leave, employee details entities.
- ⇒ In the diagram the process is the employee management system.
- ⇒ In the diagram the employee details entities is connect with the employee management system through the employee details database.
- ⇒ the Holidays entities is connect with the employee management through the Holiday about database.
- ⇒ the leave entities is connect with the admin through the leave details database.

- ⇒ the Salary entities is connect with the admin and employee management through the salary database.
- ⇒ the login entities is connect with the login database with the employee management.

Activity diagram of Employee Management System



Activity Diagram :

Activity Diagram is basically a flow chart to represent the flow from one activity to another activity.

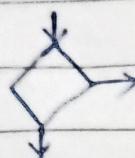
Notation :



→ Activity



→ transition



→ Decision

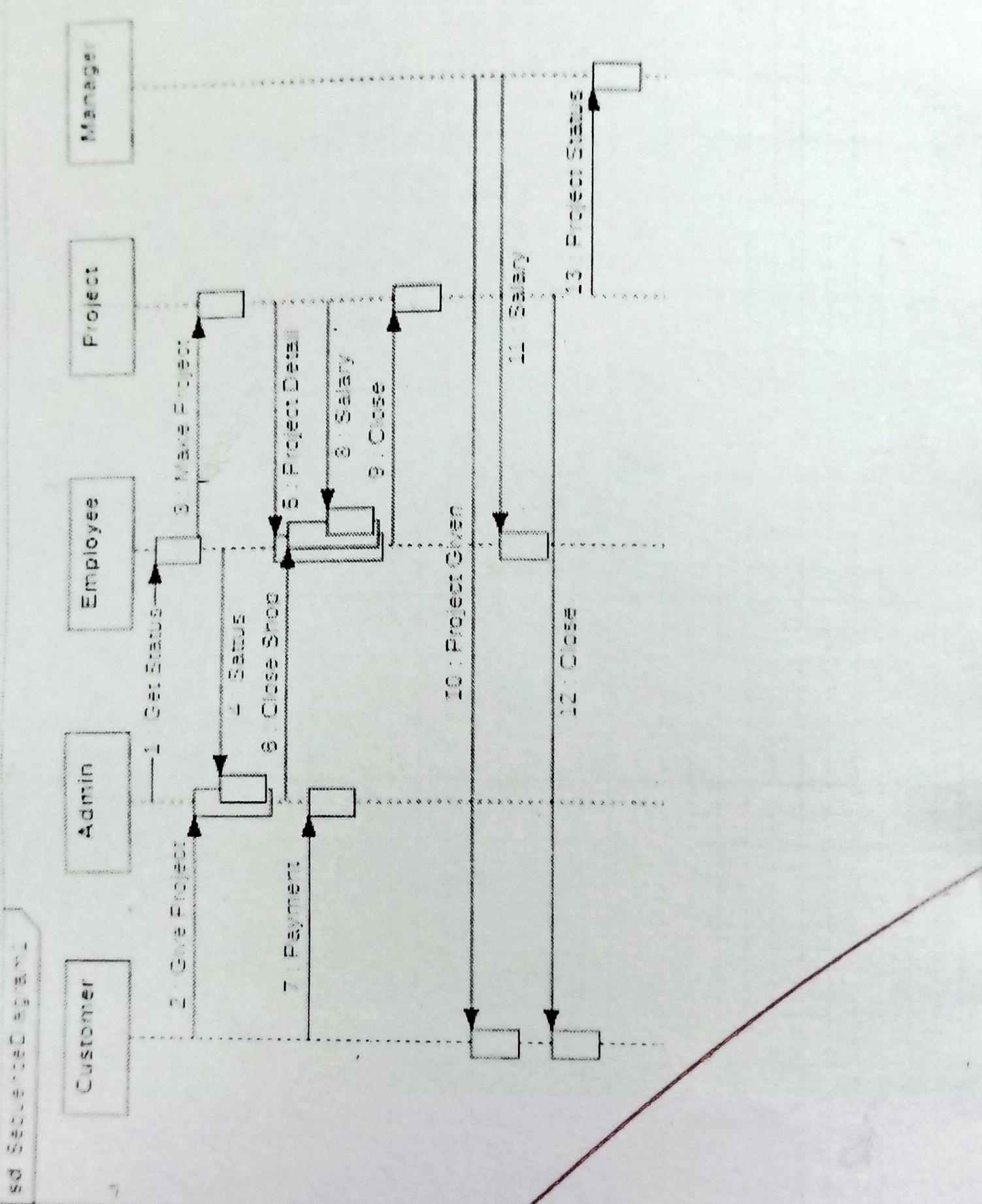


→ Fork



→ Initial State

- ⇒ first of all we Registered in the account.
- ⇒ After then we use the Enter id and Password activity for enter id.
- ⇒ After then Login through the check login id decision.
- ⇒ After then the permission required is use for the check Permission for the check it Salary attendance and Project.
- ⇒ After then we reach the final State.
- ⇒ And User access the functionality according to the Permission.



Sequence Diagram :

The sequence diagram maintains order of the few of message inside the system by depicting the common relationship b/w two life line. Just like a time order sequence event.

NOTATION :

Instant : class → OBJECT

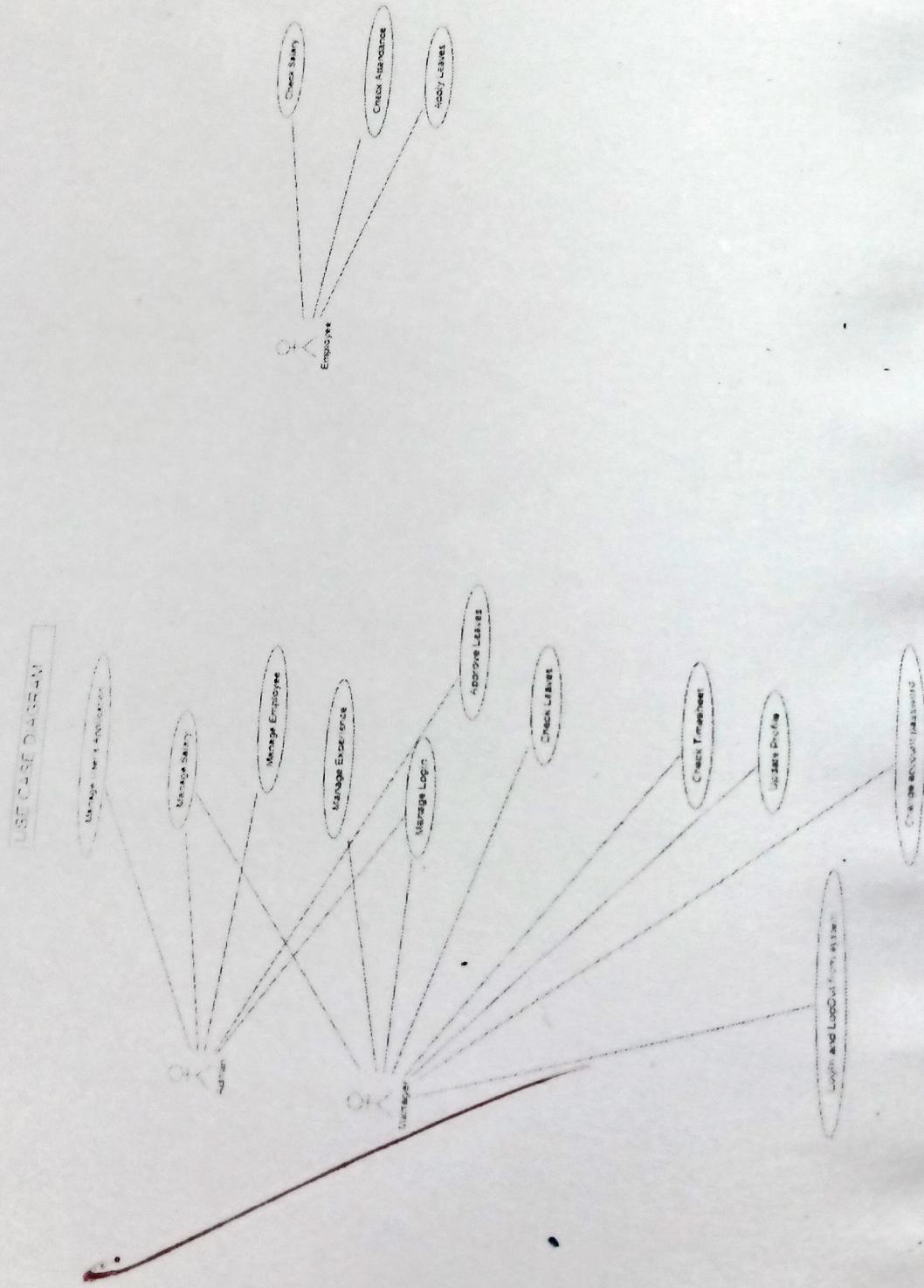
Instant : class → Life Line

→ Scope

→ → Message Transition

- ⇒ In the diagram customer, Admin, employee, Project and manager are the object.
- ⇒ The customer give the project for Admin. and Admin ask the status for em of project from employee.
- ⇒ Employee tell the status of the project to the Admin.

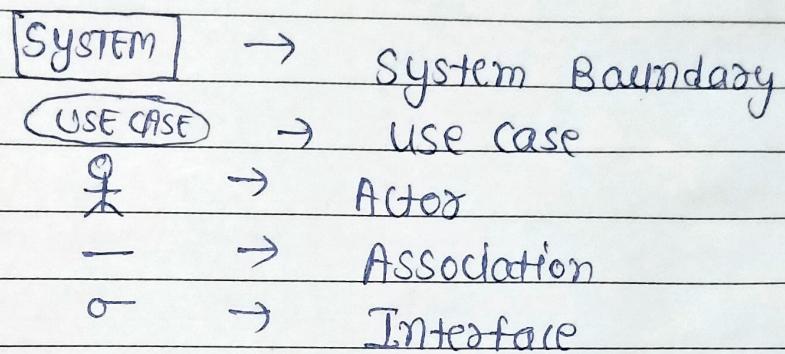
- ⇒ the manager is ask the about of the Project Status from the Project object.
- ⇒ the manager gives the salary to the employee.
- ⇒ the employee make the Project.
- ⇒ the customer tell about the closing time of Project.
- ⇒ After the complete of Project the Project will be close.



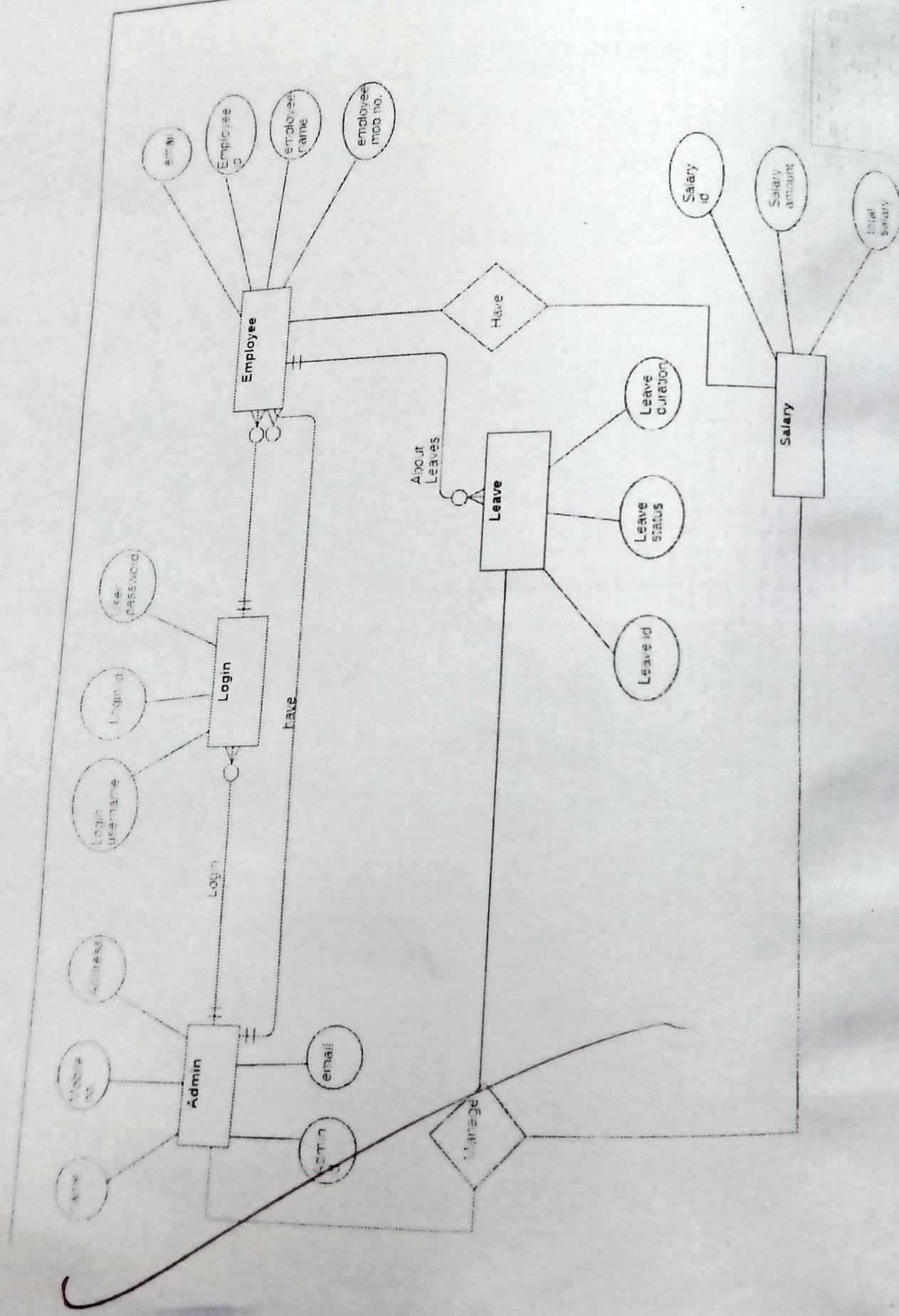
USE CASE DIAGRAM :

It represents a function of a system by utilizing actors and use case. It encapsulates the functional requirements of a system and its association with actors.

Notation :



- ⇒ In the Project the Admin as a actor and admin is associate with the manage user's application usecase, manage salary usecase, manage employee usecase, Approve Leaves and Manage login usecase.
- ⇒ the manager as a Actor and Manager associate with the Manage login usecase, Check leave usecase, Check timesheet usecase, update profile usecase, login and logout from System usecase.
- ⇒ the employee as a Actor and associate with the check salary usecase, check attendance usecase, and Apply leaves usecase.



ER Diagram :

ER diagram is a modeling method used in Software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modeling methods are called Entity-relationship or E-R diagram.

Notation :

 → Entity

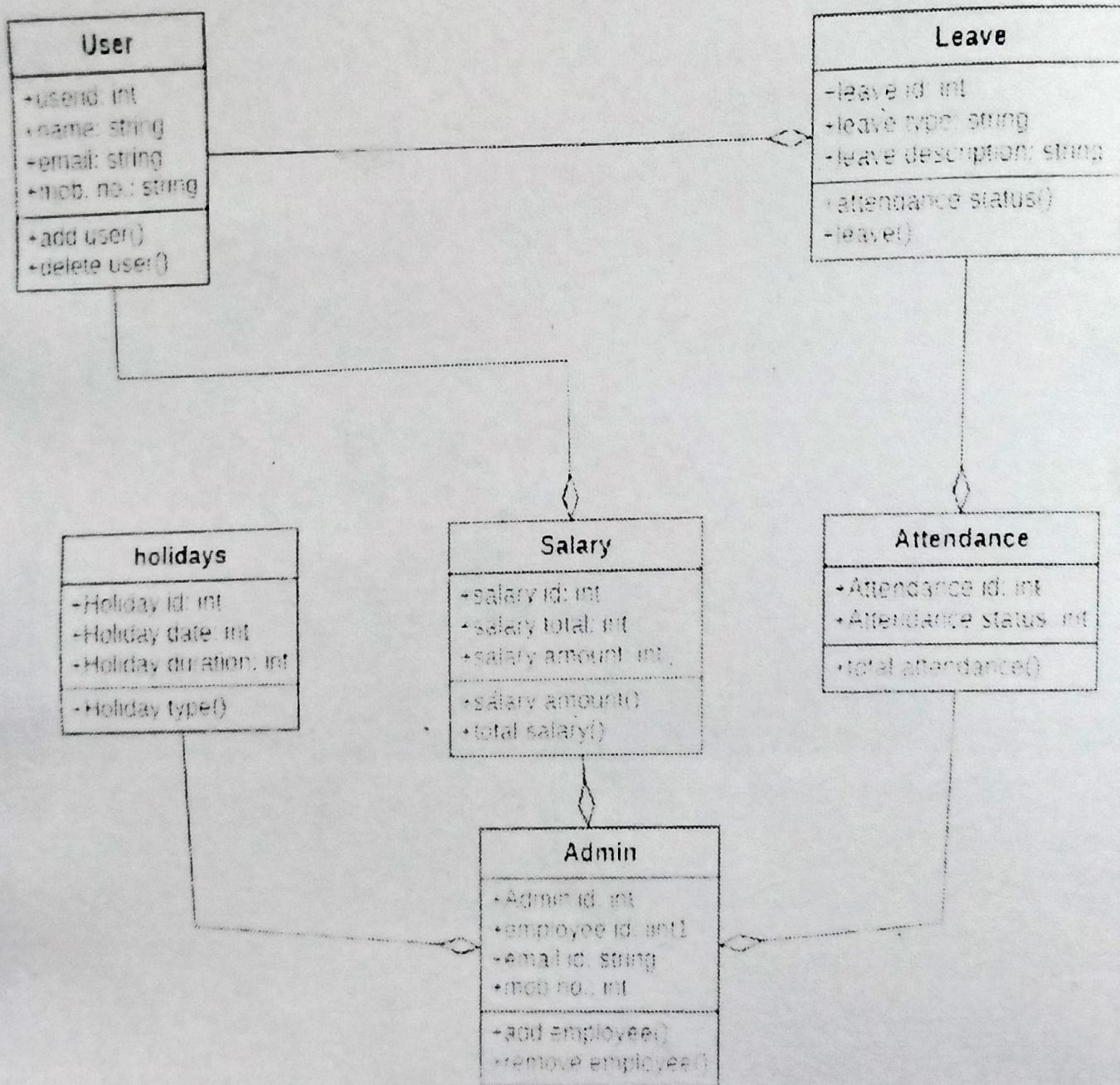
— → One to one

— ⇋ → one to many

- ⇒ In the project the Admin have the name, mab no., address, email attribute and the admin have the connection one to many to the login, employee.
- ⇒ the admin manage the leave of the employee and leave have the leave id, leave status, leave duration attributes.
- ⇒ Employee have the one to many connection to the leave and employee have the salary and about of the salary.

- ⇒ the login have the one-to-many connection to the employee for login
- ⇒ the Login Entity have the login username, login id and user Password attributes.
- ⇒ the Employee Entity have the Employee email, id name, mab.no. attributes.

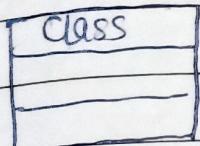
CLASS DIAGRAM



CLASS DIAGRAM :

Class Diagram is a static diagram it represents the static view of an application class diagrams not only used for visualizing describing & documenting different assets of system but also for constructing executable code of the software application.

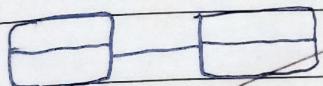
Notation :



→ Class



→ Object



→ Link

- ⇒ In the Project have the User class and it have the user id , name , email , mob. no. attributes and have the add user and delete user operation.
- ⇒ In the leave class have the leave id and leave type and leave description attributes and attendance and leave operation. Connected with the attendance
- ⇒ In the holiday class have the Holiday id and Holiday date and Holiday duration attributes and connected with the Admin

- ⇒ the Attendance Class have the attendance id and Attendance status attribute and connected with the Admin.
- ⇒ the Holiday class have the Holiday id and Holiday date and Holiday duration attributes and Holiday type operation and Connected with the Admin.
- ⇒ the Salary class have the Salary id and Salary amount and total Salary attributes and connected with the Admin.

✓
~~Refer~~
15/12/23