

DEPARTMENT OF INFORMATION TECHNOLOGY

**SOFTWARE ENGINEERING
LAB MANUAL**

**III B.Tech-I SEMESTER
(ACADEMIC YEAR 2020-21)**

Prepared by

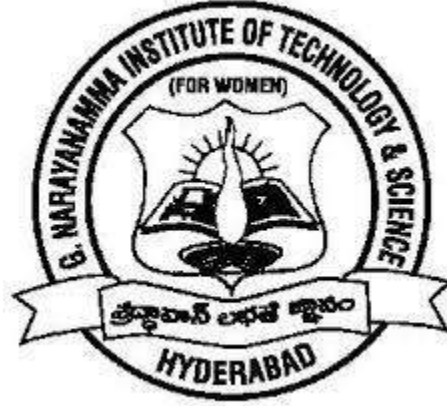
Dr.S.Rama Charan, Assoc.Prof.

Ms.D.Vandana, Asst.Prof


Approved by

Dr. I.Ravi Prakash Reddy

HOD IT



SOFTWARE ENGINEERING LAB

Department of IT, GNITS

Index

S. No	Contents	Pg. No
	Introduction	1
	Index	2-3
1	Academic Domain 1.1 Student Marks Analysing System 1.1.1 Problem Analysis 1.1.2 Requirement (fun & non fun) 1.1.3 E-R Diagram 1.2 Course Registration System. 1.2.1 Problem Analysis 1.2.2 Requirement (fun & non fun) 1.2.3 E-R Diagram	4-24 4 5-13 14 15 16-23 24
2	Airways Domain 2.1 Online ticket Reservation System. 2.1.1 Problem Analysis 2.1.2 Requirement (fun & non fun) 2.1.3 E-R Diagram 2.2 Flight information System. 2.2.1 Problem Analysis 2.2.2 Requirement (fun & non fun) 2.2.3 E-R Diagram	25-34 25 26-29 30 31 32-34 35
3	Medicine Domain 3.1 Expert System to prescribe the medicines for the given Symptoms. 3.1.1 Problem Analysis 3.1.2 Requirement (fun & non fun) 3.1.3 E-R Diagram 3.2 Remote Computer Monitoring 3.2.1 Problem Analysis 3.2.2 Requirement (fun & non fun) 3.2.3 E-R Diagram	36-50 36 37-41 42 43 44-49 50
4	Finance Domain 4.1 ATM System. 4.1.1 Problem Analysis 4.1.2 Requirement (fun & non fun) 4.1.3 E-R Diagram 4.2 Stock Maintenance. 4.2.1 Problem Analysis 4.2.2 Requirement (fun & non fun) 4.2.3 E-R Diagram	51-64 51 51-56 57 58 59-63 65
5	Human Resource Management	66-72

	5.1 Quiz System. 1.1.1 Problem Analysis 1.1.2 Requirement (fun & non fun) 1.1.3 E-R Diagram 5.2 E-Mail Client System. 1.2.1 Problem Analysis 1.2.2 Requirement (fun & non fun) 1.2.3 E-R Diagram	66 67 68 69 70-72 73
6	UML DIAGRAMS	74-122
7	FORWARD ENGINEERING	123-126
8	APPLICATION IMAGES	127-132
9	APPLICATION CODE	133-154

Department of IT, GNITS

ACADEMIC DOMAIN

1. COURSE REGISTRATION SYSTEM

Problem Analysis

Definition

The Course Registration System allows the students to register their academic subject based on their own study plan by determining the appropriate courses, timeslots and rooms. The students are required to make the combination of the study plan to make sure their timetable is in accordance with their preferred time.

Scope

Through the Course Registration System, the students are allowed to register a subject and are also permitted to drop the registered subject at any point after registering. However, only the courses uploaded by the faculty are available in the registration database.

Objectives

Course Registration System is created to make the process of course registration hassle free to a student. It is designed:

- To develop a prototype that allows students to register courses based on their preferred timeslots.
- To develop a system that is able to suggest courses on the available timeslots.
- To offer student satisfaction by facilitating heuristic method concept
- To view course information with more specific details and convenience

Software Requirement Specification Document

Introduction

This document specifies the requirements for a Course Registration System. The requirements are grouped by their stakeholders, and functional and non-functional requirements are separated.

Stakeholders

S	Students
L	Lecturers
B	Administration M
C	System maintainer

General requirements

Some requirements are shared by all stakeholders.

1	The system shall provide, store, represent <i>static course information</i>
2	The system shall provide, store, represent <i>dynamic course information</i>
3	The system shall provide a messaging system

Requirements of Students

Functional requirements

1	The system shall enable students to retrieve contact information of students and lecturers of subscribed courses
---	--

2	The system shall provide the history of a course (view contents of a course over the years)
3	The system shall provide the history of attended courses
4	The system shall enable students to subscribe to courses
5	The system shall enable students to subscribe/unsubscribe to online exams
6	The system shall be able to provide a collaboration environment in a course (so students can share files and notes within a team)
7	System shall allow students to upload co-curricular activities
8	The system shall be able to let students upload and download files
9	The system shall facilitate searches in all <i>static information</i> of courses.
10	The system shall facilitate searches within all <i>dynamic information</i> and files in a course
11	The system shall allow students to edit their personal information
12	The system shall provide a password reset function, which resets the password and mails it to the user
13	The system shall notify students of events, assignments, exams online as well as offline (posted news messages, team invites and scheduled exams)

Non-functional requirements

- Privacy

1	The system shall protect the user's privacy
2	The system shall prevent students from viewing grades of others
3	The system shall provide a user-customizable visibility policy for the <i>personal information</i>

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime
3	The system shall have downtime at most 4 hours/month
4	The system shall have its expected downtime announced at least 48 hours in advance

- User friendliness

1	The system will be user friendly
2	The system shall have bilingual support (Hindi and English)
3	The system shall have a maximum of 3 clicks to reach any content
4	The system shall have a single login to access all content
5	The system shall have a consistent and descriptive UI (in all the views and dialogs, the UI elements behave and are placed in a similar way)
6	The system shall have a UI which is intuitive (the behaviour of the system is according to the intuition of a standard end user)

- Accessibility

1	The system shall have high accessibility
2	The system shall be accessible by disabled (blind) users, who should be able to navigate the system and have access to all content and functionality

- Security

1	The system will be secure
2	The system shall allow only students to change <i>study information</i> of others

- Interoperability

1	The system shall be highly interoperable
2	The system shall provide an export to commonly used calendar formats (allowing users to import scheduled lectures into a personal calendar)

Requirements of Lecturers

Functional requirements

1	The system shall allow lecturers to create courses
2	The system shall allow lecturers to recreate a course (copied from a previous period)
3	The system shall allow lecturers to register <i>assistant lecturers</i>
4	The system shall allow lecturers to prepare lecture schedules (roster)
5	The system shall allow lecturers to upload course material for lectures
6	The system shall enable lecturers to <i>manage</i> grades or to change the grading policy (insert, update, calculate final grade)
7	The system shall enable lecturers to mail multiple students at once
8	The system shall provide a “one-click” function which will mail all students of the course their grades (no template should be entered or altered)
9	The system shall allow lecturers to manage <i>static, dynamic course information</i>
10	The system shall allow lecturers to specify enrolment policies based on grade, first-come first-serve and department
11	The system shall prevent students from subscribing to a course they don't qualify for (not completing required courses, or not from the right department(s))
12	The system shall enable lecturers to plan meetings with students or student teams
13	The system shall allow lecturers to post news messages
14	The system shall allow lecturers to set the visibility of archived items (enabling them to gradually expose content to students)
15	The system shall allow only lecturers to <i>manage</i> student teams

16	The system shall allow lecturers to enter grades for teams (so each team member will get that grade)
17	The system shall allow only lecturers to create, insert, remove delete, rename new and existing teams
18	The system shall provide grade statistics (averages, standard deviation, per department, per year)
19	The system shall enable lecturers to compare grade statistics with other courses
20	The system shall allow lecturers to duplicate courses and import materials from other courses into another course, but only from their own courses

Non-functional requirements

- Security

1	The system shall allow lecturers to view the <i>dynamic course information</i> of courses given by other lecturers
2	The system shall allow lecturers to <i>manage</i> the dynamic content visibility (visible for students and lecturers, visible for lecturers, visible to self only)
3	The system shall allow students to view only their own grade
4	The system shall allow lecturers to view all grades of all students in the course

- Interoperability

1	The system shall be able to import BOZ roster information into the course roster
---	--

- Privacy

1	The system shall protect the user's privacy
2	The system shall prevent students from viewing grades of others
3	The system shall provide a user-customizable visibility policy for the <i>personal information</i>

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime
3	The system shall have downtime at most 4 hours/month
4	The system shall have its expected downtime announced at least 48 hours in advance

- User friendliness

1	The system will be user friendly
2	The system shall have bilingual support (Hindi and English)
3	The system shall have a maximum of 3 clicks to reach any content
4	The system shall have a single login to access all content
5	The system shall have a consistent and descriptive UI (in all the views and dialogs, the UI elements behave and are placed in a similar way)
6	The system shall have a UI which is intuitive (the behaviour of the system is according to the intuition of a standard end user)

- Accessibility

1	The system shall have high accessibility
2	The system shall be accessible by disabled (blind) users, who should be able to navigate the system and have access to all content and functionality

Requirements of Maintainer

Functional requirements

1	The system shall allow maintainers to create back-ups of the entire system
2	The system shall allow maintainers to restore partial and complete back-ups of a specific date
3	The system shall provide latest updates to the students and lecturers from different sources.
4	The system shall allow maintainers to limit the total available space for specific courses

Non-functional requirements

- Extensibility / Evolvability

1	The system shall be easily extensible and evolvable
---	---

- Testability

1	The system shall be easily testable
---	-------------------------------------

- Scalability

1	The system shall be scalable and append able
---	--

- Maintainability

1	The system shall be easily maintainable
---	---

- Interoperability

1	The system shall be interoperable with secondary university systems
---	---

Requirements of the Administration

Functional requirements

1	The system shall allow only the administration to <i>manage, create new</i> courses or delete courses
2	The system shall allow only the administration to update <i>static course information</i>
3	The system shall allow only the administration to appoint (principal) lecturers to courses
4	The system shall allow only the administration to specify the minimum Number of students for a course. If there are too little subscriptions in a Semester, that course will not be given during that semester.
5	The system shall have no maximum limit for the number of course participants ever
6	The system shall allow only the administration to specify the course, department as prerequisites for students
7	The system shall allow only the administration to specify the department as prerequisite of a course
8	The system shall allow the administration to retrieve all <i>study and personal information</i> of students, lectures

Non-functional requirements

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime
3	The system shall have downtime at most 4 hours/month
4	The system shall have its expected downtime announced at least 48 hours in advance

- User friendliness

1	The system will be user friendly
2	The system shall have bilingual support (Hindi and English)
3	The system shall have a maximum of 3 clicks to reach any content
4	The system shall have a single login to access all content
5	The system shall have a consistent and descriptive UI (in all the views and dialogs, the UI elements behave and are placed in a similar way)
6	The system shall have a UI which is intuitive (the behaviour of the system is according to the intuition of a standard end user)

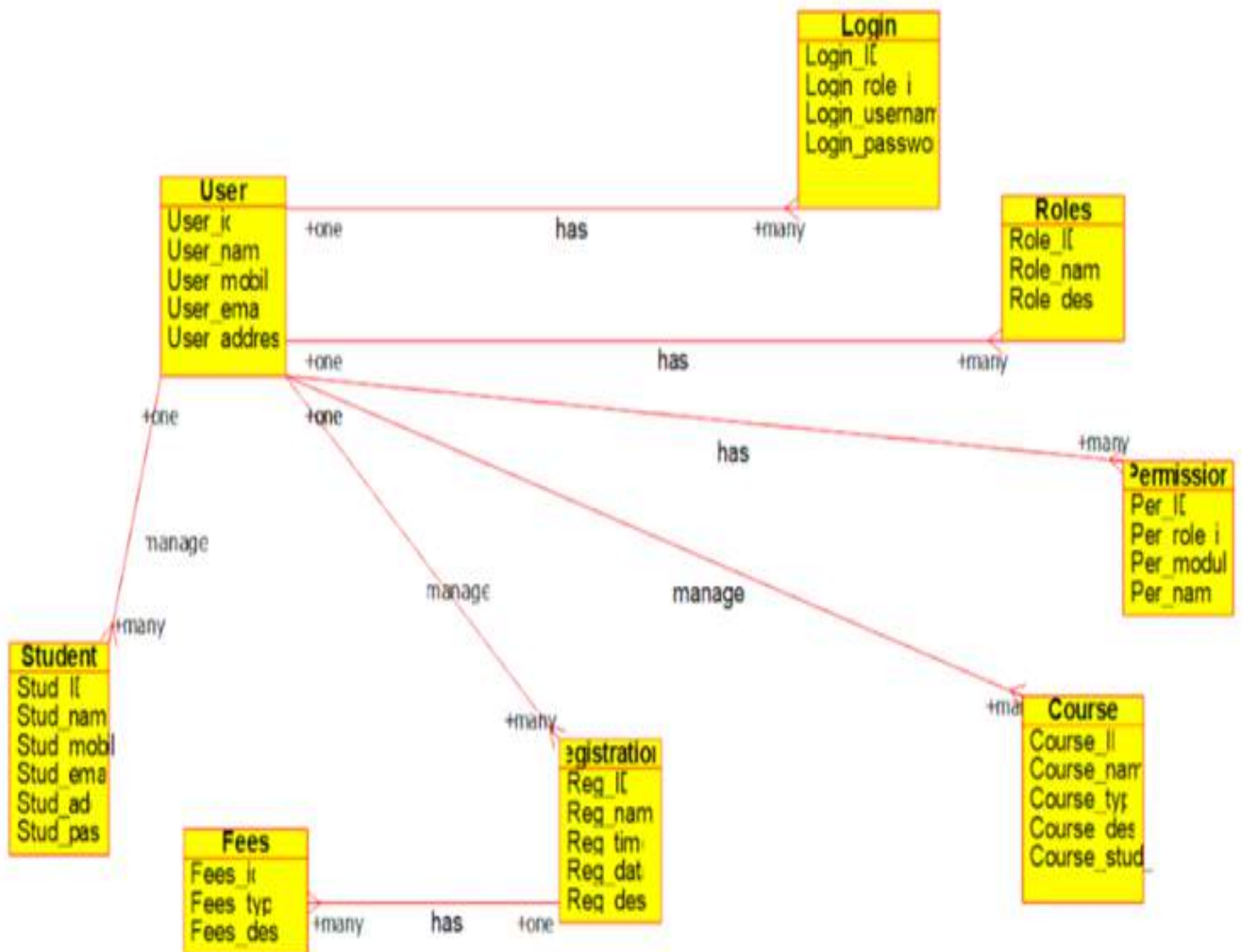
- Interoperability

1	The system shall be interoperable with <i>secondary university systems</i>
---	--

- Extensibility

1	The system shall allow the administration to make exceptions with regard to student enrolment to courses
---	--

Data Modelling



2. STUDENT MARKS ANALYSING SYSTEM

Problem Analysis

Definition

The Student Mark Analysis System deals with the complete academic details of the students. It comprises of the Roll No, Name, Mark, Total and average. It can be accessed by the faculty who alone can change or update the marks of the student. It is also the duty of the faculty to maintain the records, the duty of the administrator is to generate the report cards to the faculty members. The faculty will calculate the total marks based on the percentage obtained by the student. The role of the student is to just to view the marks entered by the faculty.

Scope

The main scope of the project is to get the student mark and update it whenever necessary and then to display it for the students and faculties. This system is very essential for every educational institution as it reduces man power.

Objectives

- This system reduces manual work to great extent.
- The mark analysis is carried out by the system in an efficient manner.
- The marks can be easily updated.
- The examination can be conducted in a secure environment.
- Once the papers have been submitted the analyser can enter the marks into the database.
- The system assigns the grade automatically based on the marks secured by the student.
- The students can easily check their scores. Access to database can be setup on username / password basis with multiple users and privileges

Software Requirement Specification Document

Introduction

This document specifies the requirements for Students Marks Analysing System. The requirements are grouped by their stakeholders, and functional and non-functional requirements are separated.

Stakeholders

S	Students
L	Lecturers
P	Principal (HOD)
A	Administrator
S	System Maintainer

General Requirements

Some requirements are shared by all stakeholders.

1	The system shall provide, store, represent <i>static course information</i>
2	The system shall provide, store, represent <i>dynamic course information</i>
3	The system shall provide a messaging system

Requirements of Students

Functional Requirements

1	The system should provide students a username and a temporary password through which students can access the Students Marks Analyzing System.
2	The system should allow students to reset their password and change their profile information.
3	The system should display the information of the marks obtained to the students.
4	The student should be able to see semester wise and subject wise marks.
5	The student should also be able to see the aggregate of all the semesters.
6	The students should also be able to compare their marks with other students.

Non-functional requirements

- Privacy

1	The system shall protect the user's privacy
---	---

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime

- User friendliness

1	The system will be user friendly
2	The system shall have a single login to access all content
3	The system shall have a consistent and descriptive UI

- Accessibility

1	The system shall have high accessibility
---	--

- Security

1	The system will be secure
---	---------------------------

- Interoperability

1	The system shall be highly interoperable
---	--

Requirements of Lecturers

Functional requirements

1	The system should provide lecturers a username and a temporary password through which students can access the Students Marks Analyzing System.
2	The system should allow lecturers to reset their password and change their profile information.
3	The system should facilitate lecturers to view class result.
4	The system should facilitate lecturers to view subject wise result.
5	The system should facilitate lecturers to calculate the percentage (internal or external or both).
6	The lecturers should also be able to make comparative analysis of students marks.

Non-functional requirements

- Privacy

1	The system shall protect the user's privacy
---	---

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime

- User friendliness

1	The system will be user friendly
2	The system shall have a single login to access all content
3	The system shall have a consistent and descriptive UI

- Accessibility

1	The system shall have high accessibility
---	--

- Security

1	The system will be secure
---	---------------------------

- Interoperability

1	The system shall be highly interoperable
---	--

Requirements of principal (HOD)

Functional requirements

1	The system should provide Principal a username and a temporary password through which Principal can access the Students Marks Analyzing System.
2	The system should allow Principal to reset their password and change their profile information.

3	The system should facilitate principal to view subject-wise result.
4	The system should facilitate principal to view branch-wise result.
5	The system should facilitate principal to view year-wise result.
6	The system should facilitate principal to calculate the pass percentage of students.
7	The principal should also be able to make comparative analysis of students' marks.

Non-functional requirements

- Privacy

1	The system shall protect the user's privacy
---	---

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime

- User friendliness

1	The system will be user friendly
2	The system shall have a single login to access all content
3	The system shall have a consistent and descriptive UI

- Accessibility

1	The system shall have high accessibility
---	--

- Security

1	The system will be secure
---	---------------------------

- Interoperability

1	The system shall be highly interoperable
---	--

Requirements of Administrator

Functional requirements

1	The system should provide administrator a username and a temporary password through which students can access the Students Marks Analyzing System.
2	The system should allow administrator to reset their password and change their profile information.
3	The system should allow administrator to upload subject-wise marks of all the students according to the year and semester.
4	The system should allow administrator to upload the students' details.
5	The system should allow administrator to update the student database.
6	The system should allow administrator to upload supplementary results of students.

Non-functional requirements

- Privacy

1	The system shall protect the user's privacy
---	---

- Availability

1	The system shall have high availability
2	The system shall not have unexpected downtime

- User friendliness

1	The system will be user friendly
2	The system shall have a single login to access all content
3	The system shall have a consistent and descriptive UI

- Accessibility

1	The system shall have high accessibility
---	--

- Security

1	The system will be secure
---	---------------------------

- Extensibility

1	The system shall allow the administrator to make exceptions with regard to students results
---	---

- Interoperability

1	The system shall be highly interoperable
---	--

Requirements of system maintainer

Functional requirements

1	The system maintainer should protect and store the data of students and their marks.
2	The system maintainer should be able to set authorization permissions to students, lecturers, principal and administrator.
3	The system administrator should update the Students Marks Analyzing System regularly

Non-functional requirements

- Extensibility

1	The system shall be easily extensible and evolvable
---	---

- Testability

1	The system shall be easily testable
---	-------------------------------------

- Scalability

1	The system shall be scalable and append able
---	--

- Maintainability

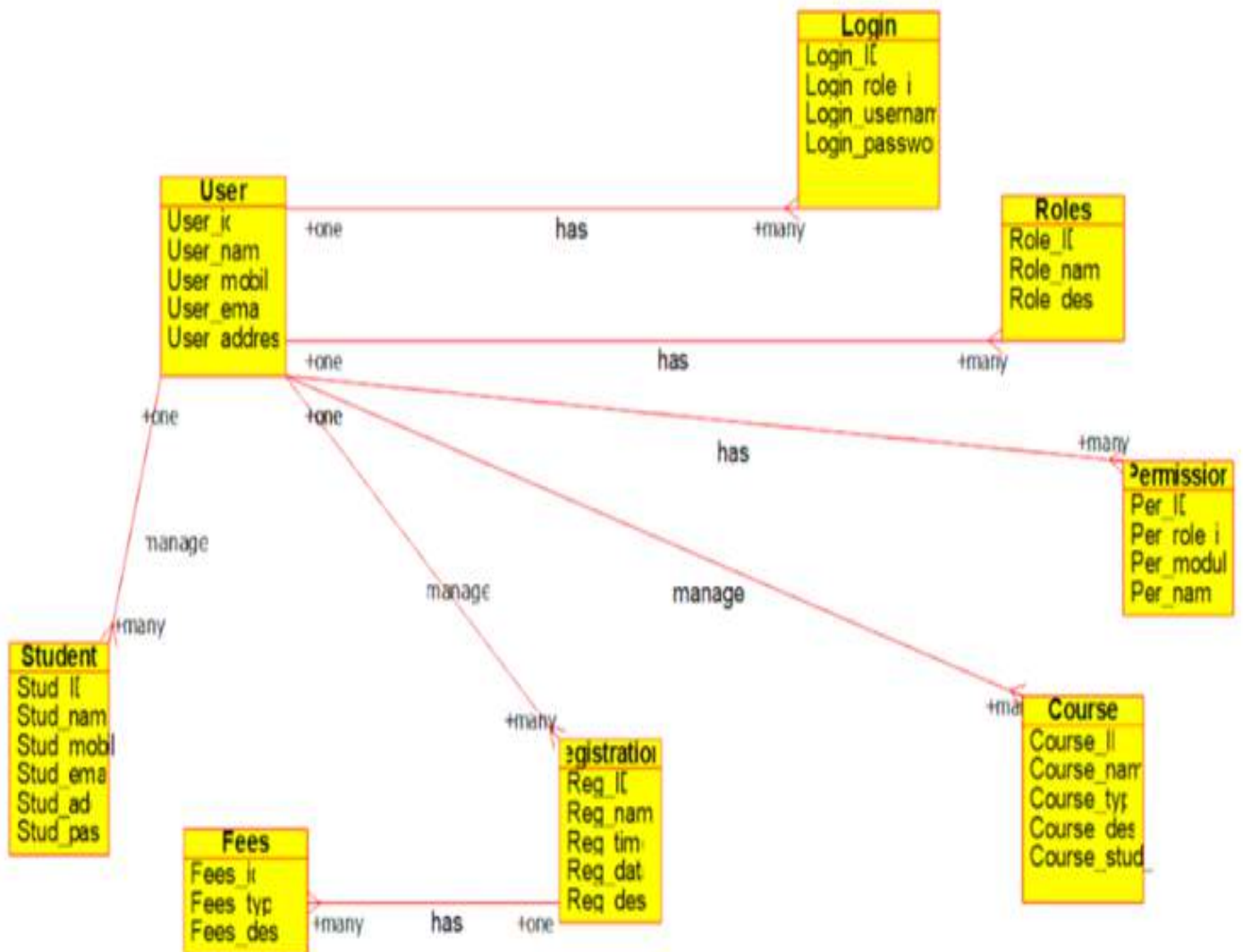
1	The system shall be easily maintainable
---	---

- Interoperability

1	The system shall be interoperable with secondary university systems
---	---

Department of IT, GNITS

Data Modelling



AIRWAYS DOMAIN

ONLINE TICKET RESERVATION SYSTEM

Problem Definition:

Ticket Reservation System is a web-based application for the easy booking of online tickets by showing the availability of the seats based on the criteria given. The customer can search different routes from the source to the destination and then select which one they want, check flight timings and book a ticket through online. The seats (tickets) will be of First Come First Serve mode, only some seats will be reserved for booking which will be available 24 hours before the departure time.

Scope:

It is a secure website with authentication. Cost can vary time to time. You can cancel the registration within the given deadline or lose the money after it. It saves passenger details, ticket details, flight schedules etc. in the database. Server issues may occur some times.

Objectives:

- It is user friendly.
- Provides search facility to search for flights according to time and date.
- The customer can choose the way which we wish to travel and afford and book the ticket respectively.
- You will get seasonal discounts and additional discounts using some credit cards.
- Payment option is available through debit/credit card, net banking and also UPI transactions
- Passenger can modify their personal information when needed.
- Database stores records of flights, passengers, flight schedules, etc.
- Passengers can view their previous booking history.
- It also shows that either female or male booked the respective seat.
- It shows whether tickets are available or not.
- Tickets for flights can be booked 3 hours before the flight flies.

- Cancellation of the ticket should take place before 1 week or otherwise refund will not be provided.

SRS DOCUMENT:

Introduction

This document specifies the requirements for a Online Ticket Reservation System. The requirements are grouped by their stakeholders, and functional and non-functional requirements are separated.

Stakeholders

C	Customer
A	Administration
M	System Maintainer

Requirements of Customer

Functional Requirements

R1	System shall allow customer to signup/login
R2	System shall allow customer to edit his/her information
R3	The system shall allow customer to view his travel history
R4	The system shall enable customer to filter flights available by date ,time ,destination and boarding time
R5	The System shall enable customer to filter available flights based on cost
R6	The System enables customer to filter and chose flights based on class of travel
R7	The System shall enable customer to reserve and pay for the ticket
R8	The System shall provide e-ticket to the customer

Non Functional Requirements

Privacy

R9	The system shall protect user's privacy
R10	The system shall provide user-customizable visibility policy for personal information

Availability

R11	The system shall have high availability
R12	The system shall not have unexpected downtime
R13	The system shall have downtime only during low-intensity hours

User friendliness

R14	The system will be user friendly
R15	The system should provide language selection support
R16	The system shall have single login to access all content

Security

R17	The system will be secure
-----	---------------------------

Interoperability

R18	The system shall be highly interoperable
-----	--

Requirements of Administration

Functional Requirements

R19	The System shall allow only administration to manage flight information
R20	The system shall allow administration to update delete or add flight information
R21	The system shall allow administration to retrieve flight information
R22	The system shall allow administration to retrieve required passenger information

Non Functional Requirements

Privacy

R9	The system shall protect user's privacy
R10	The system shall provide user-customizable visibility policy for personal information

Availability

R11	The system shall have high availability
R12	The system shall not have unexpected downtime
R13	The system shall have downtime only during low-intensity hours

User friendliness

R14	The system will be user friendly
R15	The system should provide language selection support
R16	The system shall have single login to access all content

Security

R17	The system will be secure
-----	---------------------------

Interoperability

R18	The system shall be highly interoperable with other devices /systems
-----	--

Requirements of Maintainer

Functional requirements

R19	The system shall allow maintainers to create back-ups of the entire system
R20	The system shall allow maintainers to restore partial and complete back-ups of a specific date

Testability

R21	The system shall be easily testable
-----	-------------------------------------

Scalability

R22	The system shall be scalable and append able
-----	--

Maintainability

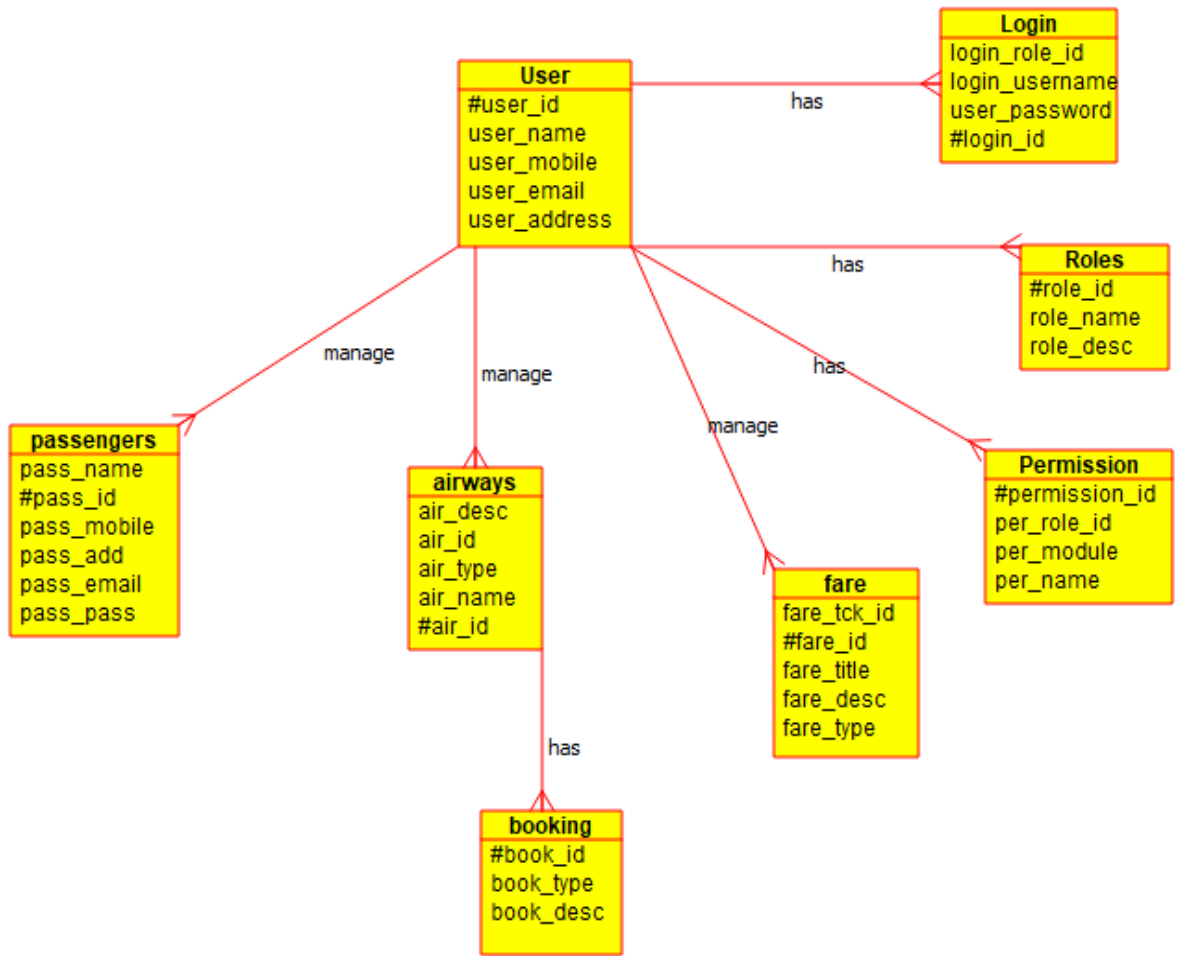
R23	The system shall be easily maintainable
-----	---

Interoperability

R24	The system shall be highly interoperable with other devices /systems
-----	--

Department of IT, GNITS

ER DIAGRAM



FLIGHT INFORMATION SYSTEM

Problem definition:

A flight information display system (FIDS) is a computer system used in airports to display flight information to passengers, in which a computer system controls mechanical or electronic display boards or TV screens in order to display arriving and departing flight information in real-time.

Scope:

When surroundings are unfamiliar and time is critical, people rely on display systems to guide them to their desired destination. Sophisticated flight information systems have to take existing language barriers into consideration and must provide accurate and complete information at all relevant locations.

Objectives:

- FIDS are used to inform passengers of boarding gates, departure/arrival times, destinations, notifications of flight delays/flight cancellations, and partner airlines, etc.
- Emergency override message displays .

SRS DOCUMENT:

Introduction

This document specifies the requirements for a Flight Information System. The requirements are grouped by their stakeholders, and functional and non-functional requirements are separated.

Stakeholders

P	Passenger
A	Administration
M	System Maintainer

Requirements of Passenger

Functional Requirements

R1	System shall allow passenger to signup/login
R2	System shall allow customer to edit his/her information
R3	The system shall allow customer to view his travel history
R4	The System shall enable customer view his/her e-ticket
R5	The system shall enable customer view expected boarding time/date
R6	The system shall enable customer view expected reaching time/date
R7	The system shall enable customer view gate no ,airline neme and other required information

Non Functional Requirements

Privacy

R9	The system shall protect user's privacy
R10	The system shall provide user-customizable visibility policy for personal information

Availability

R11	The system shall have high availability
R12	The system shall not have unexpected downtime
R13	The system shall have downtime only during low-intensity hours

User friendliness

R14	The system will be user friendly
R15	The system should provide language selection support
R16	The system shall have single login to access all content

Security

R17	The system will be secure
-----	---------------------------

Interoperability

R18	The system shall be highly interoperable
-----	--

Requirements of Administration

Functional Requirements

R19	The System shall allow only administration to manage flight information
R20	The system shall allow administration to update delete or add flight information
R21	The system shall allow administration to retrieve flight information
R22	The system shall allow administration to retrieve required passenger information

Non Functional Requirements

Privacy

R9	The system shall protect user's privacy
R10	The system shall provide user-customizable visibility policy for personal information

Availability

R11	The system shall have high availability
R12	The system shall not have unexpected downtime
R13	The system shall have downtime only during low-intensity hours

User friendliness

R14	The system will be user friendly
R15	The system should provide language selection support
R16	The system shall have single login to access all content

Security

R17	The system will be secure
-----	---------------------------

Interoperability

R18	The system shall be highly interoperable with other devices /systems
-----	--

Requirements of Maintainer

Functional requirements

R19	The system shall allow maintainers to create back-ups of the entire system
R20	The system shall allow maintainers to restore partial and complete back-ups of a specific date

Extensibility / Evolvability

R21	The system shall be easily extensible and evolvable
-----	---

Testability

R22	The system shall be easily testable
-----	-------------------------------------

Scalability

R23	The system shall be scalable and append able
-----	--

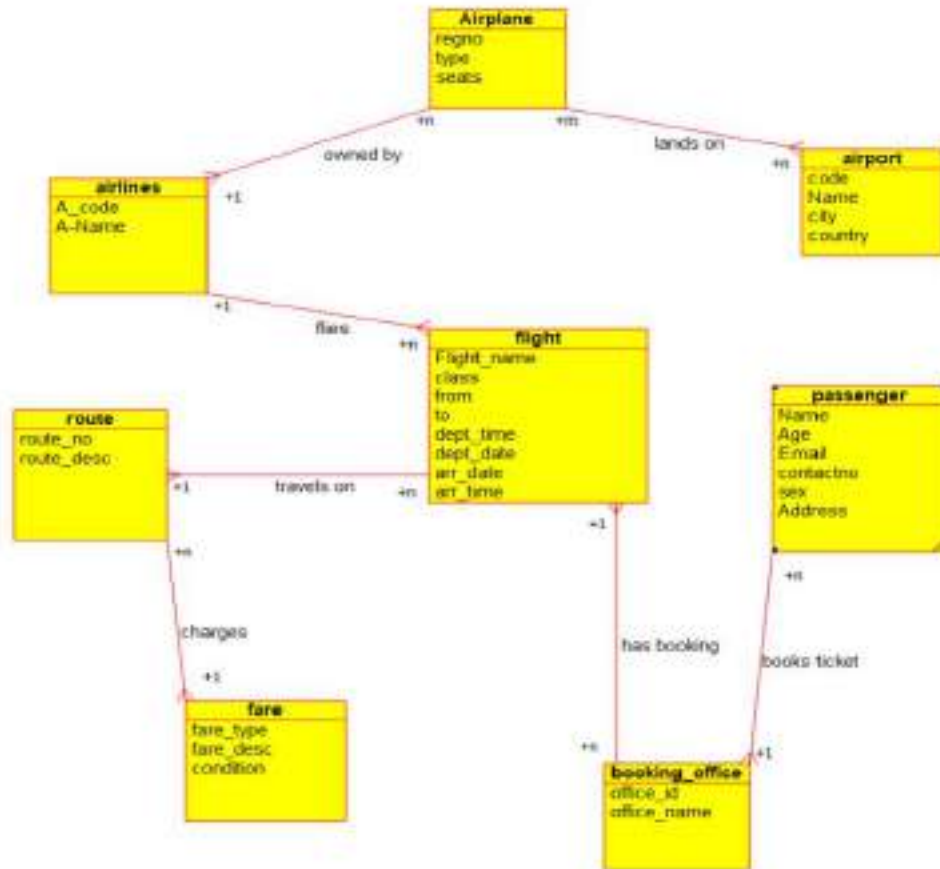
Maintainability

R24	The system shall be easily maintainable
-----	---

Interoperability

R25	The system shall be highly interoperable with other devices /systems
-----	--

ER DIAGRAM



Dex

MEDICINE DOMAIN

1. EXPERT SYSTEM TO PRESCRIBE THE MEDICINES FOR THE GIVEN SYMPTOMS

Problem Analysis

Definition

The Expert System is mainly focused on medical field where the person or patient is expected to select the symptoms given by the Expert System in the field information upon logging in with respective credentials. The Expert System will diagnose the disease using medical Database. The medical database will get all the information about medicine from pharmacy module and generates the prescription for the right symptoms and gives it to Expert System.

Scope

The Expert System is limited to commonly prescribed drugs for general clinical illnesses. The system gives the physician a list of alternative medicines available, their possible side effects and precautions based on the details entered by the patient.

Objectives

The Expert System is designed to prescribe medicines based on the symptoms mentioned by the patient. It is designed to:

- maintain medical records of a patient
- reduce the risk of data loss
- reduce paper work
- reduce medical errors
- provide accessibility to the users at the comfort of their home.

Software requirements specifications

This document specifies the requirements for Requirements for Expert system to prescribe the medicines for the given symptoms. The requirements are grouped by their stakeholders and functional and non-functional requirements are separated.

Stakeholders Include:

Stakeholders

P	Patient
P	pharmacist
M	maintainer

General requirements

Some requirements are shared by all stakeholder.

1	The system shall provide, store, represent static medical information
2	The system shall provide, store, represent dynamic medical information
3	The system shall provide a messaging system

Requirements of patient

Functional requirements

1.	Every person should enrol in the system before entering the system.
2.	Any person before entering this system should hold an account, and should provide basic details of their height, weight and any problems.
3.	The main motive of the patient is to provide honest details about themselves and what their exact problem.
4.	After providing the details a prescription is sent to the patient's phone.
5.	Daily tracking of their health is done.
6.	Daily update of their health should be given by them.
7.	The patient receives a detailed bill payment of their prescription.
8.	Patients can access the system anytime.
9.	Patients will be given remainder to put their medicines.

Non-functional requirements

Response time of patient

The system shall respond to the patient within 1 min.

Security of patient

1 The system requires the patient to identify himself/herself using their phone.

2 While logging in every time an otp is sent.

Availability of patient

The system shall be available all the time.

Maintainability of patient

The system will provide all the backup data of their pervious appointments.

User friendliness of patient

1 The system will be user friendly

2 The system shall have bilingual support (Hindi and English)

Testability of patient

1.The system shall be easily testable

Scalability of patient

1.The system shall be scalable and append able

Requirements of pharmacist

Functional requirements

1. Pharmacist will be given time slots when to do their job and in the given slot they have to prescribe medicines to the patients.
2. Pharmacist should have minimum qualification to enrol in this system.
3. Pharmacist should respond to the patient data within 15-20 mins.
4. When a medicine is not available the pharmacist should mention to the patient that the medicine given is an alternative.
5. Pharmacist can access the patient in any form of communication depending on patient's severity.
6. pharmacist should provide and tell the patients what medicine is used for what purpose.
7. A text message of medicine's should be sent to patients

Non-functional requirements

Security of pharmacist

1 The system requires the pharmacist to identify himself/herself using their phone.
2 While logging in every time an otp is sent.

Availability of pharmacist

The system shall be available all the time.

Maintainability of pharmacist

The system will provide all the backup data of their pervious appointments of the patient they are dealing with.
--

User friendliness of pharmacist

- | |
|---|
| 1 The system will be user friendly |
| 2 The system shall have bilingual support (Hindi and English) |

Testability of pharmacist

- | |
|---------------------------------------|
| 1.The system shall be easily testable |
|---------------------------------------|

Scalability of pharmacist

- | |
|--|
| 1.The system shall be scalable and append able |
|--|

Requirements of maintainer

Functional requirements

- | |
|--|
| 1. the maintainer should act as a bridge between the patient and the pharmacist. |
| 2. The maintainer should schedule the timings and slots for pharmacist |
| 3. The maintainer should have all the details of the patient. |
| 4. The maintainer should keep the detail track of patient's medicines and bill payments. |
| 5. Generation on bill should be done by the maintainer. |
| 6. Maintainer should respond to the patient, doctor, junior doctor in least possible time. |

Non-functional requirements

Security of maintainer

- | |
|---|
| 1 The system requires the pharmacist to identify himself/herself using their phone. |
| 2 While logging in every time an otp is sent. |

Availability of maintainer

- | |
|---|
| The system shall be available all the time. |
|---|

Maintainability of maintainer

The system will provide all the backup data of their pervious appointments of the patient they are dealing with.

User friendliness of maintainer

1 The system will be user friendly

2 The system shall have bilingual support (Hindi and English)

Testability of maintainer

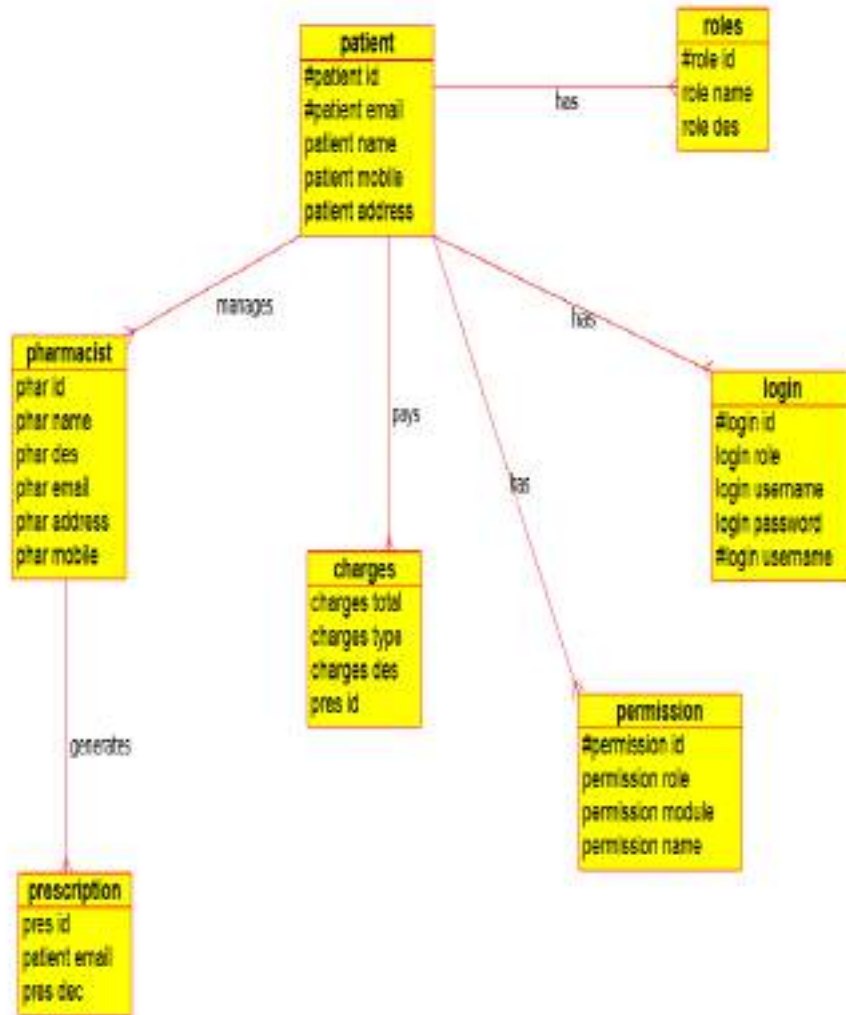
1.The system shall be easily testable

Scalability of maintainer

1.The system shall be scalable and append able

Department of IT, GNITS

ER Diagram



2. Remote Computer Monitoring

Problem Definition

Remote Computer monitoring system is designed to monitor the clients in the network with the help of databases. This system works on the basis of client-server interaction. For security reasons clients only areas pertained to them.

It involves installing software and managing all activities on the systems/network, workstations, servers or endpoints of a client, from a remote location

Scope

RCM achieves its goal of monitoring by getting some system related information of the Clients and also the Desktop Image of the Client Systems. It does not in any way disturb the Users working on the Remote Clients. Thus it works only as a Remote Monitoring Tool but not as a Remote Control Tool.

RCM is a tool specifically developed to monitor the Clients connected to the Server remotely. It does not provide remote control functionality such as access to the File System or any other feature which gives control over the remote system.

Objectives

- Remote computer monitoring offers continuous, real-time remote monitoring, covering all the devices across your network.
- Performance maintenance. You can view the resource usage from ongoing processes, kill unresponsive programs and create a check that monitors problem issues.
- Remote command line. Your remote command line is always on and is easily accessible from within a single, centralized dashboard.
- Service dependencies. View dependencies and monitor services that have grown problematic using a Windows service check.
- Service management. Execute, terminate, and restart system services.

Software Requirements specifications

This document specifies the requirements for a Remote computer monitoring. The requirements are grouped by their stakeholders and functional and non-functional requirements are separated.

Stakeholders:

D	Doctor
J	Junior doctor
P	Patient
M	maintainer

General requirements

Some requirements are shared by all stakeholder.

1	The system shall provide, store, represent static medical information
2	The system shall provide, store, represent dynamic medical information
3	The system shall provide a messaging system

Requirements of doctors

Functional requirements

1.	Doctors will be given slots to do their duty.
2.	The doctors have to login through login module.
3.	The system should allow the doctors to update their profile
4.	Doctors should give their desired time to check the patients to the maintainer
5.	Doctors can wish to take video class from the patients
6.	Doctors should discuss about the patients' health with juniors' doctors.
7.	Doctors can view their health history
8.	The system should allow doctors to prescribe certain medicine
9.	The system should allow Inter connection of doctors.

Non-functional requirements

Privacy of doctors

The system shall protect the user's privacy

The system shall prevent doctors from modifying other

doctor's cases.

The system shall provide a user-customizable visibility policy for the personal information.
--

Interoperability of doctors

The system shall be highly interoperable
--

Accessibility of doctors

The system shall have high accessibility
--

Security of doctors

1.The system will be secure.

2.The System provides the Login ID and password to maintain the security.

Testability of doctors

1.The system shall be easily testable

Scalability of doctors

1.The system shall be scalable and append able
--

Requirements of patients

Functional requirements of patients

1. The system shall allow to create an account.
2. The system shall allow the patient to see the information of doctor appointment.
3. The System shall allow the patient to browse for the required doctor.
4. The System shall allow the patient to update their Health Condition.
5. The patients should give their basic details such as weight, height, age, sugar levels and etc to the junior doctors
6. While telling the problem the patients should be honest
7. The System shall allow the patient to pay the medical bills, hospital fees etc.

Non-functional requirements

Privacy of patients

The system shall protect the user's privacy
The system shall prevent doctors from modifying other

Interoperability of patients

The system shall be highly interoperable
--

Accessibility of patients

The system shall have high accessibility
--

Security of patients

1.The system will be secure.
2.The System provides the Login ID and password to maintain the security.

Testability of patients

1.The system shall be easily testable

Scalability of patients

1.The system shall be scalable and append able
--

Requirements of junior doctor

Functional requirements of junior doctor

- | |
|--|
| 1. For enrolling into the system, the doctors should medically graduate. |
| 2. Juniors doctors will be given slots to perform their duty |
| 3. This system provides login to the junior doctors |
| 4. Juniors should act as a bridge between the doctor and the patient |
| 5. Juniors doctors should first analysis the situation of the patient |
| 6. Junior doctors should provide brief about the patient and their problem |

Non-functional requirements of junior doctor

Privacy of junior doctor

The system shall protect the user's privacy

The system shall prevent doctors from modifying other

Interoperability of junior doctor

The system shall be highly interoperable
--

Accessibility of junior doctor

The system shall have high accessibility
--

Security of junior doctor

1.The system will be secure.

2.The System provides the Login ID and password to maintain the security.

Testability of junior doctor

1.The system shall be easily testable

Scalability of junior doctor

1.The system shall be scalable and append able

Requirements of Maintainer

Functional requirements

1. the maintainer should have direct contact with the patients
2. The maintainer should schedule the timings and slots for doctors and junior doctors
3. Maintainer should have all the details of the patient
4. The maintainer should keep the detail track of patient's medicines and bill payments.
5. Generation of bill should be done by the maintainer
6. Maintainer should respond to the patient, doctor, junior doctor in least possible time

Non-functional requirements

Privacy of maintainer

The system shall protect the user's privacy

The system shall prevent doctors from modifying other

Interoperability of maintainer

The system shall be highly interoperable

Accessibility of maintainer

The system shall have high accessibility

Security of maintainer

1.The system will be secure.

2.The System provides the Login ID and password to maintain the security.

Testability of maintainer

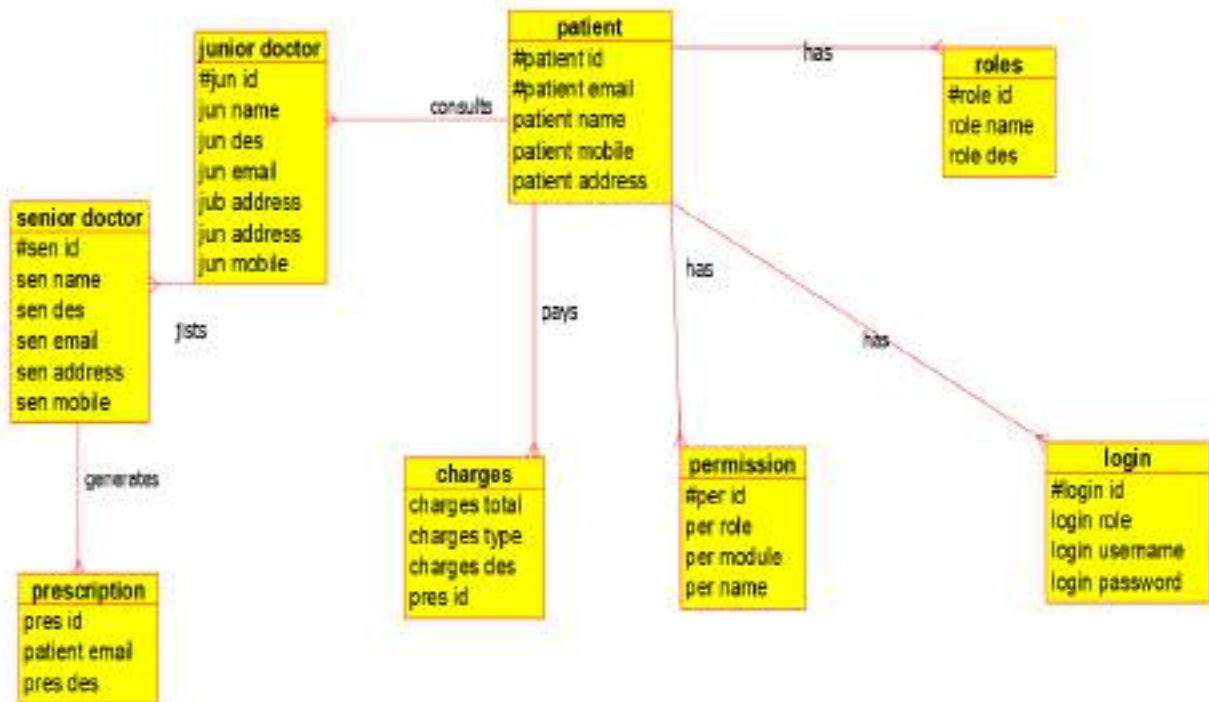
1.The system shall be easily testable

Scalability of maintainer

1.The system shall be scalable and append able

Department of IT, GNITS

ER Diagram



FINANCE DOMAIN

ATM system

Problem Definition:

A bank has several automated teller machines (ATMs) that are geographically distributed and connected via a wide area network to a central server. Each ATM machine has a card reader, a cash dispenser, a keyboard/display, and a receipt printer. By using the ATM machine, a customer can withdraw cash from either a checking or savings account, query the balance of an account, or transfer funds from one account to another.

Scope:

ATM system can be implemented in ATM machine by owner of bank or in charge of branch.

The network enables customers to complete simple bank account services via automated teller machines that may be located off premise and that need no to be owned and operated by customer's bank. It collects information about a simple account transaction, communicates the transaction information to the customer's bank, and dispenses cash to the customer.

Objectives:

The main objective is to speed up the transactions done by customers. No manual transactions needed generally. The second objective is to save the time which is very important now-a-days.

It also includes other objectives such as:

- To render accurate services to customer.
- The reduction of fraudulent activities.
- To achieve speedy processing of customer data.
- To reduce error processing, the guarantee of increase security.
- The Objective of Automated Teller Machines is convenient.

SOFTWARE REQUIREMENT SPECIFICATION

Description:

The automated teller machine (ATM) is an automatic banking machine (ABM) that allows the customer to complete basic transactions without any help from bank representatives. The basic one allows the customer to only draw cash and receive a report of the account balance.

Stake holders:

B	Bank Employee
C	Customer
A	ATM administrator
M	ATM machine

General requirements:

Some requirements are shared by all stakeholders.

R1	The system shall provide static course information
R2	The system shall be able to store static course information
R3	The system shall be able to represent static course information
R4	The system shall provide dynamic course information
R5	The system shall be able to store dynamic course information
R6	The system shall be able to represent dynamic course information
R7	The system shall provide a messaging system

Requirements of bank employee:

Functional requirements:

R1	The system shall allow Banks to verify card id.
R2	The system shall verify bank with card.
R3	The system shall allow banks to verify card password.
R4	The system Should allow banks verify amount request.
R5	The system Should provide bank to debit/credit amount to account.

Non-functional requirements:

Privacy:

R1	User accessibility is ensured in all ways.
R2	Users are advised to change their pin on the first use.
R3	Users are advised not to tell their pin to others.
R4	Maximum number of attempts to enter pin is 3 times.
R5	The system shall protect the user's privacy.

Availability:

R1	The system shall have high availability.
R2	The system shall not have unexpected downtime.
R3	The system shall have downtime only during low-intensity hours.

User friendly:

R1	The system will be user friendly.
R2	The system shall have trilingual support (Hindi, Regional language and English).

Accessibility:

R1	The system shall have high accessibility.
R2	The system Should be able to navigate the system and have access to all content and functionality.

Security:

R1	The system shall have a Internet link secure layer on top.
----	--

Interoperability:

R1	The system should be highly interoperable.
----	--

Requirements of customer:

Functional requirements:

R1	The system shall allow customer to Insert an atm card.
R2	The system shall allow Validating the atm card of customer.
R3	The system shall allow customer to enter the product task.

R4	The system shall allow the customer to Enter the pin.
R5	Customer should enter the valid pin
R6	The system shall allow customer in Validating for account type if the task is banking.
R7	The system shall ask customer for amount to be withdrawn.
R8	The system shall allow customer to debit the amount if there is sufficient balance availability.
R9	Error message is displayed otherwise.
R10	The system shall Ask for a printing advice if the task is balance enquiry.

Non-functional requirements:

Privacy:

R1	User accessibility is ensured in all ways.
R2	Users are advised to change their pin on the first use.
R3	Users are advised not to tell their pin to others.
R4	Maximum number of attempts to enter pin is 3 times.
R5	The system shall protect the user's privacy.

Availability:

R1	The system shall have high availability.
R2	The system shall not have unexpected downtime.
R3	The system shall have downtime only during low-intensity hours.

User friendly:

R1	The system will be user friendly.
R2	The system shall have trilingual support (Hindi, Regional language and English).

Accessibility:

R1	The system shall have high accessibility.
R2	The system Should be able to navigate the system and have access to all content and functionality.

Interoperability:

R1	The system should be highly interoperable.
----	--

Security:

R1	The system will be secured.
R2	Pin should be asked by the system for any kind of accessing.

Requirements for ATM administrator:

Functional requirements:

R1	The system Should allow administrator to monitor the technical status of ATM machine.
R2	The system shall allow administrator to upgrade the current application.
R3	The system provides administrator to ensure of platform software.
R4	The system provides graphical User Interface to administrator for authorization.

Non-functional requirements:

Privacy:

R1	User accessibility is ensured in all ways.
R2	Users are advised to change their pin on the first use.
R3	Users are advised not to tell their pin to others.
R4	Maximum number of attempts to enter pin is 3 times.
R5	The system shall protect the user's privacy.

Availability:

R1	The system shall have high availability.
R2	The system shall not have unexpected downtime.
R3	The system shall have downtime only during low-intensity hours.

User friendly:

R1	The system will be user friendly.
R2	The system shall have trilingual support (Hindi, Regional language and English).

Accessibility:

R1	The system shall have high accessibility.
R2	The system Should be able to navigate the system and have access to all content and functionality.

Interoperability:

R1	The system should be highly interoperable.
----	--

Requirements for ATM machine:

Functional requirements:

R1	The system shall know a type of transaction to be performed.
R2	The system Should contain information regarding States/Screen display.
R3	The system should be able to Reverse the last transaction.
R4	The system shall provide timeout parameters based on requirement.

Non-functional requirements:

Privacy:

R1	User accessibility is ensured in all ways.
R2	Users are advised to change their pin on the first use.
R3	Users are advised not to tell their pin to others.
R4	Maximum number of attempts to enter pin is 3 times.
R5	The system shall protect the user's privacy.

Availability:

R1	The system shall have high availability.
R2	The system shall not have unexpected downtime.
R3	The system shall have downtime only during low-intensity hours.

User friendly:

R1	The system will be user friendly.
R2	The system shall have trilingual support (Hindi, Regional language and English).

Accessibility:

R1	The system shall have high accessibility.
R2	The system Should be able to navigate the system and have access to all content and functionality.

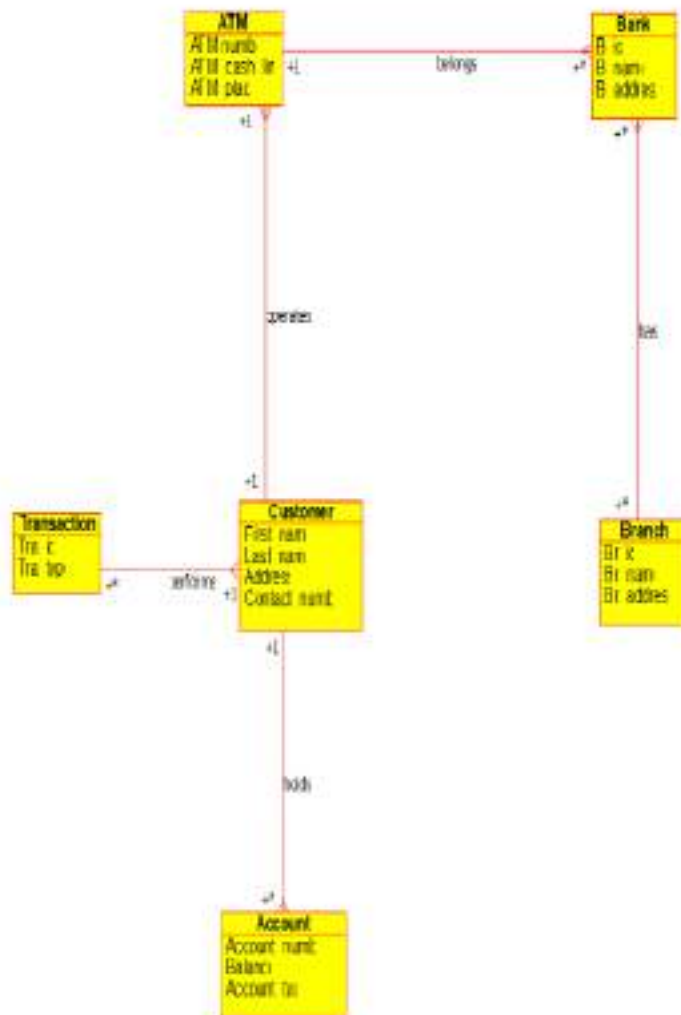
Interoperability:

R1	The system should be highly interoperable.
----	--

Security:

R1	Screen and keyboard must be secured.
----	--------------------------------------

E-R DIAGRAM:



Stock Management System:

Problem Definition:

A new stock maintenance system for a book store is to replace the existing maintenance system which is inefficient. The new stock maintenance system will allow the employee to record information of the books available in the book store and generate reports based on the total amount of sales.

The new system will have a windows-based desktop interface to allow employee to enter the information of sale, purchase orders, change employee preferences and create reports. Employee can only access the information and purchase orders for security purpose. The system retains information on all the books in the shop. The system retains the records of the cost, edition, author, publication of the books. The employee maintains the information of the sale of books. He can add the books at right time and update the database. The customer can view the availability of the required books and the price of the books. The customer can just view them but cannot make any changes.

Scope:

This supplementary specification applies to the stock maintenance system. This specification defines the non-functional requirements of the system, such as reliability, usability, performance and supportability as well as functional requirements that are common across a number of use cases.

Objectives:

The purpose of the document is to define the requirements of the stock maintenance system. This supplementary specification lists the requirements that are not readily captured in the use cases of the use case model. The supplementary specification & the use case model together capture a complete set of requirements on the system.

- To ensure a continuous supply of materials and stock so that production should not suffer at the time of customer's demand.
- To avoid both overstocking and under-stocking of inventory.

- To maintain the availability of materials whenever and wherever required in enough quantity.
- To maintain minimum working capital as required for operational and sales activities.
- To optimize various costs indulged with stock like purchase cost, carrying a cost, storage cost, etc.
- To keep material cost under control as they contribute to reducing the cost of production.
- To eliminate duplication in ordering stocks.
- To minimize loss through deterioration, pilferage, wastages, and damages.
- To ensure everlasting inventory control so that materials shown in stock ledgers should be physically lying in the warehouse.
- To ensure the quality of goods at reasonable prices.
- To facilitate furnishing of data for short and long-term planning with a controlled stock.
- To supply the required material continuously.
- To maintain a systematic record of stock.
- To make stability in price.

Department of I.T.G.N.I.S

SOFTWARE REQUIREMENT SPECIFICATION

Stock maintenance System:

Description:

Stock management is the function of understanding the stock mix of a company and the different demands on that stock. The demands are influenced by both external and internal factors and are balanced by the creation of purchase order requests to keep supplies at a reasonable or prescribed level. The stock maintenance system is basically for the customers who access the information about the stock.

Stake holders:

E	Employees
S	suppliers
A	administrator

General requirements:

R1	The system shall provide static course information
R2	The system shall be able to store static course information
R3	The system shall be able to represent static course information
R4	The system shall provide dynamic course information
R5	The system shall be able to store dynamic course information
R6	The system shall be able to represent dynamic course information
R7	The system shall provide a messaging system

Requirements of employee:

Functional requirements:

R1	The System holds all the details of the all the employees who are working in the organization
R2	The system holds the details of all the go downs and employees which are part of the organization
R3	The system also allows them to view the list of inward entries.

R4	The system allows the go down employee to log into the system and enter their inwards entries related to their go down.
R5	The system allows the users to change their password for future security.
R6	The system allows the go down manager to log into the system and enter stock return entries and the reason for return.

Non-functional requirements:

Security:

R1	The system shall allow the users should be able to change their passwords for increased security
----	--

Availability:

R1	The system should be available over the intranet so that the Users like the godown managers & clerks can use the system from their respective locations which could be anywhere in the organization.
R2	The system shall not have unexpected downtime

User friendliness:

R1	The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.
----	--

Privacy:

R1	For gaining entry into the system the user should enter user info and the user should be able use login & passwords for gaining access to the system.
----	---

Interoperability:

R1	The system should be designed in such a way that it is easy to enhance it with more functionality. It should be scalable & easily maintainable.
----	---

Accessibility:

R1	The system should be easy to accessible.
----	--

Requirements of supplier:

Functional requirements:

R1	It holds the details of all Product Stocks held in the ware-house of the company
R2	The system allows the go down supplier to log into the system and enter their outward entries and their purpose related to their go down.
R3	Whenever an inwards entry is entered then accordingly the stock number will be automatically updated.
R4	Whenever an outward entry is entered then accordingly the stock number will be automatically updated.
R5	Whenever a return entry is entered then accordingly the stock number will be automatically updated if the reason is order cancelled otherwise It need not update the stock.

Non-functional requirements:

Security:

R1	The system shall allow the users should be able to change their passwords for increased security
----	--

Availability:

R1	The system should be available over the intranet so that the Users like the go down managers & clerks can use the system from their respective locations which could be anywhere in the organization.
R2	The system shall not have unexpected downtime

User friendliness:

R1	The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.
----	--

Privacy:

R1	For gaining entry into the system the user should enter user info and the user should be able use login & passwords for gaining access to the system.
----	---

Interoperability:

R1	The system should be designed in such a way that it is easy to enhance it with more functionality. It should be scalable & easily maintainable.
----	---

Accessibility:

R1	The system should be easy to accessible.
----	--

Requirements of administrator:

Functional requirements:

R1	The system allows the admin to view the list of users and take the print.
R2	The system allows admin to generate go down details report.
R3	The system allows admin to generate inwards details report.
R4	The system allows admin to generate outwards details report.
R5	The system allows admin to generate returns details report.
R6	The system allows admin to generate stock statement report.

Non-functional requirements:

Availability:

R1	The system should be available over the intranet so that the Users like the godown managers & clerks can use the system from their respective locations which could be anywhere in the organization.
R2	The system shall not have unexpected downtime

User friendliness:

R1	The system should be easy to understand and organized in a structured way. The users should also receive feedback about any errors that occur.
----	--

Privacy:

R1	For gaining entry into the system the user should enter user info and the user should be able use login & passwords for gaining access to the system.
----	---

Accessibility:

R1	The system should be easy to accessible.
----	--

Security:

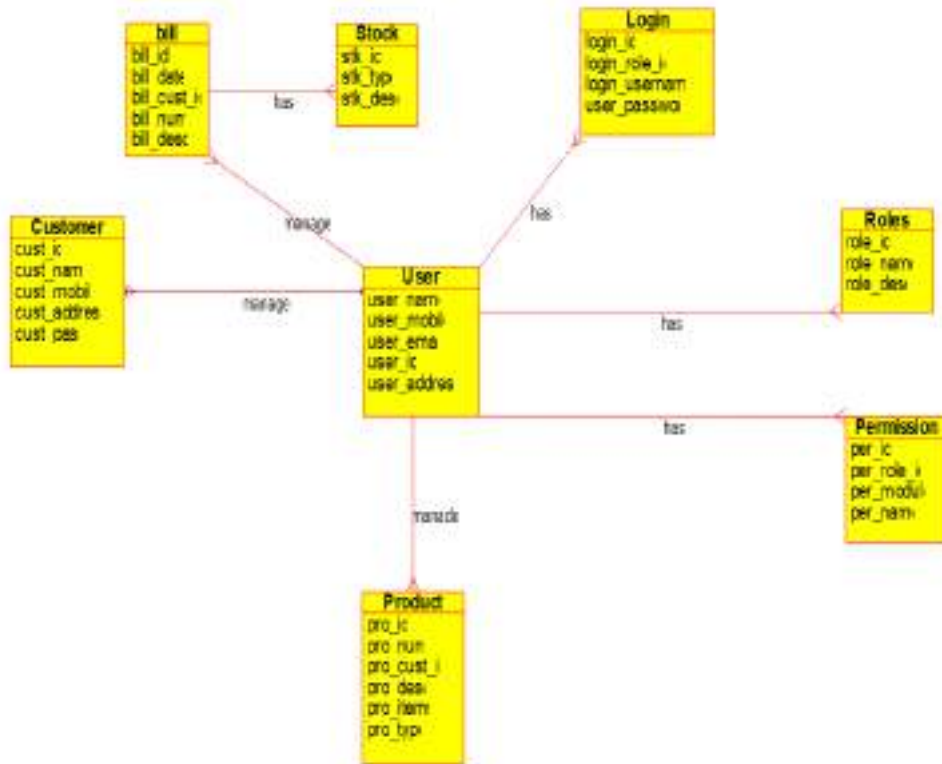
R1	For gaining entry into the system the admin should register user info and the user should be able use login & passwords for gaining access to the system.
----	---

Interoperability:

R1	There should be no limitation about the hardware platform that is to be used to run the system.
----	---

Department of IT/GNITS

ER DIAGRAM:



Depc

HUMAN RESOURCE MANAGEMENT DOMAIN-

1. QUIZ SYSTEM –

Problem Definition-

Quiz System is a web-based application which is majorly used for conducting quizzes. Here the quizzes will be conducted with time constraints. The faculty or any lecturer can post the quiz and the students can take the quiz at the given time and only the faculty members have the access to make change to the quiz at any time. Students are provided the flexibility to choose among different types of aptitude and programming language tests. It provides an online comprehensive solution to manage quiz where the individuals are participating in a team.

Scope-

The users/students should have a valid user id and password to access the quiz and participate in them. Each question will have a time limit to answer if the participant does not answer the question, they will not be awarded any marks and the next question appear.

Objectives-

- For people to participate they should have valid credentials.
- Time limit will be there for each question compared as the normal one of having a time limit for the whole quiz to make sure that copying does not take place.
- Notifications will be provided when quiz takes place or when the time limit is about to get over.
- Only faculty members can make changes to the quiz questions at any time and can schedule the quiz at their wish.
- Only marks will be visible to the students at the end of the test along with the correct answers displayed.
- This System reduces a lot of work for the faculty as the paper work decreases and also saves a lot of time as the marks are provided on the spot when conducted online, but when conducted offline on paper the faculty should correct each and every paper which takes a lot of time.
- For some quizzes web camera is a must in order to decrease the chance of copying.
- It is mainly useful for educational institutions.

SRS DOCUMENT -

Introduction-

This document specifies the requirements for Quiz System. The requirements are grouped by their stakeholders, functional and non-functional requirements are separated.

Stakeholders-

Administration	A
Users	U

Functional Requirements for Users-

1	The system should ensure that all the users should have valid login credentials.
2	The system displays the rules and regulations before each quiz.
3	The system should send notifications to the users when the quiz starts and just 10 to 15 mins before the quiz starts.
4	The users should register for any quiz before participating.
5	The system should automatically generate the score after the completion of each quiz, the score of each participant will only be visible to them
6	The system will only display the ranks of the participants and not their score.
7	The system provides a feature of messaging in which users can communicate and chat with one another.
8	The system allows the users to edit their personal information.
9	The system will also send notification in the form of e-mail's to the users on the quizzes based on their previous history and their interests.
10	The system ensures that participants get disqualified if window proctoring occurs more than once, for the 1st time a warning will be given.

Functional Requirements for Administration-

1	The system should ensure that all the users should have valid login credentials.
2	Only the administrators of the quiz can change the timings and cancel the quiz at any time.
3	The system ensures that the administrators can only view all the users scores and ranks.
4	The system ensures that the administrators can chat with the users and other faculty members.
5	The system ensures that they can change the format and other details of the quiz at any time as per the guidelines given.

Non-Functional Requirements for Administration & Users-

1.Security-

1	The system will provide security to the user's data
2	The system will be secure.
3	The system shall allow students to view only their own grade
4	The system will allow the lecturers/Faculty to view the grades of all the students.

2.Accessibility-

1	The system can be easily accessed.
2	The system is user friendly.
3	The system will have a single login for all content access.

3.InterOperability-

1	The system is highly interoperable.
2	The system shall provide an export to commonly used calendar formats

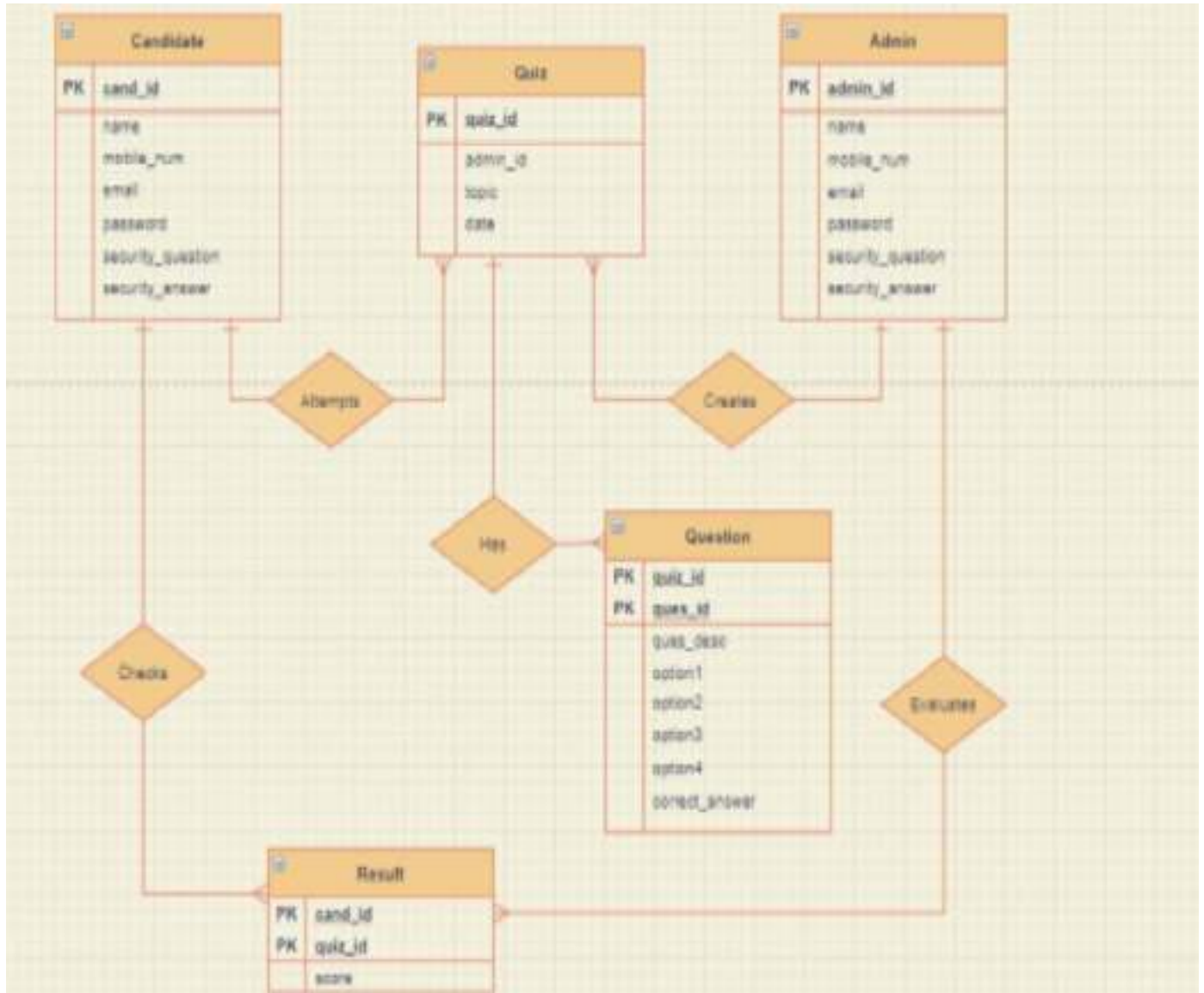
4.Reliable-

1	The system is highly reliable.
---	--------------------------------

5.Availability –

1	The system has high availability.
2	The system shall not have unexpected downtime
3	The system shall have downtime only during low-intensity hours

ER DIAGRAM –



2.EMAIL CLIENT SYSTEM –

Problem definition -

The email client system is a virtual application that has a strongly built email client that allows users to compose and send emails to one or more email addresses and at the same time allows to receive, read emails to making communication very efficient and effective. It provides a central interface for receiving, composing and sending emails of configured email addresses.

Scope -

It is highly secure and reliable as it uses advanced SMTP and POP3 services to send and receive emails. The system provides a robust understanding of an email client and allows for efficient sending and receiving of emails. Saves a lot of time and resources as it is accessible through internet.

Objectives -

- This software system allows users to compose emails as per user requirements along with title destination email and the message body.
- Reduces a lot of time, human effort, it is well authenticated and can be accessed from the comfort of any geographical location(virtually)
- The email client composes of an inbox as well as spam folder for receiving emails.
- Emails consisting of content as per predefined rules (not overly Populated with images and links) are saved in email inbox.
- Emails containing images or links above a certain limit are saved in the spam folder.
- The system is developed having high robustness and security.
- The system prevents the unsolicited bulk messages by putting it into spam folder and the appropriate ones in inbox folder.
- It also provides clients identity protection.
- The system prevents unauthorized access of clients email through Illicit means. Prevents hacking.

- Sometimes the system can consider the unsolicited emails valid and can put it in inbox instead of putting in spam folder.
- Searching can be done based on the title of the email, date and generally, they are filtered by date received.
- There is module called All Mail which displays all the mails received.
- Important mails can be starred, archived or sent to Important mail.
- The sent messages can viewed in the Sent box, similarly all the draft messages can be viewed in Drafts.
- There is a trash folder where all the deleted messages can be seen.

SRS DOCUMENT –

Introduction-

This document specifies the requirements for E-mail client system. The requirements are grouped by their stakeholders, functional and non-functional requirements are separated.

Stakeholders-

Sender	S
Receiver	R

Functional Requirement Document for E-mail Client System(Both sender and receiver have common)-

1	The system should ensure that the users have valid Mail account and credentials.
2	If the mail does not reach the receiver or if the receiver's mail ID entered is incorrect then an error message will be displayed to the sender.
3	The system will send a mail to the user if any new login attempt is made for another system.
4	The system will also ensure that the user sets a strong password.
5	The system also provides the users the option of 2 step authentication if required.
6	After a mail is deleted the system ensures that it stays in the bin for 30 days so that the user can recover it whenever they want.
7	Once a mail is sent it cannot be deleted but before it is sent it can be edited many times and will be saved in drafts.

8	The system sends a notification in the form of a vibration or buzz sound when an email is received.
9	The system also allows the user to send not just text messages but also files in the form of PDF's and all.
10	The system also allows the user to classify the mails as spam, promotions and social so that they need not waste time looking after every mail.
11	The system also maintains a huge database to store the user's data and the mails.

Non-Functional Requirements for Sender & Receiver-

1.Maintainability-

1	The system shall be easily maintainable.
---	--

2.Reliable-

1	The system is reliable.
---	-------------------------

3.User-Friendly-

1	The system is user-friendly.
2	The system shall have a single login to access all the content.
3	The system shall have a consistent UI.

4.Security-

1	The system protects the user's privacy.
2	The system allows the users to edit only their personal information.

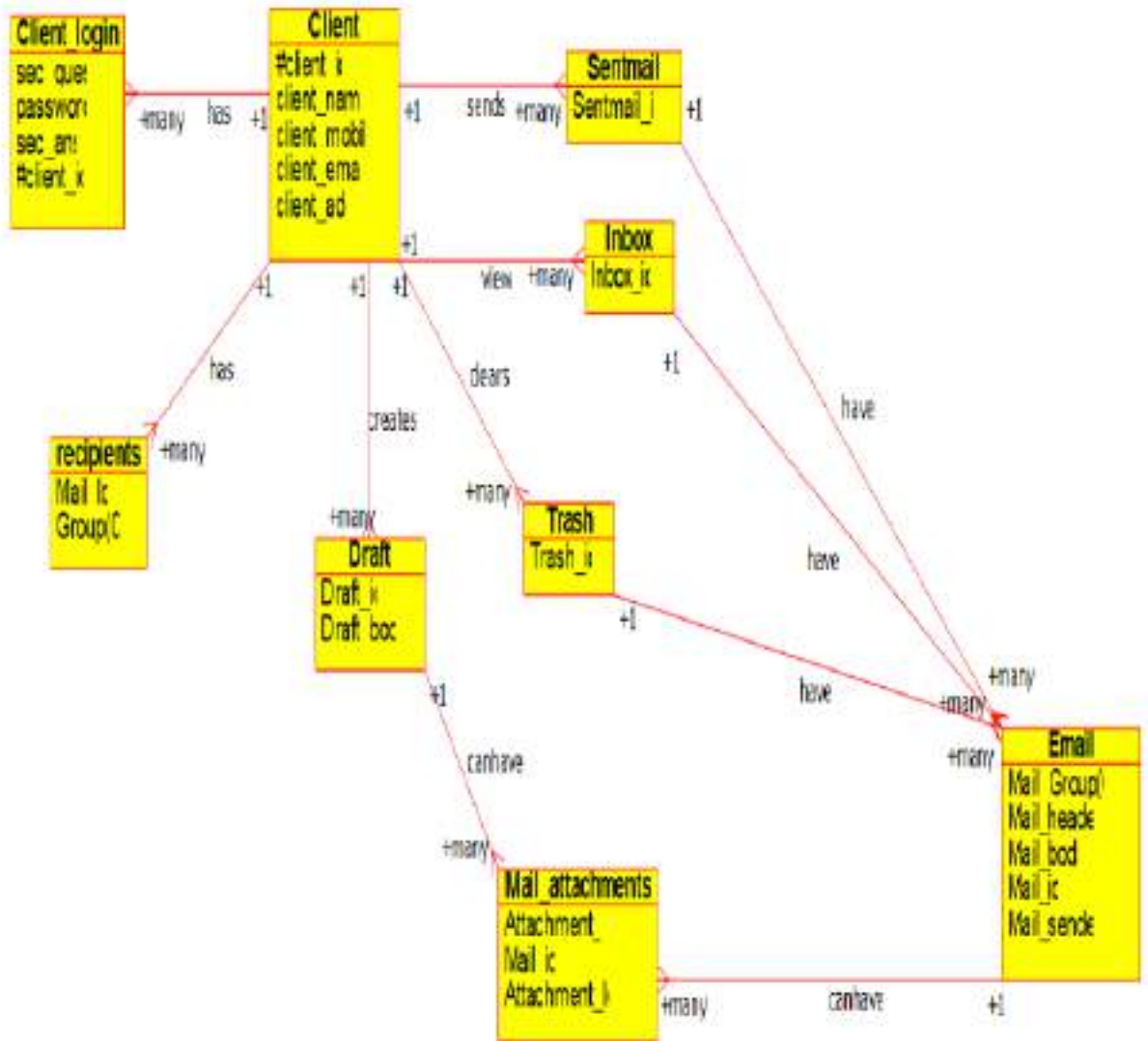
5.Interoperability-

1	The system is highly interoperable.
---	-------------------------------------

6.Other-

1	The system shall be accessible by disabled (blind) users, who should be able to navigate the system and have access to all content and functionality.
2	The system shall only allow those who send and receive the data to view it, others have no access and cannot view it at any cost.

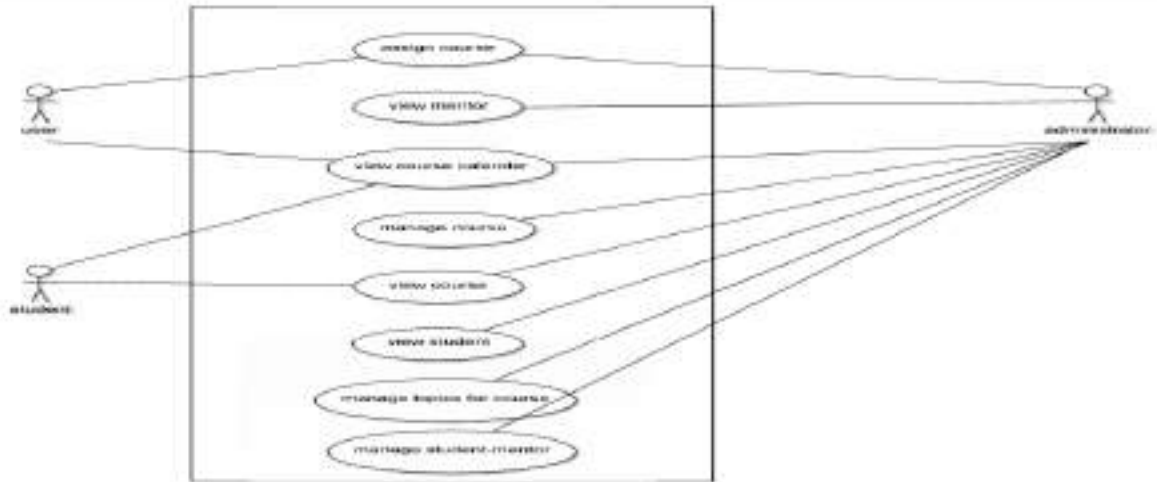
ER DIAGRAM –



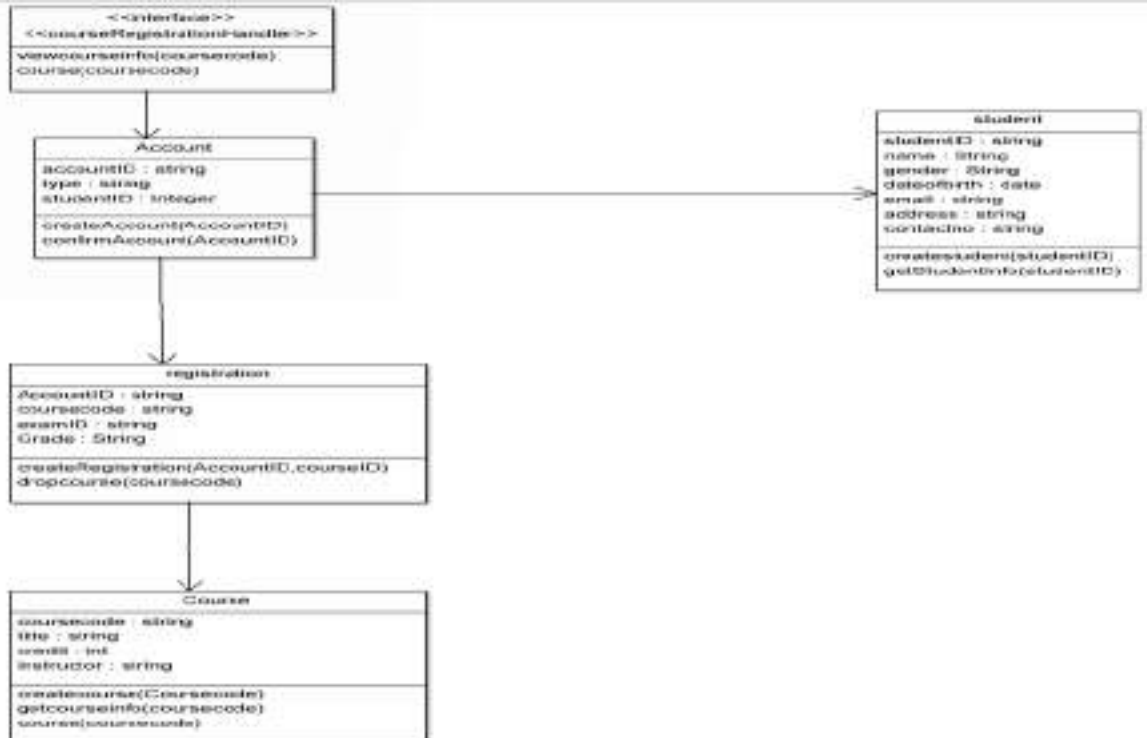
UML DIAGRAMS

COURSE REGISTRATION SYSTEM

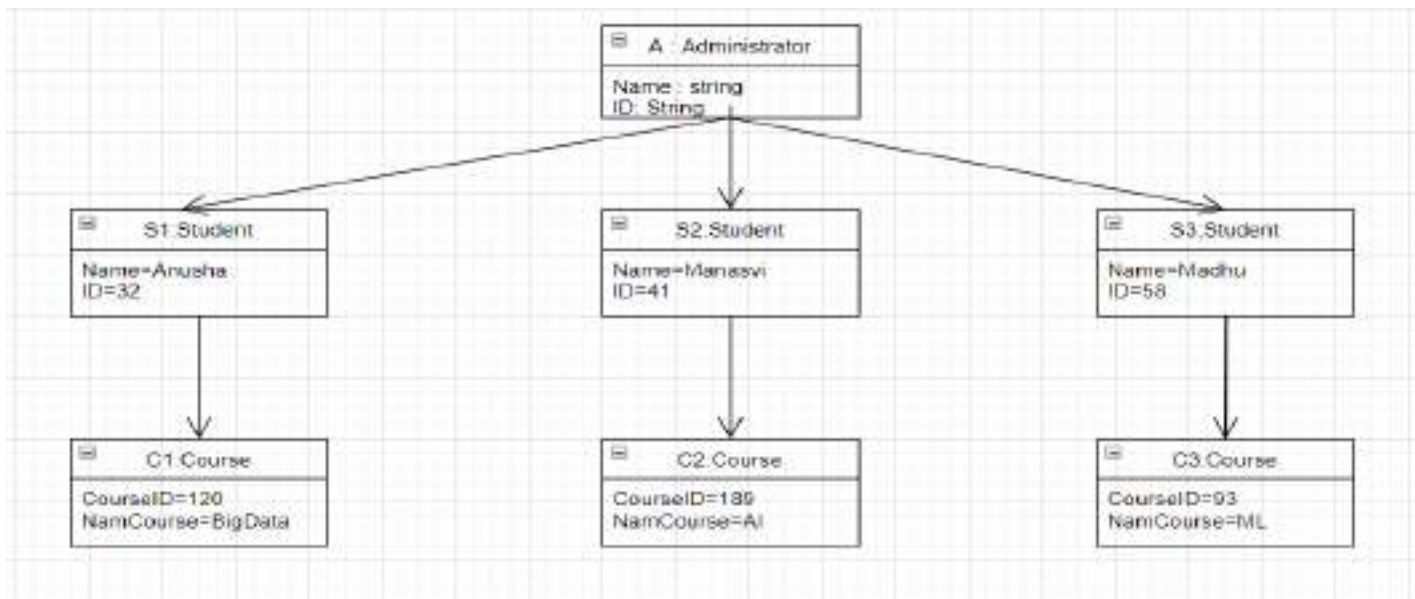
USE CASE DIAGRAM-



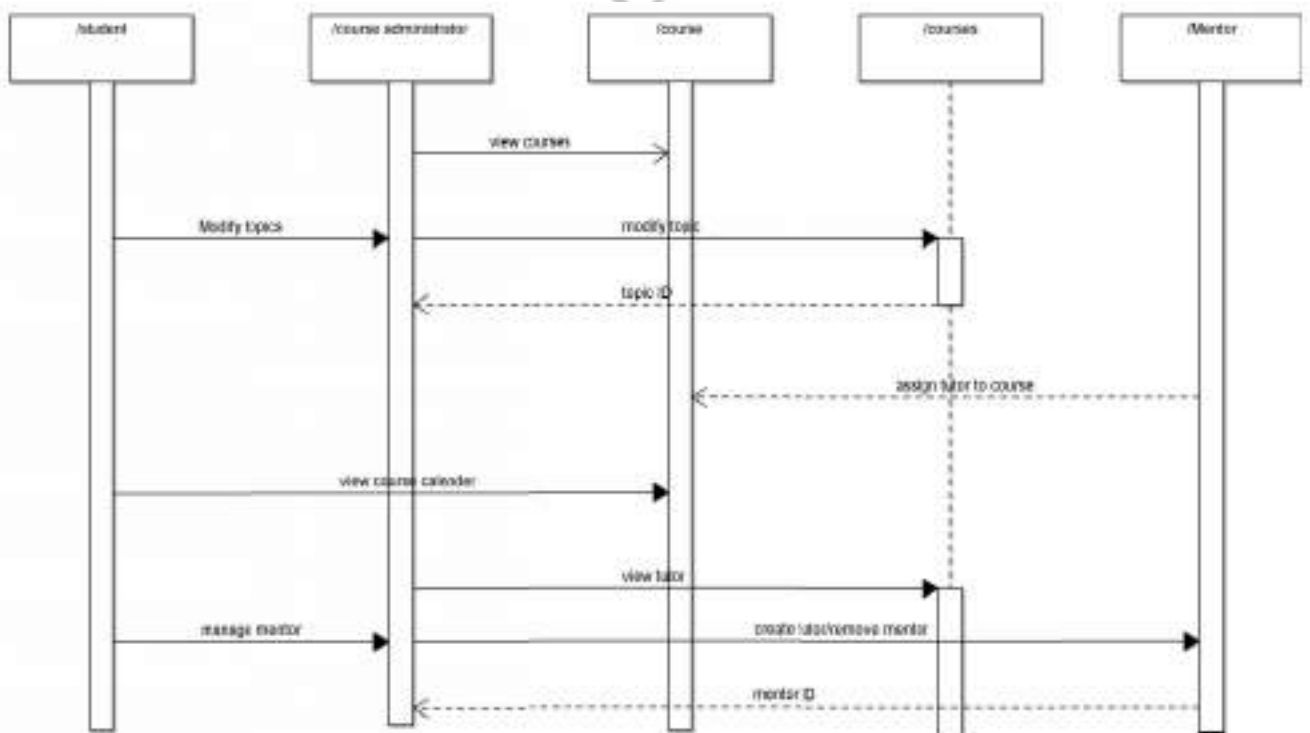
CLASS DIAGRAM –



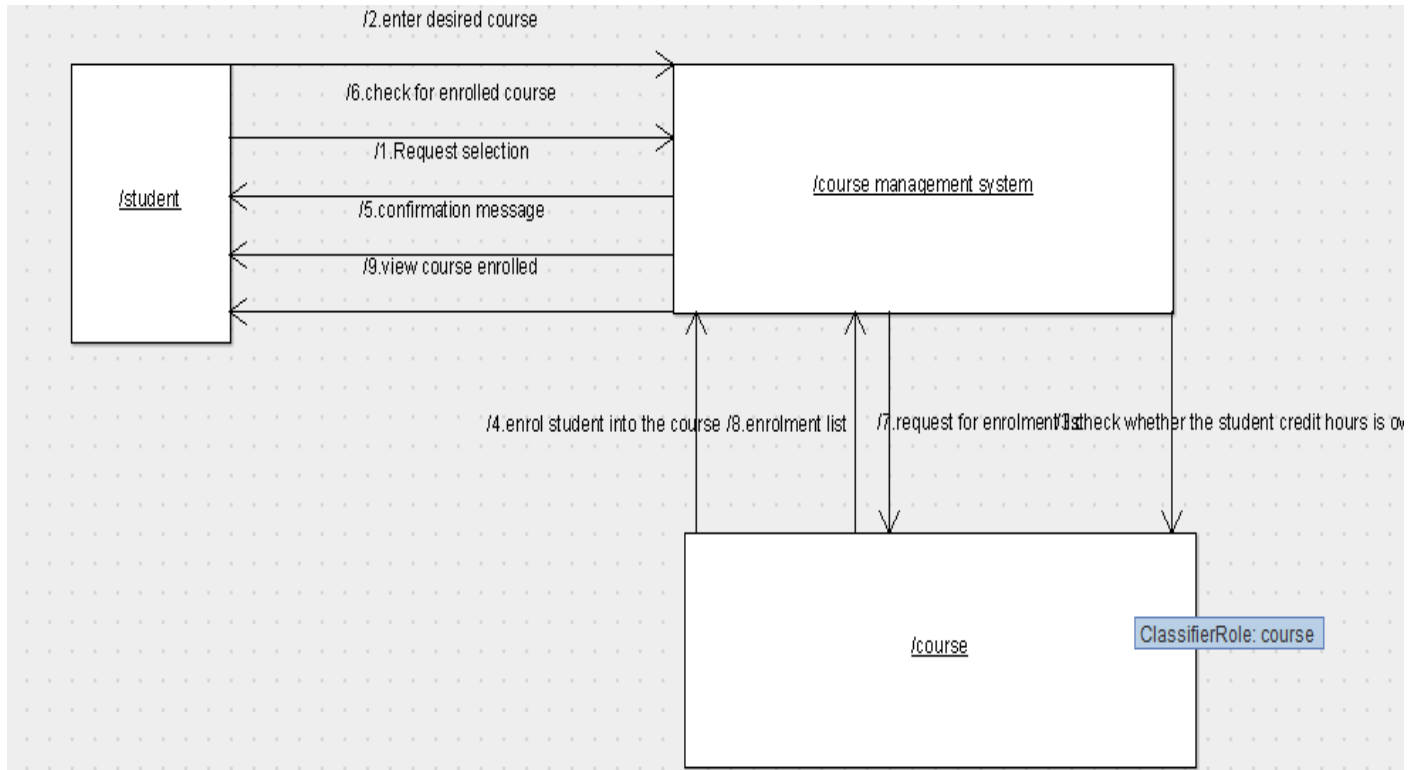
OBJECT DIAGRAM –



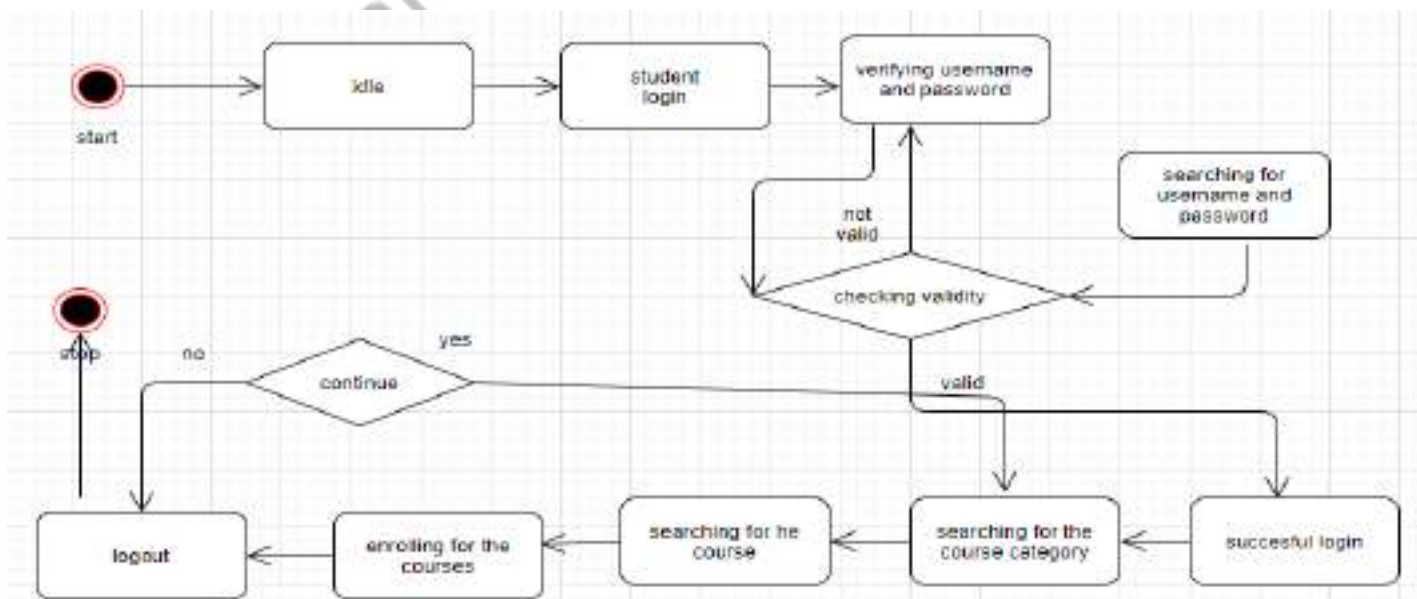
SEQUENCE DIAGRAM –



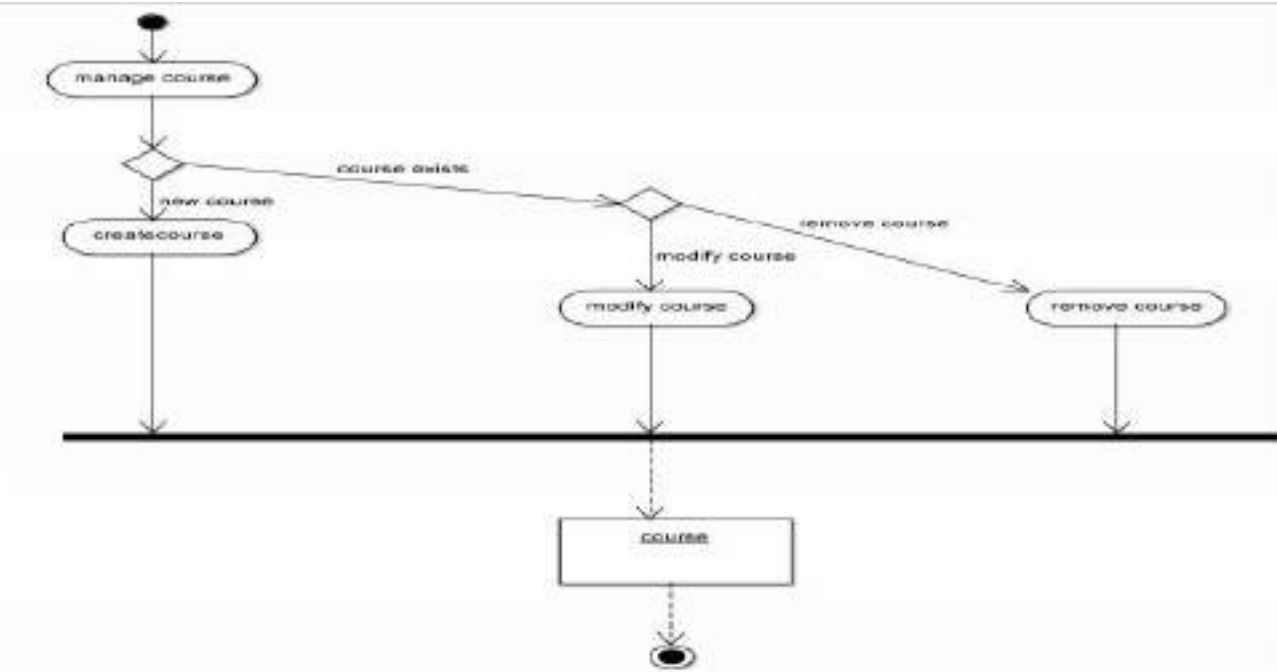
COLLABORATION DIAGRAM –



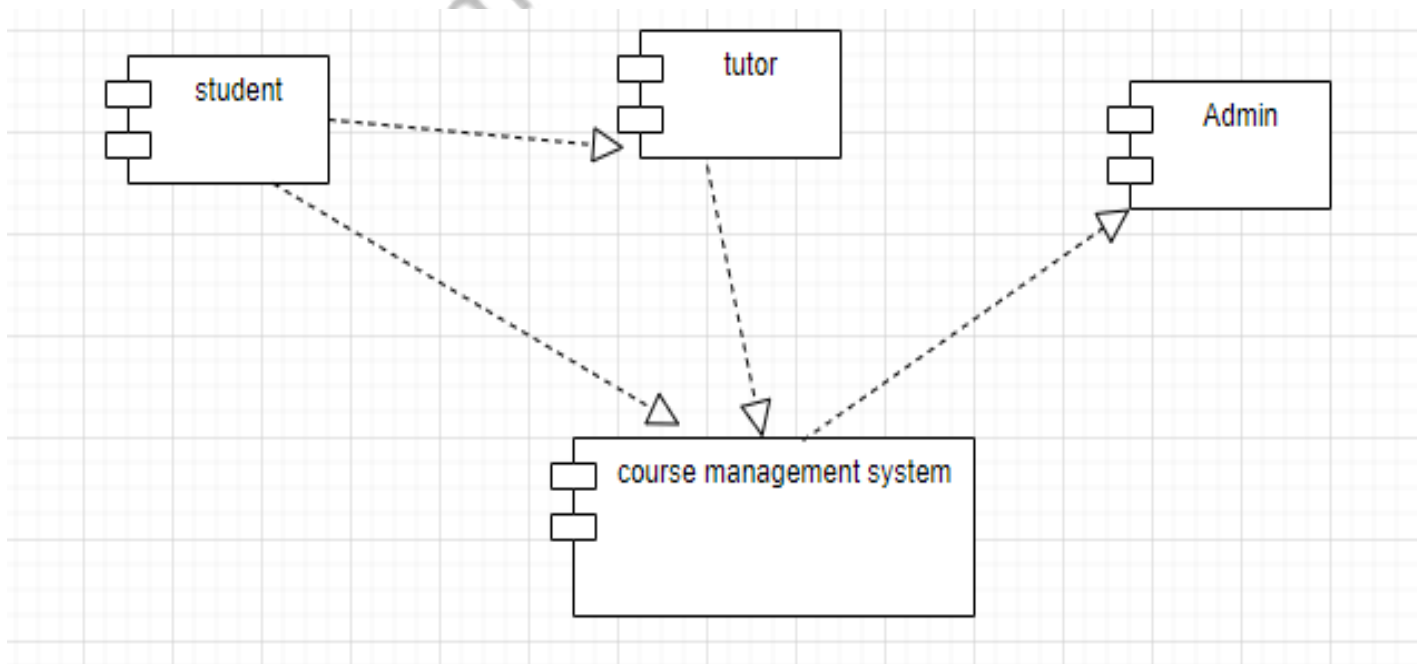
STATECHART DIAGRAM –



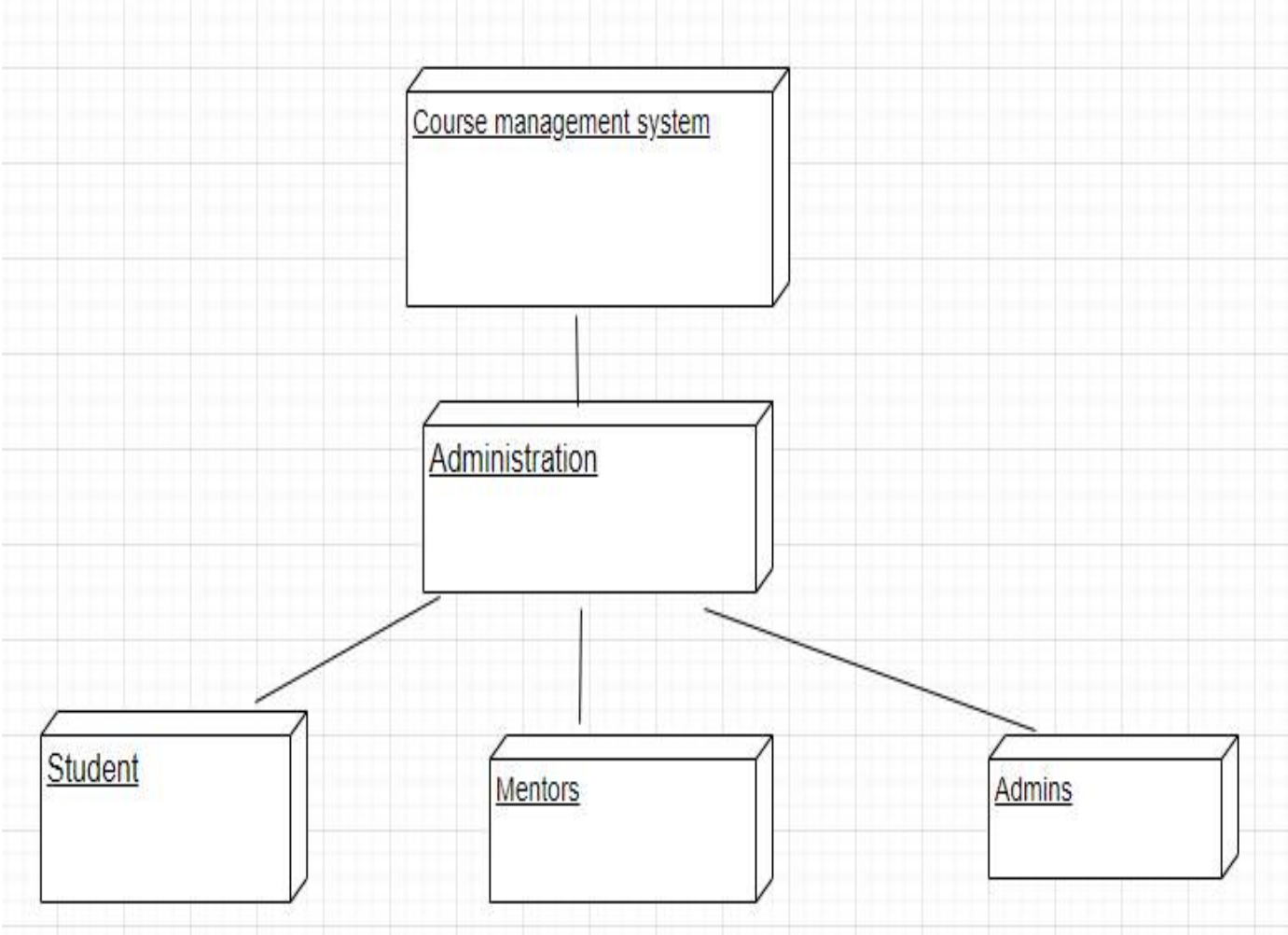
ACTIVITY DIAGRAM –



COMPONENT DIAGRAM -

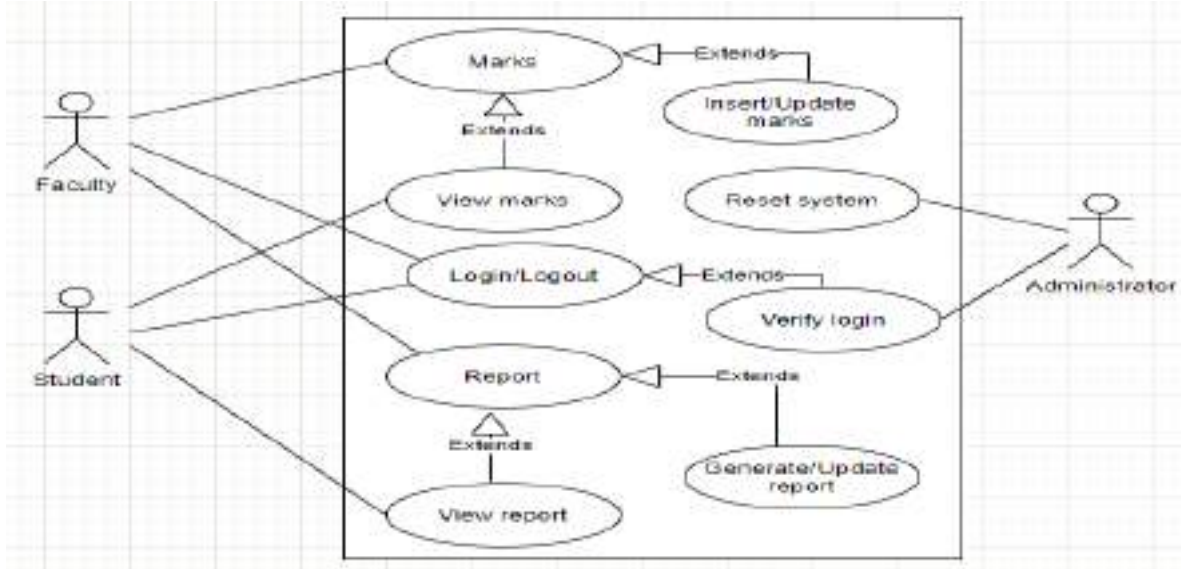


DEPLOYMENT DIAGRAM –

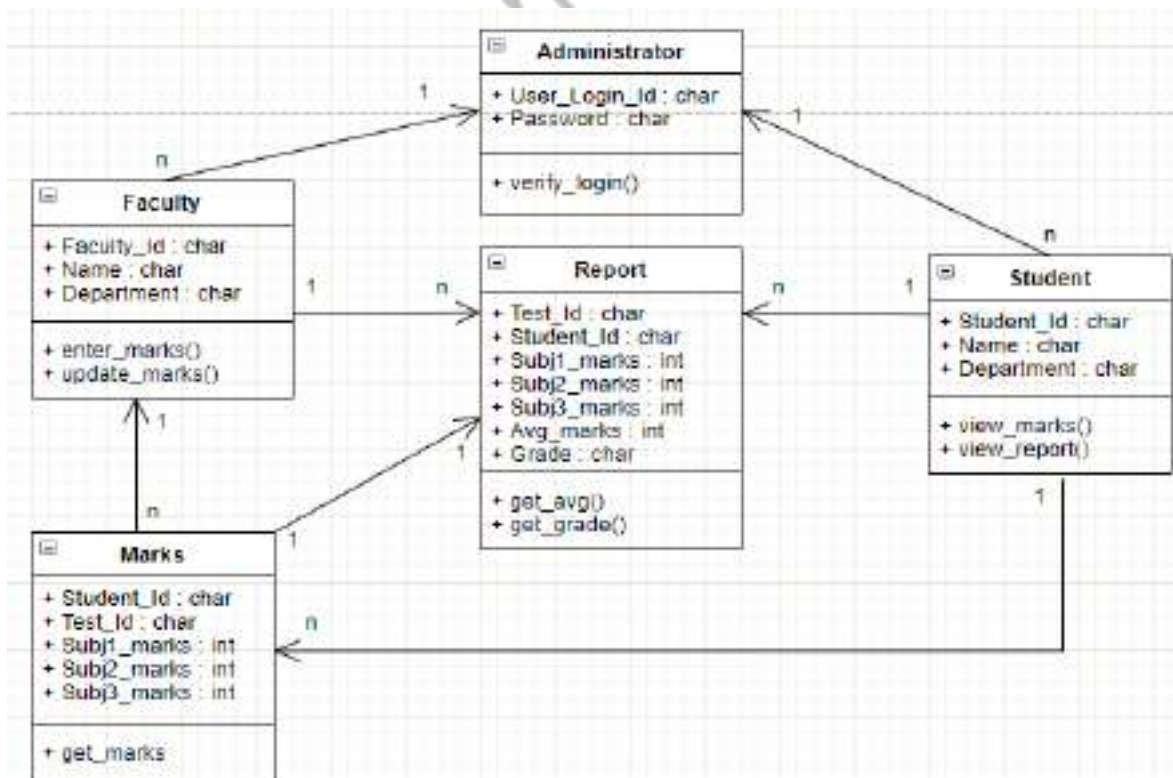


STUDENT MARKS ANALYZING SYSTEM

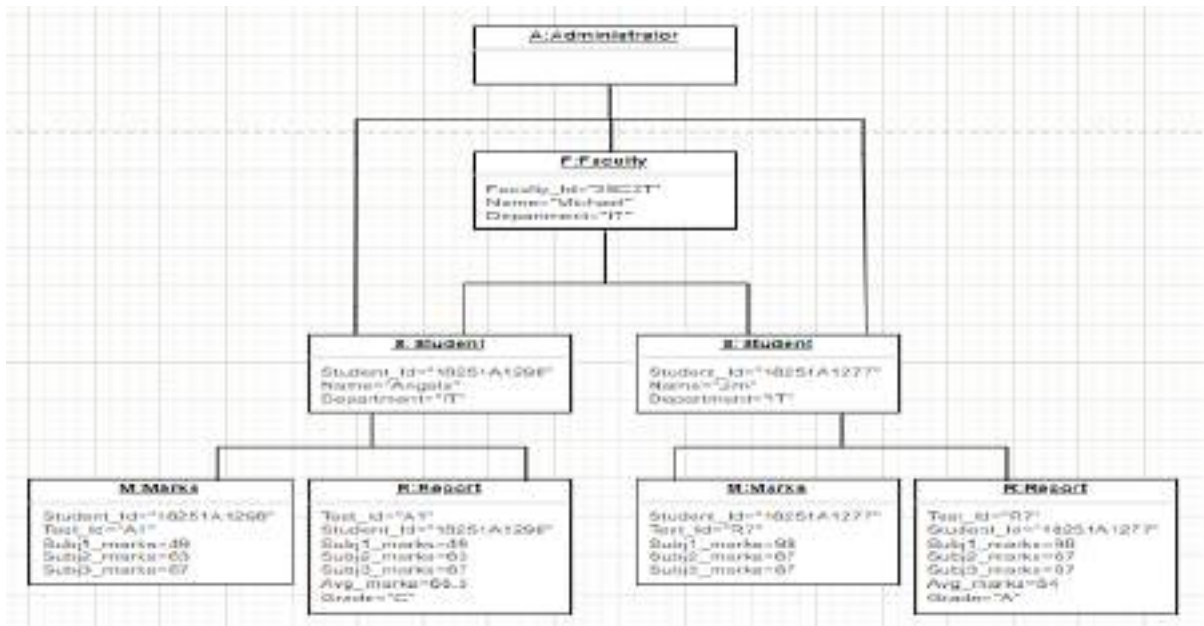
Use-case Diagram



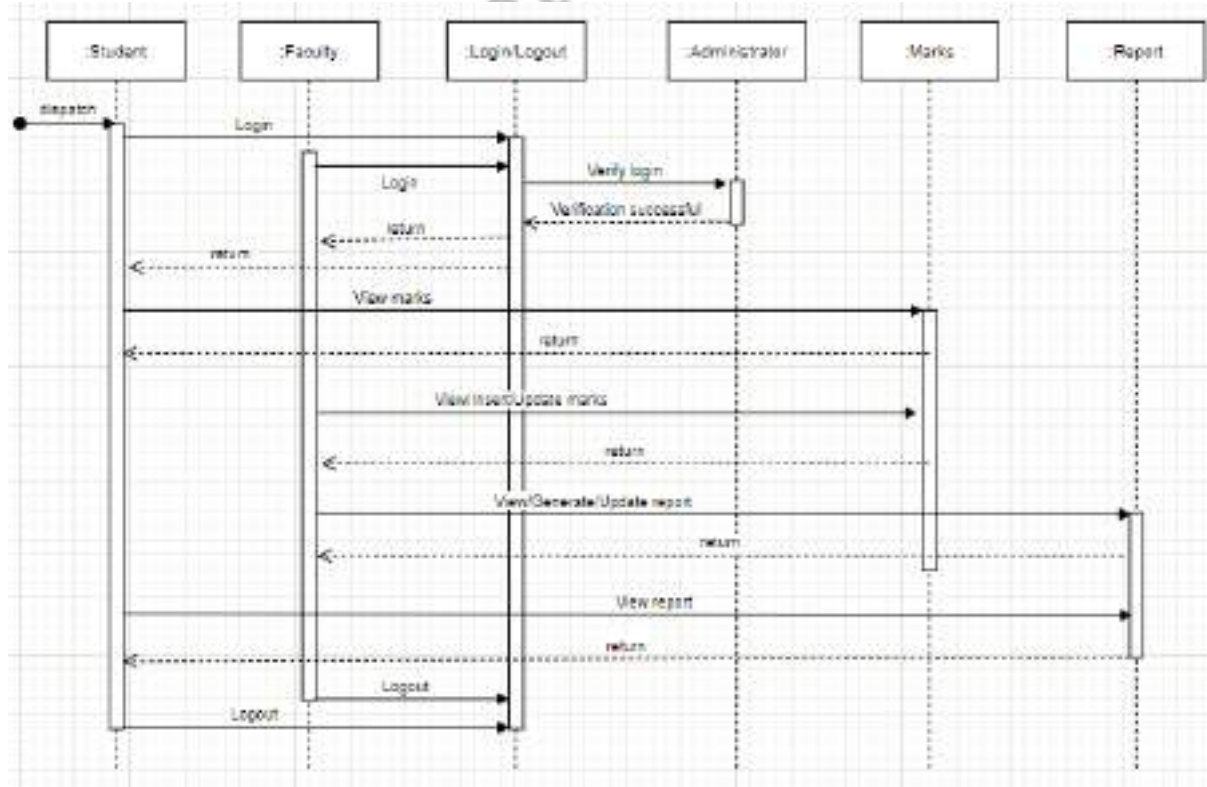
Class Diagram



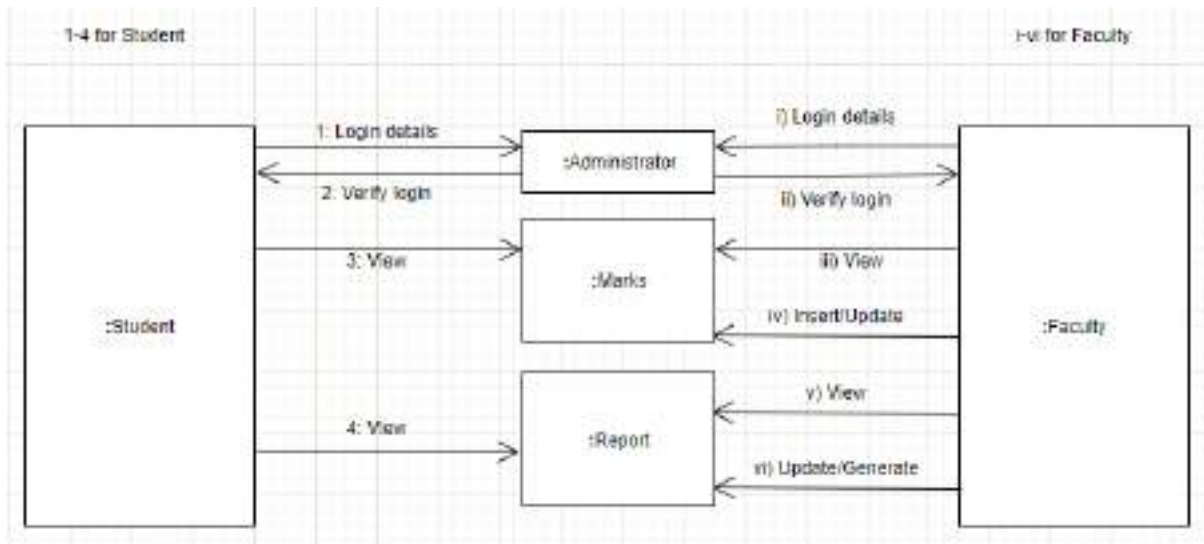
Object Diagram



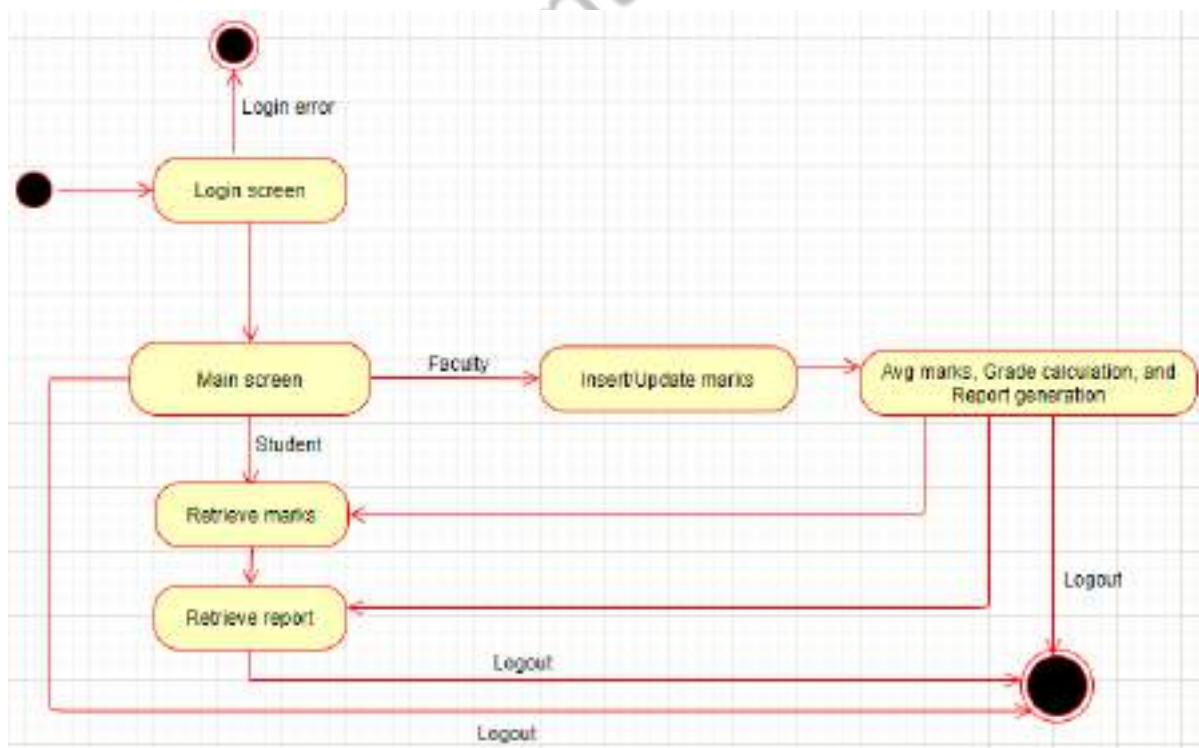
Sequence Diagram



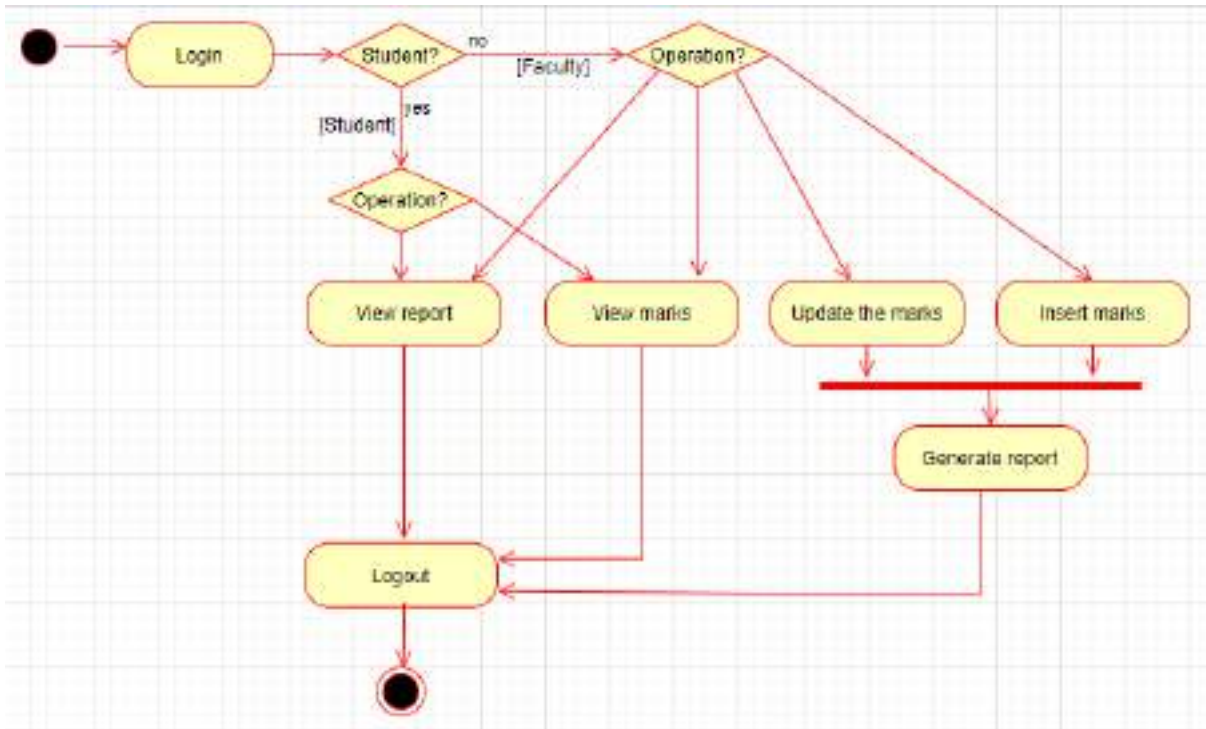
Collaboration Diagram



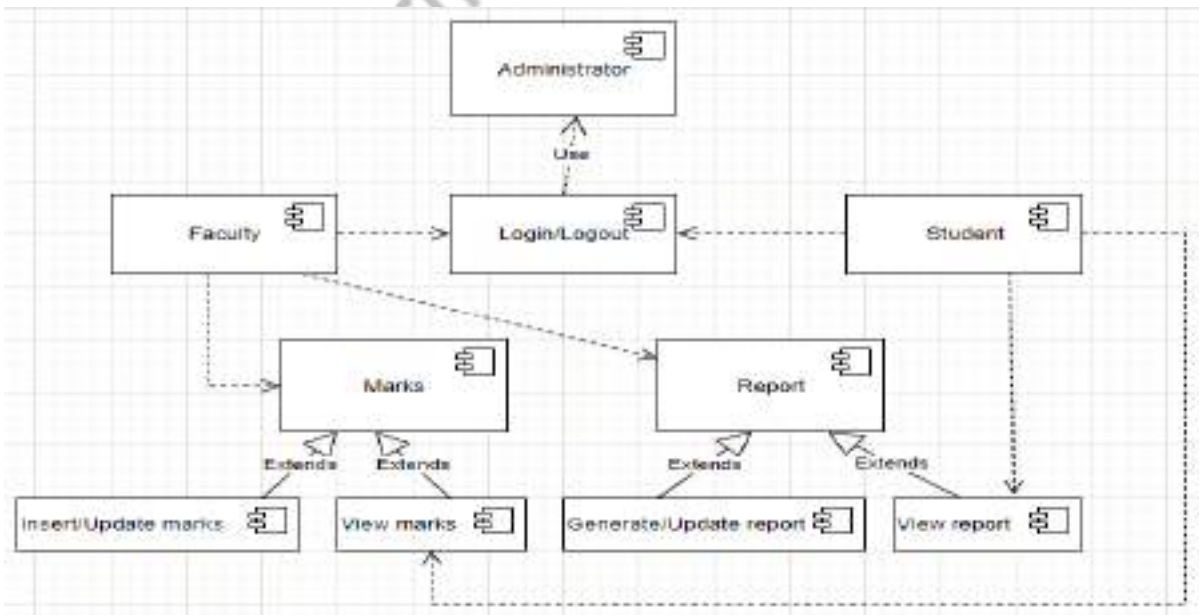
State Machine Diagram



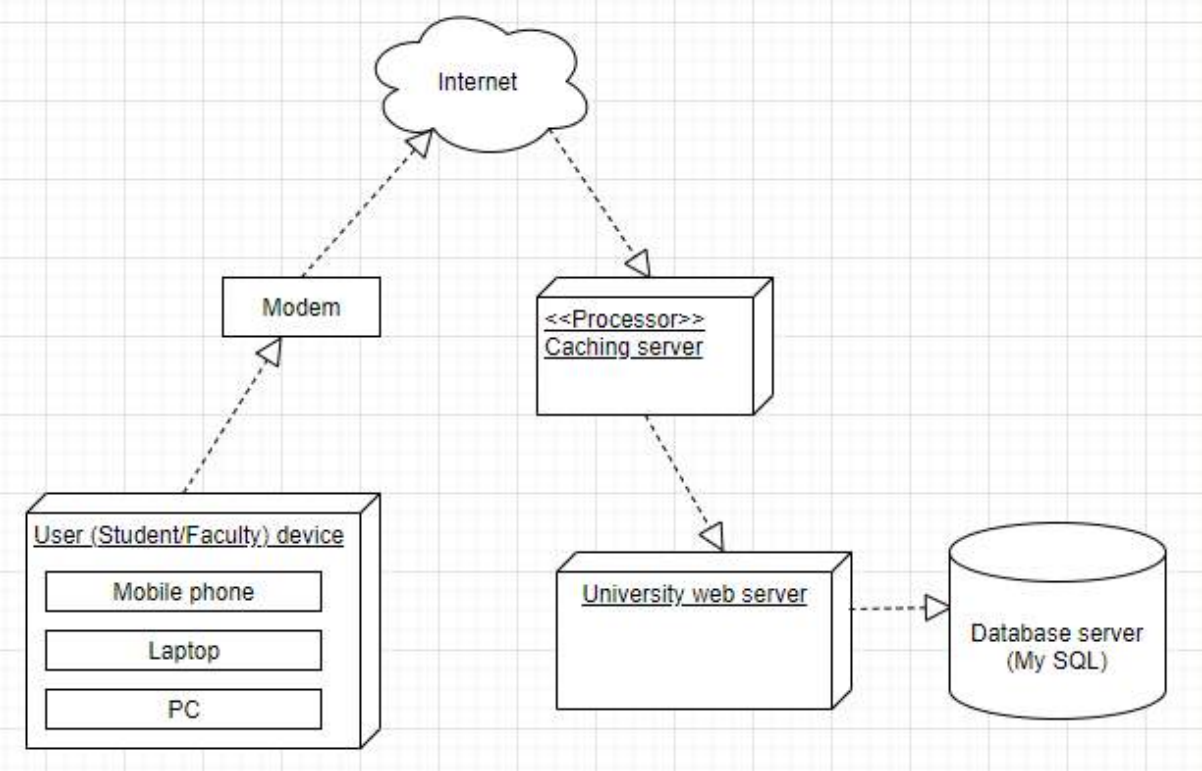
Activity Diagram



COMPONENT DIAGRAM



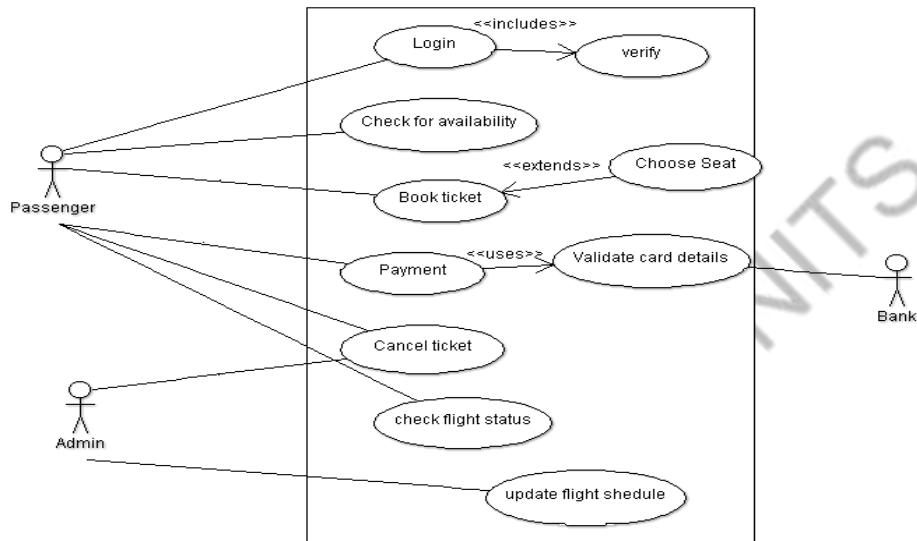
Deployment Diagram



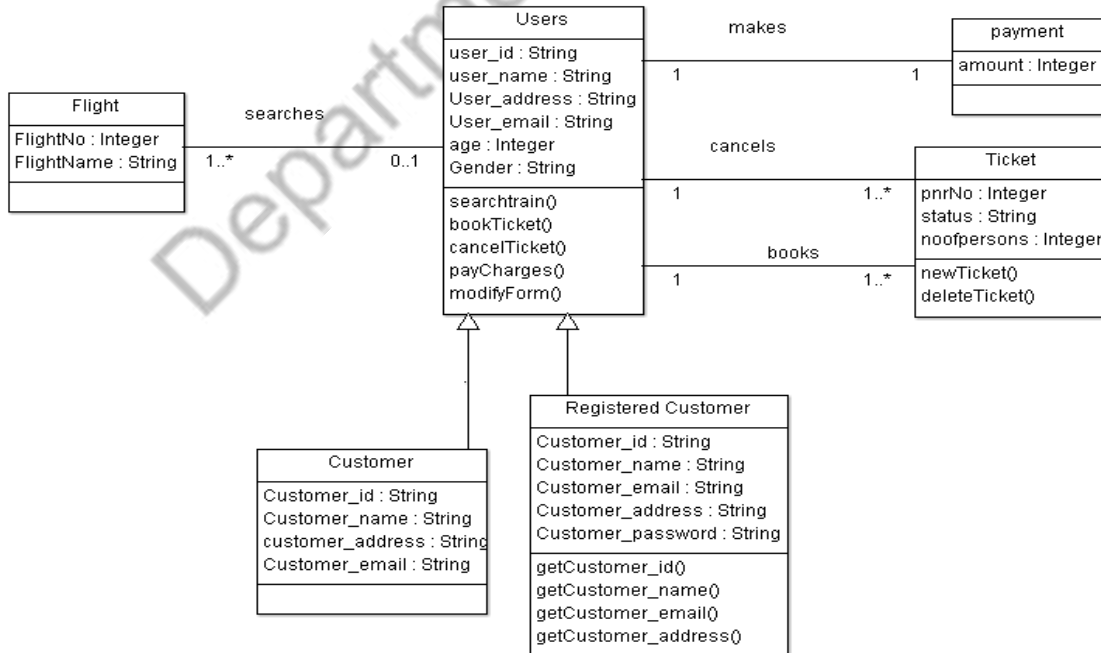
Depar...

ONLINE TICKET RESERVATION SYSTEM

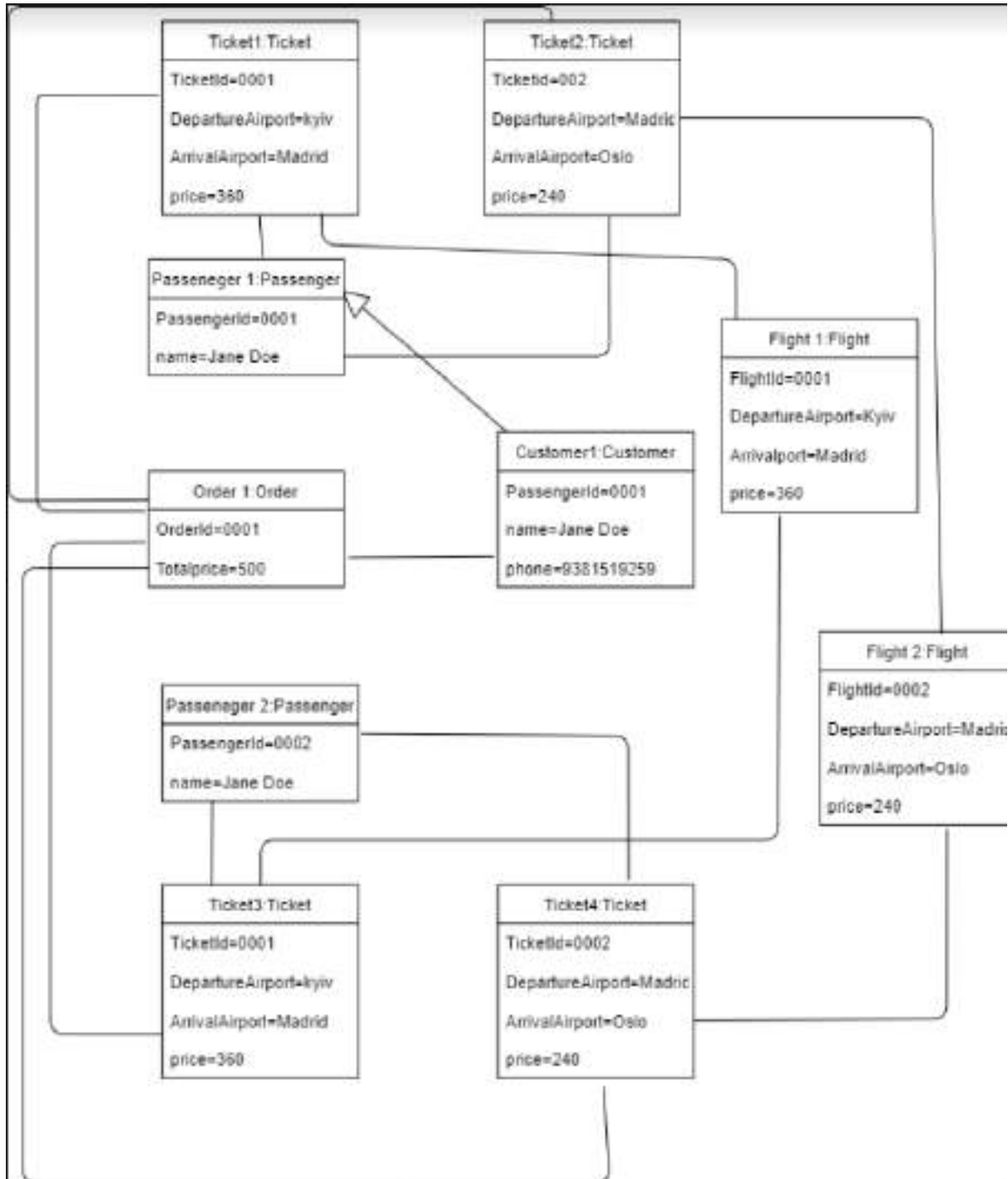
USECASE DIAGRAM:



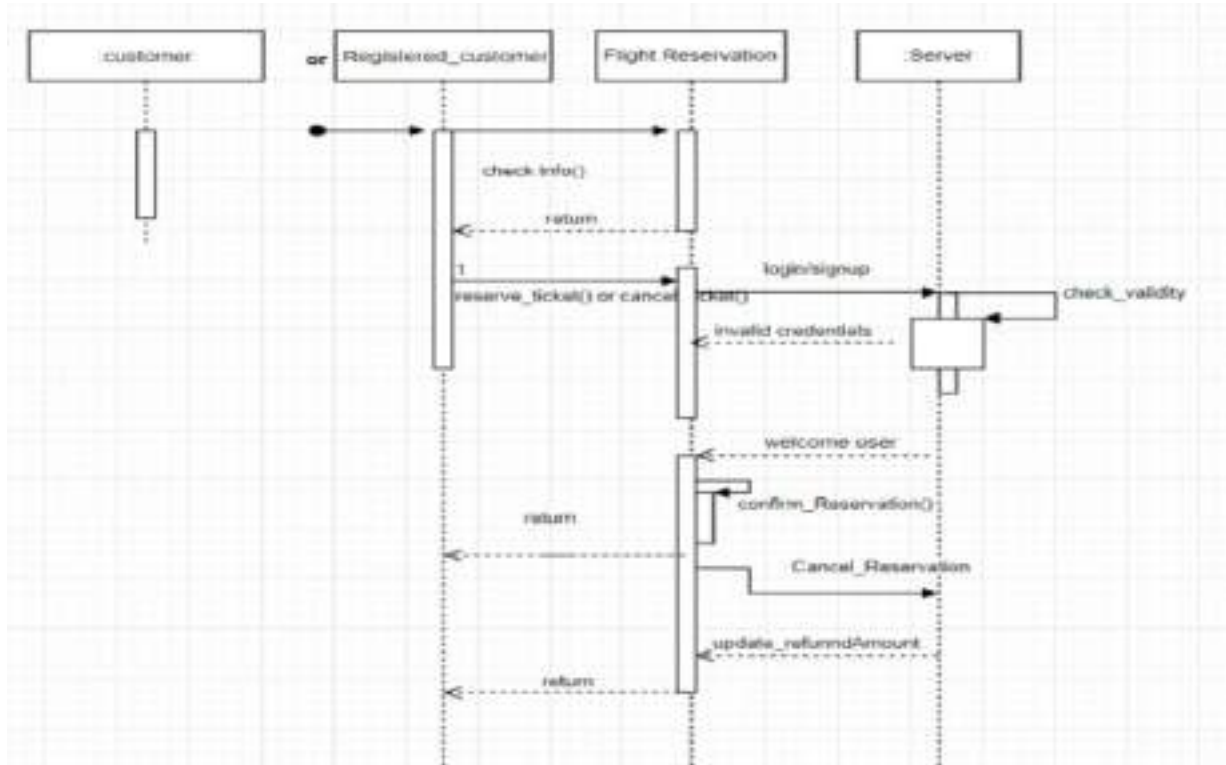
CLASS DIAGRAM



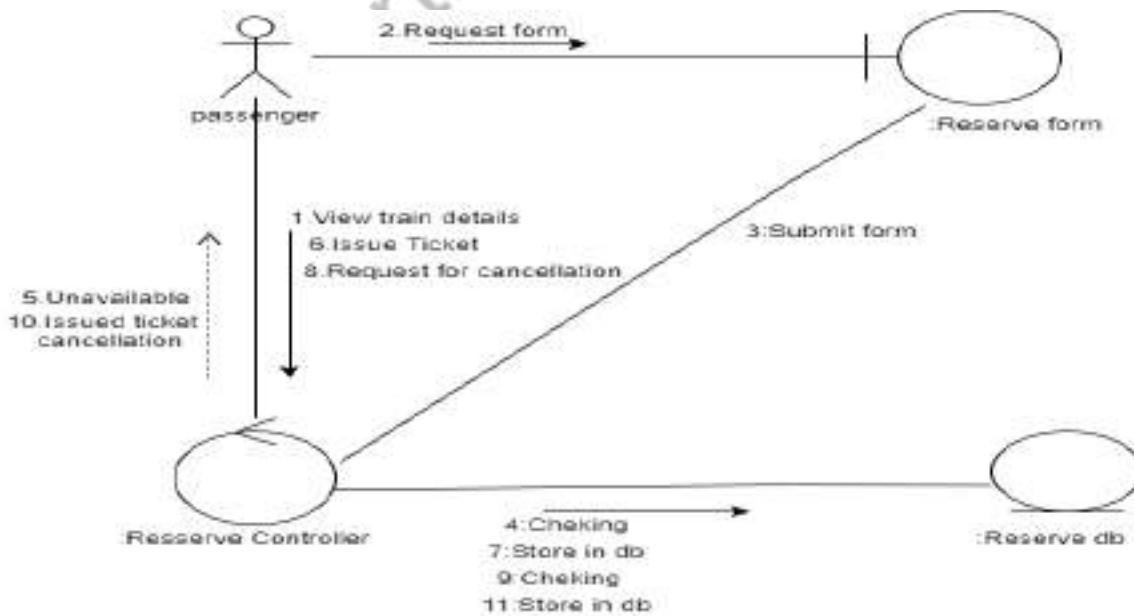
OBJECT DIAGRAM



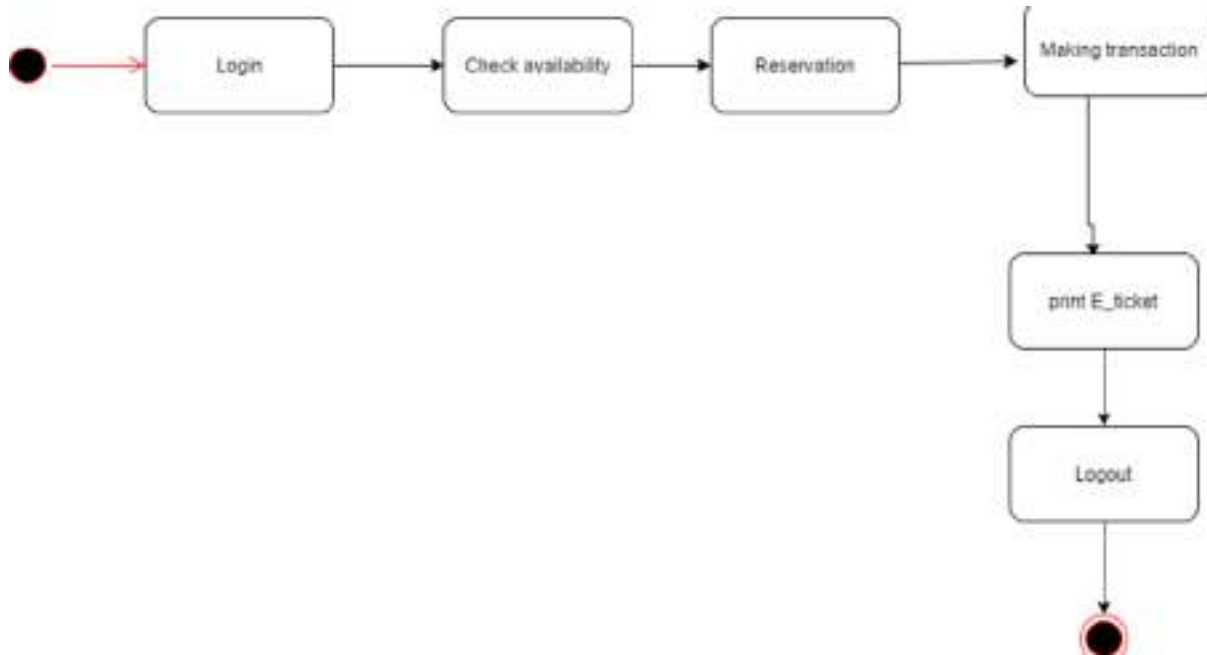
SEQUENCE DIAGRAM



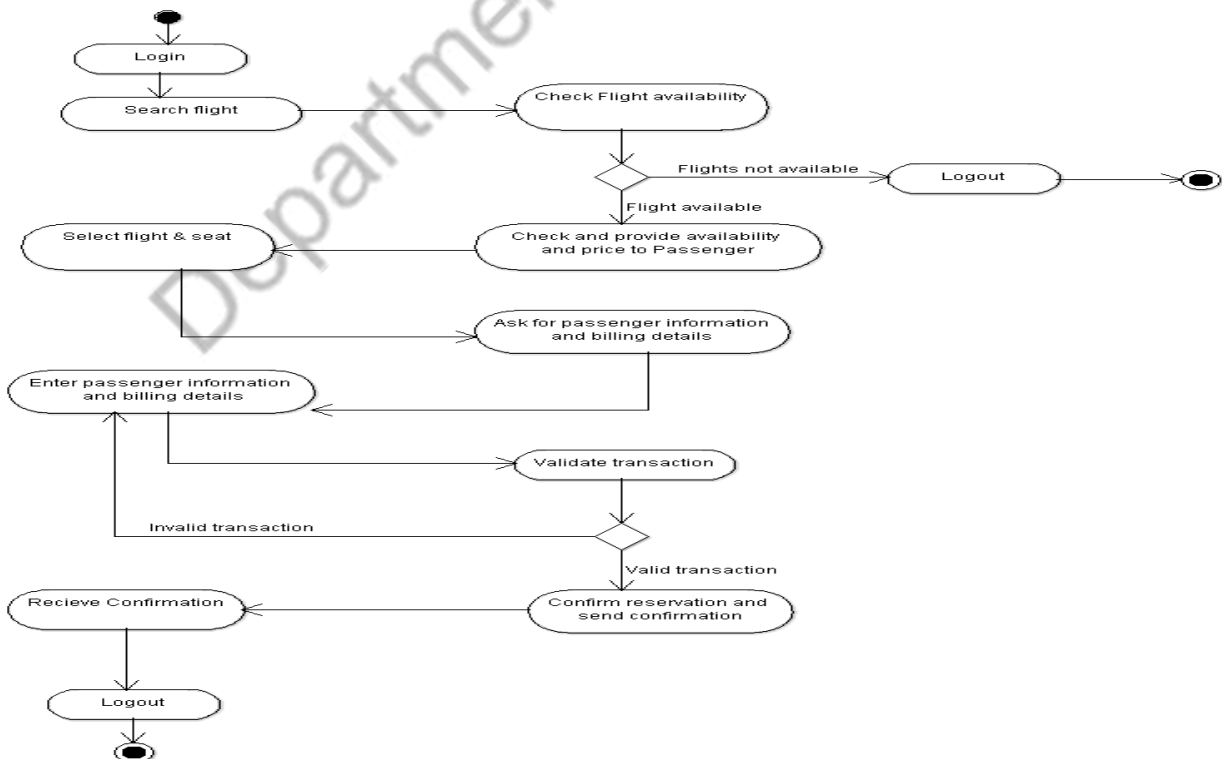
COLLABORATION DIAGRAM



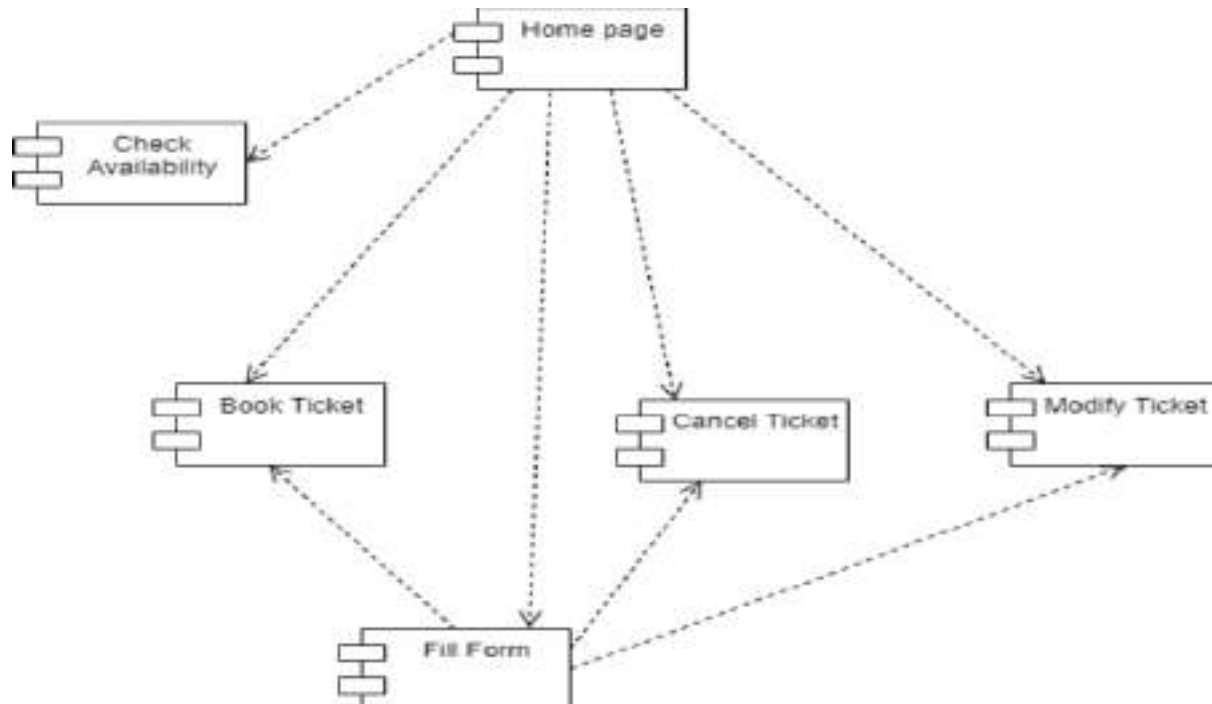
STATE MACHINE DIAGRAM



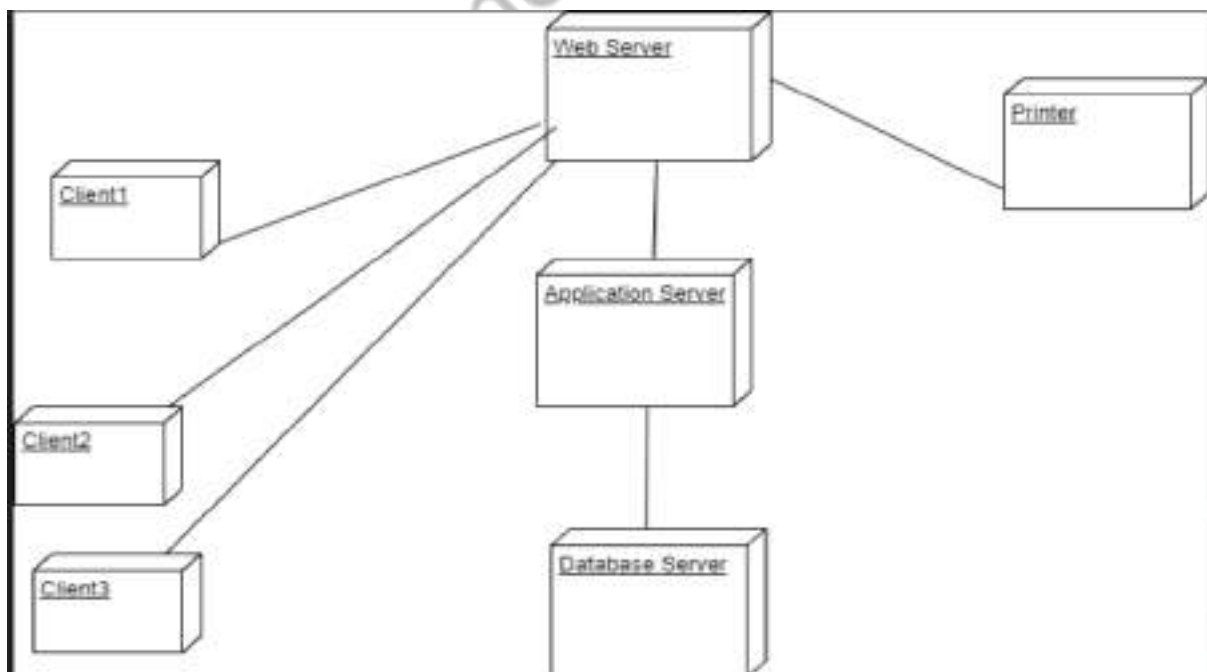
ACTIVITY DIAGRAM



COMPONENT DIAGRAM

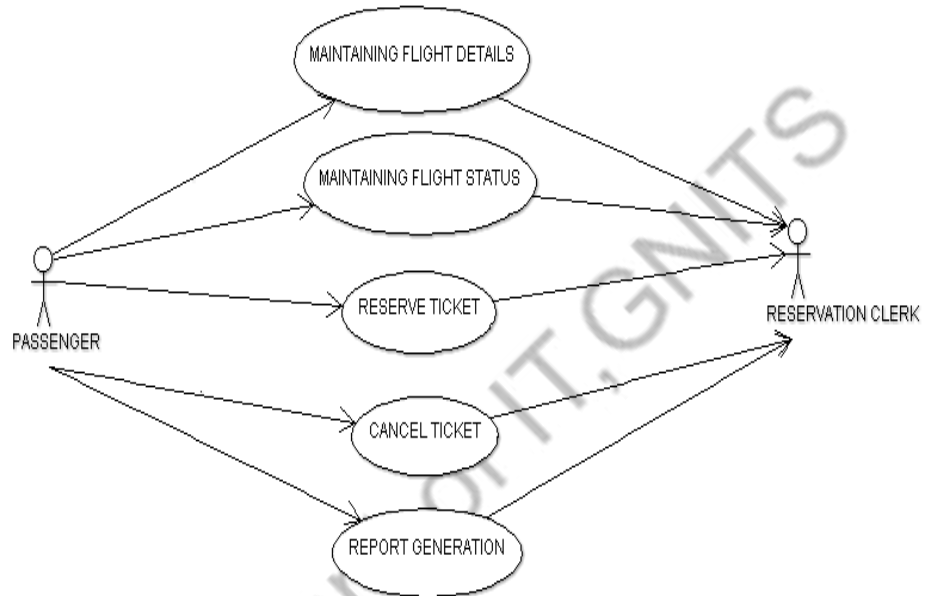


DEPLOYMENT DIAGRAM

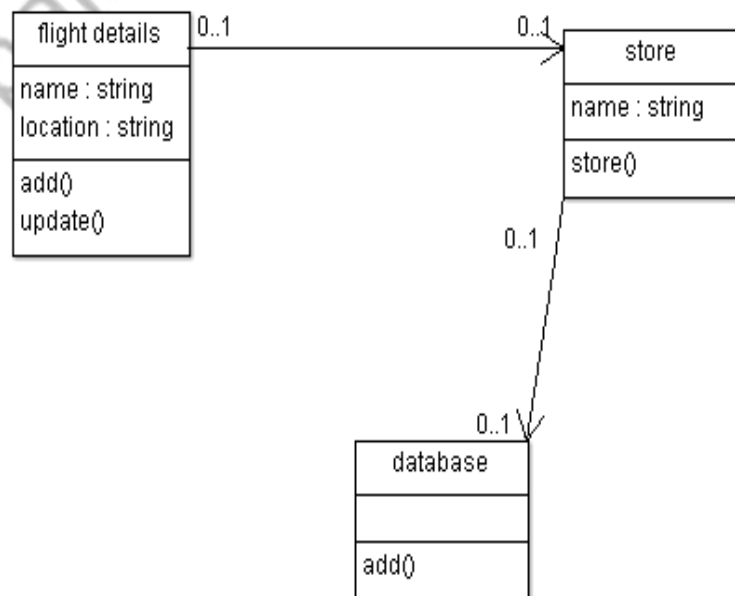


FLIGHT INFORMATION SYSTEM

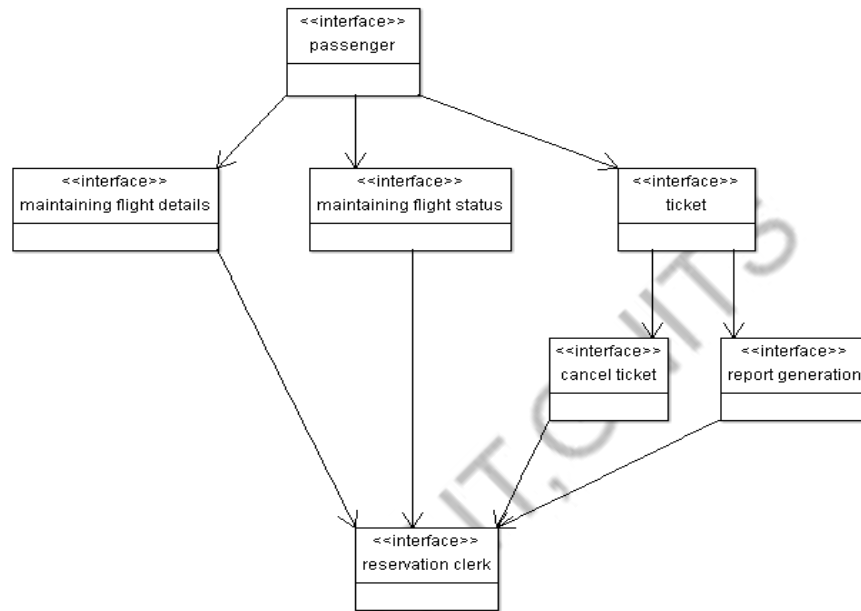
USE CASE DIAGRAM:



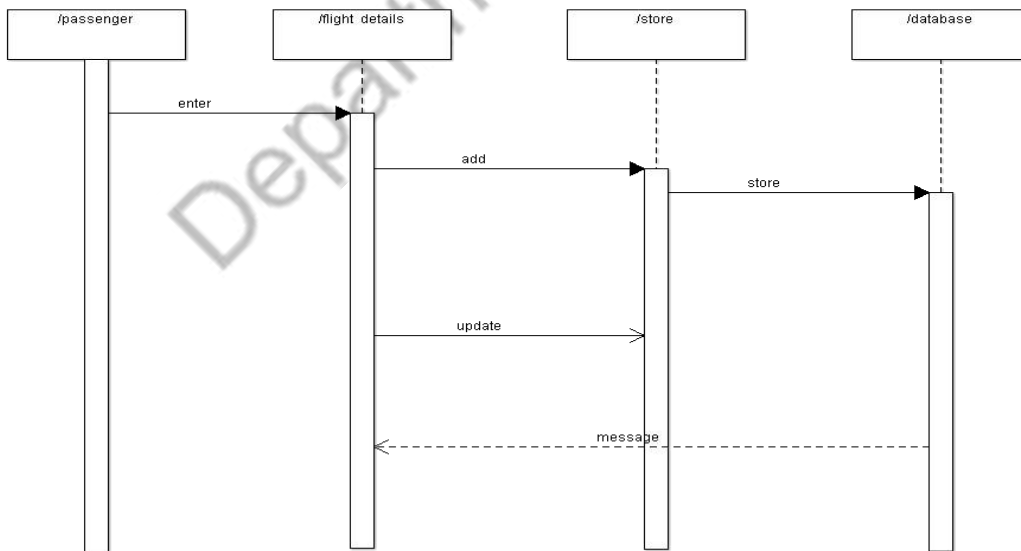
CLASS DIAGRAM:



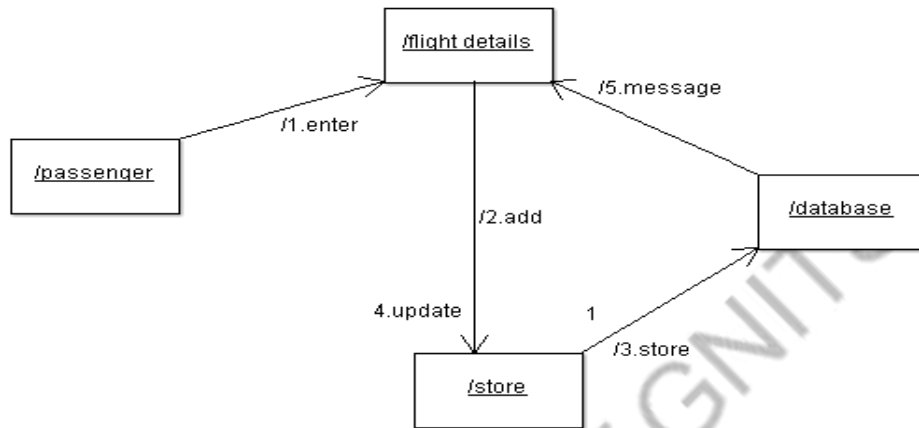
OBJECT DIAGRAM



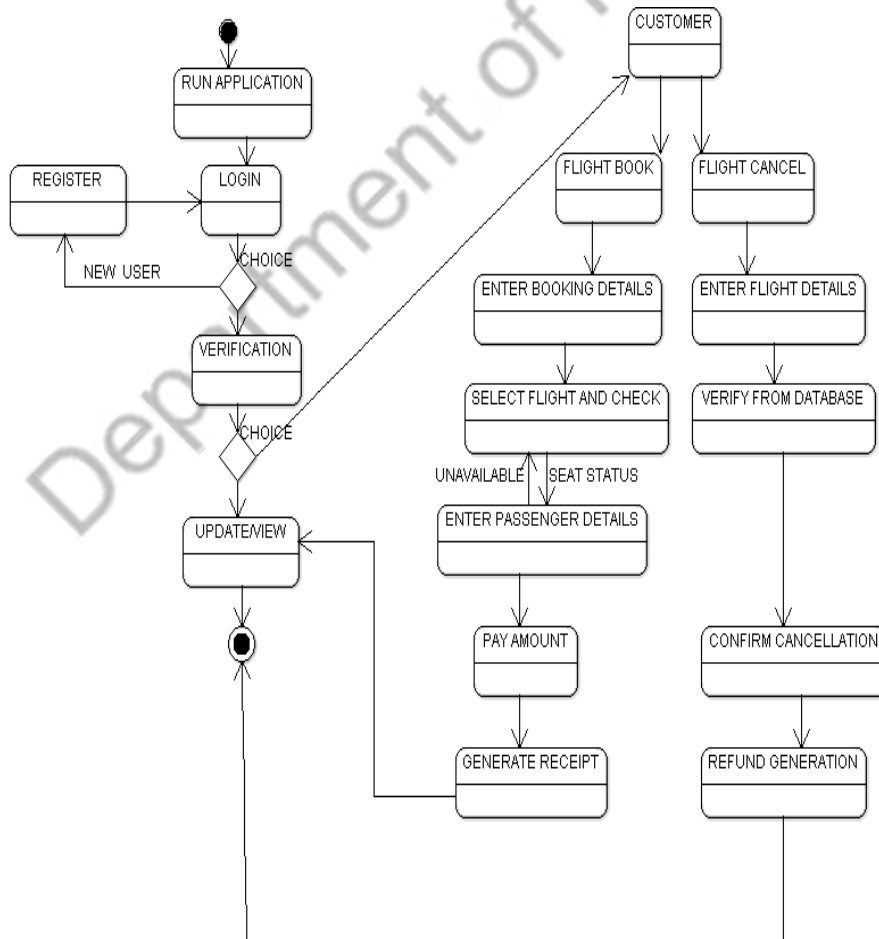
SEQUENCE DIAGRAM:



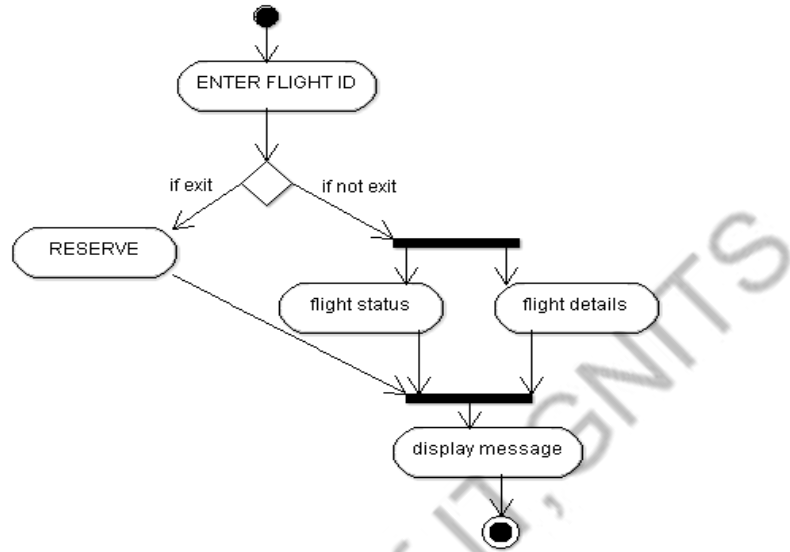
COLLABORATION DIAGRAM:



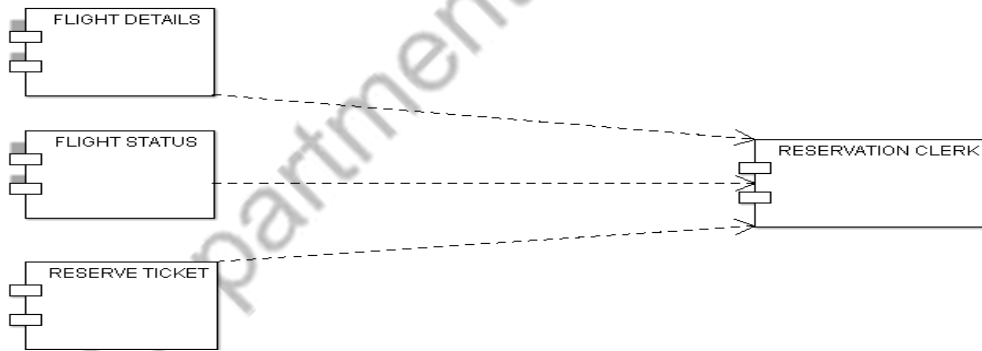
STATE DIAGRAM:



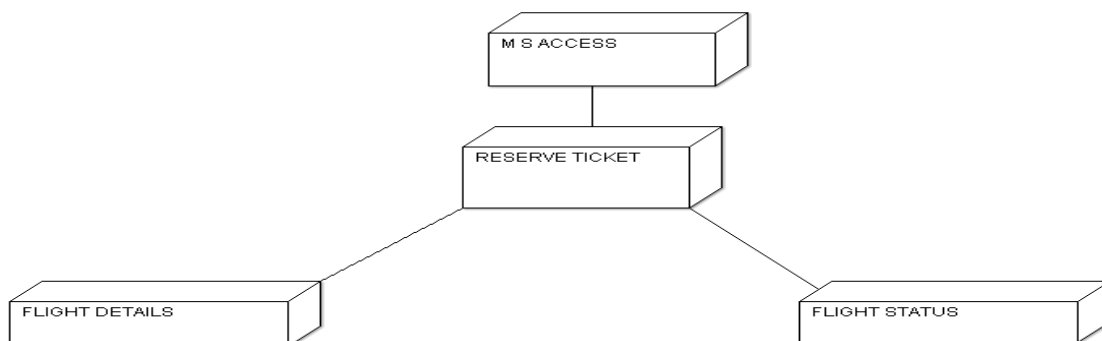
ACTIVITY DIAGRAM:



COMPONENT DIAGRAM:

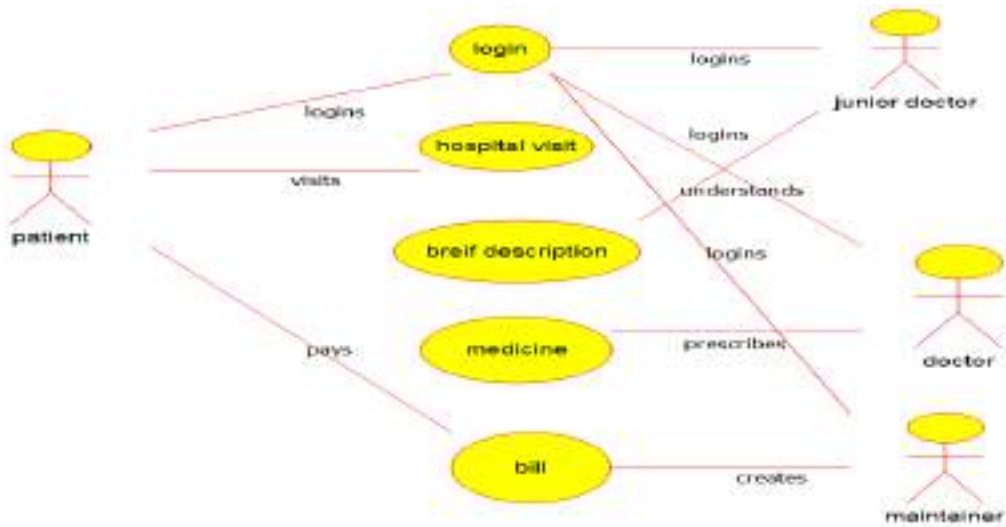


DEPLOYMENT DIAGRAM:

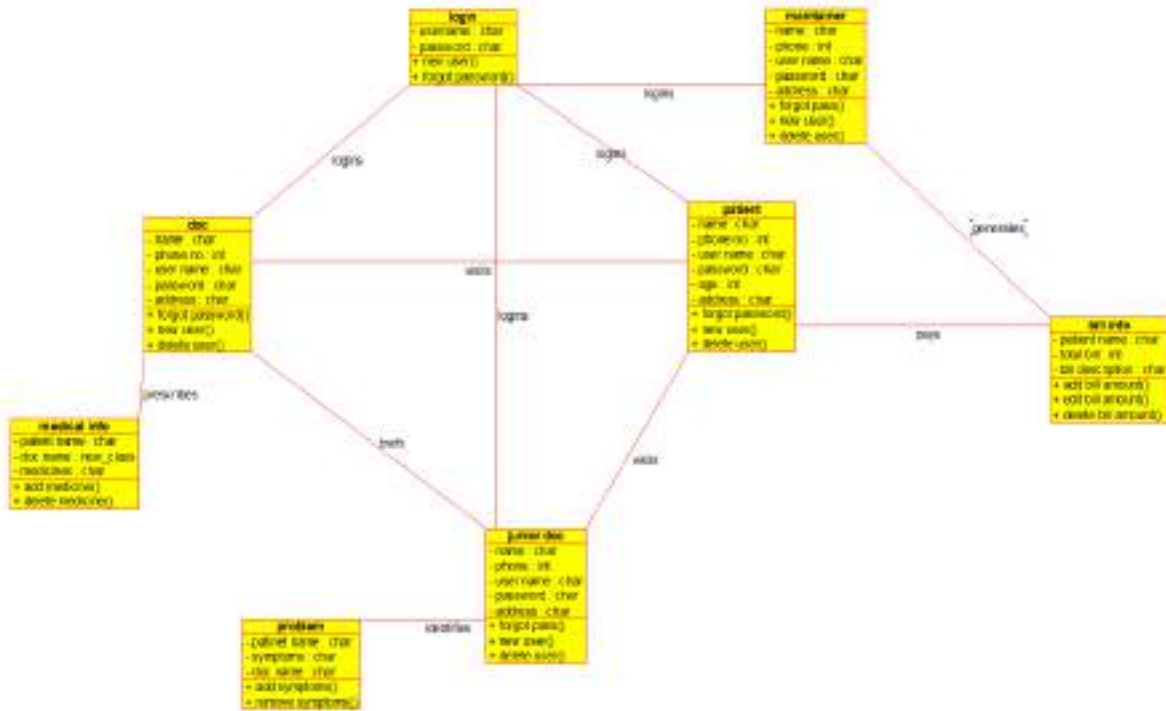


EXPERT SYSTEM

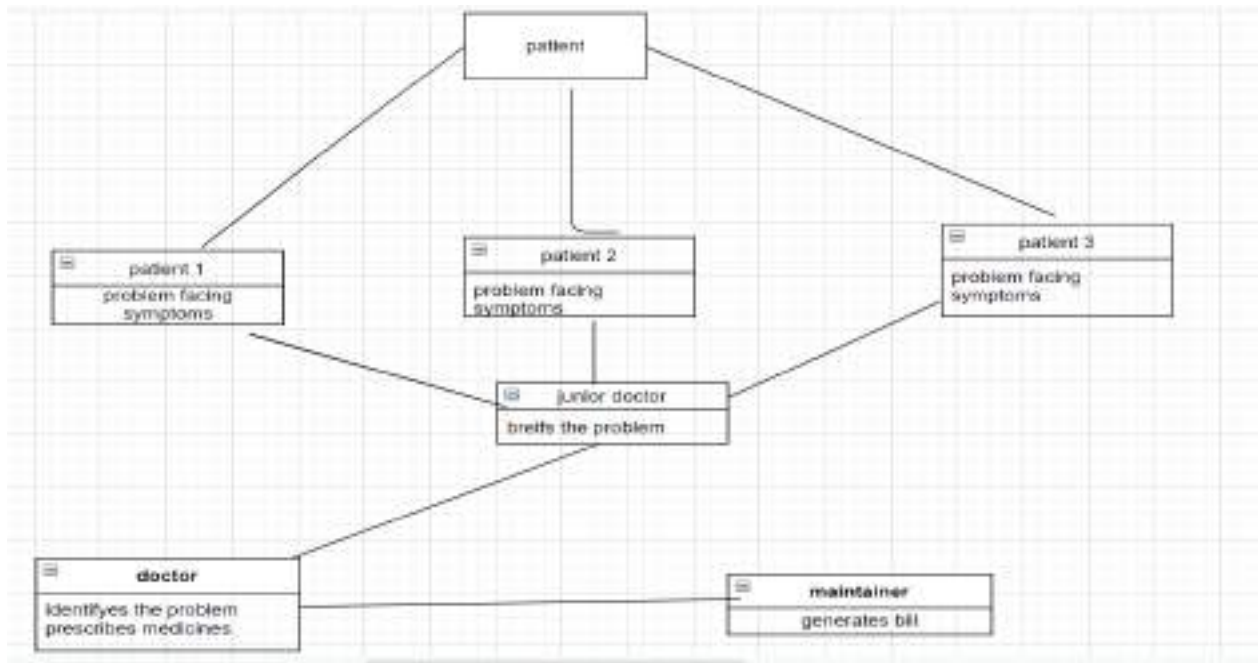
Use case diagram -



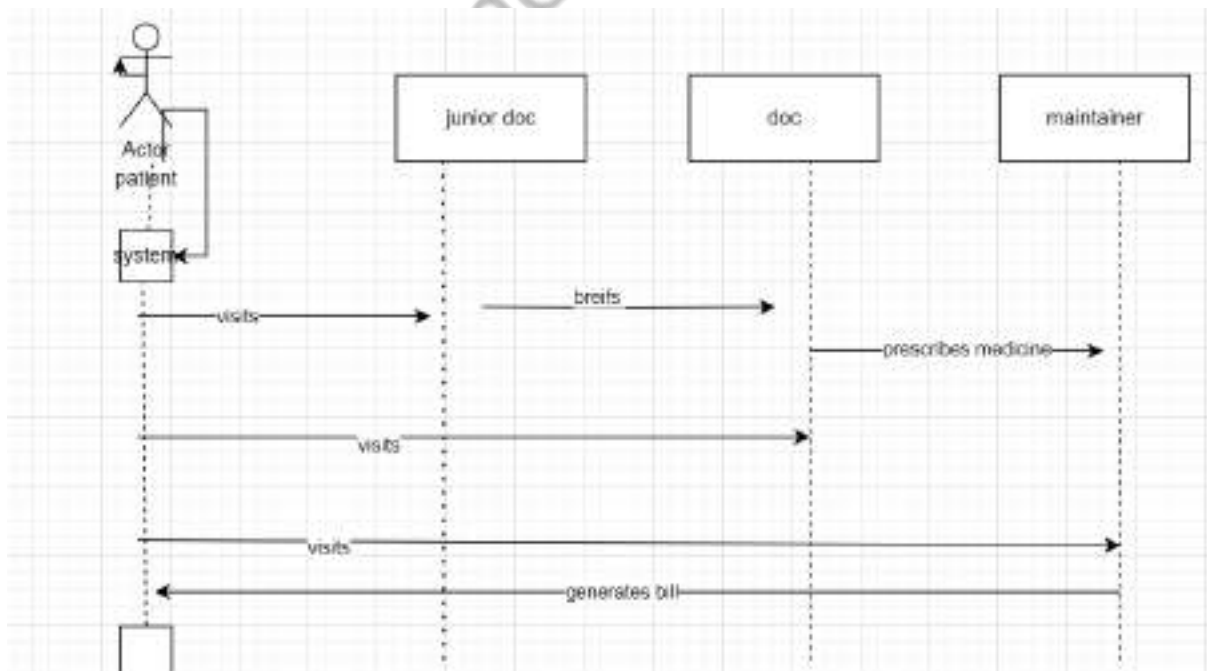
Class diagram –



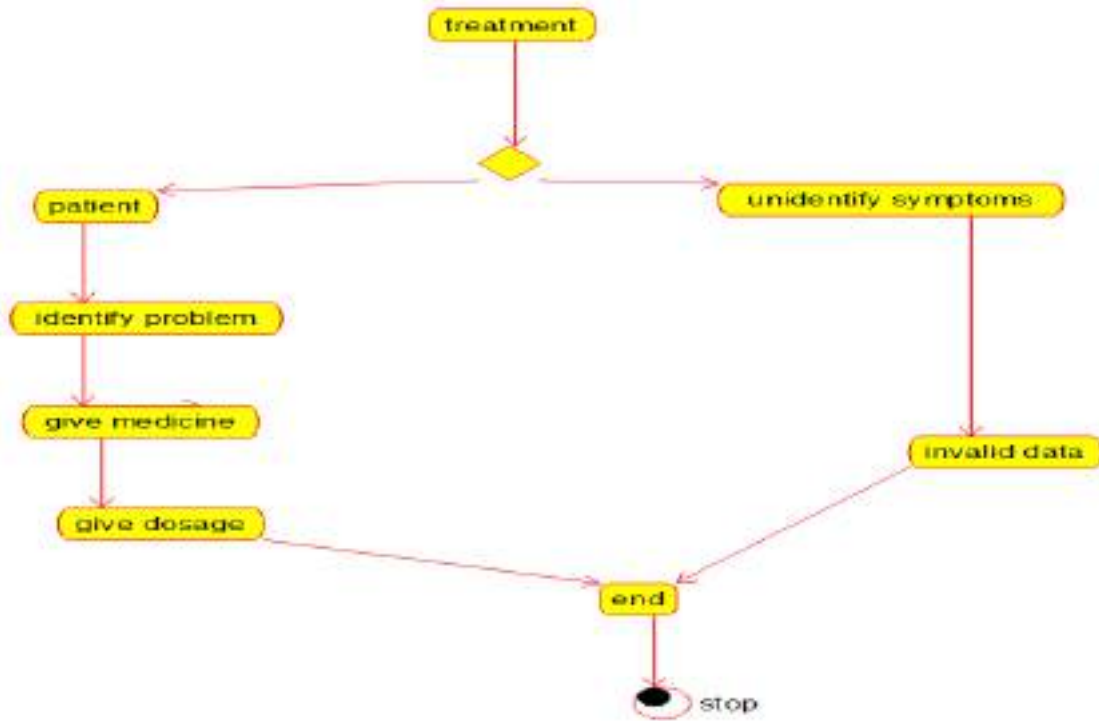
Object Diagram –



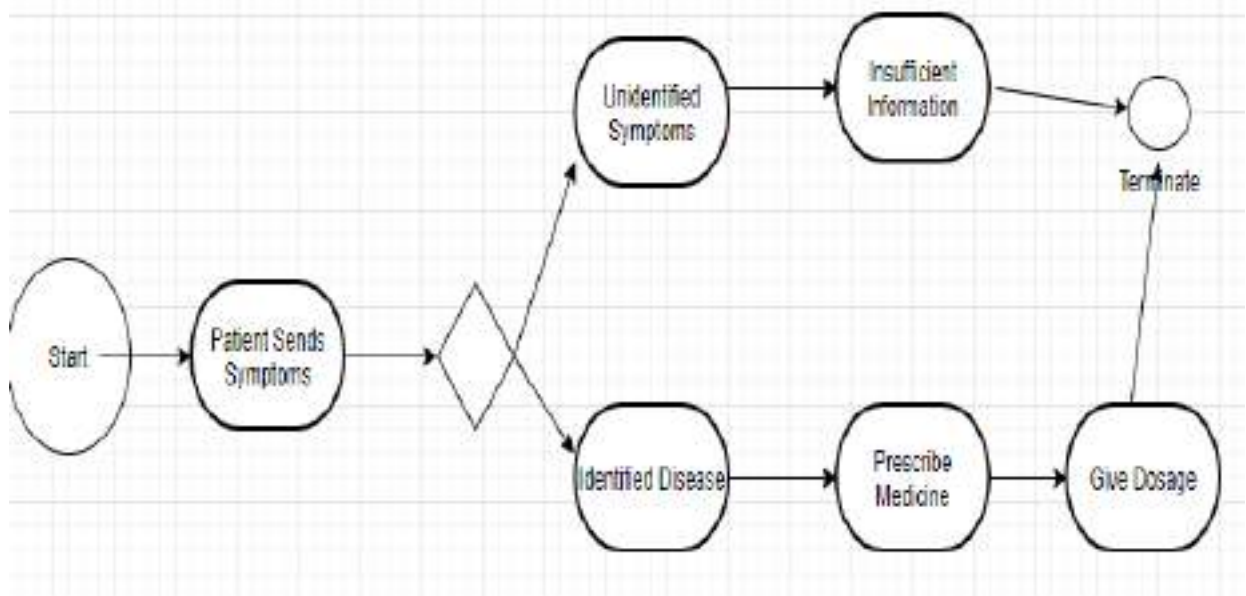
Sequence Diagram –



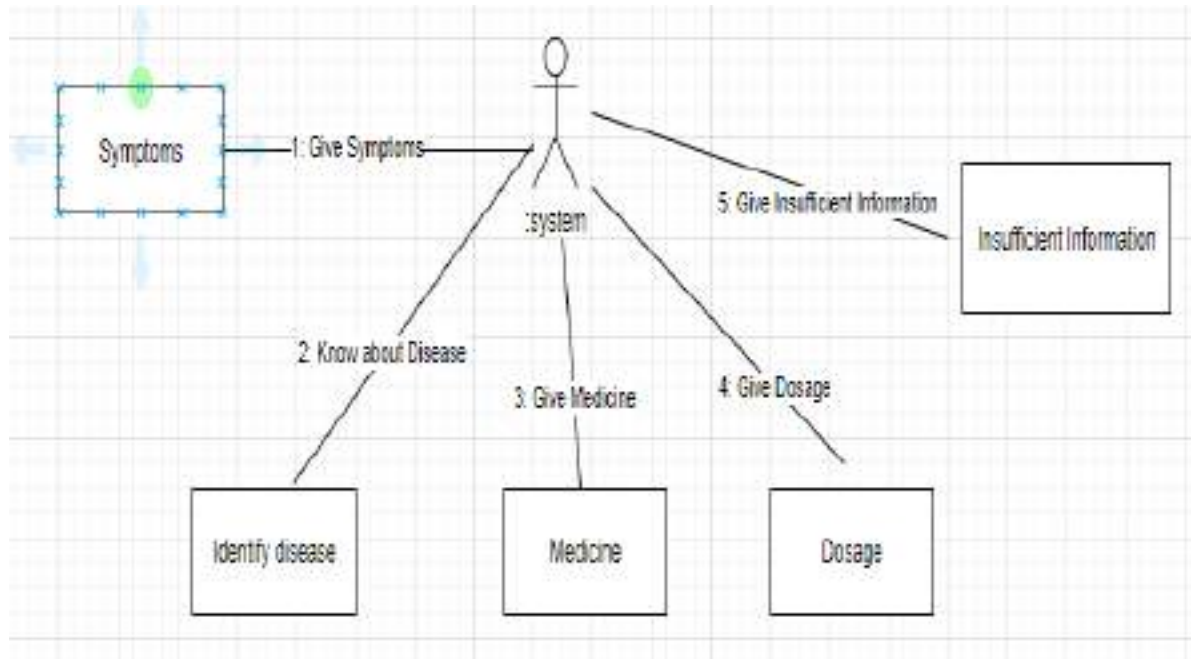
State Diagram -



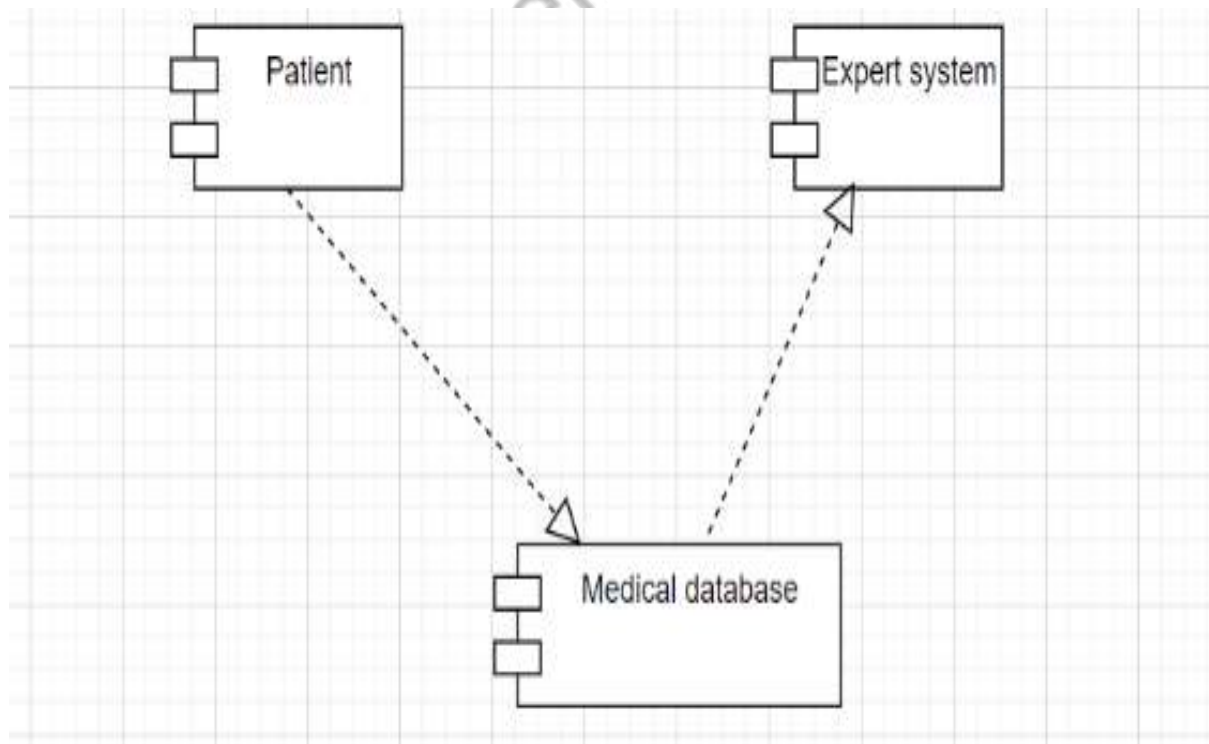
Activity Diagram -



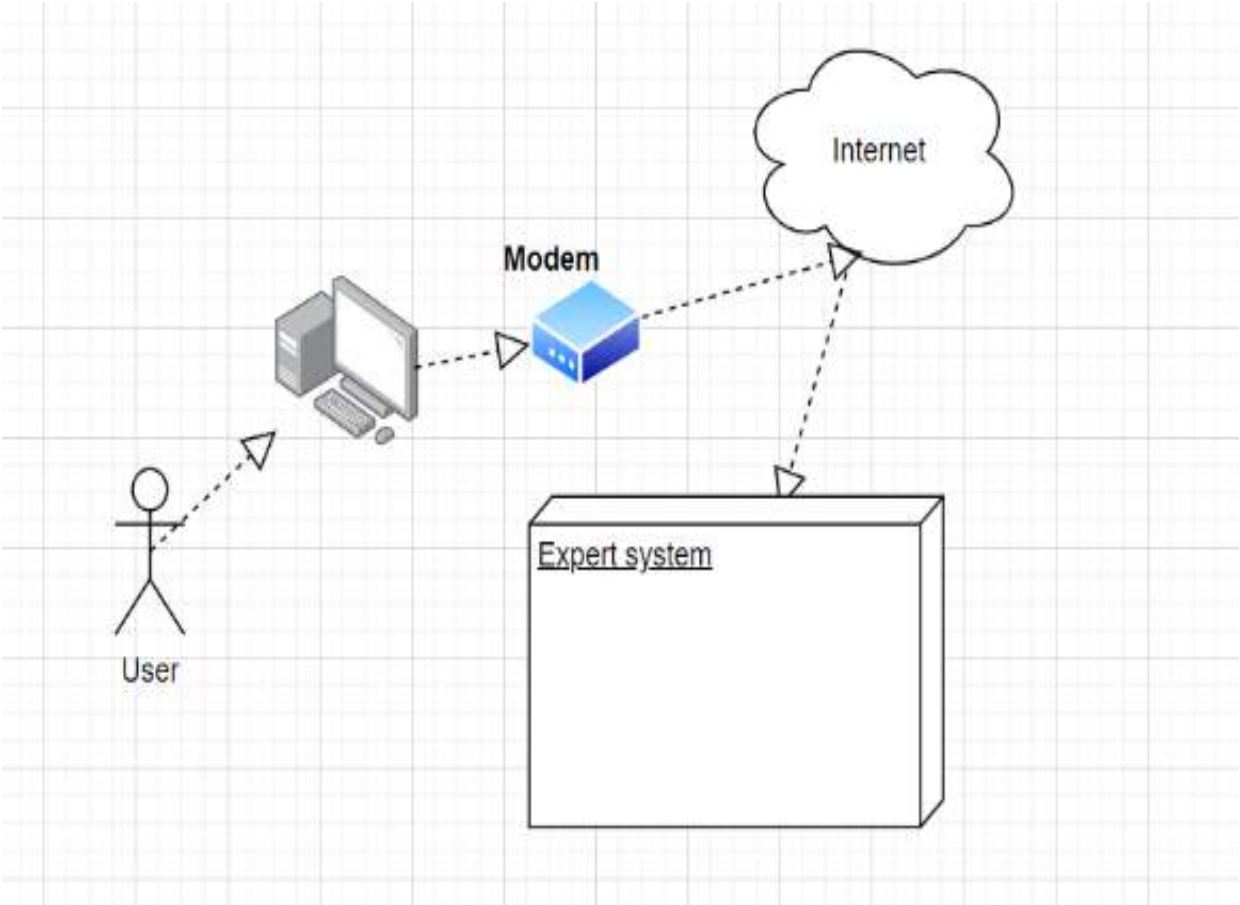
Collaboration Diagram -



Component Diagram -



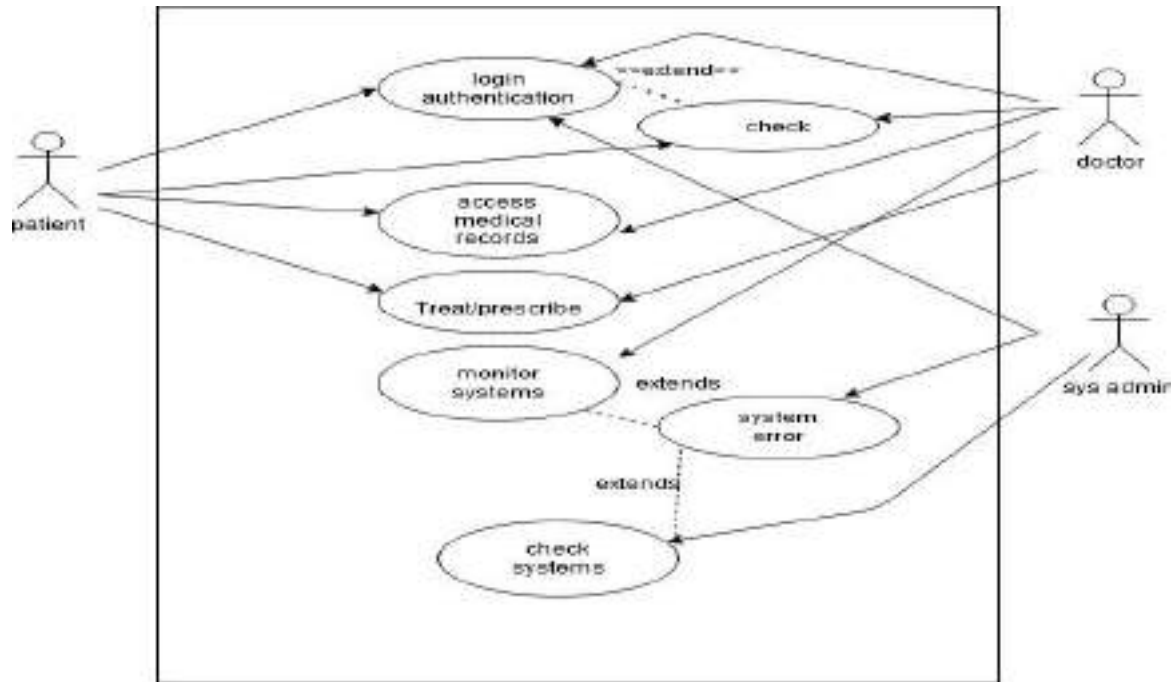
Deployment Diagram –



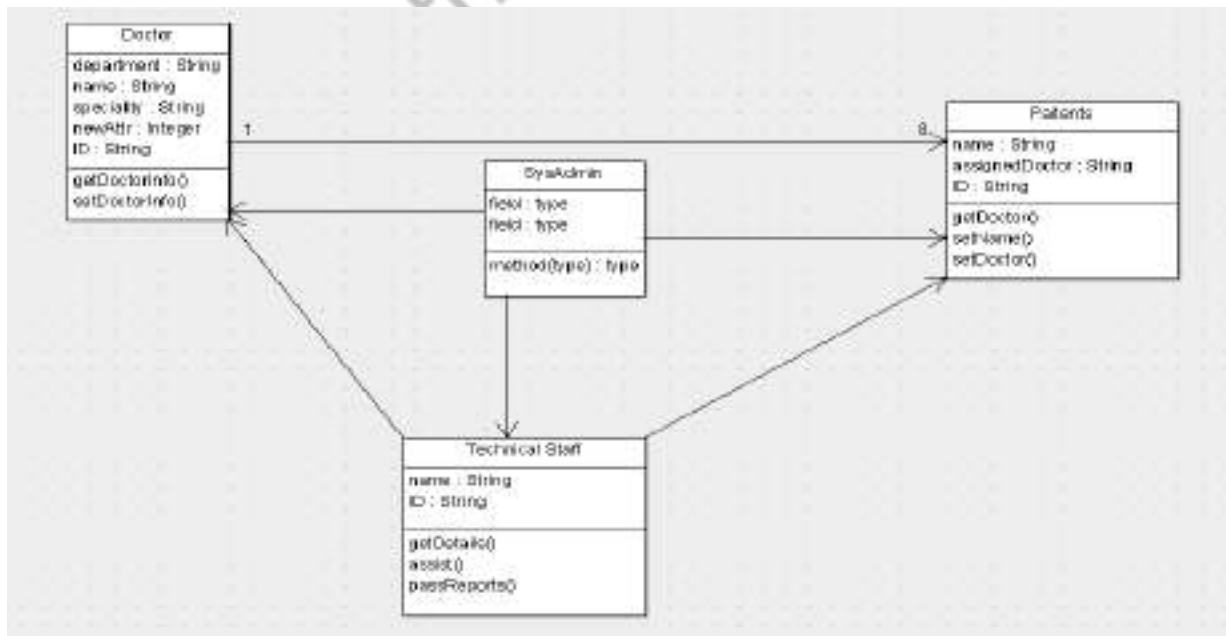
Dep

REMOTE COMPUTER MONITORING

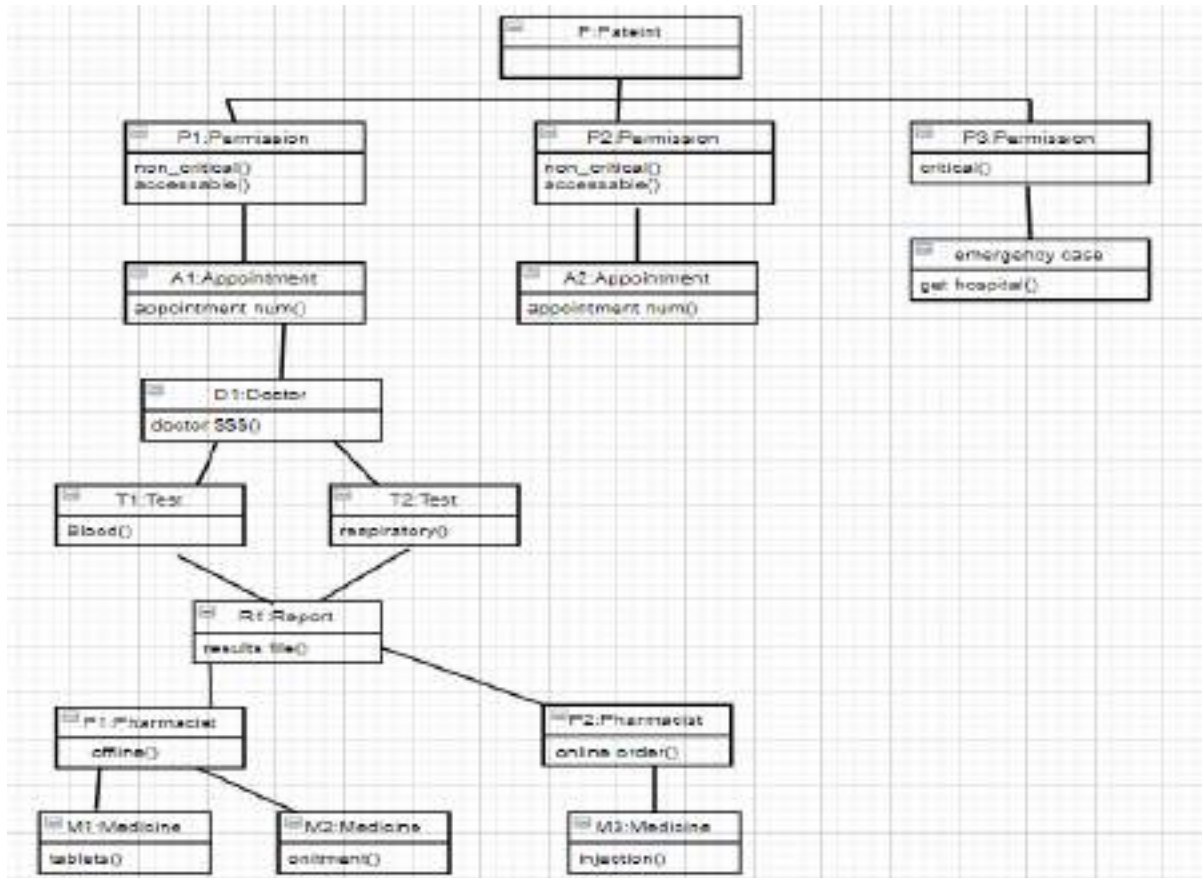
USE CASE DIAGRAM



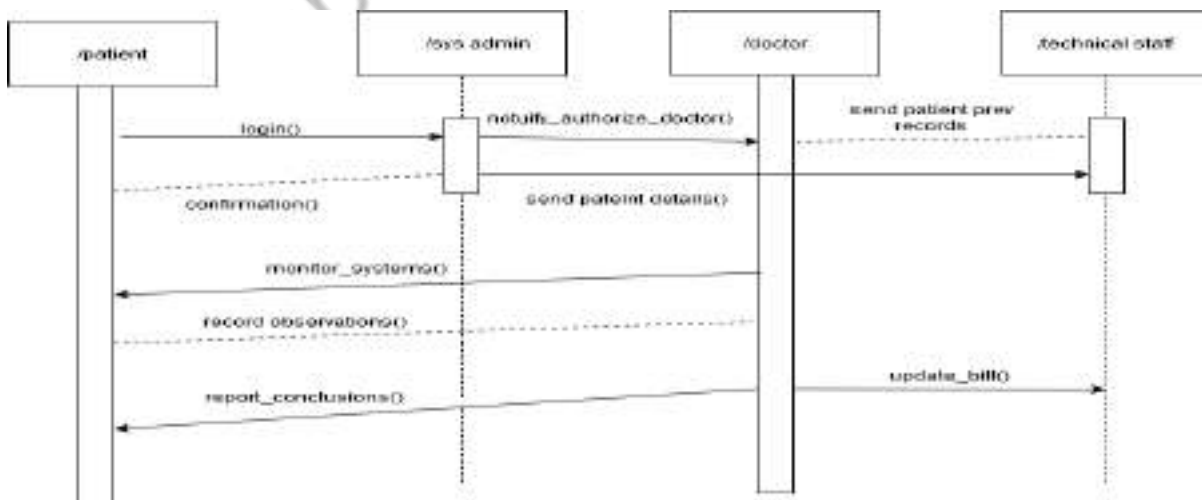
CLASS DIAGRAM



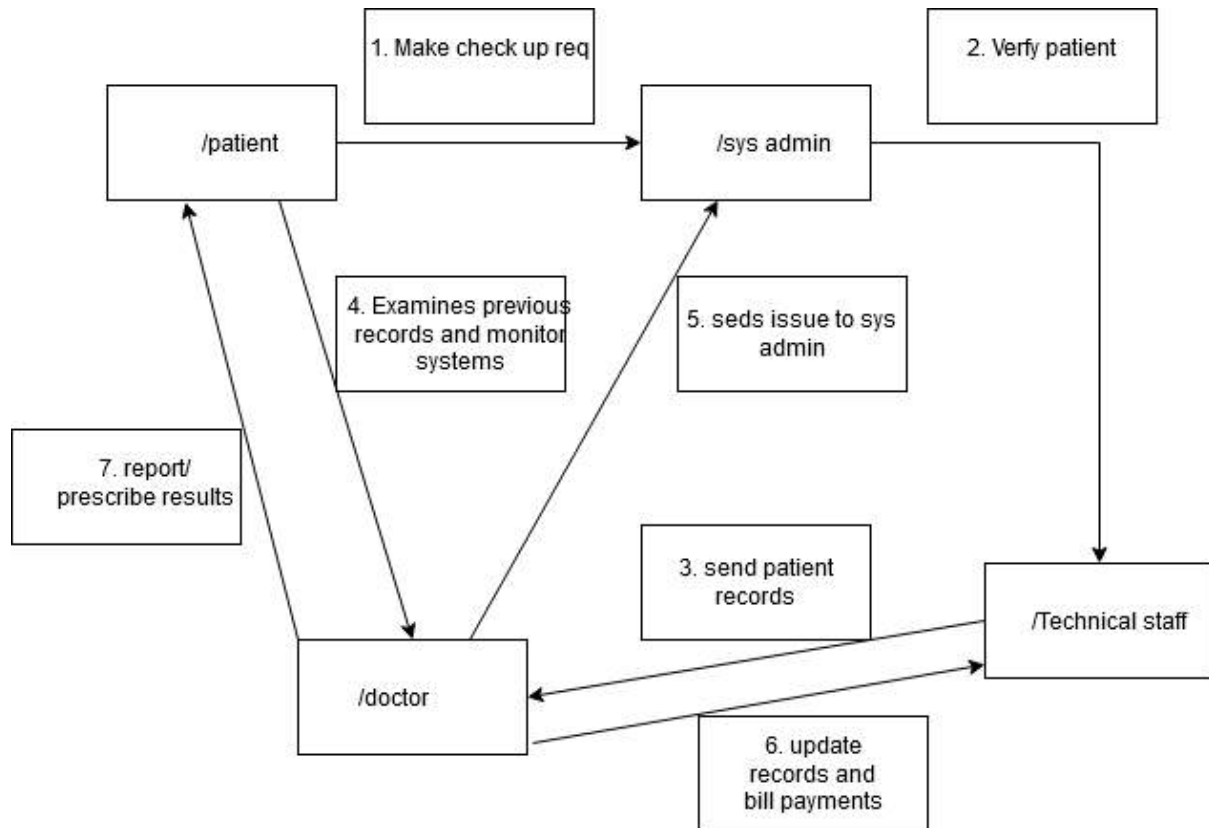
OBJECT DIAGRAM



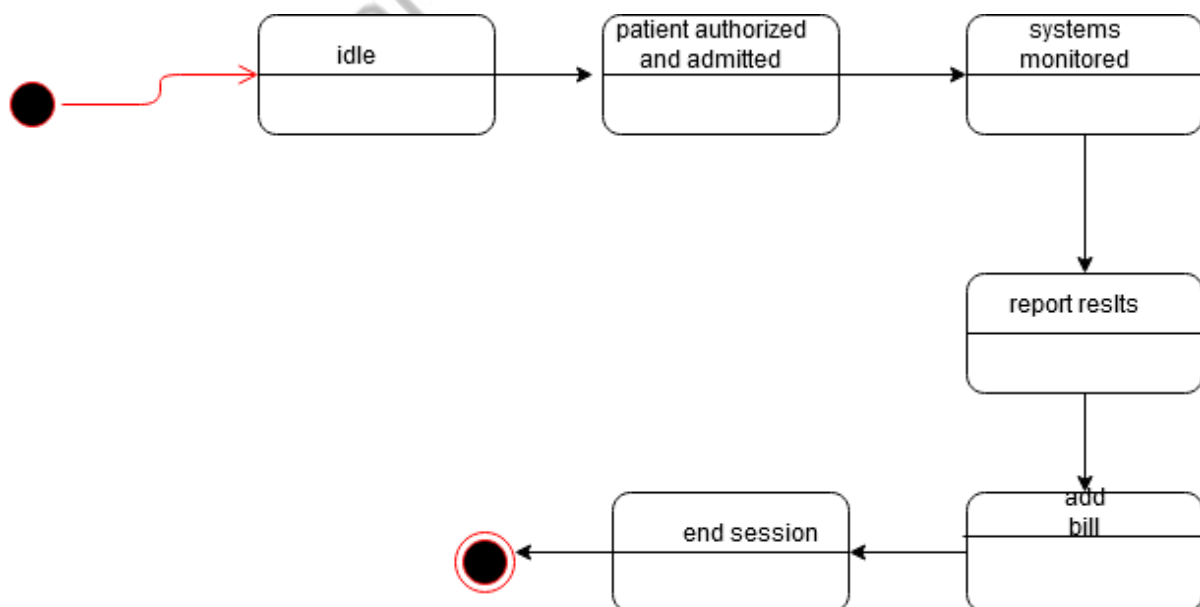
SEQUENCE DIAGRAM



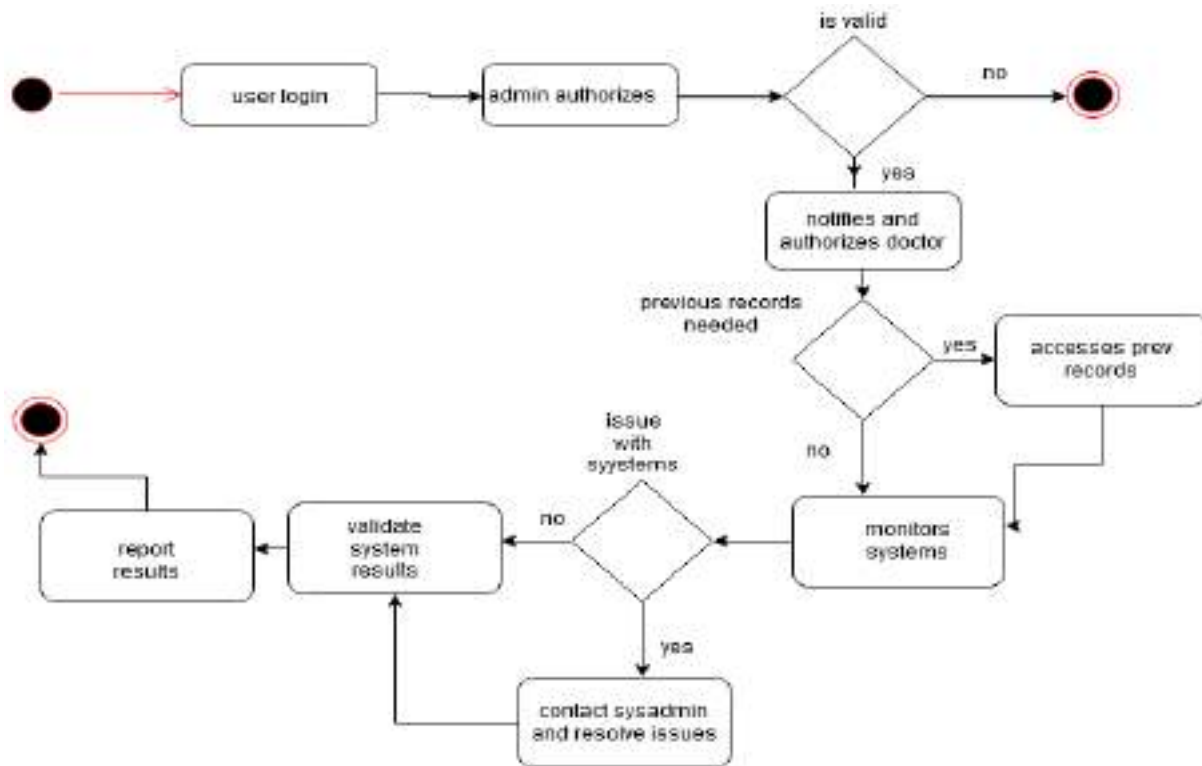
COLLABARATION DIAGRAM



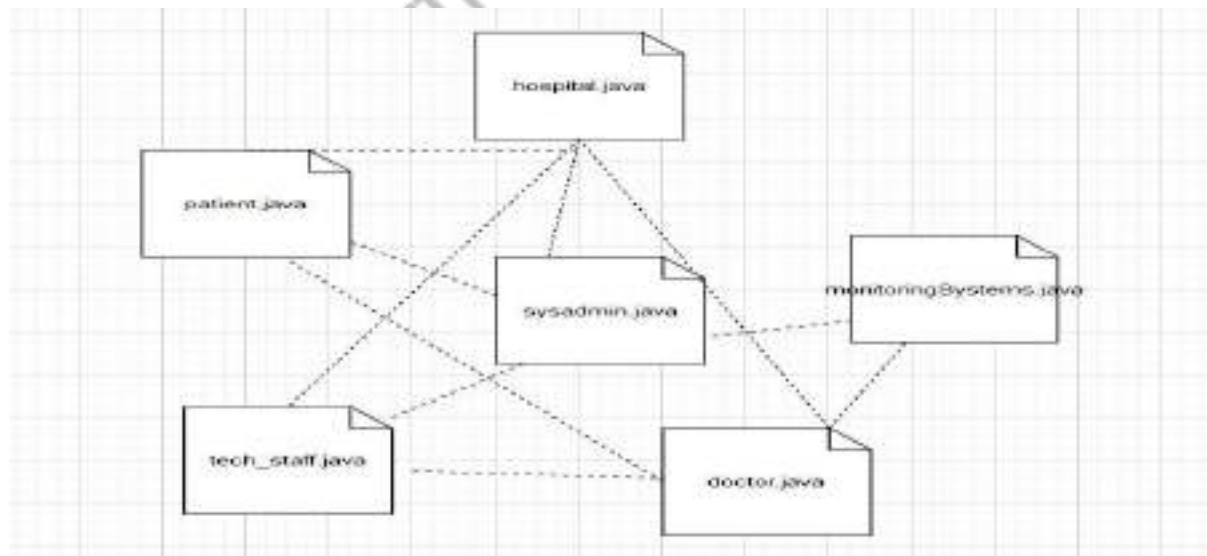
STATE CHART DIAGRAM-



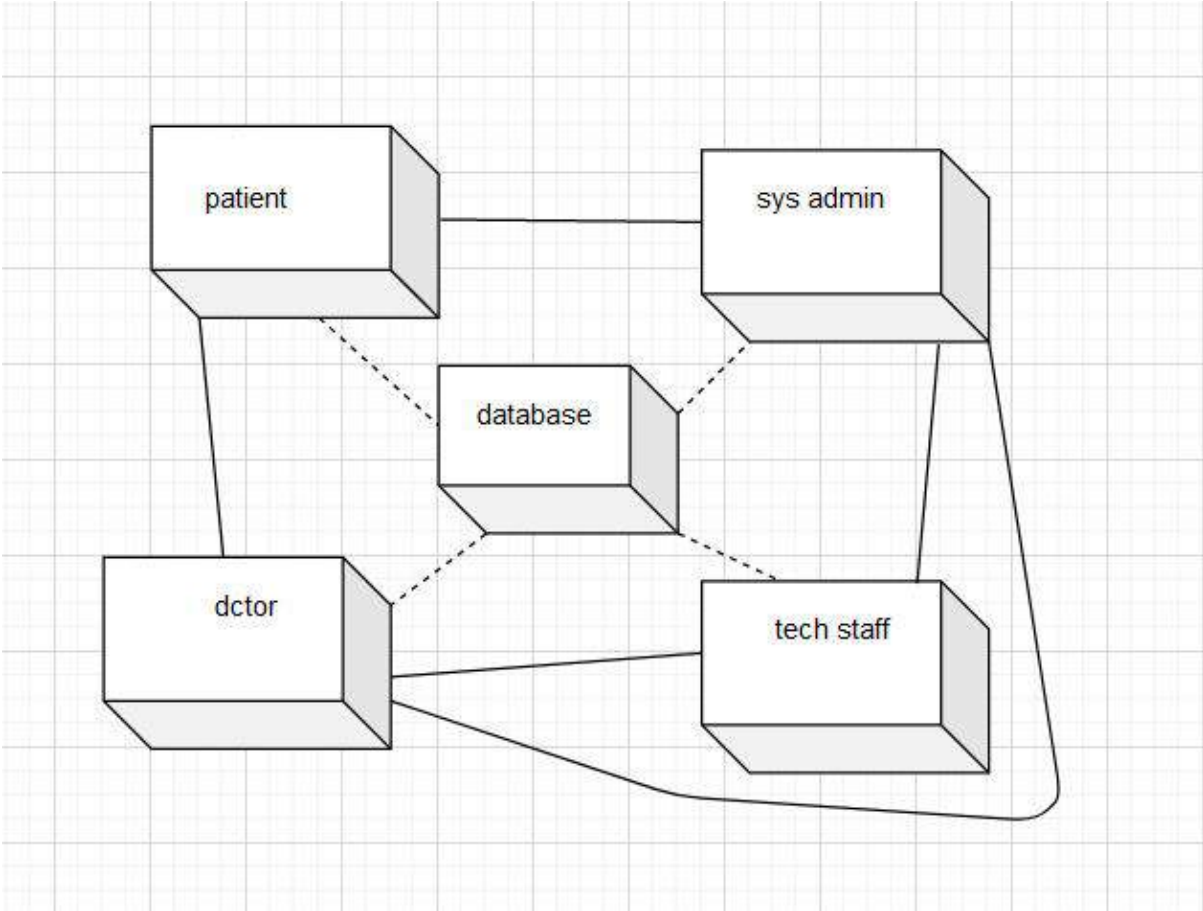
ACTIVITY DIAGRAM



COMPONENT DIAGRAM-



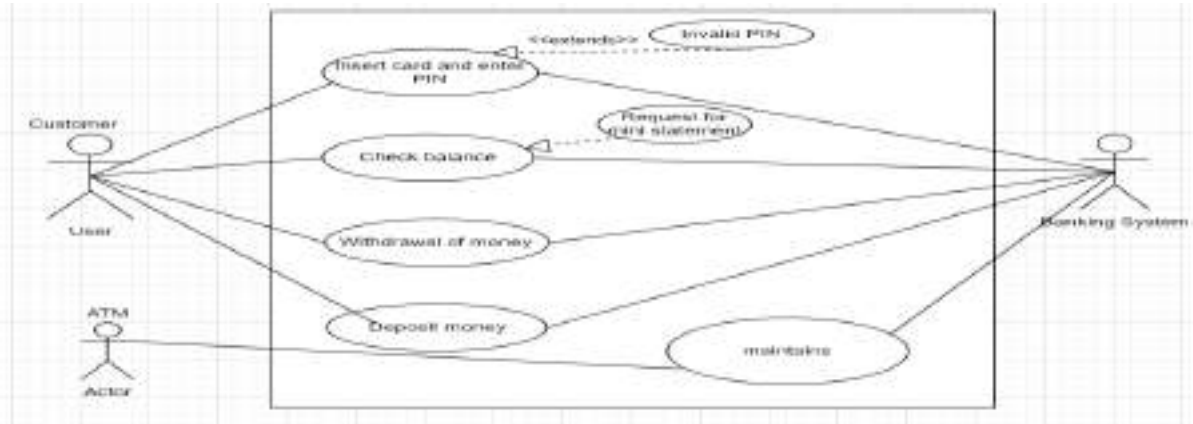
DEPLOYMENT DIAGRAM



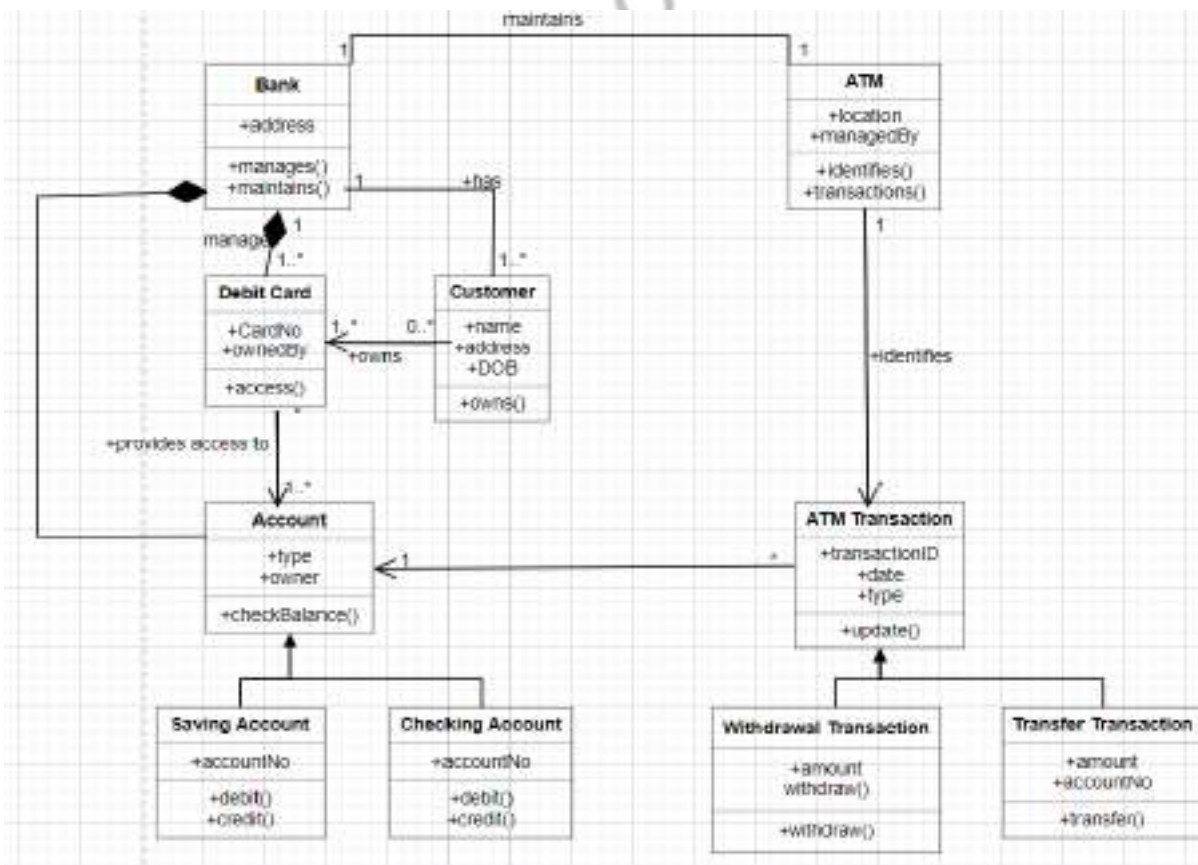
Depa

ATM SYSTEM

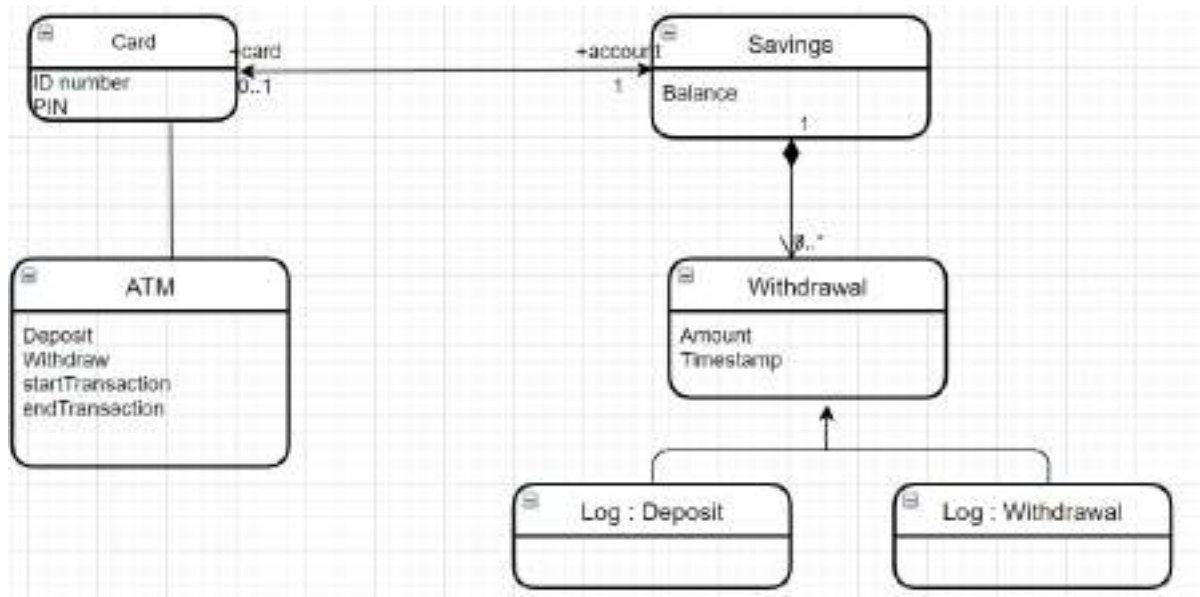
Use Case Diagram -



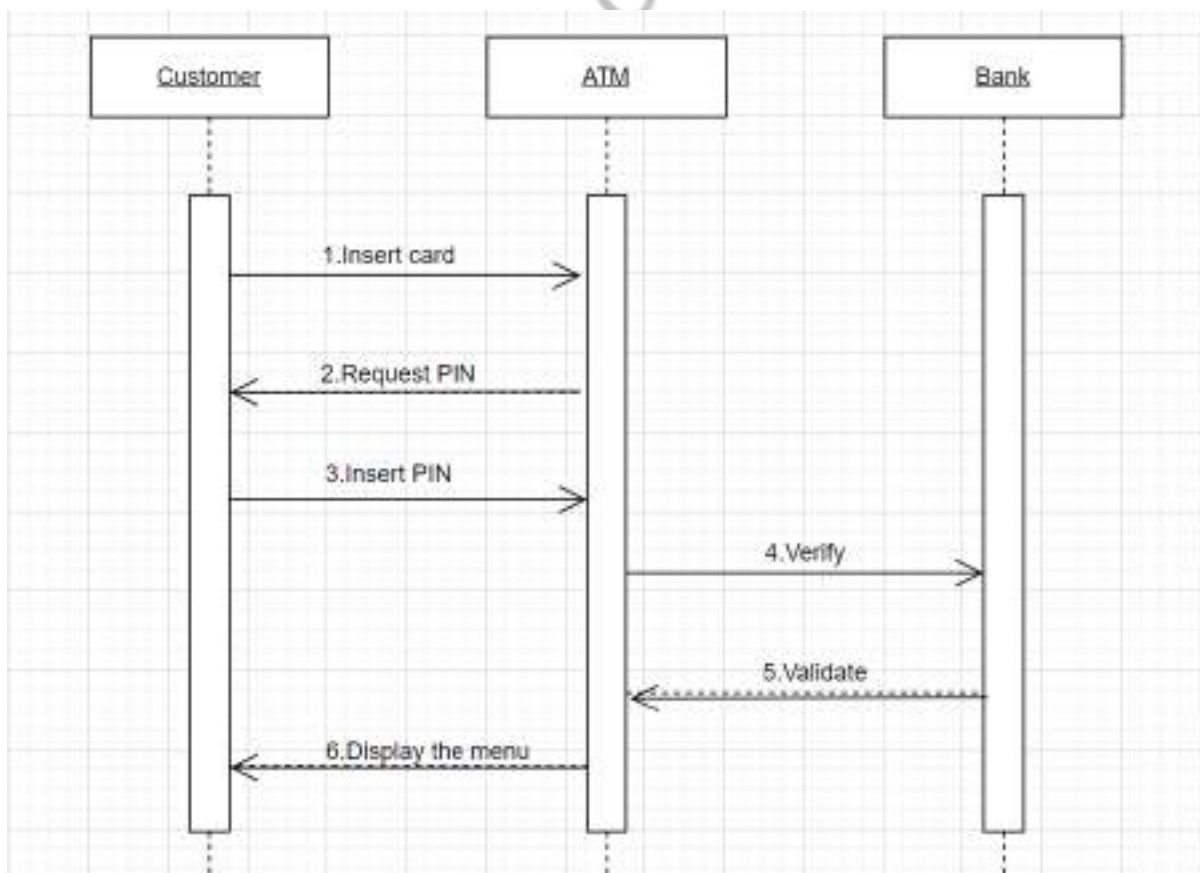
Class Diagram-



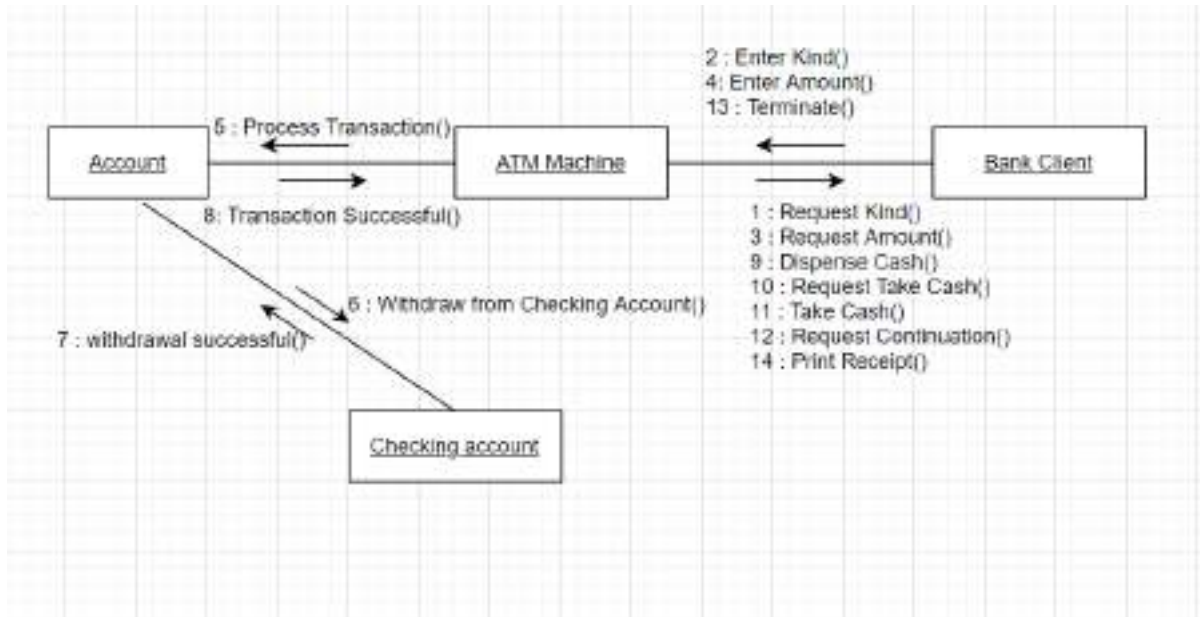
Object Diagram-



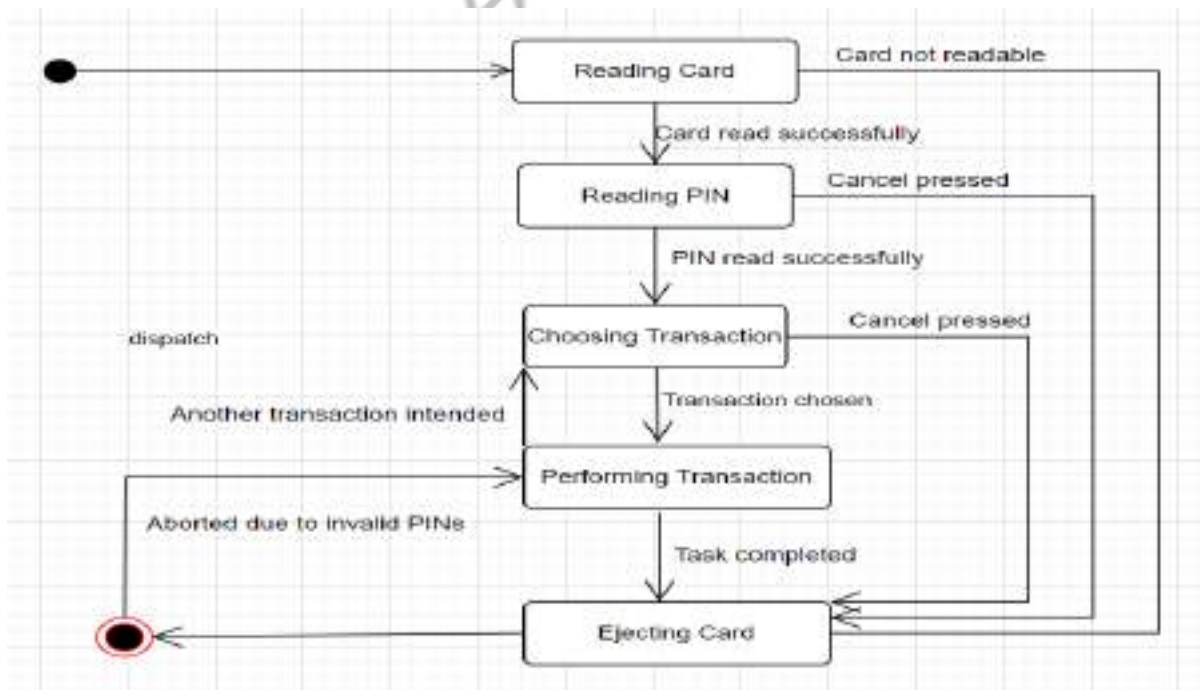
Sequence Diagram –



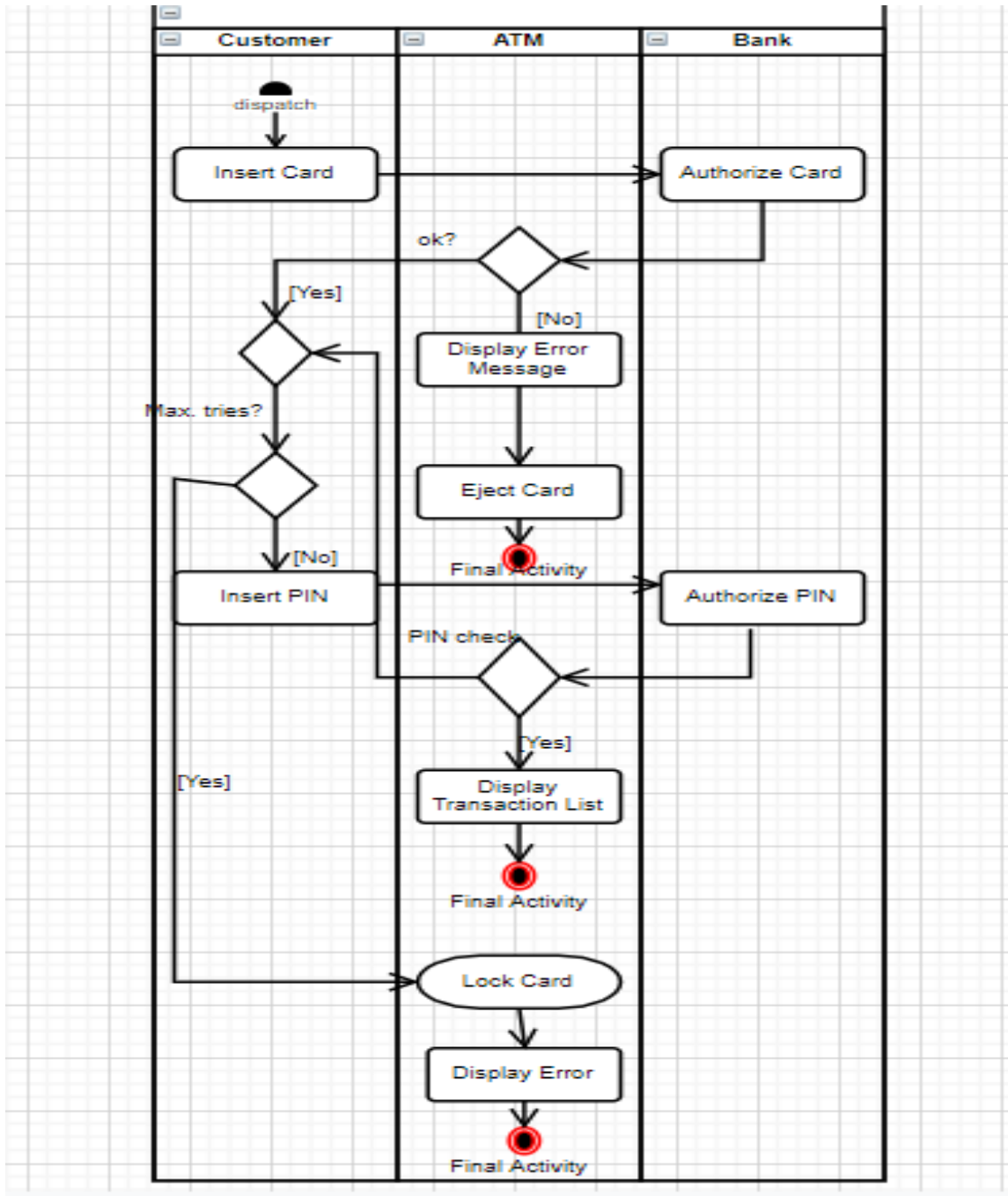
Collaboration diagram



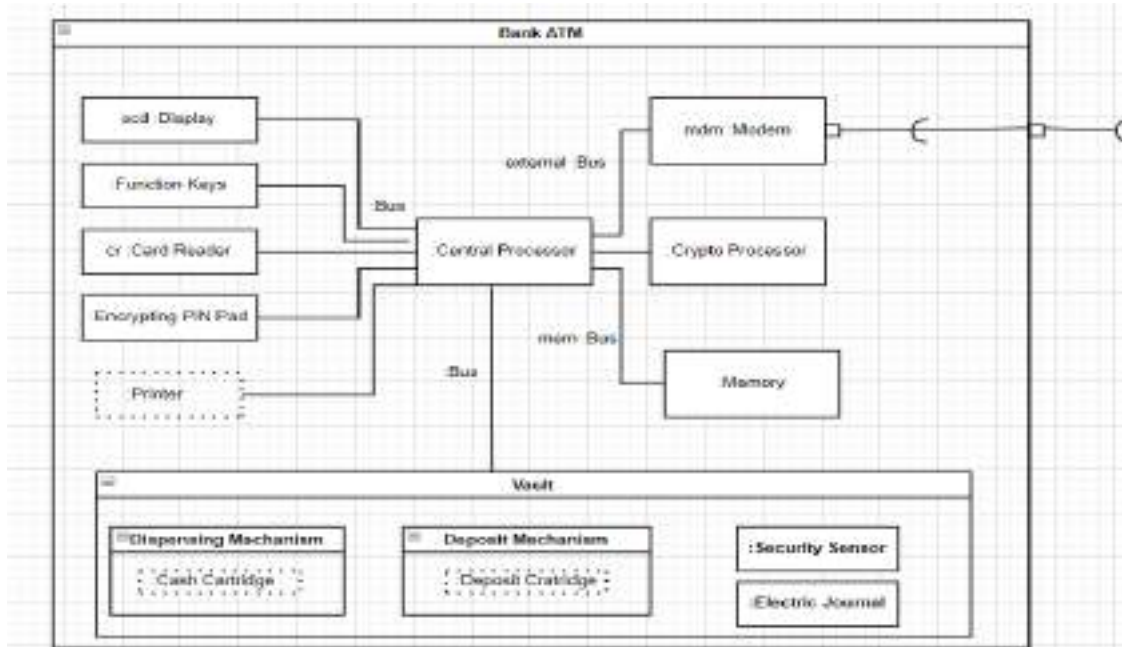
State diagram-



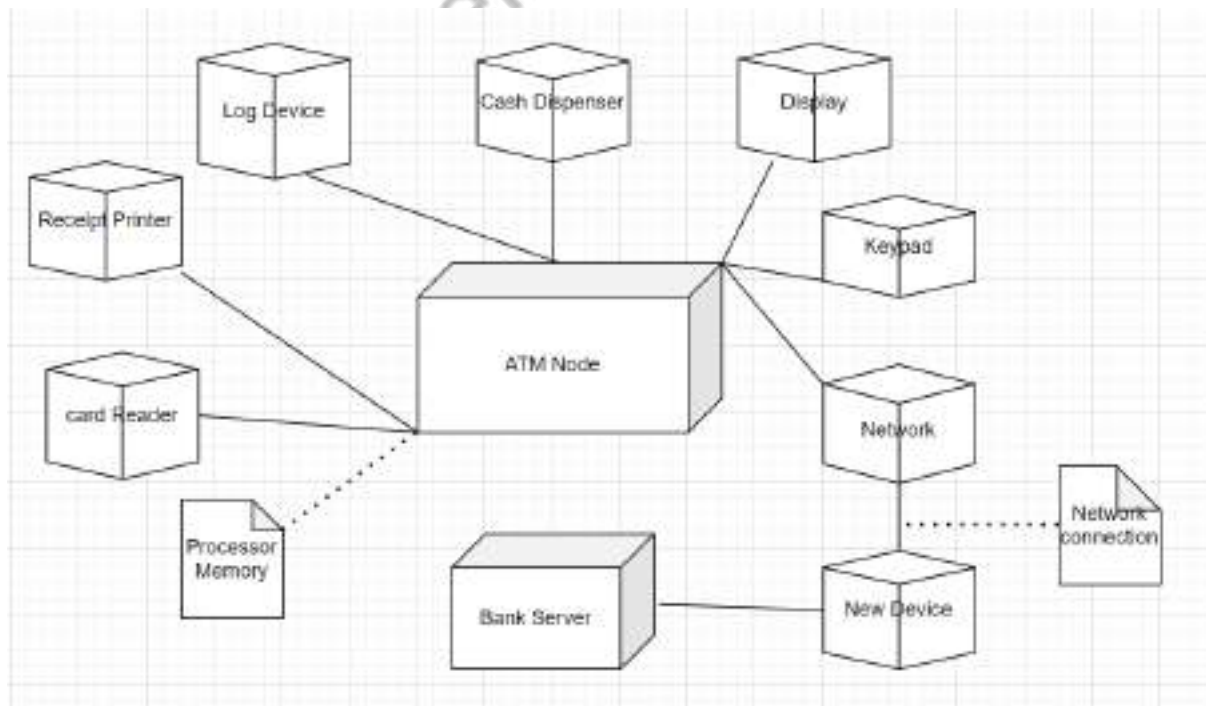
Activity diagram



Component diagram

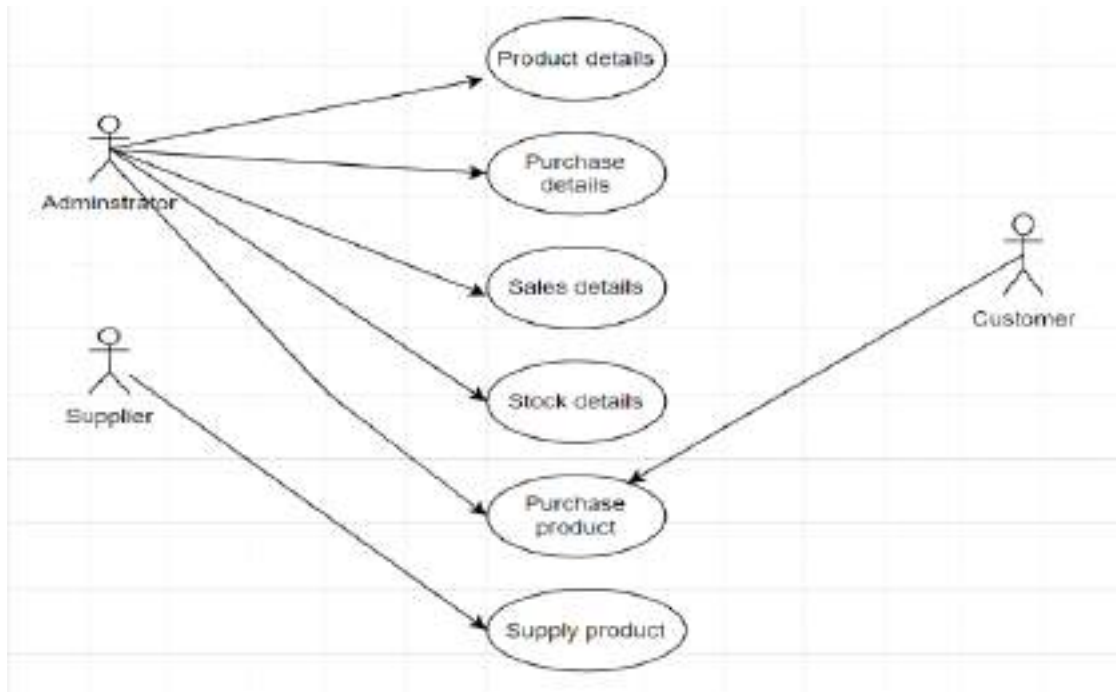


Deployment diagram

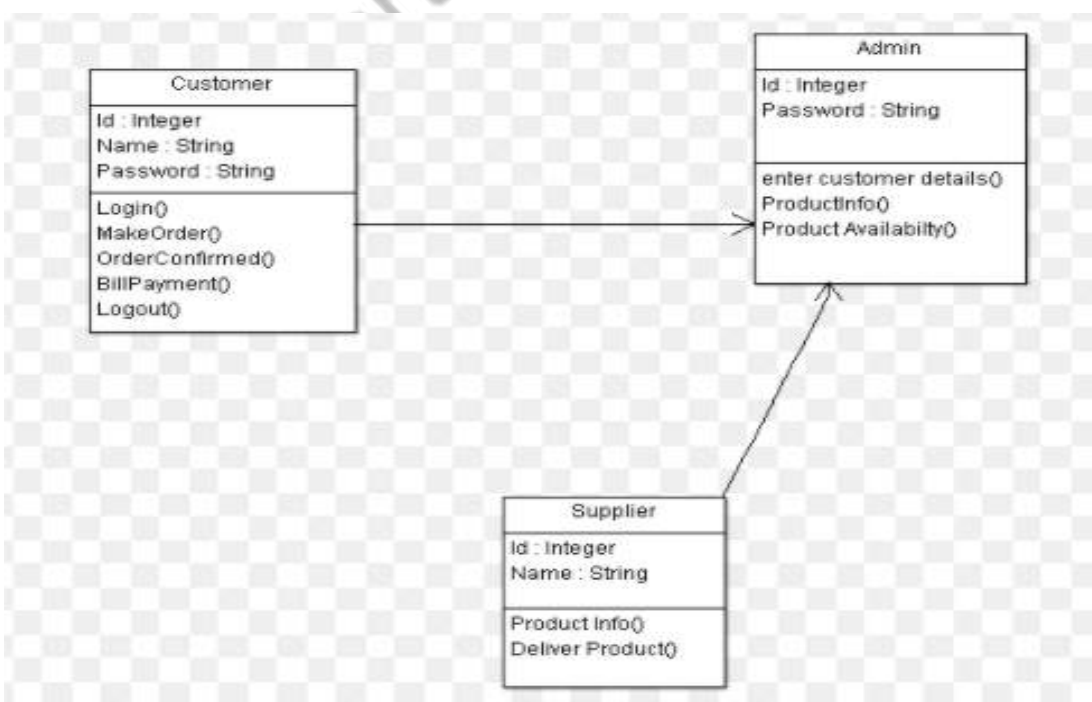


STOCK MANAGEMENT SYSTEM

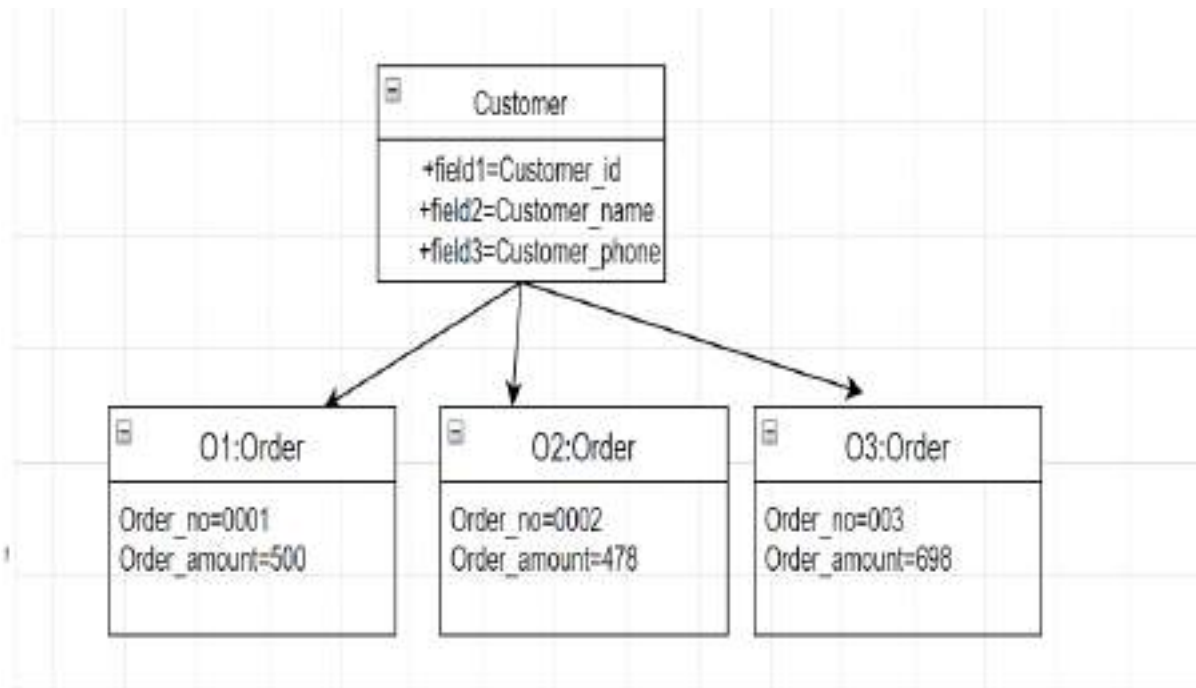
Use Case Diagram-



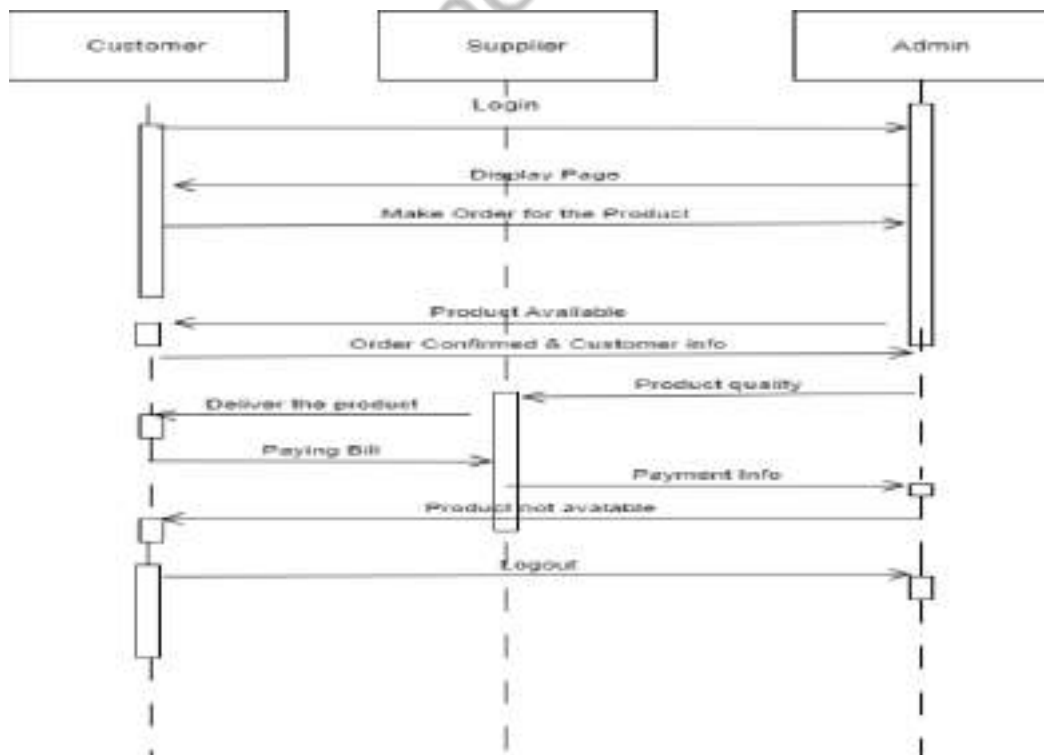
Class Diagram-



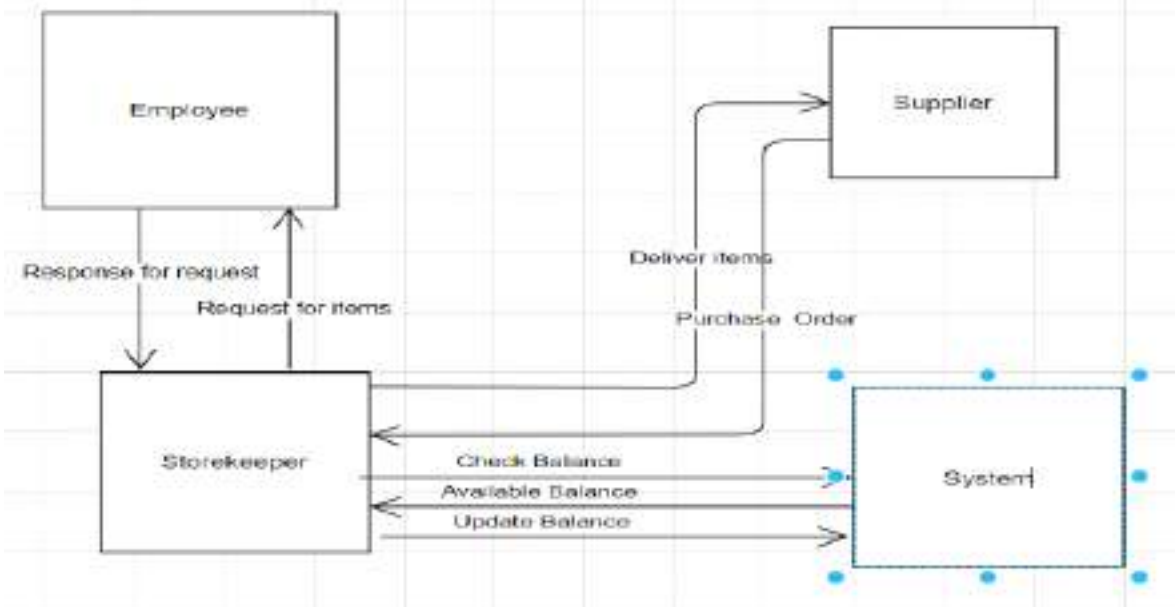
Object Diagram-



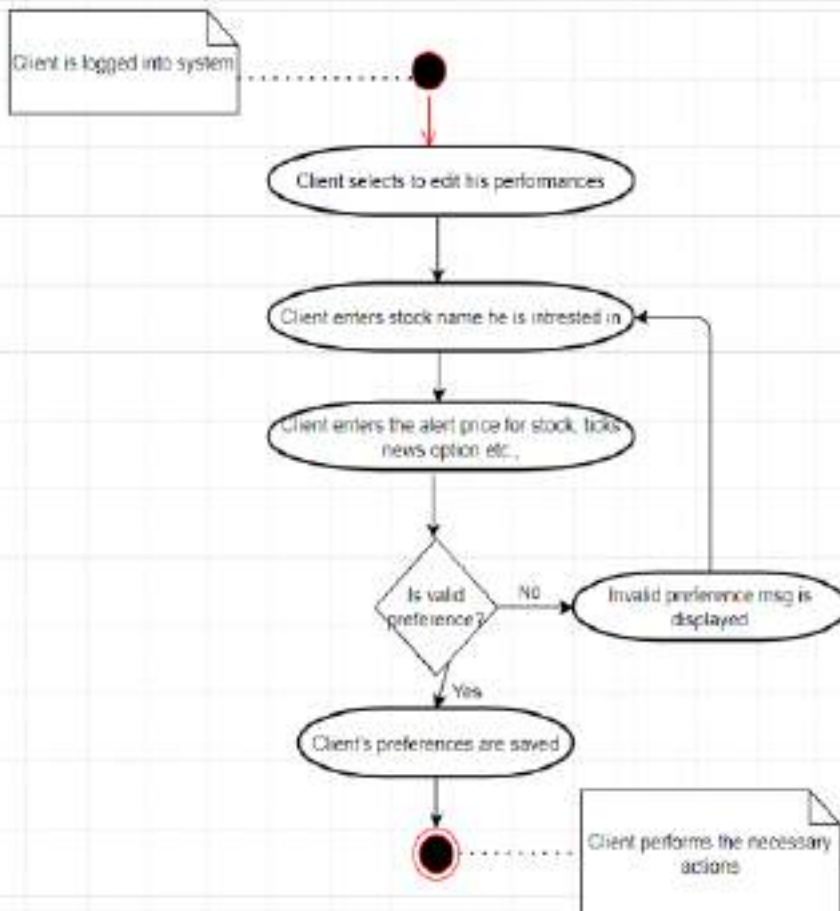
Sequence Diagram-



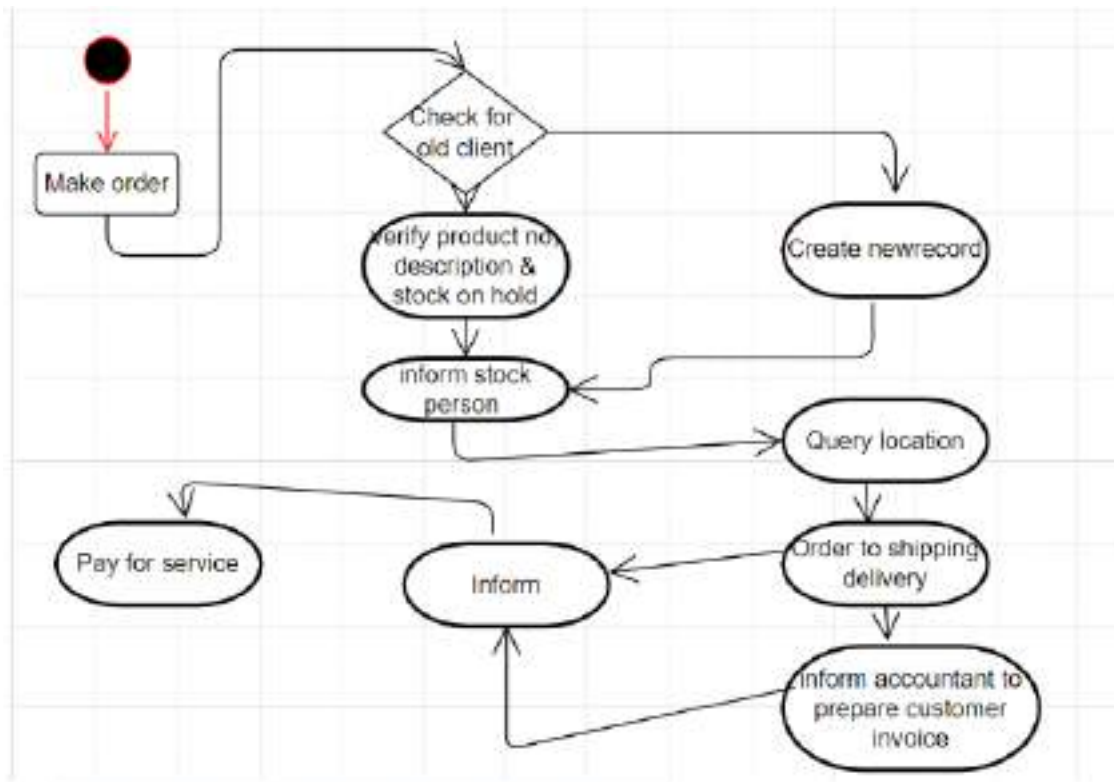
Collaboration Diagram-



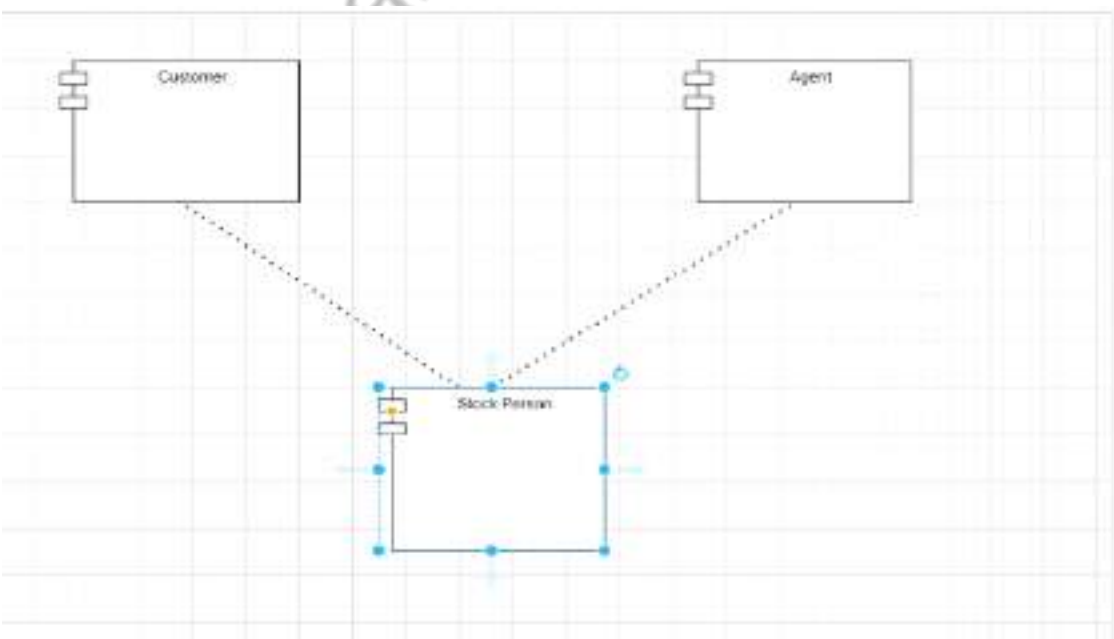
State chart Diagram-



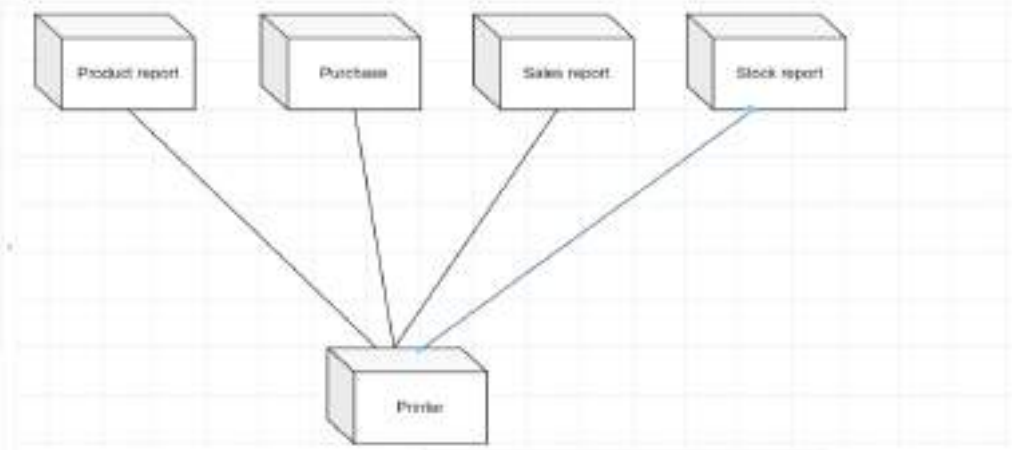
Activity Diagram:



Component Diagram:



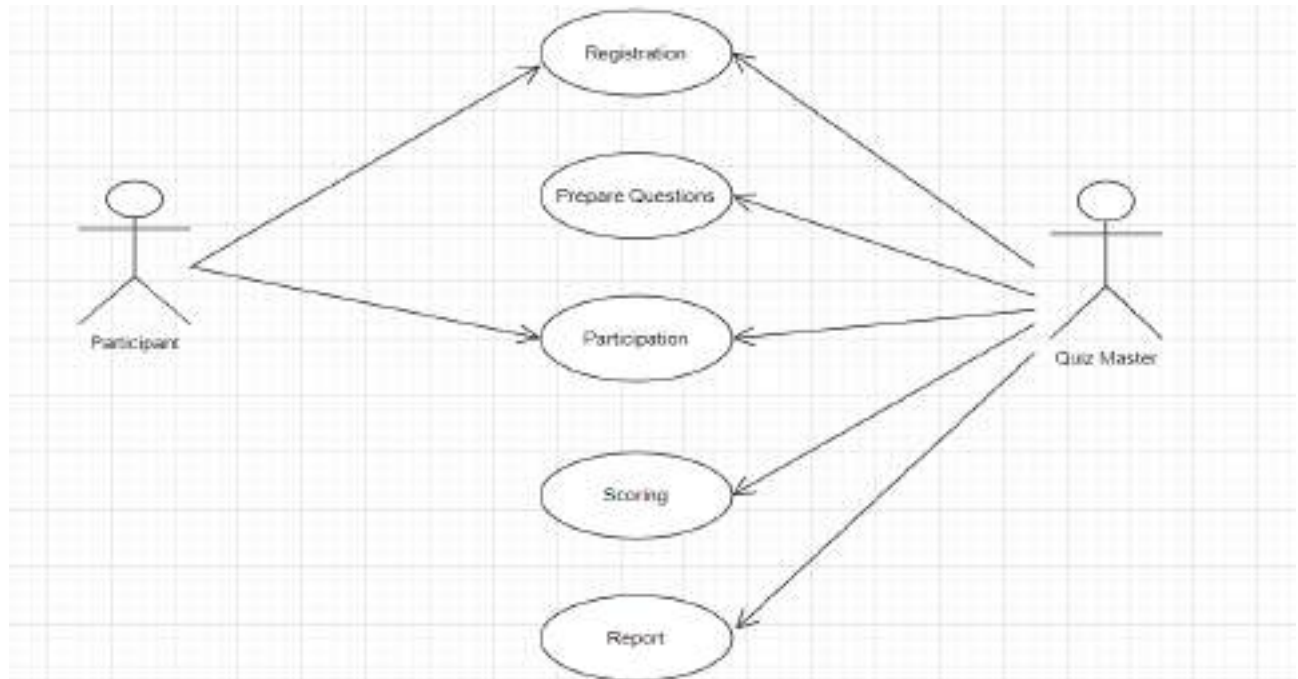
Deployment Diagram:



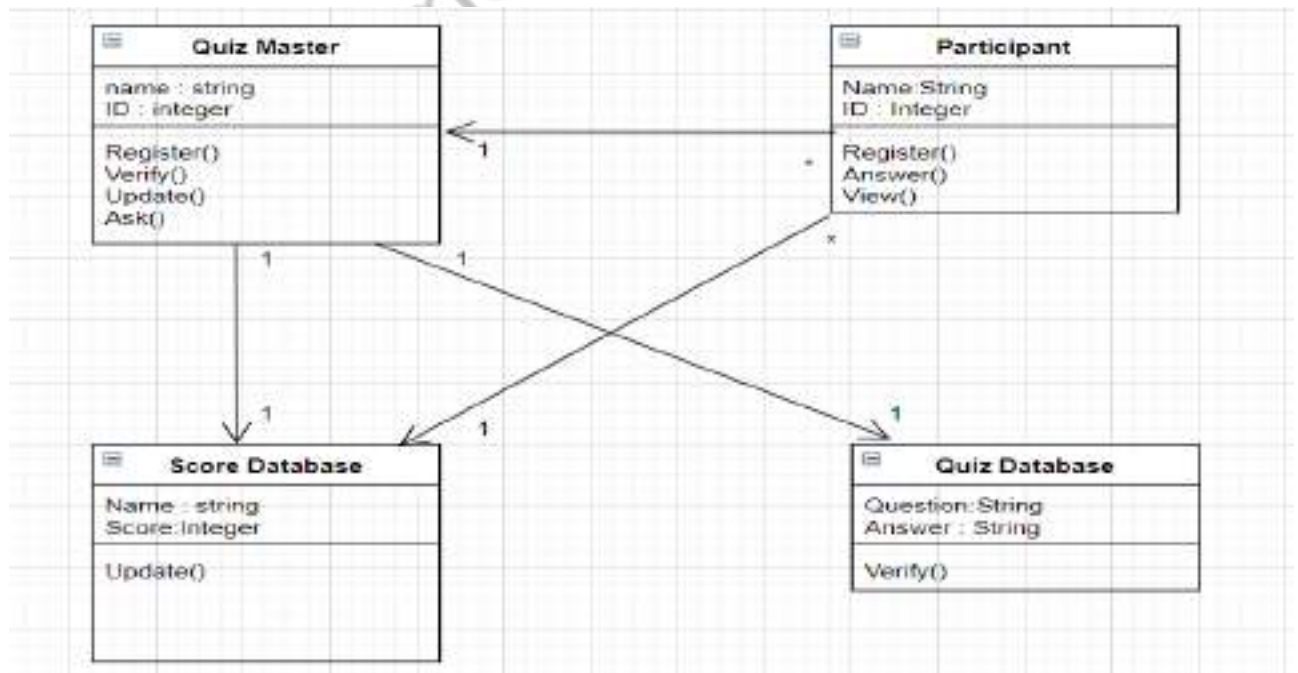
Department of

QUIZ SYSTEM

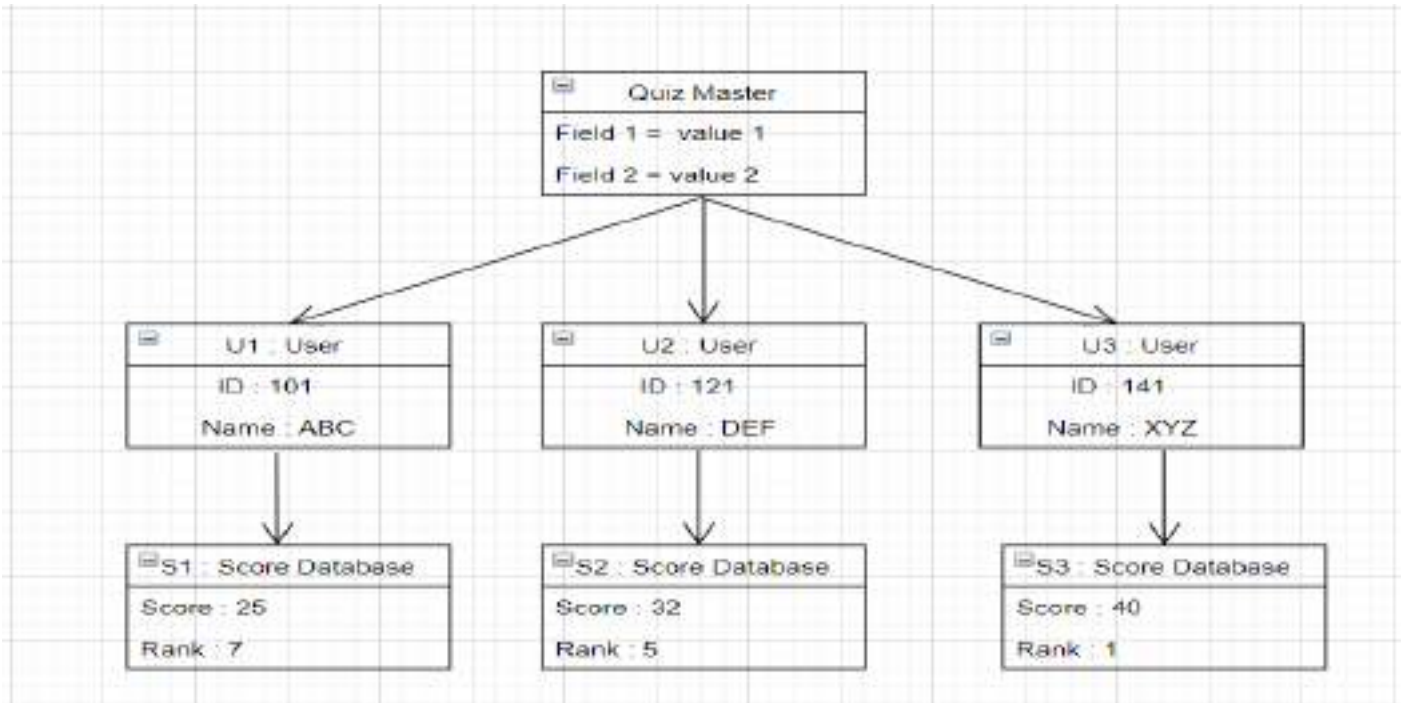
USE CASE DIAGRAM-



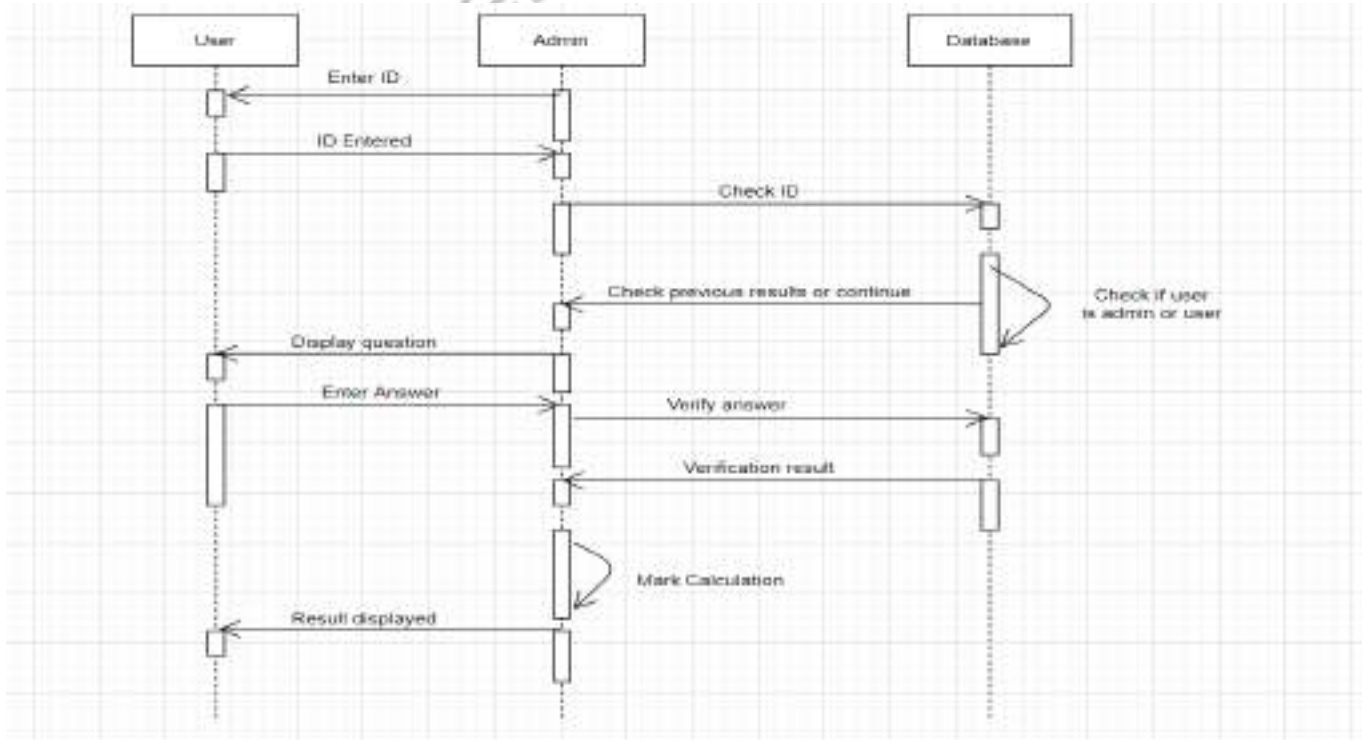
CLASS DIAGRAM-



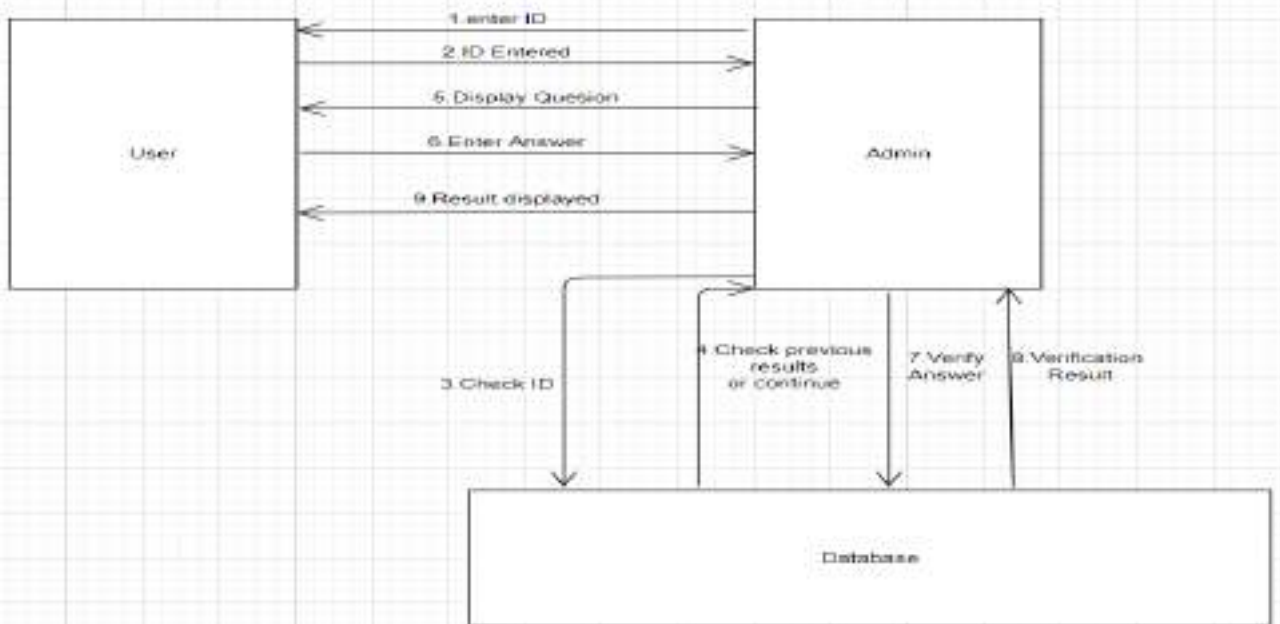
OBJECT DIAGRAM –



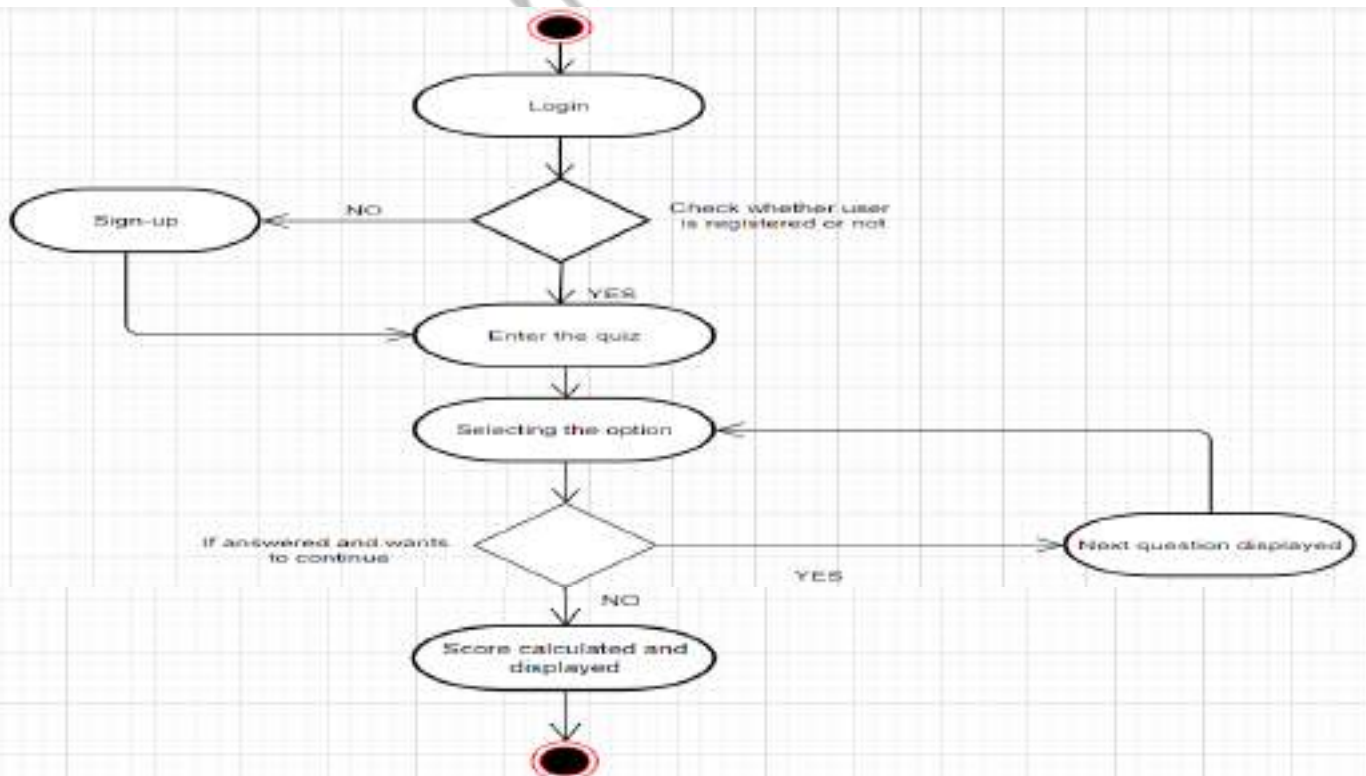
SEQUENCE DIAGRAM-



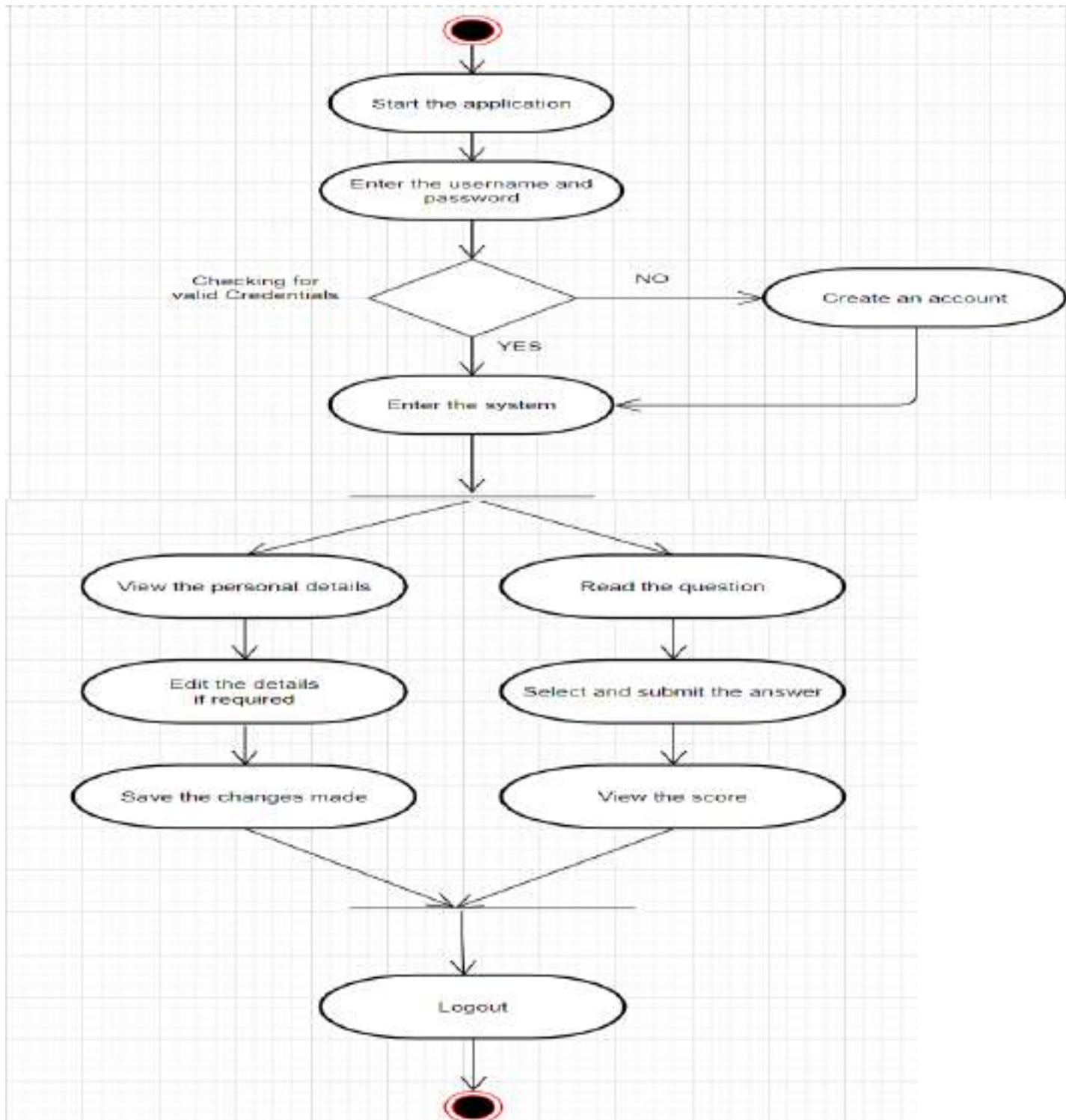
COLLABORATION DIAGRAM-



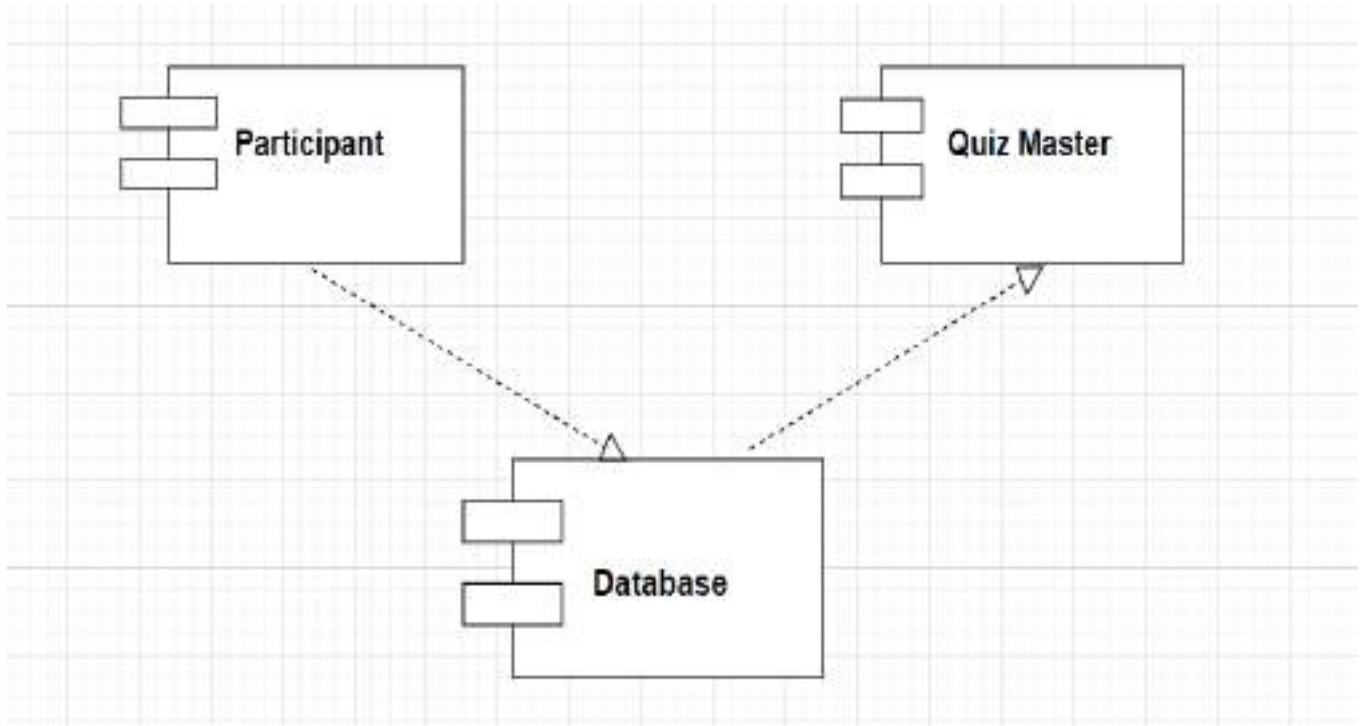
STATECHART DIAGRAM –



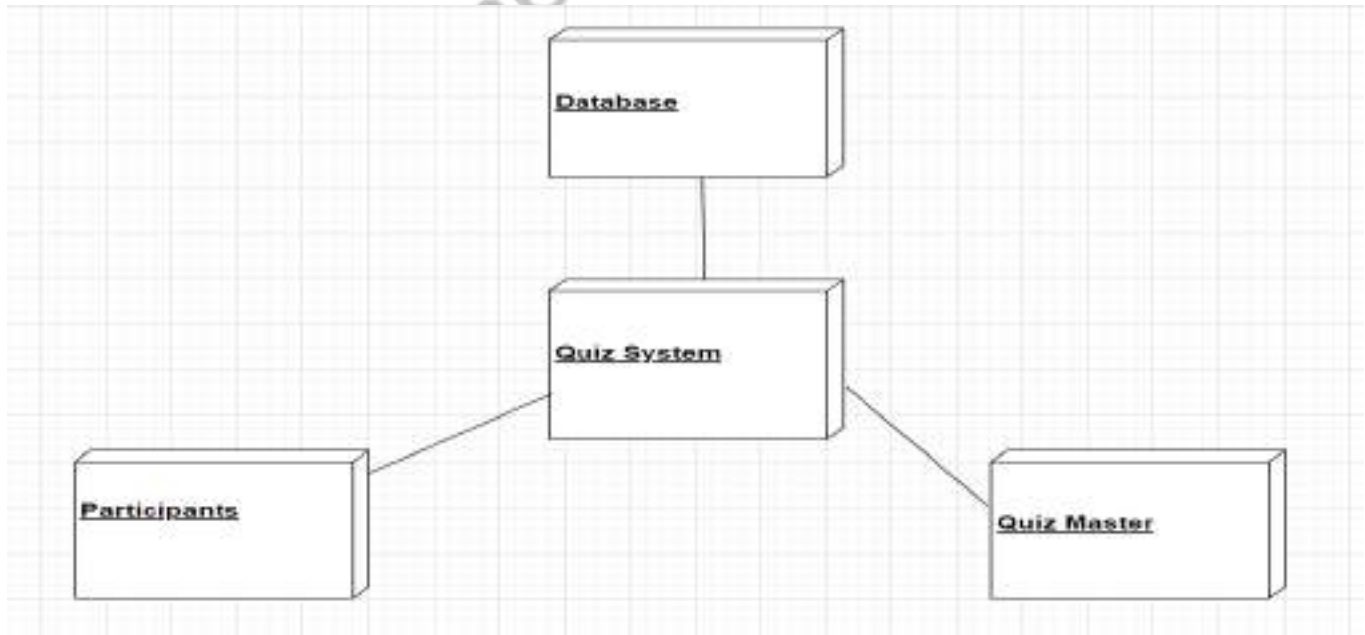
ACTIVITY DIAGRAM –



COMPONENT DIAGRAM -

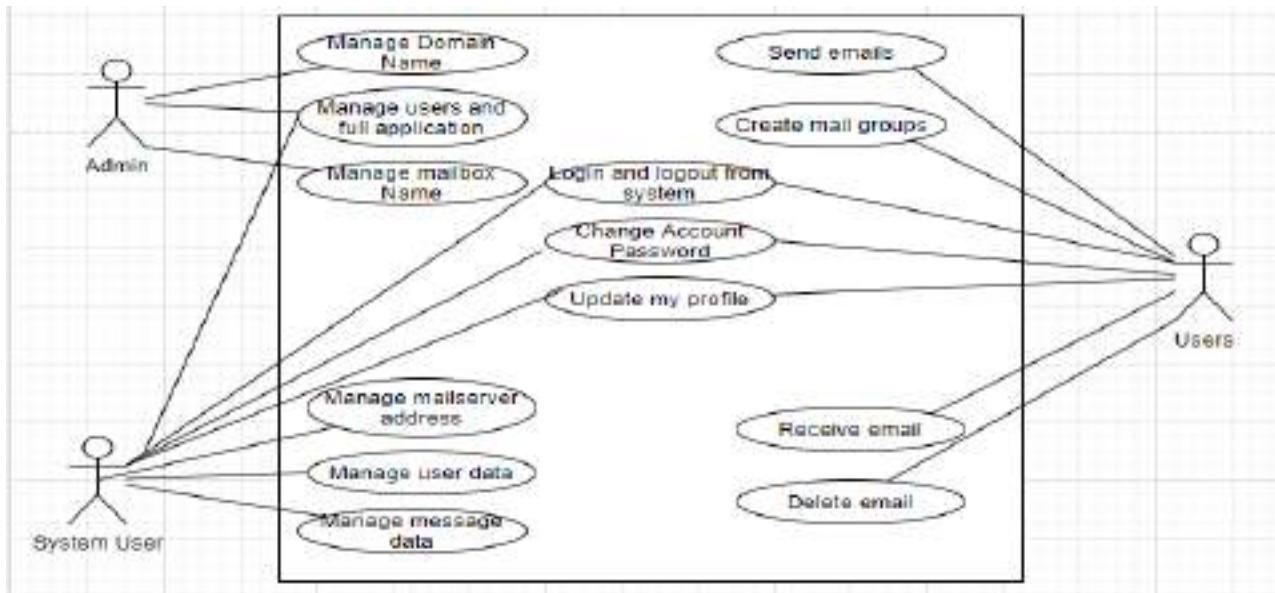


DEPLOYMENT DIAGRAM -

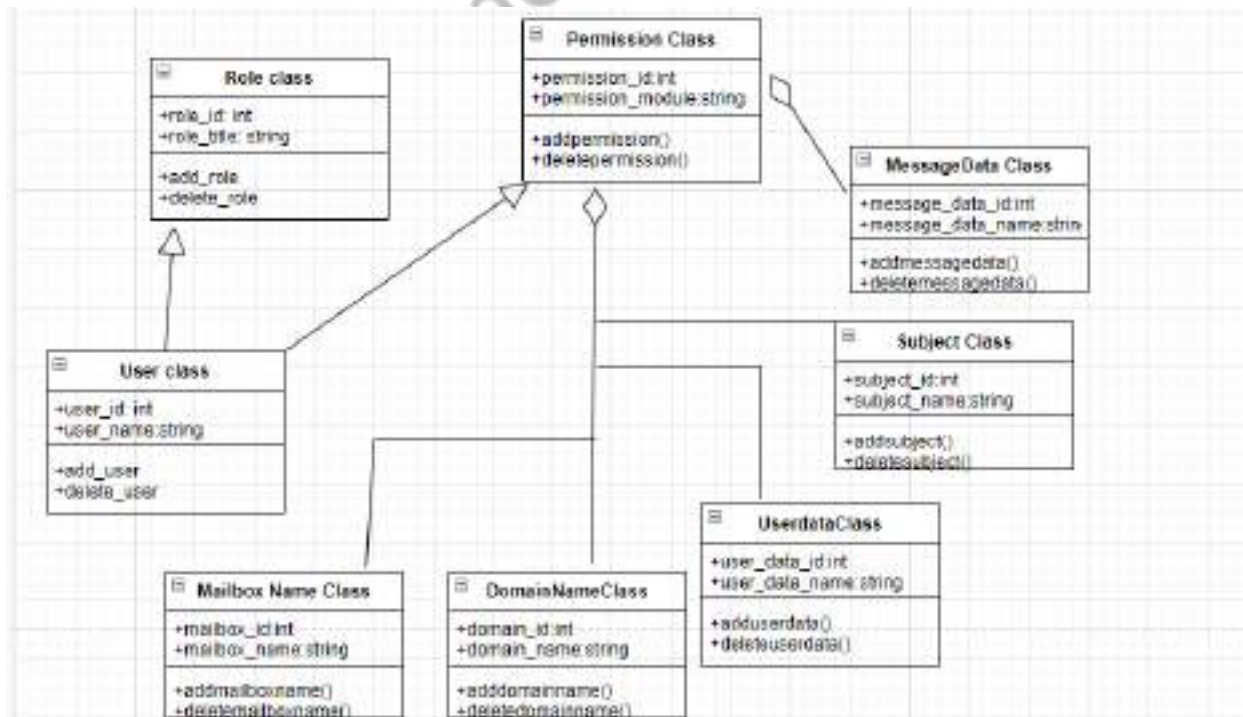


EMAIL CLIENT SYSTEM

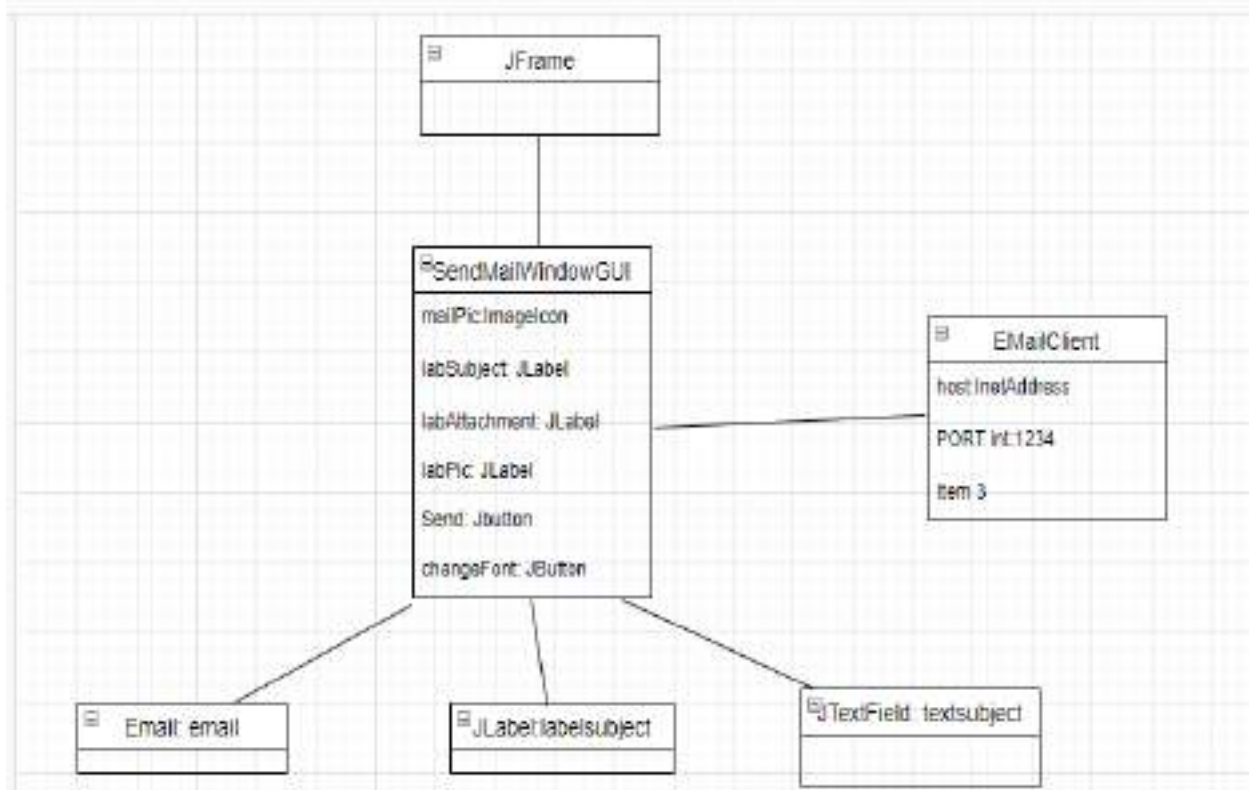
USE CASE DIAGRAM



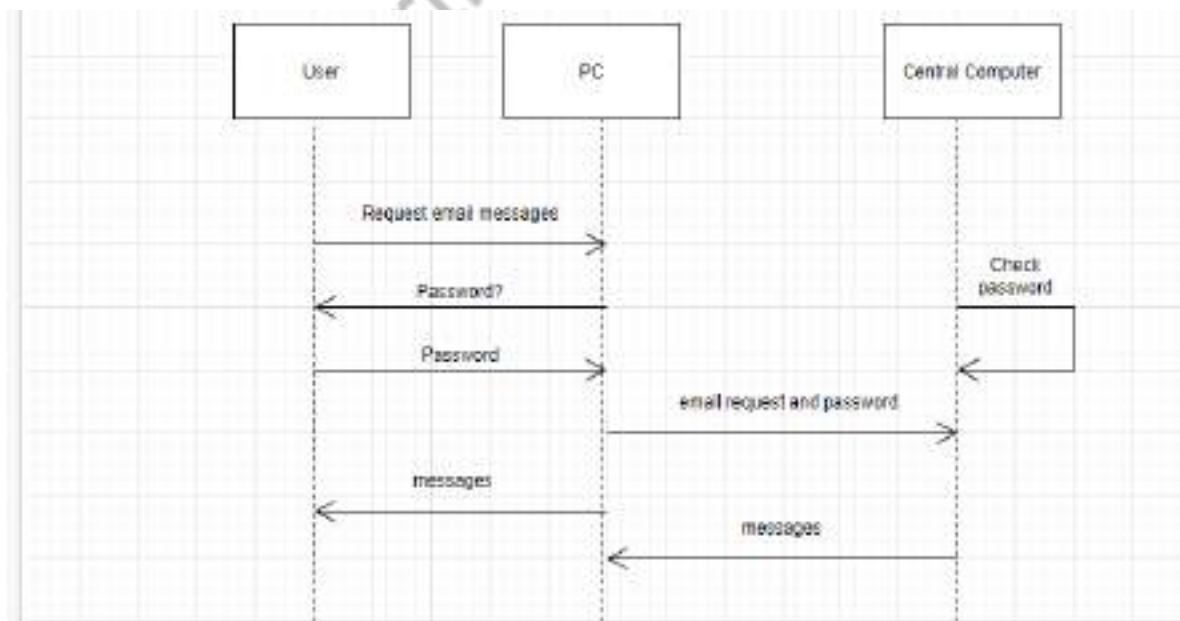
CLASS DIAGRAM –



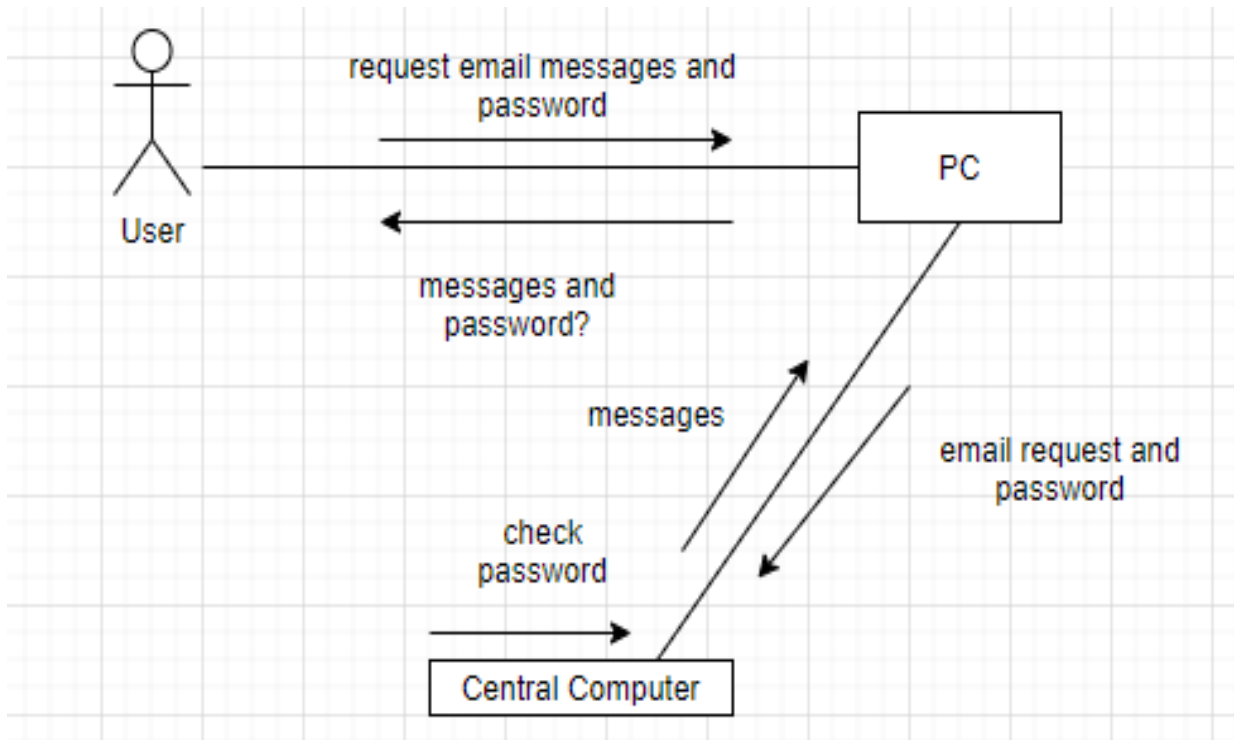
OBJECT DIAGRAM -



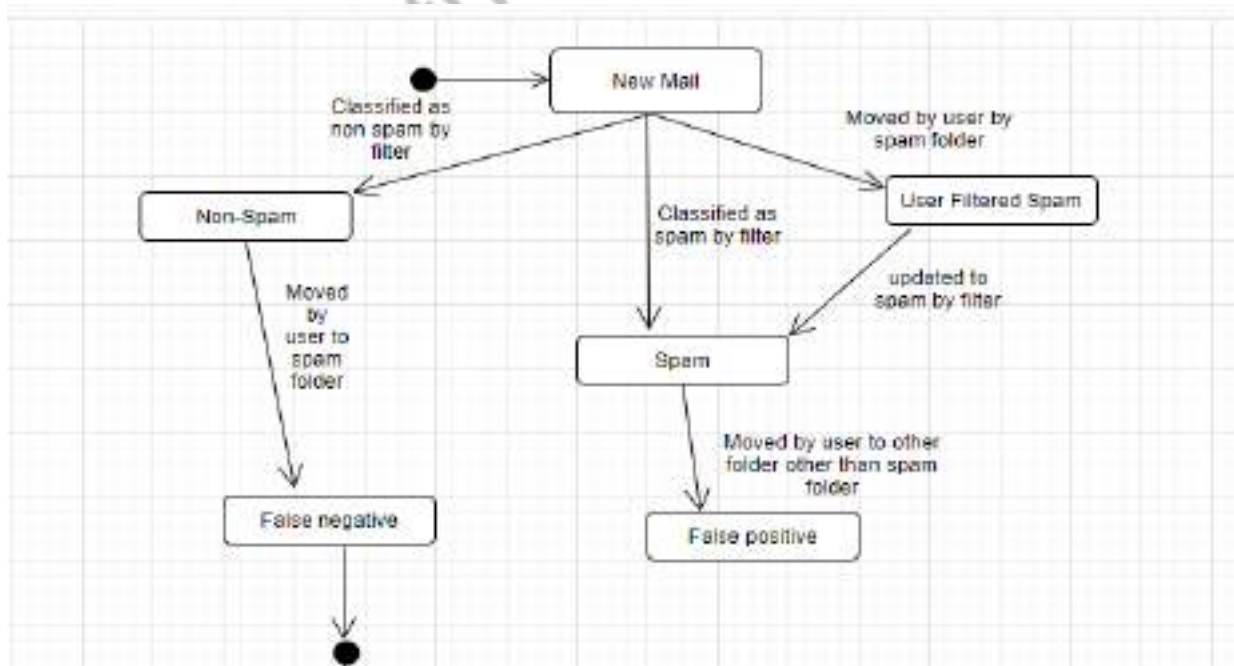
SEQUENCE DIAGRAM -



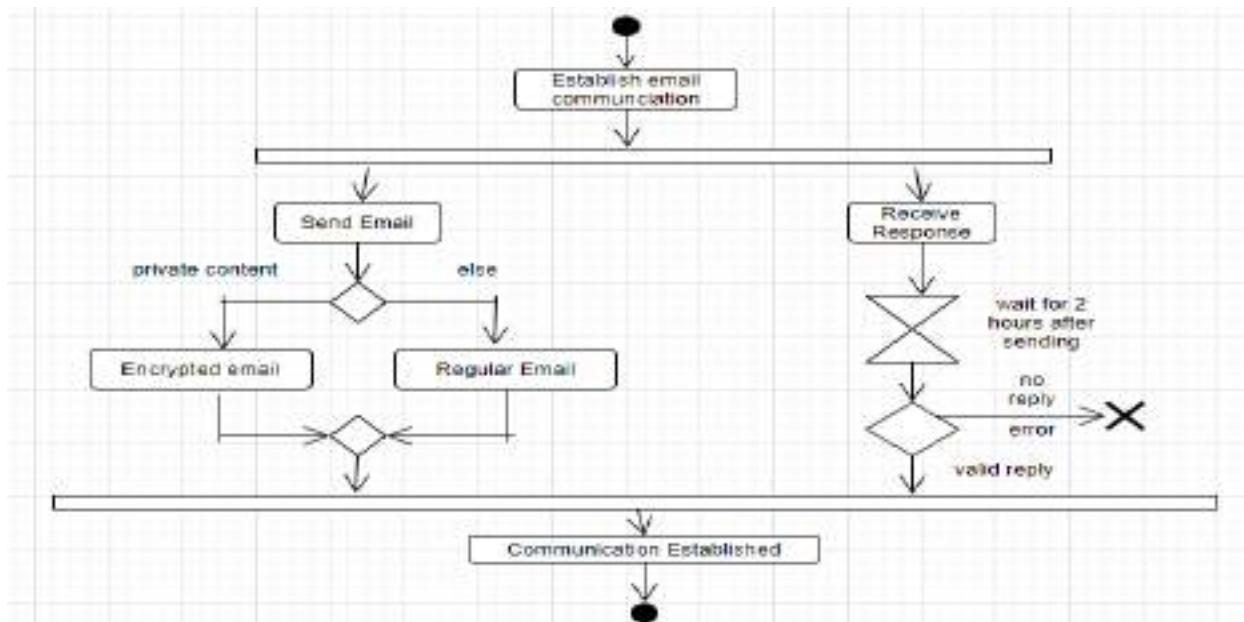
COLLABORATION DIAGRAM -



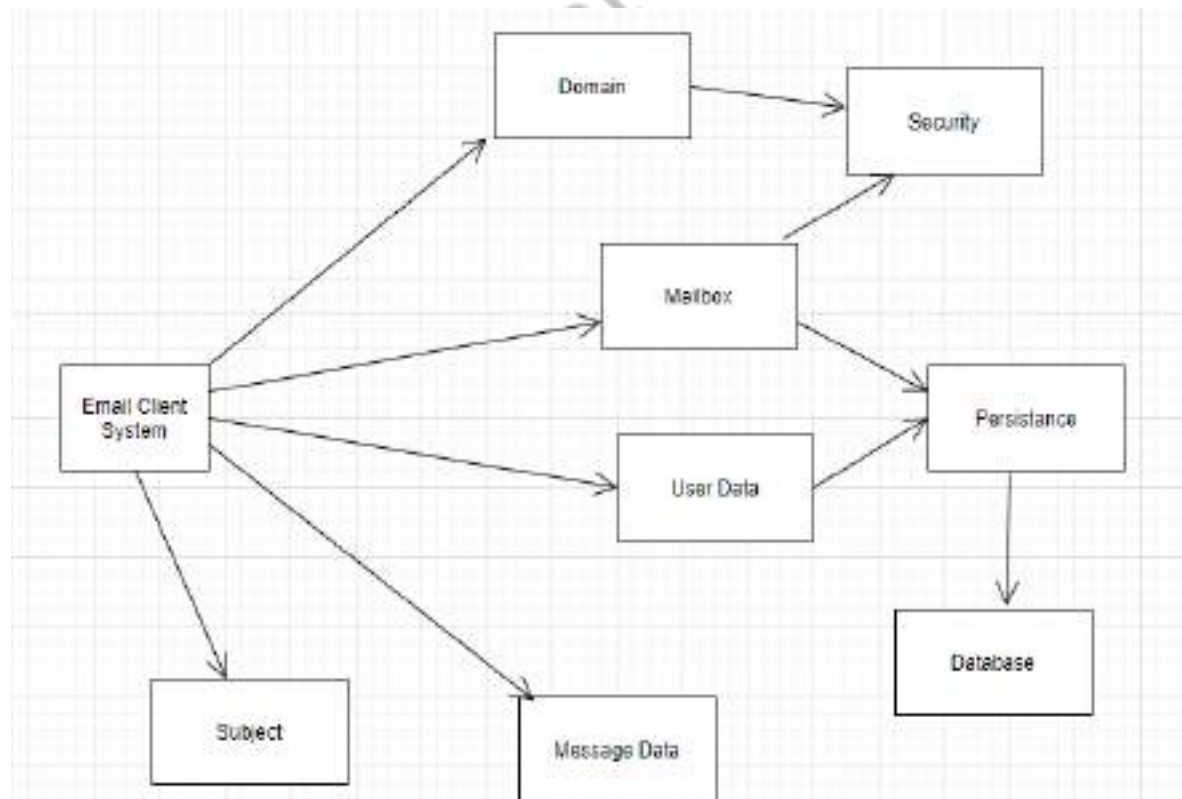
STATE DIAGRAM -



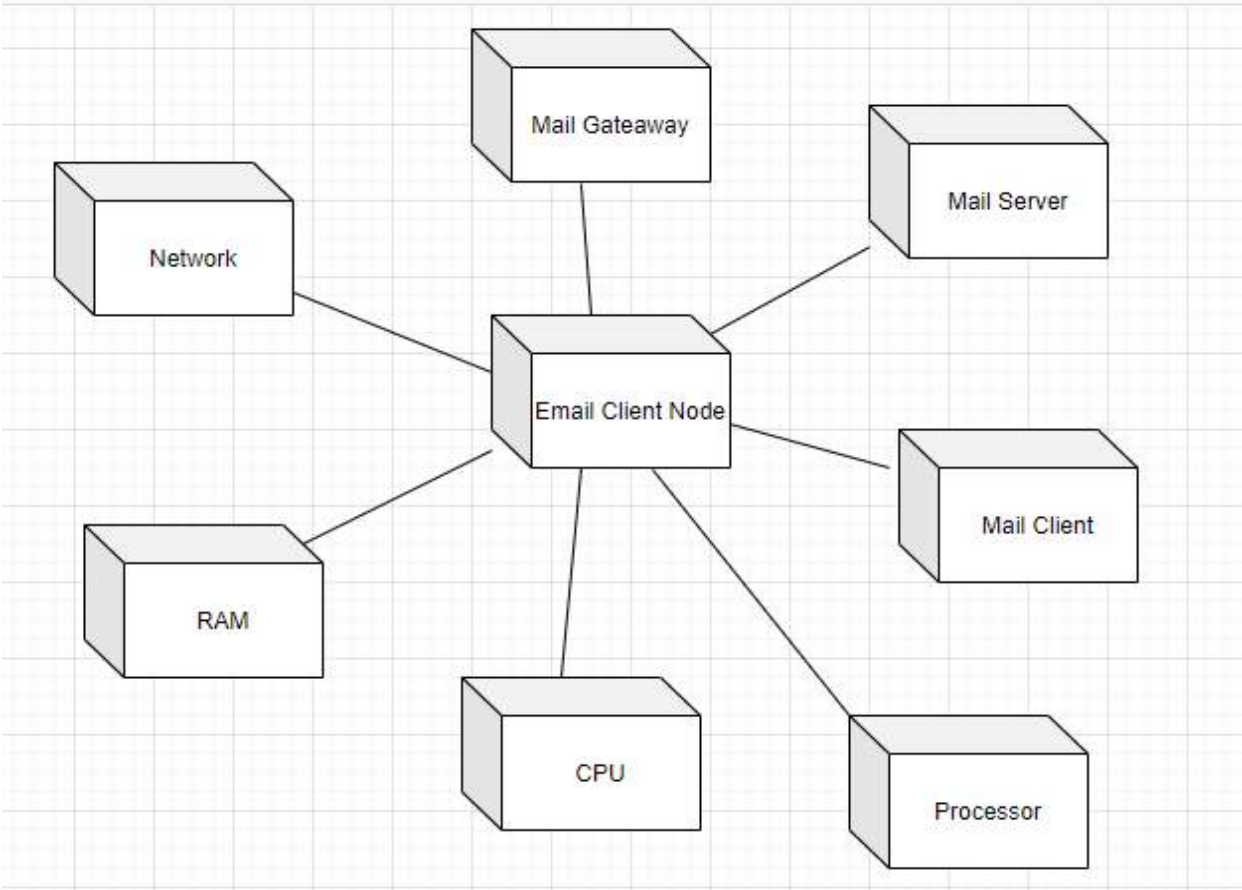
ACTIVITY DIAGRAM -



COMPONENT DIAGRAM -



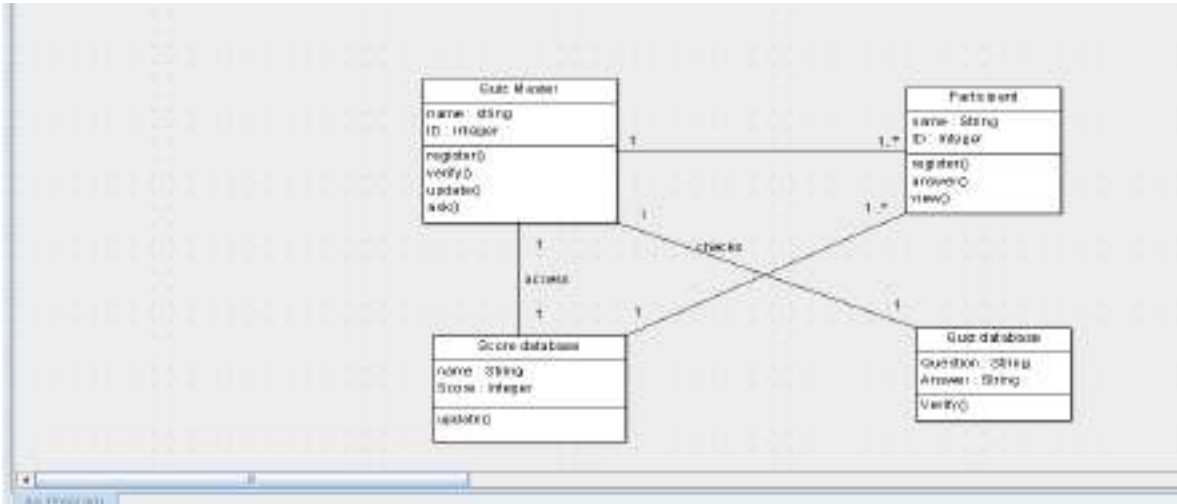
DEPLOYMENT DIAGRAM -



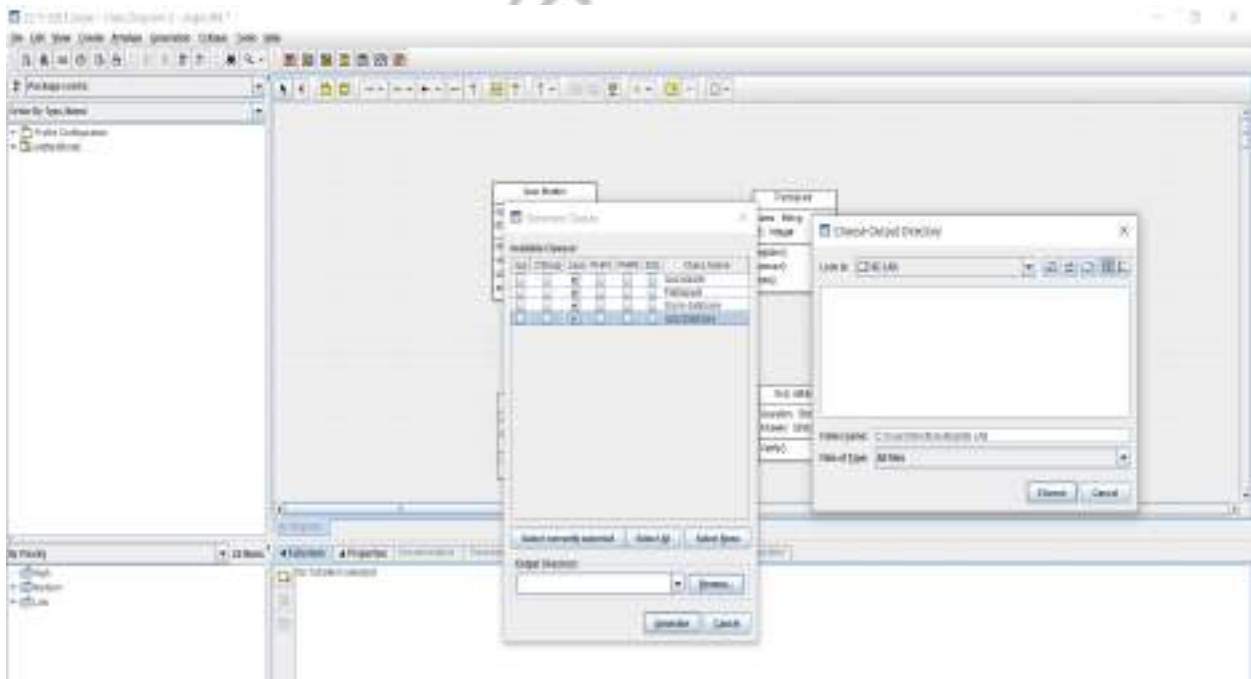
Depa.

FORWARD ENGINEERING

Step 1 - Draw the class diagram of the application and save it.



Step2-Click on Generation and select generate code for current diagram. A checkbox will appear. Select the language in which you want your code to be generated. For the generated code to be saved select a location.



Step3 - Once the language and location are selected click on generate code. Now check the folder location to view the generated code.

Name	Date modified	Type	Size
22-1-2021	22-01-2021 03:24 PM	ArgoUML Project F...	7 KB
22-1-2021.zargo-	22-01-2021 03:24 PM	ZARGO- File	0 KB
Participant.java	12-02-2021 07:33 AM	JAVA File	1 KB
Quiz database.java	12-02-2021 07:33 AM	JAVA File	1 KB
Quiz Master.java	12-02-2021 07:33 AM	JAVA File	1 KB
S.drawio	10-01-2021 09:25 PM	DRAWIO File	19 KB
Score database.java	12-02-2021 07:33 AM	JAVA File	1 KB

Step4 - View the generated code.

Participant.java - Notepad

File Edit Format View Help

```
public class Participant {
```

```
    public String name;
```

```
    public Integer ID;
```

```
    public Quiz Master quiz Master;
```

```
    public Score database checks;
```

```
    public void register() {
```

```
    }
```

```
    public void answer() {
```

```
    }
```

```
    public void view() {
```

```
    }
```

```
}
```

QuizDatabase.java - Notepad

File Edit Format View Help

```
public class Quiz database {  
  
    public String Question;  
  
    public String Answer;  
  
    public Quiz Master quiz Master;  
  
    public void Verify() {  
    }  
  
}
```

QuizMaster.java - Notepad

File Edit Format View Help

```
import java.util.List;  
  
public class Quiz Master {  
  
    public string name;  
  
    public integer ID;  
    public Score database access;  
    /**  
     *  
     */  
    public List<Participant> participant;  
    public Quiz database quiz database;  
  
    public void register() {  
    }  
  
    public void verify() {  
    }  
  
    public void update() {  
    }  
  
    public void ask() {  
    }  
}
```

```
import java.util.List;
```

```
public class Score database {
```

```
    public String name;
```

```
    public Integer Score;
```

```
    public Quiz Master access;
```

```
    /**
```

```
     *
```

```
     *
```

```
    */
```

```
    public List<Participant> checks;
```

```
    public void update() {
```

```
    }
```

```
}
```

Depai

APPLICATION IMAGES

HOME PAGE-



STUDENT LOGIN PAGE-



SIGNUP PAGE-

Sign Up

Please fill in this form to create an account:

Email

Password

Repeat Password

Remember me

By creating an account you agree to our [Terms & Privacy](#).

STUDENT HOME PAGE-

Home Logout

Welcome to the Quiz.

[1. Check your previous test results](#)

[2. Attempt the quiz](#)

There are various topics which can be selected as per your choice.

- 1.C Programming Language
- 2.Python Programming Language
- 3.JAVA
- 4.Software Engineering
- 5.Database Management System
- 6.General Knowledge

ALL THE BEST!

© 2020 All rights reserved. | [Admin Panel](#)

STUDENTS PREVIOUS RESULTS PAGE-

Previous Marks

Customise Subject & Max. Marks							
Class : IX							
Sr.	Subject	PT1	PT2	PT3	NS	SEA	Annual Exam
1	English	10	10	10	5	5	80
2	Hindi	10	10	10	5	5	80
3	Mathematics	10	10	10	5	5	80
4	Science	10	10	10	5	5	80
5	Social Studies	10	10	10	5	5	80
6	Computer Sc.	10	10	10	5	5	80

QUIZ PAGE FOR STUDENT-

Quiz

Question 1 / Total 1

What is 1 + 1

1

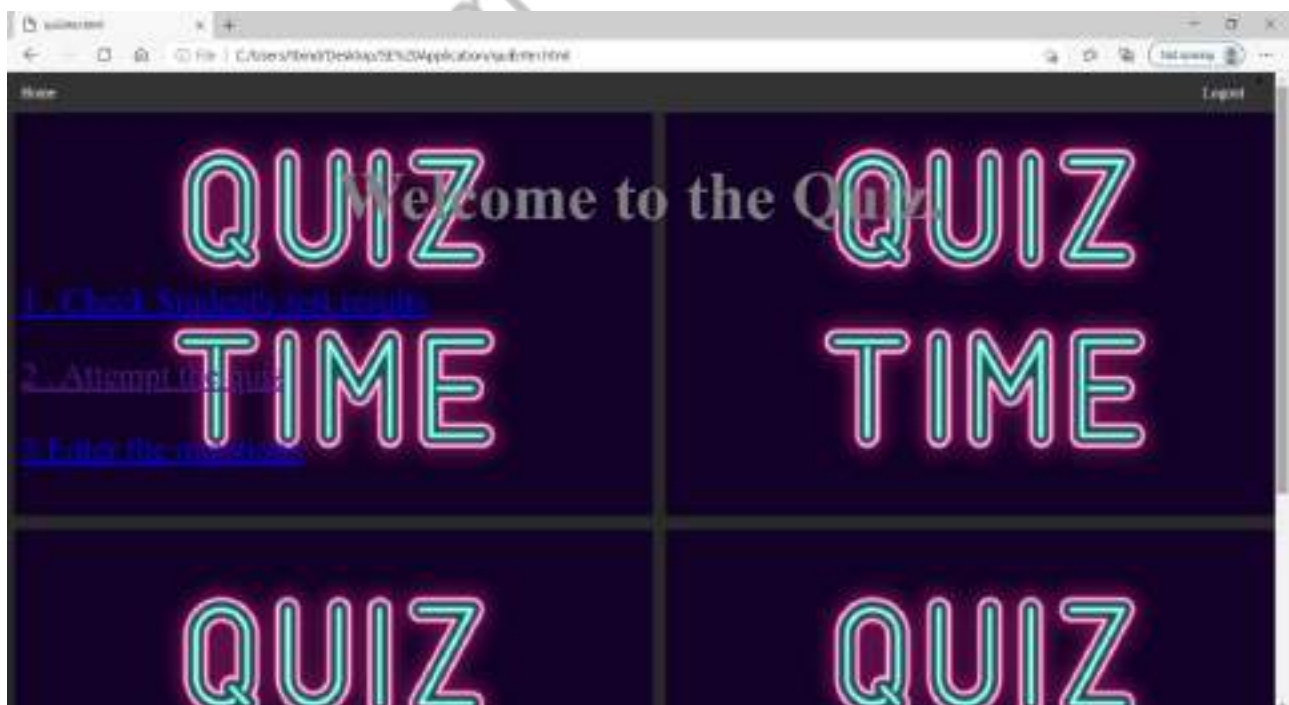
2

3

QUIZ MASTER LOGIN PAGE –



QUIZ MASTER HOME PAGE-



PREVIOUS TEST SCORES OF STUDENTS PAGE –



Previous Results

Roll No.	Name	English				Hindi				Mathematics							
		FT-001-1001	Assess Exam	Total	Remarks	Class Mark	FT-001-1001	Assess Exam	Total	Remarks	Class Mark	FT-001-1001	Assess Exam	Total	Remarks		
	Max. Marks	20	50	70		20	50	70		20	50	70		20	50		
1	Anam Kaur	15.00	34.00	49.00	A2	4	15.50	19.00	B1	4	16.00	32.00	48.00	B2	1	15.00	40.00
2	Moni Kaur	16.50	35.00	51.50	A2	5	17.50	18.00	B1	4	16.00	34.00	48.00	B2	5	17.00	54.00
3	Mansik Singh	16.50	38.00	54.50	A2	3	14.00	18.00	A1	3	13.00	18.00	31.00	B2	5	18.00	56.00
4	Praty Singh	15.50	38.00	53.50	A1	2	16.00	14.00	B3	6	15.50	35.00	50.50	B1	1	15.50	51.00
5	Shani C	14.50	38.00	52.50	B1	8	15.50	14.00	B3	8	16.50	18.00	34.50	B1	2	14.00	57.00
6	Darsh	16.50	45.00	61.50	C1	9	16.00	16.00	A1	2	14.50	14.00	28.50	B2	6	16.00	71.00
7	Jyoti	17.00	48.00	65.00	A2	4	14.50	18.00	C1	9	16.50	17.00	33.50	B1	5	17.50	65.00
8	Laksh	16.50	35.00	51.50	B1	7	15.50	17.00	A1	1	13.00	16.00	29.00	B2	5	16.50	62.00
9	Rishi	16.50	35.00	51.50	A1	1	16.00	14.00	B3	6	16.00	14.00	30.00	B1	4	15.00	61.00
10	Rohan	16.50	38.00	54.50	A1	3	15.00	16.00	B3	7	16.00	17.00	33.00	B2	8	16.50	74.00

QUIZ MASTER'S QUIZ PAGE –



QUIZ MASTER'S QUESTION ENTRY PAGE –



Please enter the questions for the test.

Question:

Answer:

Depart

QUIZ SYSTEM CODE

HOME PAGE CODE-

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
  background-image: url("quiz.png");
}
body {margin:0;}
.navbar {
  overflow: hidden;
  background-color: #333;
  position: fixed;
  top: 0;
  width: 100%;
}
.navbar a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.navbar a:hover {
  background: #ddd;
  color: black;
}
.main {
  padding: 16px;
  margin-top: 30px;
  height: 1000px;
}
h1{
  color:grey;
  font-size:500%;
  text-align: center;
}
```

```

p {
  margin: 4px;
  font-size: 90%;
}
</style>
</head>
<body>
<div class="navbar">
  <a href="home1.html">Home</a>
  <a href="web.html">Student Login</a>
  <a href="web1.html">Quiz Master Login</a>
  <li style="float:right"><a class="active" href="#web">Contact No:9876543210</a></li>
</div>
<div class="main">
<style>
body {
  background-color:lightgreens;
}
</style>
<h1 >Welcome to the Quiz System</h1>
</body>
</html>

```

STUDENT LOGIN PAGE CODE-

```

<!DOCTYPE html>
<html>
<head>
  <title>STUDENT LOGIN</title>
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<style>
body {
  background-image: url("quiz.png");
}
.login{
  width: 382px;
  overflow: hidden;
  margin: auto;
  margin: 20 0 0 450px;
  padding: 80px;
}
h2{
  text-align: center;
  color: #277582;

```

```
padding: 20px;
font-size:50px
}
label{
color: gray;
font-size: 25px;
}
#Uname{
width: 300px;
height: 30px;
border: none;
border-radius: 3px;
padding-left: 8px;
}
#Pass{
width: 300px;
height: 30px;
border: none;
border-radius: 3px;
padding-left: 8px;
}
}
#log{
width: 300px;
height: 30px;
border: none;
border-radius: 17px;
padding-left: 7px;
color: black;
}
#sign{
width: 300px;
height: 30px;
border: none;
border-radius: 17px;
padding-left: 7px;
color: black;
}
span{
color: white;
font-size: 20px;
}
}
```

Department of IT, GNITS


```

a{
  float: centre;
  background-color: grey;
}
h2{
  font-size:350%;
}
</style>
<body>
  <h2 style="color:gray;">STUDENT LOGIN</h2><br>
  <div class="login">
    <form id="login" method="get" action="login.php">
      <label><b>User Name</b>
      </label>
      <input type="text" name="Uname" id="Uname" placeholder="Username">
      <br><br>
      <label><b>Password</b>
      </label>
      <input type="Password" name="Pass" id="Pass" placeholder="Password">
      <br><br>
      <a href="stuEnter.html"><input type="button" name="log" id="log" value="Log In">
      </a>
      <br><br>
      <a href="signup.html"><input type="button" name="sign" id="sign" value="Sign Up">
      </a>
      <br><br>
      <input type="checkbox" id="check">
      <span>Remember me</span>
      <br><br>
    </form>
  </div>
</body>
</html>

```

STUDENT SIGNUP PAGE CODE –

```

<!DOCTYPE html>
<html>
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}

/* Full-width input fields */
input[type=text], input[type=password] {

```

```
width: 100%;
padding: 15px;
margin: 5px 0 22px 0;
display: inline-block;
border: none;
background: #f1f1f1;
}

/* Add a background color when the inputs get focus */
input[type=text]:focus, input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}
/* Set a style for all buttons */
button {
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}
button:hover {
  opacity: 1;
}
/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color: #f44336;
}
/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
  float: left;
  width: 50%;
}

/* Add padding to container elements */
.container {
  padding: 16px;
}
```

```
/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
  overflow: auto; /* Enable scroll if needed */
  background-color: #474e5d;
  padding-top: 50px;
}
/* Modal Content/Box */
.modal-content {
  background-color: #fefefe;
  margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
  border: 1px solid #888;
  width: 80%; /* Could be more or less, depending on screen size */
}
/* Style the horizontal ruler */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}
/* The Close Button (x) */
.close {
  position: absolute;
  right: 35px;
  top: 15px;
  font-size: 40px;
  font-weight: bold;
  color: #f1f1f1;
}
.close:hover,
.close:focus {
  color: #f44336;
  cursor: pointer;
}
```

```

/* Clear floats */
.clearfix::after {
  content: "";
  clear: both;
  display: table;
}
/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
  .cancelbtn, .signupbtn {
    width: 100%;
  }
}
</style>
<body>

<h2>Signup Form</h2>
<button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Sign Up</button>
<div id="id01" class="modal">
  <span onclick="document.getElementById('id01').style.display='none'" class="close"
title="Close Modal">&times;</span>
  <form class="modal-content" action="/action_page.php">
    <div class="container">
      <h1>Sign Up</h1>
      <p>Please fill in this form to create an account.</p>
      <hr>
      <label for="email"><b>Email</b></label>
      <input type="text" placeholder="Enter Email" name="email" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>

      <label for="psw-repeat"><b>Repeat Password</b></label>
      <input type="password" placeholder="Repeat Password" name="psw-repeat" required>

      <label>
        <input type="checkbox" checked="checked" name="remember" style="margin-
bottom:15px"> Remember me
      </label>

```

```
<p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms & Privacy</a>.</p>
```

```
  <div class="clearfix">
    <button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
    <a href="stuEnter.html"><button type="submit" class="signupbtn">Sign Up</button></a>
    <a href="stuEnter.html"><input type="button" name="sign" id="sign" value="Sign
Up"></a>
  </div>
</div>
</form>
</div>
```

```
<script>
// Get the modal
var modal = document.getElementById('id01');
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
  if (event.target == modal) {
    modal.style.display = "none";
  }
}
</script>
```

```
</body>
</html>
```

STUDENT HOME PAGE CODE-

```
<!DOCTYPE html>
<html>
<head>

<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {margin:0;}
.navbar {
  overflow: hidden;
  background-color: #333;
  position: fixed;
  top: 0;
  width: 100%;}
```

```

.navbar a {
  float: left;
  display: block;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}
.navbar a:hover {
  background: #ddd;
  color: black;
}
.main {
  padding: 16px;
  margin-top: 30px;
  height: 1000px; /* Used in this example to enable scrolling */
}
p {
  margin: 4px;
  font-size: 90%;
}
footer {
  text-align: center;
  padding: 3px;
  background-color: DarkSalmon;
  color: white;
}
</style>
</head>
<body>
<div class="navbar">
  <a href="#home">Home</a>
  <li style="float:right"><a class="active" href="home1.html">Logout</a></li>
</div>
<div class="main">
<style>
body {
  background-color:lightgreens;
}
</style>
<h1 style="font-size:250%;">Welcome to the Quiz.</h1>

```

```

</head>
<body>
<p style="font-size:150%;"><a href="preResults.html">1 . Check your previous test
results</a></p>
<p style="font-size:150%;"><a href="student.html">2 . Attempt the quiz</a></p>
<p style="font-size:200%;">There are various topics which can be selected as per your choice.
</p>
<p style="font-size:150%;"> 1.C Programming Language</p>
<p style="font-size:150%;"> 2.Python Programming Language </p>
<p style="font-size:150%;"> 3.JAVA </p>
<p style="font-size:150%;"> 4.Software Engineering</p>
<p style="font-size:150%;"> 5.Database Management System</p>
<p style="font-size:150%;"> 6.General Knowledge</p>
<p style="font-size:300%;">ALL THE BEST!</p>
</div>
<footer>
  <p>Regarding any queries email : <br>
  <a href="#mail">quizsystem@gnits.ac.in</a></p>
</footer>

</body>
</html>

```

STUDENT PREVIOUS RESULTS CODE-

```

<!DOCTYPE html>
<html>
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;
}
h2{
  font-size:200%;
  color:Blue;
  text-align:center;
}
</style>
<body>
<h2 >Previous Marks</h2>

```

```

</body>
</html>
```

QUIZ CODE –

```
<!DOCTYPE html>
```

```
<html lang="en" xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta charset="utf-8" />
  <title>Quiz</title>
</head>
<body>
  <style>
    #table_1 {
      background-color: PaleVioletRed;
      border: 4px solid black;
      width: 1200px;
      height: 700px;
      color: white;
      text-align: center;
    }

    #center001 {
      text-align: center;
    }
  </style>
  <div id="table_1">
    <h2 style="font-size:300%;">Quiz</h2>
    <hr>
    <div id="center001" >Question: <text id="q001">0</text> Score: <text
id="score001">0</text> </text></div>
    <p id="question01"></p>
    <p id="msg01"></p>
    <br />
    <br />
    <div id="starter01">
      <button onclick="begin001()" id="b001" style="font-size:150%;">Begin</button><br />
    </div>
    <p id="n001"></p>
  </div>
  <script>
```



```

var b = 0;
b++;
var c = 2;
c--;
var d = 0;
d++;

function begin001() {
    document.getElementById("q001").innerHTML = d++;
    document.getElementById("question01").innerHTML = "What is 5 + 5?" + "<br /><br /><br />" + "<button onclick=ans001() id=q1>10</button><br />" + "<button onclick=ans002() id=q2>8</button><br />" + "<button onclick=ans002() id=q3 >25</button><br />";
    document.getElementById("starter01").innerHTML = "";
}
function ans001() {
    document.getElementById("score001").innerHTML = b++;
    document.getElementById("question01").innerHTML = "Correct";
    document.getElementById("starter01").innerHTML = "";
    document.getElementById("n001").innerHTML = "<button onclick=nxt1()>Next</button><br />";
}
function ans002() {
    document.getElementById("question01").innerHTML = "Incorrect";
    document.getElementById("starter01").innerHTML = "";
    document.getElementById("n001").innerHTML = "<button onclick=nxt1()>Next</button><br />";
    document.getElementById("attempts").innerHTML = c--;
    if (c < 0) {
        document.getElementById("question01").innerHTML = "";
        document.getElementById("msg01").innerHTML = "Sorry. You lost!";
        document.getElementById("n001").innerHTML = "";
    }
}
function nxt1() {
    document.getElementById("question01").innerHTML = "What is 2 x 8?" + "<br /><br /><br />" + "<button onclick=ans004() id=q4>14</button><br />" + "<button onclick=ans004() id=q5>20</button><br />" + "<button onclick=ans003() id=q6>16</button><br />";
    document.getElementById("n001").innerHTML = "";
    document.getElementById("q001").innerHTML = d++;
}
function ans003() {
    document.getElementById("score001").innerHTML = b++;
}

```

```

    document.getElementById("question01").innerHTML = "Correct";
    document.getElementById("starter01").innerHTML = "";
    document.getElementById("n001").innerHTML = "<button
onclick=nxt2()>Next</button><br />";
}
function ans004() {
    document.getElementById("question01").innerHTML = "Incorrect";
    document.getElementById("starter01").innerHTML = "";
    document.getElementById("n001").innerHTML = "<button
onclick=nxt2()>Next</button><br />";
    document.getElementById("attempts").innerHTML = c--;
if (c < 0) {
    document.getElementById("question01").innerHTML = "";
    document.getElementById("msg01").innerHTML = "Sorry. You lost!";
    document.getElementById("n001").innerHTML = "";
}
}
function nxt2() {
document.getElementById("question01").innerHTML = "";
document.getElementById("msg01").innerHTML = "End of Quiz.";
document.getElementById("n001").innerHTML = "";
}
</script>
</div>
</body>
</html>

```

QUIZ MASTER LOGIN CODE –

```

<!DOCTYPE html>
<html>
<head>
    <title>Quiz System</title>
    <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<style>
body {
    background-image: url("quiz.png");
}
.login{
    width: 382px;
    overflow: hidden;

```

```
        margin: auto;
        margin: 20 0 0 450px;
        padding: 80px;
    }
    h2{
        text-align: center;
        color: #277582;
        padding: 20px;
        font-size:50px
    }
    label{
        color: gray;
        font-size: 25px;
    }
    #Uname{
        width: 300px;
        height: 30px;
        border: none;
        border-radius: 3px;
        padding-left: 8px;
    }
    #Pass{
        width: 300px;
        height: 30px;
        border: none;
        border-radius: 3px;
        padding-left: 8px;
    }
    #log{
        width: 300px;
        height: 30px;
        border: none;
        border-radius: 17px;
        padding-left: 7px;
        color: black;
    }
    #sign{
        width: 300px;
        height: 30px;
        border: none;
        border-radius: 17px;
```

```

padding-left: 7px;
color: black;
}
span{
color: white;
font-size: 20px;
}
a{
float: centre;
background-color: grey;
}
</style>
<body>
<h2 style="color:gray;">Quiz Master Login</h2><br>
<div class="login">
<form id="login" method="get" action="login.php">
<label><b>User Name </b>
</label>
<input type="text" name="Uname" id="Uname" placeholder="Username">
<br><br>
<label><b>Password
</b>
</label>
<input type="Password" name="Pass" id="Pass" placeholder="Password">
<br><br>
<a href="quiEnter.html"><input type="button" name="log" id="log" value="Log In">
</a>
<br><br>
<a href="signup1.html"><input type="button" name="sign" id="sign" value="SignUp">
</a>
<br><br>
<input type="checkbox" id="check">
<span>Remember me</span>
<br><br>
</form>
</div>
</body>
</html>

```

QUIZ MASTER SIGNUP CODE –

```
<!DOCTYPE html>
<html>
<style>
body {font-family: Arial, Helvetica, sans-serif;}
* {box-sizing: border-box;}
/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 15px;
  margin: 5px 0 22px 0;
  display: inline-block;
  border: none;
  background: #f1f1f1;
}
/* Add a background color when the inputs get focus */
input[type=text]:focus, input[type=password]:focus {
  background-color: #ddd;
  outline: none;
}
/* Set a style for all buttons */
button {
  background-color: #4CAF50;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
  opacity: 0.9;
}
button:hover {
  opacity: 1;
}
/* Extra styles for the cancel button */
.cancelbtn {
  padding: 14px 20px;
  background-color: #f44336;
}
/* Float cancel and signup buttons and add an equal width */
.cancelbtn, .signupbtn {
```

```
float: left;
width: 50%;
}
/* Add padding to container elements */
.container {
padding: 16px;
}
/* The Modal (background) */
.modal {
display: none; /* Hidden by default */
position: fixed; /* Stay in place */
z-index: 1; /* Sit on top */
left: 0;
top: 0;
width: 100%; /* Full width */
height: 100%; /* Full height */
overflow: auto; /* Enable scroll if needed */
background-color: #474e5d;
padding-top: 50px;
}
/* Modal Content/Box */
.modal-content {
background-color: #fefefe;
margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
border: 1px solid #888;
width: 80%; /* Could be more or less, depending on screen size */
}
/* Style the horizontal ruler */
hr {
border: 1px solid #f1f1f1;
margin-bottom: 25px;
}
/* The Close Button (x) */
.close {
position: absolute;
right: 35px;
top: 15px;
font-size: 40px;
font-weight: bold;
color: #f1f1f1;
.close:hover,
.close:focus {
```

```

    color: #f44336;
    cursor: pointer;
}
/* Clear floats */
.clearfix::after {
    content: "";
    clear: both;
    display: table;
}
/* Change styles for cancel button and signup button on extra small screens */
@media screen and (max-width: 300px) {
    .cancelbtn, .signupbtn {
        width: 100%;
    }
}
</style>
<body>
<h2>Signup Form</h2>
<button onclick="document.getElementById('id01').style.display='block'"
style="width:auto;">Sign Up</button>
<div id="id01" class="modal">
    <span onclick="document.getElementById('id01').style.display='none'" class="close"
title="Close Modal">&times;</span>
    <form class="modal-content" action="/action_page.php">
        <div class="container">
            <h1>Sign Up</h1>
            <p>Please fill in this form to create an account.</p>
            <hr>
            <label for="email"><b>Email</b></label>
            <input type="text" placeholder="Enter Email" name="email" required>
            <label for="psw"><b>Password</b></label>
            <input type="password" placeholder="Enter Password" name="psw" required>
            <label for="psw-repeat"><b>Repeat Password</b></label>
            <input type="password" placeholder="Repeat Password" name="psw-repeat" required>
            <label>
                <input type="checkbox" checked="checked" name="remember" style="margin-
bottom:15px"> Remember me
            </label>
            <p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms &
Privacy</a>.</p>
            <div class="clearfix">

```

```

        <button type="button" onclick="document.getElementById('id01').style.display='none'"
class="cancelbtn">Cancel</button>
        <a href="quiEnter.html"><button type="submit" class="signupbtn">Sign Up</button></a>
        <a href="quiEnter.html"><input type="button" name="sign" id="sign" value="Sign
Up"></a>
    </div>
</div>
</form>
</div>
<script>
// Get the modal
var modal = document.getElementById('id01');
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
</script>
</body>
</html>

```

QUIZ MASTER HOME PAGE-

```

<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {
    background-image: url("quiztime.png");
}
body {margin:0;}
.navbar {
    overflow: hidden;
    background-color: #333;
    position: fixed;
    top: 0;
    width: 100%;
}
.navbar a {
    float: left;
    display: block;

```



```

color: #f2f2f2;
text-align: center;
padding: 14px 16px;
text-decoration: none;
font-size: 17px;
}
.navbar a:hover {
background: #ddd;
color: black;
}
.main {
padding: 16px;
margin-top: 30px;
height: 1000px; /* Used in this example to enable scrolling */
}
p {
margin: 14px;
font-size: 90%;
}
footer {
text-align: center;
padding: 3px;
background-color: DarkSalmon;
color: white;
}
h1{
text-align: center;
color:grey;
}
</style>
</head>
<body>
<div class="navbar">
<a href="#home">Home</a>
<li style="float:right"><a class="active" href="home1.html">Logout</a></li>
</div>
<div class="main">
<style>
body {
background-color:lightgreens;
}
</style>
<h1 style="font-size:500%;">Welcome to the Quiz.</h1>

```

```
</head>
<body>
<p style="font-size:250%;"><a href="preResults1.html">1 . Check Student's test
results</a></p>
<p style="font-size:250%;"><a href="student.html">2 . Attempt the quiz</a></p>
<p style="font-size:250%;"><a href="enterQues.html">3.Enter the questions</a></p>
</div>
<footer>
  <p>Regarding any queries email : <br>
  <a href="#mail">quizsystem@gnits.ac.in</a></p>
</footer>
</body>
</html>
```

STUDENT'S PREVIOUS TEST SCORES CODE -

```
<!DOCTYPE html>
<html>
<style>
img {
  display: block;
  margin-left: auto;
  margin-right: auto;}
h2{
  font-size:200%;
  color:Blue;
  text-align:center;}
</style>
<body>
<h2 >Previous Results</h2>

</body>
</html>
```

QUIZ MASTER'S TEST QUESTION'S ENTRY CODE –

```
<!DOCTYPE html>
<html>
<style>
body {
  background-color: violet;}
h2{
  font-size:300%;
  text-align:center;
}
form{
  text-align:center;
  font-size:250%;
  padding:20px;}
</style>
<body>
<h2 >Please enter the questions for the test.</h2>
<form action="/action_page.php">
  <label for="fname">Question:</label><br>
  <input type="text" id="Question" name="Question" ><br>
  <label for="lname" >Answer:</label><br>
  <input type="text" id="Answer" name="Answer" ><br><br>
  <a href="#submit" ><input type="button" name="submit" id="submit" value="Submit">
</form>
</body>
</html>
```