



# Software Requirements specification (SRS) Document for the course registration system.

## 1. Introduction :

### 1.1 Purpose :

The Software Requirements specifications document explain the functional and non-functional software requirements of student course registration tracking system for the university. The intention of the document is that it is to be used by the members of the group making this purpose is also to ensure that all of the requirements present in the document are catered during the development phase of the system.

### 1.2 Intended audience :

The document is intended to be read by the customer. This is a technical document and terms should be understood by the customer. The customer needs to understand this document fully. So that they can draft a design document using the SRS presented to them by the analyst.

### 1.3 Intended use :

There are various kinds of users for this course registration. These registrations can be applied via online by many companies, students, professors etc... Web based applications offer a good way of accessing centralised data securely. Servers will be accessed directly by the person or team managing them. Through the use of cloud computing, servers can be fully redundant and replicable to prevent downtime as a result of a disaster.

## 1.4 Scope :-

- The system provides online interface to the user where they can fill in their personal details and submit the necessary documents.
  - The registrar concerned with the issue of course can use this system to reduce his workload and process the application in a speedy manner.
  - Provide a communication platform between student and registrar.

## 1.5 Definitions and Acronyms:

CRS :- course Registration system

AIS :- Academic information system

SIS :- Student information system.

## 2. Overall Description:

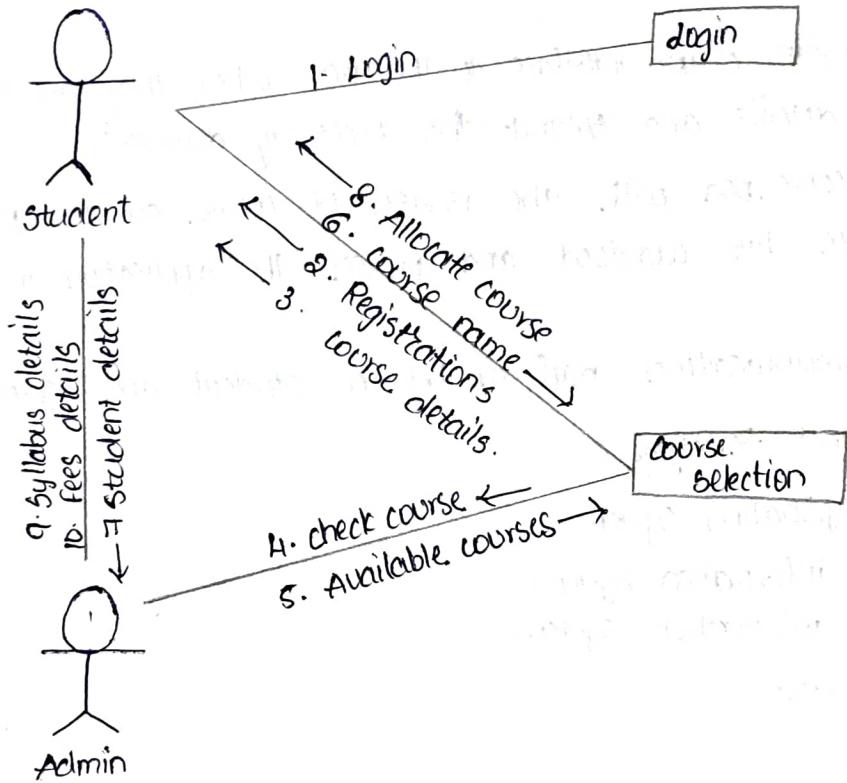
## 2.1 User needs :-

- secure payment process.
  - further communications.
  - contact information
  - conditional login
  - Registration fee details
  - complete details of the course that available.

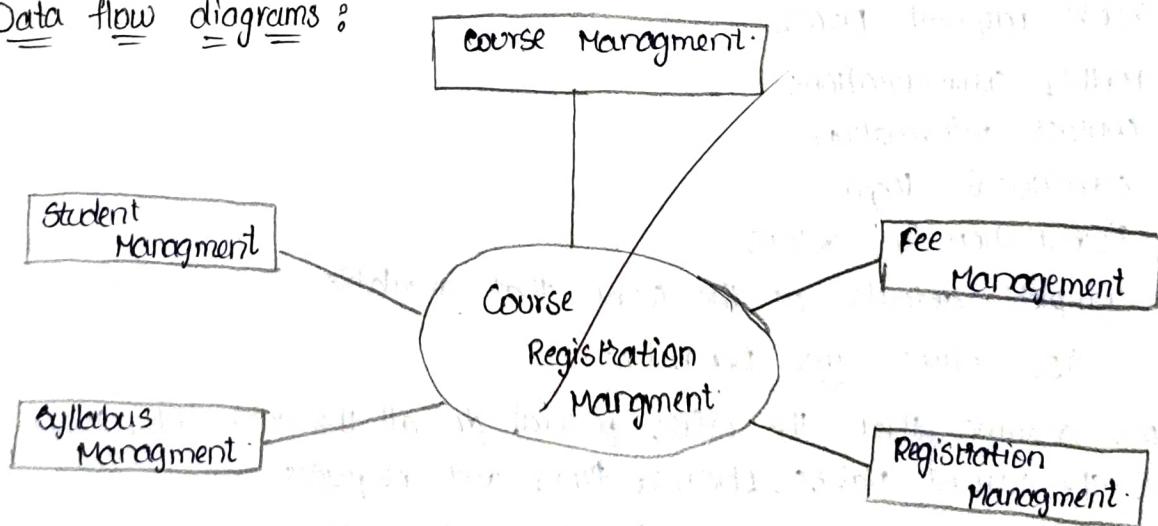
## 2-2 Assumptions and Dependencies?

- we assume that the office personal do all the data entry based and the current values obtained from and registers.
  - We assume that the computer that will use the software will be part of the having proper platform to run it.
  - User with administrator access should be careful in deleting or modifying only information knowingly or unknowingly which will lead to data base.

→ collaboration diagrams:



Data flow diagrams:



- the end users of this software are assumed to have basic level of the computer knowledge i.e., point and click.

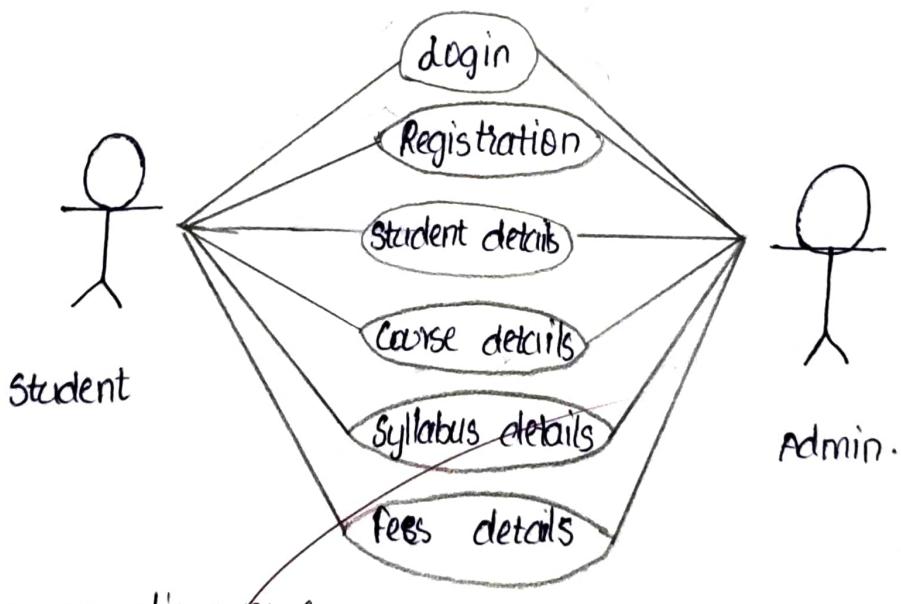
### 3. Systems features and Requirements :-

#### 3.1 Functional requirements :-

UML Diagrams :-

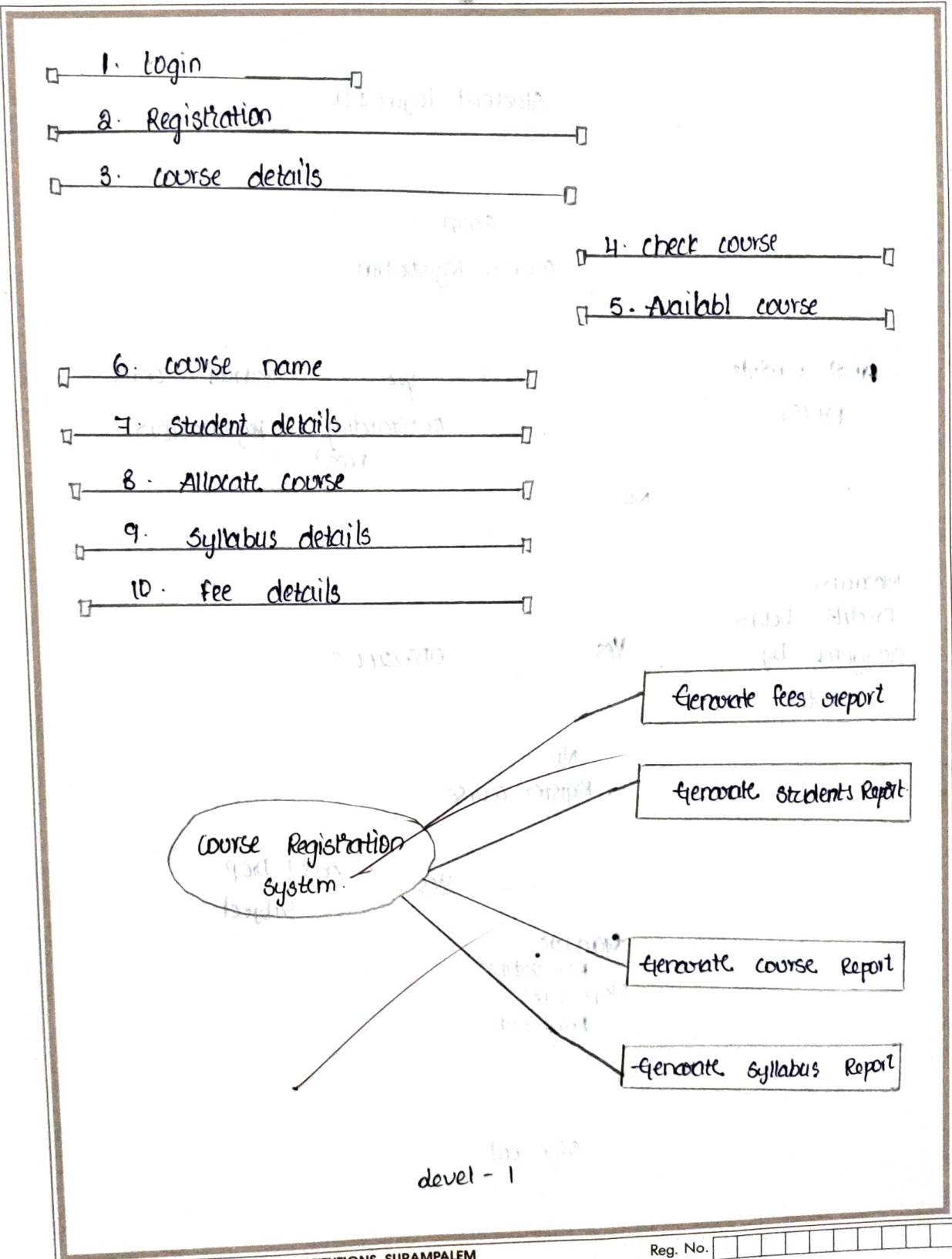
- usecase diagram.
- sequence diagram.
- collaboration diagram.

→ use case diagrams :-

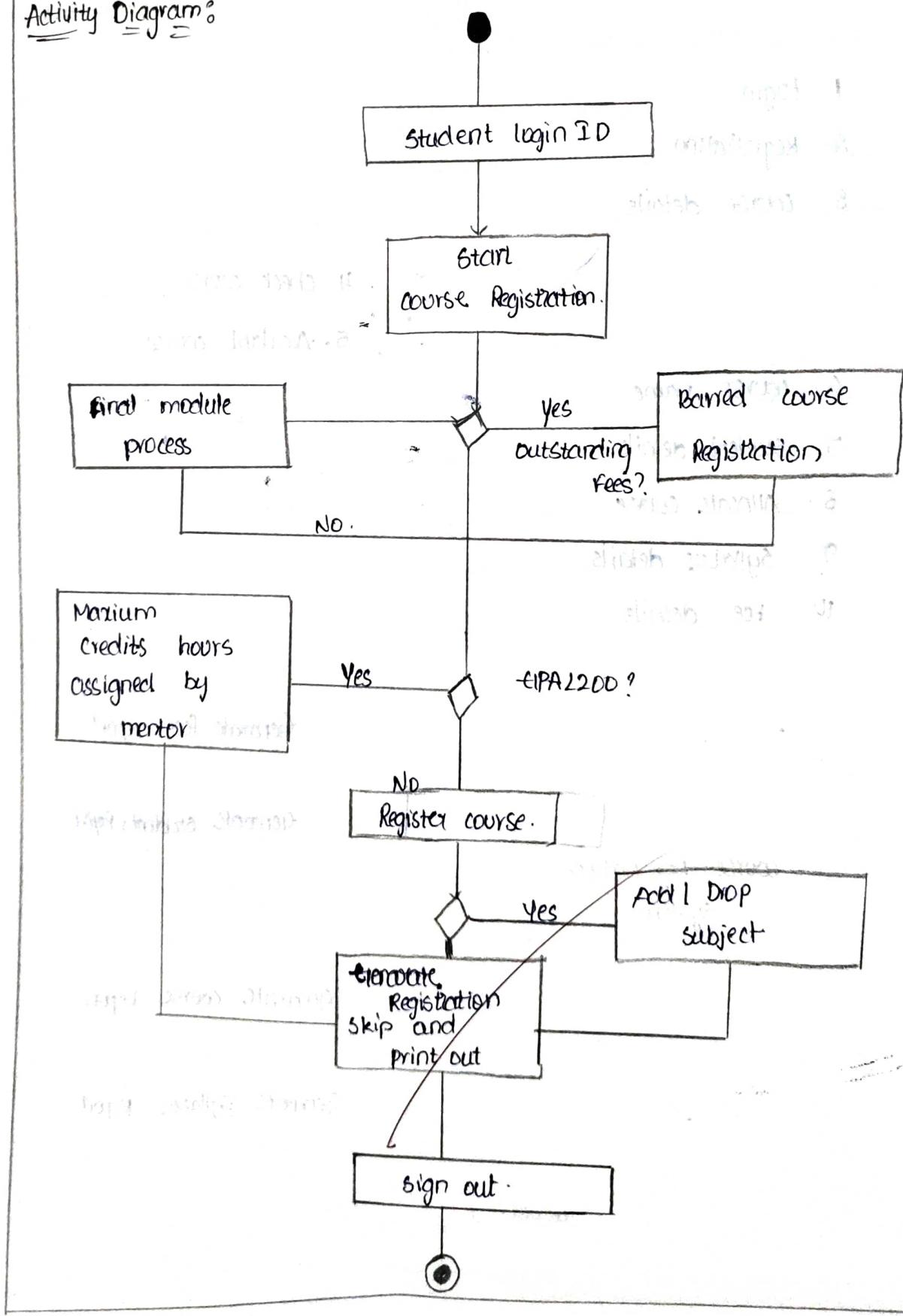


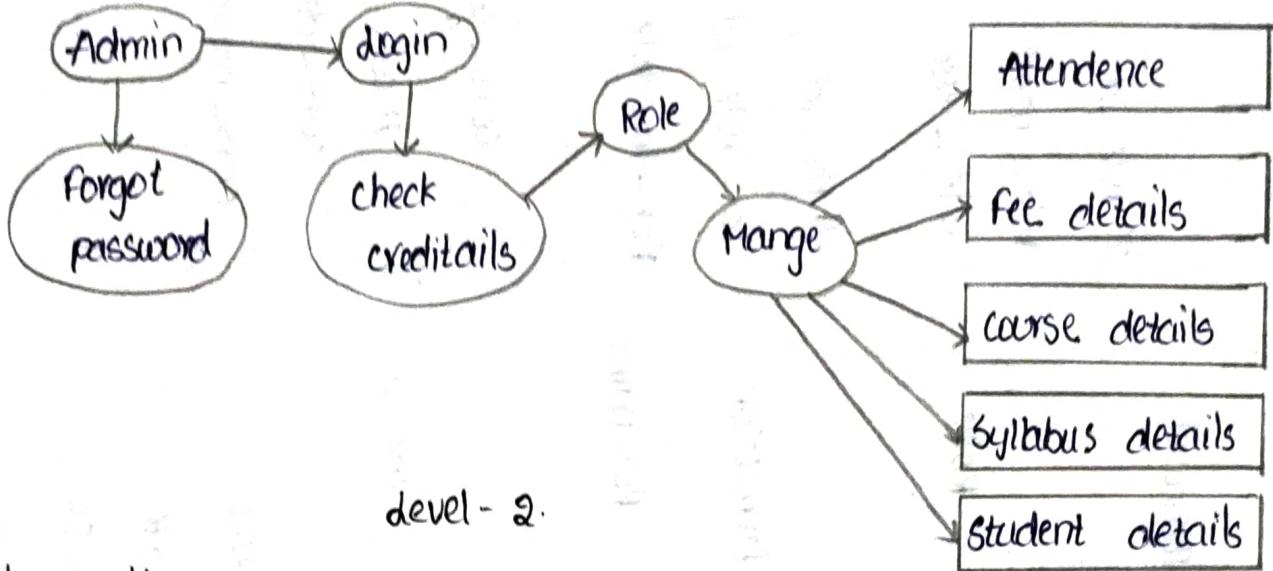
→ sequence diagrams :-



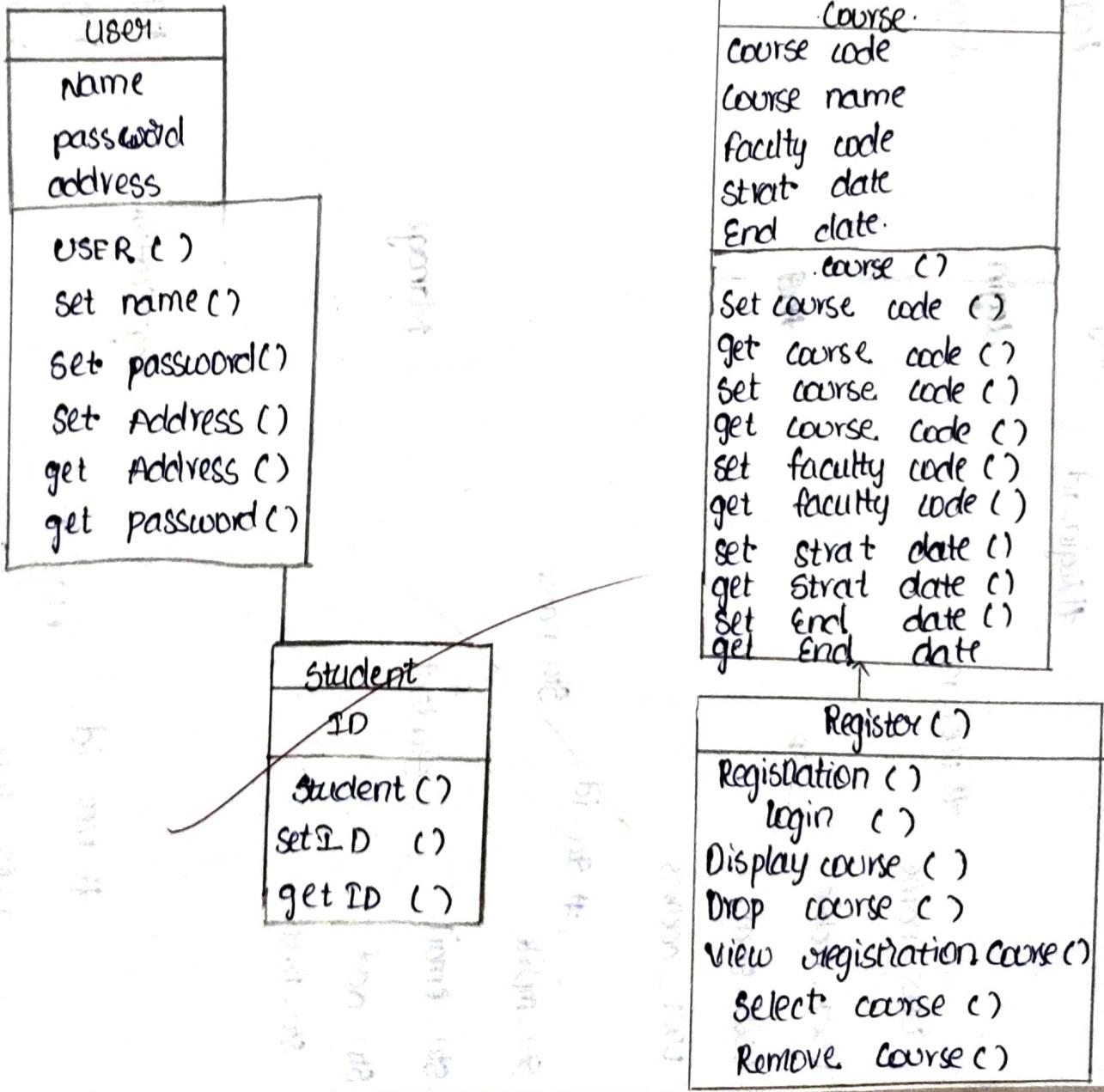


## Activity Diagram:

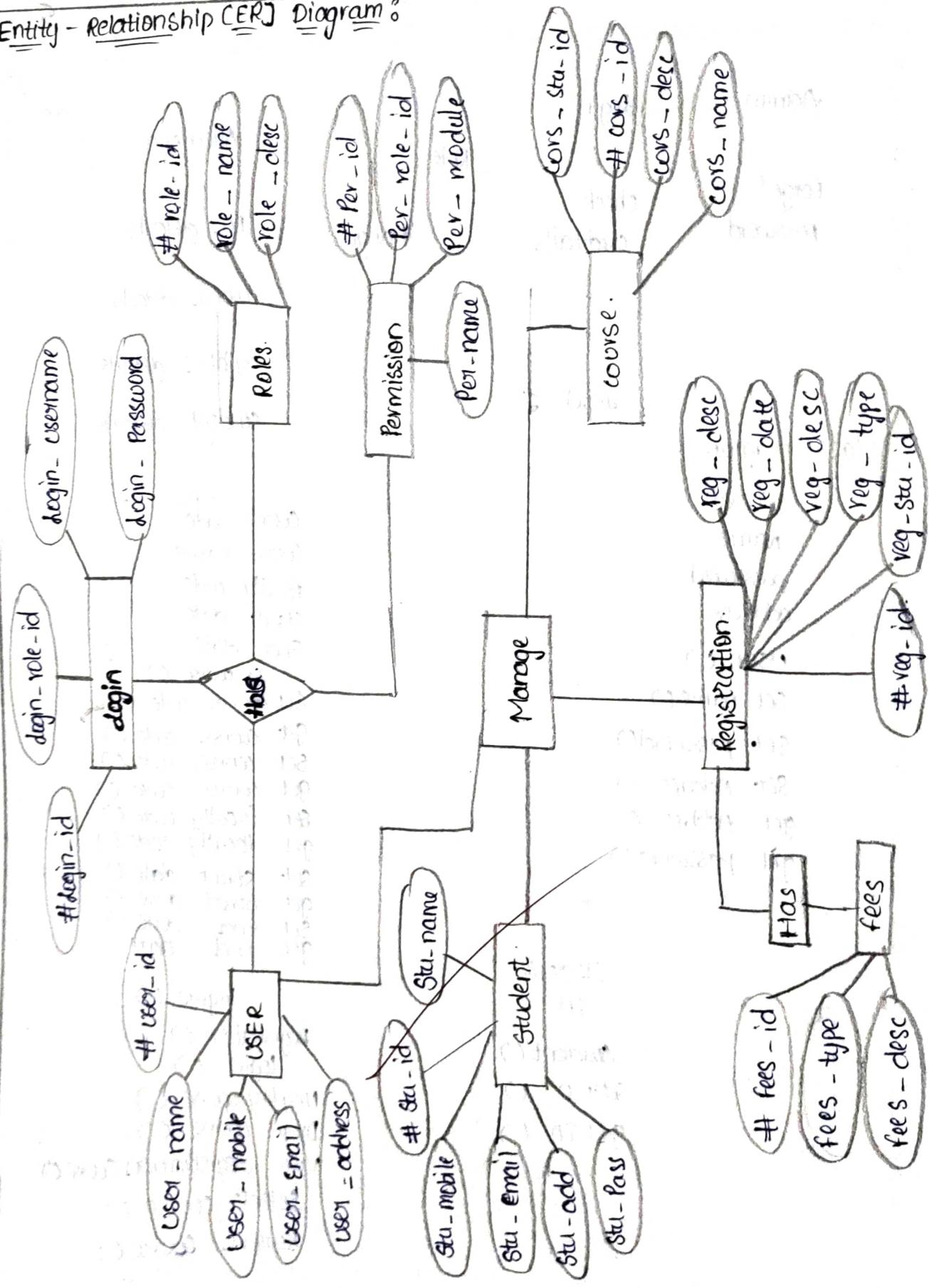




class diagram:-



## Entity - Relationship (ER) Diagram



### 3.2 External interface Requirement :-

#### → User interface :-

The user is going to interact with the system through different interface.

- When a student logs in to the online registration system, they will have to create a profile if they are new users of the system.
- Once login, user will choose and action to do, new users can view course catalogue and proceeds to registration.
- Old users can cancel the registration or replace courses registered.

#### → Hardware Interface :-

This system will not be engaging with the hardware in terms of exchanging information or functionalities, it will run on the hardware. The system will be dependent on the capabilities of the hardware and will not alter any functionalities of the hardware.

#### → Software Interface :-

The system will require a lot of software connections so it has to be compatible to external software to be able to connect. The main system will be composed of the web portal, the operating system used and the database all communicating together and exchanging information.

#### → Communication Interface :-

The main communication link that the system will be using the internal. The information transmission methods are:- Web file transfer, E-mail, social network information transfer.

### 3.3 System features :-

- provides complete web site solution, including student registration, status update - complete web base administration.

- Students can register and login into the system to check registration status online without needing to visit university campus.
  - Admin can login and review registration details and update status.
  - Generates accurate data as data is entered by student, system enforces validation to check data purity while saving the record.
  - We can edit the marks of the students and generate the report again.

### 3.4 Non-functional requirements :

→ performance :-

- The system shall support up to 2000 simultaneously users against the original data base at any given time, and up to 500 simultaneous users against the local servers at any one time. The system shall provide access to the legacy.

→ Reliability :-

the system shall be available all hrs a day 7days a week. user access at any time

→ scalability:

Scaling the system to large number of users · large courses  
will have hundreds of students · parallel

~~275~~  
N. Danell

→ Security :

- The system must prevent students from changing any schedules other than their own, professors from modifying assignment course offering for other professors
  - Only professors can enter grade for students
  - Only the registrar is allowed to change any student information.

Expt. No. :

Date :



# Software Requirements Specifications (SRS) Document for Student Marks Analysis System

## 1. Introduction :-

### 1.1 Purposes :-

This specification document describes the capabilities that will be provided by the software application STUDENT MARKS ANALYSIS SYSTEM. It also states the various constraints by which the student system will abide. The intended audience for this document are the development team, testing team and end users of the product.

### 1.2 Intended audience :-

The intended audience for this document are the development team, testing team and end users of the product.

### 1.3 Intended Use :-

Student marks analyzing system has been designed to carry out the mark analysis process in education institute. The result of respective departments can be efficiently computed without much of manual involvement. The purpose of this document is to define the requirement of mark analysis system.

### 1.4 Scope :-

The application will manage the informations about various students enrolled in this course in different years. The subjects offered during different semester of the course, the marks obtained by various students in various subjects in different semesters.

### 1.5 Definitions and Acronyms :-

- SRS - Software requirement specifications
- IEEE - The Institute of Electrical and Electronics Engineers.
- DFD - Data flow diagrams
- DB - Data Base
- OS - operating system.

## 2. Overall Description:

## 2.1 User Needs :

The projects is available on the internet. the university website should provide a user manual on how to use an student marks analyzing System. Users of the System will be guided by the System all the way to do their work.

This subsection of the SRS should describe the general characteristics of the intended uses of the product including educational level, experience and technical expertise. Their characteristics:

- The users should be familiar with the Internet.
  - User should be computer literate.

## 2.2 Assumptions and Dependence :

The assumptions and dependences relevant to the system are as follows:

- User should have internet access.
  - The user must have access or be on a reliable network.
  - Faculty, Student and admin should have been admitted to the University.
  - Faculty enters & updates subject Marks.
  - The number of subjects to be taken by the students in each semester doesn't change.

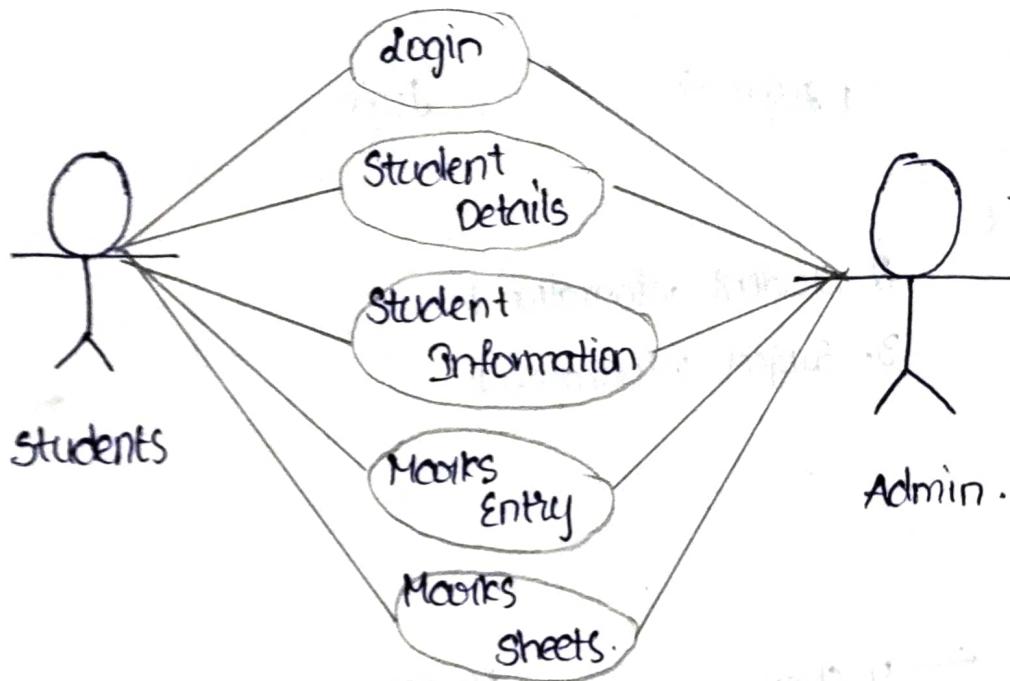
### 3. System Features and Requirements:

### 3.1 functional Requirements:

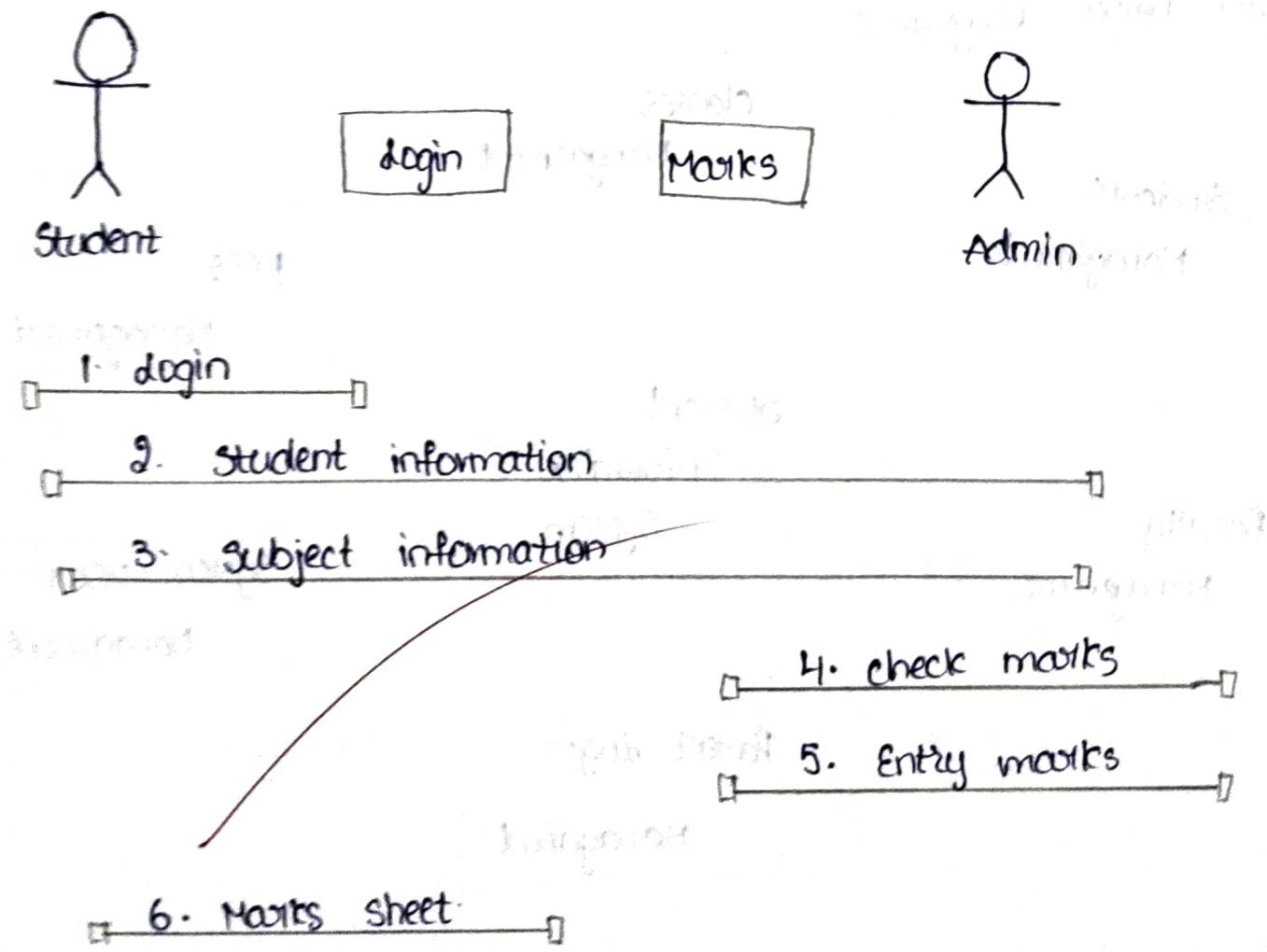
## UML Diagrams :-

- Use case diagram
  - sequence diagram
  - collaboration diagram

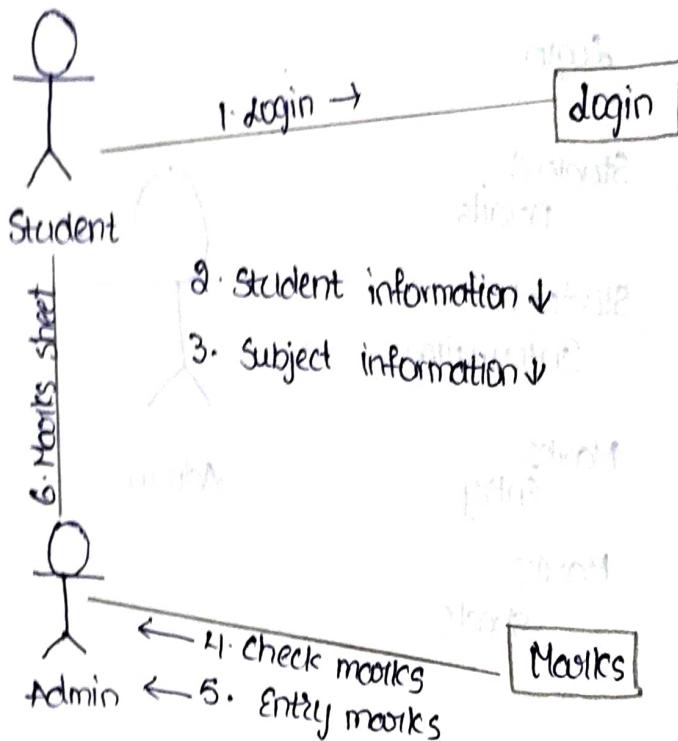
→ Use case diagram:



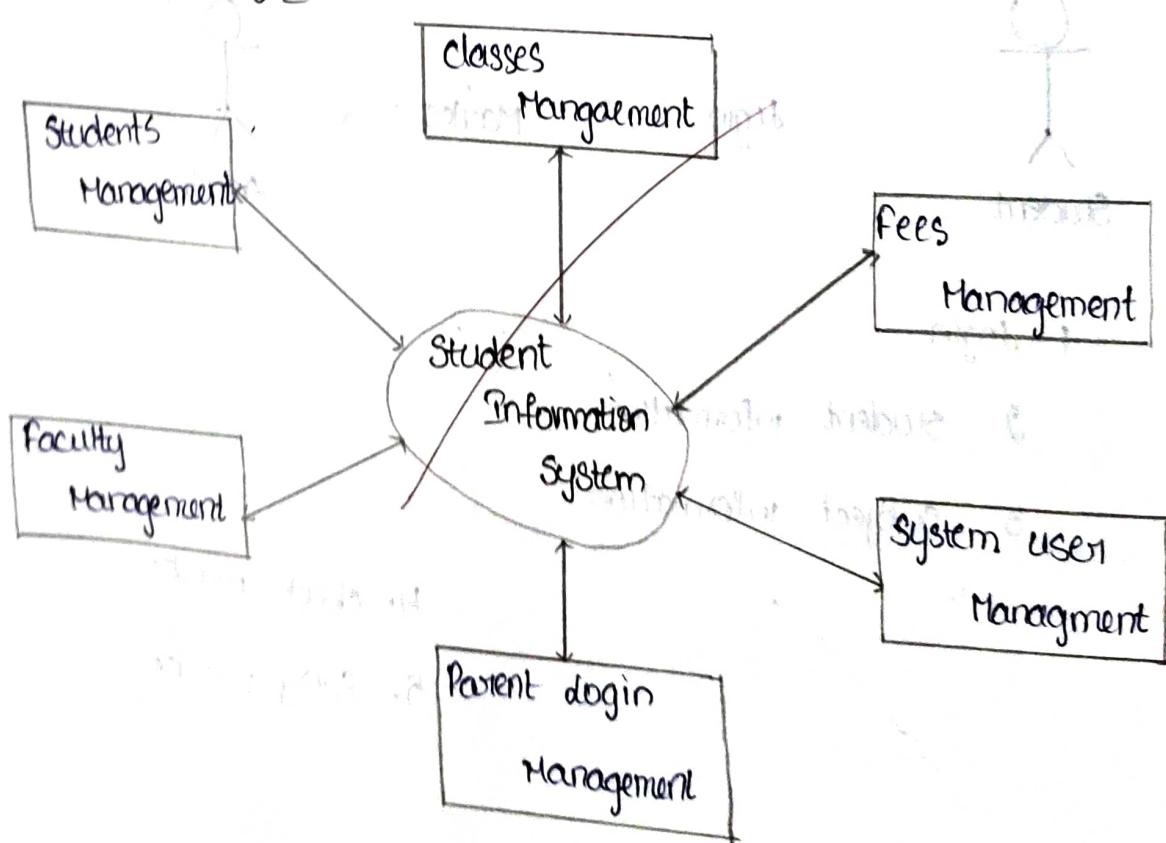
→ Sequence Diagram:



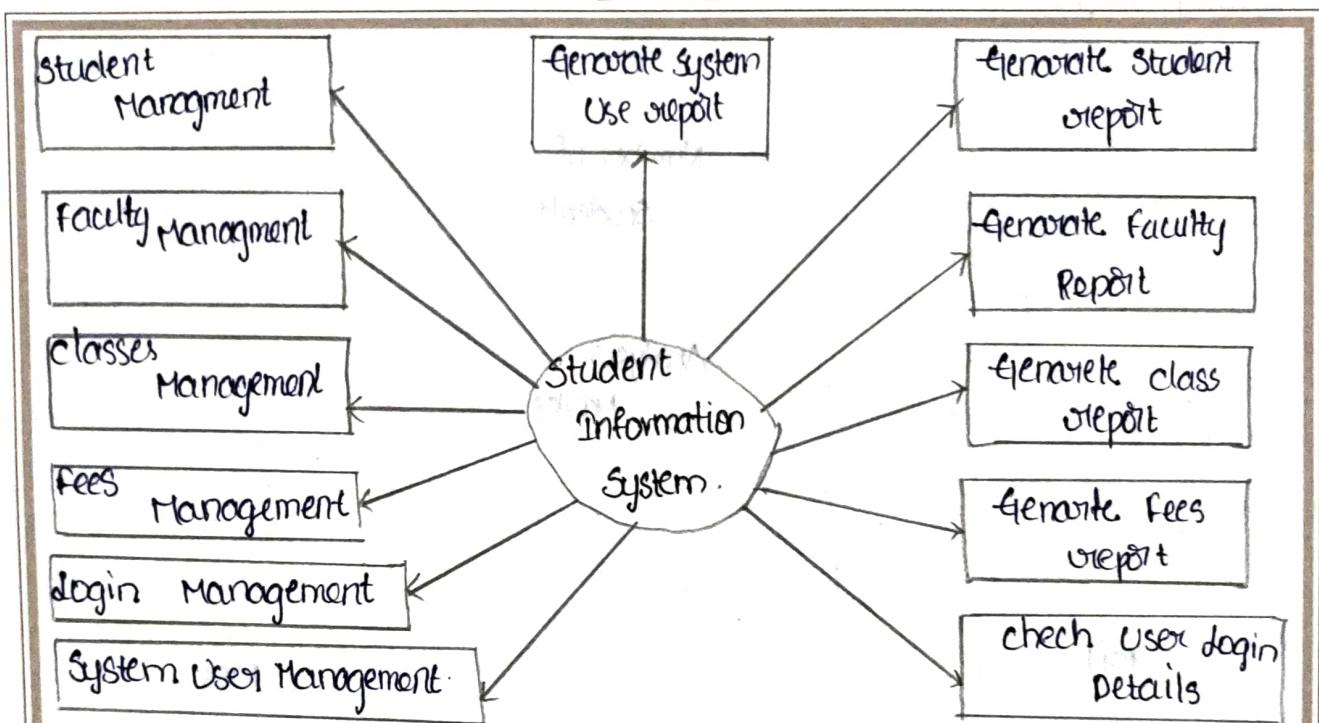
→ collaboration Diagram :-



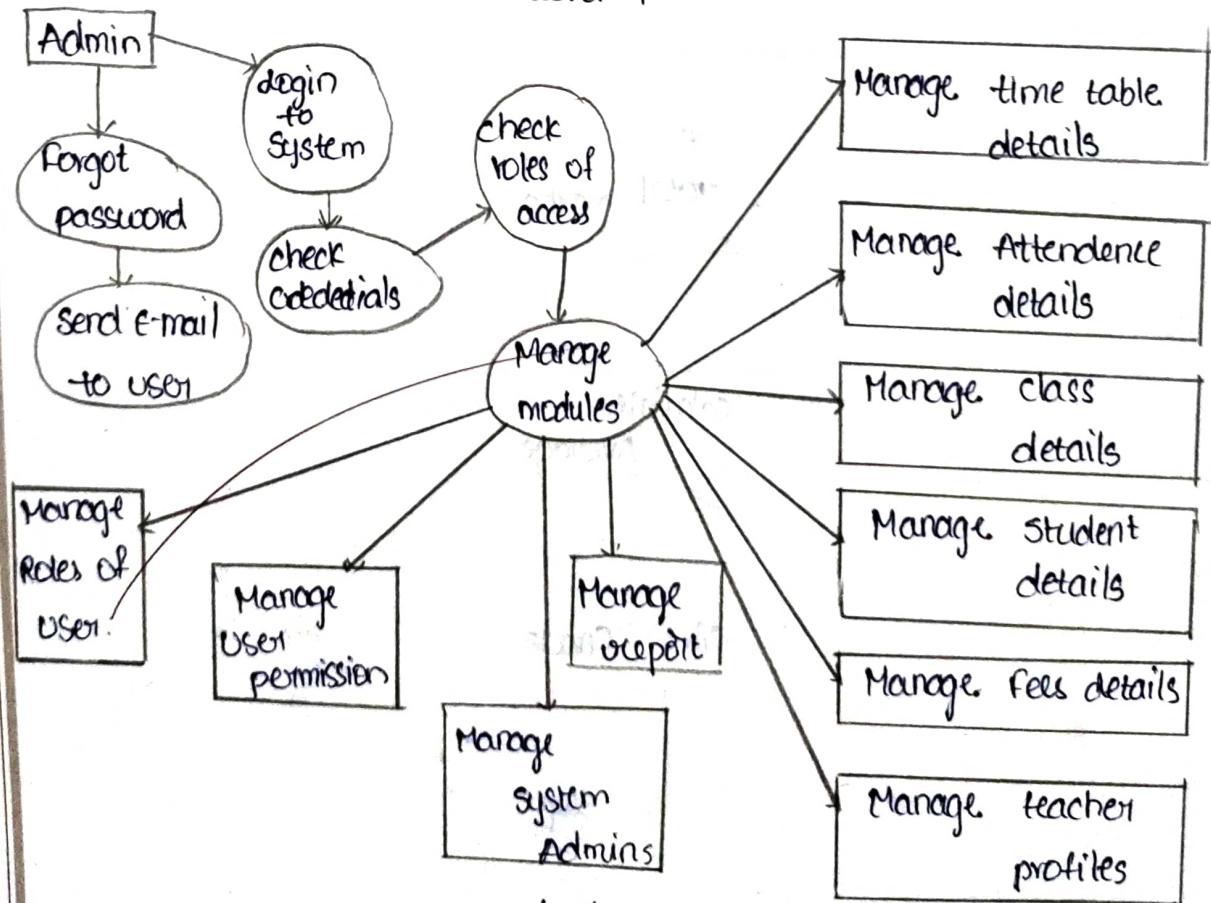
Data flow Diagram :



devel - 0

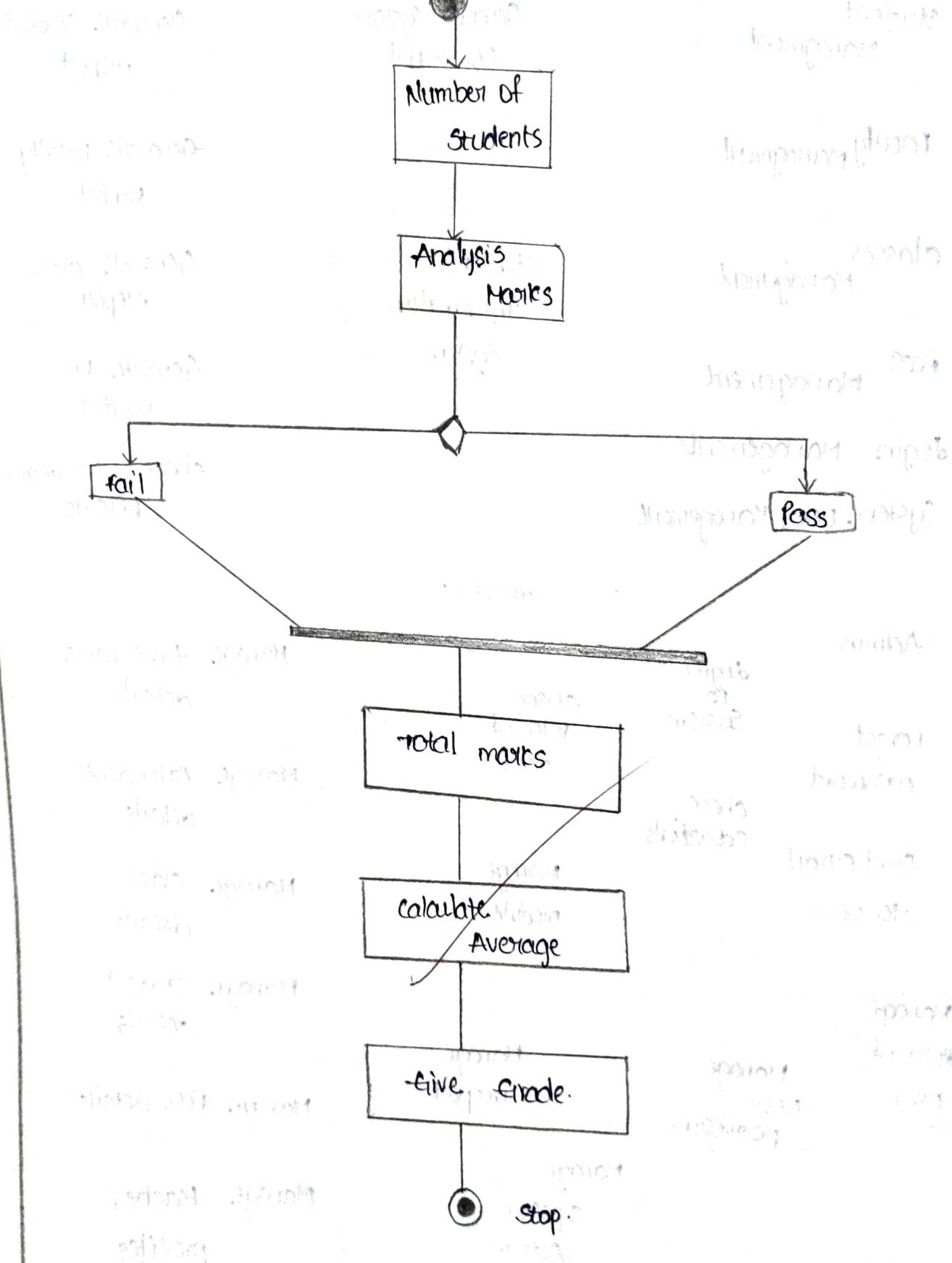


level - 1



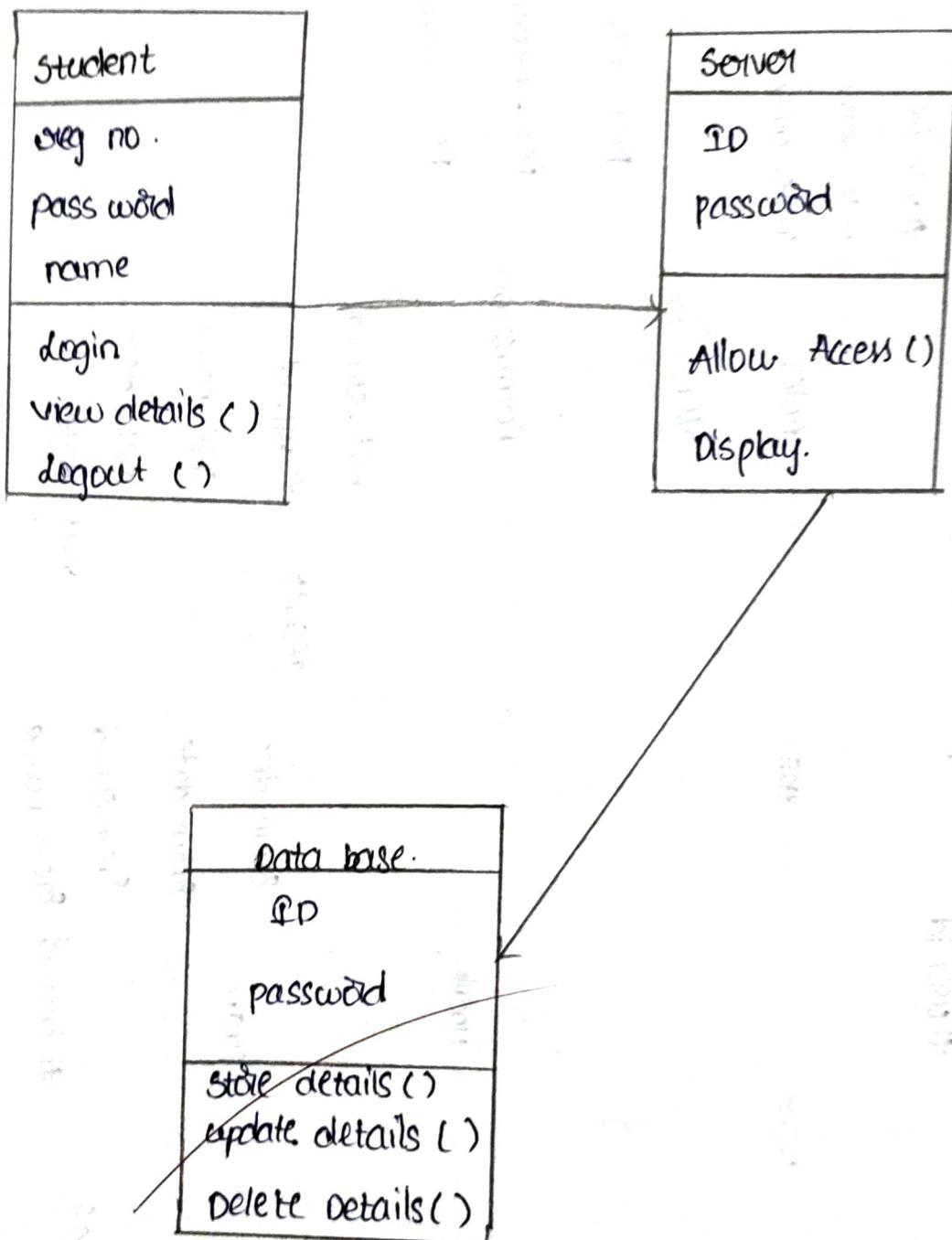
level - 2

## Activity diagram :-

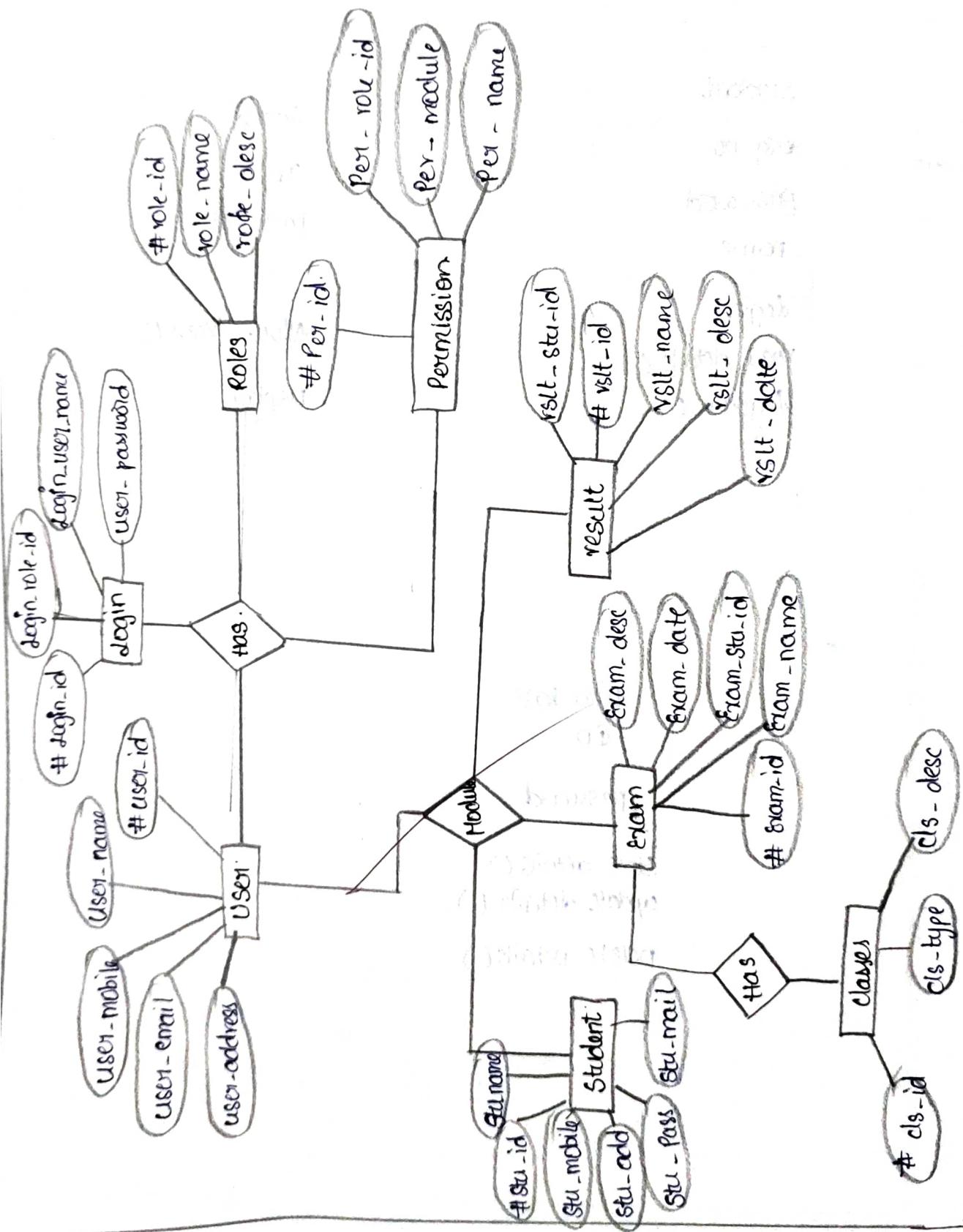




## class Diagram :-



## Entity - Relationship (E-R) Diagrams :





### 3.2 External Interface Requirements:

#### → User Interface :-

The user is going to interact with the system through different interface. Listed below are the different components of user interface listed below the respective headings.

- Faculty log-in
- Entering or updating marks process
- Admin login
- Generating progress report process
- View and print progress process.

#### → Hardware Interface :-

The system will run on different hardware gadgets. Below are the minimum hardware requirements for the smooth running of the system

- 1GB RAM PC
- 14<sup>th</sup> color monitor
- 120 fCB +HDD CPU
- Proper functioning internet.

#### → Software Interface :-

Data base - Oracle

Operating system - Any OS like Windows 7, Linux, Mac OS

front end - HTML, JAVA Script.

#### → Communication Interface :-

The month communication link that the system will be using the Internet.

### 3.3 System Features:

The STUDENT MARKS ANALYSIS SYSTEM comprises of two main features, namely internet connectivity which will enables users to communicate with the server through a browser & web agent, and secondly the system requires data base service to store the user's data. In a nutshell this system is web application and this only operational.

### 3.4 Non-functional Requirements:

→ Performance:

- The system is required a fair amount of speed especially while browsing.
  - The database shall be able to accommodate a minimum of 10000 students
  - the software will support multiple users ,with their respective amounts.

→ safety :

It is required provide a protection for the marks data base of the university.

→ Security:

~~The main security concern is for users account hence. proper login mechanism should be used to avoid hacking. the student marks analysing system shall not disclose. personal information of students to unauthorised user or the public.~~

→ Software Quality Attributes :-

**Availability** :- program reports can be viewed and printed at any time

i.e., 2417.



### → Business Rules :

- The Student movies Analyzing system shall include two types of accounts: the administration and the faculty.
- To login to the System user name and password is required. User name shall be the faculty id and the password as they prefer. Same for admin login.

N. Praveen



# Software Requirements specifications SRS Document for

## online ticket reservation System

### 1. Introduction:

#### 1.1 Purpose:

This document is intended for the following group of people.

- Developers for the purpose of maintenance and updates of the website.
- Documentation writers.
- Management and tests.

#### 1.2 Intended audience:

- Members are given a provision to check their accounts information and change it.
- User can search presently available shows & upcoming shows in different theaters.
- users can book the required number of seats and then pay for them.

#### 1.3 Intended use:

The electronic ticket replaces the paper ticket and allows customers to purchase, change, and refund transportation transactions over telephone. Eliminating the need to visit the airport ticket counter.

#### 1.4 Scope:

This document applies to e-ticketing website of a movie theater. This software is for designing tickets, managing reservations and creating a unique bar code for every ticket. It allows the user to book a ticket for a movie at a linked theater and wished time.



the software takes a input. the input e-mail id or phone number for a primary verification to create an account . the software produces an e-ticket as an output . the user can download it in a PDF formate . the software is expected to complete in duration of 1 Month and the estimated cost in RS 10 Lakhs.

### 1.5 Definitions and Acronyms :

- Software Requirements specifications
- PDF - Portable Document formate
- FTP - File transfer Protocol.

### 2. Overall Description :

#### 2.1 User Needs :

- User needs the following things to book an online reservations
- voter photo identity card issued by Election commission of India
  - Pan card issued by income tax Department
  - photo identity card having serial number issued by central / state Government.
  - Students identify card with photograph issued by recognized School / college for their students.

#### 2.2 Assumption And Dependencies :

The requirement stated in the SRS could be affected by the following factors:

- One major dependence that the project might face. the changes that need to be incorporated with the change in the policies regarding different services . As policies changes the system.



- Another constraint related to the operating environment is that we are specific to oracle data base.
- The projected could be largely affected if two different user book same sent at a same time. Such condition shall be taken off.
- At this stage no quantitative measurements are imposed on the software in terms of speed and memory although it is implied that all functions will be optimized with respect to speed and memory. It is further more assumed that the scope of the package will increase considerably in the future.

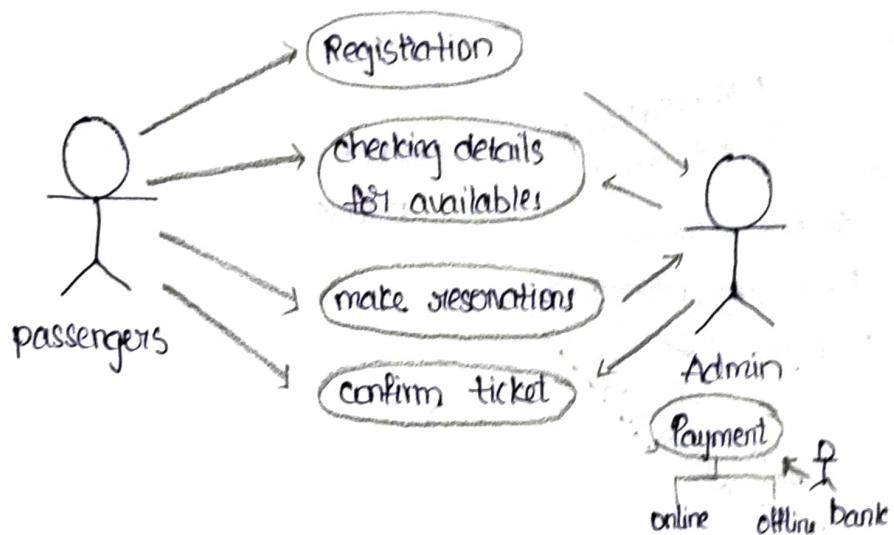
### 3. System features and Requirements:

#### 3.1 functional Requirements:

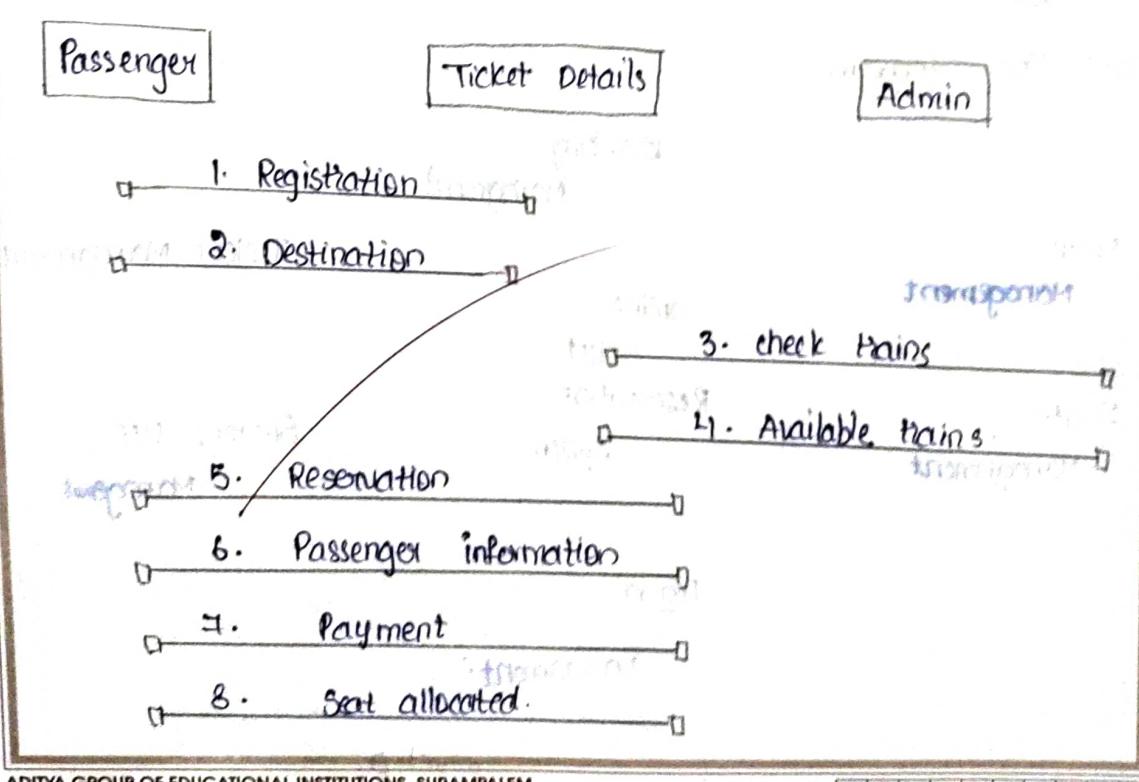
UML diagrams:

- usecase diagram
- sequence diagram
- collaboration diagram

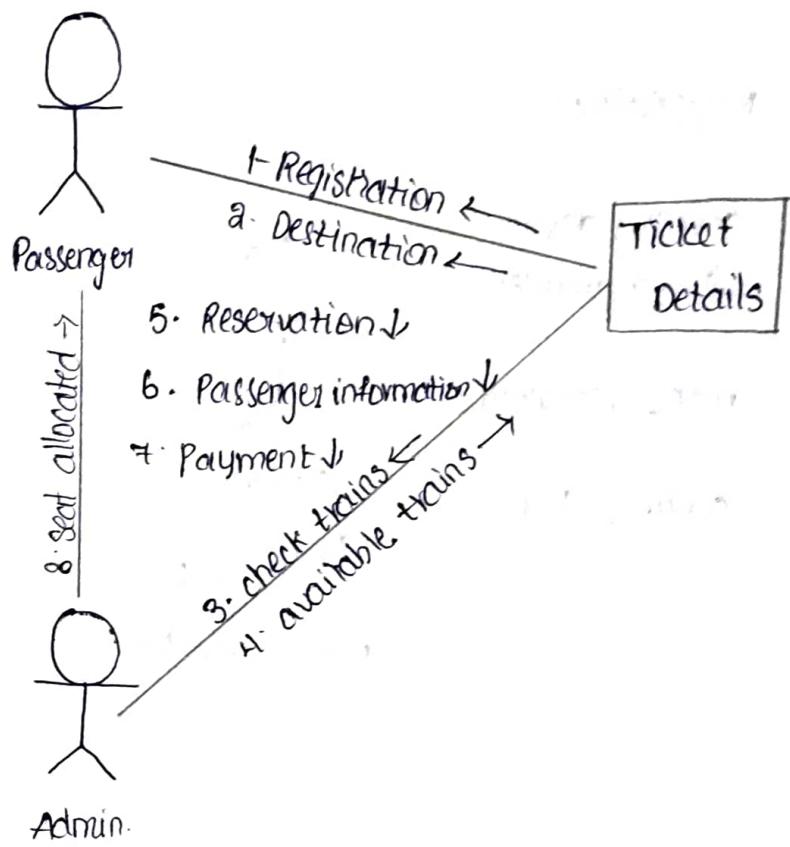
→ use case diagram:



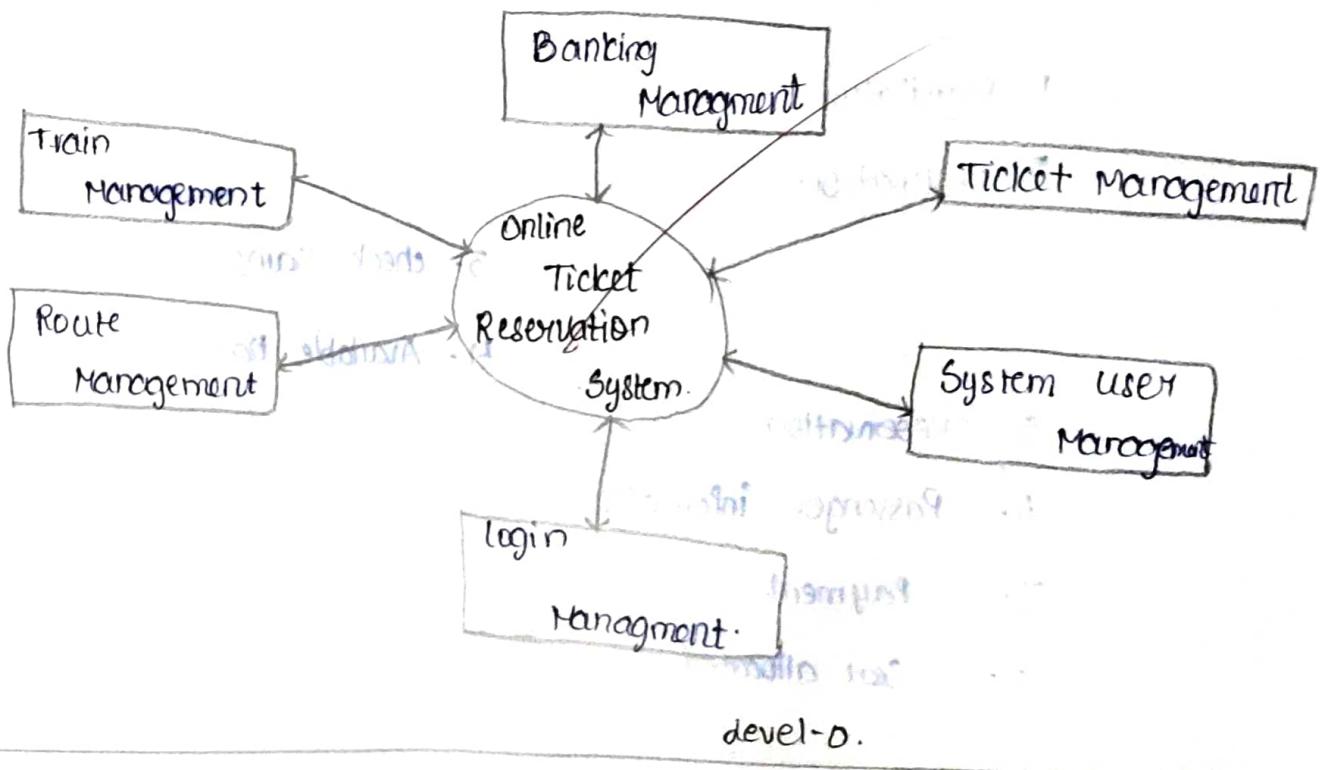
→ Sequence Diagram:

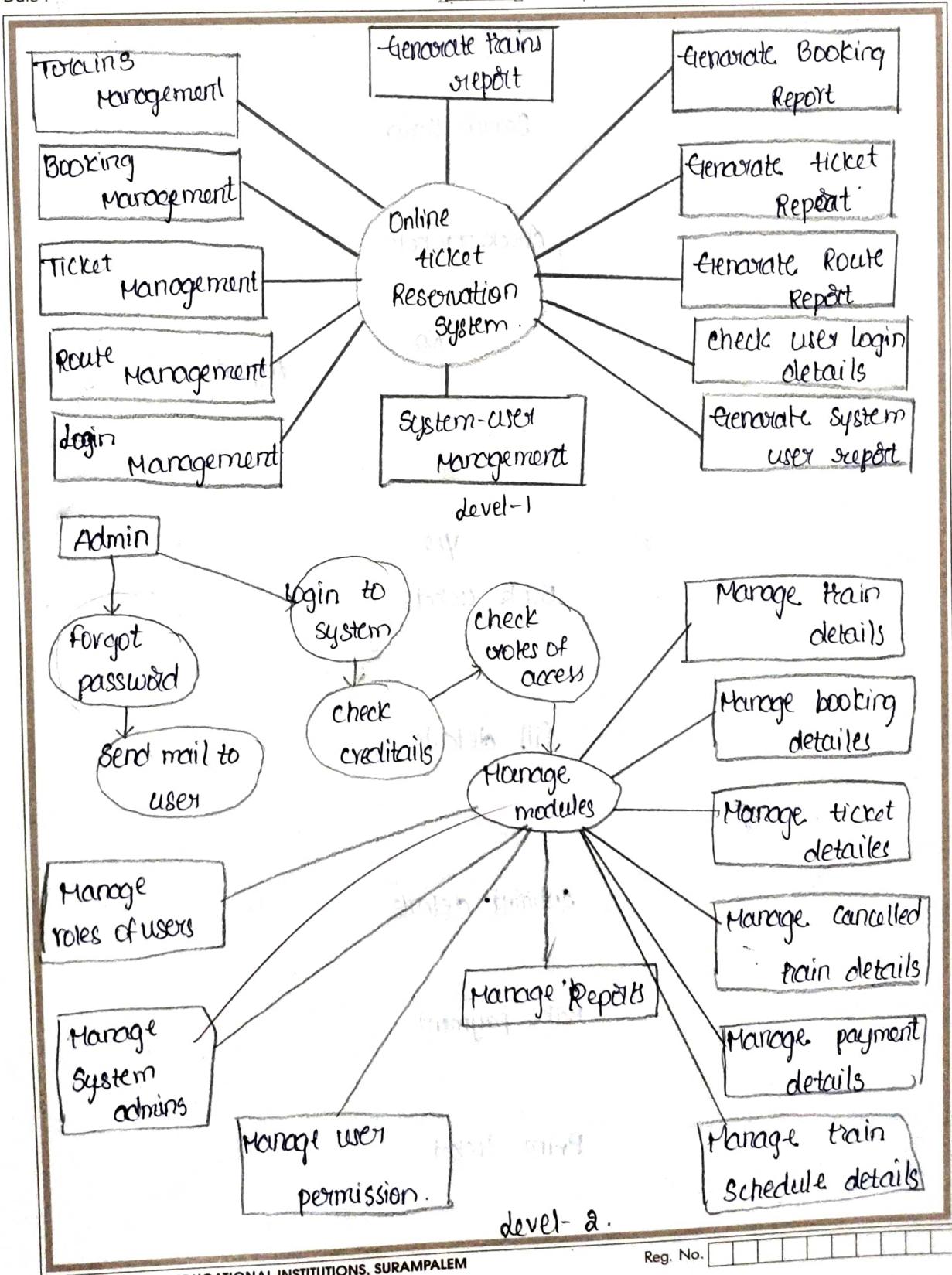


→ collaboration diagram:

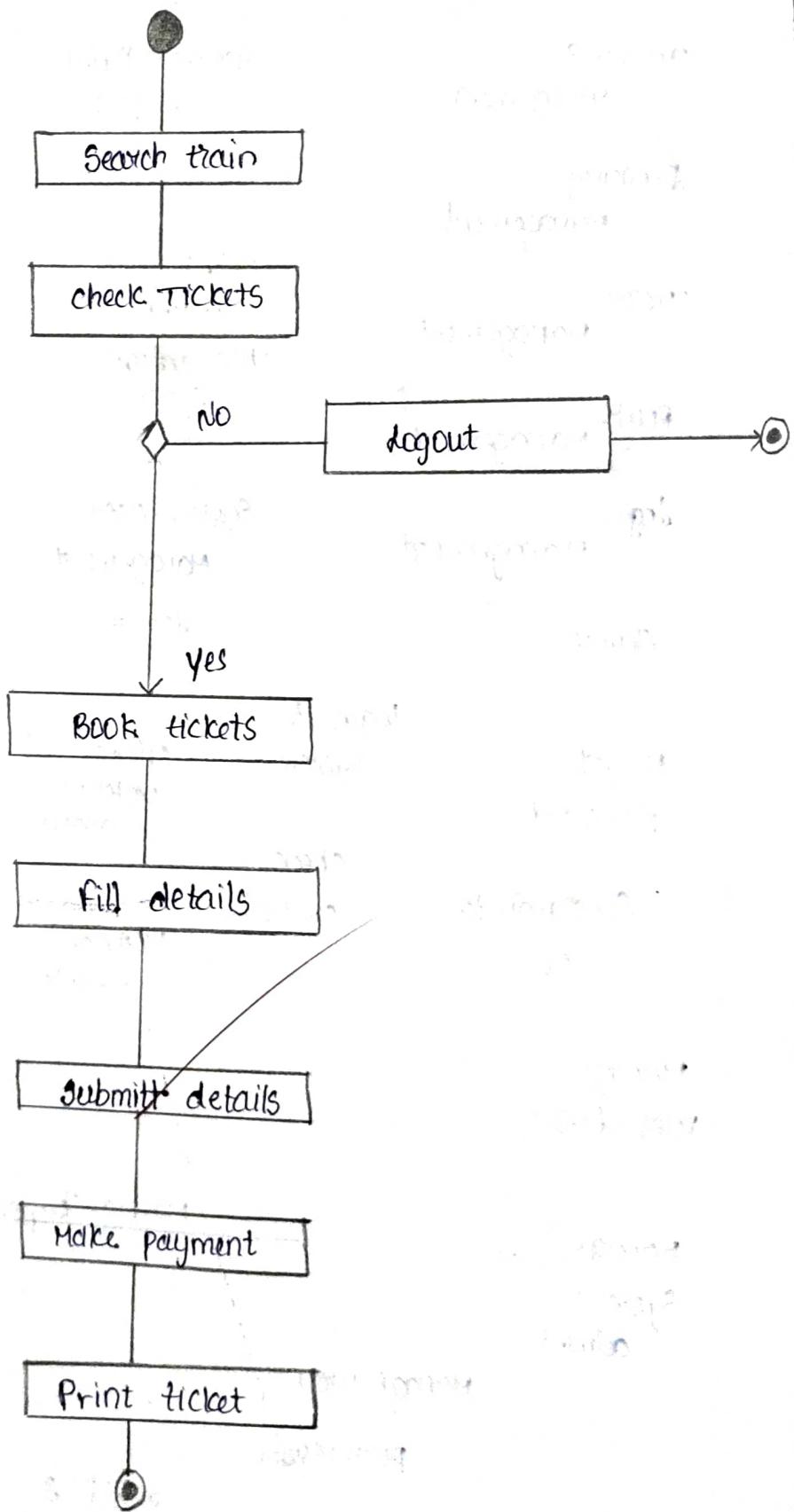


Data flow diagram:





## Activity diagram:



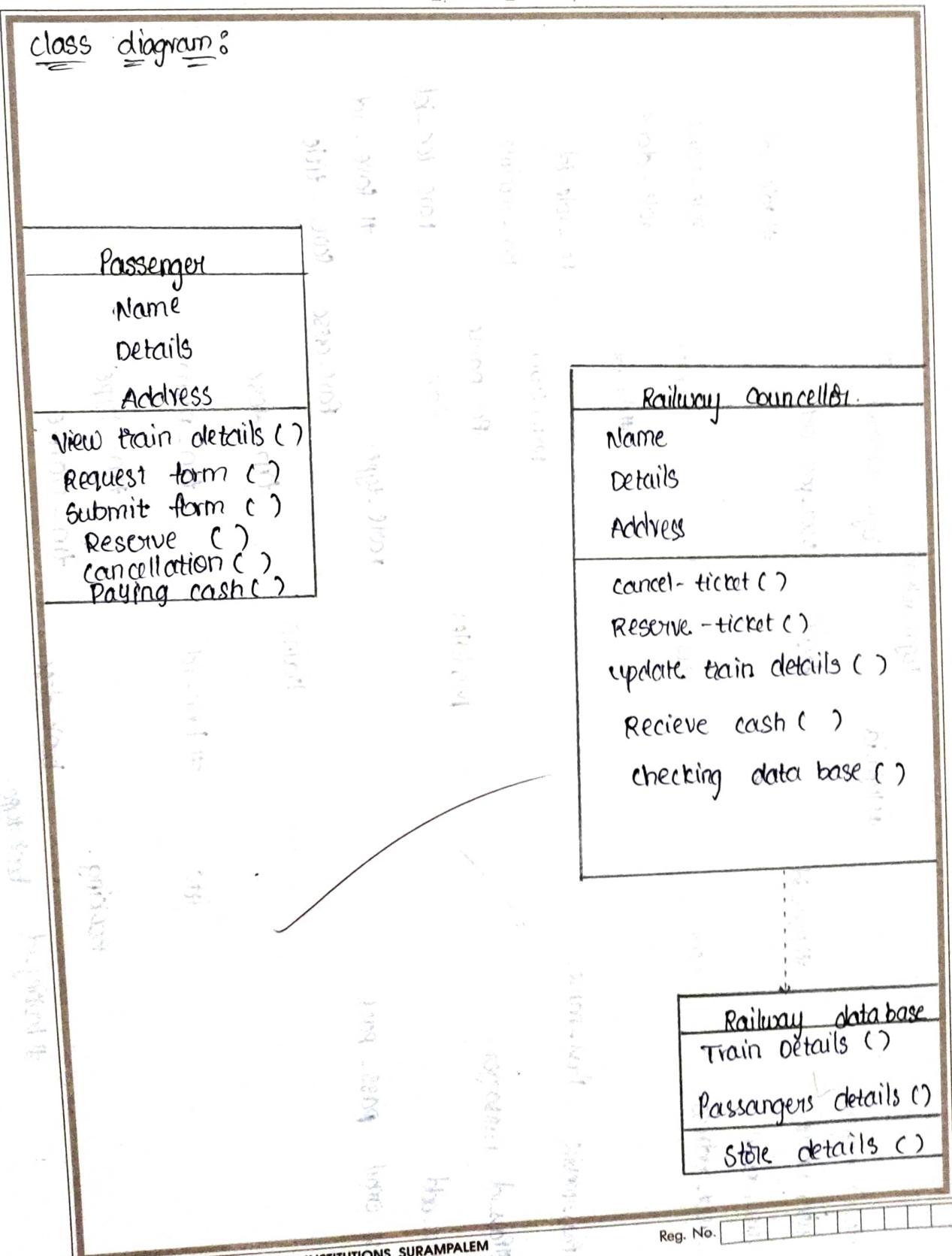
Expt. No. :

Date :

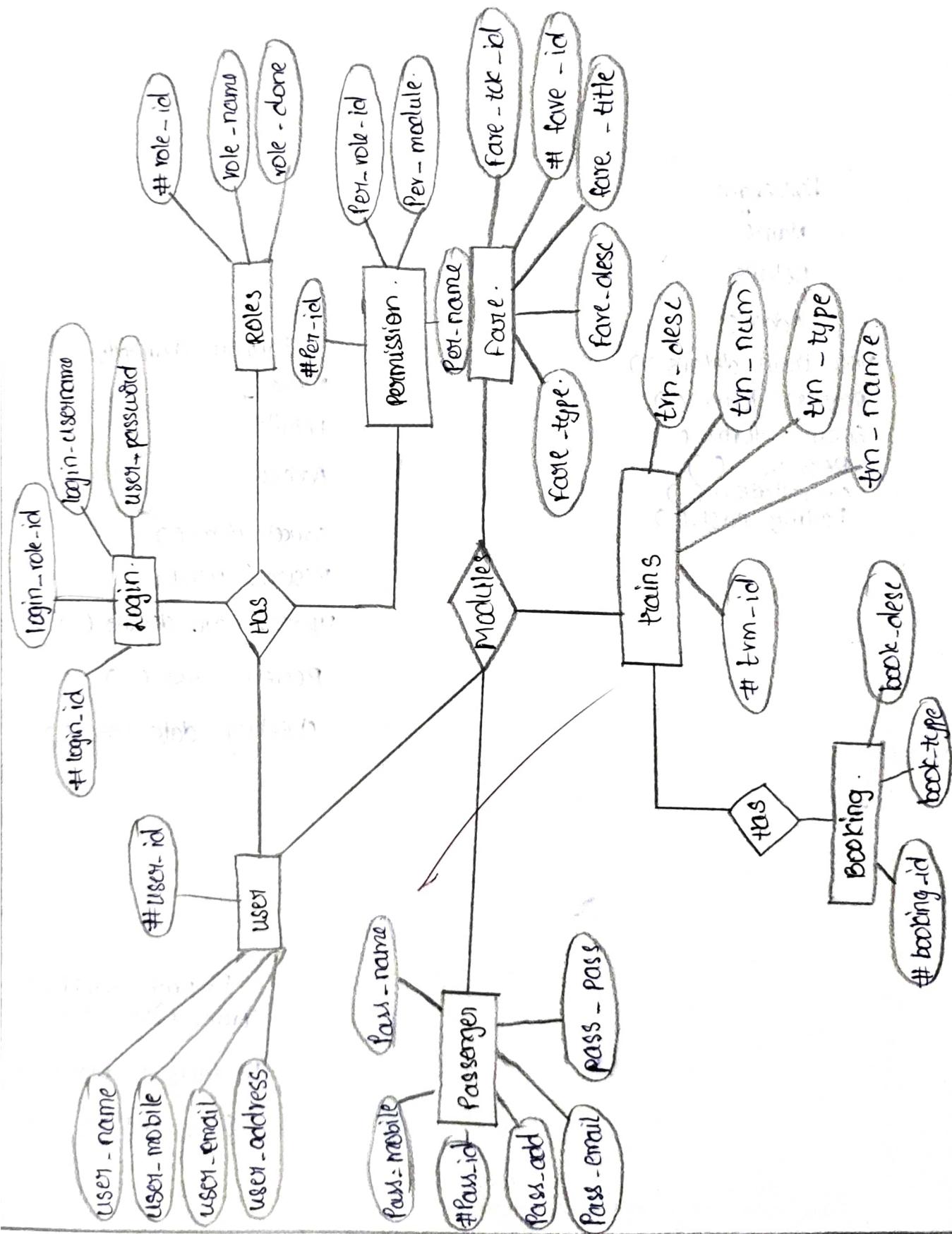


Page No. : 21

## class diagram:



## Entity Relationship (E-R) diagram :-



### 3.2 External Internal Requirements :-

#### → User Interface :-

- The login screen is provided in the begining for Entering the required user/name /Pin no. and account number
- In case of administration , a screen will be shown having option to reboot system, shut down system, disable any services.

Incase of reboot/ shut down , a screen is displayed to confirm the user's will to reboot and also allow the user to take any backup if needed.

#### → Hardware Interface :-

- the ATM power supply shall have a 101220 VAC manual switch
- the card reader shall be a magnetic strip reader
- there shall be a 40 column dot matrix receipt printer
- the slot for a card in the card reader include an extra indentation for the embossed area of the card.

#### → Software Interface :-

The transaction management software used to manage the transaction and keep track of resources shall be BMS version 2.0. the database used to keep record user account shall be oracle version 7.0.

#### → communication Interface :-

The transaction management software used to manage the transactions and keep track of resources shall be BMS version 2.0. the database used to keep record user account shall be oracle version 7.0.

### 3.3 System Features :

- It is an easy registration process.
- It has more multiple languages & currency options.
- Easy payment is possible.
- Data ownership Event copy features
- It gives reserved seating
- customized drag-and-drop registration form builder
- Inbuilt marketing tools.

### 3.4 Non-Functional Requirements :

#### → Performance :

The following list provides a brief summary of the performance requirements for the software.

i) Capacity : The booking portal shall provide customer a 24-hrs-service.

ii) Dynamic : The initial loading time must not exceed 4 secs under normal server workload and 7 sec under peak work load. Registration and login need password verification which should not exceed 2 secs to login. Time unit to search a show must not cross 3 secs.

iii) Quality : The primary objective is to provide quality software. As the quality of a piece of software is difficult to measure quantitatively, the following guideline will be used when judging the quality of software.

#### → Reliability :

- The system provides storage of all databases on redundant computers with automatic switchover.
- The reliability of the overall program depends on the reliability of the separate or distinct components.

# Software Requirement specification (SRS) Document for the

## Stock Maintenance System

### 1. Introduction:

#### 1.1 Purpose :

The purpose of stock maintenance is to reduce the costs of holding stock while ensuring you can meet customer demand and making sure that there's enough material for production.

Business should always have a 'safe' amount of stock so, that they are to react and cover any unforeseen issues.

#### 1.2 Intended audience :

The intended audience of the document are shop keeper's innovative team members, our development members, and all the business enterprise. If any changes made, it should be accounted.

#### 1.3. Intended use :

There are various kinds of users for this product. Those products are purchased via online by many customers (e.g:- shopping). User classes may be differentiated based on frequency of use.

#### 1.4 Scope :

- Effective storage of stocks
- Advanced and customized search option
- Discipline storage area
- Improved and optimized service .



### 1.5 Definition and Acronyms :

- Market data provider : one who analysis the product and distributed the news.
- customer : one who takes order of product.
- Sales person : one who maintain the stock details.

The stock maintenance system has to handle records for number of product and the maintenance was difficult though it has used an information system, it was totally manual.

### 2. Overall Description :

#### 2.1 User needs :

- Control your costs
- Improve your delivery
- Manage planning and forecasting
- Reduce the time for managing stock.

#### 2.2 Assumption and dependence :

- There is a known, continuous and constant demand.
- costs are known and constant
- shortage is not permitted
- Bundle product
- catalog
- catalog inventory (legally)
- configurable Product
- EAV (Entity - Attribute - Value)
- grouped product
- Input - Export
- sales

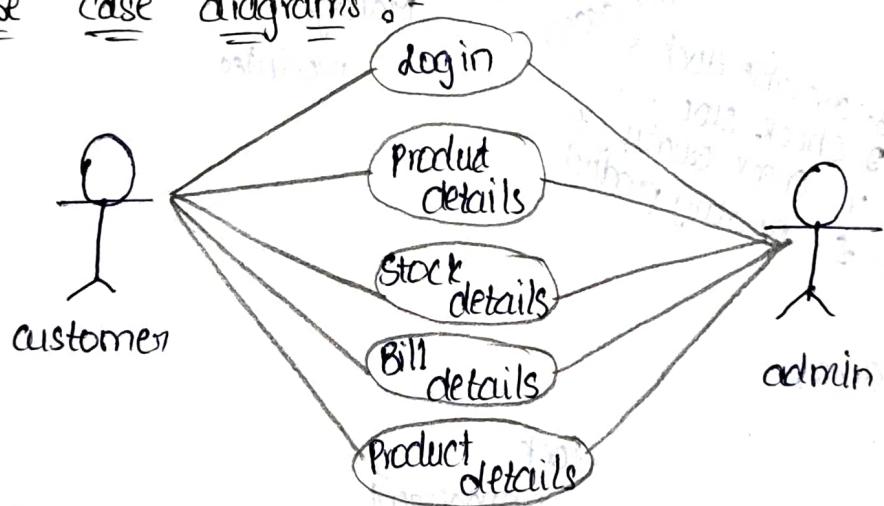
### 3. System features and Requirements

#### 3.1 functional Requirements :-

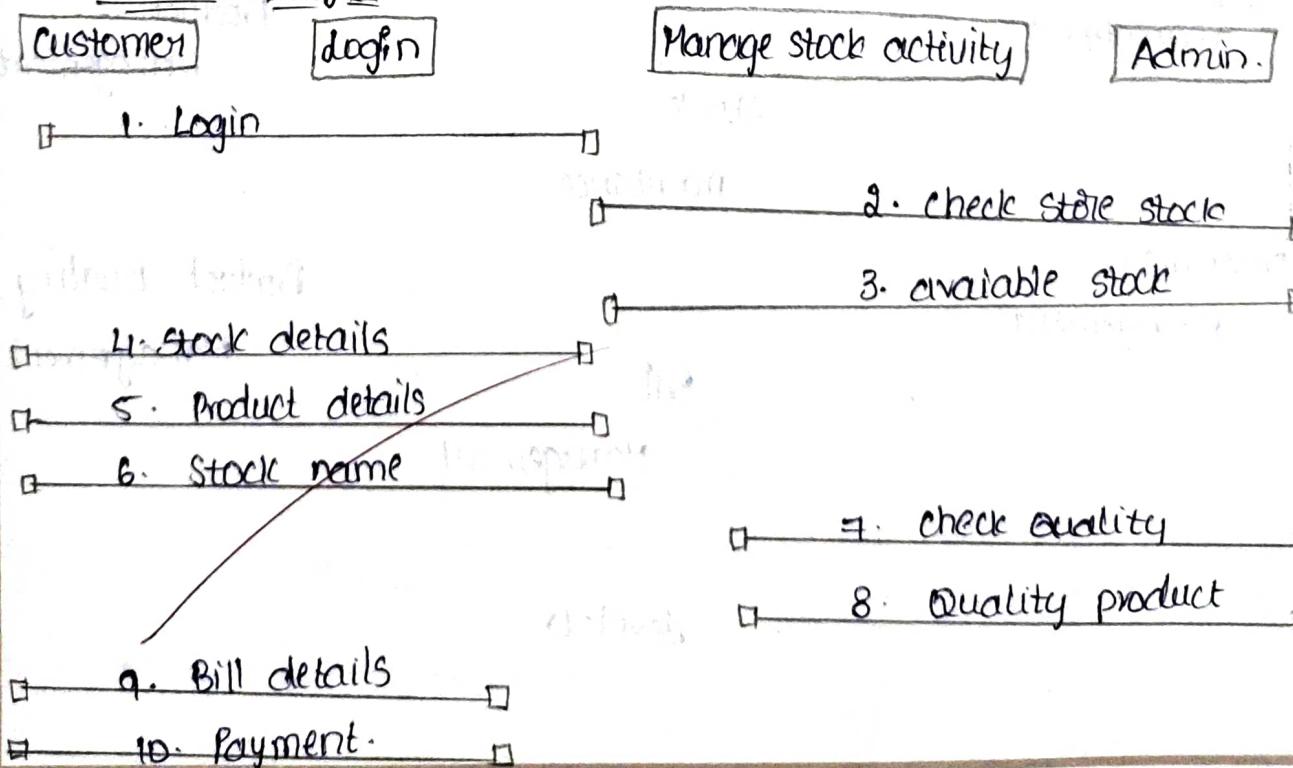
##### UML diagrams :-

- Use case diagrams
- Sequence diagrams
- collaboration diagrams.

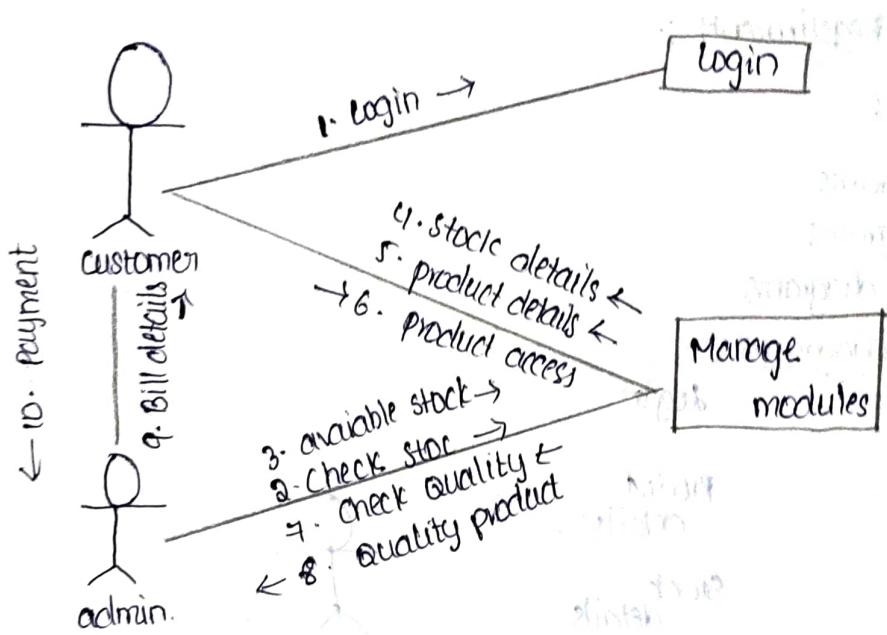
→ Use case diagrams :-



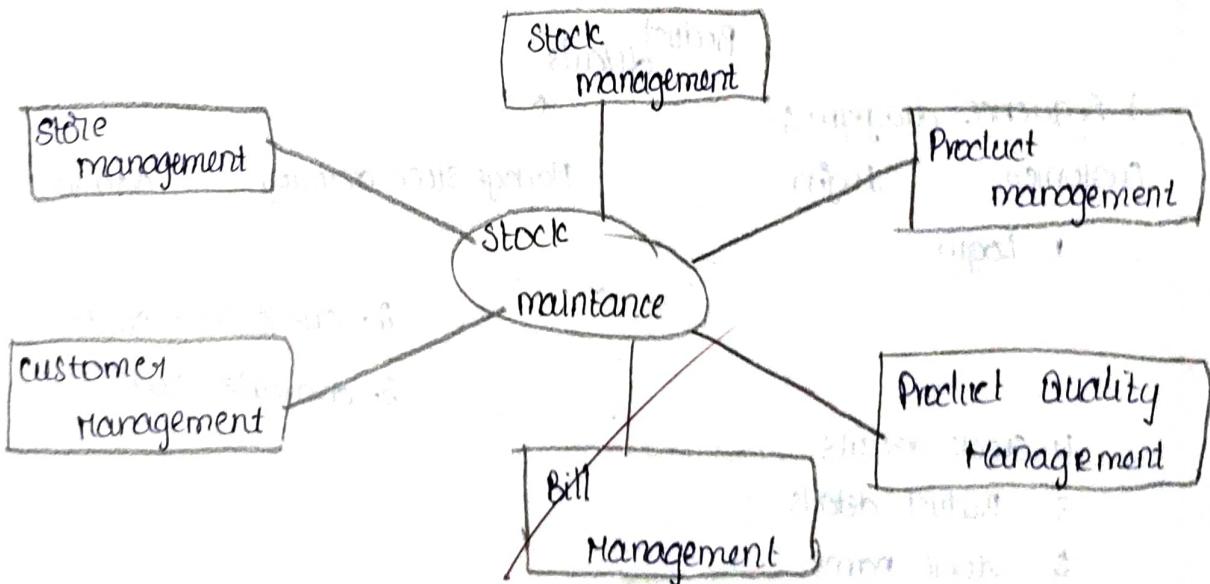
→ Sequence diagram :-



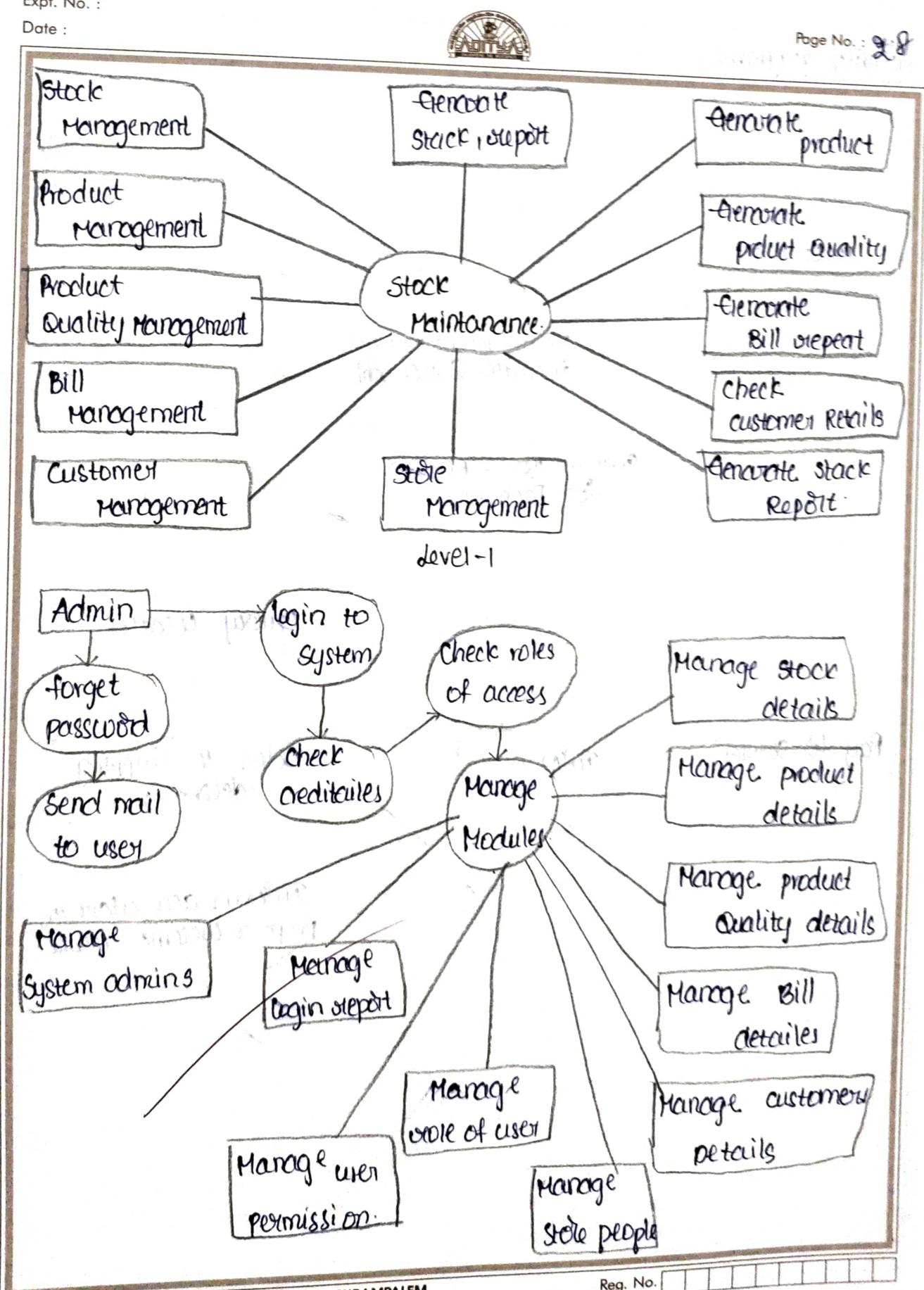
→ collaboration diagram :-



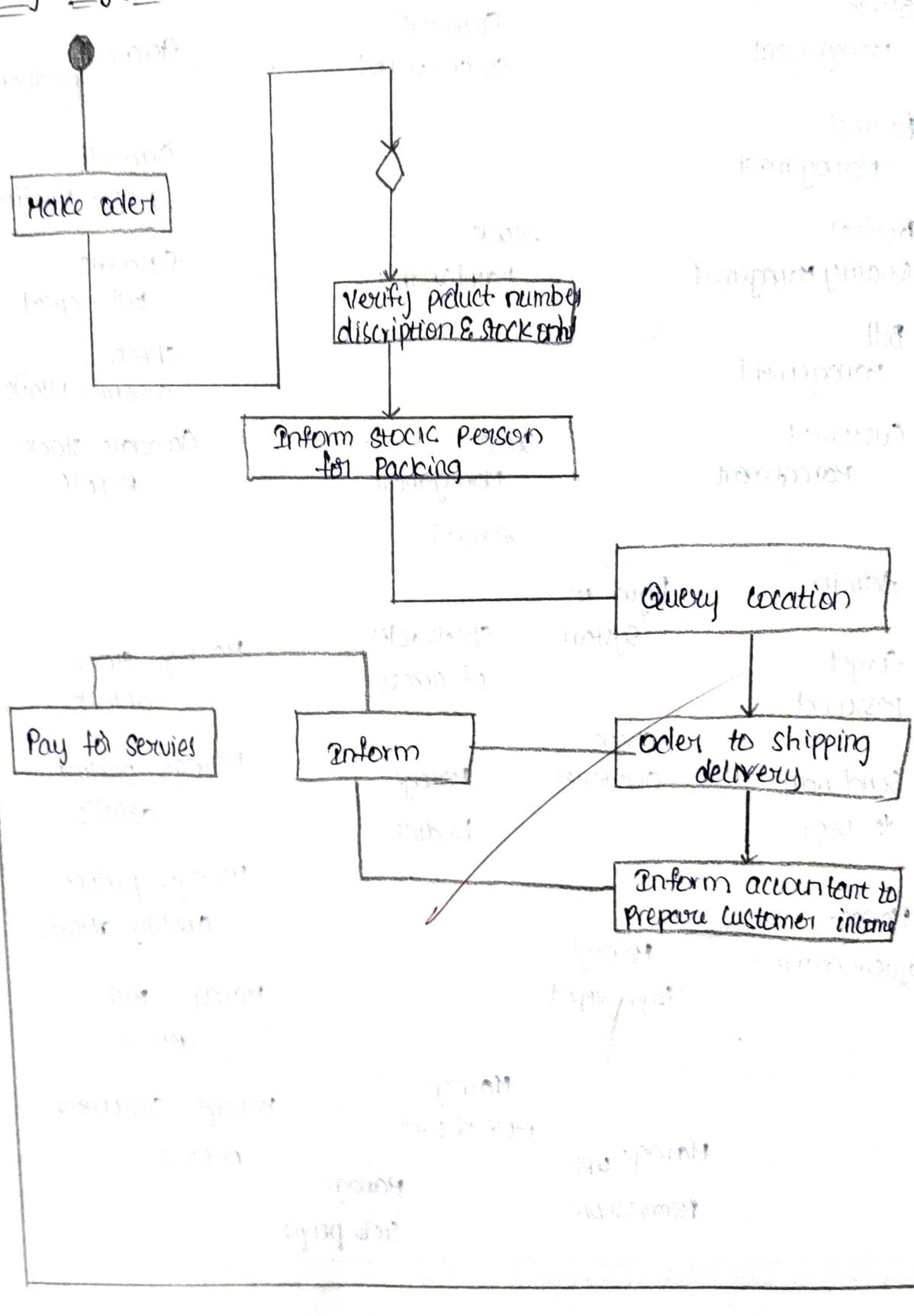
Data flow diagrams :-



level-0

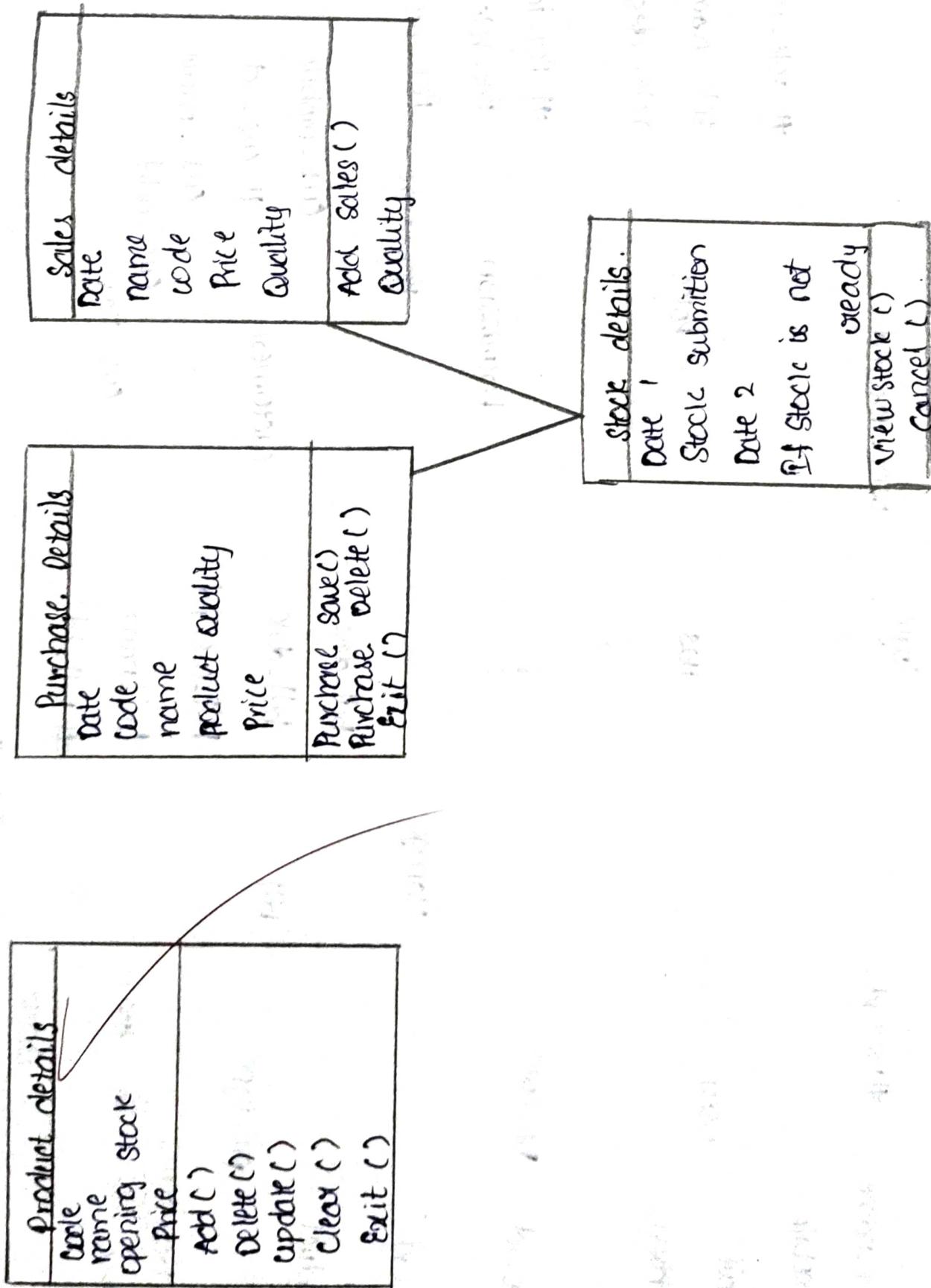


## Activity diagrams :-

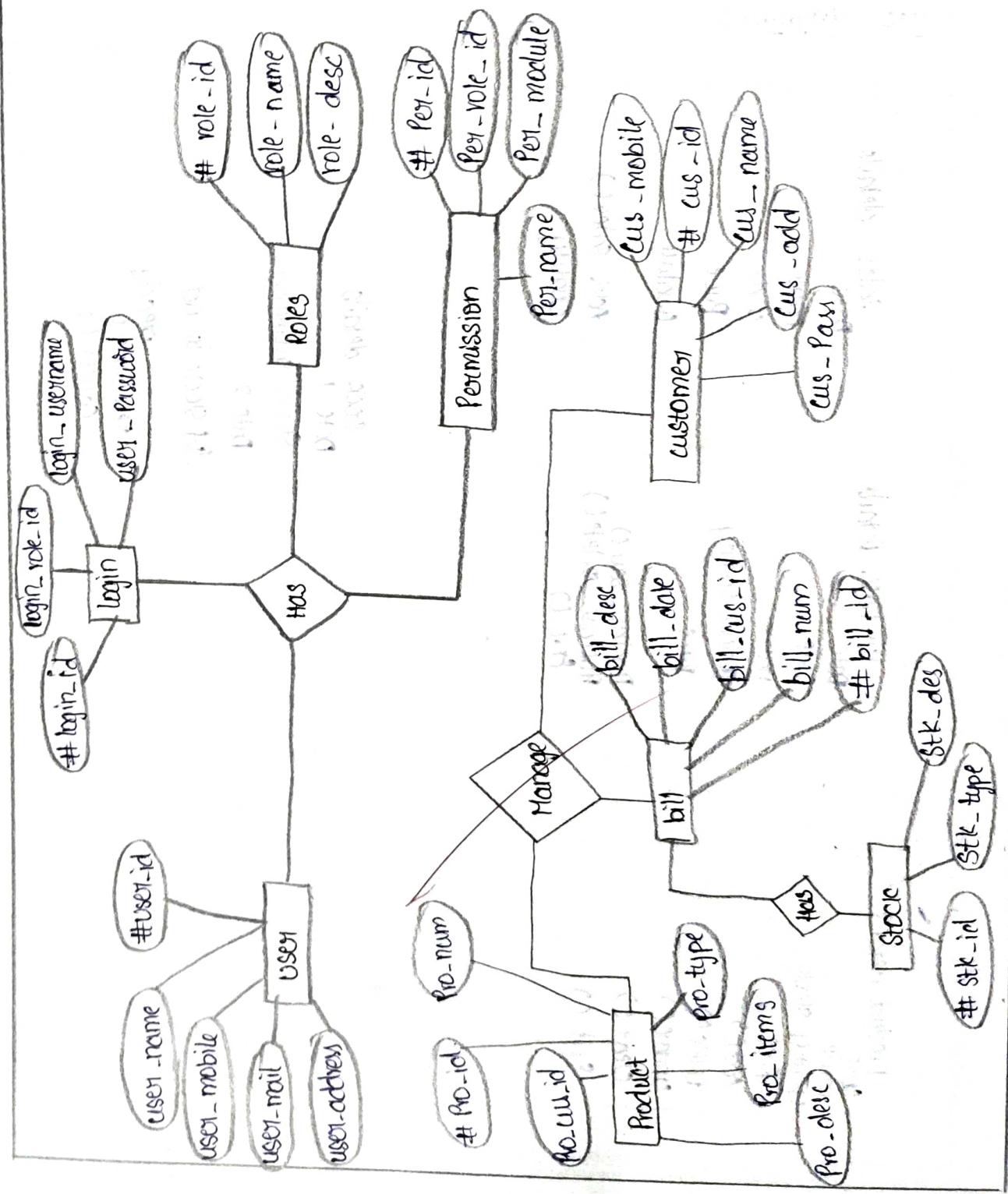




## class diagram :-



## Entity Relation (ER) Diagram :-





### 3.2 External Interface Requirements

→ User Interface:

USER NAME:	<input type="text"/>
PASSWORD :	<input type="password"/>
<input type="button" value="OK"/>	<input type="button" value="CANCEL"/>

→ Hardware Interface: A hardware stock management is the storing and tracking of hardware assets and it inventory in an organisation hardware stock software is a tool that automatically track and fetch all hardware data and provides administrators with a snapshot.

→ Software Interface:

Front End	Java
Back End	MySQL 5.1.36

When invalid inputs are given to the modules then the error message will be popped up in order to inform the user that the input provided is not taken by the database. When the incomplete information is provided by the user and the user tries to submit the form in order to store the details in the database the system will popup a message box using the user to enter all the details required.

→ Communication Interface: - The machine will have to be part of the college local area network to access the central data base.

### 3.3 System Features

The most important stock maintenance software features

- Real time stock Backing.
- Real time inventory value.
- Reorder points & low stock alerts
- Purchase management & supplier management
- unlimited SKUs
- Proper Inventory reporting
- Assemblies
- Bills of Material (BOM) management.



### 3.4 Non-functional requirements:

#### → Performance:

Easy tracking of records and updating can be done. All the requirements related to performance characteristics of the system are specified in the section below.

i. static Requirements: these requirements do not impose any constraint on the execution characteristics of the system. they are

- Number of terminals: the software makes use of an underlying data base that will reside at the same system while the first and will be available to the administration computer.
- Number of users: the no.of. users can be administrator only. but this software can be extended application for almost all staff member of the organization.

ii. Dynamic Requirements: these specify constraints on the execution characteristics of the system. they typically include response time and throughout the system

→ Reliability: The software will not be able to connect to the database in the event of the server being down due to hardware or software failure.

→ Availability: The software will be available only to administrator of the organization & the product as well as customer details will be provided by him.

→ Security: The security requirements deal with the primary security the software should be handled only by the administrator and authorized users. only the administrator has right to create new accounts and generating stock only authorized .

→ Maintainability: Backups ~~for a database~~ are available

→ Portability: The software is a web-based application and is built in JAVA & SQL. so it is platform independent and thus independent of operating System.



\* consider any application using COCOMO model, estimate the effort.

#### \* COCOMO Model :-

→ COCOMO is constructive cost estimation model

→ COCOMO is one of the most generally used software estimation model in the world. COCOMO predicts the effort and schedule of a software product based on the size of the software.

→ COCOMO is a regression model based on LOC, i.e.; number of lines of code. It is a procedure cost estimate model for software practise and often used as a process of reliability predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

#### \* Effort & Schedule :-

→ The key parameter which define the quality of any software products, which are also an outcome of the COCOMO are primarily Effort & Schedule.

→ Effort: Amount of labour that will be required to complete a task. It is measured in person months units.

→ Schedule: simply means the amount of time required for the completion of the job which is of course, proportional to the effort put. It is measured in the unit of time such as weeks, months.

\* Categorized into three : In case, projects are categorized into three types:-

- 1) organic
- 2) semi-detached
- 3) Embedded



\* 1) Organic :-

→ A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and helps also the team members have a nominal experience regarding the problem.

→ Examples of this type of project are simple business system, simple inventory management systems, and data processing system.

\* 2) Semidetached :-

→ A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed.

\* 3) Embedded :-

→ A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, & if the stringent regulation on the operational method exists.

Example :- ATM, AIR traffic control.

\* Types of models :-

→ COCOMO consists of a hierarchy of three increasingly detailed and accurate form. Any of the three forms can be adopted according to our requirements.

1) Basic COCOMO model

2) Intermediate COCOMO model

3) Extended COCOMO Model.



### → Basic COCOMO Model :-

It is the one type of static model to estimate software development effort quickly and roughly. It mainly deals with the number of lines of code and the level of estimation accuracy is level as we don't consider the all parameter belongs to the project the estimate effort and scheduled time for the project are given by the relation.

$$\text{Effort } (E) = a^* (KLOC)^b \text{ MM.}$$

$$\text{Scheduled time } (D) = c^* (E)^d \text{ Months (M)}$$

Where,

- E = total efforts required for the project in MM  
(MON - Months).

- D = total time required for project development in  
months (M).

- KLOC = the size of the code for the project in kilo  
lines of code.

- a, b, c, d = the constant parameters for a software project

\* Average resource size = EID (efforts / scheduled time)

\* productivity of software = KLOC/E

\* constants :-

Project type	a	b	c	d
organic	2.4	1.05	2.5	0.38
Semidetached	3	1.12	2.5	0.35
Embedded	3.6	1.2	2.5	0.32

### → Intermediate COCOMO model:

The intermediate model estimate software development effort in terms of size of the program and other related cost drivers parameters of the project. The estimated effort and scheduling time are given by the relationship.

$$\text{effort (E)} = a * (\text{kLOC})^b * \text{EAF MM}$$

$$\text{scheduling time (D)} = c * (\text{E})^d \text{ months (M)}$$

Where

- E = total effort required for the project in man-months(MM)

- D = total time required for project development in months(M)

- KLOC = kilo lines of code

- a, b, c, d = the constant parameters.

- EAF = It is an effort adjustment factor, which is calculated by multiplying the parameter.

values of different cost driver parameters - for ideal, the value is,

### \* Product parameter:

Project type	a	b	c	d
organic	0.75	1.03	2.5	0.58
Semidefected	3	1.12	2.5	0.35
Embedded	8.6	1.2	2.5	0.82



### → Detailed COCOMO model :

Detailed COCOMO incorporates all qualities of the standard version with an assessment of the cost drivers effect on each method of the software engineering process. The detailed model uses various effort multipliers for each cost driver property. In detailed COCOMO, the whole software is differentiated into multiple modules, and then we apply COCOMO in various modules to estimate effort and then sum the effort.

### \* Examples :

For a given project was estimated with a size of 300 KLOC. calculate the effort, schedule time for development. Also calculate the average resource size and productivity of the software for organic project type.

For Organic project

$$\text{Given Size} = 300 \text{ KLOC}$$

$$\text{Effort (E)} = C * (\text{KLOC})^B \text{ MM}$$

$$= 2.4 * (300)^{1.05} \text{ MM}$$

$$= 2.4 * 399 \text{ MM}$$

$$= 987.6 \text{ MM}$$

$$D = C * E^d \text{ M}$$

$$= 2.5 * (987.6)^{0.38} \text{ M}$$

$$= 2.5 * 13.57 \text{ M}$$

$$= 33.94 \text{ M}$$

$$\text{Productivity} = \text{KLOC / E}$$

$$= 300 / 987.6$$



$$\begin{aligned}
 &= 0.3132 \text{ kLOC} \\
 \text{Resource} &= 313.2 \text{ LOC} \\
 &= E/10 \\
 &= \frac{957.6}{33.94} = 28.21
 \end{aligned}$$

for a given project was estimated with a size of 300 kLOC. calculate the effort, schedual time for development by considering development having high application experiance and very low experiance in programming.

In this problem, we consider intermediate cocotto model -1

$$\text{Effort (E)} = a^k (\text{kLOC})^b \times \text{months (M)}$$

$$\text{application} = 0.32$$

$$\text{programming experiance} = 1.14$$

$$EAP = 0.32 \times 1.14 = 0.3648$$

$$\begin{aligned}
 E &= 2.4 \times (300)^{1.05} \times 0.3648 \\
 &= 903.39
 \end{aligned}$$

$$\begin{aligned}
 D &= 2.5 \times (903.39)^{0.38} \\
 &= 33.20
 \end{aligned}$$

$$\cancel{\text{Effort}} = 903.39$$

$$\cancel{\text{schedual time}} = 33.20.$$

~~suppose a project was estimated to be 400 kLOC. calculate the effort and development time for each of the three model i.e; organic, semi-detached & embedded.~~

## Basic

$$E = a^* c (KLOC)^b MM$$

$$D = C \otimes \epsilon^d M$$

Production = KLOC'E

Resource = EID

$$\text{Organic : } E = 2.4 \times (400)^{1.05} \text{ MM}$$

$$= 2.4 \text{ at } 539.71$$

= 1295.31

$$D = 2.5 * (1295.31)^{0.38} M$$

= 2-5 \* 15-25

$$= 38 \cdot 07 M$$

$$\text{Productive} = 4001 - 1295.31 = 308.80$$

$$\text{Resource} = \frac{1295.31}{38.07} = 34.01$$

Semi detached :-

$$f = 34 \times (400)^{1.12} - 2462.79$$

$$D = 25 * (2462 \cdot 79)^{0.35}$$

$$= 38.45$$

$$\cancel{\text{Productive}} = \frac{400}{200 \cdot 79} = 0.1624 \text{ NM} \\ = 162.4 \text{ M}$$

$$\text{Resource} = \frac{2462.79}{38.45} = 64.051$$



Embedded

$$E = 3.6 \times (200)^{1.2} = 4772.8$$

$$D = 2.5 \times (4772.8)^{0.38} = 37.59.$$

$$\text{Productive} = \frac{100}{4772.8}$$

$$= 0.083 \text{ MM} = 83.80 \text{ M}$$

$$\text{Resource} = \frac{4772.8}{37.59} = 129.96$$

- (i) A project size of 200 kloc is to be developed. software development team has average experience on similar type of project. the project schedule is not very tight. calculate the effort, development time, average staff size, and productivity of the project.

so): In this problem we consider Embedded cocotte model.

$$E = a * (kLOC)^b \text{ MM}$$

$$= 3.6 * (200)^{1.2}$$

$$= 4772.8 \text{ MM}$$

$$D = c * (E)^d \text{ M}$$

$$= 2.5 * (4772.8)^{0.32}$$

$$\text{Productive} = \frac{200}{4772.8} \text{ M}$$

$$= 0.096 \text{ MM}$$

$$\text{Resource} = \frac{4772.8}{28.81}$$

$$= 167.109.$$



6\* Consider any application calculate effort using FP oriented Estimated model.

#### \* Functional Point (FP) analysis :

- FPA is used to make estimate of the software project including its testing in terms of functionality & function size of the software
- Function Point analysis may be used for the test estimation of the point. the functional size of the product is measured in terms of the function point, which is a standard of measurements to measure the software application.
- the primary purpose of the function point analysis is to measure and provide the software application function size to the client, customer and the stock holder on their request.
- further, it is used to measure the software project development along with its maintenance, consistency throughout the project irrespective of the tools and technology.

#### \* Points regarding FP's :-

- FP's of an application is found out by counting the number and types of functions upset in the application various function upset in an application can be put under five types as shown in table:

Measurements	Parameter	Examples
1. Number of external input	(EI)	1. Input screen and tables.
2. No. of External output	(EO)	2. Output screen and reporting
3. No. of External inquiry	(EQ)	3. Promote and interrupt.



2. No. of internal field (ILF)	Database and directory
5. No. of External interface (EIF)	Shared database & shared screening

\* Functional points types:

- Transactional functional types :-
- External input (EI)
- External output (EO)
- External Enquiry (EO)
- Data functional types :-
- Internal Logical file (ILF)
- External Interface files (EIF).

\* Point regarding FP's :-

- FP characterises the complexity of the software system and hence can be used to depict the project time and the manpower requirement.
- the effort required to develop the project depend on what the software does.
- FP is Programming language independent.
- FP is method is used for data Processing Systems, business system like information system.
- the five parameter mentioned above are also known as information domain characteristics.
- All the parameter mentioned above are applied some weight that have been experimentally determined.



— the functional complexities are multiplied with the corresponding weight against each function, and the value are added up to determine the UFP (unadjusted function point) of sub system.

### Computing FP's

Measurement Parameters	Count	Weighting factor			
		Simple	Average	Complex	
1. No. of External input [EI]	-	+	3	4	6 = -
2. No. of External output [EO]	-	+	4	5	7 = -
3. No. of External inquiries [EQ]	-	+	3	4	6 = -
4. No. of internal field [INIF]	-	+	7	10	15 = -
5. No. of External interface [ELF]	-	+	5	7	10 = -

#### \* Number of user input:

Each user input that provides distinct application oriented data to the software is counted. Input should be distinguished from inquiries, which are counted separately.

#### \* Number of user outputs:

Each upset output that provides application oriented information to the user is counted. In this contexted output refers to deposits, screen, error messages etc. Individual data item within a deposit are not counted separately.



### \* Number of user inquiries:

An inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output. Each distinct inquiries is counted.

### \* Number of files:

each logical mastered file i.e; a logical grouping of data that may be one part of a large database & a separated files; is counted.

### \* Number of External interfaces:

In all machine readable interface that are used to transmit information to another system are counted. Once the data have been collected, a complexity value is associated with each count. Organizations that we function point member develop criteria for determining whether a particular entry is simple.

### \* function point (FP) calculation:

$$FP = UFP + CAF$$

UFP = UNADJUSTED FUNCTIONAL POINT

CAF = COMPLEXITY ADJUSTMENT FACTOR

$$CAF = 0.65 \rightarrow (0.61 + \Sigma(f_i))$$

$f_i$  = value adjusted factor based on respond to 14 questions.

### → function Point estimation:

- Data communication
- Distributed data processing
- Performance
- Highly upset configuration
- Transaction cycle
- online data entry
- End user efficiency.



- Online updates
- complex processing
- Reliability
- Installation cost
- Operational cost
- Multiple sites
- Facilitate change.

→ complexity adjustment factor is calculated using it as parts of processing complexity and these 14 question answered on scale of 0-5:

- 0: no influence & no importance
- 1: incidental
- 2: moderate
- 3: average
- 4: significance
- 5: Essential.

- Example - 1:-

compute the function point , productivity , documentation , cost per function for the following data .

Number of user input = 84 (avg)

Number user output = 46 (simple)

Number of inquiries = 8 (complex)

No. of files = 4 (avg)

No. of external interface = 2 (simple)

- Various proceeding complexity factors are: 1,1,0,3,3,5,4,4,3,3,2,2,4,5.



Measurement parameter	Count		Weighting factor
1. No. of External input (EI)	24	*	4 = 96
2. No. of External output (EO)	46	*	4 = 184
3. No. of External inquiries (ED)	8	*	6 = 48
4. No. of Internal files (IF)	4	*	10 = 40
5. No. of External interfaces (EIF)	2	*	5 = 10

Count - Total  $\rightarrow$  378.

So sum of all  $f_i$  (~~fp~~ 170.14) =  $4+1+0+5+5+3+4+4+3+3+2+2+4+5$   
 $= 43$

*N. Jammal*

$$\begin{aligned}
 FP &= \text{Count} + \frac{\text{total}}{\text{Count}} * [0.65 + 0.01 + \sum f_i] \\
 &= 378 * [0.66 + 0.01 * 43] \\
 &= 378 * [0.65 + 0.45] \\
 &= 378 * 1.08 \\
 &= 4108.24.
 \end{aligned}$$



Design the test cases for e-commerce application (flipkart, amazon).

## Context:

1. General test case for E-commerce application (Amazon).

ii, Test case : verify that the user is able to navigate through all the product across different categories.

Verification: There are four types of products and each is classified based on commerce habit, price and product characteristics; convenience goods, shopping goods, speciality products, and unsought goods. Amazon bots or category specialist may make the changes if they don't agree with initial category section. so that the user is able to navigate through all the products across different categories.

Result :- success.

12. Test case: search - Verify that count of products is correctly displayed on the search result page for a particular search term.

Verification: Amazon search terms are keywords that shapers use to find products. Amazon's algorithm matches customer search with the term included in product listing - backward and compiles relevant products for each search results.

result: success

3, test case: filtering - Verify that filtering works correctly on categories pages.

Verification: Amazon filter helps you sort Amazon search result for the fastest delivery item, item availability, and unsponsored items. finding exactly what you want to order from Amazon can be a point if you need to think about delivery time availability & relevant.

result: success.



4, Test case: sorting Verify that all the sort options work correctly - correctly sort the products based on the sort option chosen.

Verification: Amazon can sort from high to low because Amazon does not allow low to high sorting. You also might be surprised that the sort takes into account shipping costs. You will be sorting, scanning and stacking packages on pallets, and helping to get customer orders ready for delivery.

Result: Success.

5, Product Buy flow test cases for Amazon website

1, Test case: Verify that the user can add to cart one or more products

Verification: Once you find the item you're looking for, click add to cart on any item's product detailed page. If you wish to add more than one item to your order, enter a new search item in the search bar.  
Result: Success

2, Test case: Verify that the user can add products to the wishlist

Verification: Amazon wish list is basically a gift registry. It's different than your regular baby or wedding register where people can create a wish list and share it with their friends and family. Being a wishlist it guarantees that they only receive gifts that they really want.

Result: Success

3, Test case: Verify that the user can successfully buy more than one product that was added to his/her cart

Verification: Buyer shopping carts can hold upto 1000 items combined for all unique items in cart. A cart can have 10 unique item. Once user finds the item they are looking for click add to cart on

Result: Success.

1) Test case: Verify that the cash on delivery option of payment is working fine.

Verification: Pay on delivery is available as a payment method for all the seller items that are fulfilled by Amazon, prime eligible and some seller fulfilled items. pay on delivery orders can be paid using cash or via SMS using Amazon SMS Pay Link.

Result: Success

2) User (Buyer) Registration - test cases

1) Test case: Verify that the page has both submit and cancel / next buttons at the end.

Verification: User can cancel items of orders by visiting the your orders sections in your account. Go to your orders. click cancel items. Select the check box next to each item you wish to remove from orders.

Result: Success.

2) Test case: Verify that clicking submit button after entering all the required fields, submit the data to the server.

Verification: User can submit a data request in addition to accessing a lot of your personal data & information in our account. Amazon collect data from user as they navigate the site. Such as the time spent browsing.

Result: Success

3) Test case: Verify that the clicking cancel / reset button after entering all the required fields, cancels the submit request.

Verification: When you cancel an order, Amazon automatically update the order status in the buyer's Amazon account and sends an email notification to the buyer.

Result: Success

Verification: In Amazon, it is possible to update price and quantity information using the inventory leader or product-specific inventory files. Use large inventory files only in the following situation.

you have additional items to list · you need to change the details of an existing item ·

Result: Success

14, test case: verify that product created by sellers get visible on the website after a certain period of time.

Verification: user should make their visuals appear to boost retail sales. Ensure their display reflects their brand. think about their target customer's experience. How m signage clarity when making their products stand out. Here, customer service is key

Result: success

N. Nansen



Design the testcases for a mobile application (consider any example from App store).

⇒ App : WhatsApp.

\* chat setting test scenario :

- 1) verify that the user can set a chat wallpaper for particular person or everyone.
- 2) verify that the user set privacy setting like turning on/off last seen, read receipts, status privacy etc.
- 3) verify that user can update notification setting like - notification sound on/off
- 4) verify that the user can block unwanted person or not
- 5) verify that the user can delete his whatsapp chat or not.

\* call setting scenario :

- 1) verify that the user can receive whatsapp calls from person
- 2) verify that the user can do make video call, audio call, conference call.

\* WhatsApp :

WhatsApp is most frequently used application in daily life. Although there would be thousands of test cases for WhatsApp but we are limiting the test of some of the most common and high level activities.

1. verify that an downloading the whatsapp application users can register using a new mobile number.
2. verify that for a new mobile number user will get a verification code on his mobile and filled the same verifies the new user account.
3. check the maximum number of incorrect attempts allowed while filled the verification code.

DP settings:

1. verify that the user can set DP and status on whatsapp.
2. verify that the user can update the existing DP and what's app status.
3. verify that the user can send multiple status to the person at a time
4. verify that the user can send the status to the particular person.

-group features:

N. Praveen

1. check the admin can change information / group name.
2. check the admin can change group image.
3. check if the person is removed from the group then that user will can't see the updates.

These are other app similar whatsapp with additional features like AB whatsapp.

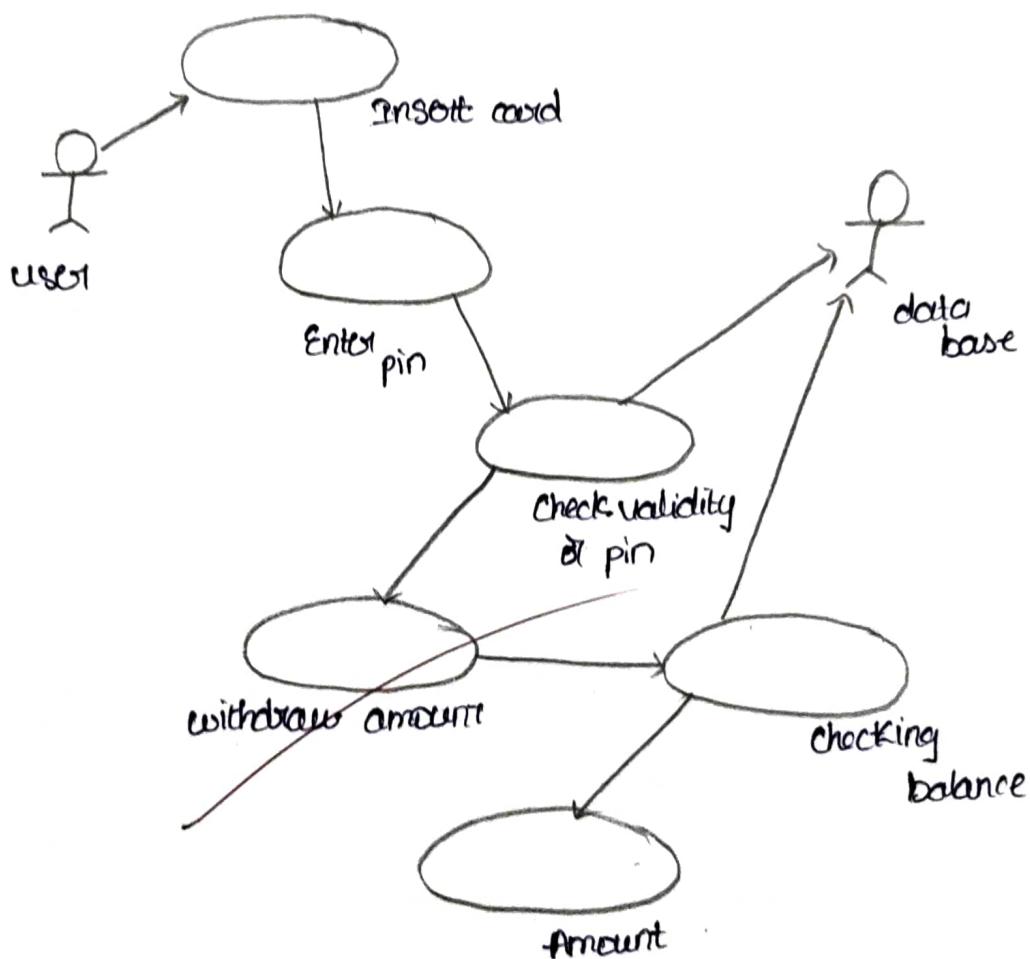
Design and implement ATM system through UML diagrams.

→ UML diagrams are classified into three categories that are given below:

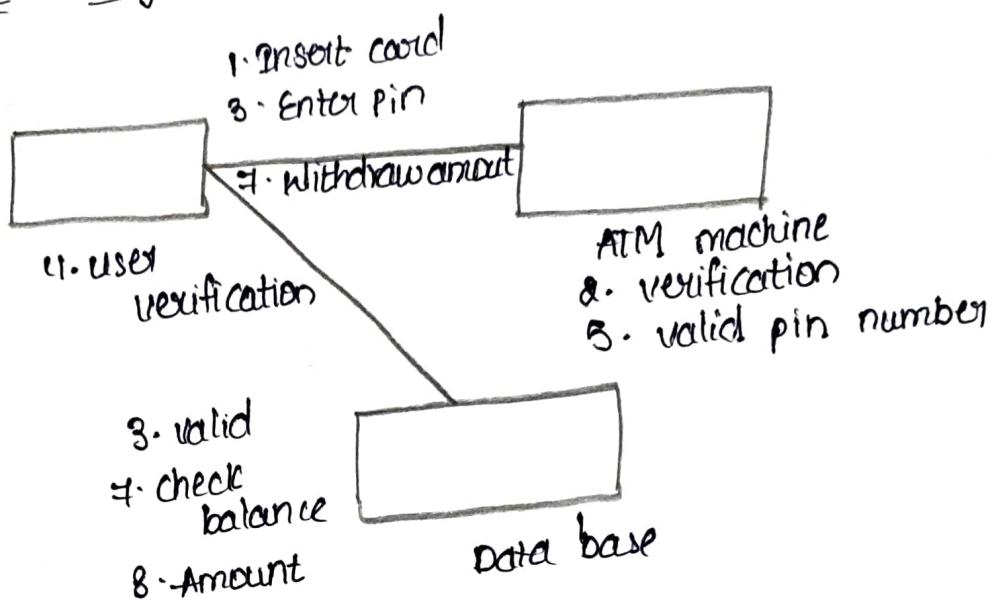
- 1) Structural diagram
- 2) Behaviour diagram
- 3) Interaction diagram

UML diagram:

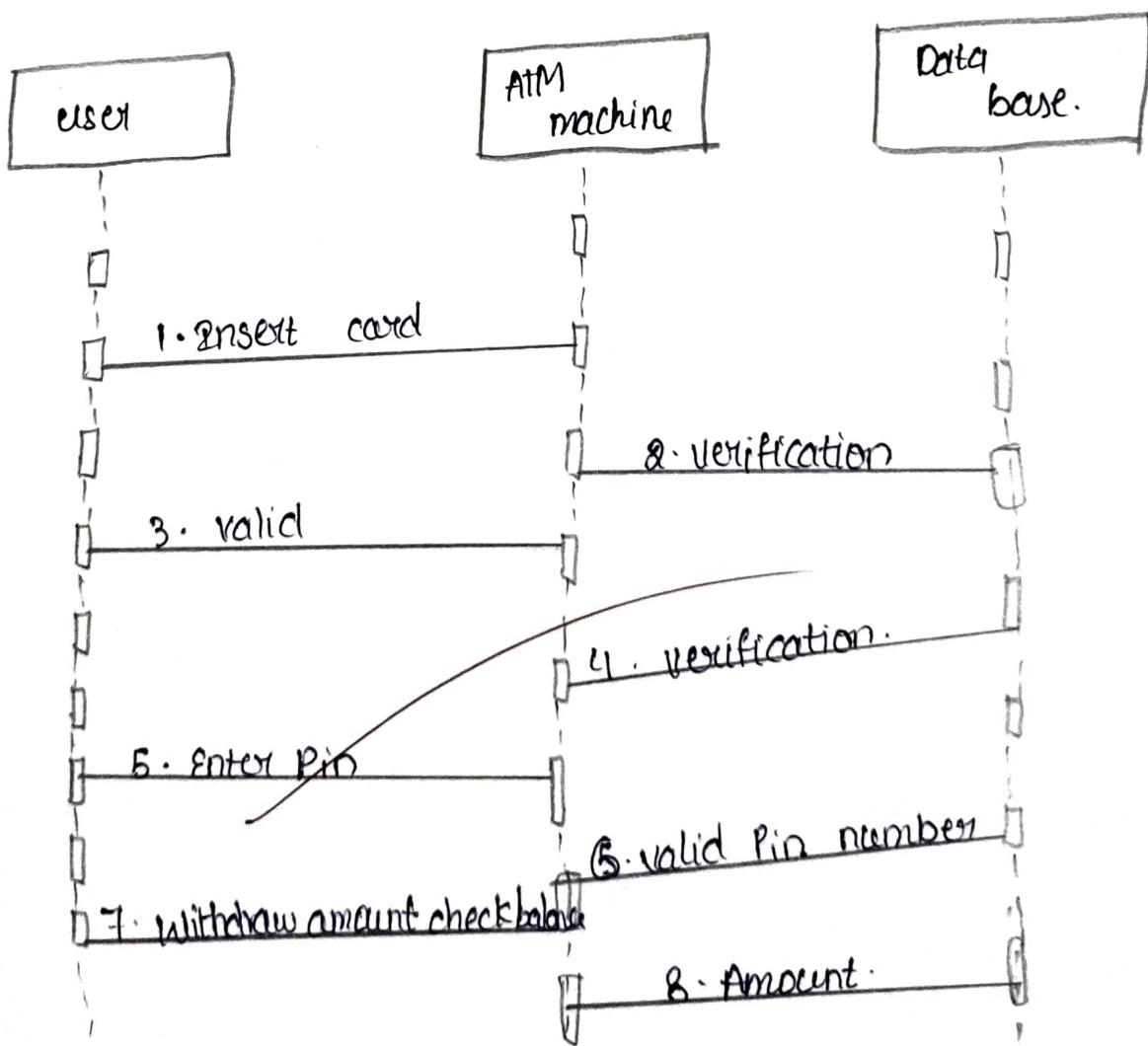
→ usecase diagram:



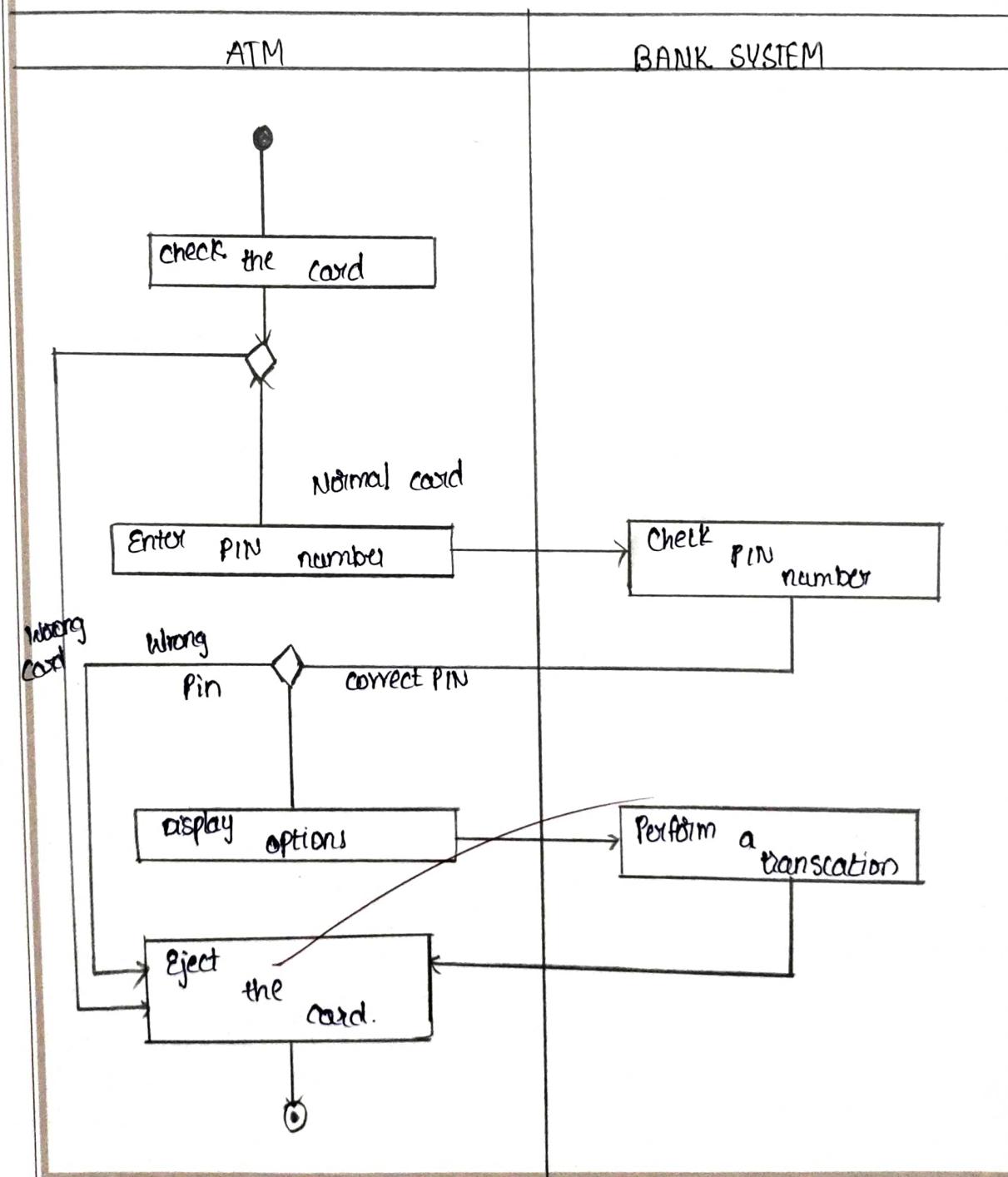
→ collaboration diagram :-



→ Sequence diagram :-



→ Activity diagram:



→ class diagrams:

