

Expt No: 1	INTRODUCTION TO UML DIAGRAM
Date:	

AIM

To study about the various UML diagram and its types.

INTRODUCTION

Unified Modeling Language ("UML") is the industry standard "language" for describing, visualizing, and documenting object-oriented (OO) systems. UML is a collection of a variety of diagrams for differing purposes. Each type of diagram models a particular aspect of OO design in an easy to understand, visual manner. The UML standard specifies exactly how the diagrams are to be drawn and what each component in the diagram means. UML is not dependent on any particular programming language, instead it focuses on the fundamental concepts and ideas that model a system. Using UML enables anyone familiar with its specifications to instantly read and understand diagrams drawn by other people. There are UML diagram for modeling static class relationships, dynamic temporal interactions between objects, the usages of objects, the particulars of an implementation, and the state transitions of systems. In general, a UML diagram consists of the following features:

- Entities: These may be classes, objects, users or systems behaviours.
- Relationship Lines that model the relationships between entities in the system.
- Generalization -- a solid line with an arrow that points to a higher abstraction of the present item.
- Association -- a solid line that represents that one entity uses another entity apart of its behaviour.
- Dependency -- a dotted line with an arrowhead that shows one entity depends on the behaviour of another entity.

Types:

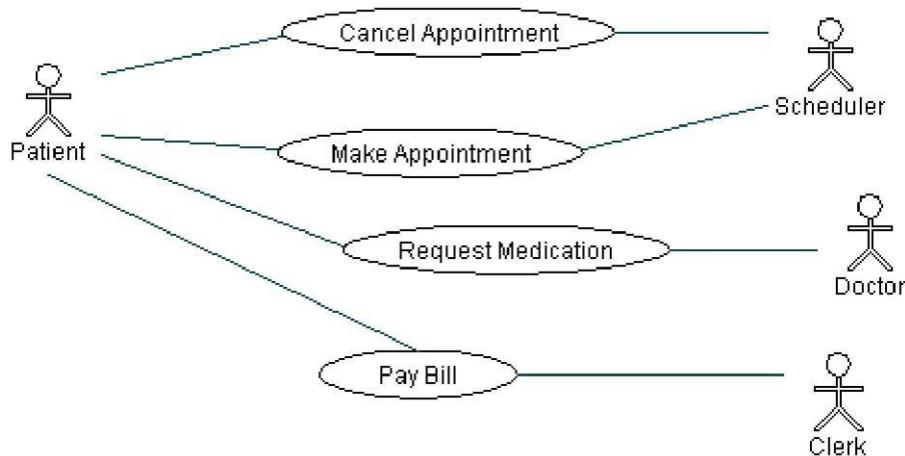
The UML diagrams are characterized into the following

- **Static:** The static characteristic of a system is essentially the structural aspect of the system. The static characteristics define what parts the system is made up of.
- **Dynamic:** The behavioural features of a system; for example, the ways a system behaves in response to certain events or actions are the dynamic characteristics of a system.
- **Implementation:** The implementation characteristic of a system is an entirely new feature that describes the different elements required for deploying a system.

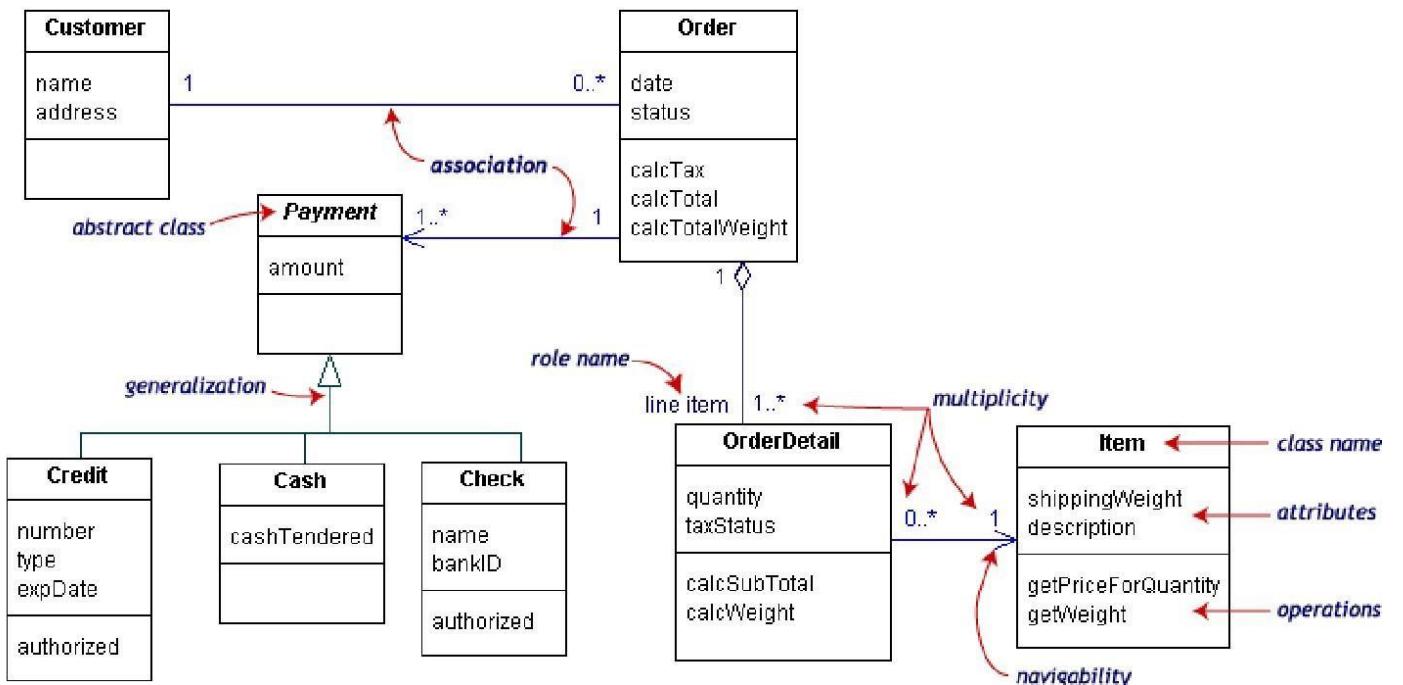
The UML diagrams that fall under each of these categories are:

- Statico Use case diagram o Class diagram
- Dynamico Object diagram o State diagram o Activity diagram o Sequence diagram o Collaboration diagram
- Implementationo Component diagram o Deployment diagram

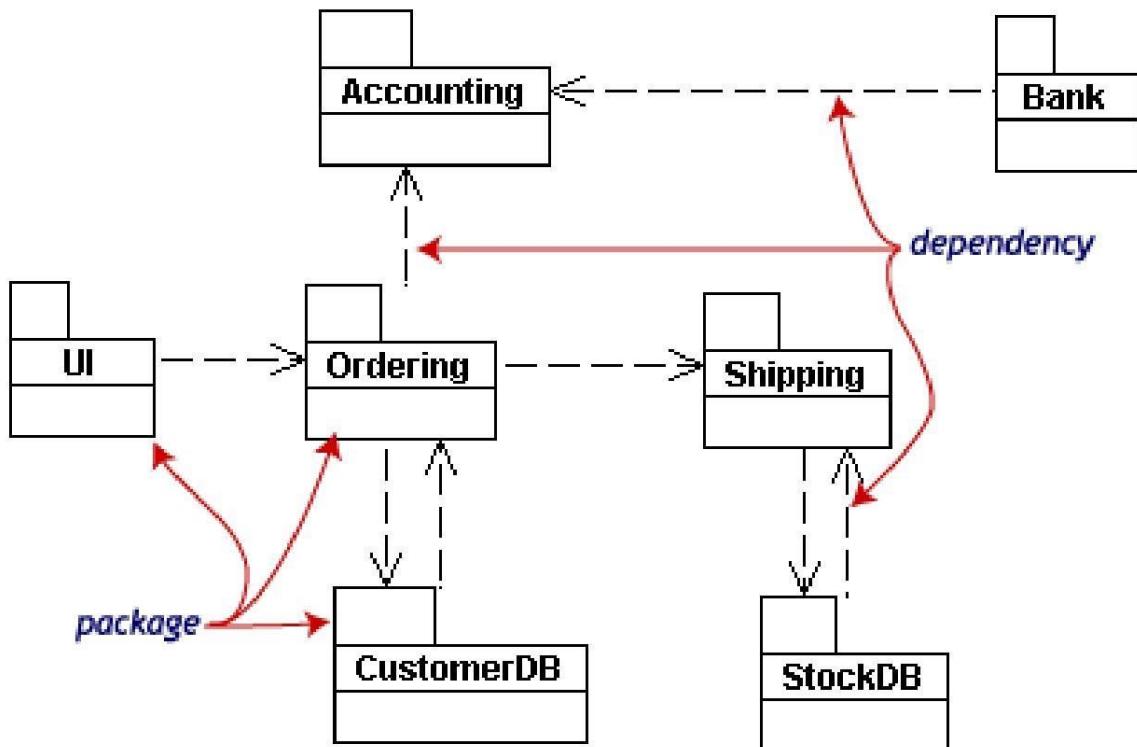
Use case diagram: The use case diagram is used to identify the primary elements and processes that form the system. The primary elements are termed as "actors" and the processes are called "use cases." The use case diagram shows which actors interact with the use cases.



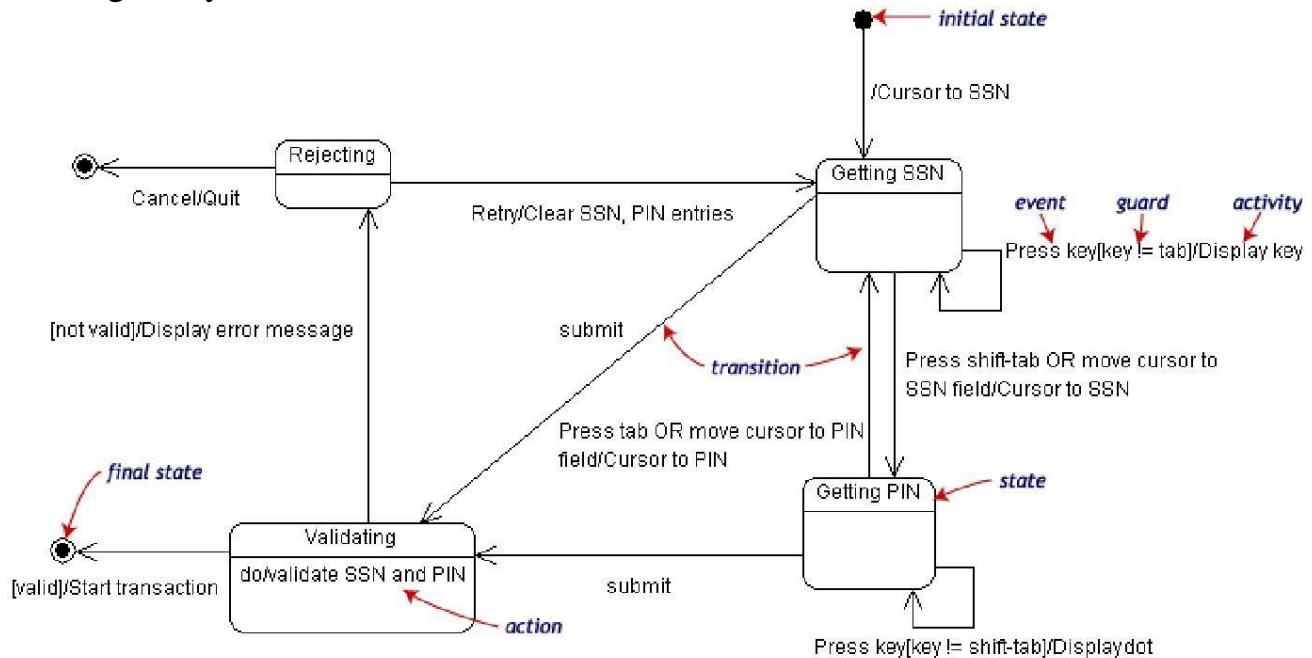
Class diagram: The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



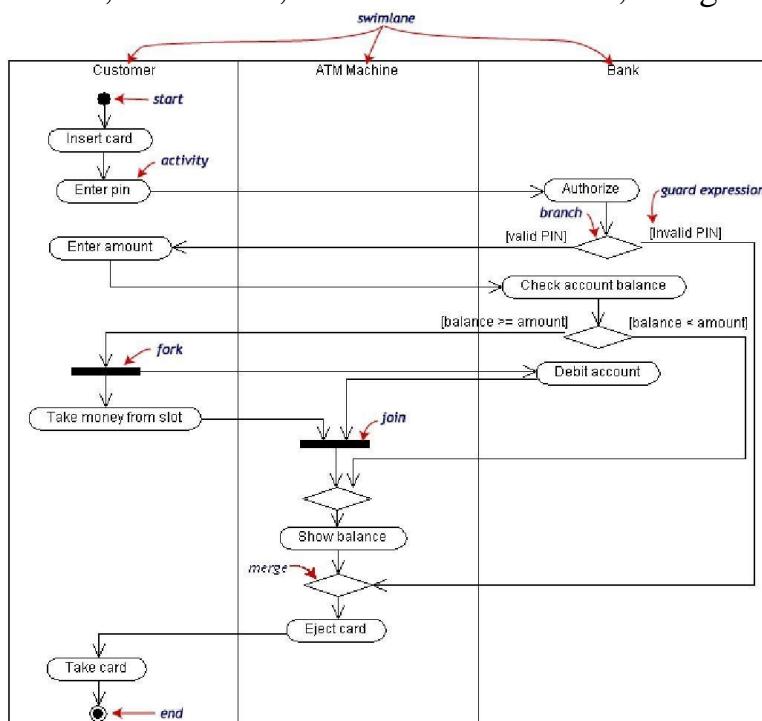
Object diagram: The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running. The object diagram captures the state of different classes in the system and their relationships or association at a given point of time.



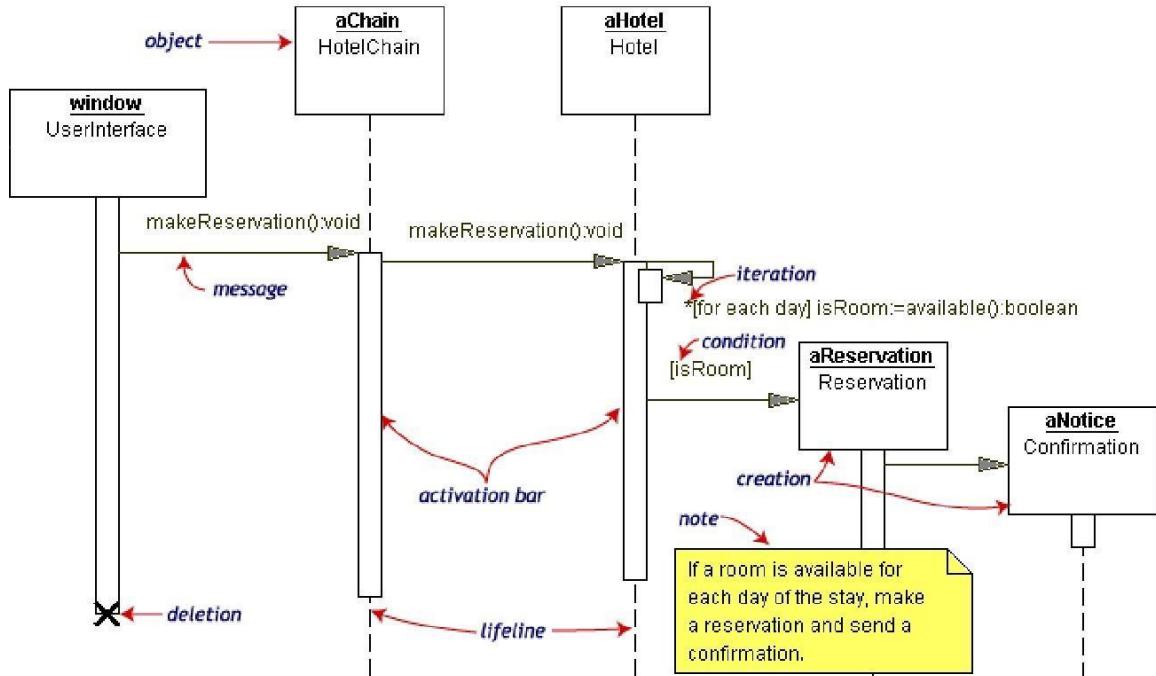
State diagram: A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.



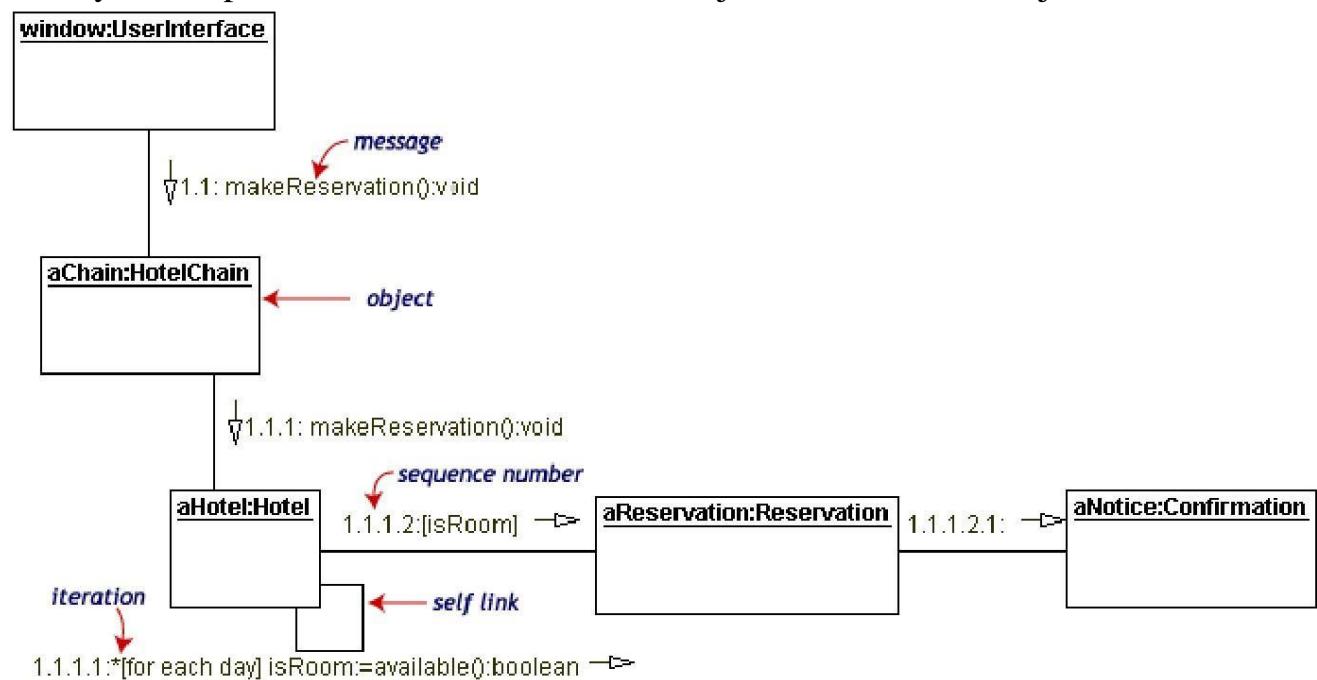
Activity diagram: The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.



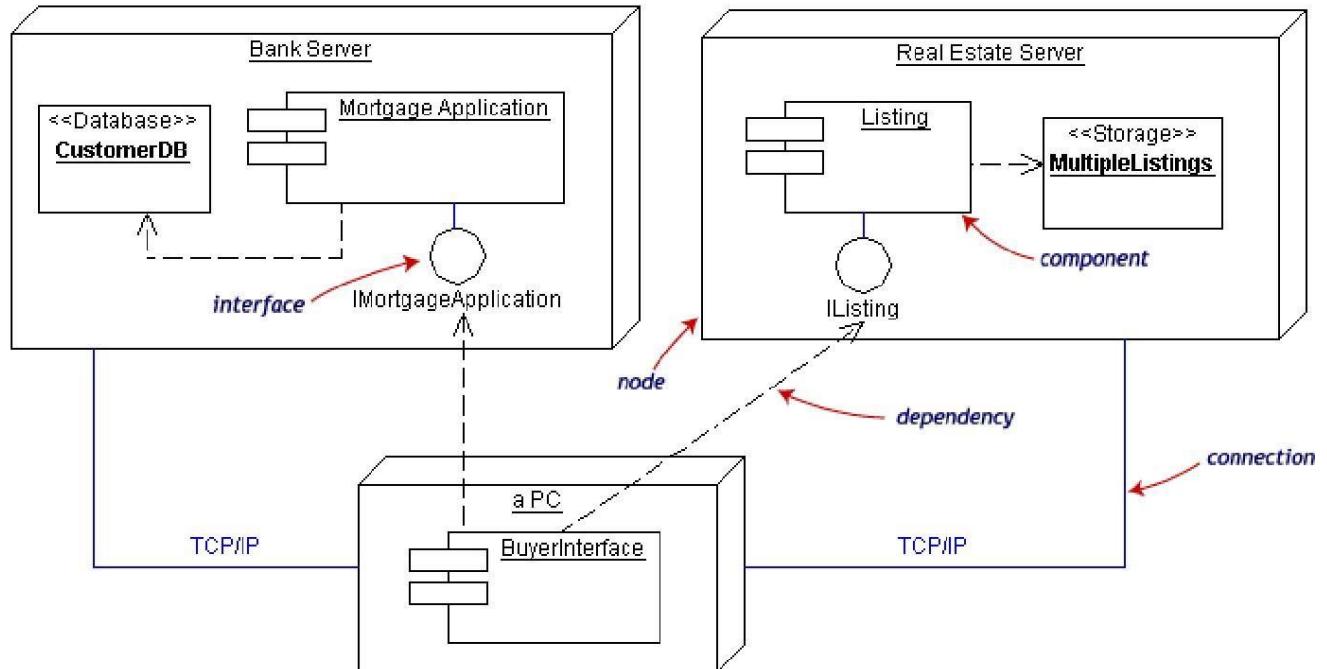
Sequence diagram: A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing messages.



Collaboration diagram: A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.



Component diagram: The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



Deployment diagram: The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

RESULT:

Thus the Various UML diagrams and their types are studied.

Expt No: 2	
Date:	PASSPORT AUTOMATION SYSTEM

AIM:

To analyze and Design UML Diagrams for Passport Automation System using Umbrello software.

1. SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux

Case Tool: Umbrello

2. ANALYSIS:

2.1 Identifying Actors:

User

Admin

2.2 Identifying Use Cases:

Fill Applicant Details

Login

Apply Passport

Issue Passport

Cancellation

Verify Applicant
Details

Check Status

Logout

2.3 Identifying Classes:

Login

Apply Passport

Registration

Cancel Passport

Check Status

2.4 Identifying Attributes:

Name

Last Name

Age

Sex

DOB

Qualification

Address

Date

Pan card No

Ration Card NO Father

Name

Contact No

2.5 Identifying Methods:

Login

Apply passport

Registration

Cancel passport

Check status

2.6 Identifying Relationship: Association **3. PROJECT SCOPE:**

The main scope of the project is to provide a Transparent, User-friendly, Scalable and Reliable Passport Automation System which would be beneficial for the citizens.

4. OBJECTIVE:

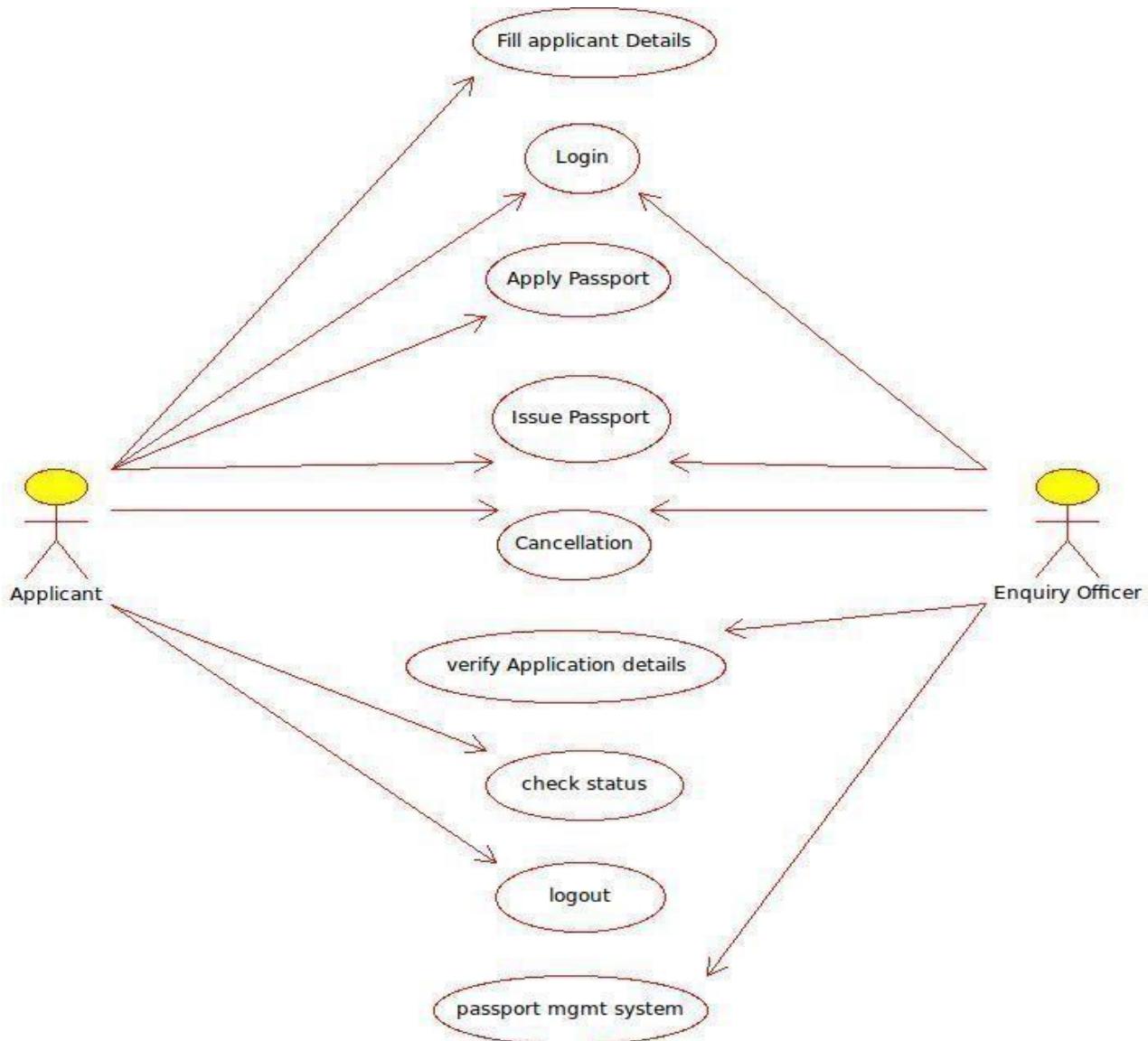
The Ultimate aim of this Passport Automation System is to provide with a system which proves to be manually beneficial i.e., benefits the citizens of the country who wishes to apply Passport. This system depicts each and every activity involved in passport automation process.

5. PROBLEM STATEMENT:

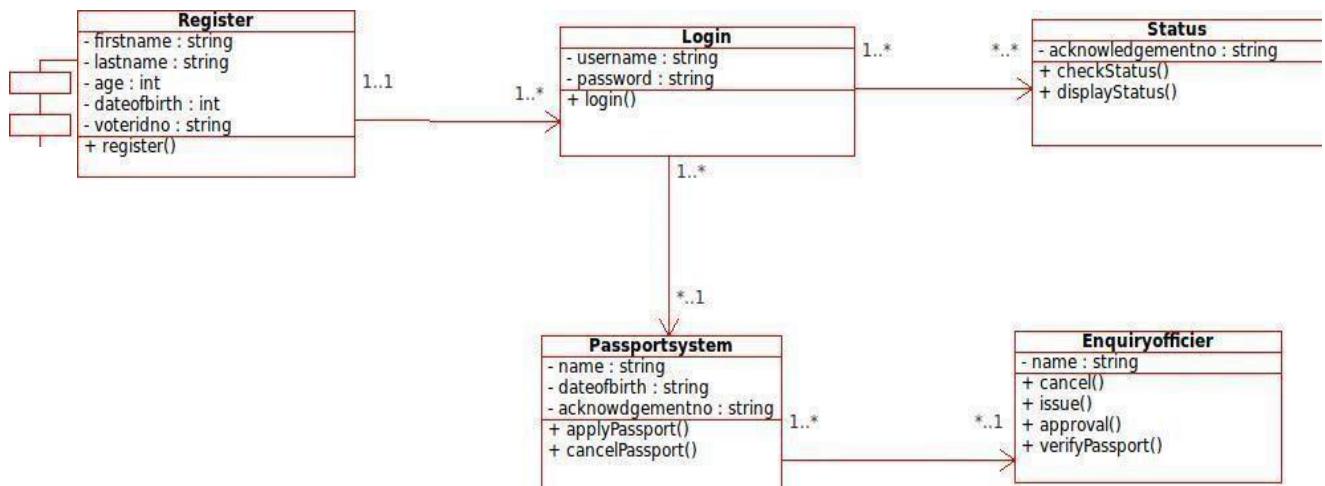
- 1 In this Passport Automation System, we assume ourself as citizen (Applicant) and proceed with the process steps as follows.
- 2 Applicant applies for Bonafide certificate in college and college issues the certificate to the applicant.
- 3 Applicant downloads the passport form either from an online server or by directly from the passport administrative office.
- 4 The Passport office issues the form stating rules & regulations, and the Applicant studies and fills the form and attaches the necessary photocopies and submits it.
- 5 Then the Passport administrator verifies the information found on the form with that of the original certificates and stores it in database.
- 6 The police receives intimation from the Passport office to perform a personal verification of that Applicant.
- 7 The police enquires and sends acknowledgement to the Passport administrative office.
- 8 The passport administrator calls up the Applicant and then issues the passport.

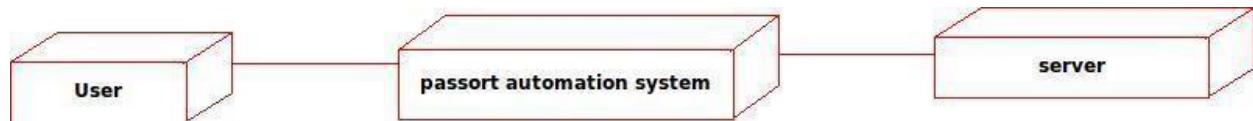
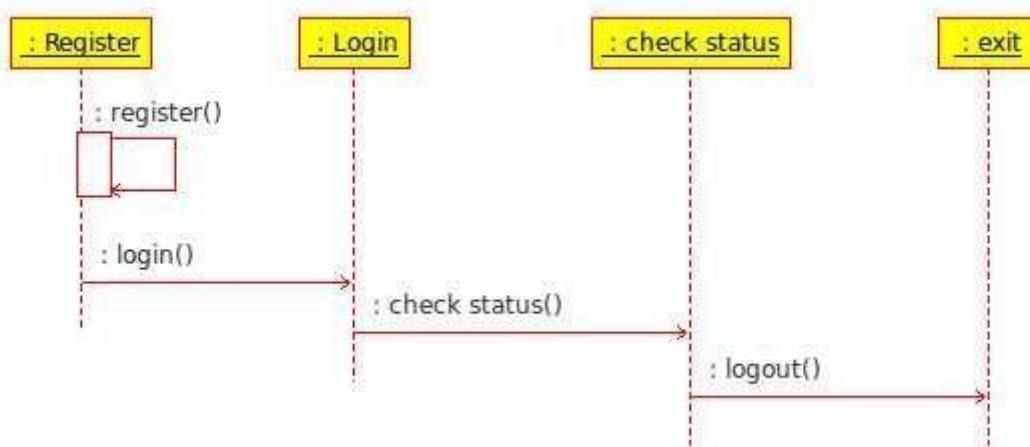
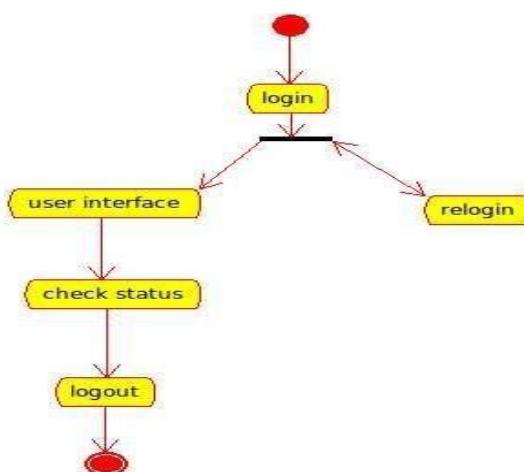
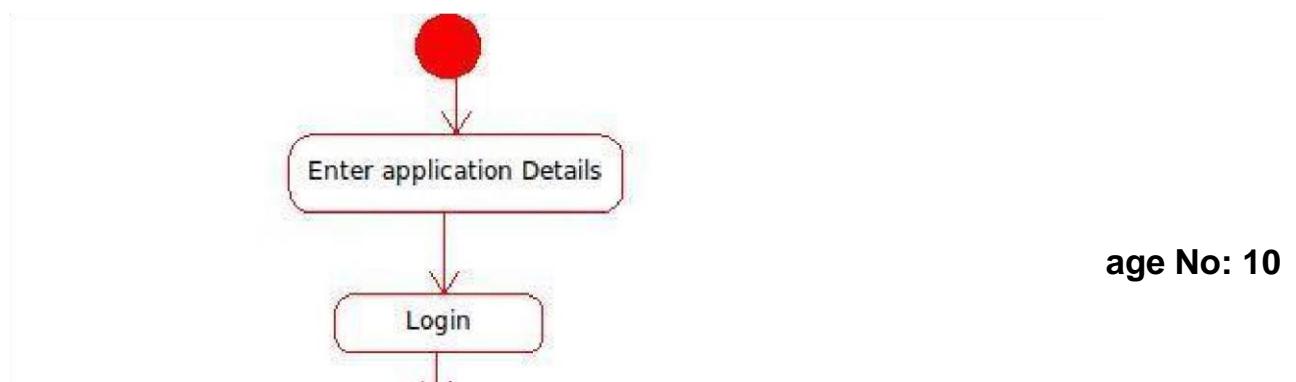
The Applicant collects the passport and if he wants to renew it, he can follow the same steps as before.

USE CASE DIAGRAM



CLASS DIAGRAM COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM**SEQUENCE DIAGRAM****STATE CHART DIAGRAM****ACTIVITY DIAGRAM****OUTPUT:**

Login.java

```
/***
 * Class Login
 */
public class Login {

    // Fields
    private String
        username;
    private String password;

    // Constructors
    public Login () { };

    // Methods
    // Accessor methods

    /**
     * Set the value of username
     * @param newVar the new value of username
     */
    private void setUsername ( String
        newVar ) {
        username = newVar;
    }

    /**
     * Get the value of username
     * @return the value of username
     */
    private String getUsername (
    ) {
        return username;
    }

    /**
     * Set the value of password
     * @param newVar the new value of password
     */
    private void setPassword ( String
        newVar ) {
        password = newVar;
    }
}
```

```
* Get the value of password  
* @return the value of password  
*/ private String getPassword (  
) {  
    return password;  
} // Other
```

methods

```
public void login( )  
{  
}  
}
```

RESULT:

Thus the “Passport Automation system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 3	
Date:	BOOK BANK SYSTEM

AIM:

To analyze and Design UML Diagrams for Book bank system using Umbrello software

1.SOFTWARE REQUIREMENT SPECIFICATION:**SYSTEM REQUIREMENT:**

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux

Tool: Umbrello

2.ANALYSIS:**2.1 Identifying Actors:**

Passenger

Admin

2.2 identifying use cases

Login

issue ticket

Check available

paid

Enter the website

print ticket

modify

cancelling ticket

ticket detail

logout

2.3 Identifying Classes:

Login

Apply ticket

Registration

Cancel ticket

Check status

2.4 Identifying Attributes:

Name

Address

Identification marks

Age

Father name

Last name

Contact no

Sex

DOB

2.5Identifying Methods:

REGISTER NO:110819104501

NAME: MUKESH.V

Login
Check statusS
Apply ticket

Get applicant details

Cancel ticket

2.6Identifying Relationship:

Association

3.PROJECT SCOPE:

The main scope of the project is to provide a exhaustive, flexible and reliable Book bank system which would be beneficial for both librarian and members.

4.OBJECTIVE:

The main objective of designing and developing a Book bank system is to provide with a system which proves to be manually beneficial i.e., benefits both the librarian and the member. The complete activities and the process right from buying of books and issuing it to members, every single activity is incorporated in this system.

5.PROBLEM STATEMENT:

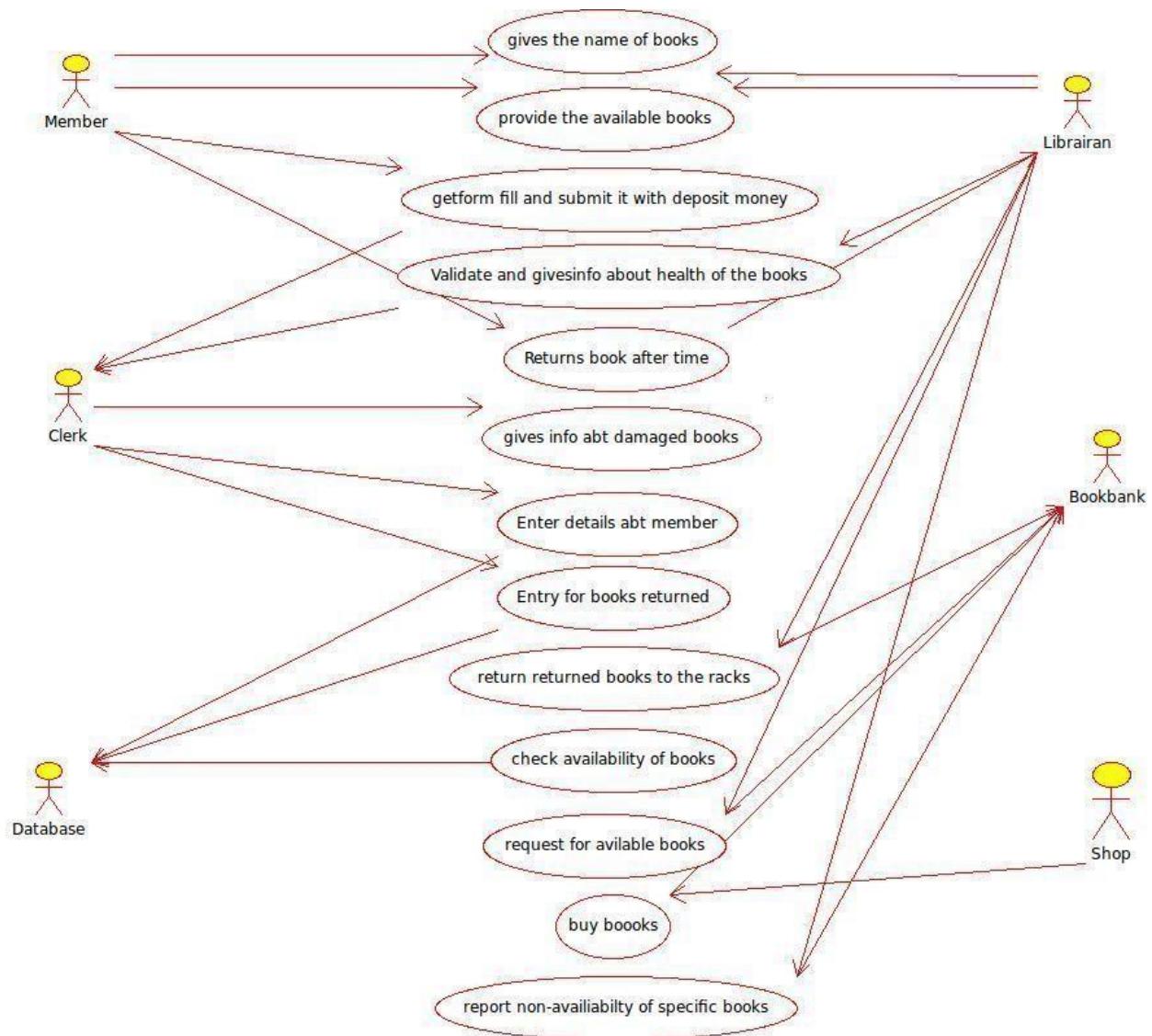
Book banks are used to provide books for amount. In our proposed the various problem statement are as follows,

1. Initially book bank buys books from book shops and provides form to members .The clerk makes entry of members and books to the database. member will give the details of books that he wants.

2.Librarian will checks the availability of books in database which contains information about members and books .If the book is available means he will issue the books else he will give the report of non-available books to book bank. The book bank will buy those books from book shops and makes entry into database.

3. When the member returns the book the clerk will check whether the book has been returned on or before due date and its condition, if he finds out any fault he will collect from member and makes entry into database.

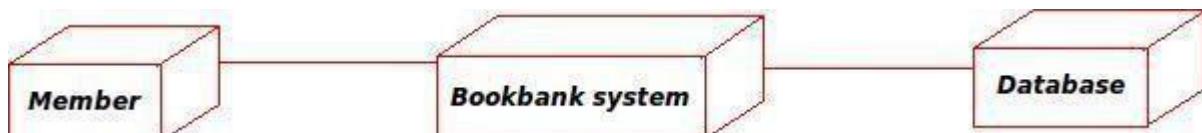
USE CASE DIAGRAM



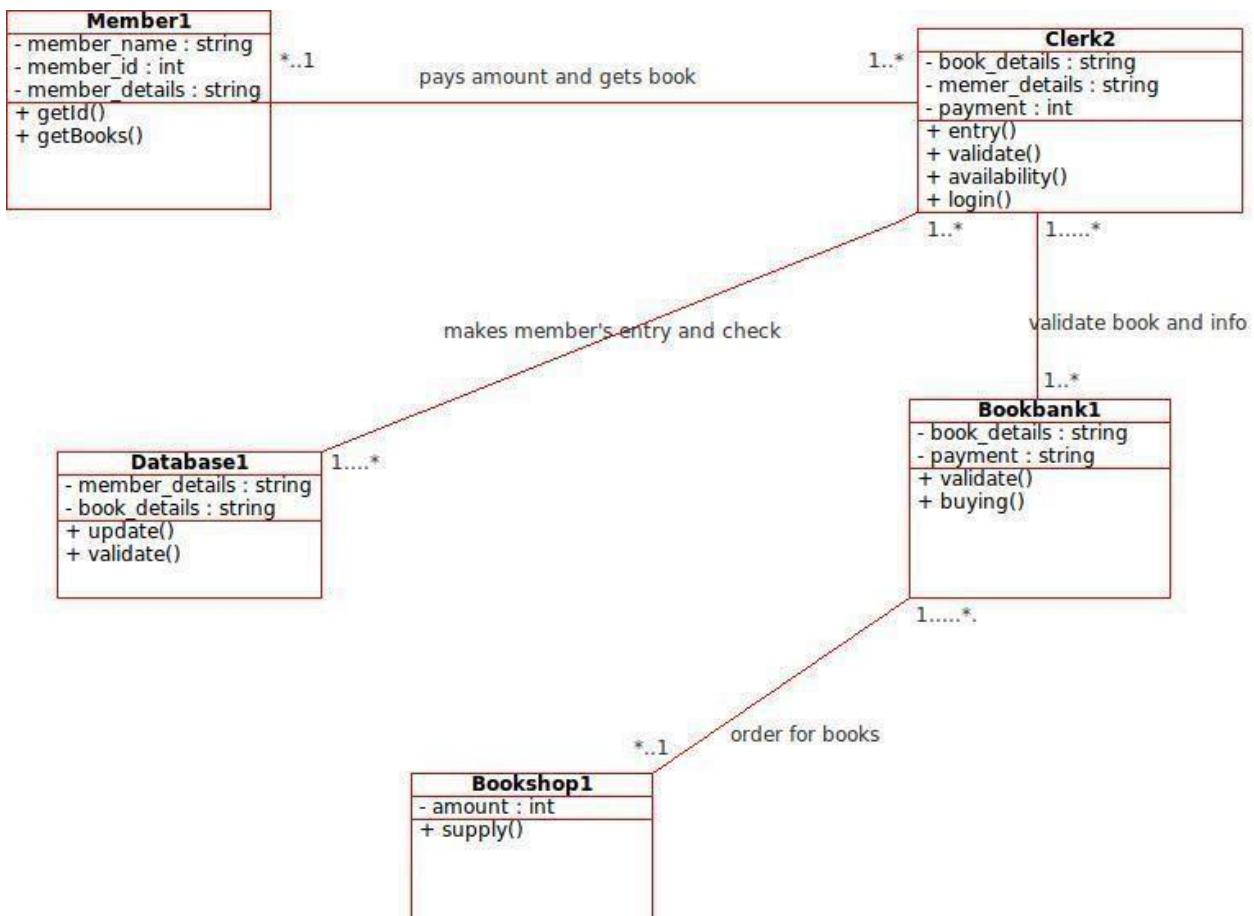
COMPONENT DIAGRAM



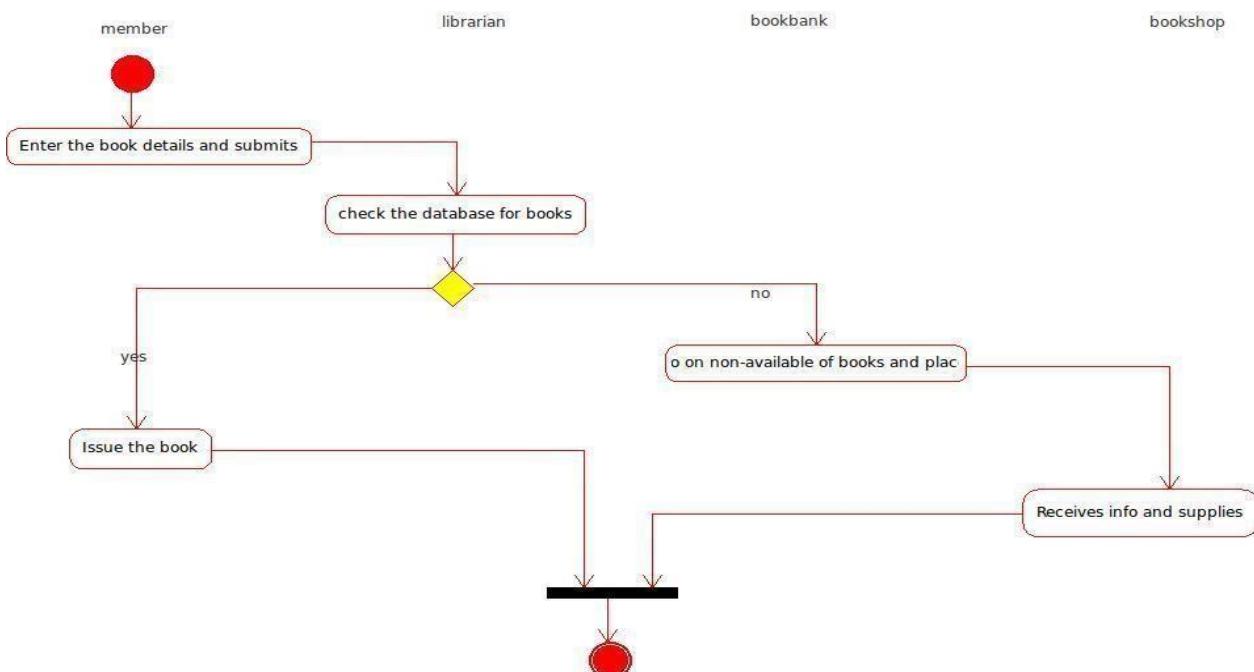
DEPLOYMENT DIAGRAM



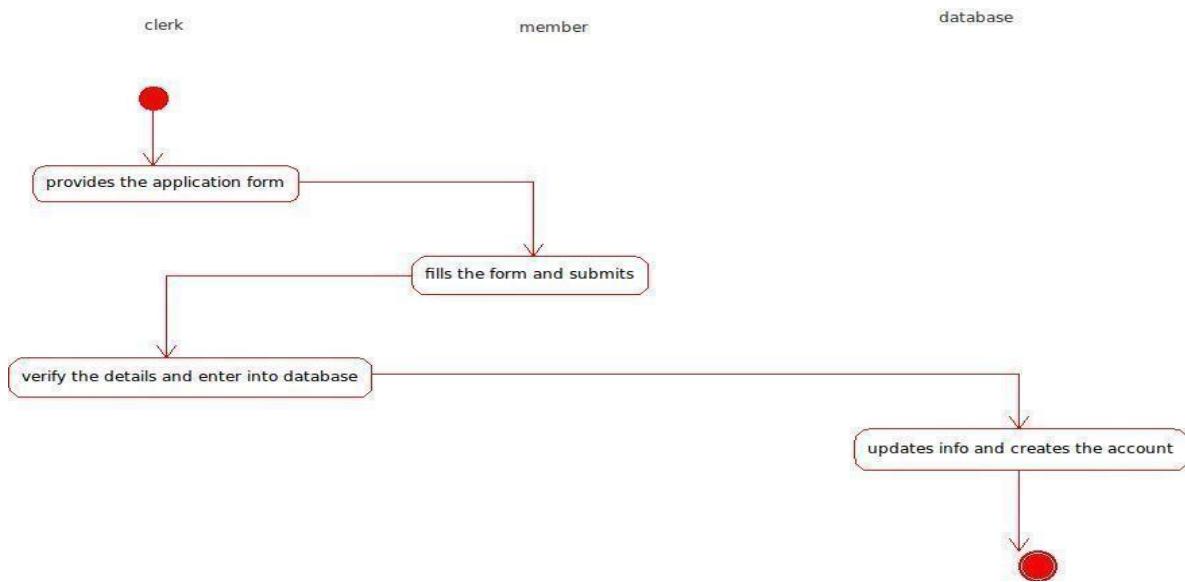
CLASS DIAGRAM



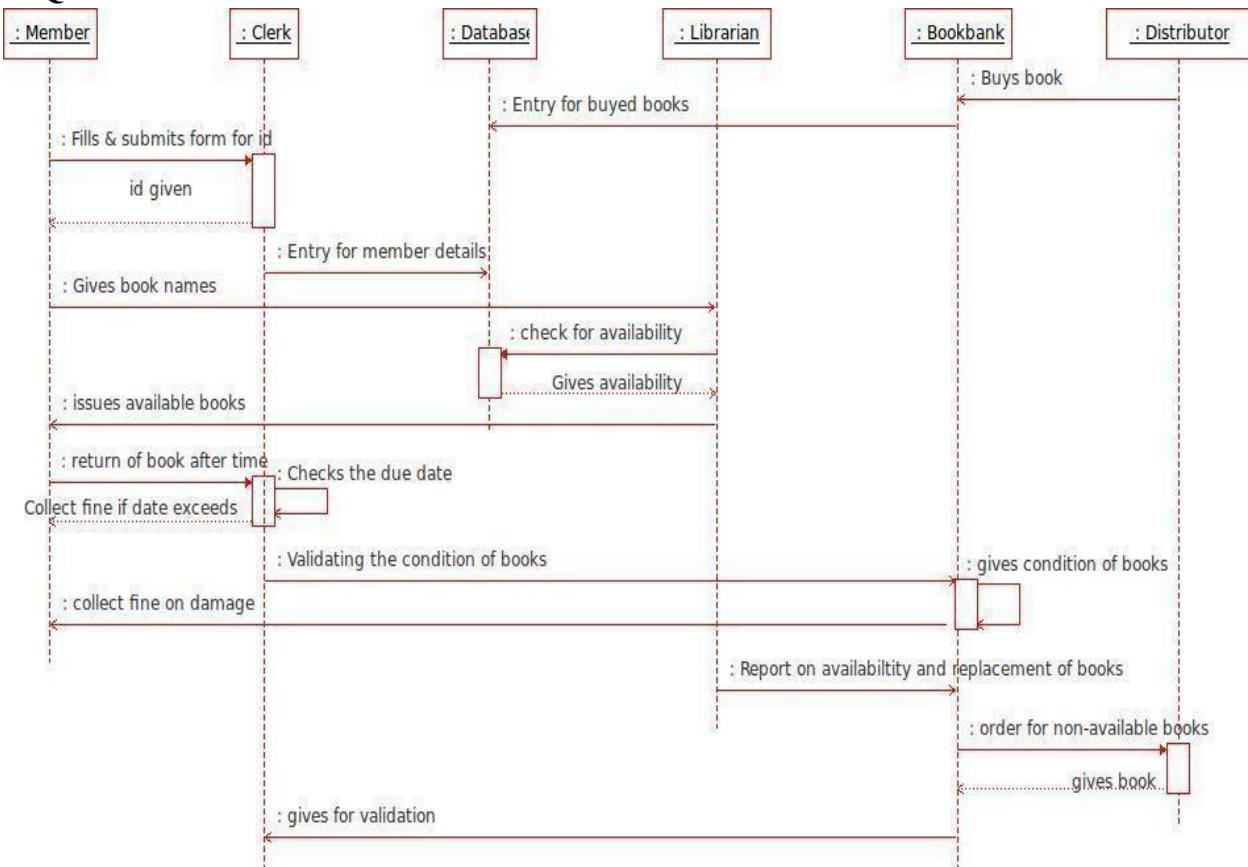
ACTIVITY DIAGRAM FOR MEMBERS IN DATABASE



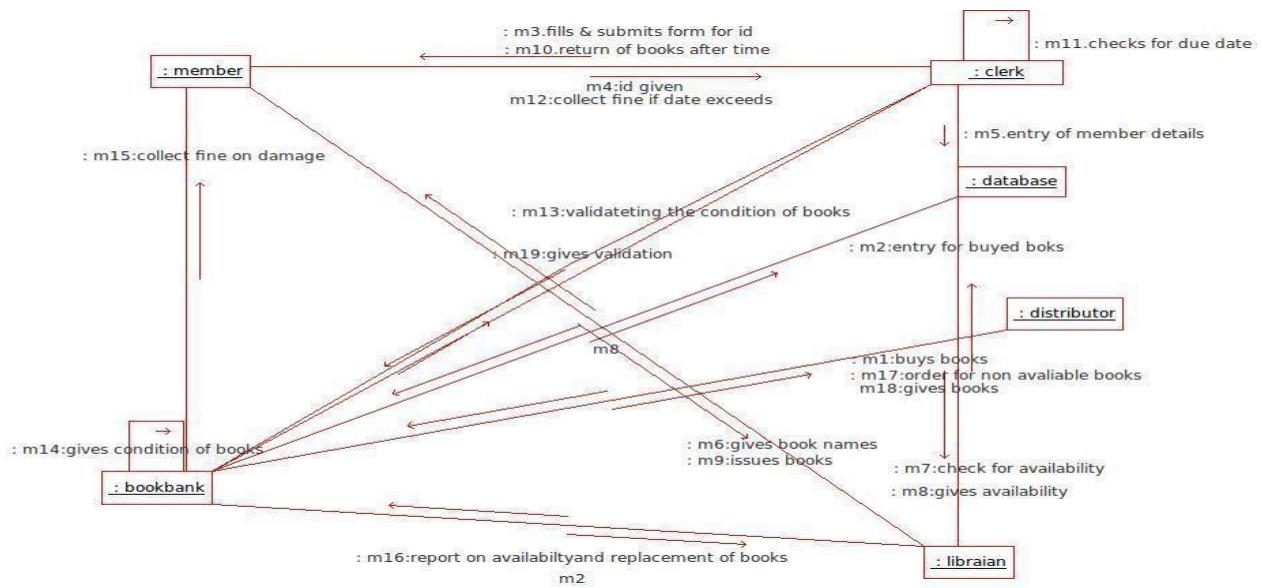
ACTIVITY DIAGRAM FOR BOOK CHECKING



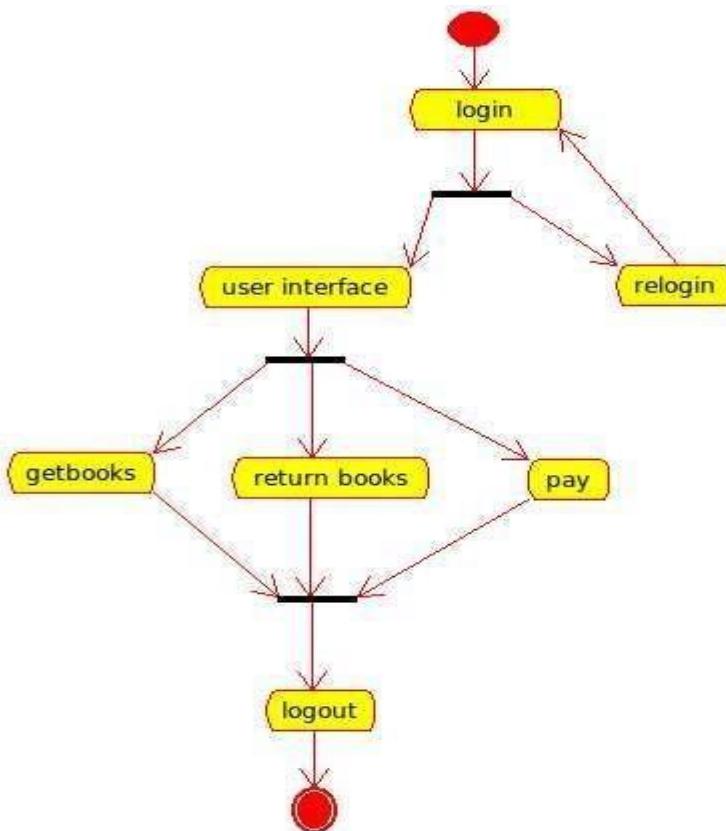
SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



STATE CHART DIAGRAM



OUTPUT:

Bookshop1.java

```
import java.util.*;  
/**  
 * Class Bookshop1  
 */  
public class Bookshop1 {  
    //  
    // Fields  
    //  
    private int amount;  
    //  
    // Constructors  
    //  
    public Bookshop1 () { };  
    //  
    // Accessor methods  
    //  
    /**  
     * Set the value of amount  
     * @param newVar the new value of amount  
     */  
    private void setAmount ( int newVar ) {  
        amount = newVar;  
    }  
    private int getAmount ( ) {  
        return amount;  
    }  
    // Other methodS  
    /**  
     */  
    public void supply( )  
    {  
    } }
```

RESULT:

Thus the “Bookbank system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 4	
Date:	
EXAM REGISTRATION SYSTEM	

AIM:

To analyze and Design UML Diagrams for exam registration system using Umbrello Software.

SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

ANALYSIS:

2.1 Identifying Actors:

Member

Shop

Librarian

Clerk

Book bank

2. Identifying Use Cases:

Give the name of book

Provide the available books

Give info about damaged books

Enter details about member

Buy the book

Return the book

3. Identifying Classes:

Member

Clerk

Book bank

Book shop

4. Identifying Attributes:

Member name

Member id

Member detail

Book detail

Amount

payment

5. Identifying Methods:

Get id
Entry

Get book
ValidateLogin

Update

Buying supply

Identifying Relationship:

Association

PROJECT SCOPE:

The main scope of this project is to maintain a student details with secure and directly maintain by both college and university. students can get the hall ticket with correct details without any irrelavant datas.

OBJECTIVE

The main objective of designing and developing a exam registration system is to provide with a system which proves to be manually beneficial i.e benefits for both university and college for easy maintaining.

PROBLEM STATEMENT:

In exam registration proposed system various problem statement are as follows, Initially university provides the exam details with fee structure to the respected college.

Then the college ask the students for them details and fee submission conditions.

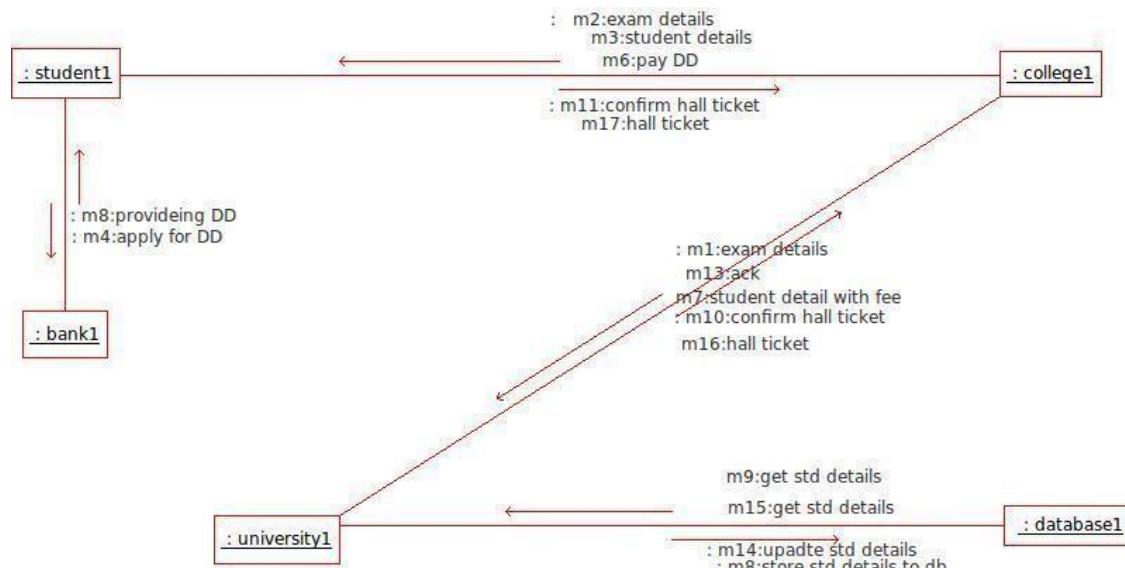
Student contact the various nationalized banks for getting DD and submit them details with the DD in the corresponding date.

When the students submit them details college will forward that to the universtiy

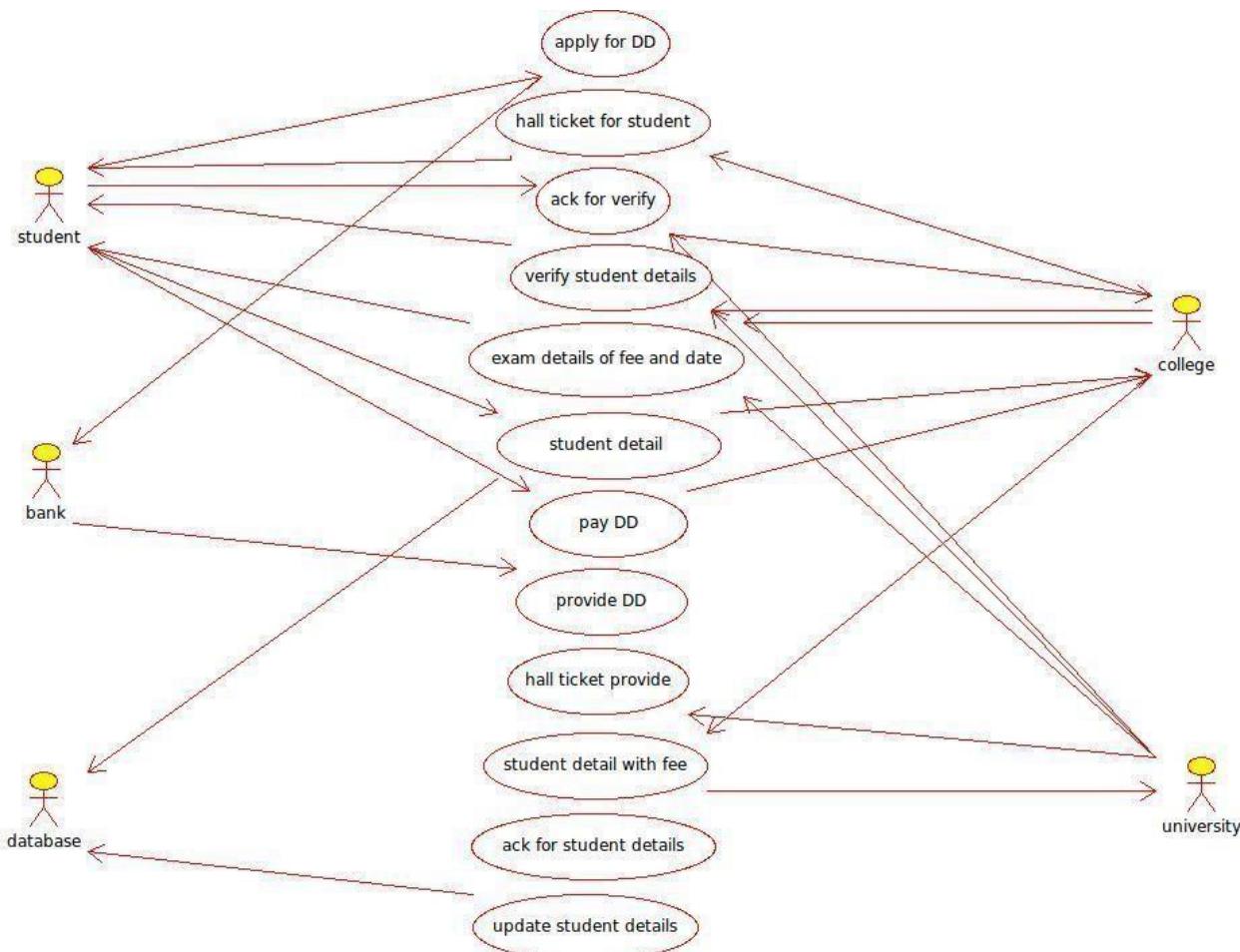
University maintaines one database and store the student details with year, semester,subjects

etc., With the valid datas university prepare the hall ticket to related college After the related colleges provide hall tickets to the students. If any problem occur in the hall ticket means students can apply for change and it will quickly done by the university.

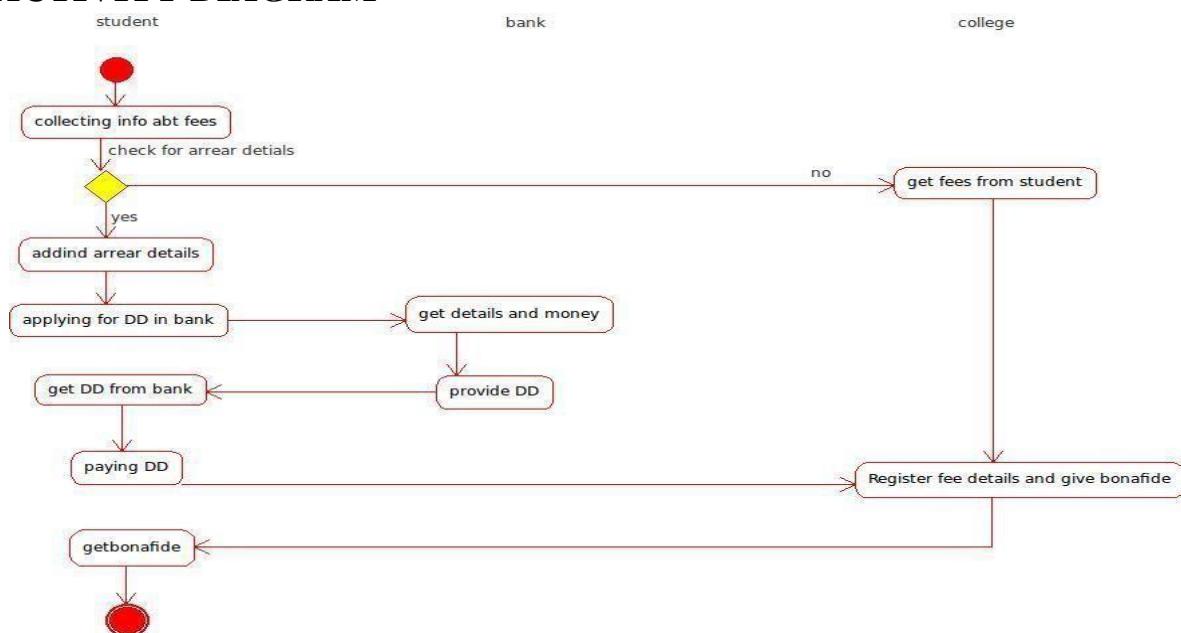
COLLABORATION DIAGRAM



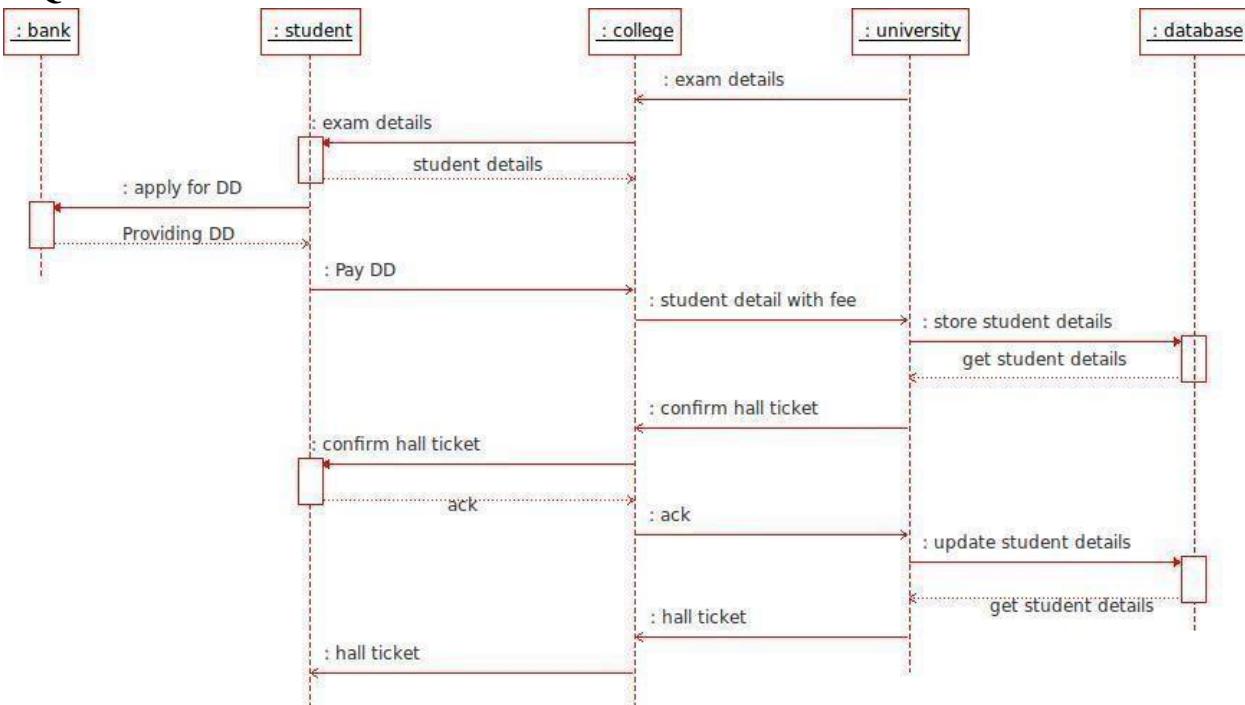
USECASE DIAGRAM



ACTIVITY DIAGRAM



SEQUENCE DIAGRAM



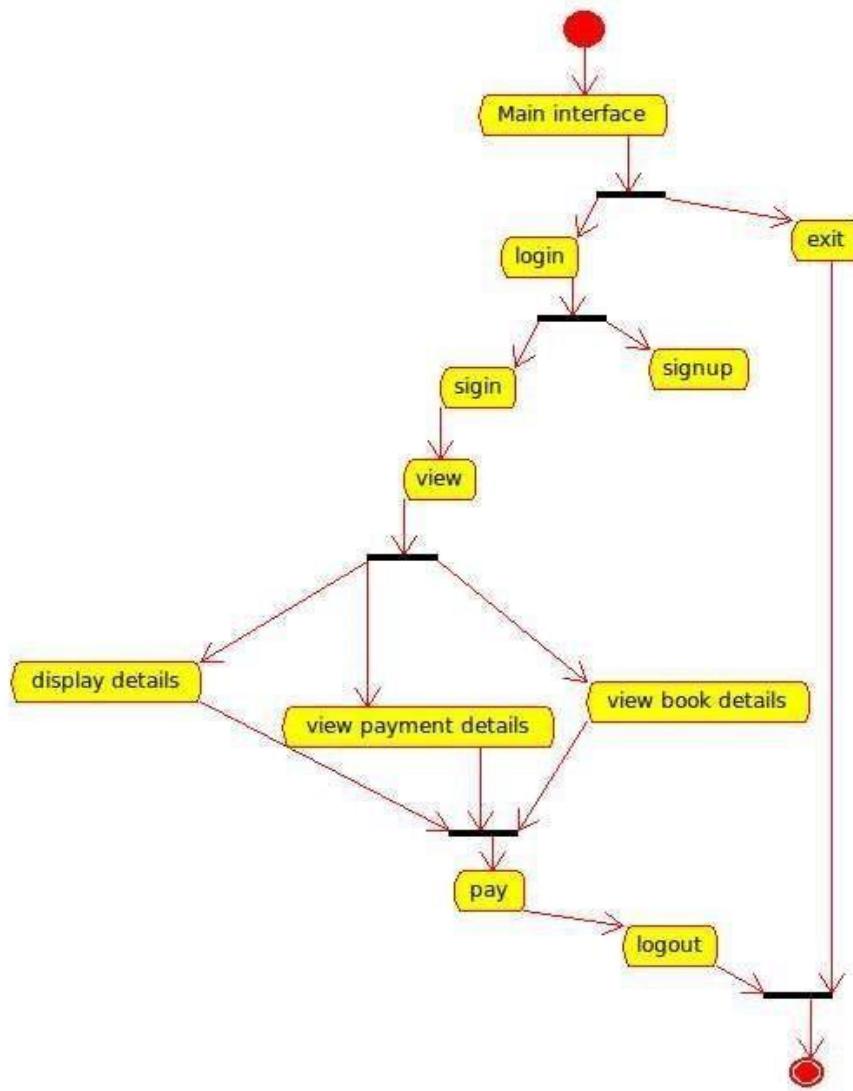
COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



STATE CHART DIAGRAM



OUTPUT

BANK.java

```

/**
 * Class Bank
 */
public class Bank
{
    // Fields
    private
    String name;
    private
    String addr;
    private
    String branch;
    // Constructors
    public
    Bank () { };
    // Methods
    // Accessor methods
    /**
     * Set the value of name
     * @param newVar the new value of name

```

```

 */ private void setName ( String
newVar ) {
    name = newVar;
}
/***
 * Get the value of name
 * @return the value of name
 */ private String getName (
) { return
    name;
}
/***
 * Set the value of addr
 * @param newVar the new value of addr
 */ private void setAddr ( String
newVar ) {
    addr = newVar;
}
/***
 * Get the value of addr
 * @return the value of addr
 */ private String getAddr (
) { return addr;
}
/***
 * Set the value of branch
 * @param newVar the new value of branch
 */ private void setBranch ( String
newVar ) {
    branch = newVar;
}
/***
 * Get the value of branch
 * @return the value of branch
 */ private String getBranch (
) { return
    branch;
}
// Other methods public
void getMoney( )

```

```

{
}

public void provideDD( )

{
} public class Student
{ // Fields private
String stuname; private
int stuid; //

Constructors
public Student () { } // Methods
// Accessor methods
/***
 * Set the value of stuname
 * @param newVar the new value of stuname
 */ private void setStuname ( String
newVar ) {
    stuname = newVar;
}
/***
 * Get the value of stuname
 * @return the value of stuname
 */ private String getStuname (
) {
    return stuname;
}
/***
 * Set the value of stuid
 * @param newVar the new value of stuid
 */ private void setStuid ( int
newVar ) {
    stuid = newVar;
}
/***
 * Get the value of stuid
 * @return the value of stuid
 */ private int getStuid (
) { return
    stuid;
}
// Other methods
public void provid_Details( )

```

```
{ } public void  
getExamdet( )  
{  
}  
public void getHallticket( )  
{  
} public void  
giveMoney( )  
{  
} public void  
getDD( )  
{  
}  
}
```

RESULT:

Thus the “Exam registration system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 5	
Date:	

STOCK MAINTENANCE SYSTEM

AIM:

To analyze and Design UML Diagrams for stock maintenance system using Umbrello software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

2.ANALYSIS:

2.1. Identifying Actors:

customer	database
stock manager	

2.2. Identifying Use Cases:

Check for avail	Deliver pdt
Gives order	Reject the order
Sufficient pdt	Update the DB
Insufficient pdt	

3. Identifying Classes:

Customer	company
Stock manger	
Database	

4. Identifying Attributes:

name	price
pdt id	phone no
pdt	no.of.avail
name address	

5. Identifying Methods:

demand	check avail
store	sales
update	producing

6. Identifying Relationship: Association

PROJECT SCOPE:

The main scope of the project is to provide a exhaustive, flexible and reliable stock maintenance system which would be beneficial for both stock manager and retailer

OBJECTIVE:

The main objective of designing and developing a stock maintenance system is to Provide with a system which proves to be manually beneficial i.e., benefits both the stock manager and the retailer. The complete activities and the process right from product's manufacture till the product gets sold out, every single activity is incorporated in this system.

PROBLEM STATEMENT:

INVENTORY SYSTEM is a real time application used in the merchant's day to day system. This is a database to store the transaction that takes places between the Manufacturer, Dealer and the Shop Keeper that includes stock inward and stock outward with reference to the dealer. Here we assume our self as the Dealer and proceed with the transaction as follows:

The Manufacturer is the producer of the items and it contains the necessary information of the item such as price per item, Date of manufacture, best before use, Number of tem available and their Company Address.

The Dealer is the secondary source of an Item and he purchases Item from the manufacturer by requesting the required Item with its corresponding Company Name and the Number of Items required. The Dealer is only responsible for distribution of the Item to the Retailers in the Town or City.

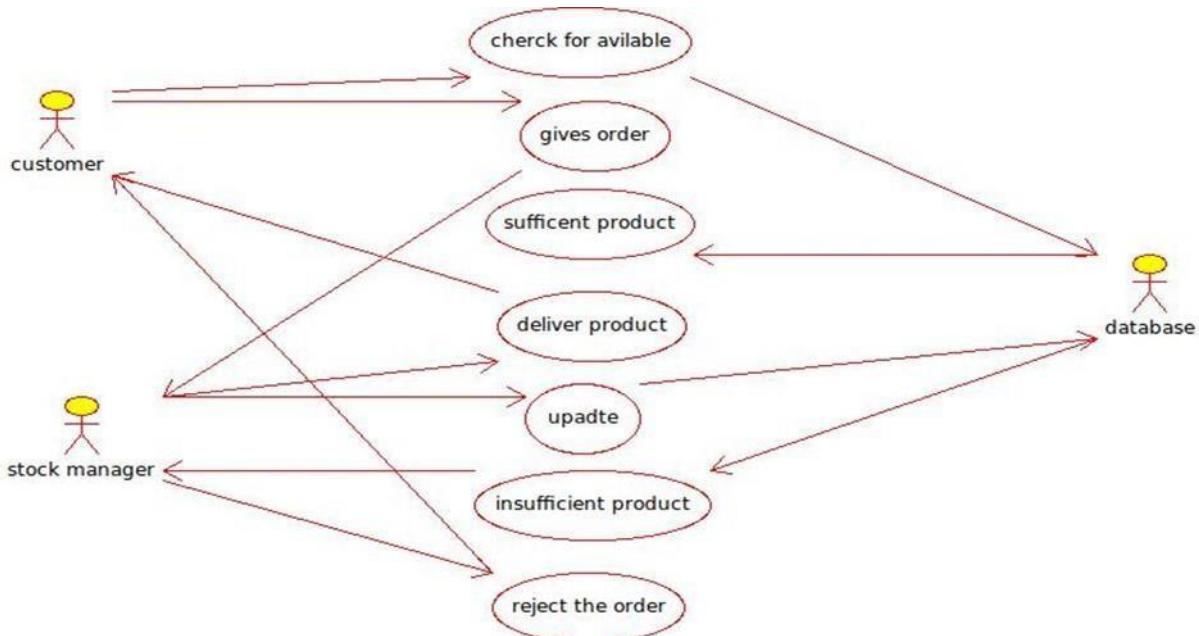
The Shop Keeper or Retailer is the one who is prime source for selling items in the market. The customers get Item from the Shop Keeper and not directly from the

CS8582 - OOAD LABORATORY

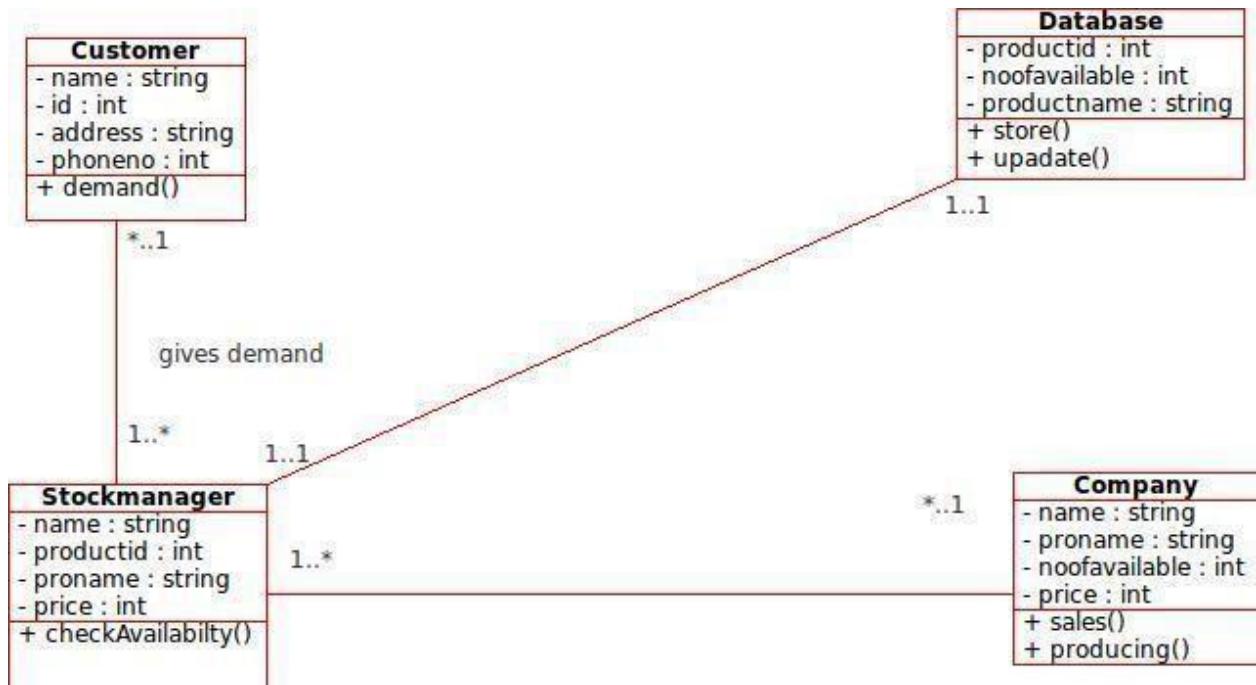
Manufacturer or the Dealer.

The Stock is the database used in our System which records all transactions that takes place between the Manufacturer and the Dealer and the Dealer and the Retailer.

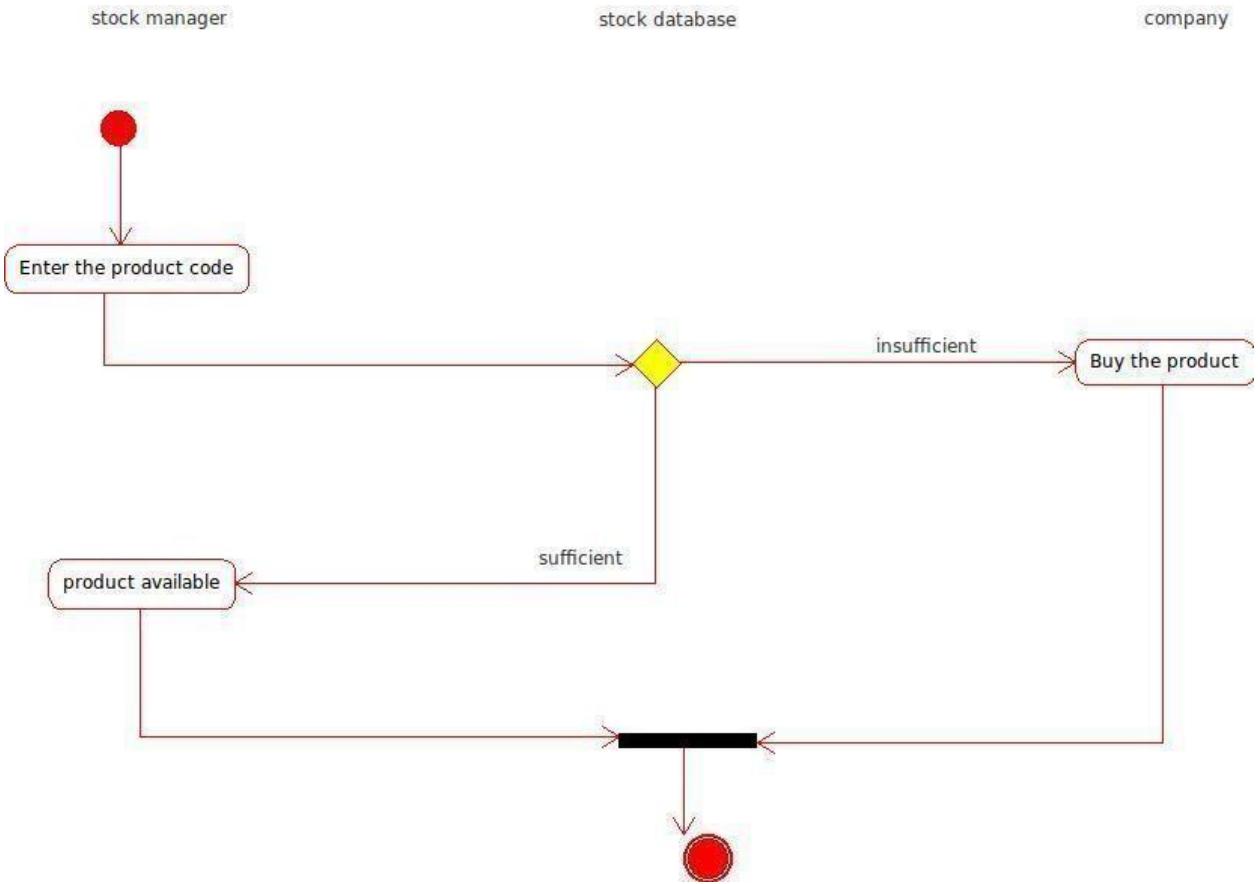
USE CASE DIAGRAM



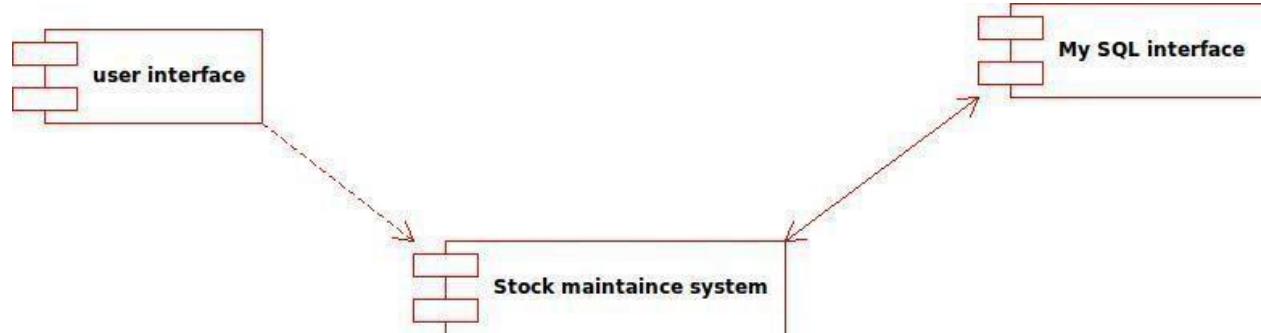
CLASS DIAGRAM



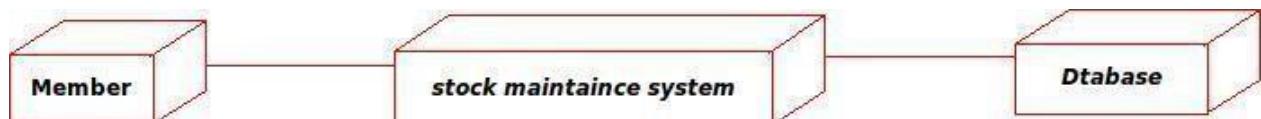
ACTIVITY DIAGRAM FOR GET ORDER FROM CUSTOMER



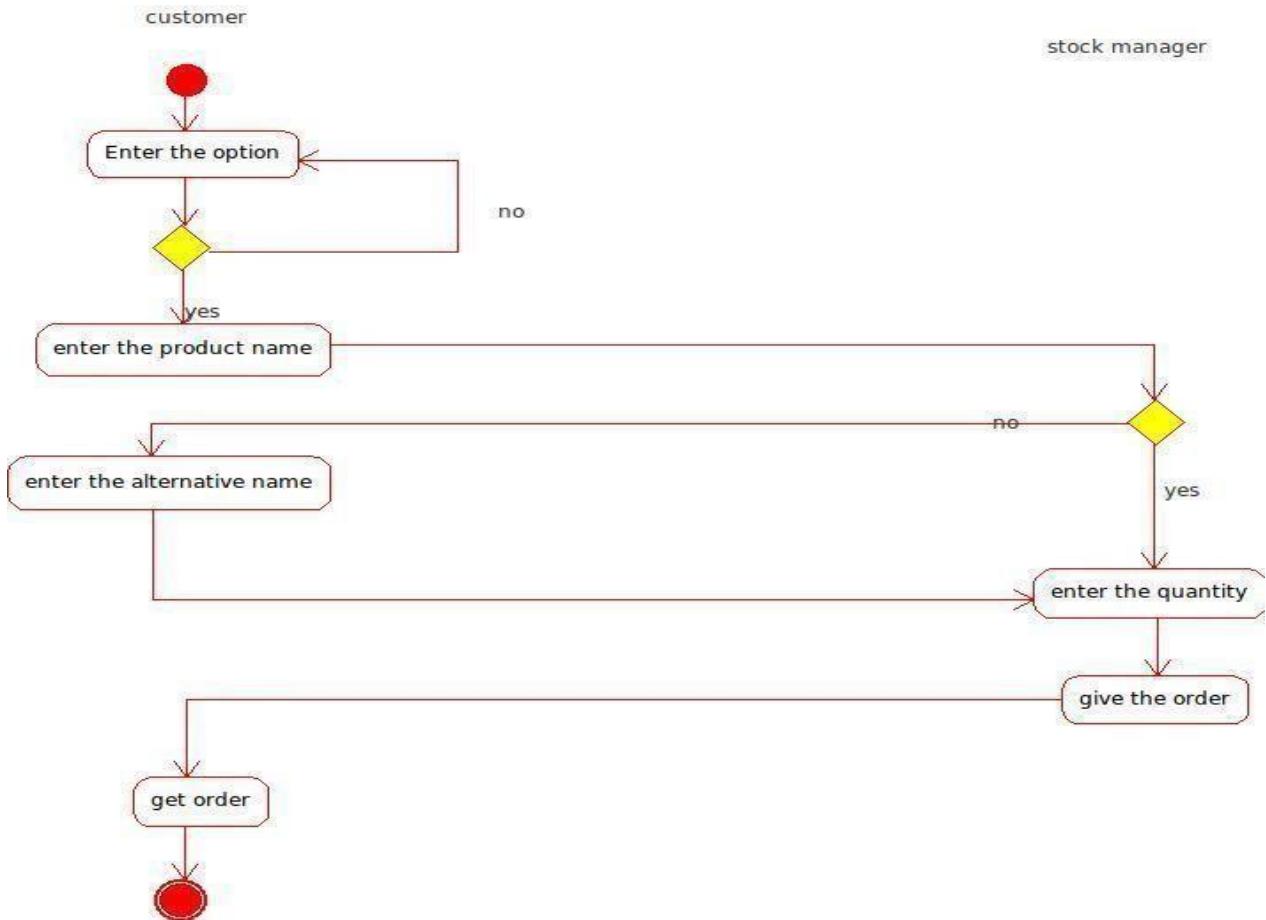
COMPONENT DIAGRAM



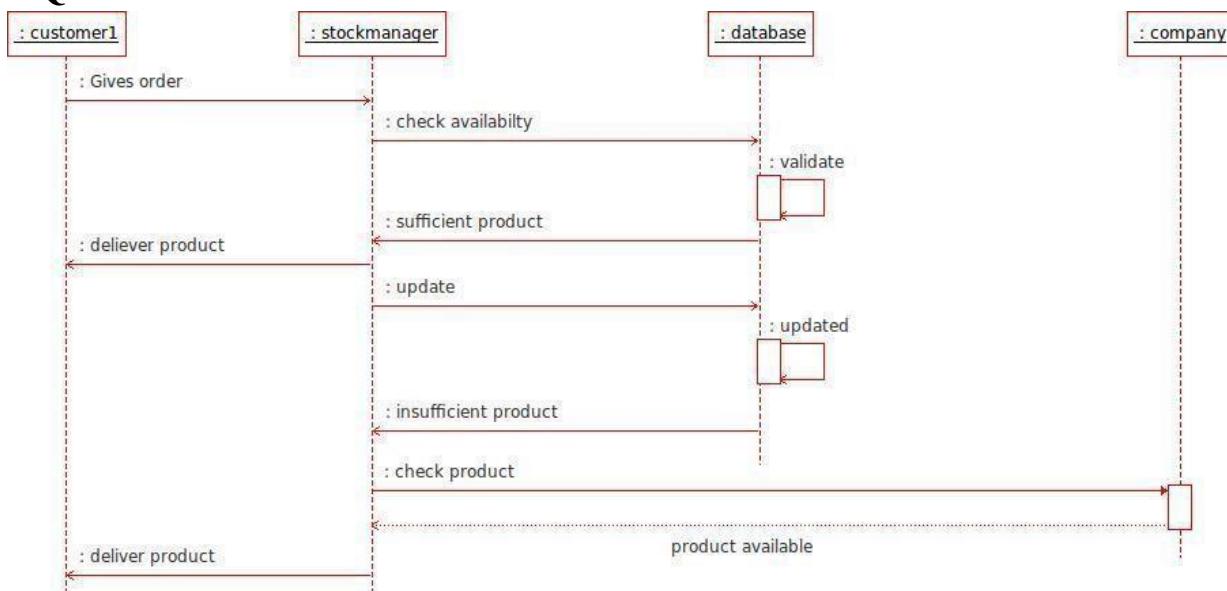
DEPLOYMENT DIAGRAM



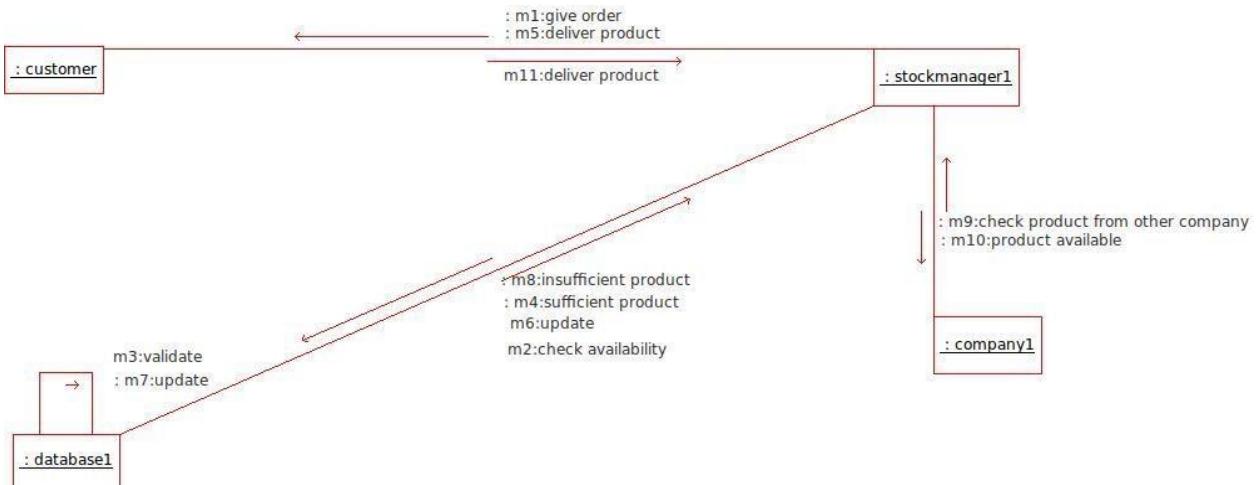
ACTIVITY DIAGRAM



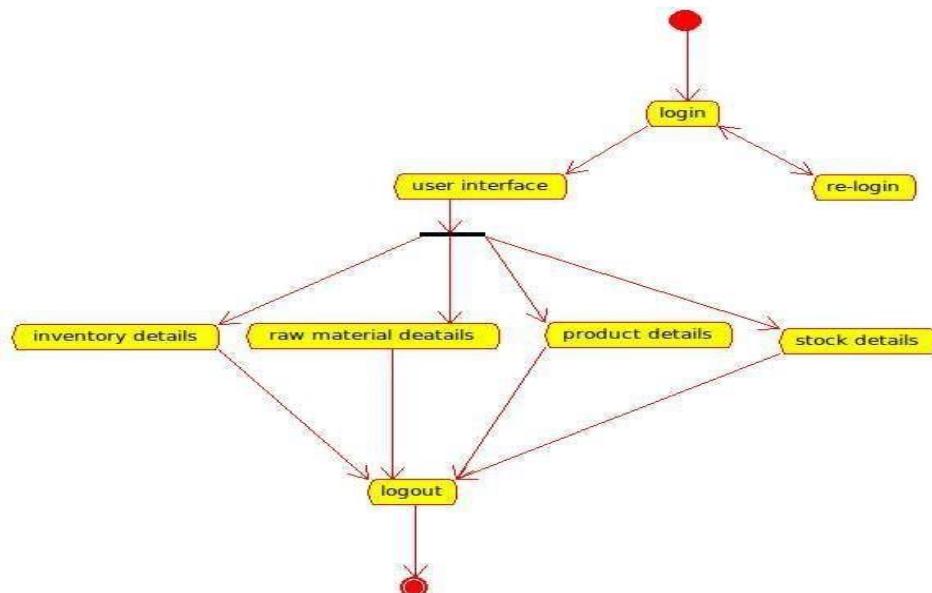
SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



STATE CHART DIAGRAM



OUTPUT:

Bank.java

```

/**
 * Class Bank*/
public class Bank { // Fields private String name; private String
addr; private String branch; // Constructors public Bank () { }; // Methods // Accessor
methods
/**
 * Set the value of name
 * @param newVar the new value of name
 */ private void setName ( String
newVar ) {
    name = newVar;
}
/**
 * Get the value of name
  
```

```

*   @return the value of name
*/ private String getName (
) { return
    name;
}
/***
*   Set the value of addr
*   @param newVar the new value of addr
*/ private void setAddr ( String
newVar ) {
    addr = newVar;
}
/***
*   Get the value of addr
*   @return the value of addr
*/ private String getAddr (
) { return
    addr;
}
/***
*   Set the value of branch
*   @param newVar the new value of branch
*/ private void setBranch ( String
newVar ) {
    branch = newVar;
}
/***
*   Get the value of branch
*   @return the value of branch
*/ private String getBranch (
) { return
    branch;
}
// Other methods
/***
*/ public void
getMoney( )
{
}
/** */ public void
provideDD( )
{
}

```

}

RESULT:

Thus the “Stock maintenance system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 6	
Date:	
ONLINE COURSE RESERVATION SYSTEM	

AIM:

To analysis and design UML Diagrams for Online Course Reservation System using Umbrello software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux

Case Tool: Umbrello

2. ANALYSIS:

2.1. Identifying Actors:

student university

college database bank

2.2. Identifying Use Cases:

hall ticket for stud

providing DD ack

for verify hall

ticket provide

verify stud detail

stud detail with fee

university

exam detail of fee and date

apply for DD

update stud detail

pay DD

2.3. Identifying Classes:

bank university

database college

student

2.4. Identifying Attributes:

stud name univ

name stud ID

collage name

bank name

college ID

bank address

2.5. Identifying Methods:

Provid_details

Get details

Get_exam detail

Update

Get_hall ticket

Provide detail

Give money

Get_DD

2.6. Identifying Relationship: Association**3.PROJECT SCOPE:**

The main scope of the project is to develop a Course Registration System, to Enquire the available courses using enquiry model and then register the required course using registration model, if the candidate want to cancel the course using cancellation model, if the candidate wants to modify or change the course, a modification model has been developed.

4.OBJECTIVE:

The main objective of designing and developing the course reservation system is to make the reservation and other related processes of this system more flexible and easier for the students who applied for the Anna University counseling.

5.PROBLEM STATEMENT:

Course reservation system is helpful for the student to reserve the course the problem statements are as follows,

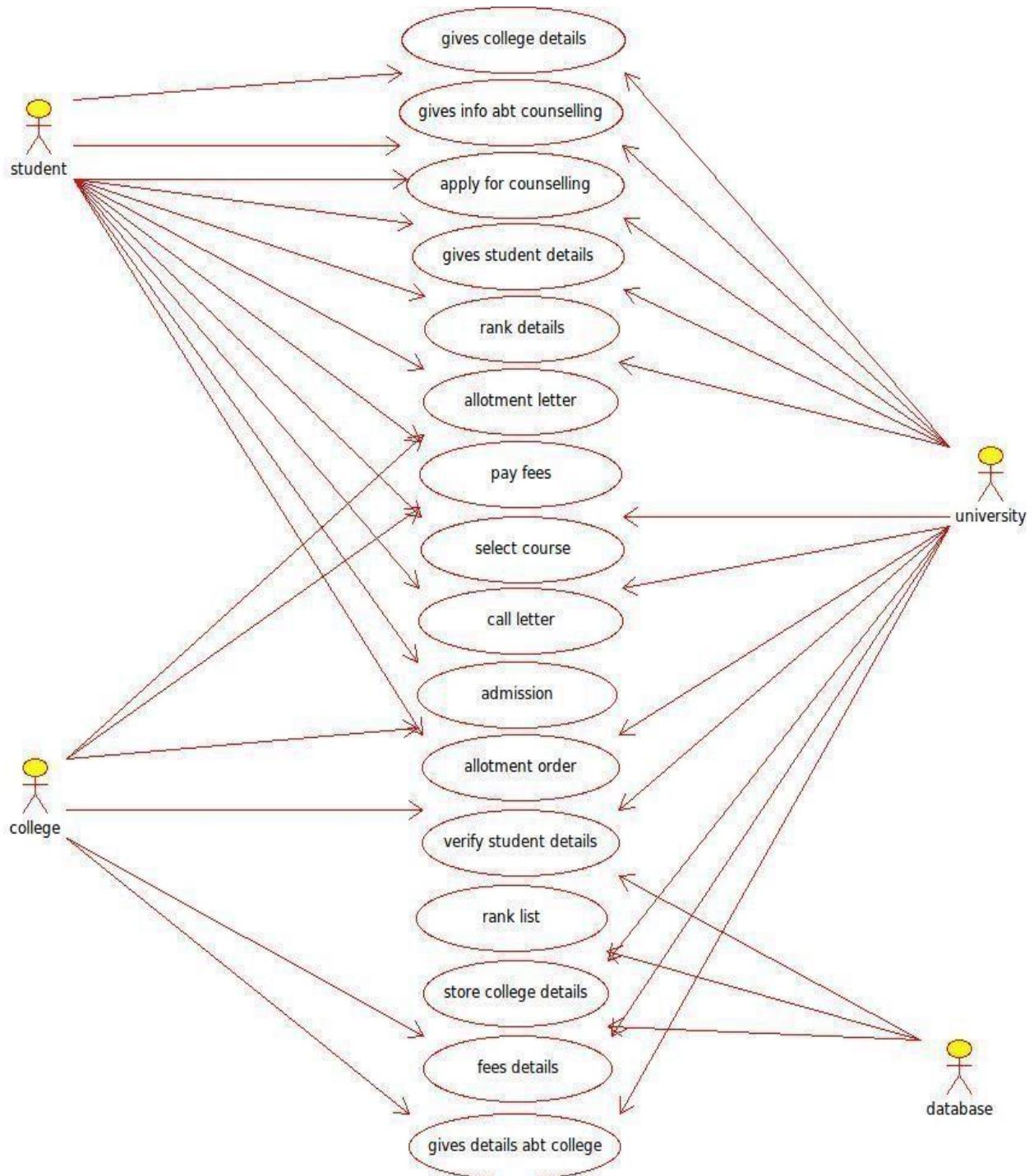
1. Initially University gives advertisement about various colleges and the students who are interested will apply for the counseling in the university
2. Then the University will intimate the student to attend the counseling in the particular date. Then the student will attend the counseling in the university
3. The University will display the caste, available courses and college list to

CS8582 - OOAD LABORATORY

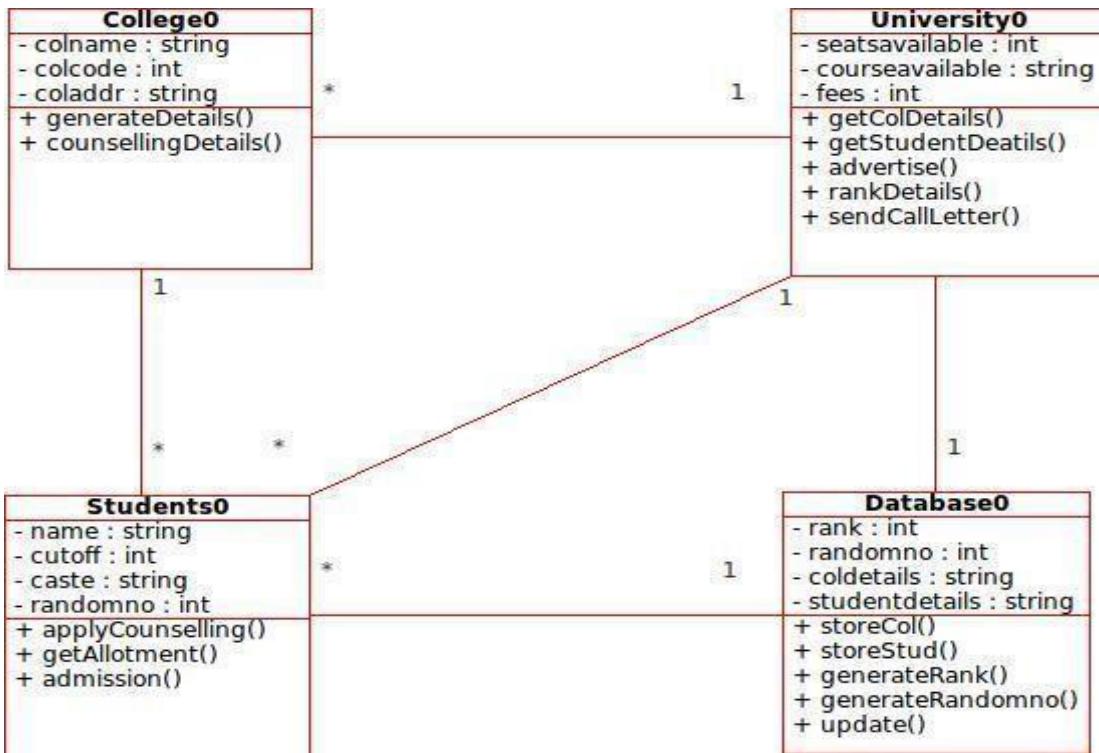
the student and then it distribute the random number to the student according to the DOB

4. Then the student will select the course and college in the university and get the allotment order from the University

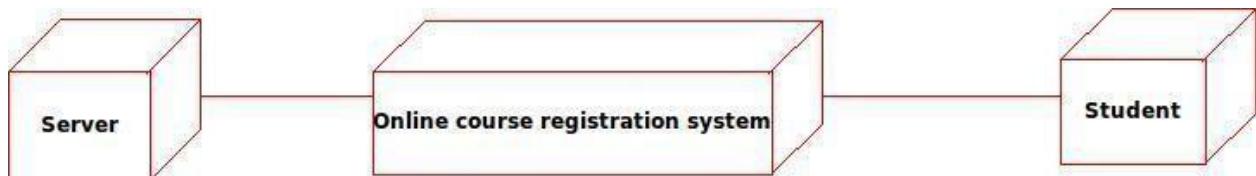
USE CASE DIAGRAM



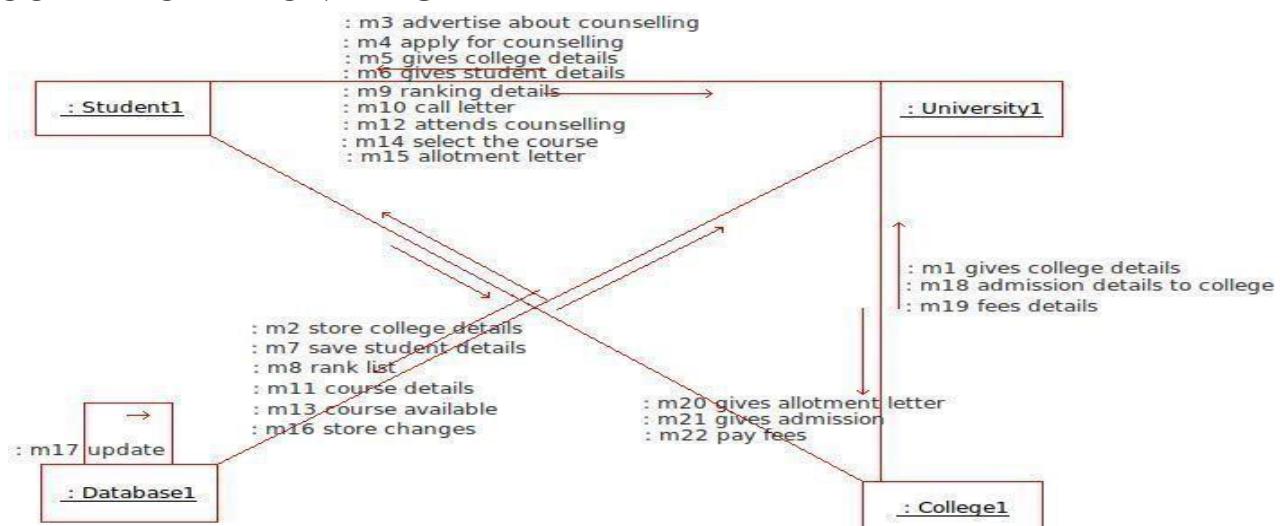
CLASS DIAGRAM



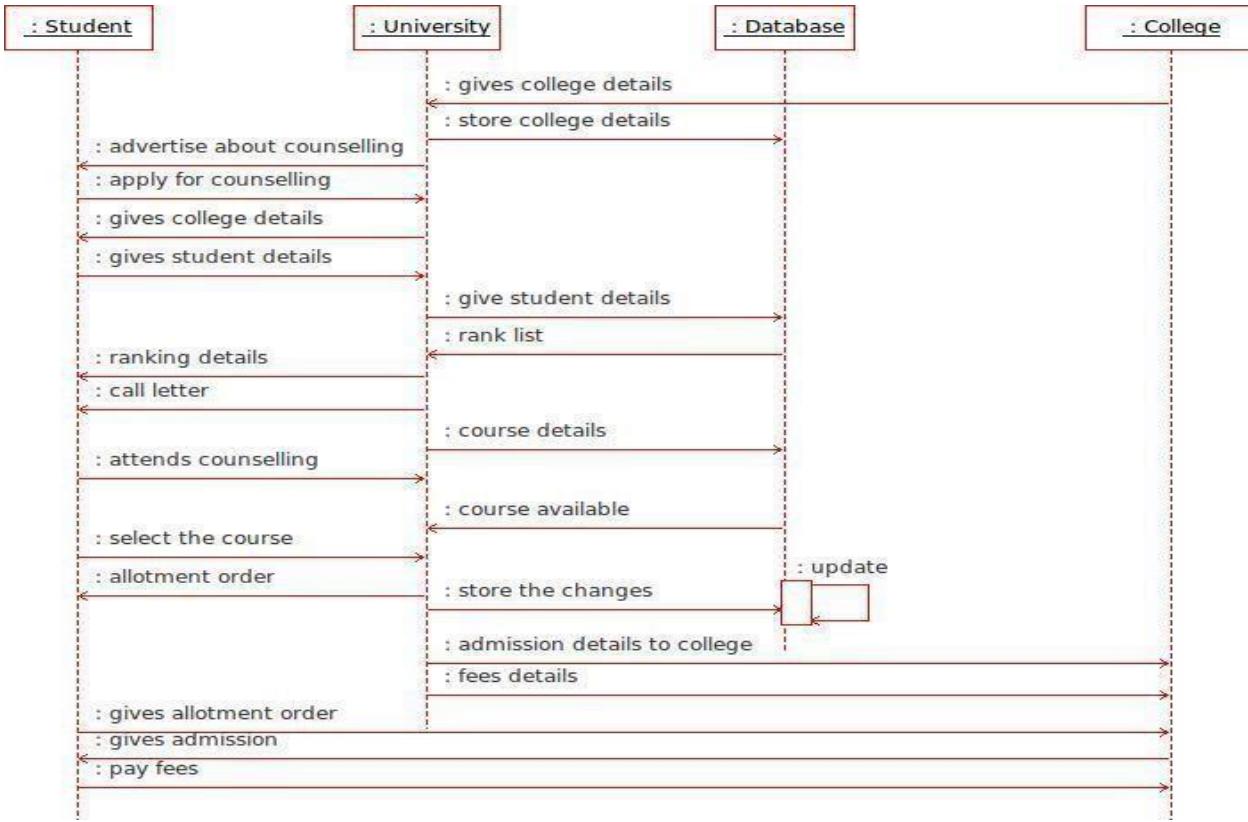
DEPLOYMENT DIAGRAM



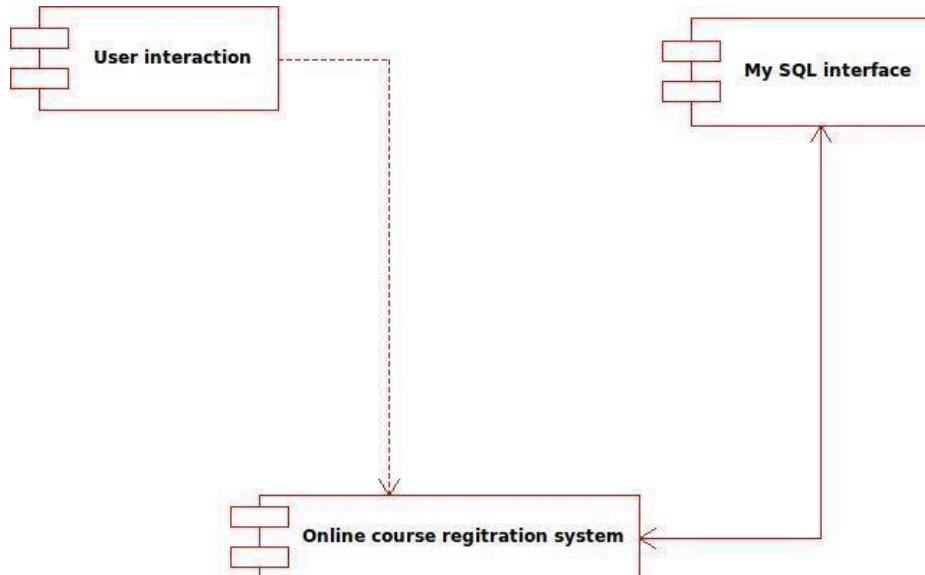
COLLABORATION DIAGRAM



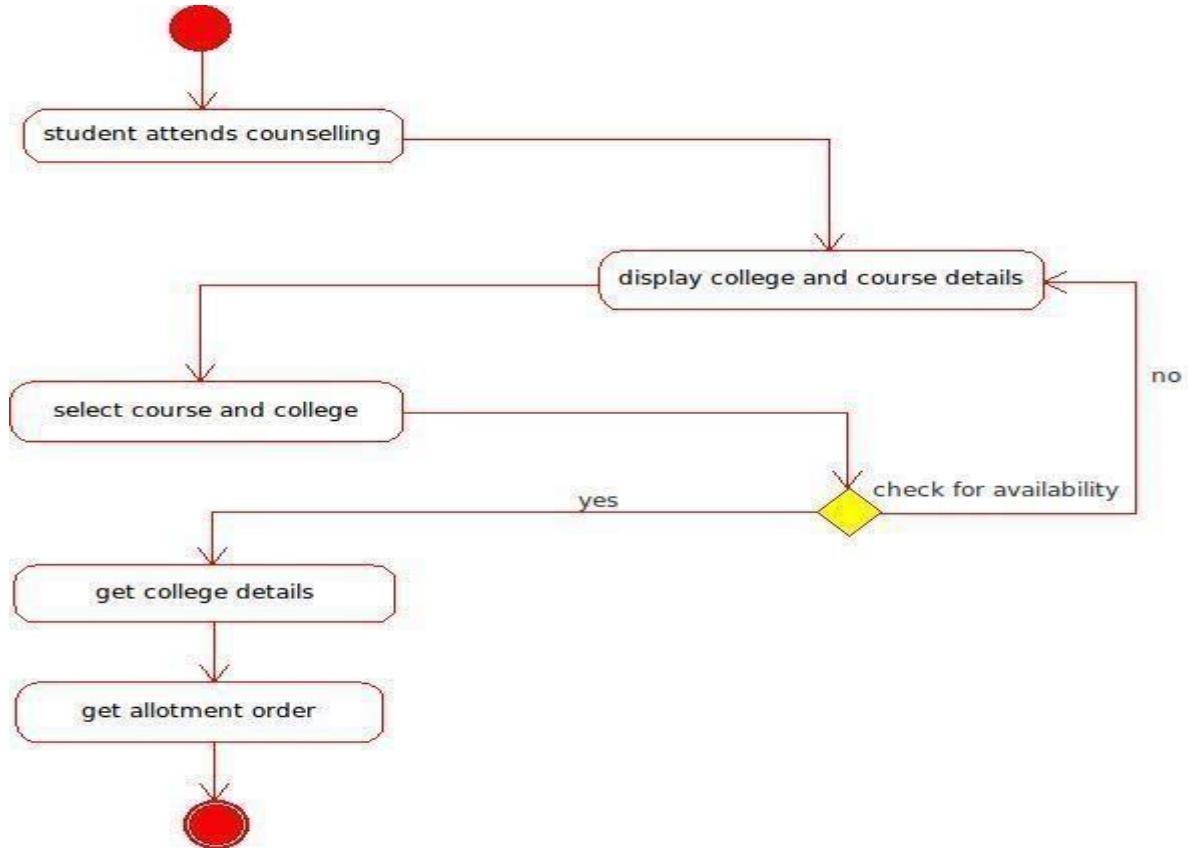
SEQUENCE DIAGRAM



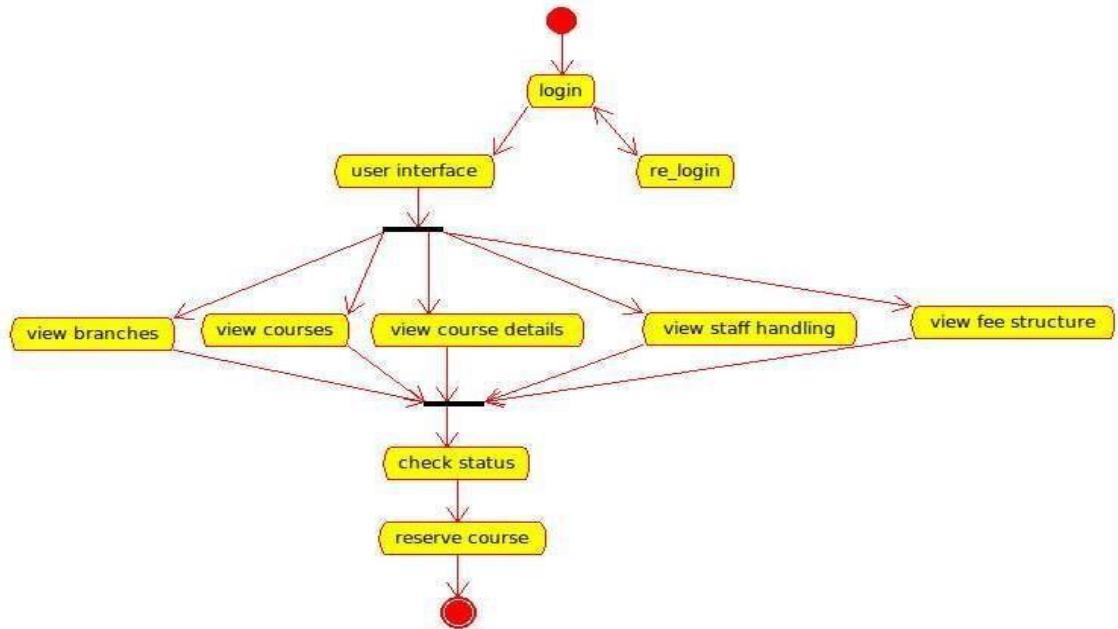
COMPONENT DIAGRAM



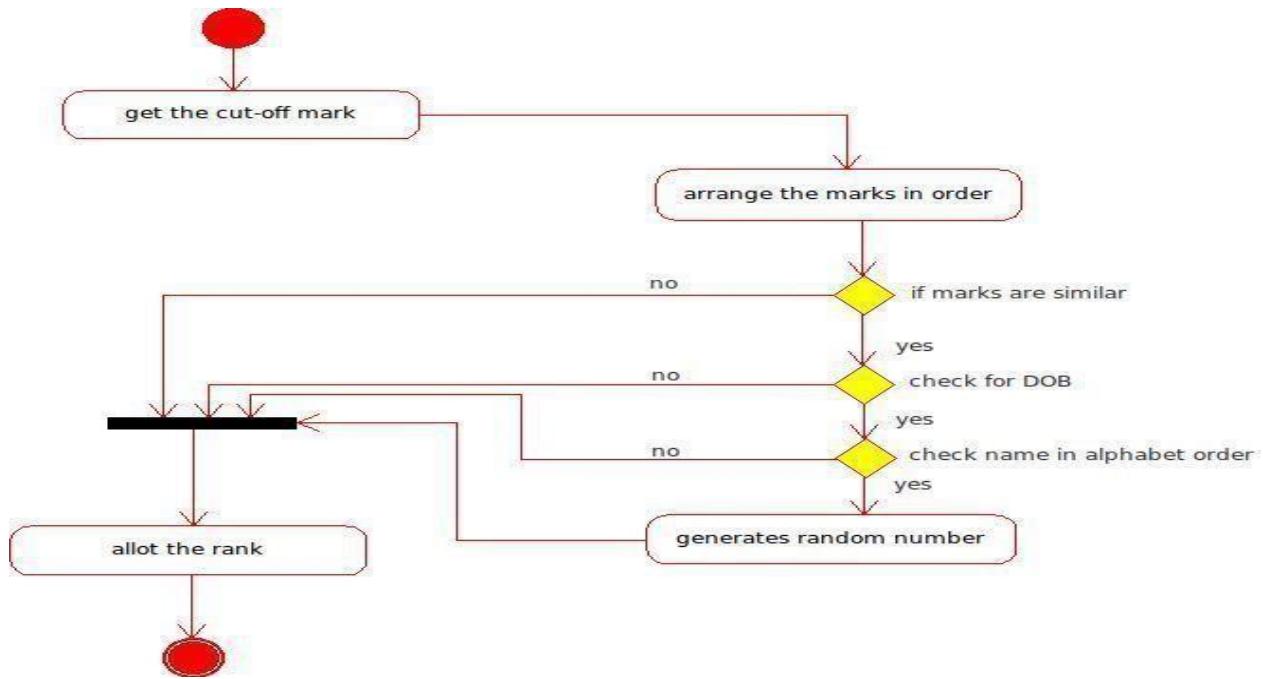
ACTIVITY DIAGRAM FOR CHECKING COURSE AVAILABILITY



STATE CHART DIAGRAM



ACTIVITY DIAGRAM FOR RANKING



OUTPUT

```

College.java
import
java.util.*; /**
 * Class College0
 */ public class College0
{ // Fields private String
colname; private int
colcode; private String
coladdr; public College0
() { };
// Accessor methods
/**
 * Set the value of colname
 * @param newVar the new value of colname
 */ private void setColname ( String
newVar ) {
    colname = newVar;
}
/**
 * Get the value of colname
 * @return the value of colname
 */ private String getColname (
) {
    return colname;
}

```

```

* Set the value of colcode
* @param newVar the new value of colcode
 */ private void setColcode ( int
newVar ) {
    colcode = newVar;
}
/***
* Get the value of colcode
* @return the value of colcode
 */ private int getColcode (
) { return
    colcode;
}
/***
* Set the value of coladdr
* @param newVar the new value of coladdr
 */ private void setColaddr ( String
newVar ) {
    coladdr = newVar;
}
/***
* Get the value of coladdr
* @return the value of coladdr
 */ private String getColaddr (
) {
    return coladdr; public void } generateDetails( )
{
}
public void counsellingDetails( )
{
}
}

```

RESULT:

Thus the “Online course reservation system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 7	
Date:	
E-TICKETING SYSTEM	

AIM:

To analyze and Design UML Diagrams for E-Ticketing system using Umbrello software.

1.SOFTWARE REQUIREMENT SPECIFICATION: SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

2.ANALYSIS:

2.1. Identifying Actors:

Client	Project
Team members	manager
Company	Testing team
Team leader	database

2.2. Identifying Use Cases:

Reqts Verifying sub modules reqts software testing validation distribution updating modules recording accessing perfect software

2.3. Identifying Classes:

agreement	feed back
soft_develop re	
modification	

2.4. Identifying Attributes:

specify	buy
allocation	comment
reqts	code
valid time	feature
limit	sample

sub module
soft

5. Identifying Methods:

Get reqt	C_test
Allocate	Process
Redevelop	Report
Validate	
Submit	

6. Identifying Relationship

Association

3.PROJECT SCOPE:

The main scope of the project is to perform the reservation, modification and cancellations for E-Ticketing system.

4.OBJECTIVE:

The main objective of designing and developing an E-Ticketing is to make the reservation and other related processes of this system more flexible and easier for the end users and without waiting in the long queue for the above mentioned processes.

5.PROBLEM STATEMENT:

Initially, the passenger will login into E-Ticketing system with his/her own credentials.

The passenger will plan for the travel by providing his/her Source Train Station, Destination Train Station, Journey Date and Quota (General/Tatkal/Ladies) details and the system will provide the train details (Train No/Train Name/Train Time/Travel Duration) to the passenger.

I f we need to check the ticket availability of a particular train, we need to select that train and the E - Ticketing System will show the ticket availability.

If the Tickets are available, We can book the ticket by entering the passenger details (Name, Age, Gender, Berth preference). Else repeat step3.

After entering the passenger details, the passenger will submit the booking

CS8582 - OOAD LABORATORY

registration form to the system and the System will confirm the booking registration details to the passenger.

Once the Passenger will confirm the details provided by him/her, the system will ask the mode of payment (Internet/Credit/Debit Banking).

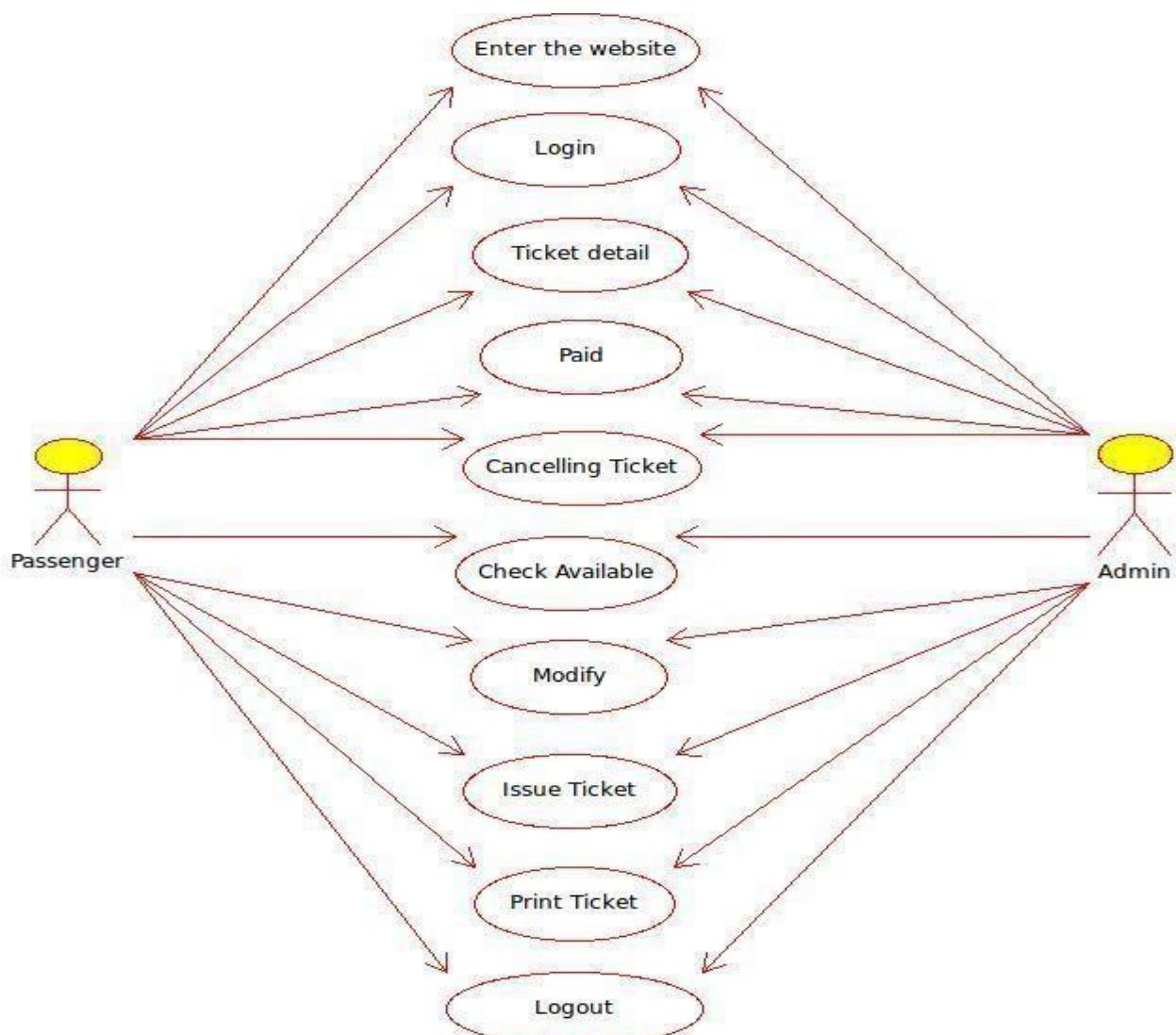
The passenger will select the mode of payment and the system will enter into the Banking site for payment.

Once the payment is done, the ticket will get booked and he/she can take a ticket print out.

If the passenger wants to cancel a ticket of a particular date, he/she can cancel a ticket by providing the train details (Train No/Name) and the ticket will be notified as cancelled.

The Passenger can check the refund status of cancelled ticket.

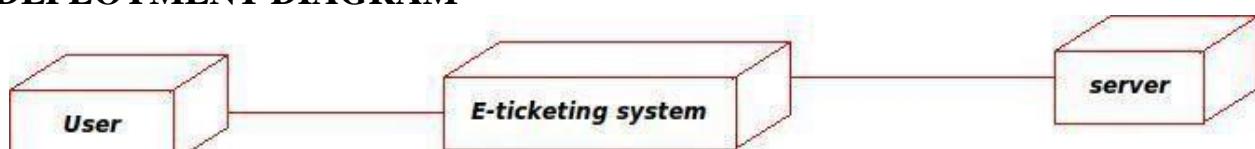
USE CASE DIAGRAM



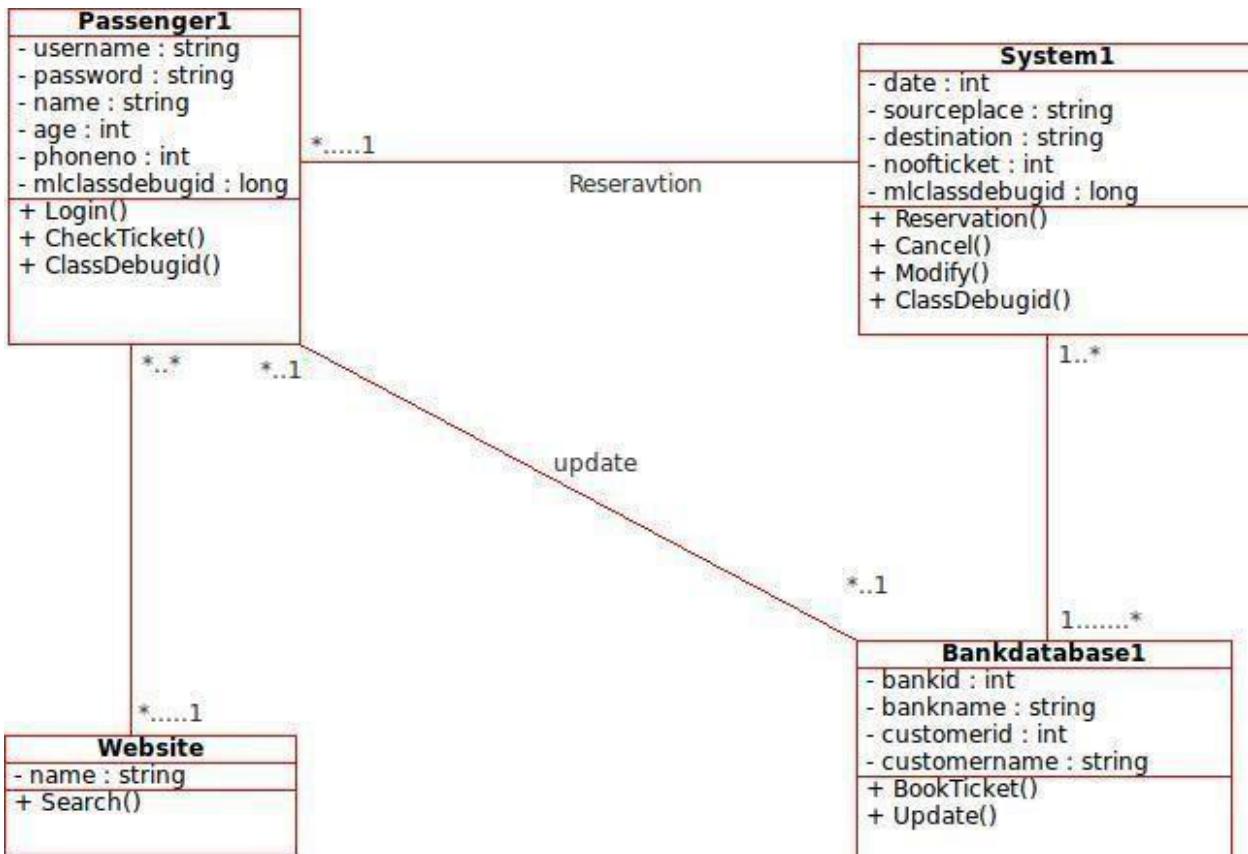
COMPONENT DIAGRAM



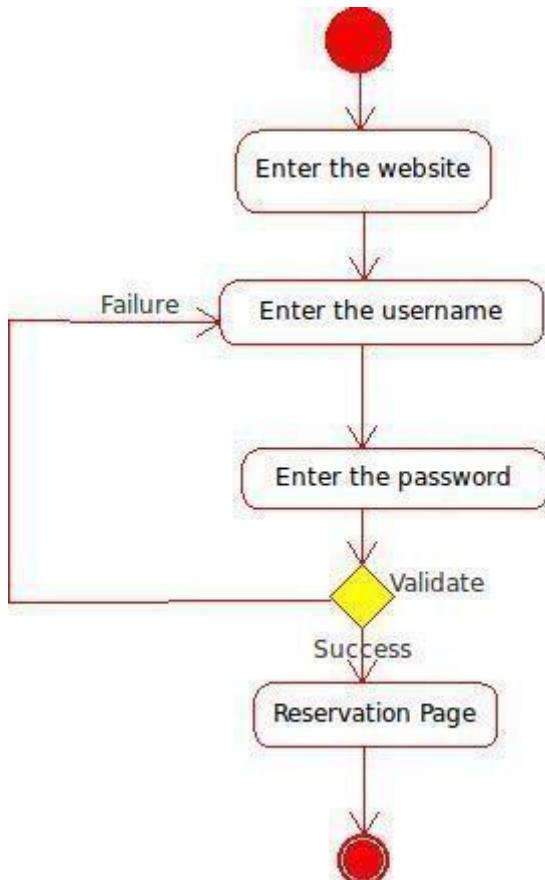
DEPLOYMENT DIAGRAM



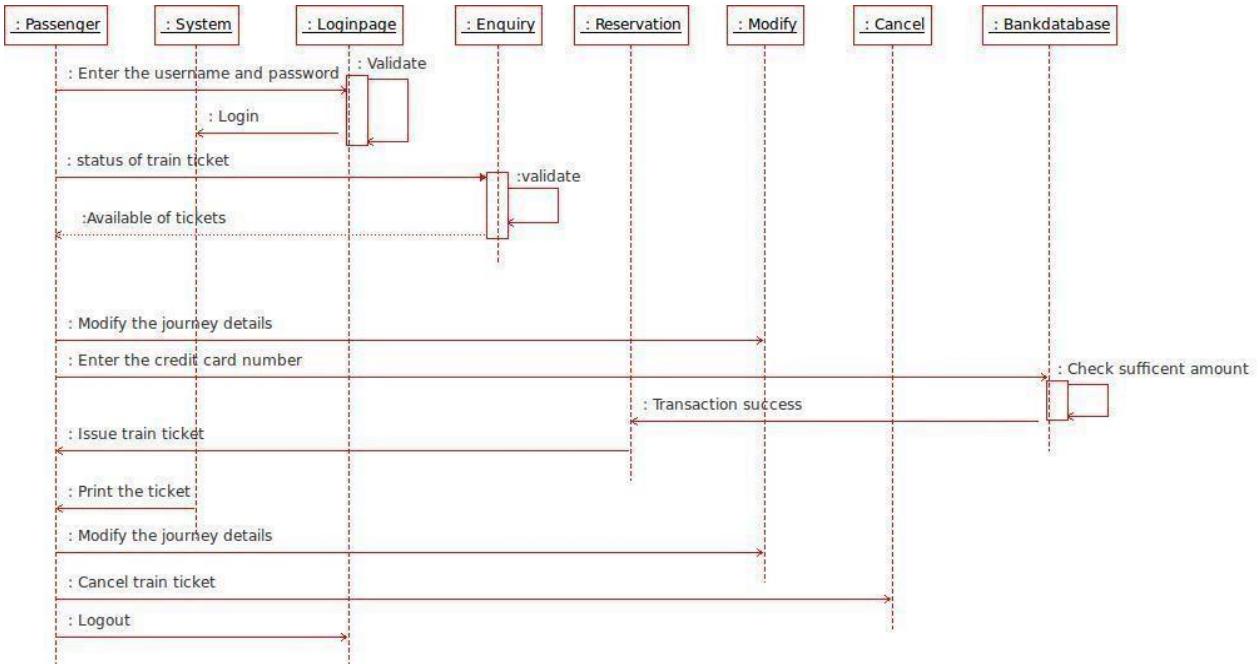
CLASS DIAGRAM



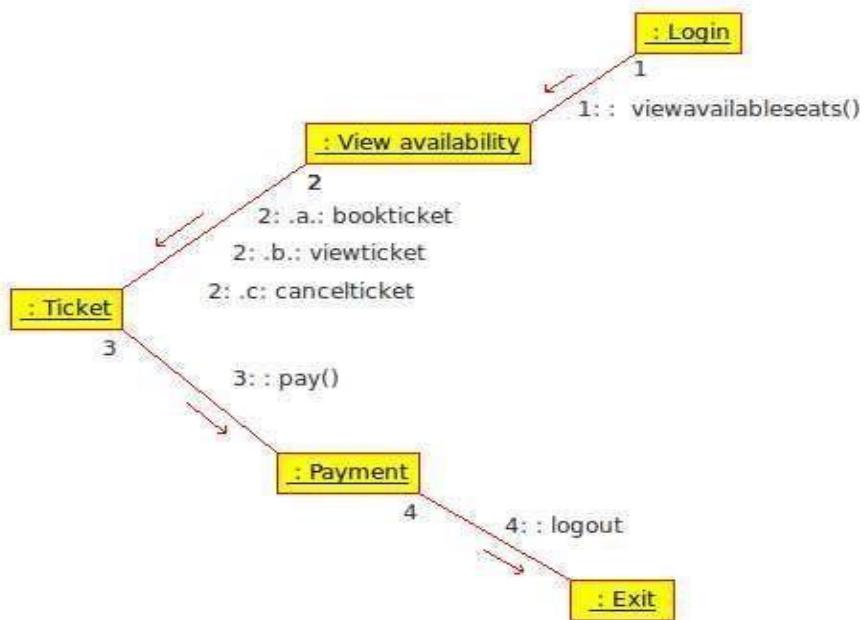
ACTIVITY DIAGRAM FOR CHECK TICKET AVAILABILITY



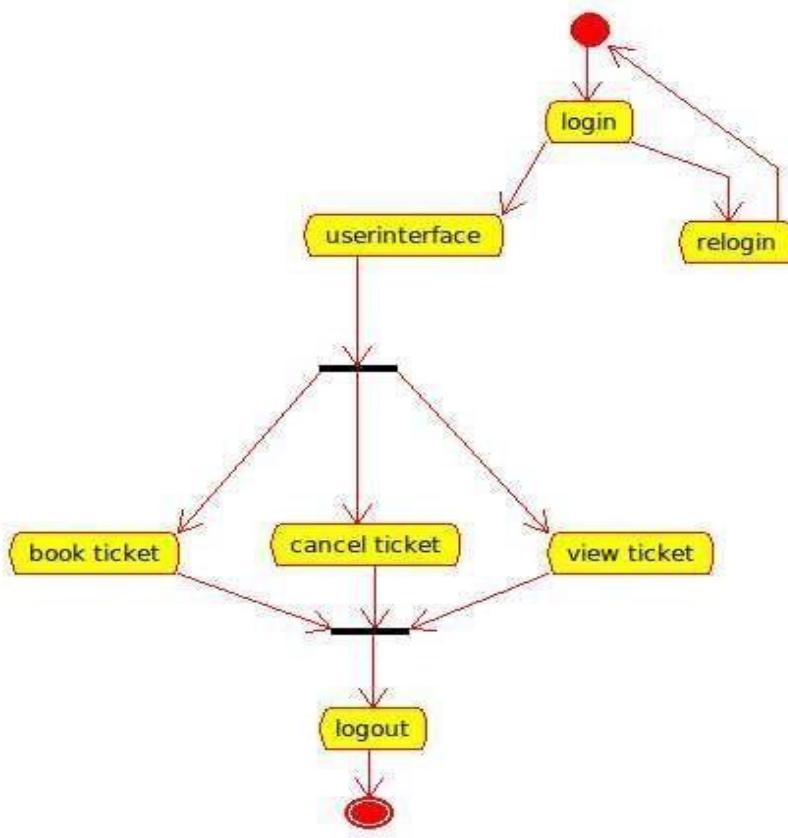
SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



STATE CHART DIAGRAM



OUTPUT:

```

Passenger1.java import
java.util.*;
/**
 * Class Passenger1
 */ public class Passenger1 {
private String username;
private String password;
private String name; private int
age; private int phoneno;
private long mlclassdebugid;
// Constructors public
Passenger1 () { };
// Accessor methods
/**
 * Set the value of username
 * @param newVar the new value of username
 */ private void setUsername ( String
newVar ) {
    username = newVar;
}
/**
 * Get the value of username
 * @return the value of username

```

```

 */ private String getUsername (
) { return
    username;
}
/***
 * Set the value of password
 * @param newVar the new value of password
 */ private void setPassword ( String
newVar ) {
    password = newVar;
}
/***
 * Get the value of password
 * @return the value of password
 */
private String getPassword () {
    return password;
}
/***
 * Set the value of name
 * @param newVar the new value of name
 */ private void setName ( String
newVar ) {
    name = newVar;
}
/***
 * Get the value of name
 * @return the value of name
 */ private String getName (
) { return
    name;
}
/***
 * Set the value of age
 * @param newVar the new value of age
 */ private void setAge ( int
newVar ) {
    age = newVar;
}
/***
 * Get the value of age
 * @return the value of age
 */ private int getAge (

```

```

) { return
    age;
}
/***
 * Set the value of phoneno
 * @param newVar the new value of phoneno
 */ private void setPhoneno ( int
newVar ) {
    phoneno = newVar;
}
/***
 * Get the value of phoneno
 * @return the value of phoneno
 */
private int getPhoneno ( ) {
    return phoneno;
}
/***
 * Set the value of mlclassdebugid
 * @param newVar the new value of mlclassdebugid
 */ private void setMlclassdebugid ( long
newVar ) {
    mlclassdebugid = newVar;
}
/***
 * Get the value of mlclassdebugid
 * @return the value of mlclassdebugid
 */ private long getMlclassdebugid (
) { return
    mlclassdebugid;
}
// Other methods
public void Login( )
{
}
public void CheckTicket(
)
{
}
public void ClassDebugid(
)
{
}
}

```

RESULT:

Thus the “E-ticketing system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 8	
Date:	
SOFTWARE PERSONNEL MANAGEMENT SYSTEM	

AIM:

To analyze and Design UML Diagrams for Software Personnel Management system using Umbrello Software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)
RAM: 256MB
Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case
Tool: Umbrello

2.ANALYSIS:

2.1. Identifying Actors:

Client	Project
Team members	manager
Company	Testing team
Team leader	database

2.2. Identifying Use Cases:

Reqts Verifying	sub modules
reqts software	testing
validation updating	distribution
modules accessing	recording

2.3. Identifying Classes:

perfect software

agreement
soft_develop re
modification

feed back

2.4. Identifying Attributes:

specify
allocation
reqts
valid time
limit

buy
comment
code
feature
sample

sub module
soft

2.5. Identifying Methods:

Get reqt	C_test
Allocate	Process
Redevelop	Report
Validate	test
Submit	

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to provide the developing company to maintain an interface to store and access the project details and the respective customer associated with it along with the assignment details with regards to the team working on the project.

4.OBJECTIVE:

The main objective of designing and developing a Software Personnel Management system is to provide with a system which proves to be manually beneficial i.e., benefits the software developing company. The complete activities and the process right from projects inception to its successful delivery is incorporated in this system.

5.PROBLEM STATEMENT:

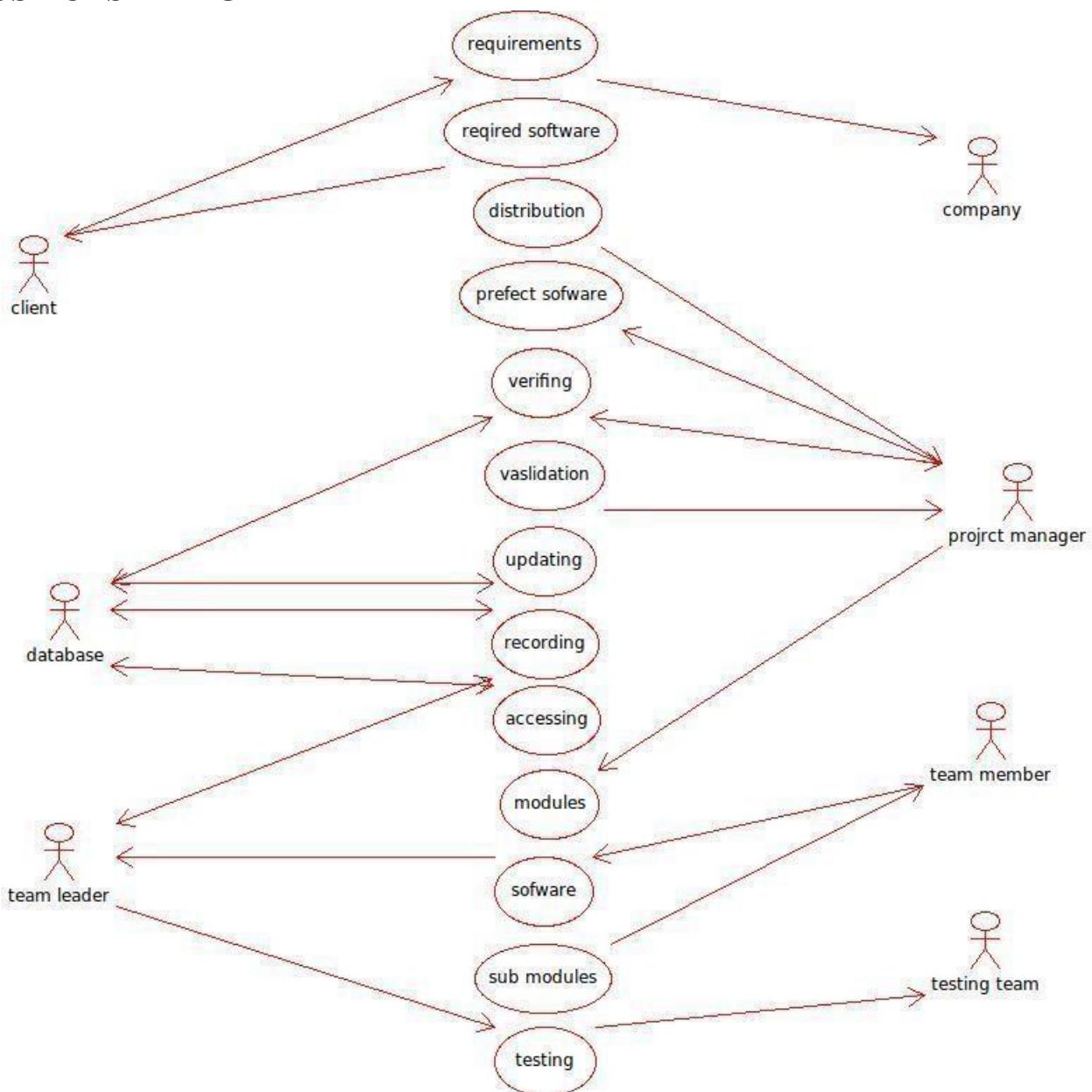
Software personnel system is a real time application used in the developer's day to day system. This is a database to store the project configuration that is specified by the customer and the software developed for it includes customer and developer requirements needed for the process. Here we assume ourselves as the developing company:

- 1.The company is the developer of the software and it contains the necessary information of the software such as requirements, time limit etc
2. The customer provides the requirements to the company and requests a

suitable solution system. The company is only responsible for developing the system **CS8582 - OOAD LABORATORY** out of the requirements of the customer.

3. The company mainly obtains the requirements from the customer and by analyzing it, prepares a requirements and feasibility documentation initially.
4. Then the company develops a software system that is apt for the integrated requirements and tests it within itself and by exposure to the concerned customer, and finally builds a complete a satisfactory solution.

USE CASE DIAGRAM



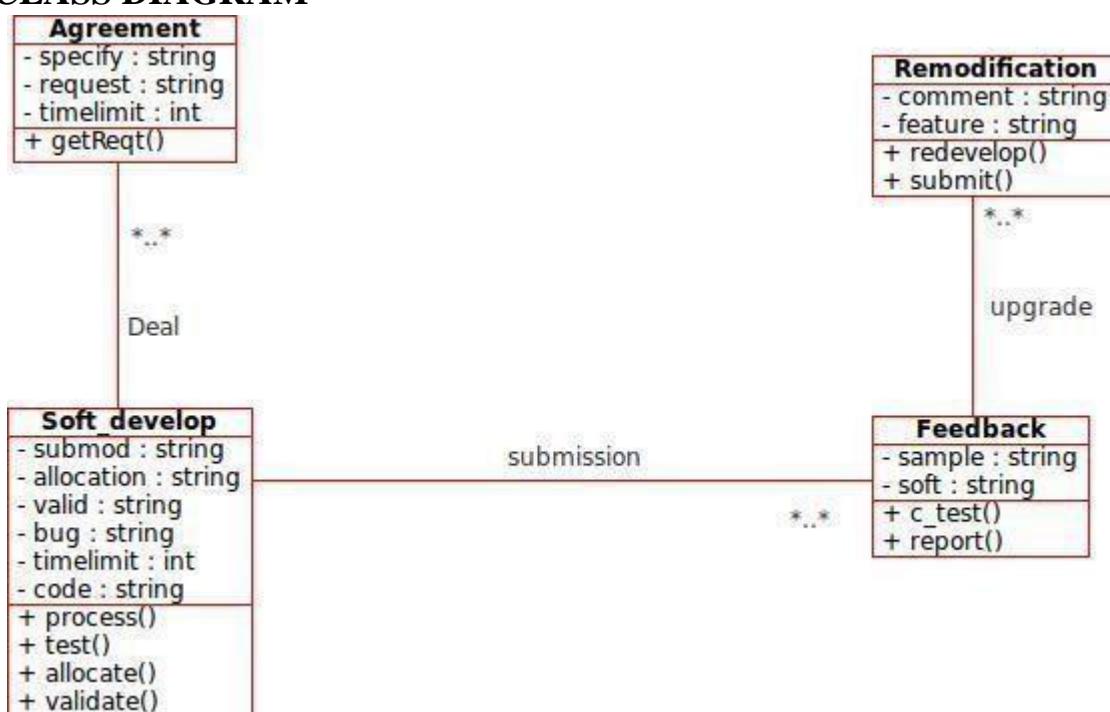
COMPONENT DIAGRAM



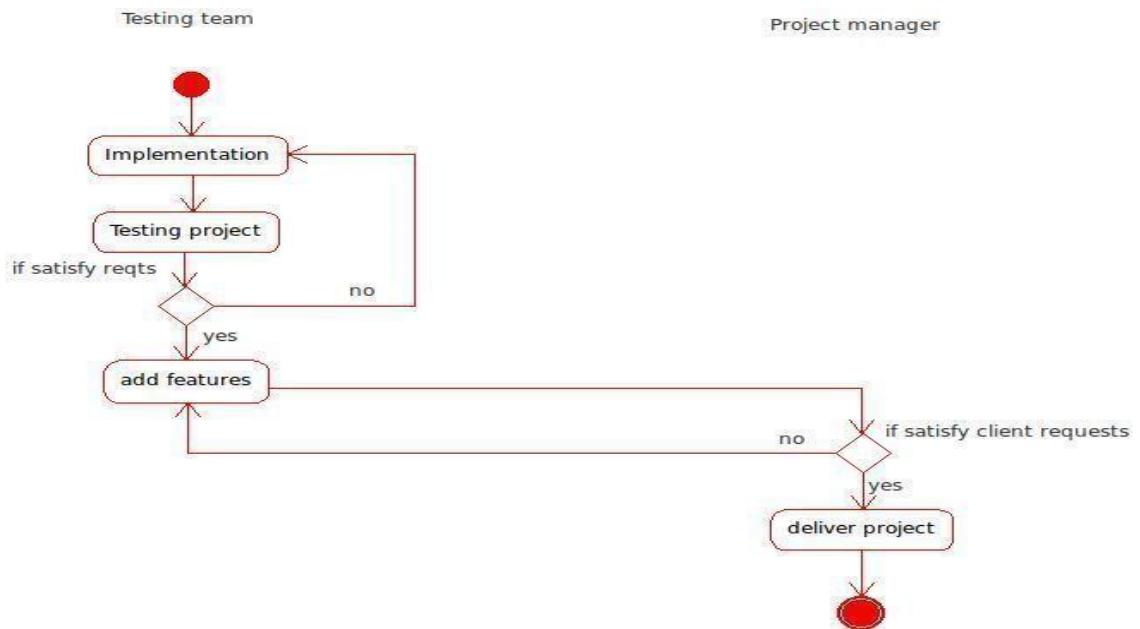
DEPLOYMENT DIAGRAM



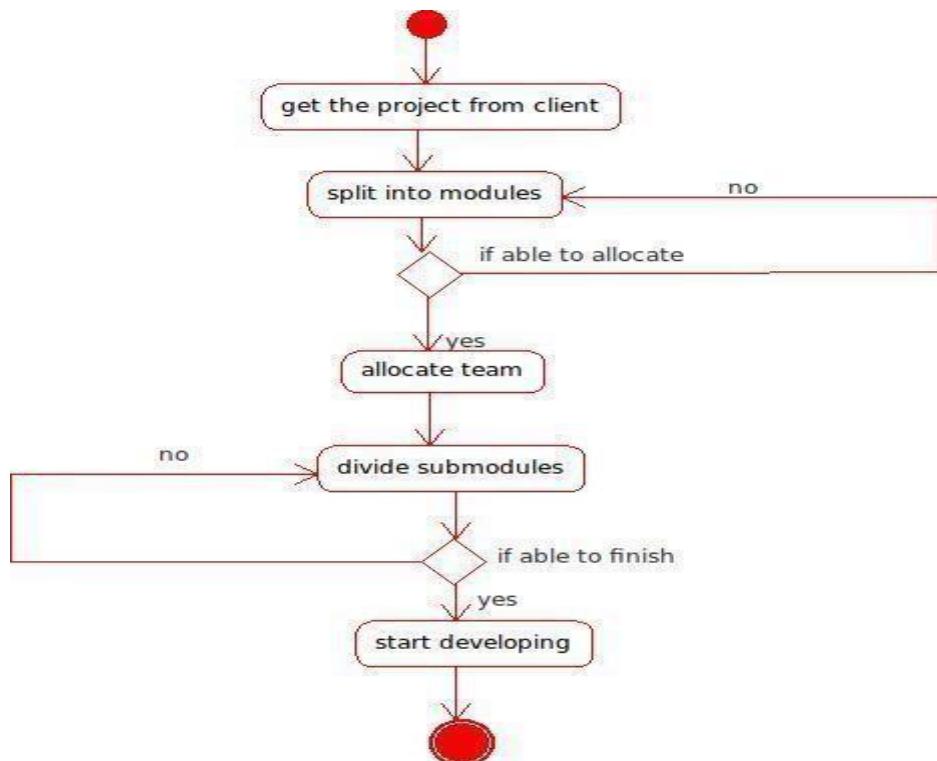
CLASS DIAGRAM



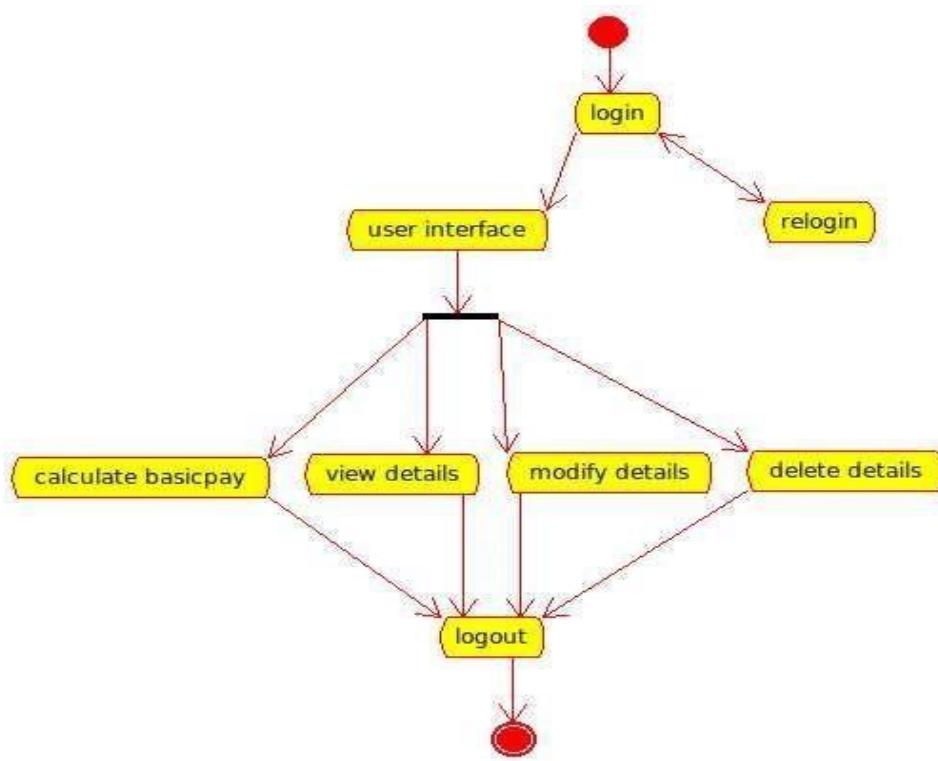
ACTIVITY DIAGRAM



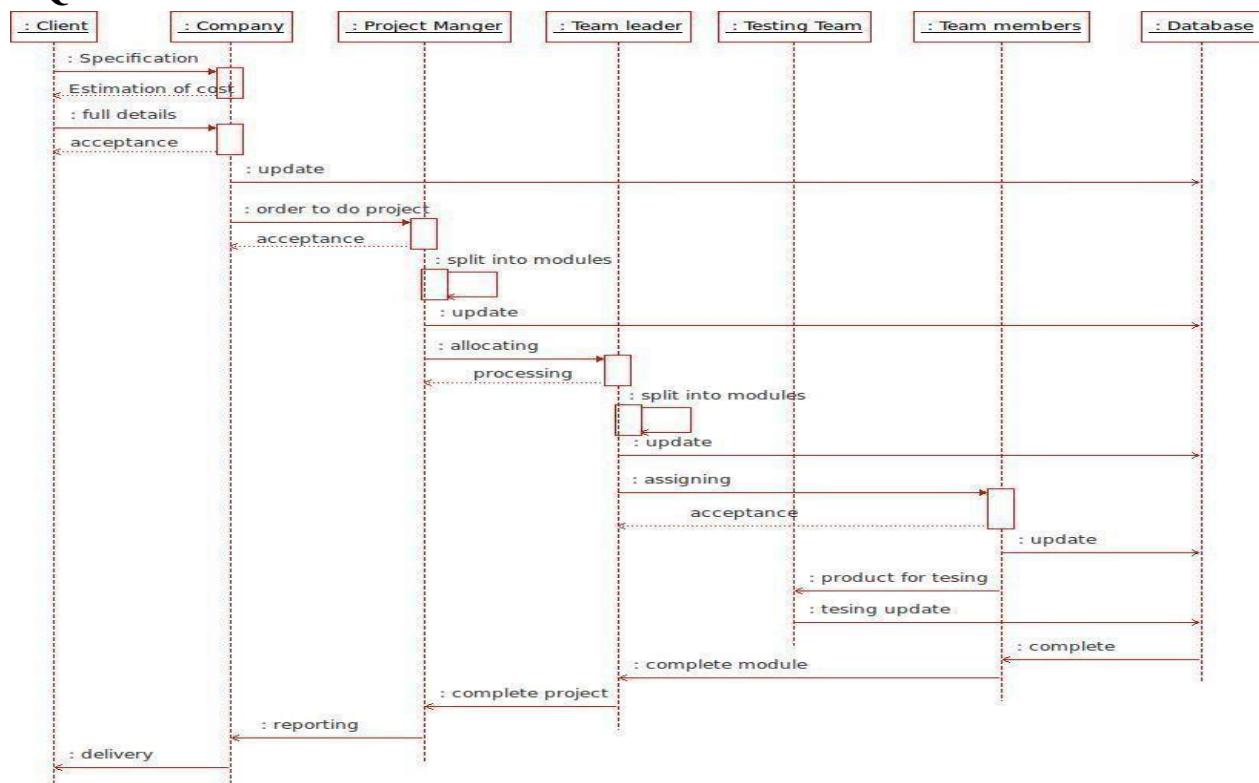
ACTIVITY DIAGRAM



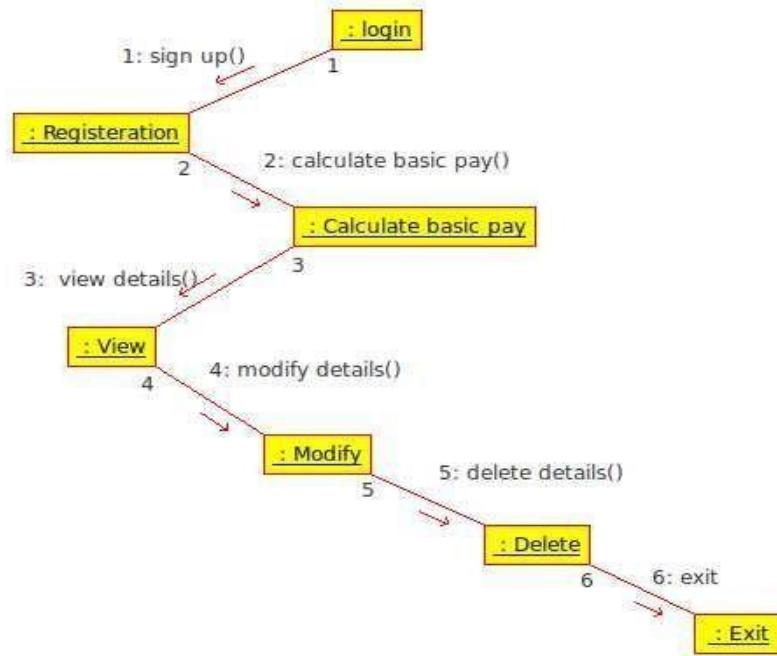
STATE CHART DIAGRAM



SEQUENCE DIAGRAM:



COLLABORATION DIAGRAM



OUTPUT:

Agreement.java

```

import java.util.*;
/**
 * Class Agreement
 */
public class Agreement {
    // Fields
    private String specify;
    private String request;
    private int timelimit;

    Constructors
    public Agreement () { }

    Methods
    // Accessor methods
    /**
     * Set the value of specify
     * @param newVar the new value of specify
     */
    private void setSpecify ( String newVar ) { specify = newVar; }

    /**
     * Get the value of specify
     * @return the value of specify
     */
}
  
```

```

 */ private String getSpecify (
) { return
    specify;
}
/***
 * Set the value of request
 * @param newVar the new value of request
 */ private void setRequest ( String
newVar ) {
    request = newVar;
}
/***
 * Get the value of request
 * @return the value of request
 */ private String getRequest (
) { return
    request;
}
/***
 * Set the value of timelimit
 * @param newVar the new value of timelimit
 */ private void setTimelimit ( int
newVar ) {
    timelimit = newVar;
}
/***
 * Get the value of timelimit
 * @return the value of timelimit
 */ private int getTimelimit (
) { return
    timelimit;
}
// Other methods
public void getReqt( )
{
}
}
}

```

RESULT:

Thus the “Software personnel management system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 9	CREDIT CARD PROCESSING SYSTEM
------------	--------------------------------------

Date:

AIM:

To analyze and Design UML Diagrams for credit card processing using Umbrello software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

2.ANALYSIS:

2.1. Identifying Actors:

Customer	Bank	DB
Retailer		

2.2. Identifying Use Cases:

Reqt for the application form	Accessible code is entered
Credit card is received	Payment is accepted
Credit card is given to the retailer	Ack is received
Billing is made	Details are satisfied

2.3. Identifying Classes:

Customer	Database
Bank	
Mediator	

2.4. Identifying Attributes:

Name	Mediator object Access_id
Receipt	
Id	
Acc_no	

2.5. Identifying Methods:

Purchases_using_creditcard	Provides detail about customer
Issues_the_card	Mediator_submit_the form
Mediatorrequest to get creditcard	Accepts the payment

Informs the current balance Submit
the form Stores detail of cust

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to perform the activities like issuing of credit card, enquiring about the balance, billing process made by customer.

4.OBJECTIVE:

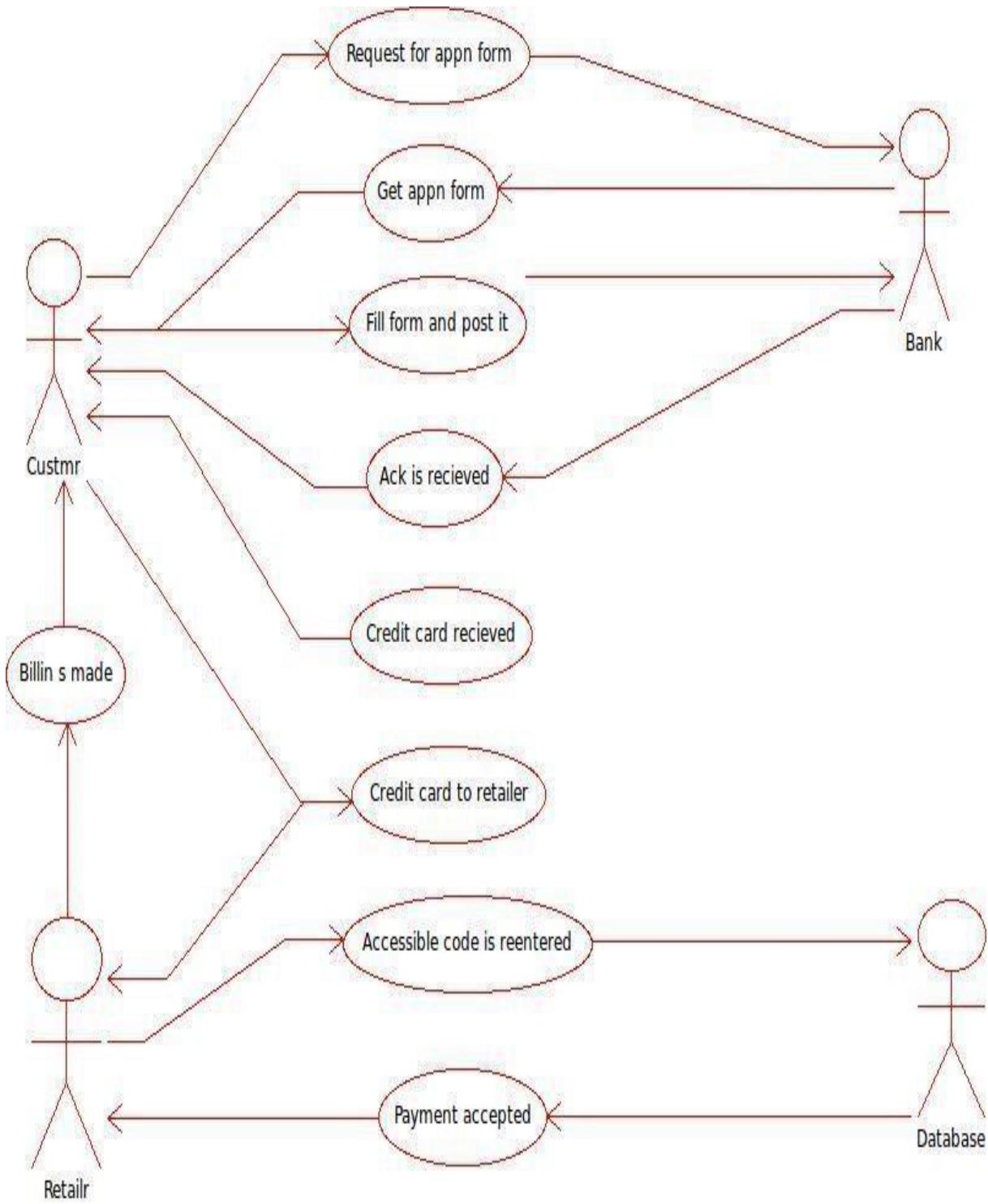
The main objective of designing and developing the credit card processing is to provide with the system which proves to be manually beneficial i.e., benefits for both customer and bank. The complete activities and the process right from accessing a bank account, getting a credit card and processing of credit card, every single activity is incorporated in this system.

5. PROBLEM STATEMENT:

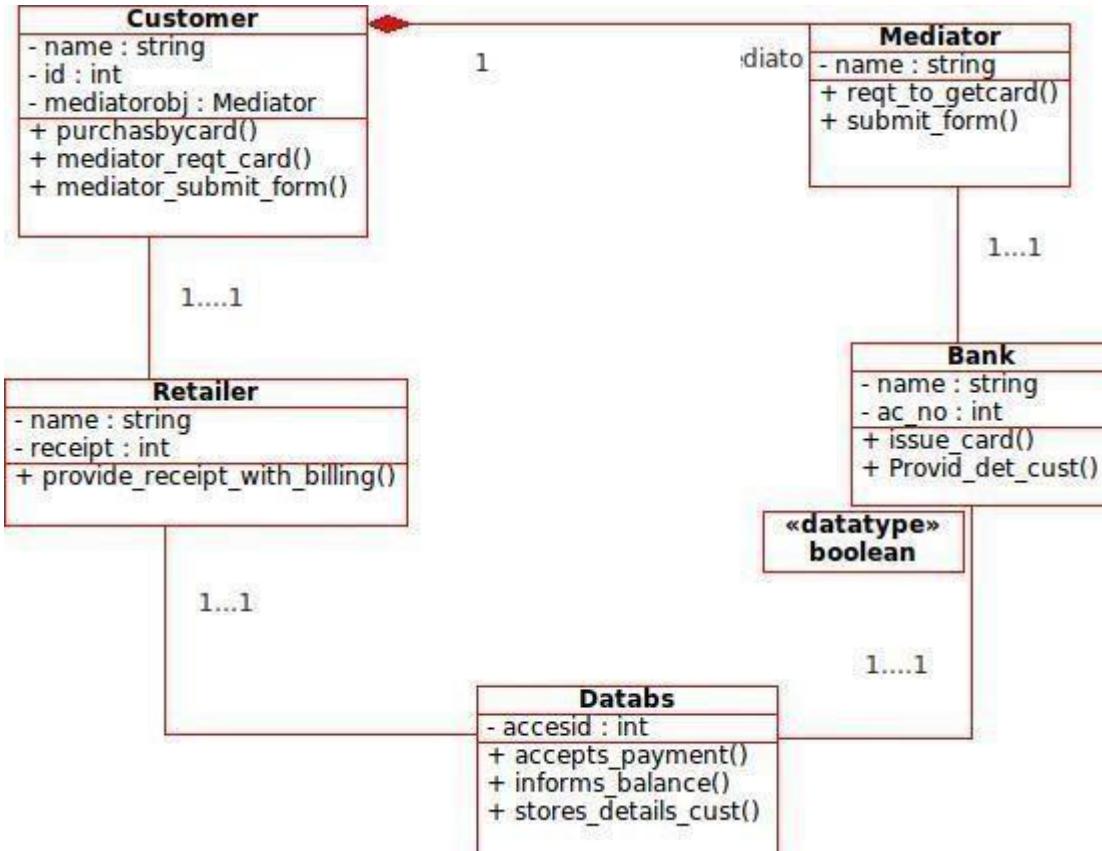
1. Submit the application form to the bank
- 2 Verification of personal details of customer is done by the bank
3. Credit card is given to the customer after issuing
4. After purchasing, the credit card is given to the customer to swipe
5. Retailer checks for the our balance enquiry from the database
- 6 Database informs bank about data access and current balance
7. The billing details are provided by the bank to the customer

CS8582 - OOAD LABORATORY

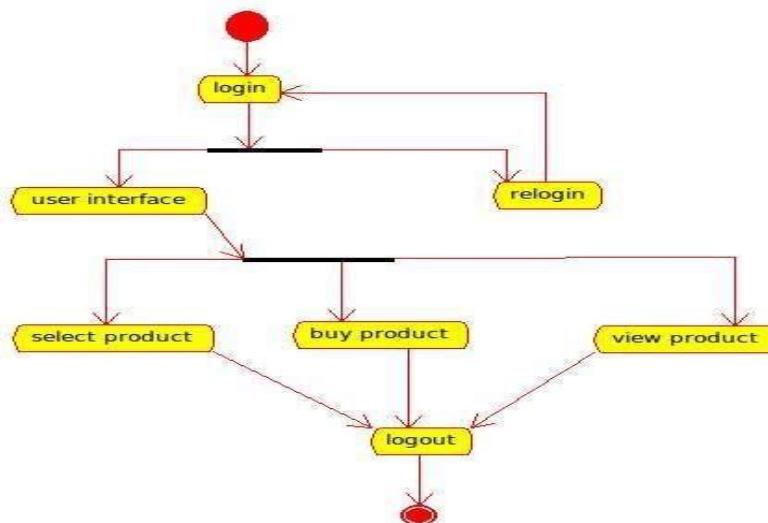
USE CASE DIAGRAM



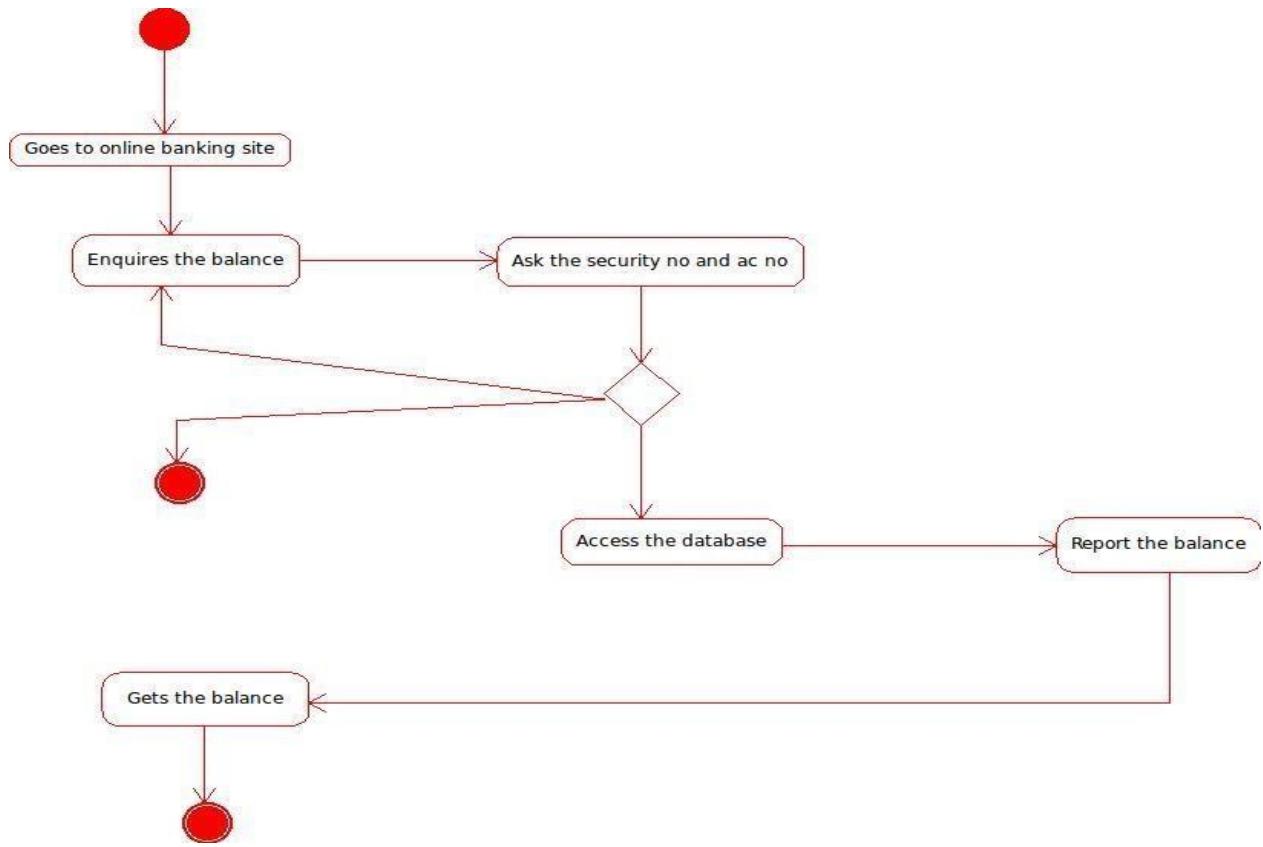
CLASS DIAGRAM



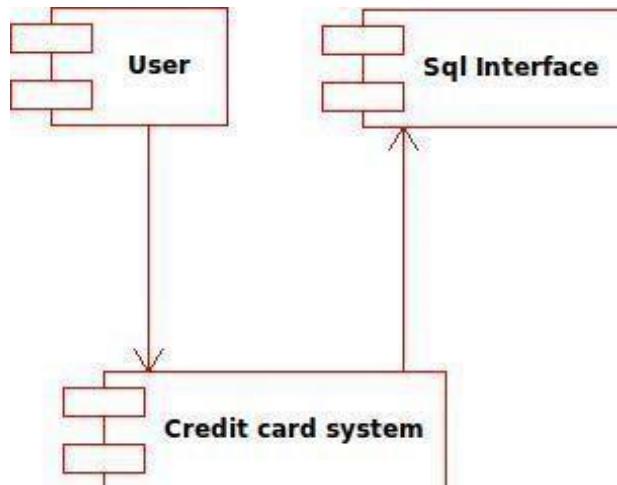
STATE CHART DIAGRAM



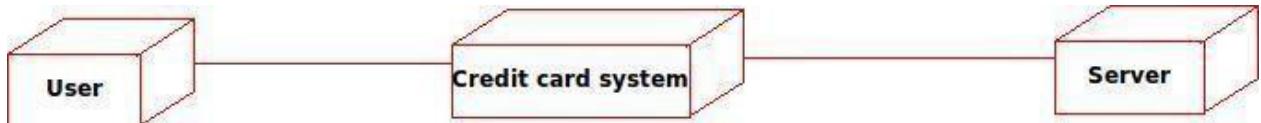
ACTIVITY DIAGRAM



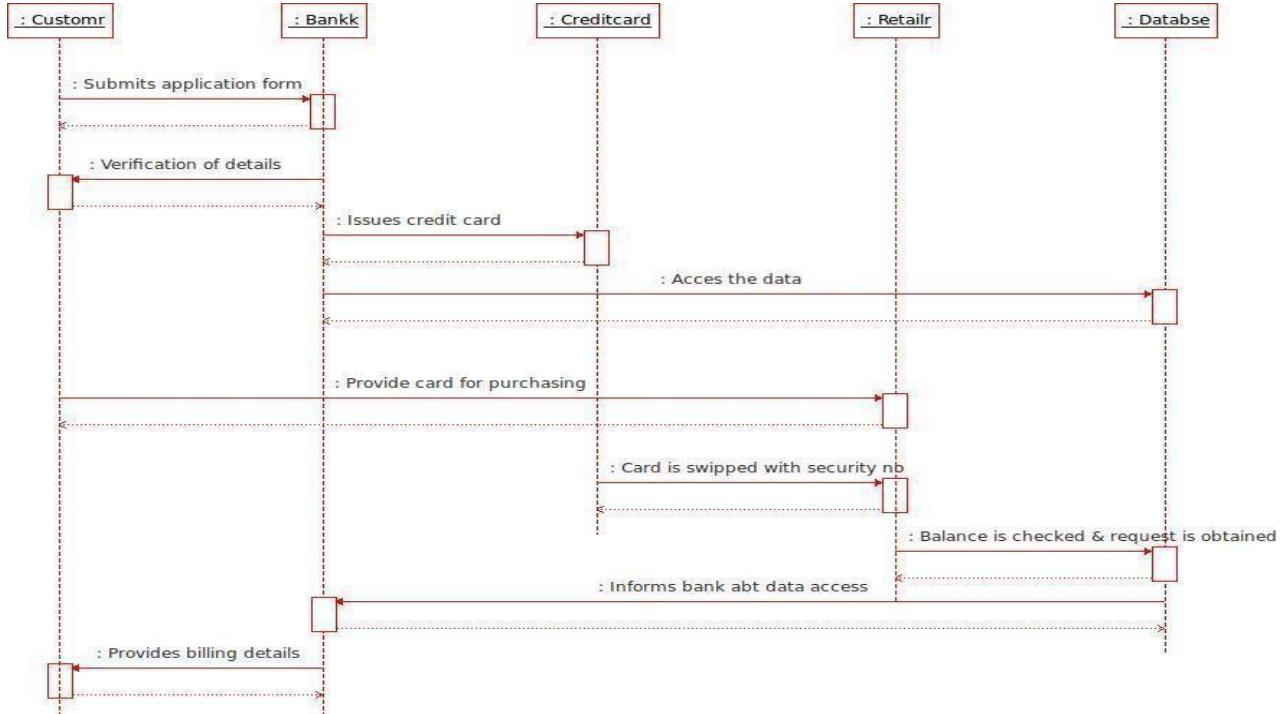
COMPONENT DIAGRAM



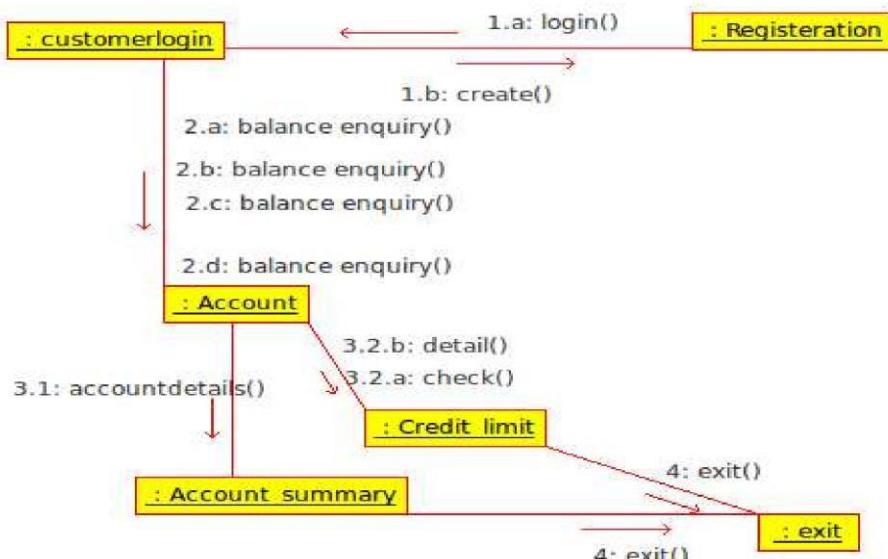
DEPLOYMENT DIAGRAM



SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



OUTPUT

```

Retailer.java import
java.util.*; /** * Class
Retailer */ public
class Retailer
{
// Fields private
String name; private
int receipt; //
Constructors public
Retailer () { } //
Methods
// Accessor methods
/***
* Set the value of name
* @param newVar the new value of name
 */ private void setName ( String
newVar ) {
    name = newVar;
}
/***
* Get the value of nam
@return the value of name
 */ private String getName (
) { return
    name;
}
/***
* Set the value of receipt
* @param newVar the new value of receipt
 */ private void setReceipt ( int
newVar ) {
    receipt = newVar;
}
/***
* Get the value of receipt
* @return the value of receipt
 */ private int getReceipt (
) { return
    receipt;
}
// Other methods public void provide_receipt_with_billing(
) { }

```

}

RESULT:

Thus the “Credit card system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 10	
Date:	
EBOOK MANAGEMENT SYSTEM	

AIM:

To analyse and design E-book management system using Umbrello Software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

ANALYSIS:

2.1. Identifying Actors:

User software browser server

2.2 Identifying Use Cases:

payment done

propose to

search the software

download accepts

book details found

the pay search

the software

the results

downloading

download

display

2.3. Identifying Classes:

User

EBook

Browser

Server

2.4. Identifying Attributes:

Name IP address

User id

Book name

Password

2.5. Identifying Methods:

Create_account

Author Address

version

Download_book

search

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to provide a exhaustive, flexible and reliable EBook Management system. This E-BOOK should contain index of the topics. When the main page is visited index of the topics is displayed. Select the required topic and double click on it. Then the page with the contents of the selected topic will be displayed. A certain option is also present in that page to go back to main page and search for other topics.

4.OBJECTIVE:

The main objective of designing and developing a EBook Management system is to provide with the system which proves to be beneficial. The activities such as search for the book online and download, every single activity is incorporated in the system.

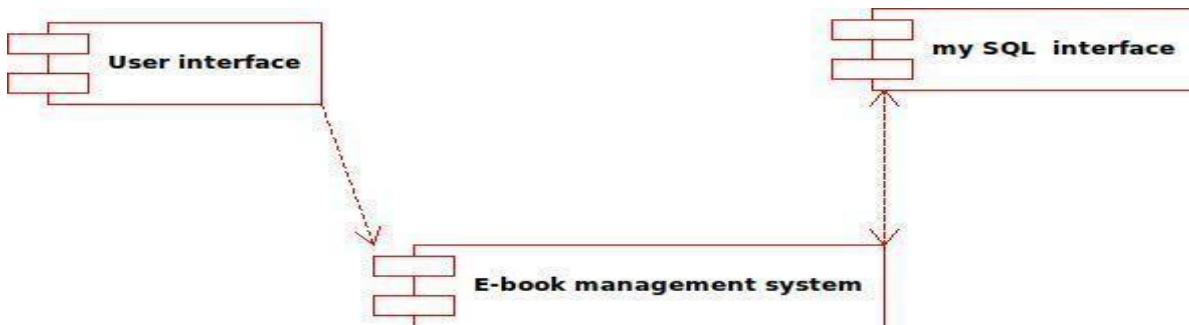
5.PROBLEM STATEMENT:

E-book management system is helpful for the download the books from internet the problem statements are as follows,

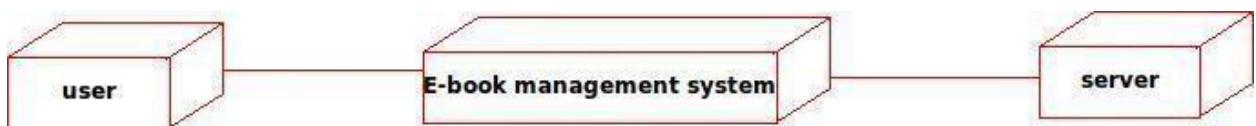
1. User accesses the web browser and gives the book details to be searched.
2. The server checks for the availability of the book, if found display the book to the user through the browser.
3. The user can also download the book .The server searches for the downloading software and downloads the book if the software is available.
4. The user can access the book by paying for it.

COMPONENT DIAGRAM

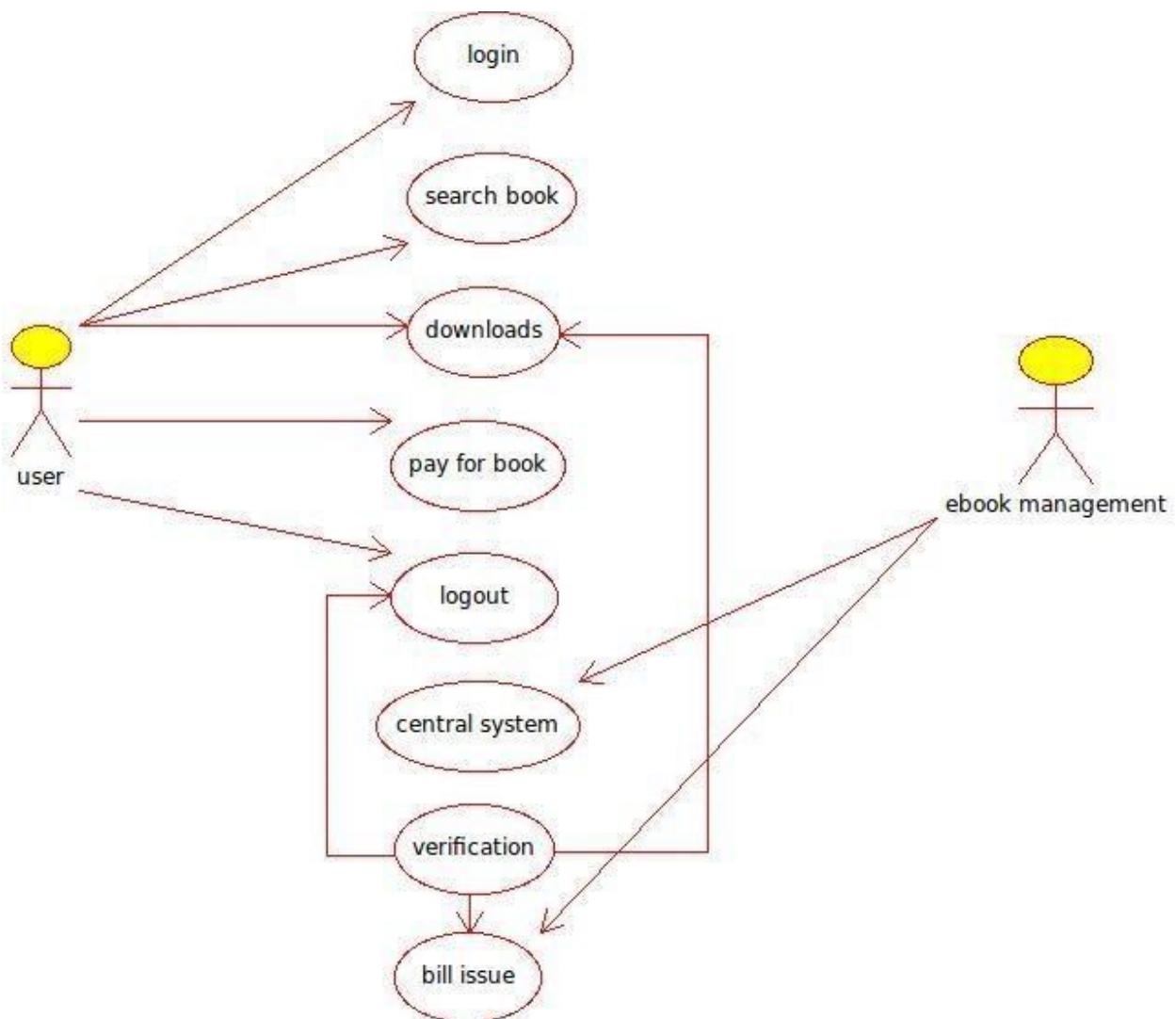
CS8582 - OOAD LABORATORY



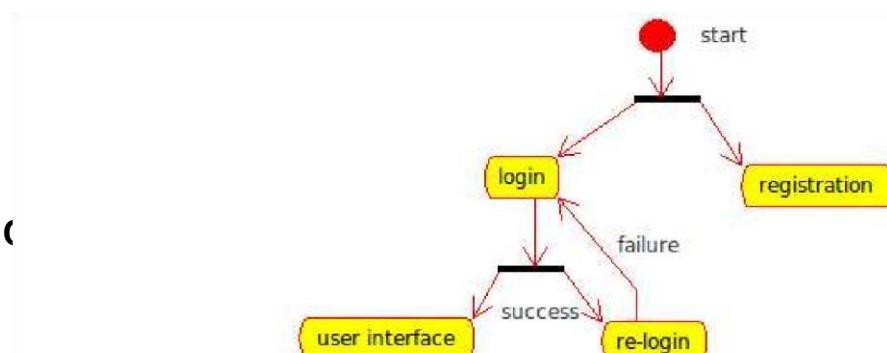
DEPLOYMENT DIAGRAM



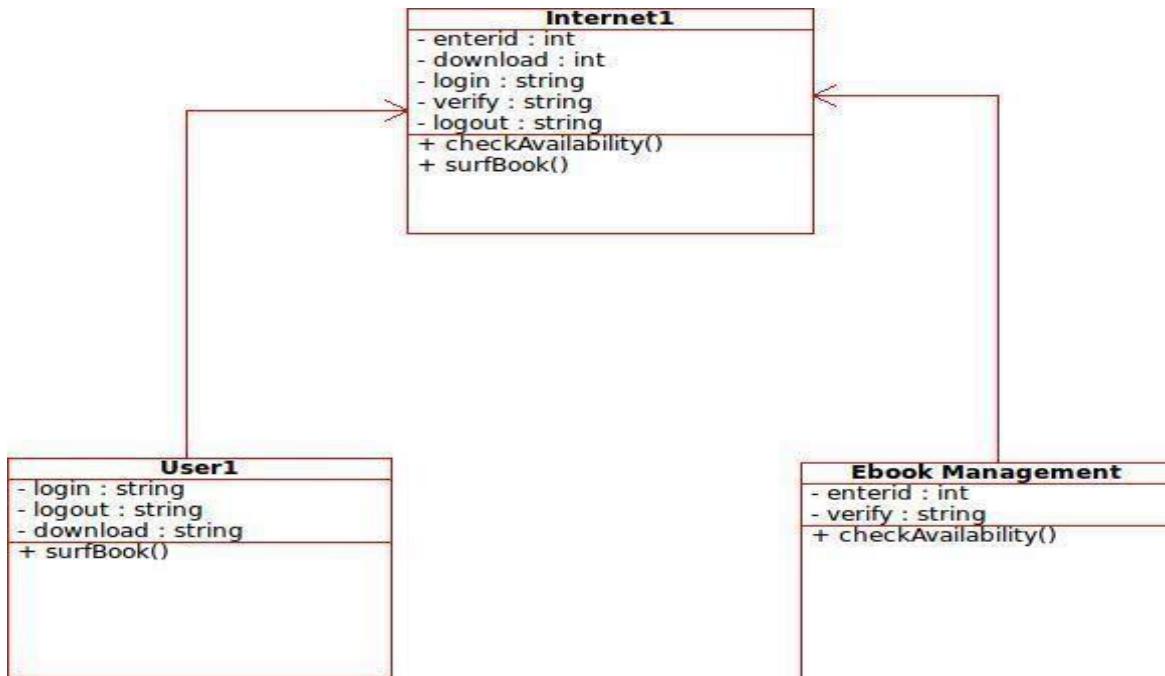
USE CASE DIAGRAM



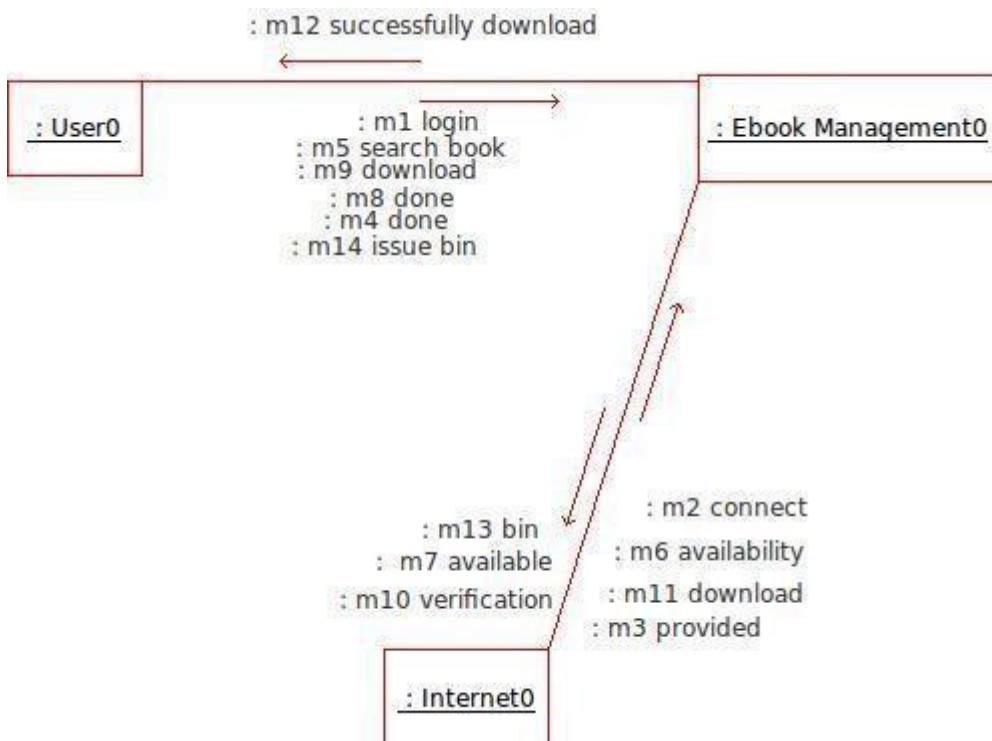
STATE CHART DIAGRAM CLASS DIAGRAM



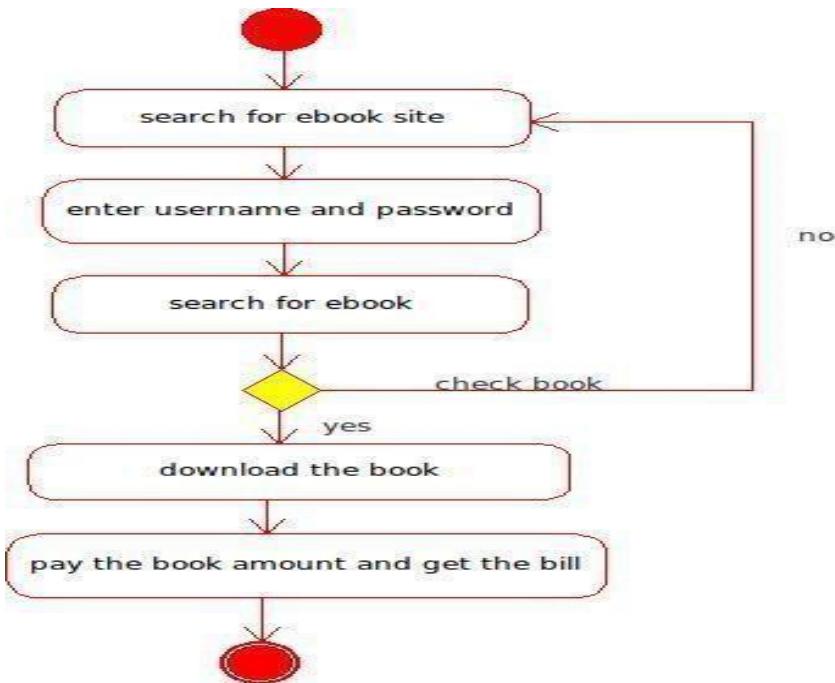
o: 76



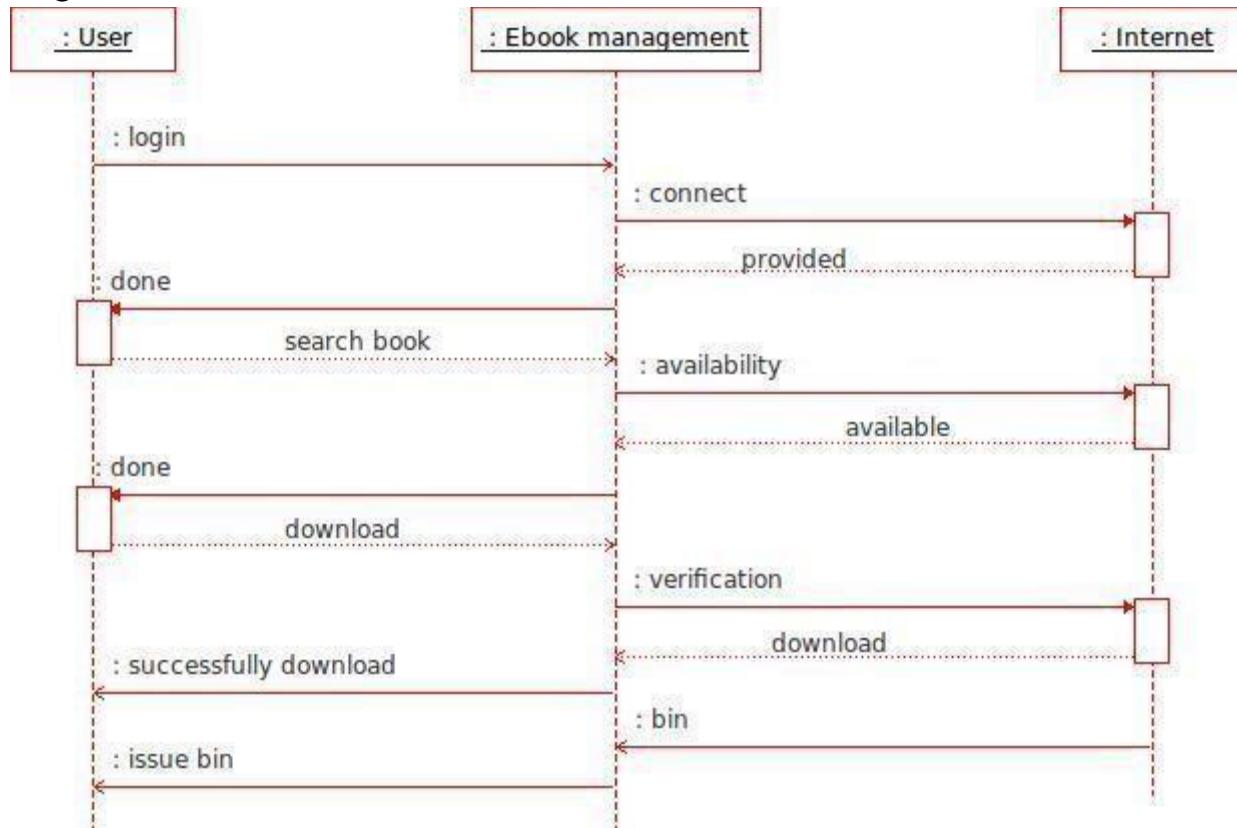
COLLABORATION DIAGRAM



ACTIVITY DIAGRAM



SEQUENCE DIAGRAM



OUTPUT:

Ebook_Management.java

```

/**
 * Class Ebook_Management
 */
public class Ebook_Management {

```

```

// Fields private int enterid;
private String verify; //
Constructors public
Ebook_Management () { };
// Methods
// Accessor methods
/**
 * Set the value of enterid
 * @param newVar the new value of enterid
 */ private void setEnterid ( int
newVar ) {
    enterid = newVar;
}
/**
 * Get the value of enterid
 * @return the value of enterid
 */ private int getEnterid (
) { return
    enterid;
}
/**
 * Set the value of verify
 * @param newVar the new value of verify
 */ private void setVerify ( String
newVar ) {
    verify = newVar;
}
/**
 * Get the value of verify
 * @return the value of verify
 */ private String getVerify
() { return
    verify;
} // Other methods public void
checkAvailability( )
{
}
}

```

RESULT:

Thus the “E-book management system” has been analysed & designed and the coding skeleton has been generated using Umbrello software

Receive

Info details

Inform

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to develop the communication gap between IT candidates and the Human Resources of various Multinational IT concerns.

4.OBJECTIVE:

The main objective is to seek employment with reputed Multinational Concerns with the candidates line of interest. In this project , Job Portal plays an important role to provide an efficient and beneficial Recruitment for job seekers and experienced person

5.PROBLEM STATEMENT:

In this Recruitment System, we assume our self as Applicant and proceed with the process steps as follows.

The applicant searching for the recruitment availability in job portal.

Job portal webpage search it in his DB which contains various list of organizations.

The organization attender collects all the form from different applicant and stores in organization DB.

HR short lists the applications according to their eligibility.

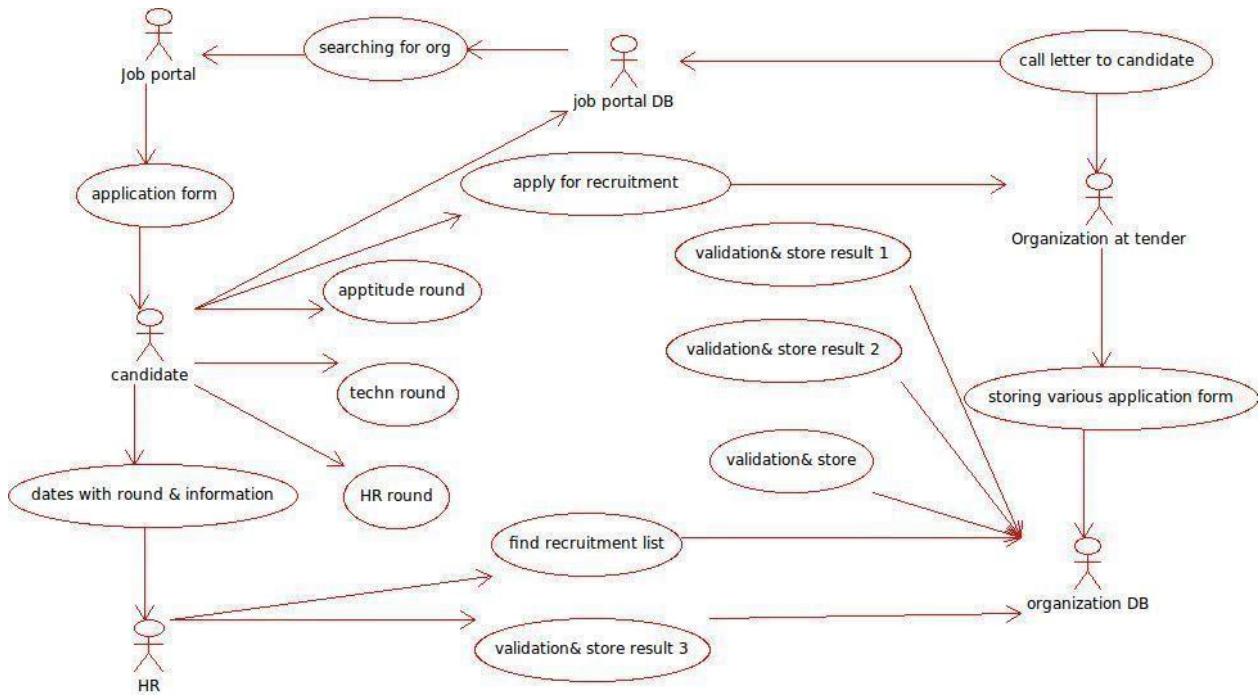
Then, dates with round information send to the short listed applicants.

The candidate attend the first round and if he selected, he will goes on further rounds.

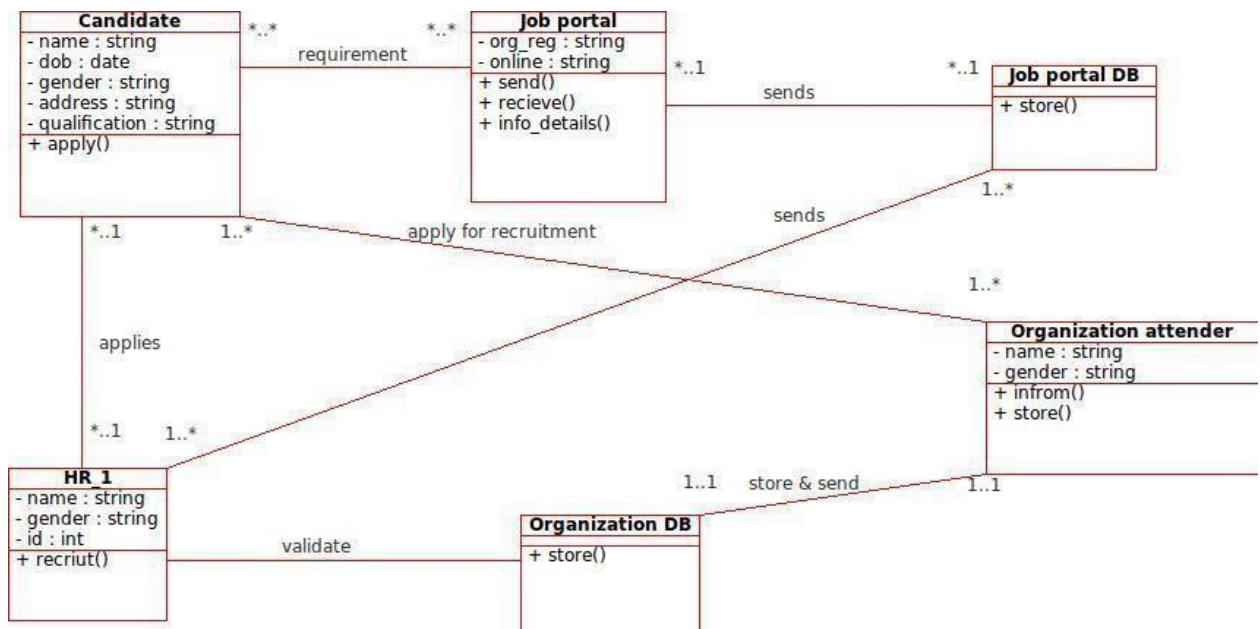
Finally, the call letter will be send to the selected applicants.

CS8582 - OOAD LABORATORY

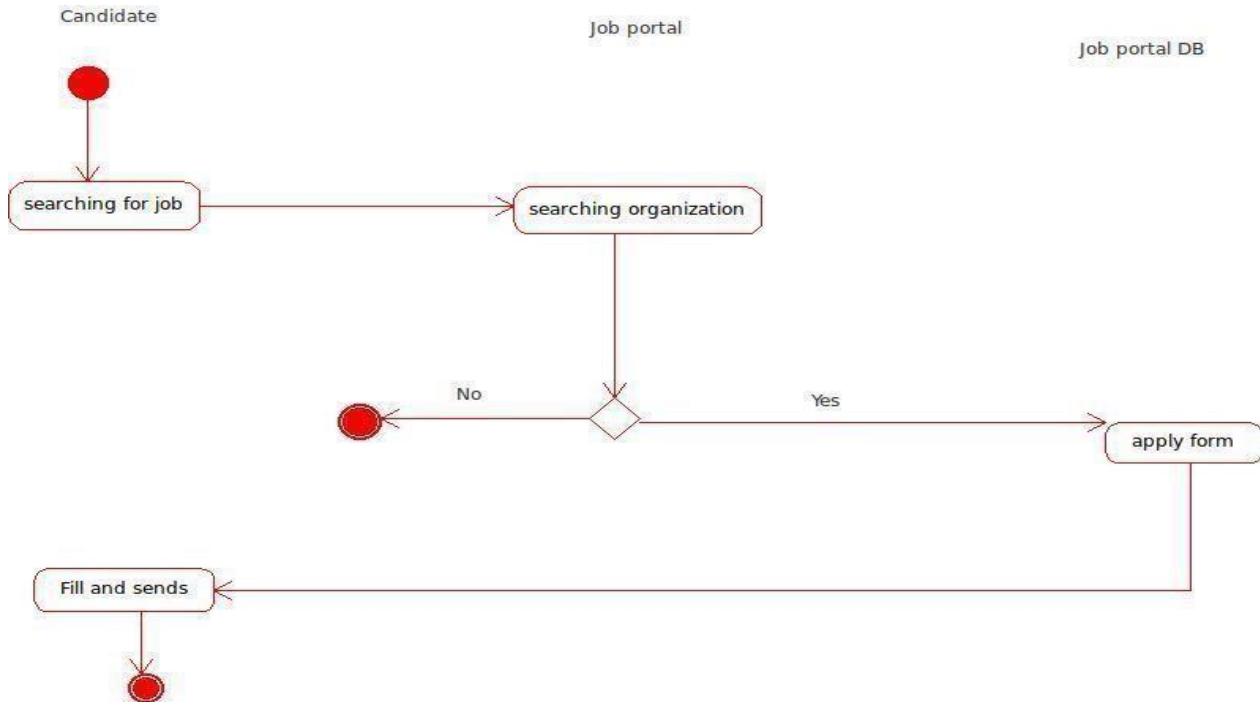
USE CASE DIAGRAM



CLASS DIAGRAM



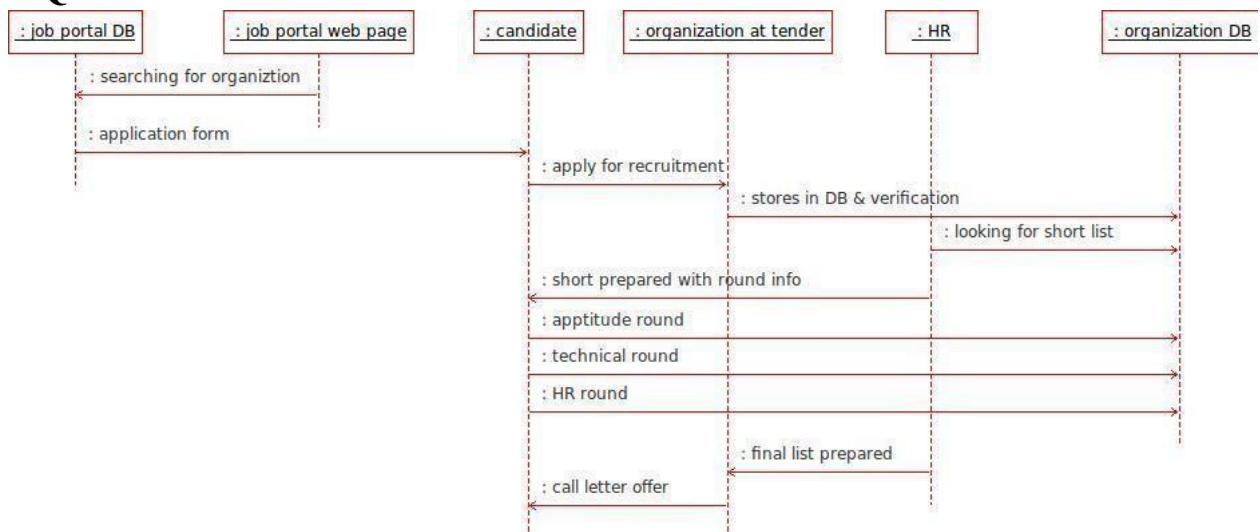
ACTIVITY DIAGRAM



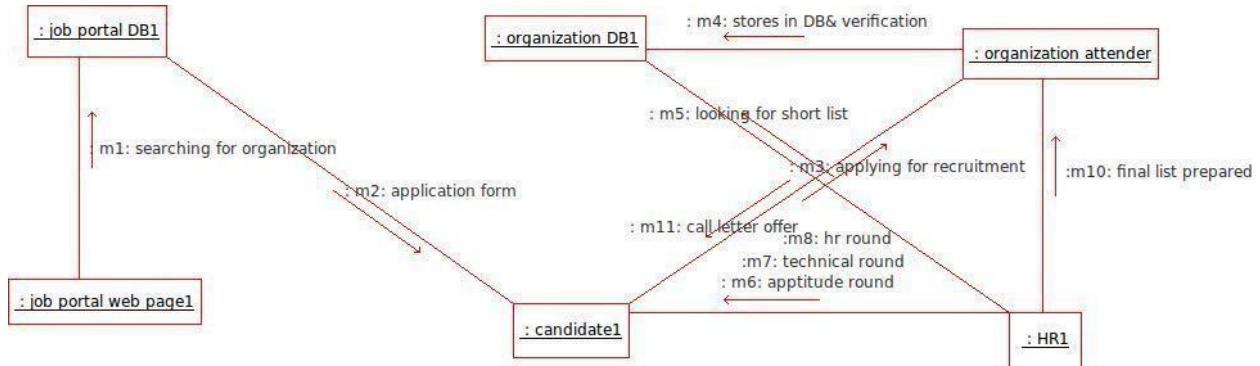
COMPONENT DIAGRAM



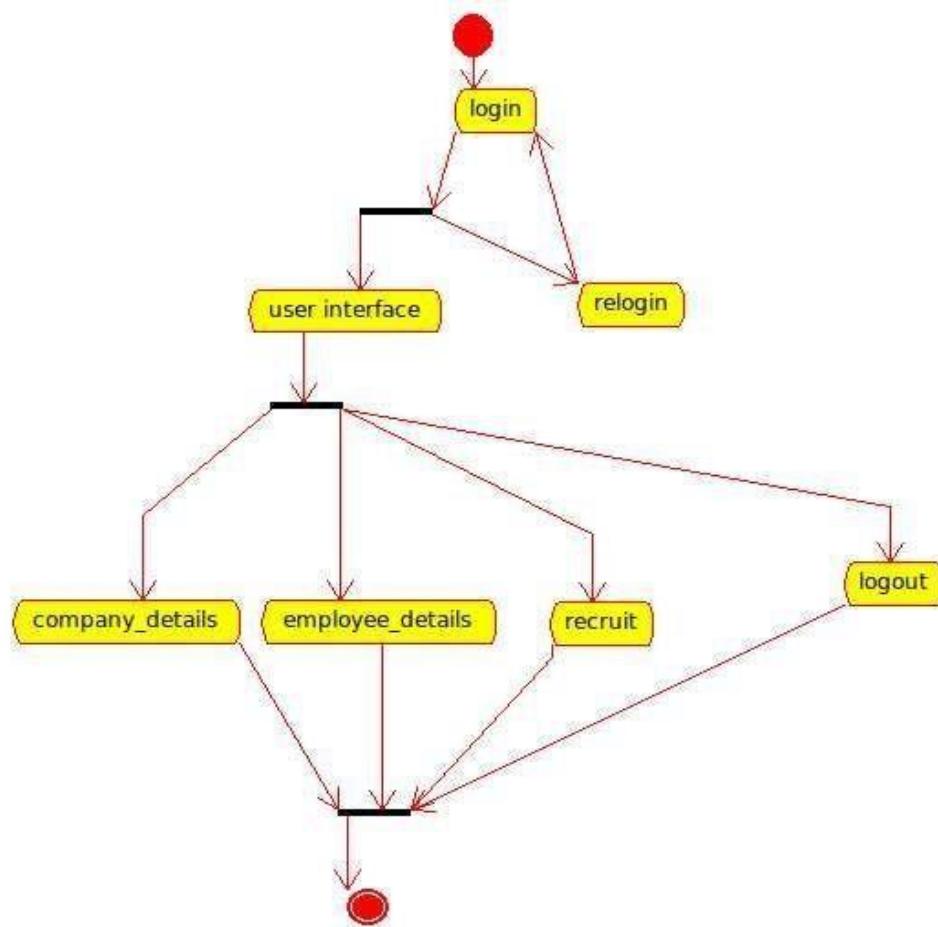
SEQUENCE DIAGRAM



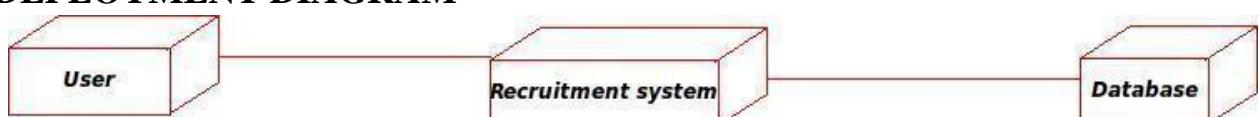
COLLABORATION DIAGRAM



STATE CHART DIAGRAM



DEPLOYMENT DIAGRAM



OUTPUT:

Passenger.java

```

import
java.util.*;
  
```

```

/**
* Class _class_module_passenger
*/
public class
_class_module_passenger {
// Fields private String
user_name; private int
password; private String
name; private int age;
private int phone_no;
private long
mlclassDebugID; //
Constructors public
_class_module_passenger () { };
// Methods
// Accessor methods
/**
* Set the value of user_name
* @param newVar the new value of user_name
 */ private void setUser_name ( String
newVar ) {
    user_name = newVar;
}
/**
* Get the value of user_name
* @return the value of user_name
 */ private String getUser_name (
) {
    return user_name;
}
/**
* Set the value of password
* @param newVar the new value of password
 */ private void setPassword ( int
newVar ) {
    password = newVar;
}
/**
* Get the value of password
* @return the value of password
 */ private int getPassword (
) { return
    password;
}

```

```

    }
    /**
 * Set the value of name
 * @param newVar the new value of name
 */ private void setName ( String newVar
) {
    name = newVar;
}
/**
 * Get the value of name
 * @return the value of name
 */ private String getName (
) { return
    name;
}
/**
 * Set the value of age
 * @param newVar the new value of age
 */ private void setAge ( int
newVar ) {
    age = newVar;
}
/**
 * Get the value of age
 * @return the value of age
 */ private int getAge ( ) { return
    age;
}
/**
 * Set the value of phone_no
 * @param newVar the new value of phone_no
 */ private void setPhone_no ( int newVar
) {
    phone_no = newVar;
}
/**
 * Get the value of phone_no
 * @return the value of phone_no
 */ private int getPhone_no (
) {
    return phone_no;
}

```

```

/**
 * Set the value of mlclassDebugID
 * @param newVar the new value of mlclassDebugID
 */ private void setMlclassDebugID ( long
newVar ) {
    mlclassDebugID = newVar;

}
/***
 * Get the value of mlclassDebugID
 * @return the value of mlclassDebugID
 */ private long getMlclassDebugID (
) {
    return mlclassDebugID;
}
// Other methods
 /**
 * public void
login( )
{
}
public void check_ticket( )
{
} public void
_get_classDebugID( )
{
}
}

```

RESULT:

Thus the “Recruitment system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 12	FOREIGN TRADING SYSTEM
Date:	

AIM:

To analyze and Design UML Diagrams for Foreign trading system using Umbrello Software.

SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

ANALYSIS:

1. Identifying Actors:

broker	buyer customs	sells
forex	consumer bank database	

2.2. Identifying Use Cases:

willingness & get	pay & payment
price notification & fix	validating price
price get product send	market value
goods updation &	
validation	

2.3. Identifying Classes:

database	forex consumer
buyers broker	bank
	seller

customs

2.4. Identifying Attributes:

pdt_id	ph_no
currency_name	agency_name
pdt_name	number cash amount

2.5. Identifying Methods:

Trade Send

Conversion

Ordered

Send
Check

Get pdt_id

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to provide a exhaustive, flexible and reliable stock maintenance system which would be beneficial for both stock manager and retailer

4.OBJECTIVE:

The main objective of designing and developing a Foreign trading system is to provide with a system which proves to be manually beneficial i.e., benefits both the buyer and the seller through an broker. The complete activities and the process right from product's manufacture till the product gets sold out, every single activity is incorporated in this system.

5.PROBLEM STATEMENT:

Foreign Trading System is a real time application used in the merchant's day to day system. This is a database to store the transaction that takes places between the Buyer seller and the Broker that includes Product selling and buying with reference to the Broker. Here we assume our self as the Broker and proceed with the transaction as follows:

1.The Seller is the producer of the items and it contains the necessary information of the item such as price per item, Date of manufacture, best before use, Number of tem available and their Company Address.

2. The Broker is the secondary source of an Item and he purchases Item from

the Seller by requesting the required Item with its corresponding Company Name and the Number of Items required. The Broker is only responsible for distribution of the Item to the Buyer in the Town or City.

CS8582 - OOAD LABORATORY

3. The Buyer is the one who is prime source for Buying items in the market. The Buyer get Item from the Broker and not directly from the Seller.

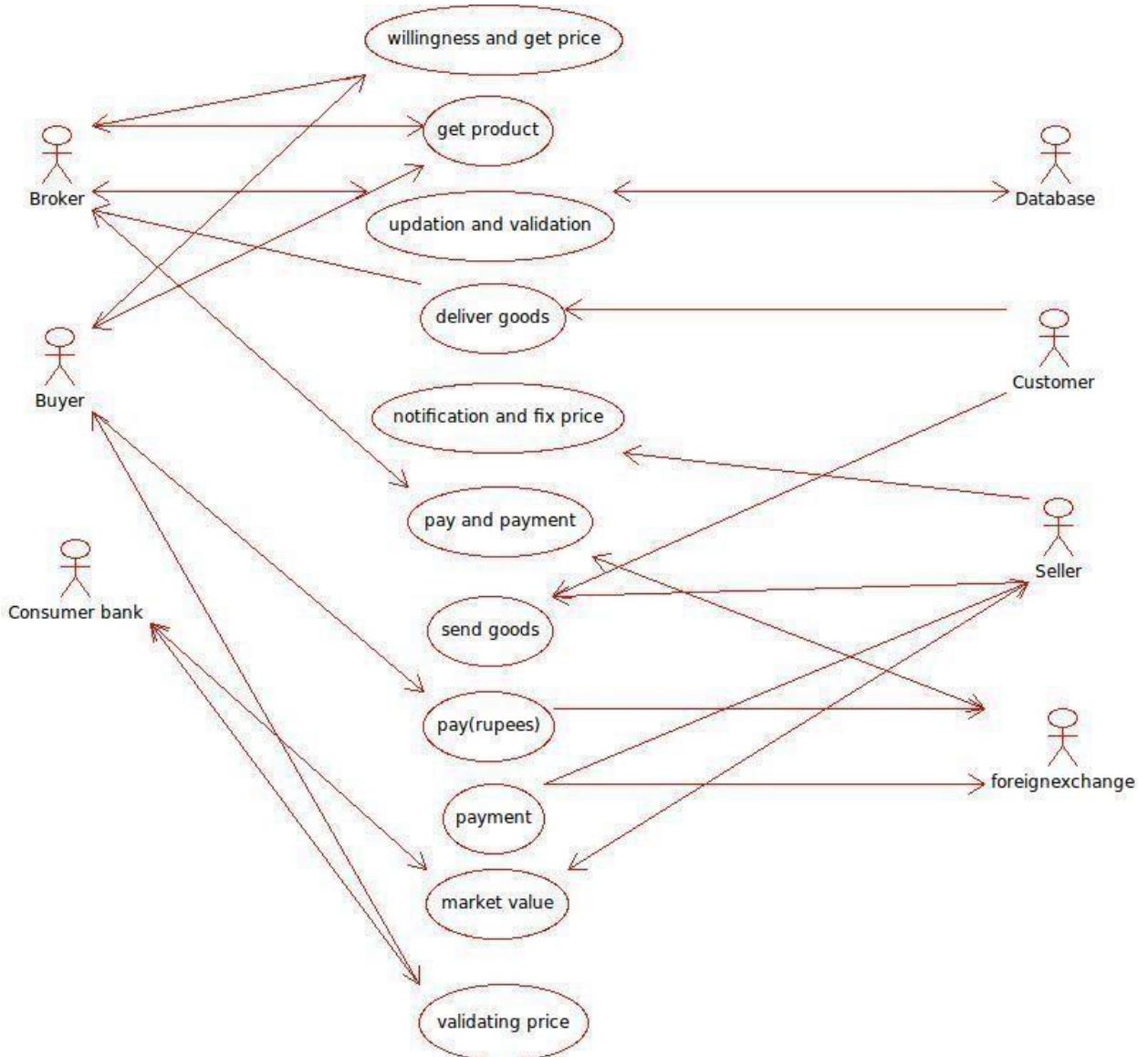
4. The Product availability is the database used in our System which records all transactions that takes place between the Broker, the buyer and the Seller.

5. Forex is the system where the buyer or broker pay the money for the product in their own money and it will paid in the form of what the seller cash.

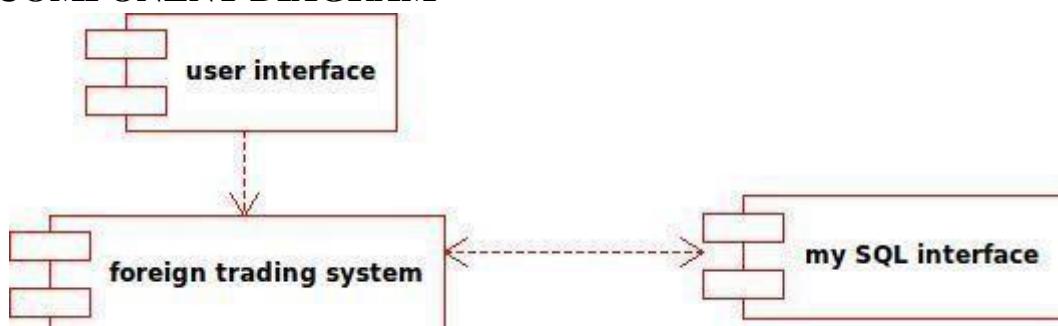
6. Customs is responsible for trading the product all over the world. If they give

approval the only we accept to trade.

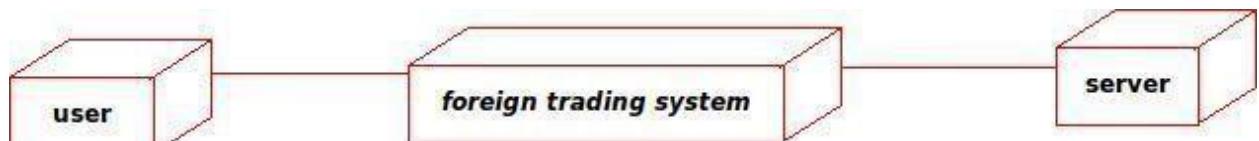
USE CASE DIAGRAM



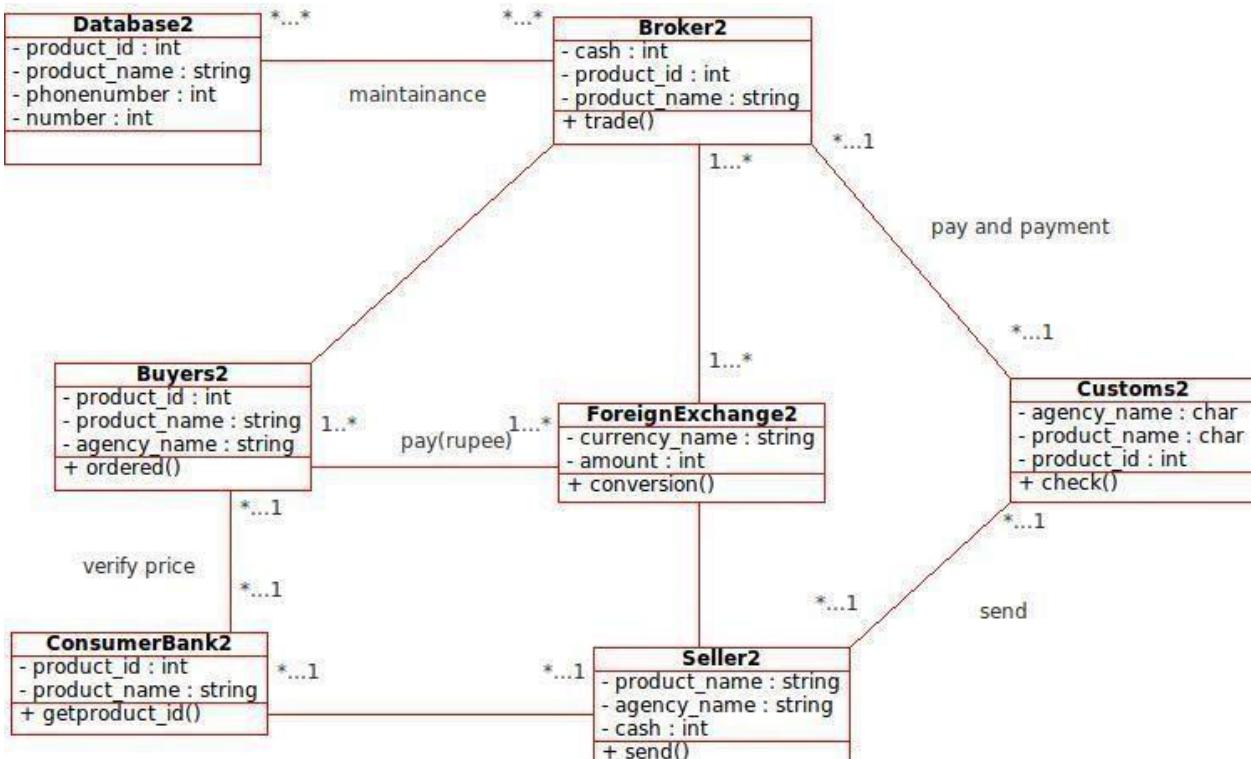
COMPONENT DIAGRAM



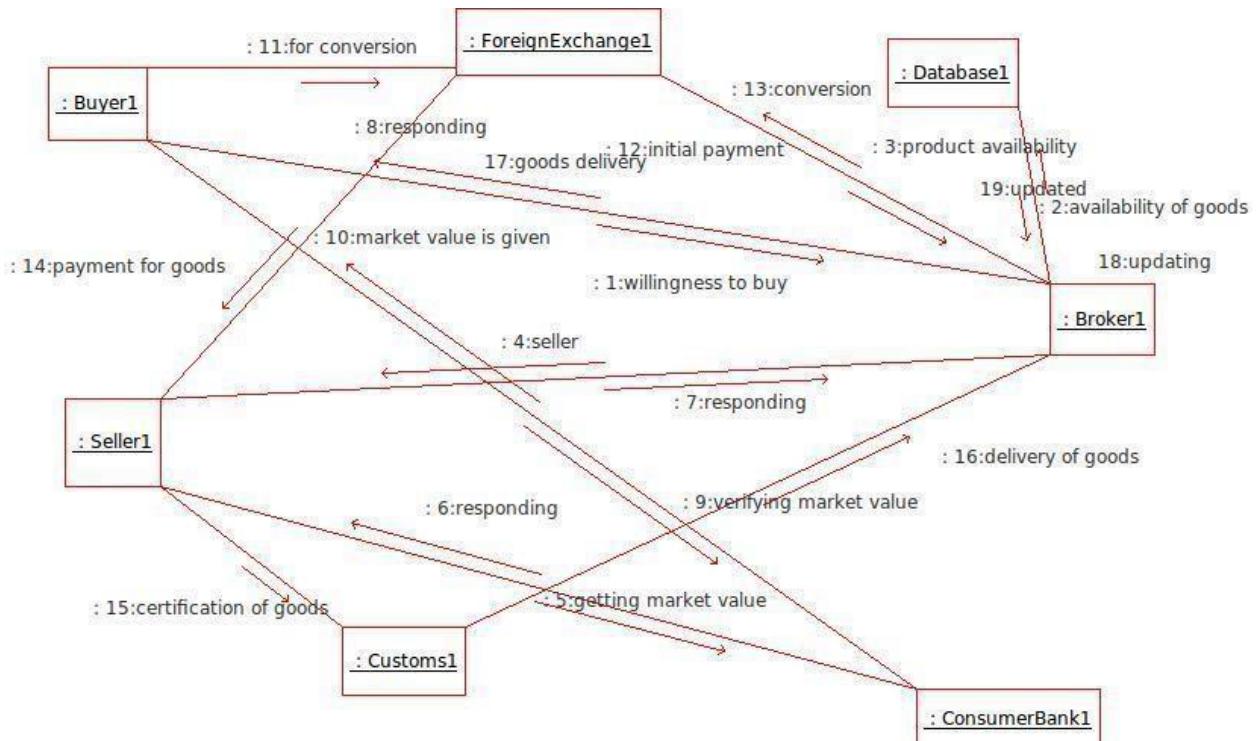
DEPLOYMENT DIAGRAM



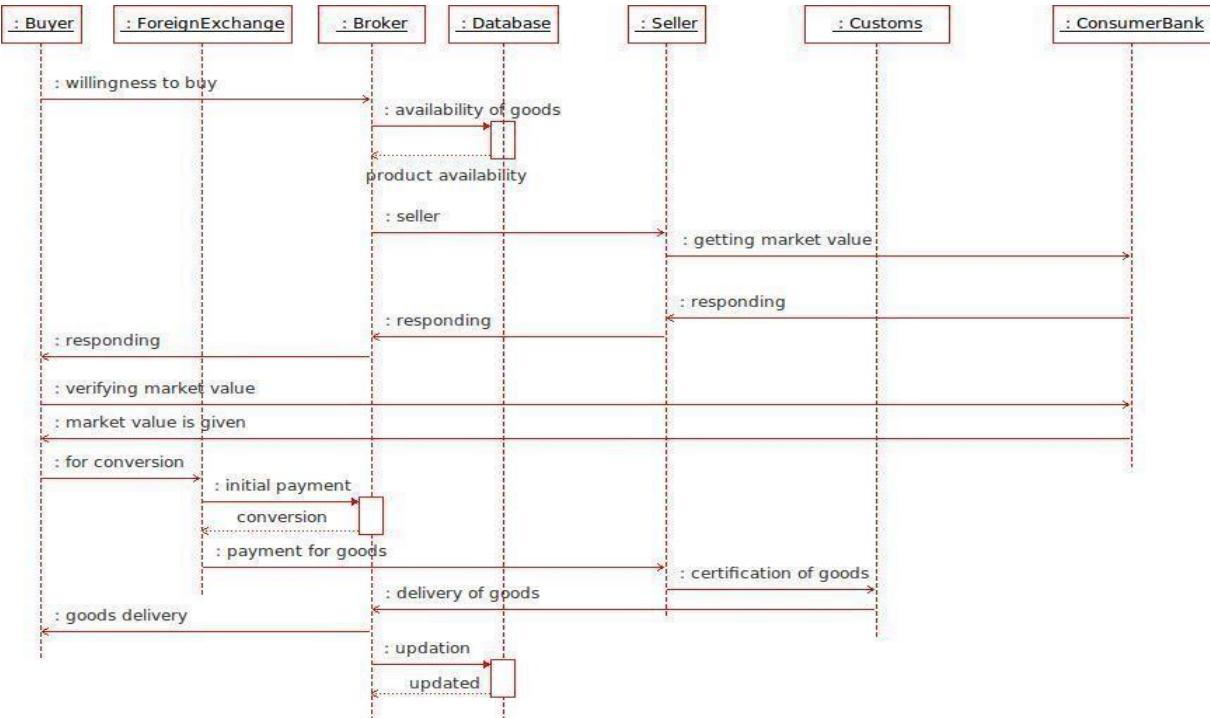
CLASS DIAGRAM



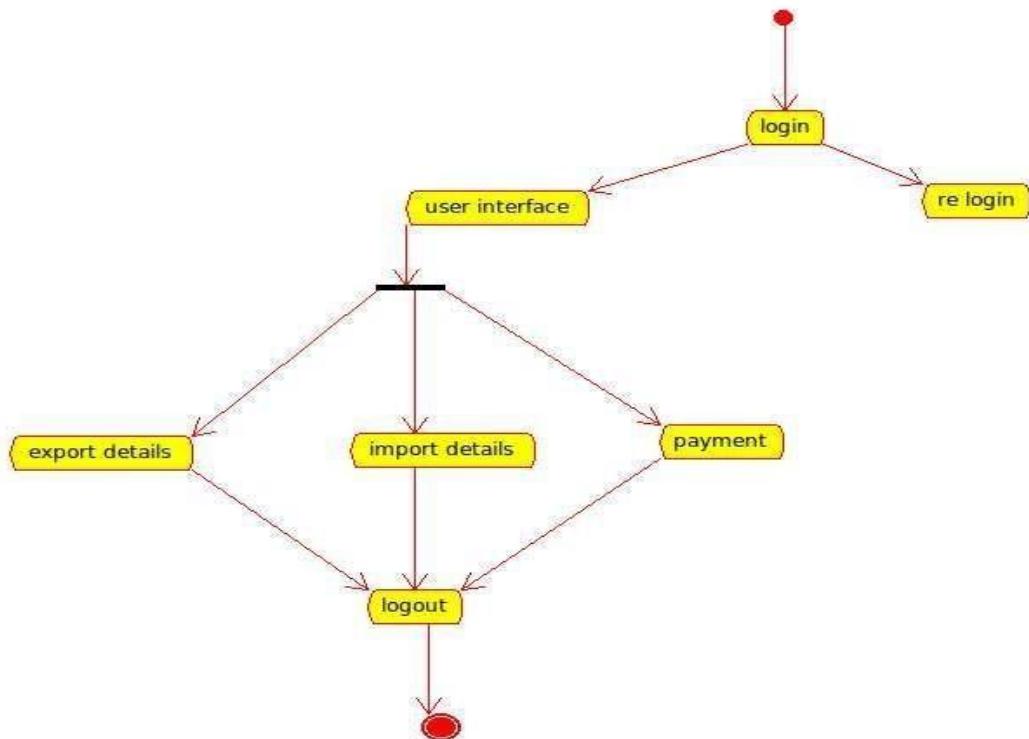
COLLABORATION DIAGRAM



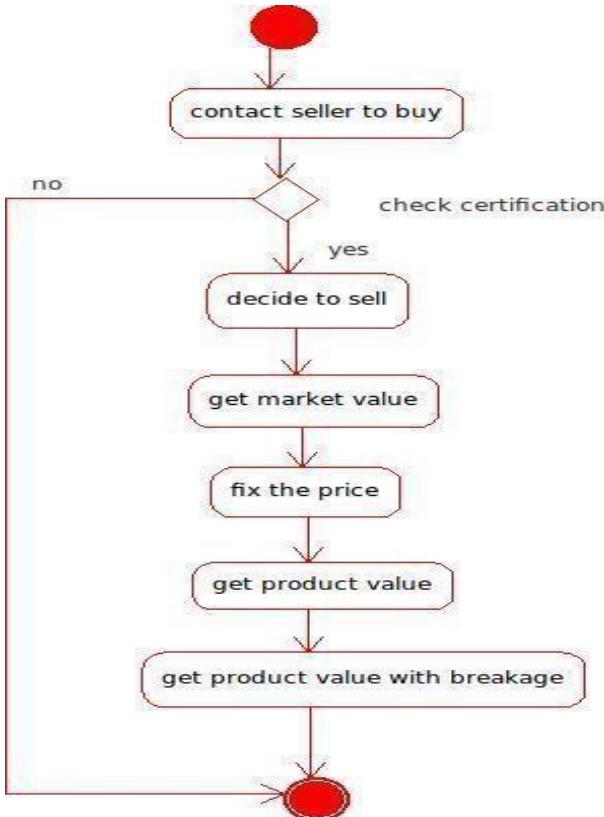
SEQUENCE DIAGRAM



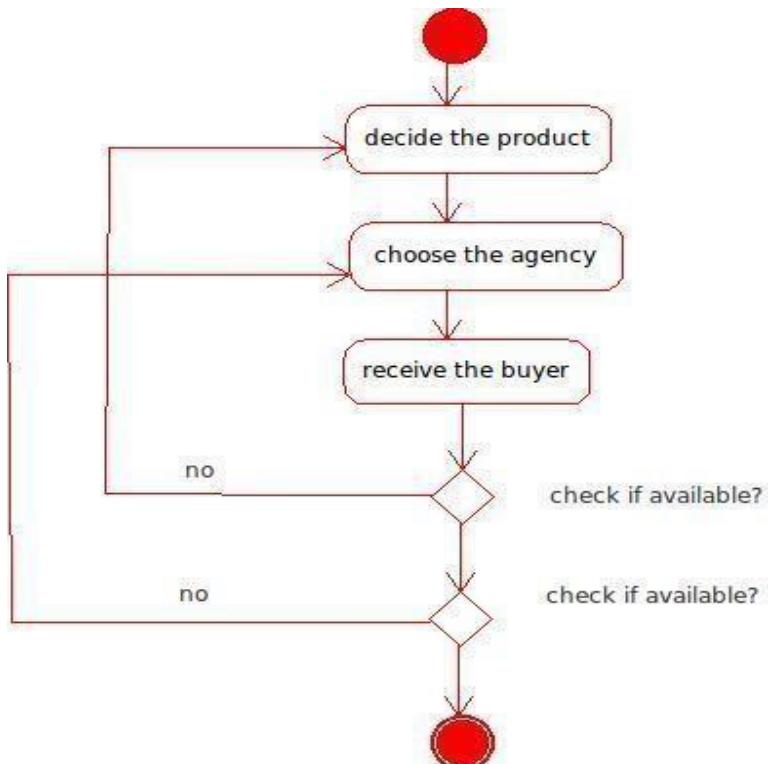
STATE CHART DIAGRAM



ACTIVITY DIAGRAM FOR PRODUCT AVAILABILITY



ACTIVITY DIAGRAM TO GET THE MARKET PRICE OF THE MARKET



OUTPUT: BankDb.java

```

import java.util.*;
/**
 * Class BankDb
  
```

```
 */ public class  
BankDb {  
    // Fields private int bankid;  
    private String bankname;  
    private int customerid;  
    private String customername;  
    // Constructors public  
    BankDb () { };//  
    Methods  
    // Accessor methods  
    /**  
     * Set the value of bankid  
     * @param newVar the new value of bankid
```

```

 */ private void setBankid ( int
newVar ) {
    bankid = newVar;
}
/***
* Get the value of bankid
* @return the value of bankid
 */ private int getBankid (
) { return
    bankid;
}
/***
* Set the value of bankname
* @param newVar the new value of bankname
 */ private void setBankname ( String
newVar ) {
    bankname = newVar;
}
/***
* Get the value of bankname
* @return the value of bankname
 */ private String getBankname (
) {
    return bankname;
}
/***
* Set the value of customerid
* @param newVar the new value of customerid
 */ private void setCustomerid ( int
newVar ) {
    customerid = newVar;
}
/***
* Get the value of customerid
* @return the value of customerid
 */ private int getCustomerid (
) { return
    customerid;
}
/***

```

```
* Set the value of customername
* @param newVar the new value of customername*/
private void setCustomername (
    String newVar ) { customername
        = newVar;
}
/***
* Get the value of customername
* @return the value of customername
*/
private String getCustomername (
) {
    return customername;
}
// Other methods
public void ticket( )
{
}
public void update( )
{
}
}
```

RESULT:

Thus the “Foreign Trading system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 13	CONFERENCE MANAGEMENT SYSTEM
Date:	

AIM:

To analyze and Design UML Diagrams for Conference Management System using Umbrello Software.

SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

ANALYSIS:

1. Identifying Actors:

Database	Reviewers
Coordinator	author
Selection committee	
Panel	

2.2. Identifying Use Cases:

advertisement update	forward the paper
reg & sends the	sends the paper
abstract forward the	
abstract request the	intimates
paper	

3. Identifying Classes:

coordinator	Panel_mem
DB	Selection_commitee

Participant Reviewing

4. Identifying Attributes:

Contacts
Paper
Money
Pin
Abstracts

Access_control
Data
Rules
List time

REGISTER

NO:110819104501 5.

NAME: MUKESH.V

Identifying Methods:

registers
intimates
listen sends
issue_prize
question

advertise
select forwarding
validate

6. Identifying Relationship: Association

PROJECT SCOPE:

The main scope of the project is to issue the prizes and national level certificates.

OBJECTIVE:

The main objective of designing and developing a Conference Management System is to provide with a system which proves to be manually beneficial i.e., benefits both the participant and the coordinator. The complete activities and the process is to present the papers and issue the prizes.

PROBLEM STATEMENTS:

The coordinator of the conference spreads the news that the conference is to be held through the advertisements.

2. The Author or the Participant after seeing the advertisement will register and then send their paper.
3. The Conference Coordinators collect the registered papers and forwards those papers to the selection committee.
4. The various selection committees split the work and selects the paper and combines all the selected papers.
5. The selected papers are passed to the reviewers.
6. The reviewers will recheck the selected paper and if required short down list.

- After the reviewing process, the reviewing panel members rank the papers and reports to the conference coordinators.

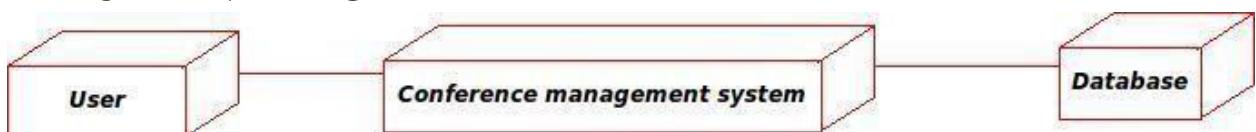
CS8582 - OOAD LABORATORY

8. The coordinator updates the database with the details of the author whose papers have been selected.
 9. after updating the coordinator intimates the author that their papers has been selected.

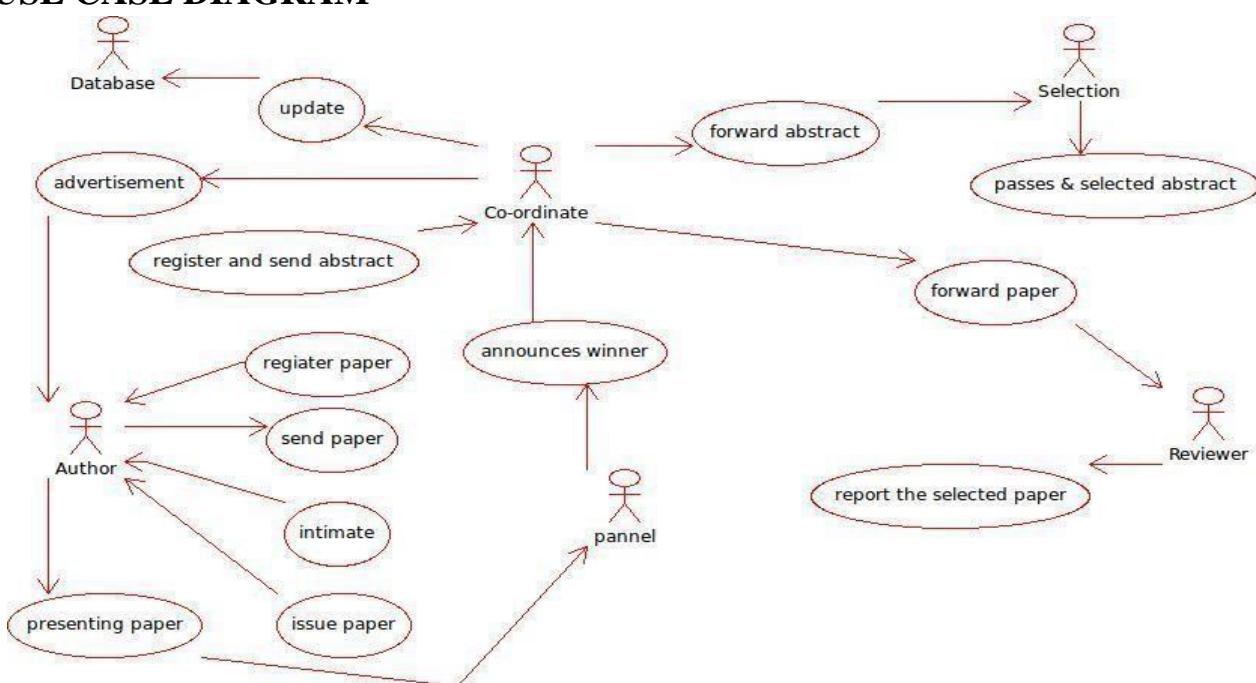
COMPONENT DIAGRAM



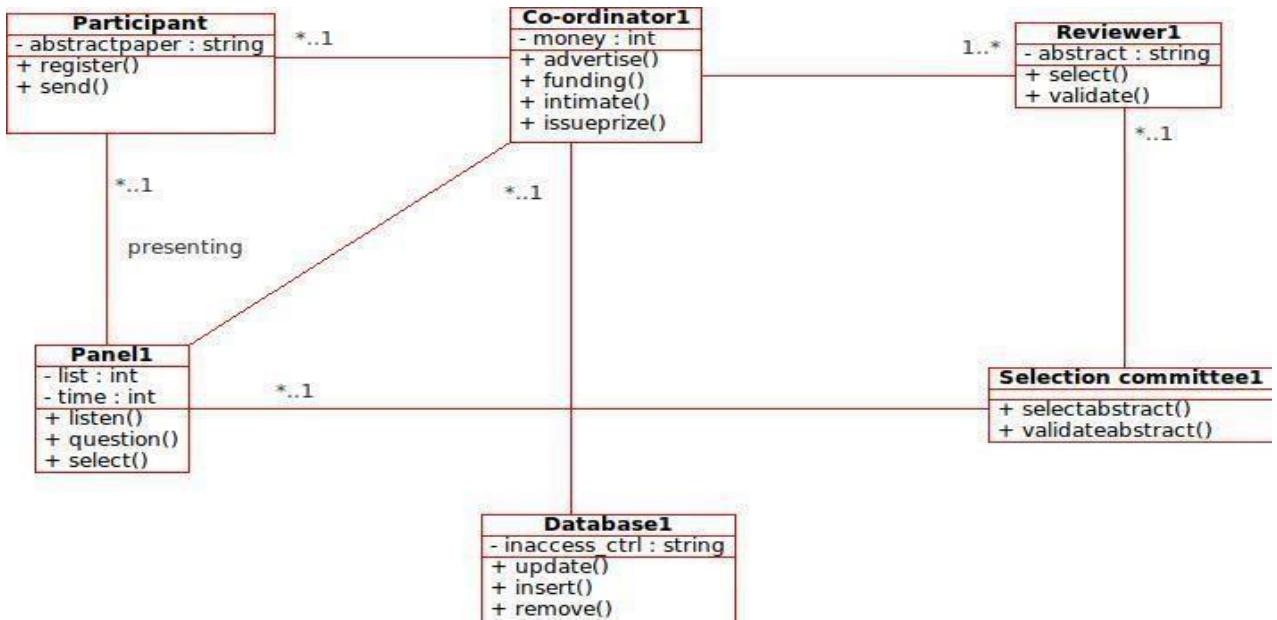
DEPLOYMENT DIAGRAM



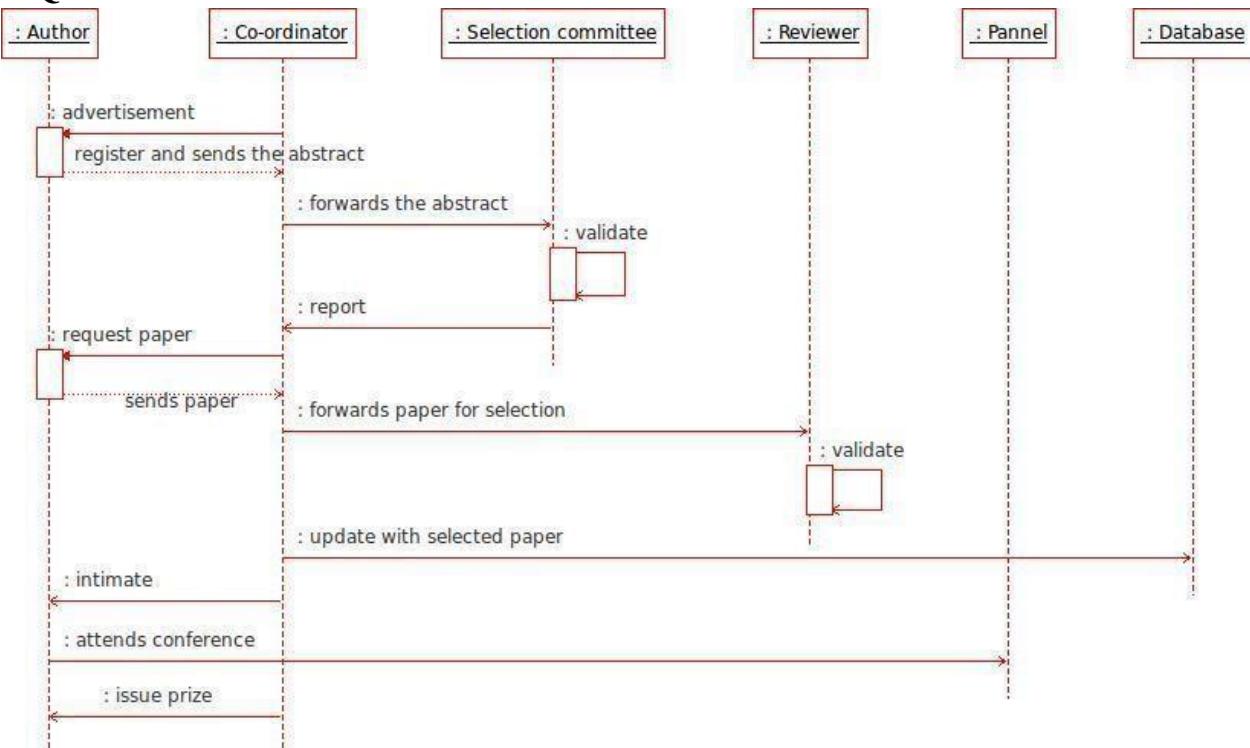
USE CASE DIAGRAM



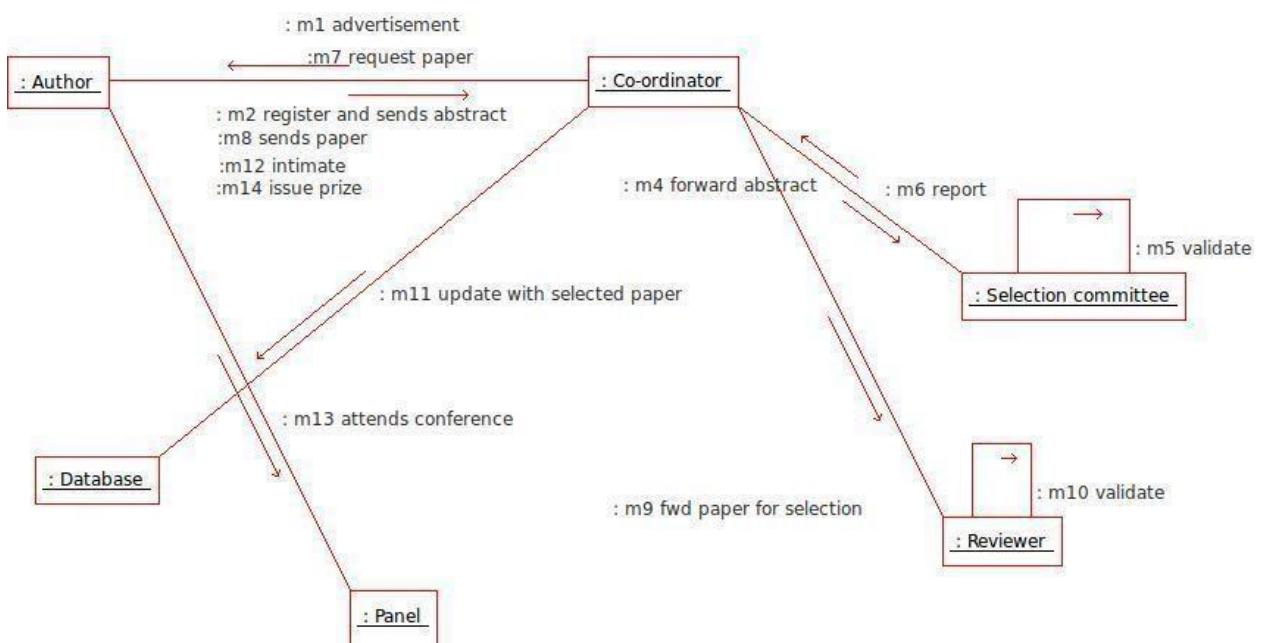
CLASS DIAGRAM



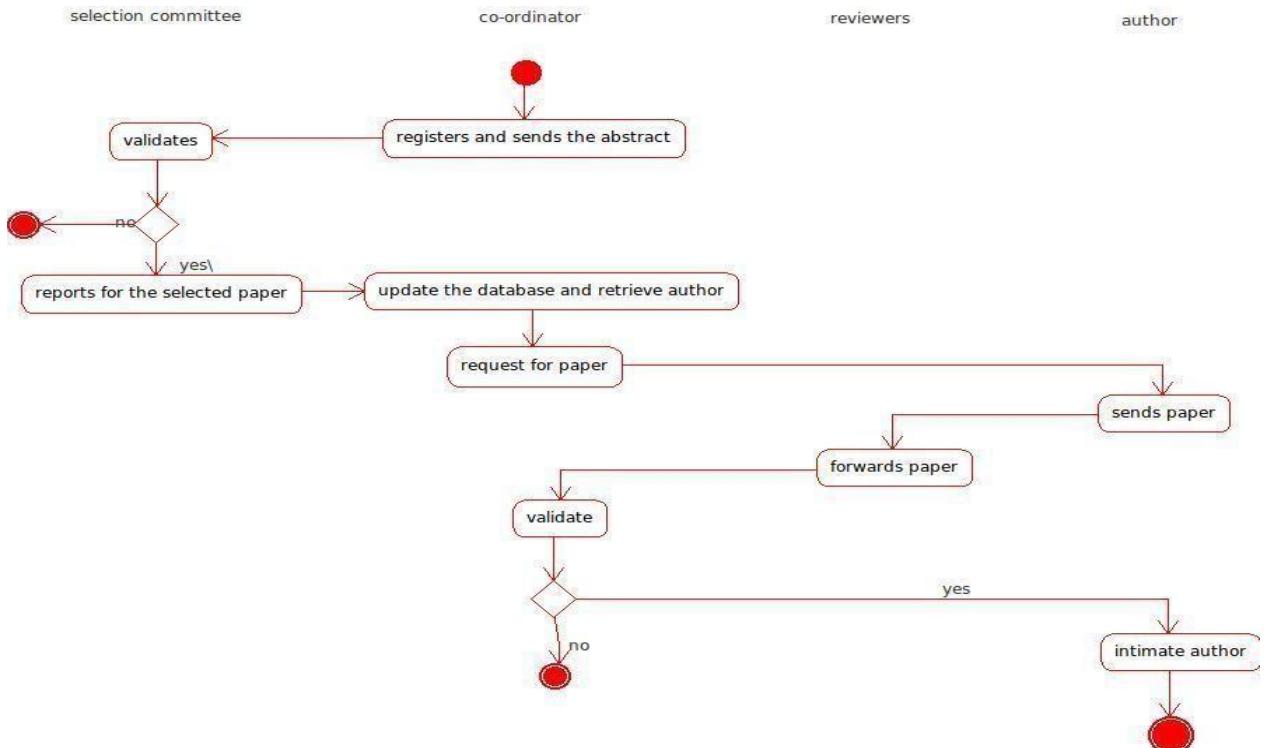
SEQUENCE DIAGRAM



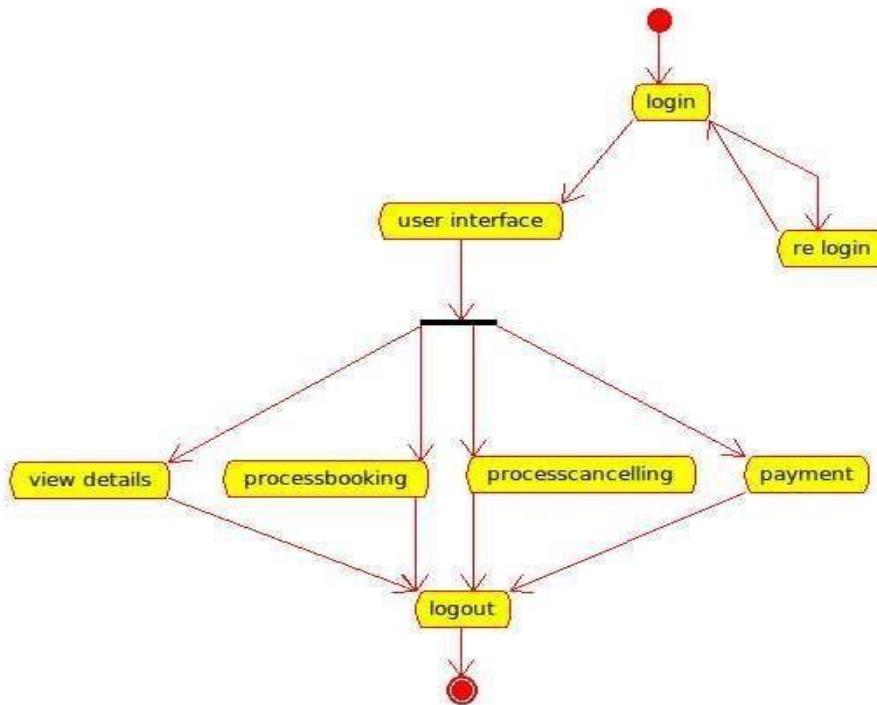
COLLABORATION DIAGRAM



ACTIVITY DIAGRAM



STATE CHART DIAGRAM



OUTPUT:

Reviewer1.java

```

import java.util.*;
/**
 * Class Reviewer1
 */
public class Reviewer1
{
    // Fields
    private String abstract;

    // Constructors
    public Reviewer1 () { };

    // Accessor methods
    /**
     * Set the value of abstract
     * @param newVar the new value of abstract
     */
    private void setAbstract ( String newVar ) {
        abstract = newVar;
    }

    /**
     * Get the value of abstract
     * @return the value of abstract
     */
    private String getAbstract (
    ) { return
        abstract;
    }

    // Other methods
    public void select( )
    {

```

```
}

public void validate( )
{
}
}
```

RESULT:

Thus the “Conference management system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 14
Date:

BPO MANAGEMENT SYSTEM

AIM:

To analyze and Design UML Diagrams for BPO Management system using Umbrello Software.

1.SOFTWARE REQUIREMENT SPECIFICATION:

SYSTEM REQUIREMENT:

Processor (32-bit Intel)

RAM: 256MB

Hard Disk: 20GB

SOFTWARE REQUIREMENT:

Operating System: GNU Linux Case

Tool: Umbrello

2.ANALYSIS:

2.1. Identifying Actors:

Company	Feedback analyst
Team leader	BPO org
Customer	Non –voiced employee
Database	
Voiced employee	

2. 2Identifying Use Cases:

Provides pdt detail	Employee convey soln
Submits team report	Enter cust info
Provide customer details	Check for query & obtain soln
Employee evaluation	Feedback to DB
Cust reports pbm	
Get employee performance	

2.3.

Identifying Classes:

Customer	BPO
Employee	

2.4. Identifying Attributes:

Cust_id	Emp_id
Emp_type	Problem
Pdt_id	Emp_performance

2.5. Identifying Methods:

Send_details	Record
Get_soln	Find_soln
Get_info	Set_info

Send_soln

Get_feedback

2.6. Identifying Relationship: Association

3.PROJECT SCOPE:

The main scope of the project is to provide the customer with an easy solution to troubleshoot their problems by direct contact with the company. It also benefits the company as it can divide its responsibilities and hire external agents(BPO) to take care of those responsibilities.

4.OBJECTIVE:

The main objective of designing and developing a BPO Management system is to provide with a system which proves to be manually beneficial i.e., benefits both the product or service providing company and the customer. The complete activities and the process right from product's sale till the product's query solving activities is incorporated in this system.

5.PROBLEM STATEMENT:

Query solving system is a real time application used in the BPO's day to day system. This is a database to store the problem statements that occur for the customer and the solution for it includes customer and product details reference to the sale. Here we assume ourselves as the BPO:

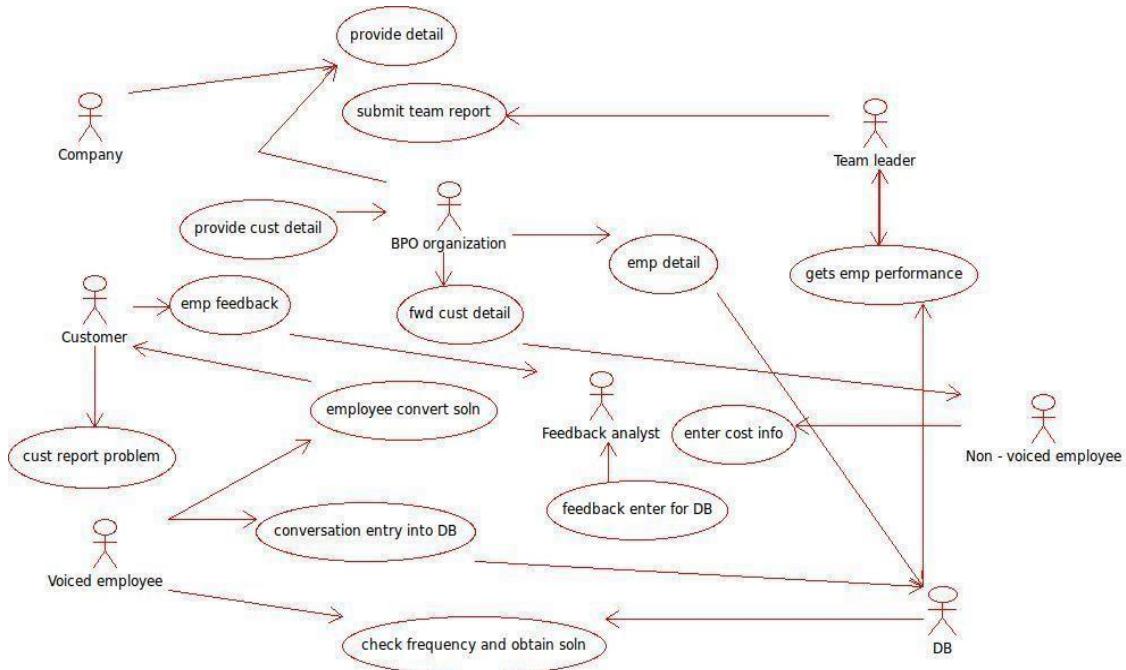
1.The company is the producer of the items and it contains the necessary information of the item such as product id.

2. The customer purchases item from the company by requesting the required item. The company is only responsible for distribution of the item to the customer.

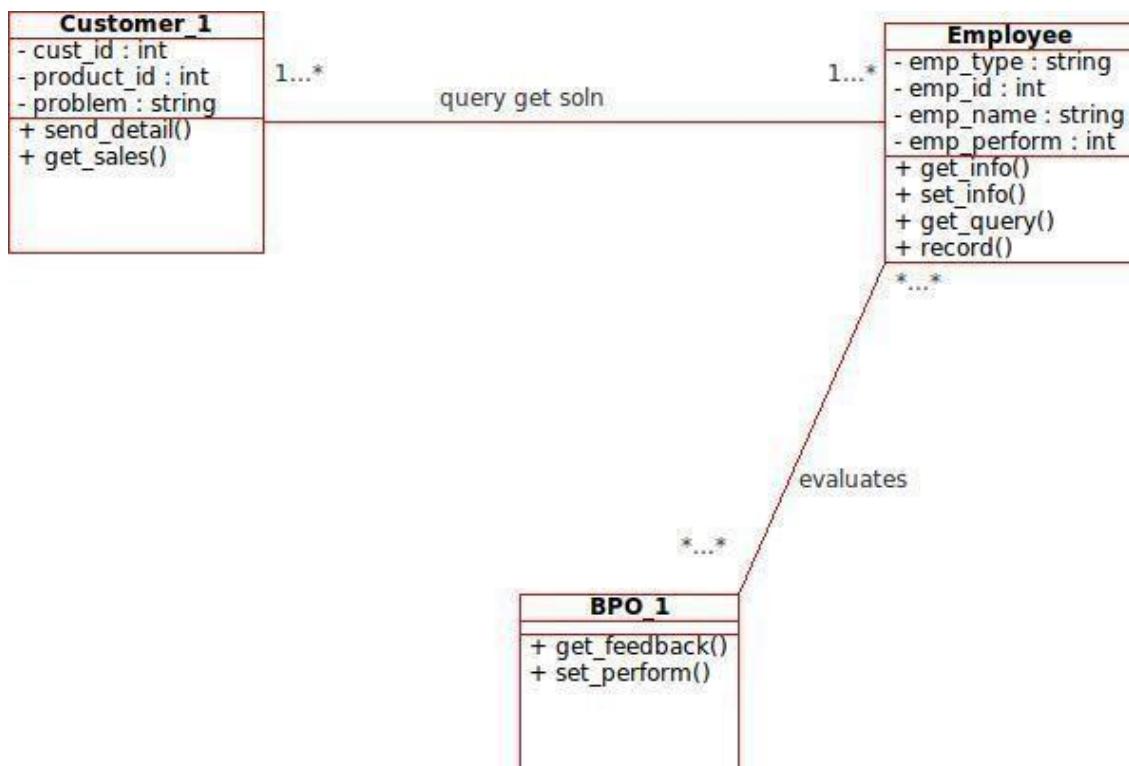
3. The BPO is mainly responsible for interacting with the customer and providing him with the required solution. The customer gets item from the company and not BPO.

4. The company provides the BPO with the product details and customer details. When the customer contacts the BPO with query, the BPO accesses this information to provide the customer with the respective solution.

USECASE DIAGRAM



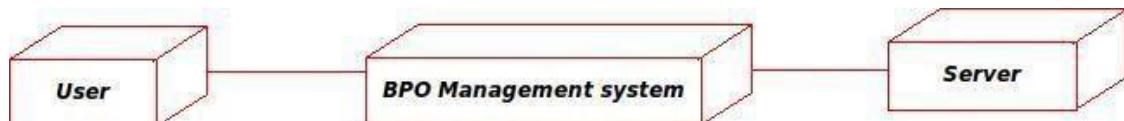
CLASS DIAGRAM



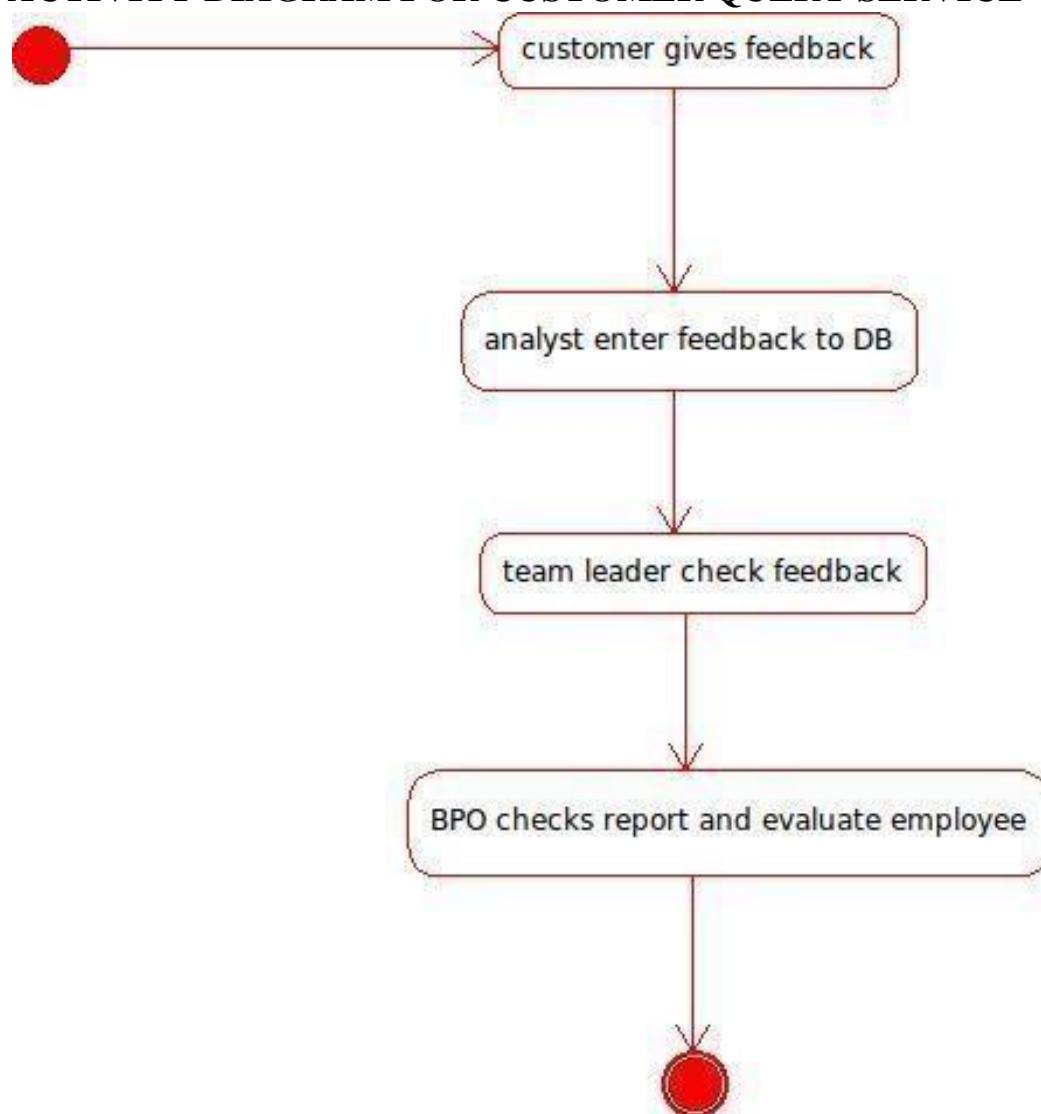
COMPONENT DIAGRAM



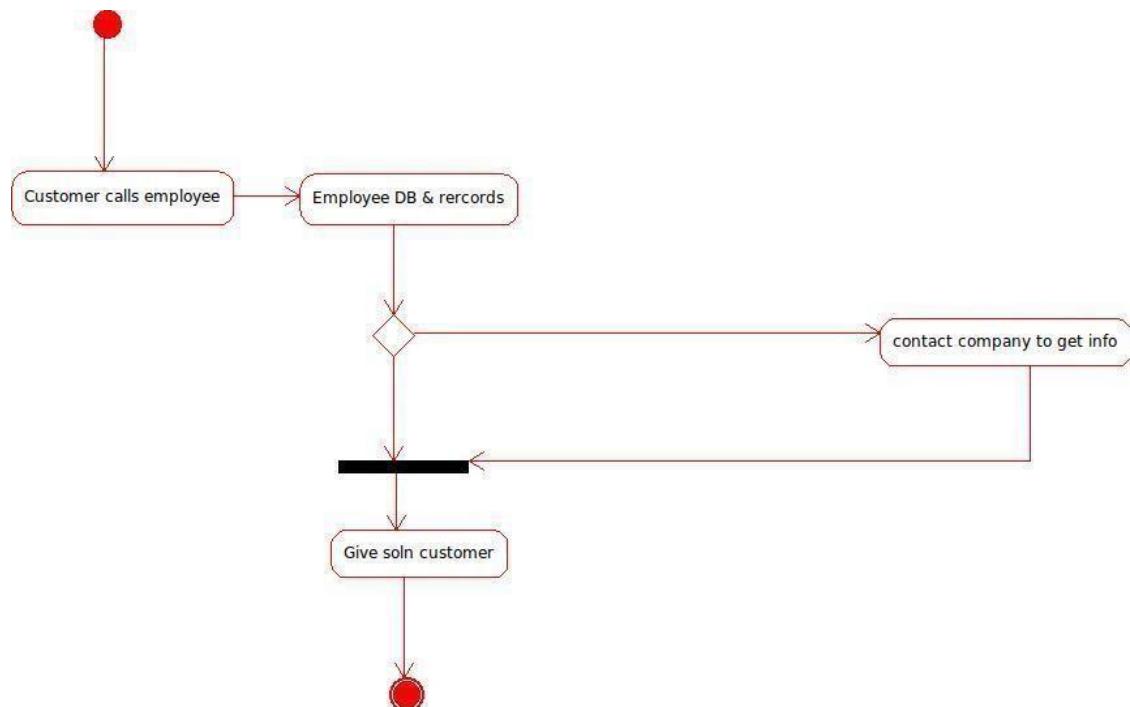
DEPLOYMENT DIAGRAM



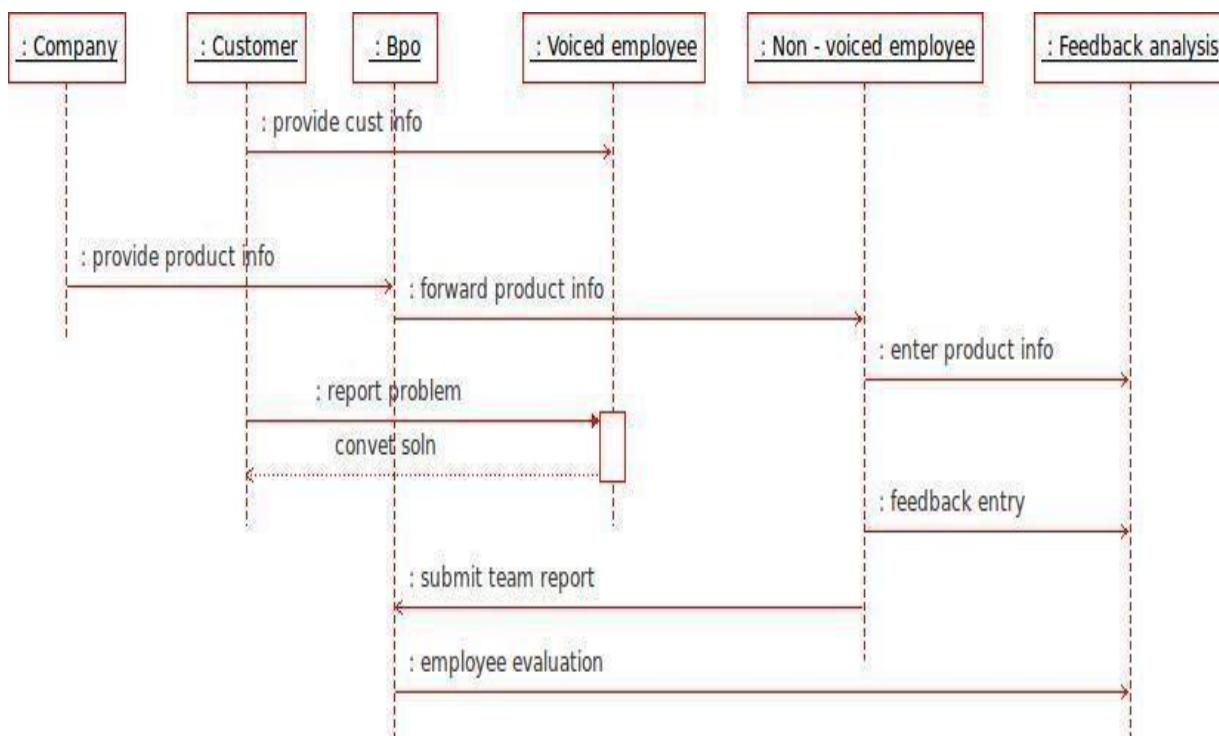
ACTIVITY DIAGRAM FOR CUSTOMER QUERY SERVICE



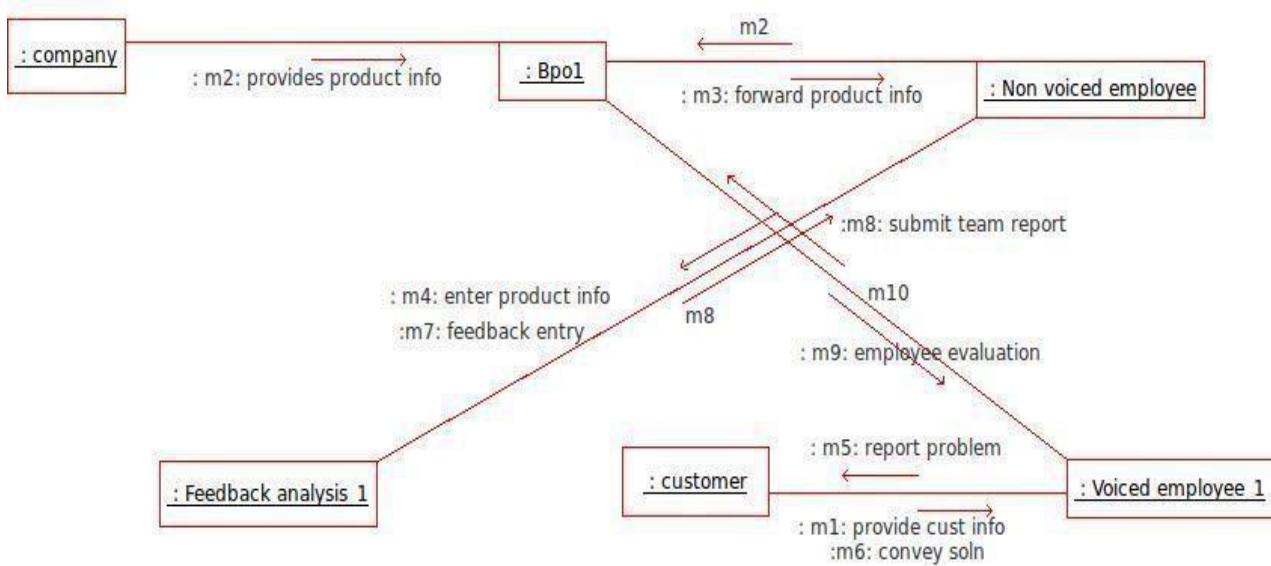
ACTIVITY DIAGRAM FOR EMPLOYEE EVALUATION



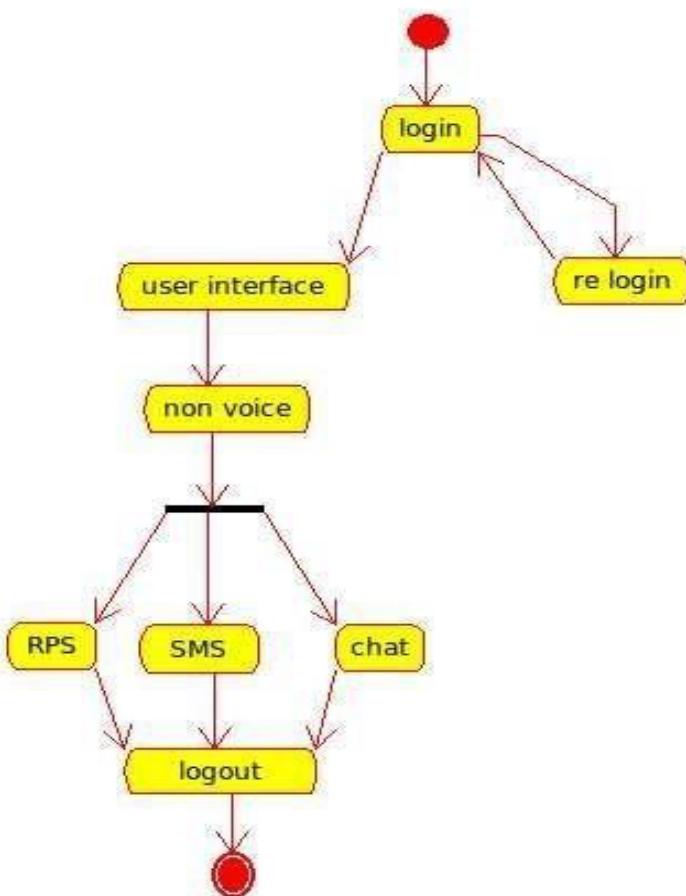
SEQUENCE DIAGRAM



COLLABORATION DIAGRAM



STATE CHART DIAGRAM



OUTPUT:

BPO_1.java

```

import java.util.*;
/**
 * Class BPO_1
  
```

Expt No: 15	
Date:	

LIBRARY MANAGEMENT SYSTEM

```
*/
public
class
BPO_1 {
```

```
//Fields // Constructors
public BPO_1 () { };
// Methods
// Accessor methods
// Other methods public void get_feedback(
)
{
}
public void set_perform( )
{
}
}
```

RESULT:

Thus the “BPO management system” has been analysed & designed and the coding skeleton has been generated using Umbrello software.

AIM:

To analyze and Design UML Diagrams for Library Management system using Umbrello Software.

PROJECT SCOPE:

The purpose of the project is to maintain the details of books and library members of different libraries.

The main purpose of this project is to maintain a easy circulation system between clients and the libraries, to issue books using single library card, also to search and reserve any book from different available libraries and to maintain details about the user

OBJECTIVE:

The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- A separate column for digital library
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- A teacher login page where teacher can add any events being organized in the college and important suggestions regarding books. ■ Online notice board about the workshop

PROBLEM STATEMENT:

Library Management System is an application which refers to library systems which are generally small or medium in size.

It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc.

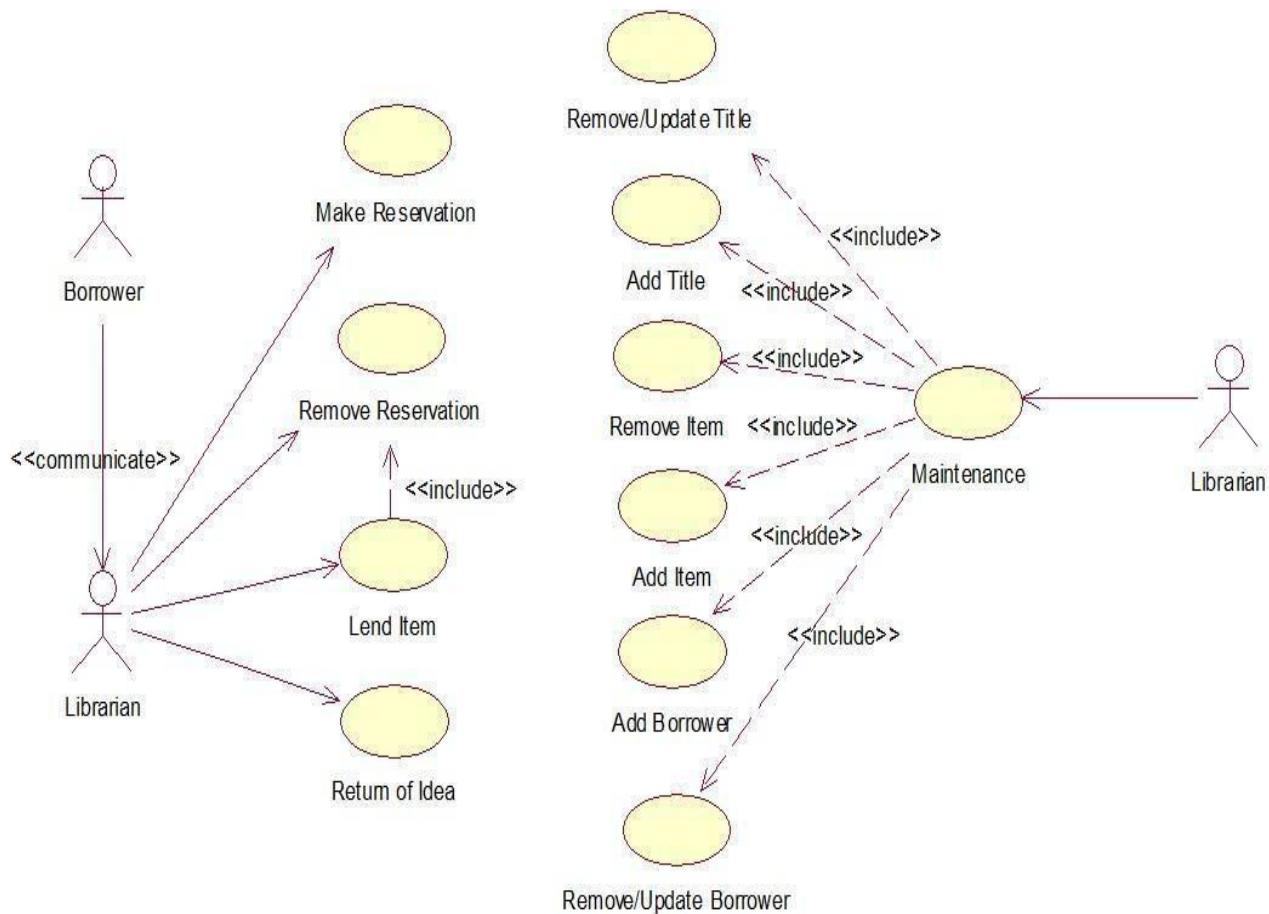
Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains.

With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

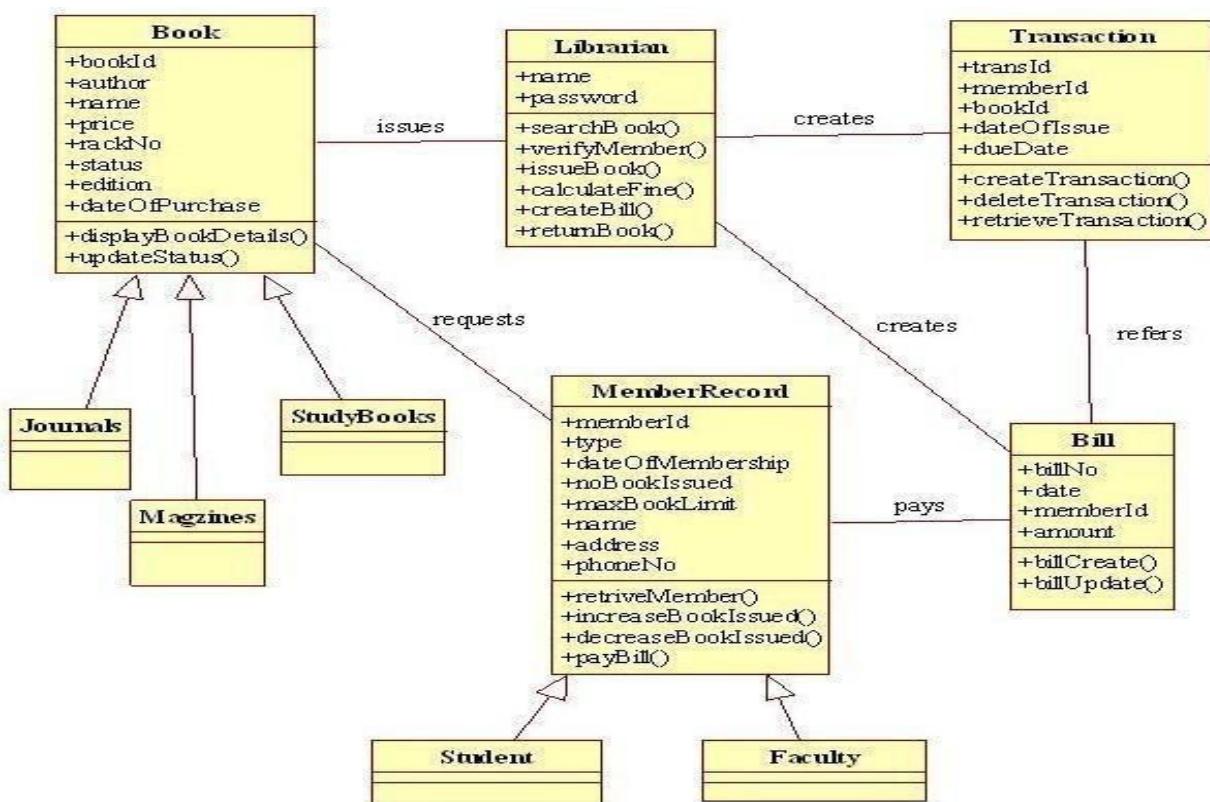
In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports.

All these modules are able to help librarian to manage the library with more efficient way as compared to library systems which are not computerized.

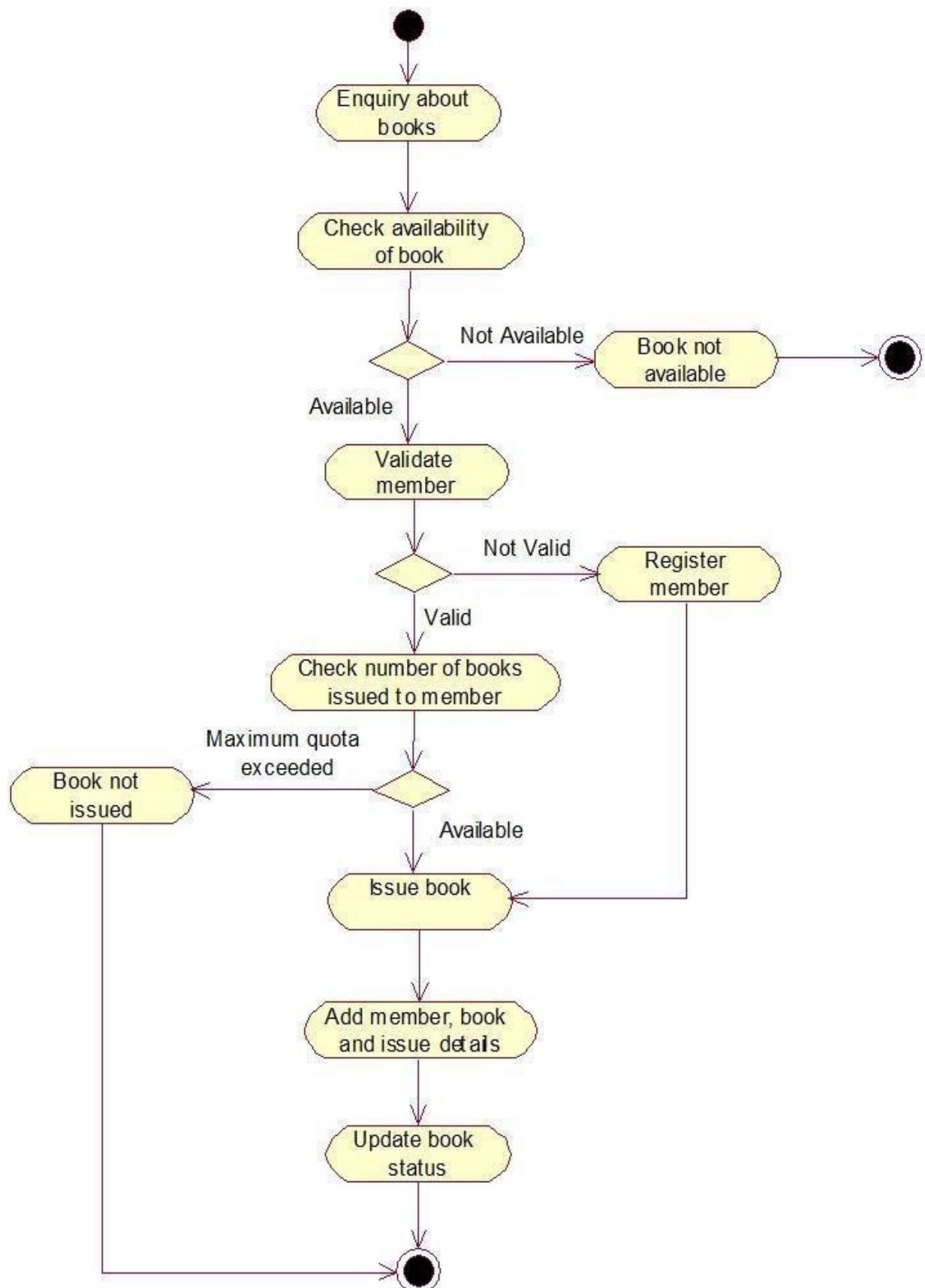
USECASE DIAGRAM



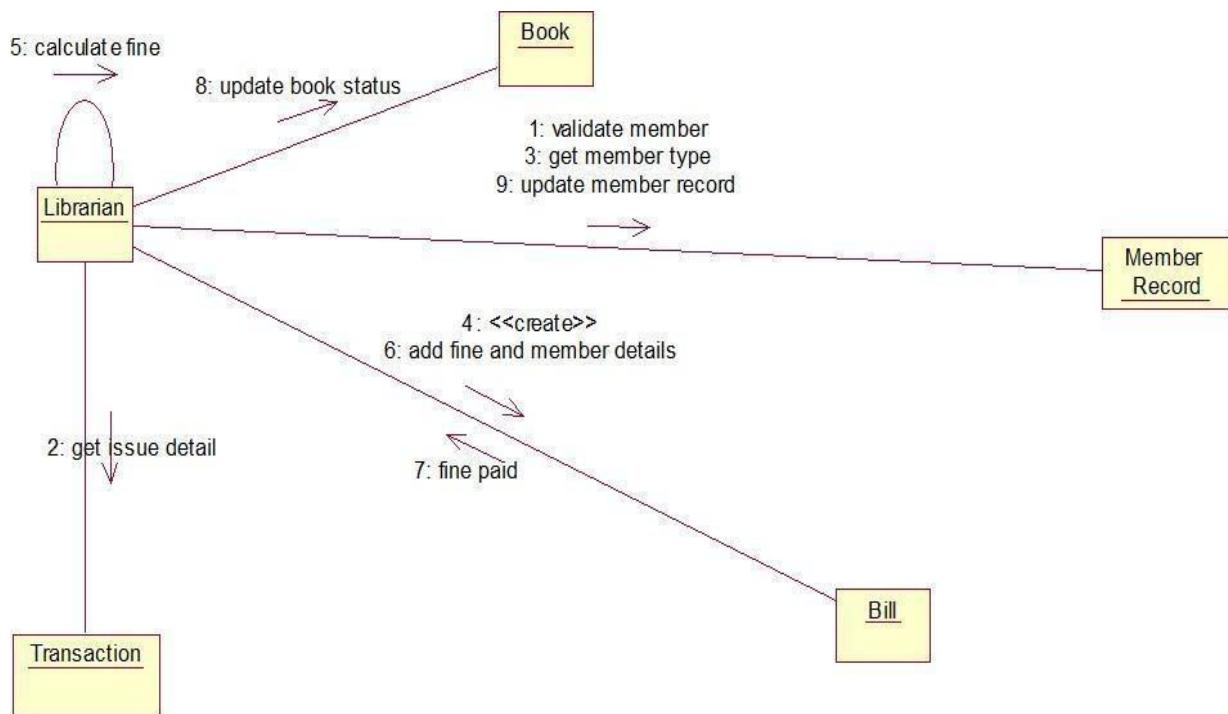
CLASS DIAGRAM



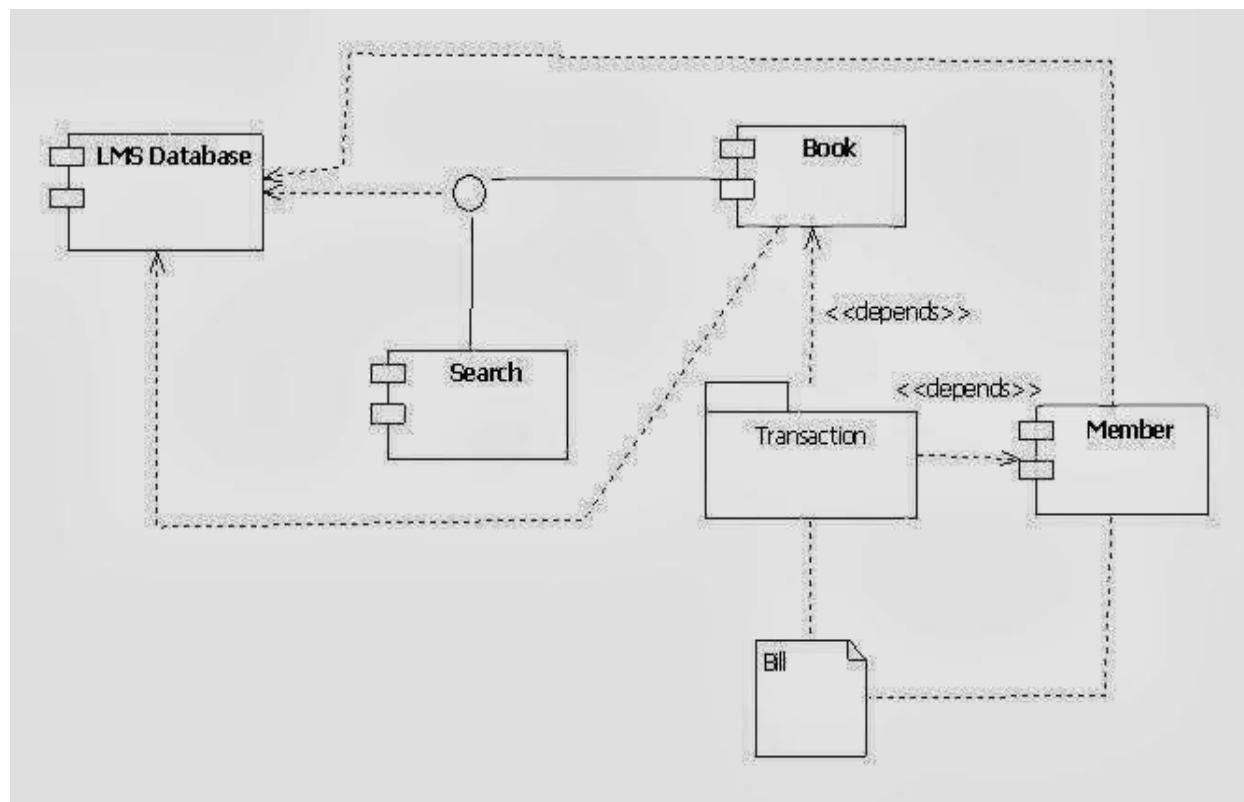
ACTIVITY DIAGRAM



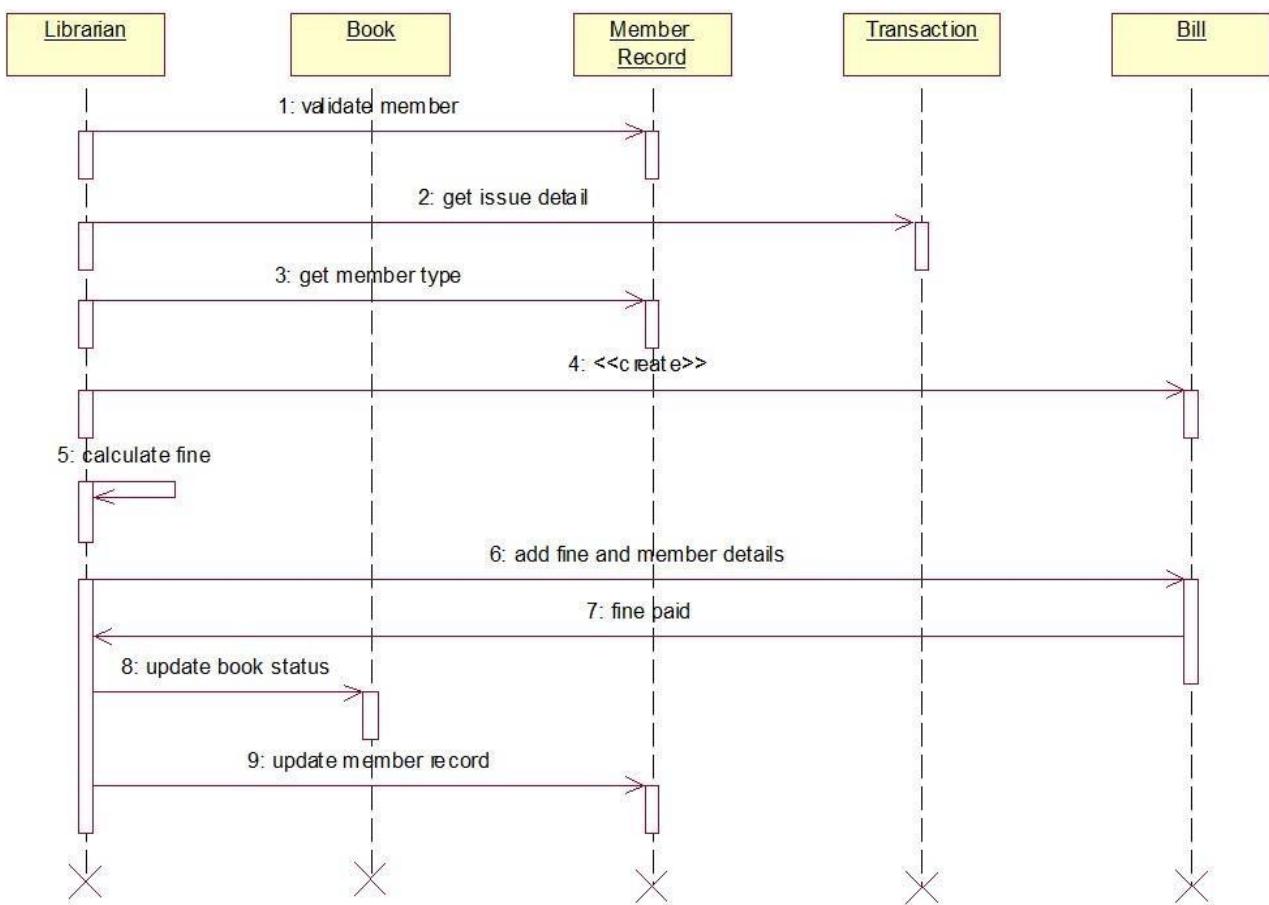
COMMUNICATION DIAGRAM



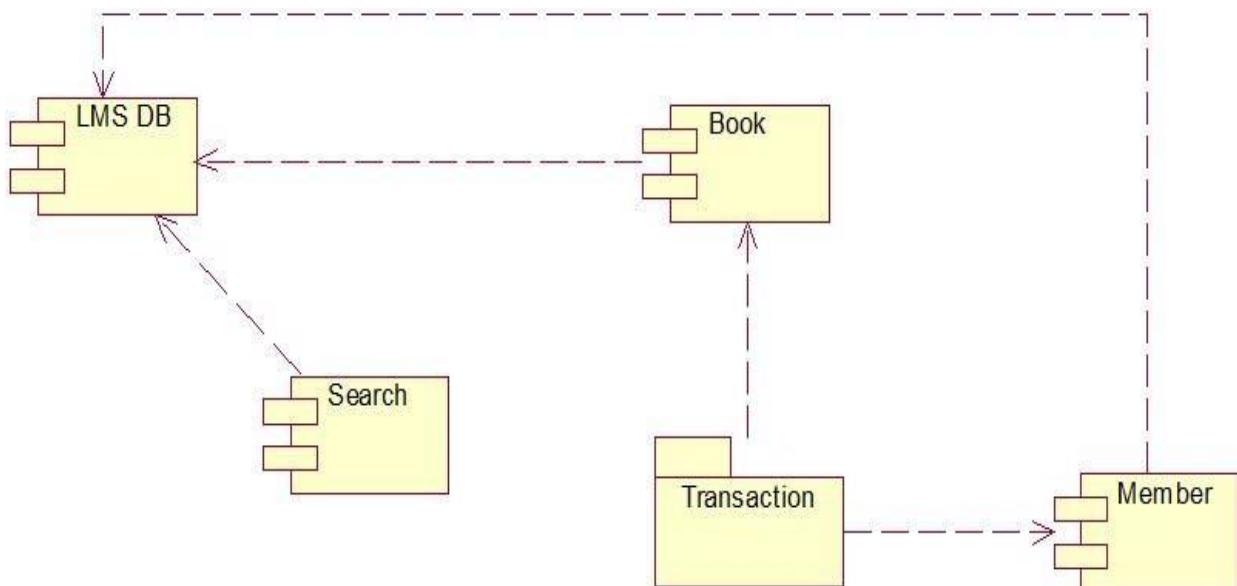
COMPONENT DIAGRAM



SEQUENCE DIAGRAM



COMPONENT DIAGRAM



RESULT:

Thus the “Library Management System has been analysed & designed and the coding skeleton has been generated using Umbrello software.

Expt No: 15
Date:

STUDENT MANAGEMENT SYSTEM

AIM:
To

analyze and Design UML Diagrams for Student Management system using Umbrello Software.

PROJECT SCOPE:

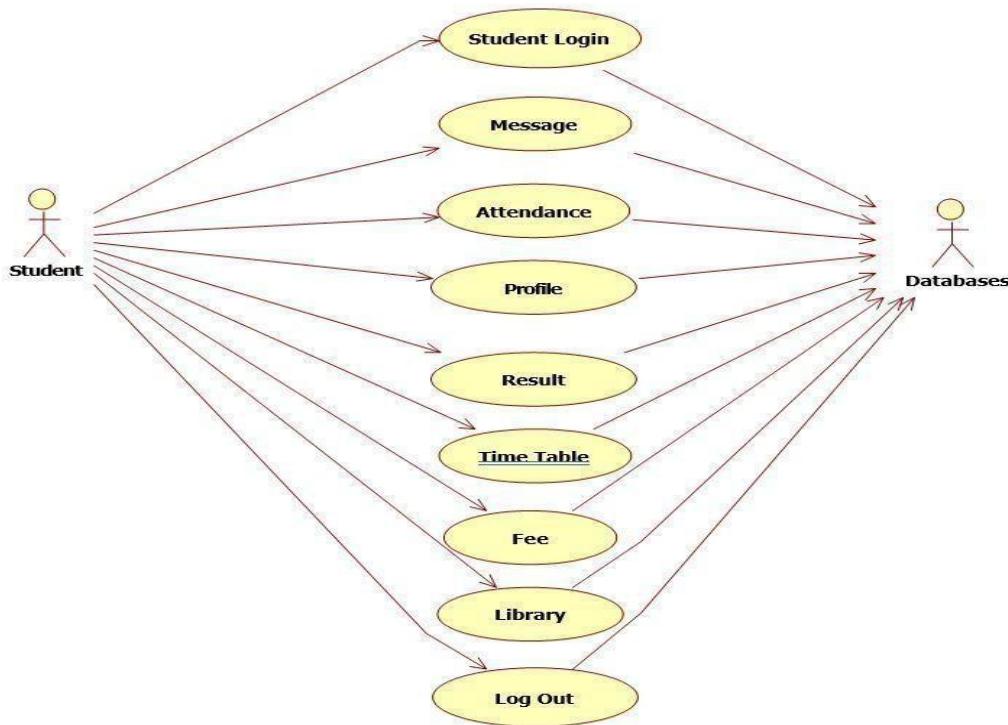
The student must register by entering the name and password to login the form. The admin select the particular student to view the details about that student and maintaining the student details. This process of student information system is described sequentially through following steps. The student registers the system. The admin login to the student information system. He/she search for the list of students. Then select the particular student. Then view the details of that student. After displaying the student details then logout.

OBJECTIVE:

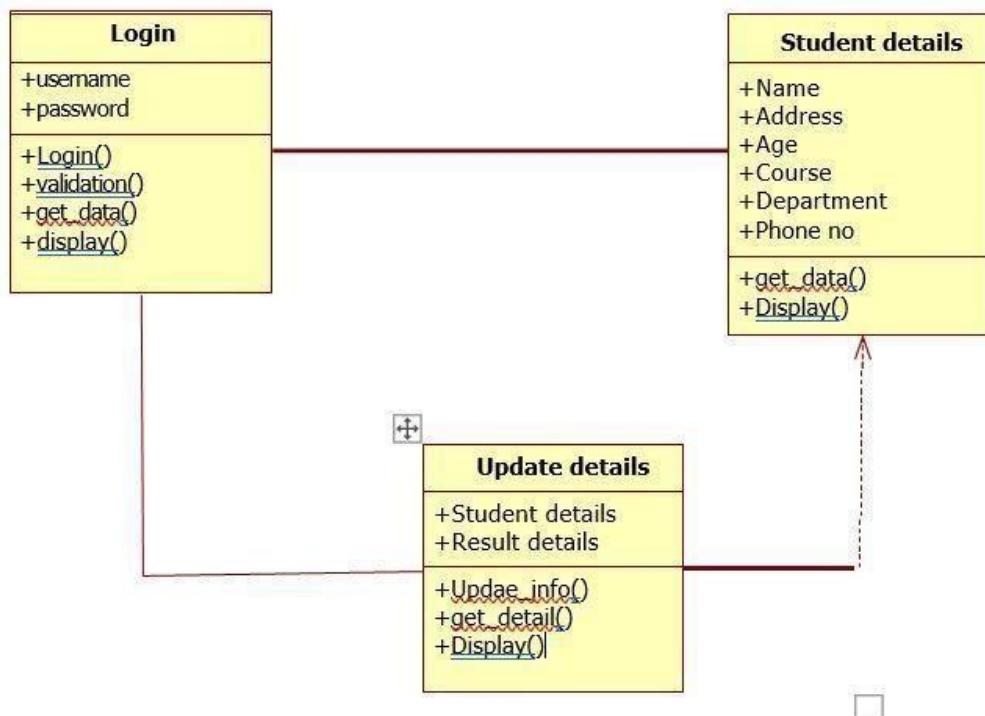
The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- A separate column for digital library
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- A teacher login page where teacher can add any events being organized in the college and important suggestions regarding books.
- Online notice board about the workshop

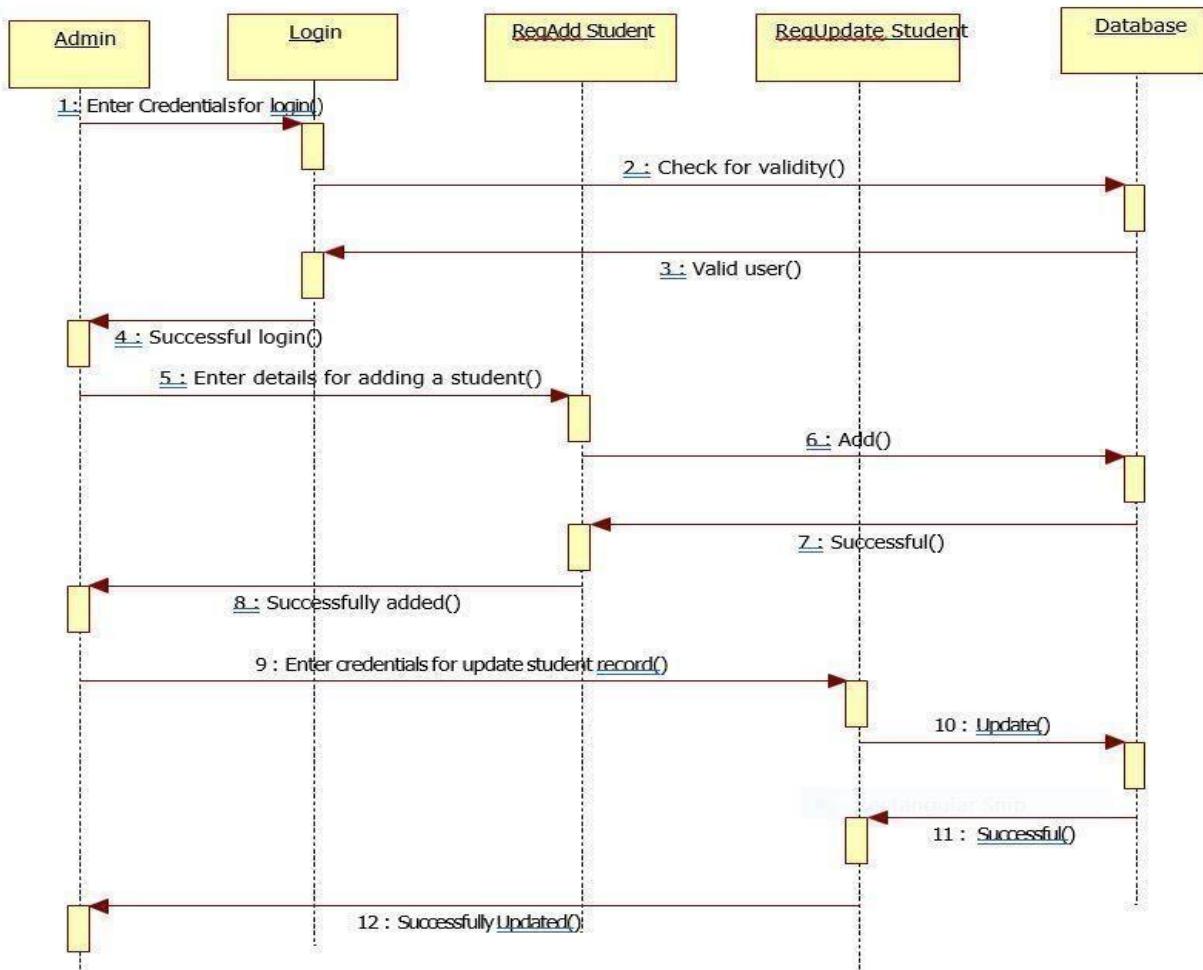
USECASE DIAGRAM



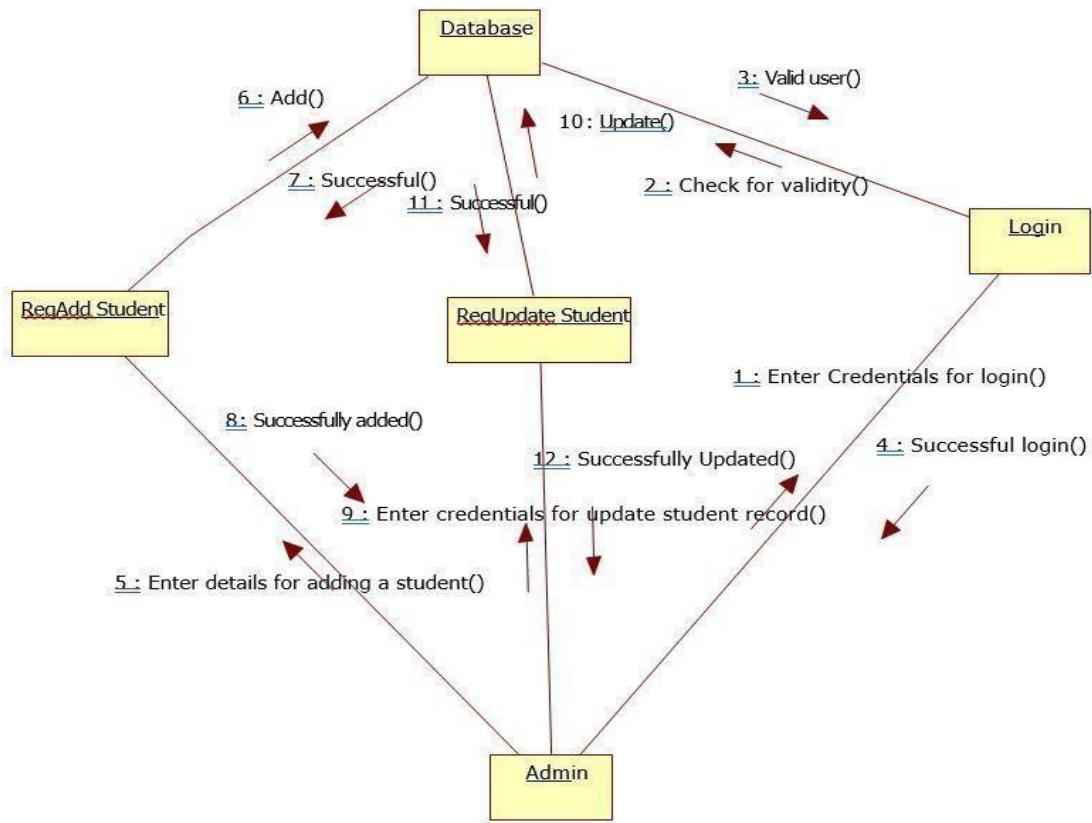
CLASS DIAGRAM



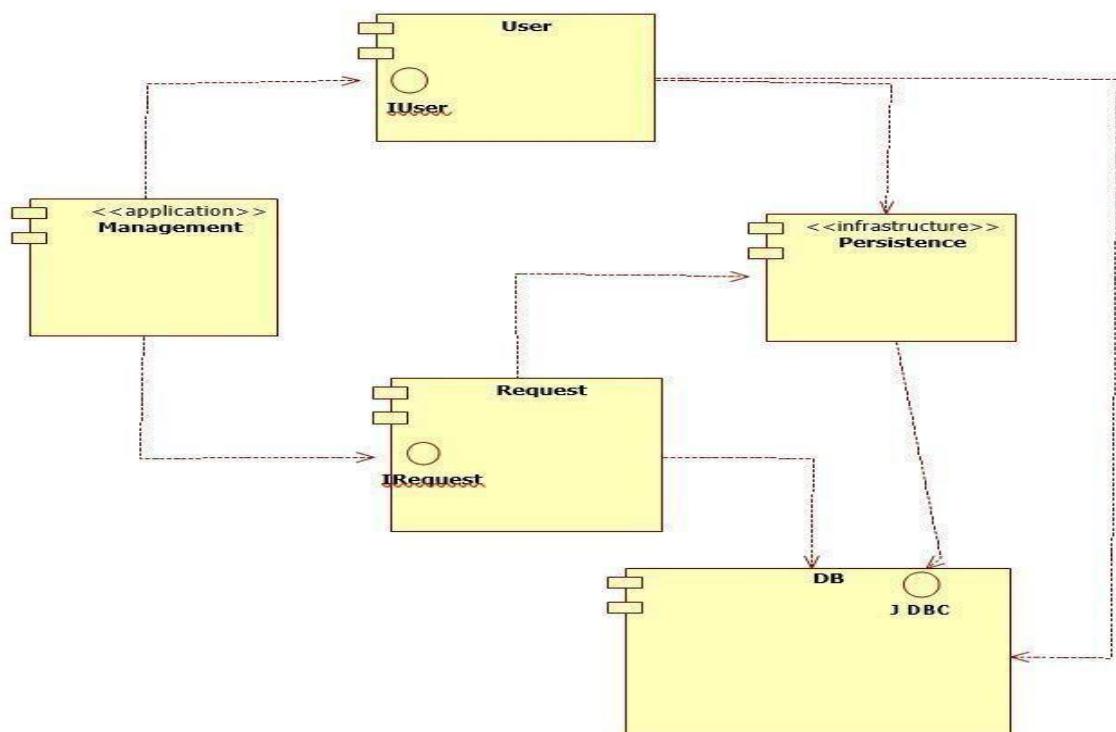
SEQUENCE DIAGRAM



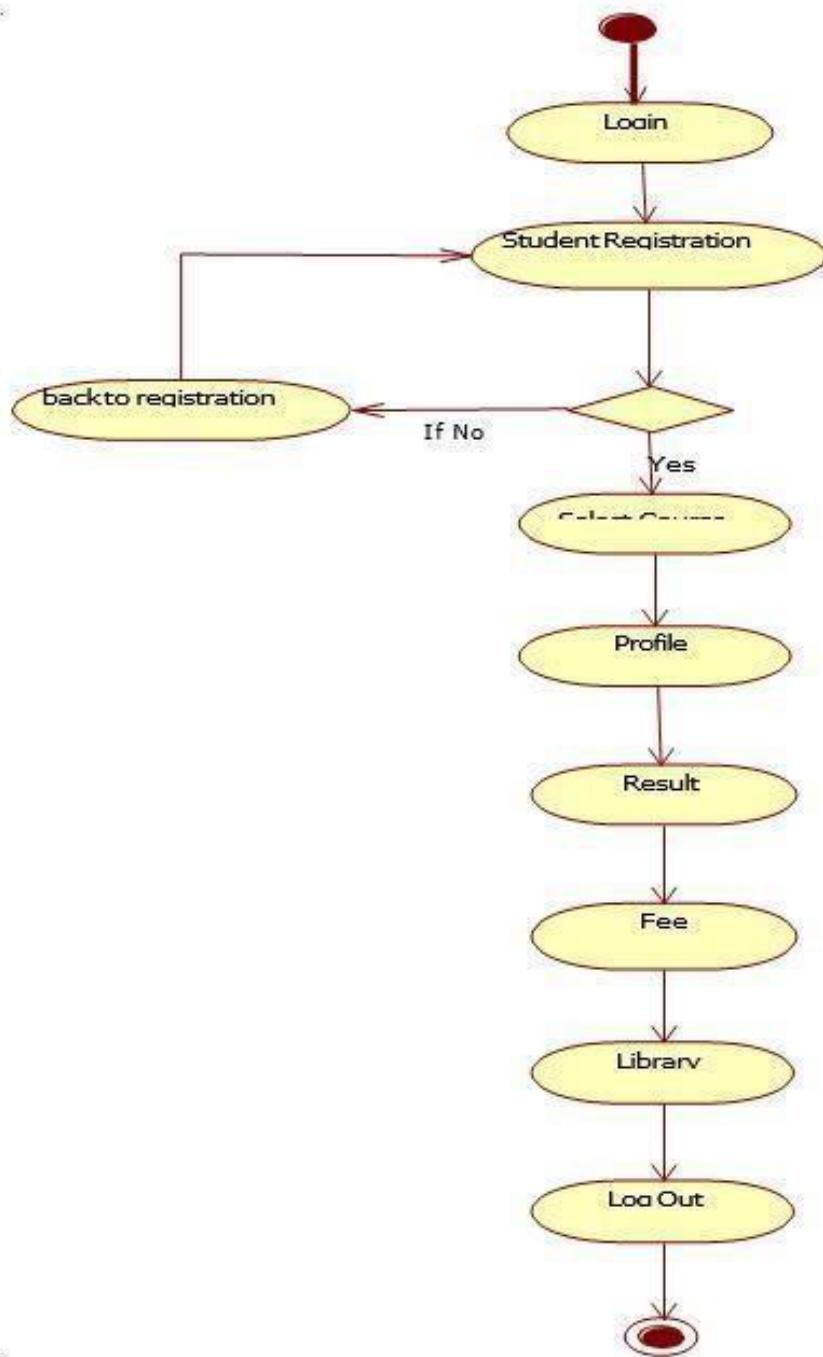
COLLABRATION DIAGRAM



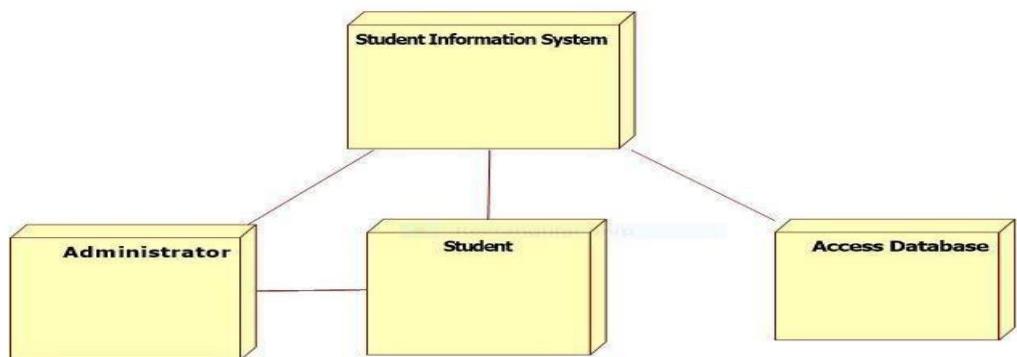
COMPONENT DIAGRAM



ACTIVITY DIAGRAM



DEPLOYMENT DIAGRAM



RESULT

Thus the “Student Management System has been analysed & designed and the coding skeleton has been generated using Umbrello software.