EX.NO.:1	PASSPORT AUTOMATION SYSTEM

AIM:

To design a Passport Automation System by using Argo-UML tool.

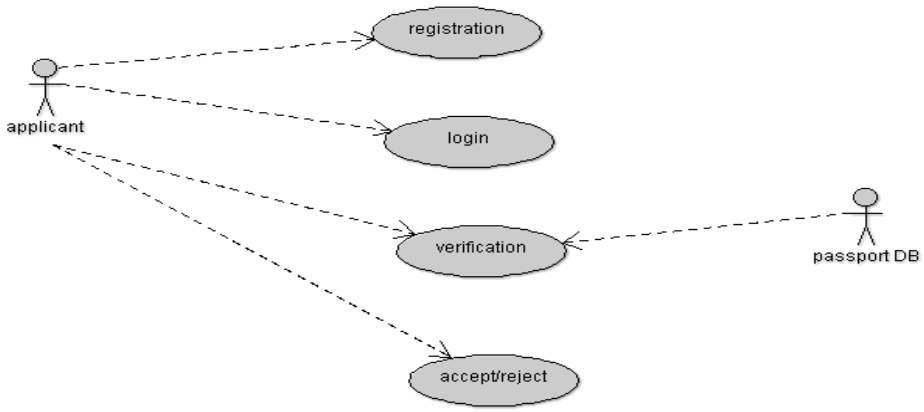
PROBLEM ANALYSIS AND PROJECT PLAN:

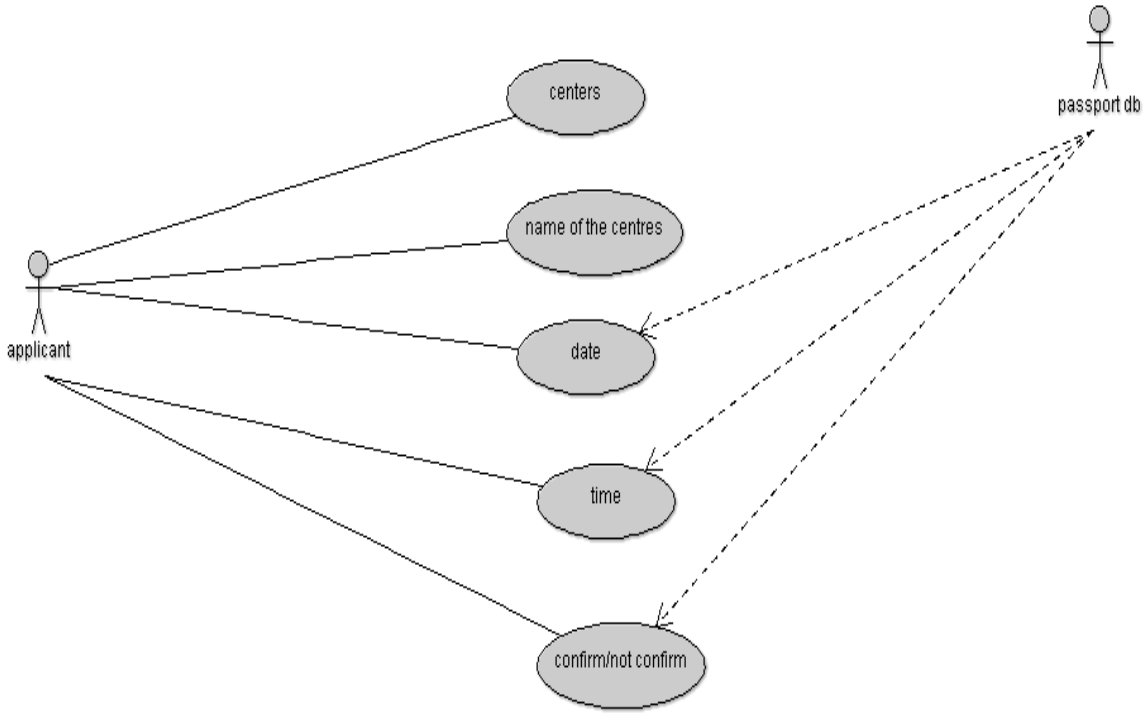
To simplify the process of applying passport, software has been created by designing through ARGO-UML tool. Initially the applicant login the passport automation system and submits his details. These details are stored in the database and verification process done by the passport administrator, regional administrator and police the passport is issued to the applicant.

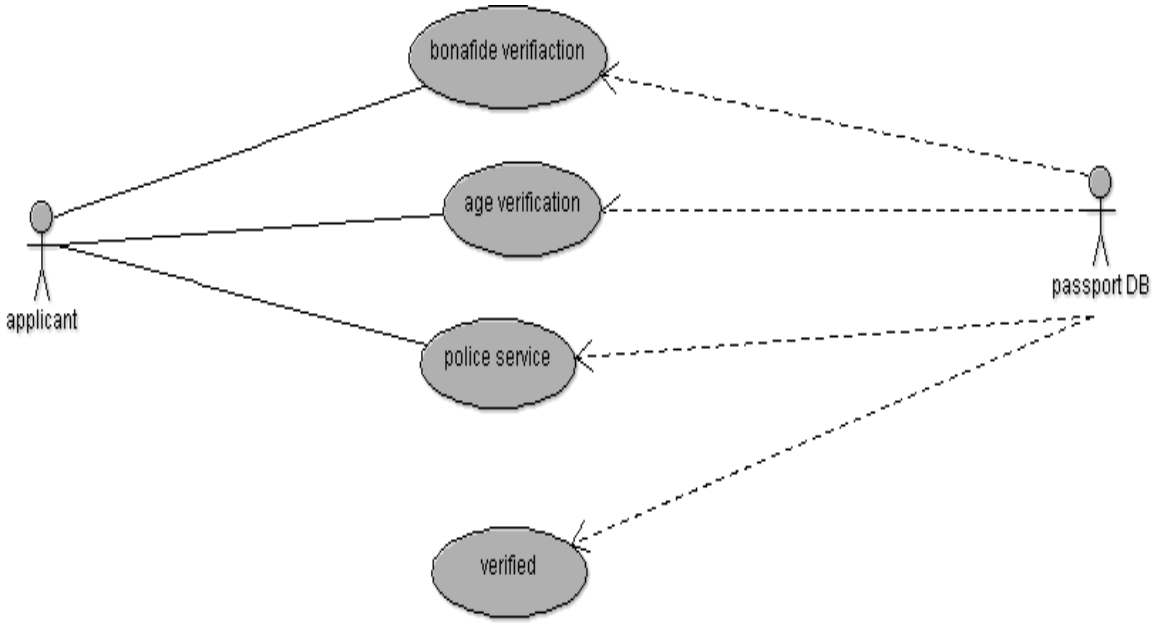
PROBLEM STATEMENT:

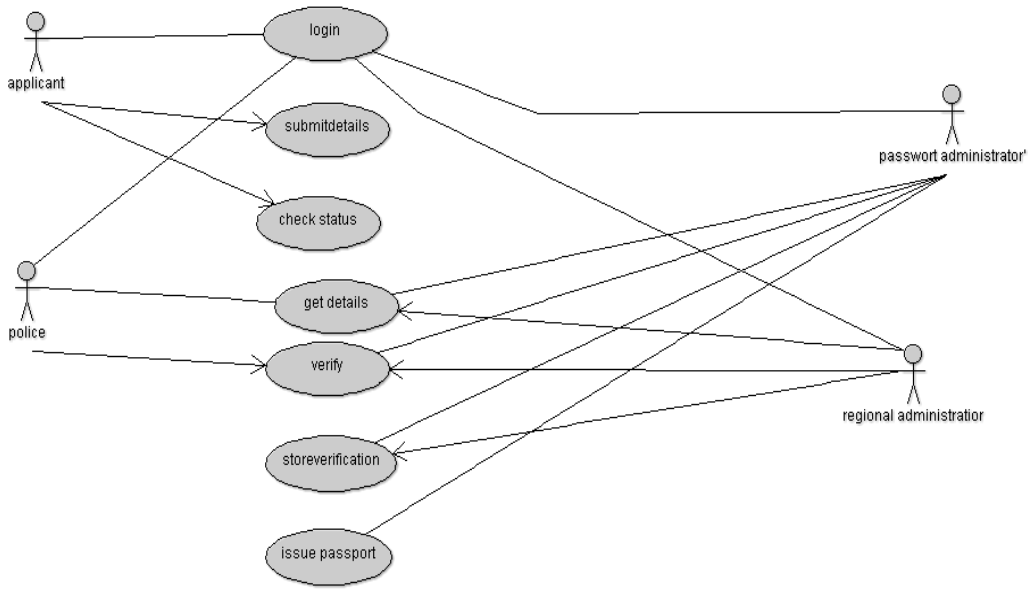
1. Passport Automation System is used in the effective dispatch of passport to all of the applicants. This system adopts a comprehensive approach to minimize the manual work and schedule resources, time in a cogent manner.
2. The core of the system is to get the online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database.
3. This forms the first and foremost step in the processing of passport application. After the first round of verification done by the system, the information is in turn forwarded to the regional administrator's (Ministry of External Affairs) office.
4. The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law.
5. The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator. After all the necessary criteria have been met, the original information is added to the database and the passport is sent to the applicant.

UML USECASE DIAGRAM:



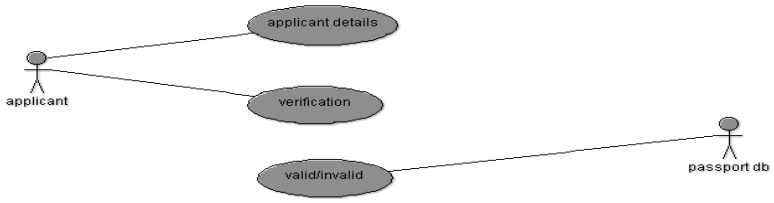

centers
passport db
name of the centres
applicant
date
time
confirm/not confirm


bonafide verifiaction
age verification
passport DB
applicant
police service
verified

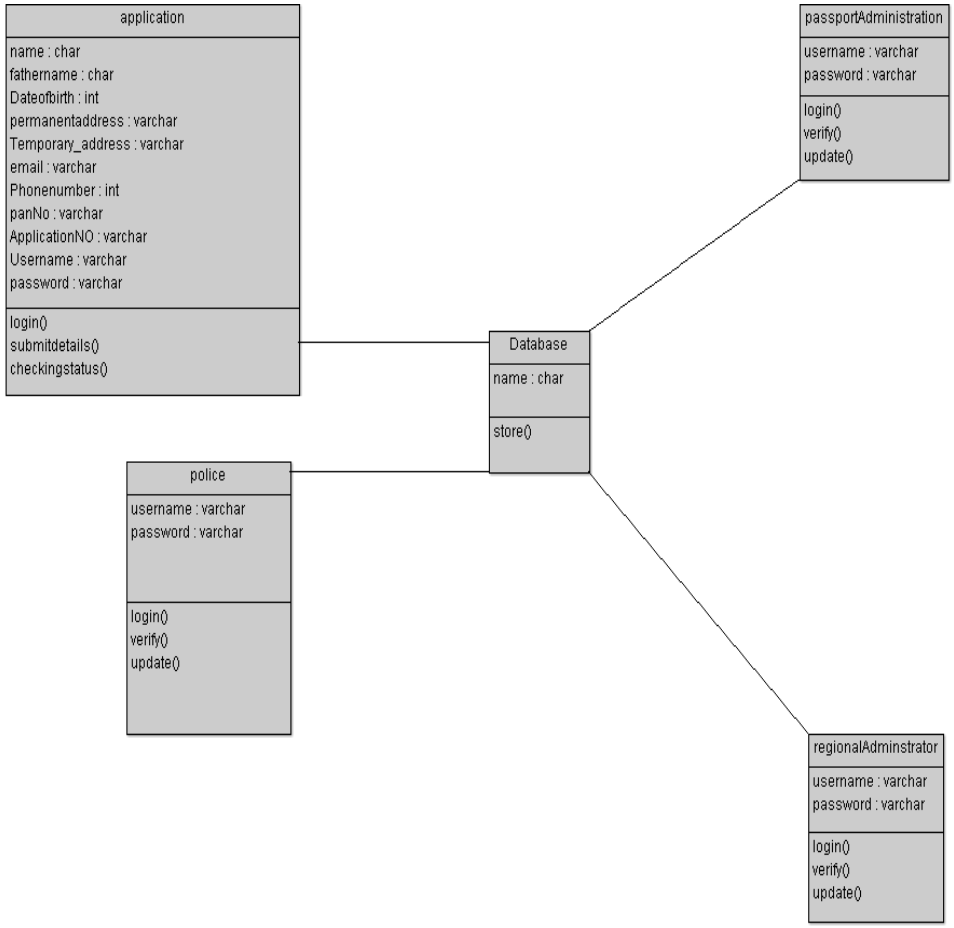


Identified Use cases are:

- APPLICATION
- PASSPORT ADMINISTRATION
- REGIONAL ADMINISTRATION
- POLICE
- DATABASE

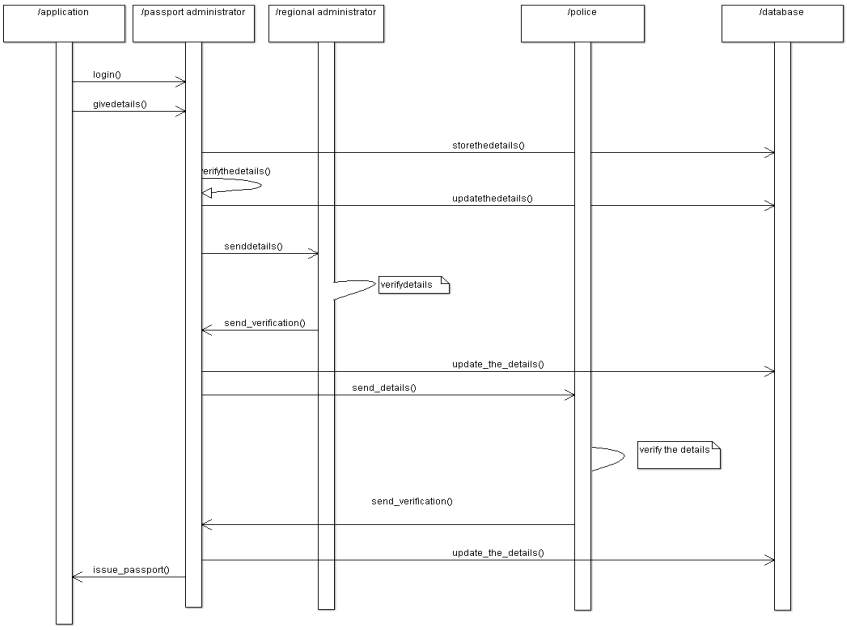


UML CLASS DIAGRAM:

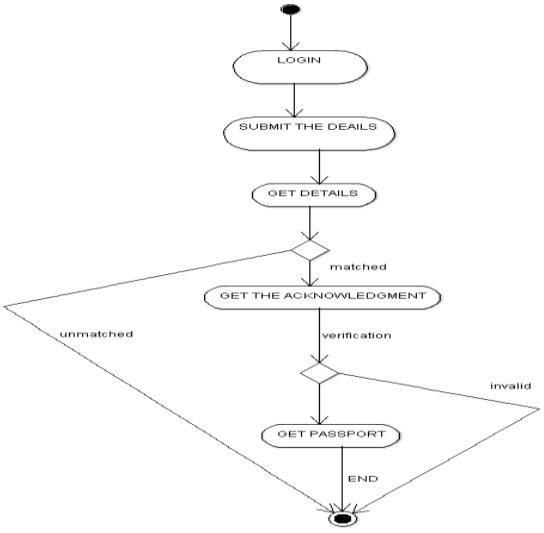


- APPLICANT
- THE DATABASE
- REGIONAL ADMINISTRATOR
- PASSPORT ADMINISTRATOR
- THE POLICE

UML INTERACTION DIAGRAM:



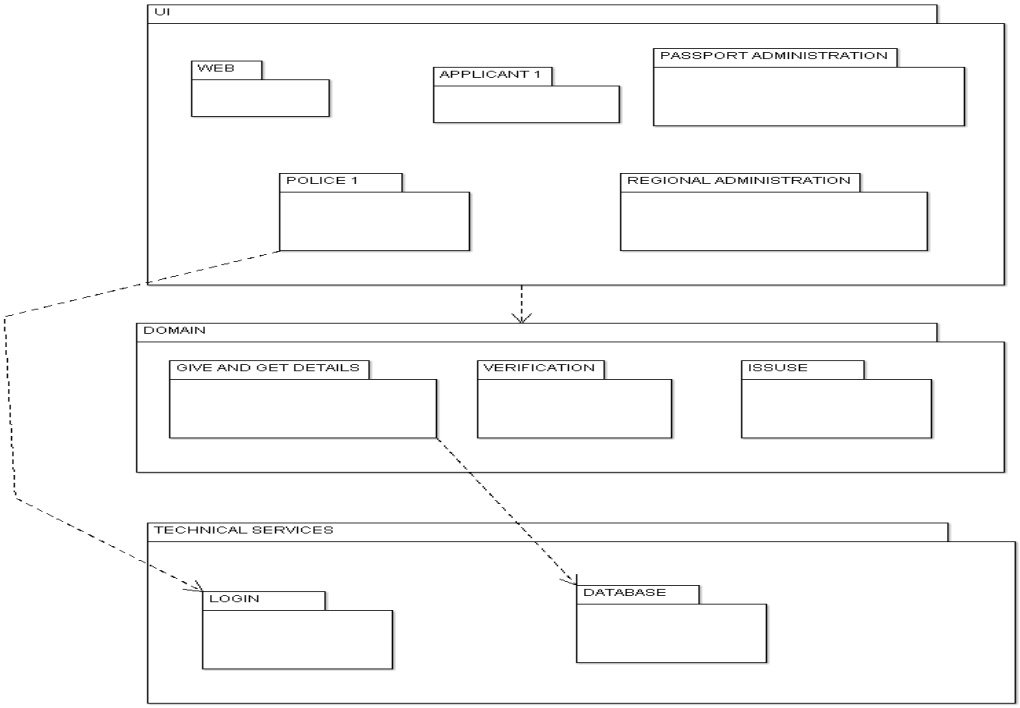
UML ACTIVITY DIAGRAM:



- The activities in the passport automation system are login, submit details, get details, issue passport and penalty and verification.
- In the login activity applicant give username and password and then login into the passport automation system after then fill the details that are required for application.

UML PACKAGE DIAGRAM:

- User-interface
  Web, applicant, police, regional administration.
- Domain
  Give and get details, Verification, Issues.
- Technical Services
  Login, Database.



UML TECHNICAL SERVICE LAYER:

ID	NAME	AGE	ADDRESS	PHONE NUMBER
1	Rajesh	19	Vellore	2343423
2	Tejesh	25	Chennai	45645645
3	Suren	42	Madurai	24254466

ID	NAME
1	Rajesh
2	Tejesh
3	Suren

ID	Appointment	Applicant ID	Date	Time
1	123	345	12-09-2016	4
2	124	234	13-09-2016	5
3	145	445	15-09-2016	7

UML DOMAIN OBJECT LAYER:

APPLICATION

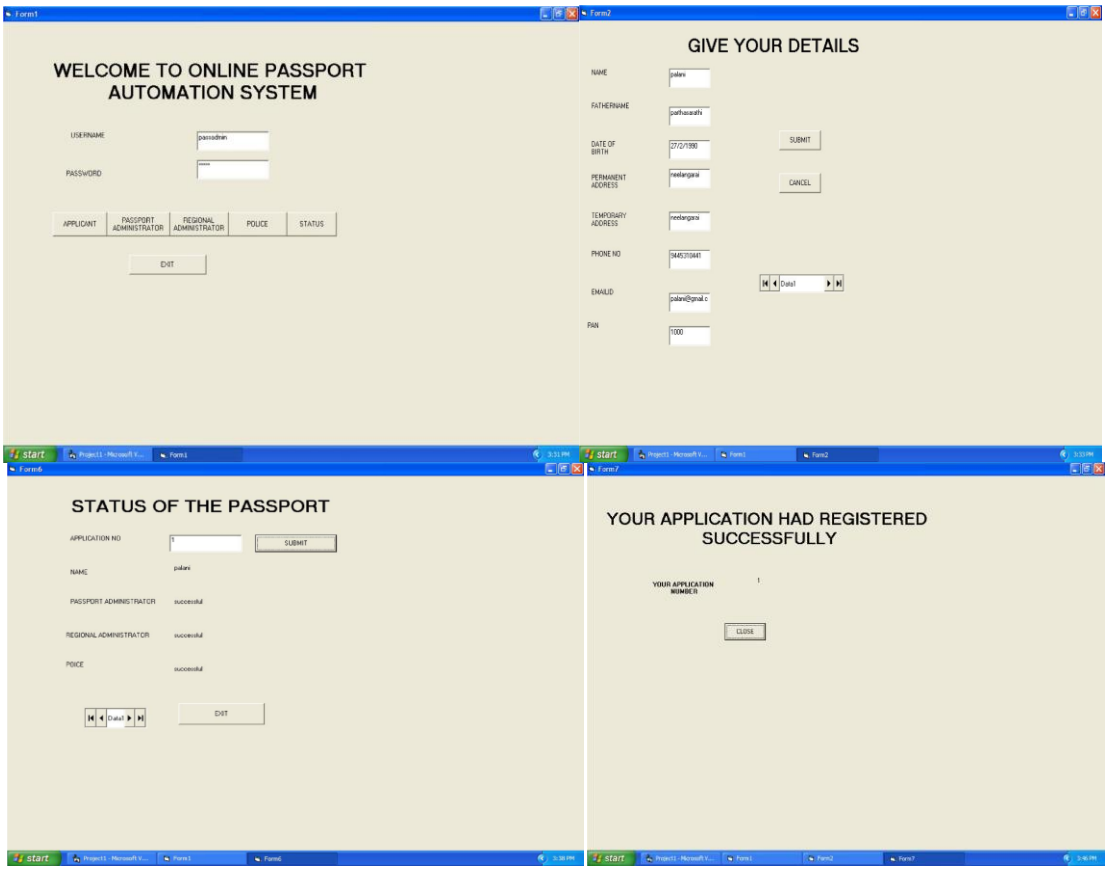
```
 import java.util.Vector;
  public class application {
  public char name;
  private char fathername;
  public int Dateofbirth;
  private varchar permanentaddress;
  private varchar Temporary_address;
  public varchar email;
  public int Phonenumber;
  public varchar panNo;
  public varchar ApplicationNO;
  public varchar Username;
  public varchar password;
   public Vector  myDatabase;

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

DATABASE

```
import java.util.Vector;
public class Database
{
  public char name;
    public Vector  myapplication;
    public Vector  mypassportAdministration;
```

```
    public Vector  myregionalAdminstrator;
    public Vector  mypolice;

  public void store()
 {
  }
}
```

USER INTERFACE LAYER:



RESULT:

Thus the Passport Automation System has been done successfully by Argo- UML tool.

EX.NO.:2	**BOOK BANK**

AIM:

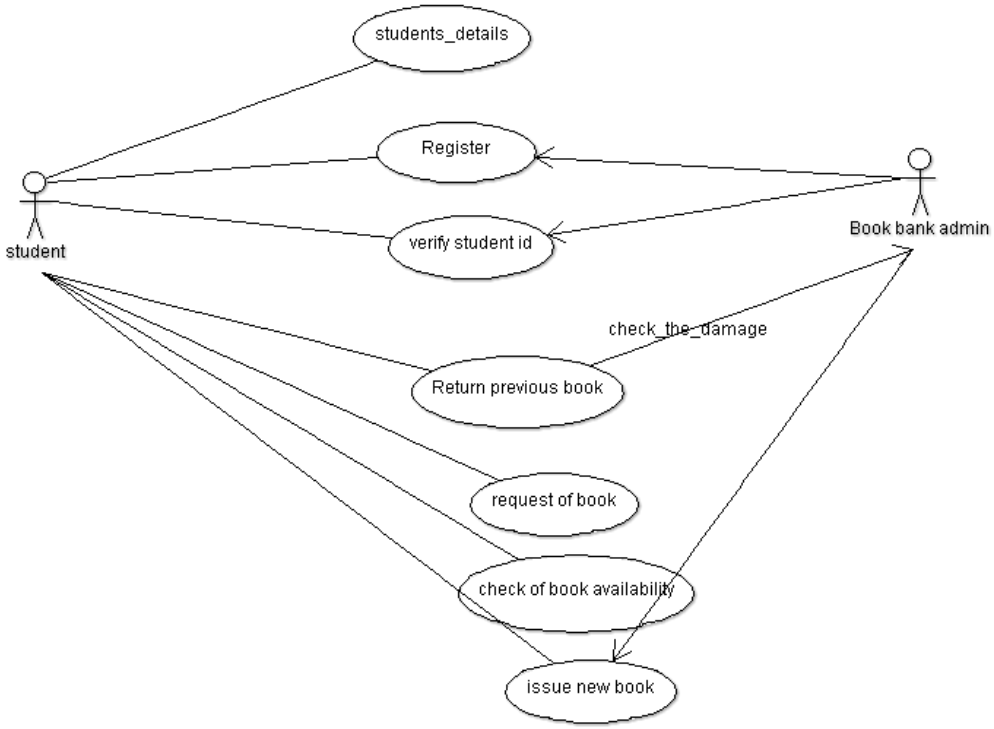
To design a Book Bank System by using Argo-UML tool. .

PROBLEM STATEMENT:

To simplify the process of applying Book Bank, software has been created by designing through ARGO-UML tool. The process of members registering and purchasing books from the book bank are described sequentially through following steps:

a. First the member registers himself if he was new to the book bank.
b. Old members will directly select old member button..
c. They select their corresponding year.
d. After selecting the year they fill the necessary details and select the book and he will be directed towards administrator
e. The administrator will verify the status and issue the book.

UML USECASE DIAGRAM:



Actor: - students, Book bank admin, user

Use case: - Students details, Date of issue, Date of return, no of books taken, Check availability.

UML CLASS DIAGRAM:

STUDENTS

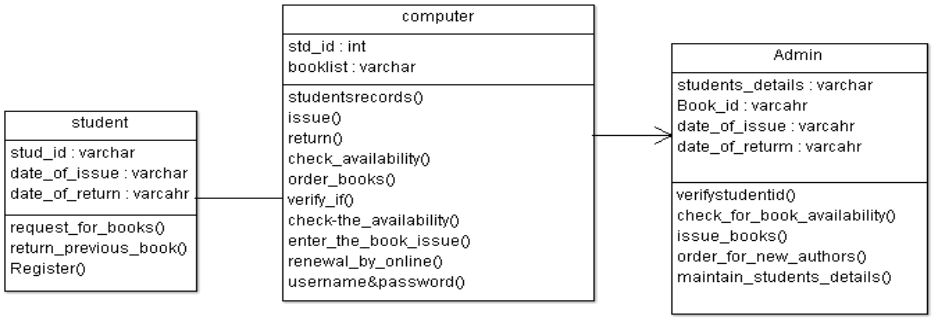
Registration, Request for book, Return previous Books

COMPUTER

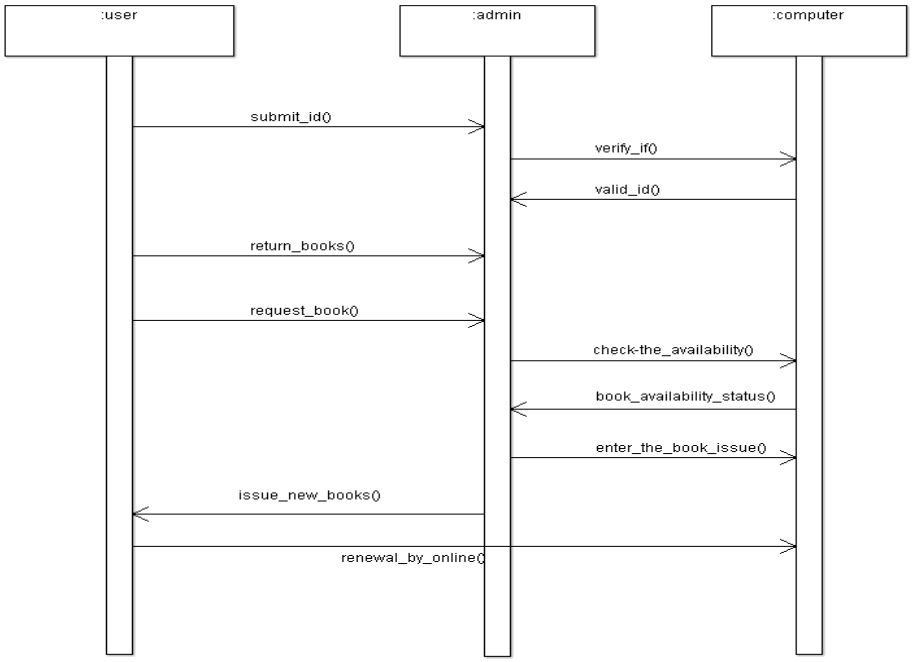
Student's record, issue, return, check availability, order books, verify id, renewal by online, username and password.

ADMIN

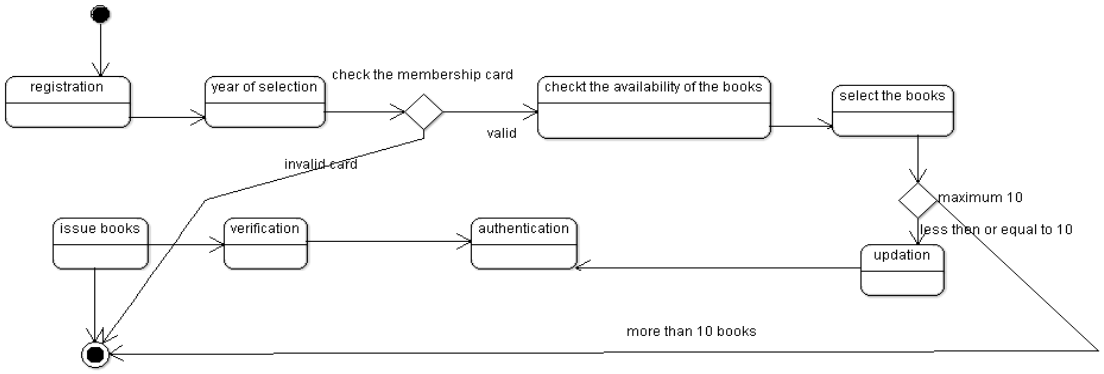
Verify students, check for book availability, Issue books, Order for new author, maintain students details



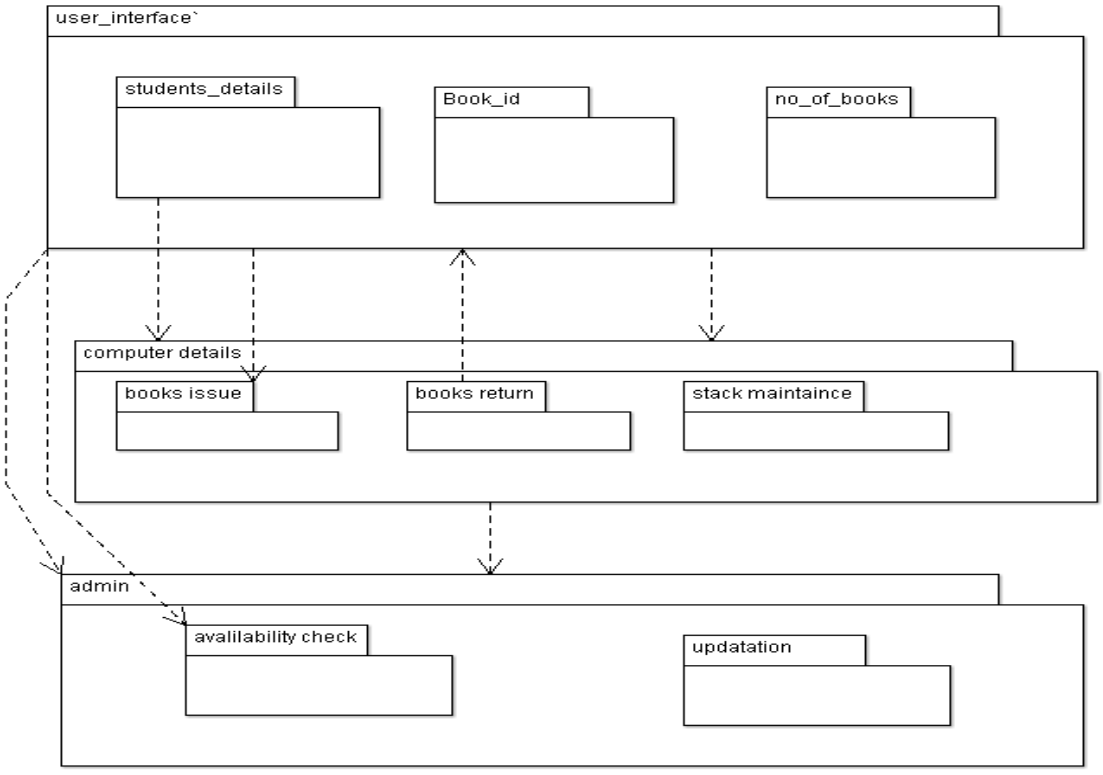
UML INTERACTION DIAGRAM:



UML STATECHART DIAGRAM:



UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.No	Std-Id	Book No	Date-of-issue
1.	15680	1234	01-06-2017
2.	15692	4102	15-07-2017
3.	15682	2011	20-06-2017

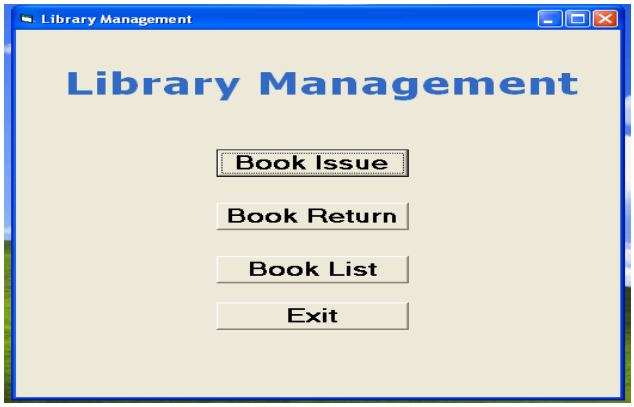
S.No	Book-No	Std-Id	Date of return
1.	1234	15680	15-06-2017
2.	4102	15692	30-07-2017
3.	2011	15682	05-07-2017

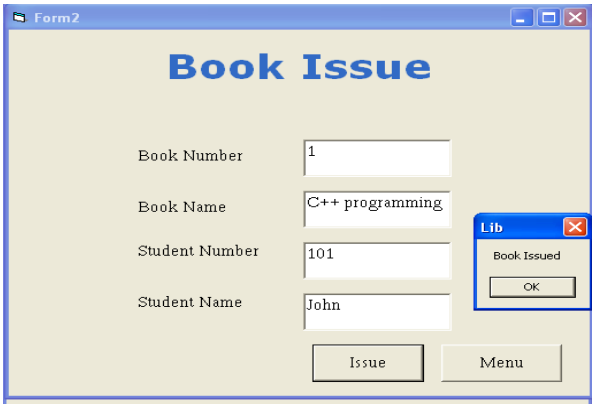
UML DOMAIN OBJECT LAYER:

```
import java.util.Vector;
  public class computer {
    public int std_id;
    public varchar booklist;
    public Vector  mystudent;
    public Vector  mystudent;
    public Vector  myAdmin;
    public Vector  myAdmin;
  private void studentsrecords() {
  }
  private void issue() {
  }
  public void return() {
  }
  public void check_availability() {
  }
  public void order_books() {
  }
  public void verify_if() {
  }
  public void check-the_availability() {
  }
  public void enter_the_book_issue() {
  }
 public void renewal_by_online() {
  }
  public void username&password() {
                }
}
import java.util.Vector;
```

```
public class student {

  public varchar stud_id;

  public varchar date_of_issue;

  public varcahr date_of_return;

     public Vector  mycomputer;

  public void request_for_books() {
  }

  public void return_previous_book() {
  }

  public void Register() {
  }

}
```

USER INTERFACE LAYER





Result:

Thus the Book Bank Management has been done successfully by using Argo-UML.

EX.NO.: 3	**EXAM REGISTRATION**

AIM:

To design Exam Registration System by using Argo-UML tool. .

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying Exam Registration, software has been created by designing through ARGO-UML tool.

The exam registration is an application in which applicant can register themselves for the exam. The details of the students who have registered for the examination will be stored in a database and will be maintained. The registered details can then be verified for any fraudulent or duplication and can be removed if found so. The database which is verified can be used to issue hall tickets and other necessary materials to the eligible students.

PROBLEM STATEMENT:

Exam registration system is used in the effective dispatch of registration from to all of the students this system adopts a comprehensive approach to minimize the manual work and schedule resources, time in cogent manner the core of the system is to get the online registration from (with details such as name , reg.no , etc.,) filled by the student whose statement is verified for it is genuineness by the exam registration system with respect to the already existing information in the database. This forms the first and fore most step in the processing of exam application.

The process of students accessing the registration application and applying for the examination by filling out the form with proper details and then the authorities verify those details given for truth and correctness are sequenced through steps

a. The students access exam registration application.

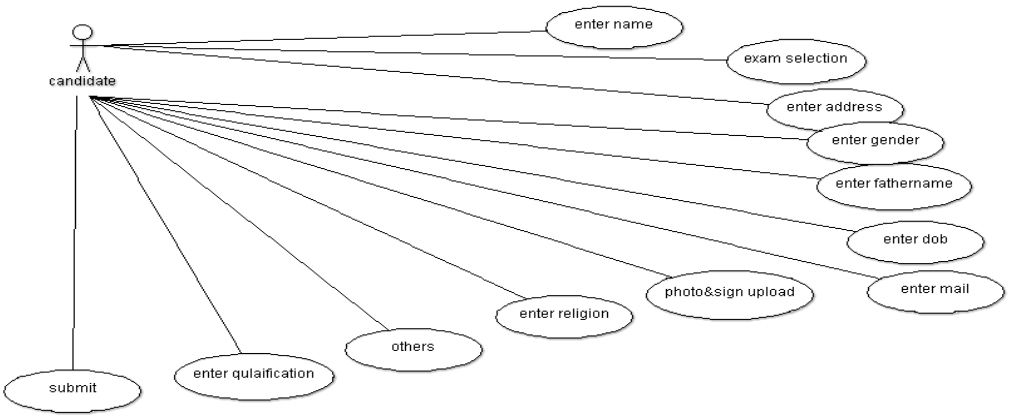
b. They fill out the form with correct and eligible details.

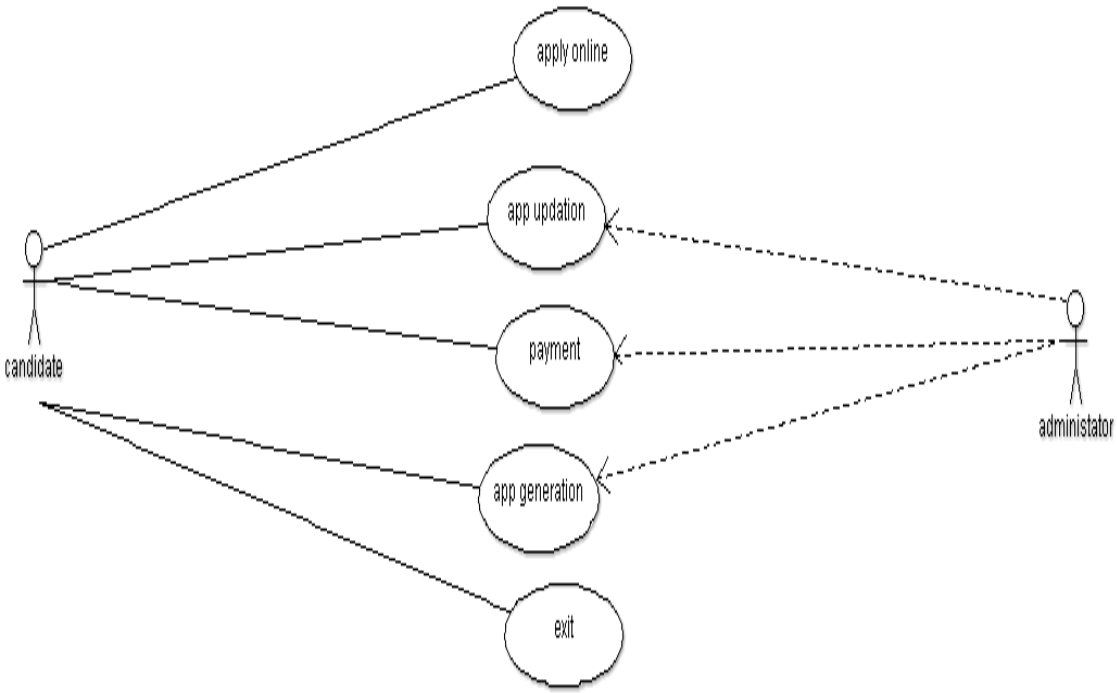
c. They complete the payment process.

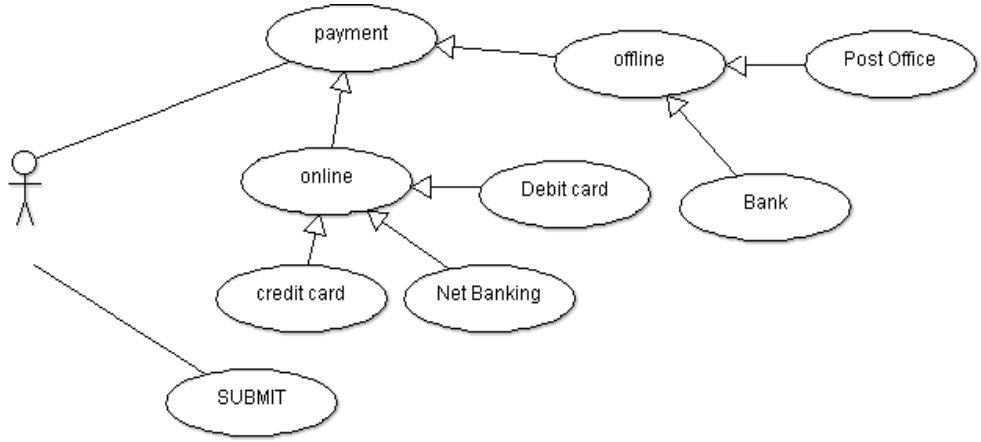
d. The authorities verify or check the details.

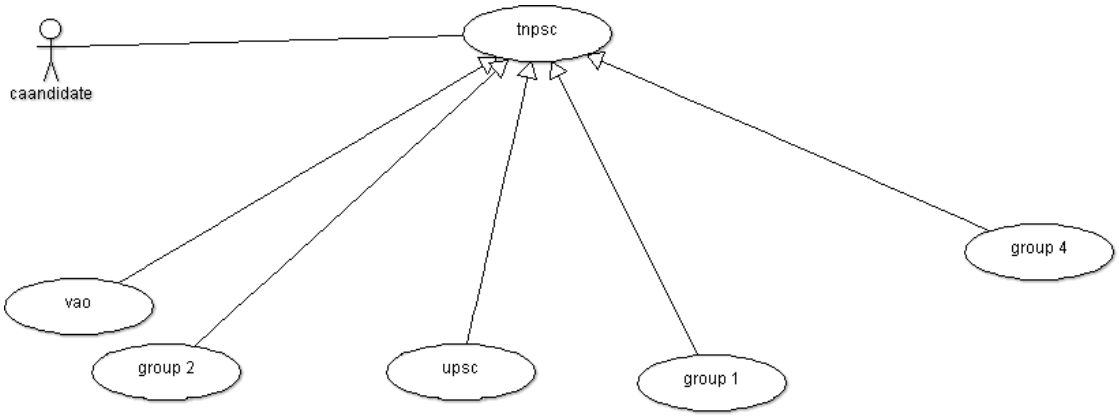
e. After all verification the exam registration database is finalized.

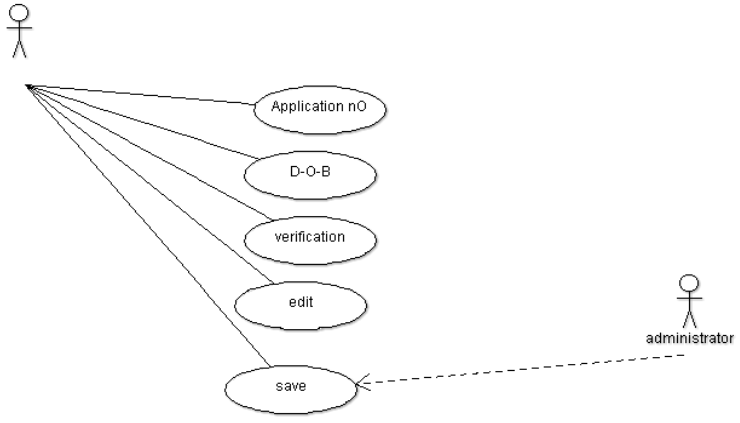
UML USECASE DIAGRAM:

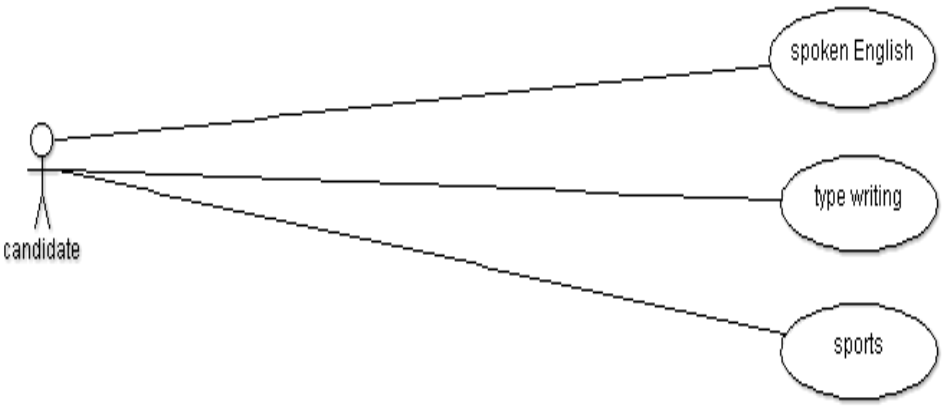


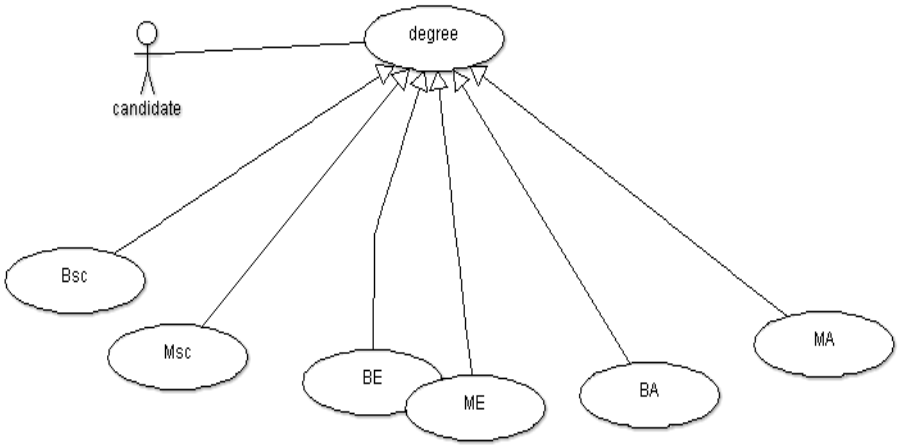



payment
offline
Post Office
online
Debit card
Bank
credit card
Net Banking
SUBMIT


caandidate
tnpsc
vao
group 2
upsc
group 1
group 4


Application nO
D-O-B
verification
edit
save
administrator





The actors in this use case diagram are Students, Interface and Database. The use cases are the activities performed by actors.

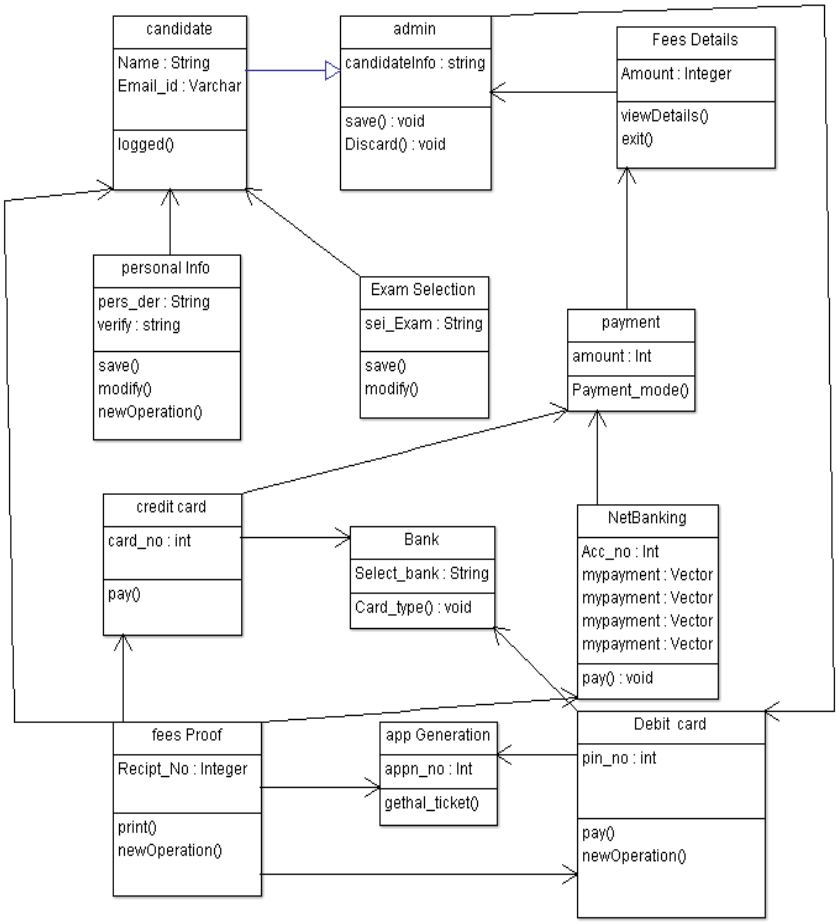
a. Student fills outs the form in the form filling process.

b. The interface checks and validates registered details.

c. Then the database is searched for details and verified.

d. Database stores the details and returns acknowledgement.

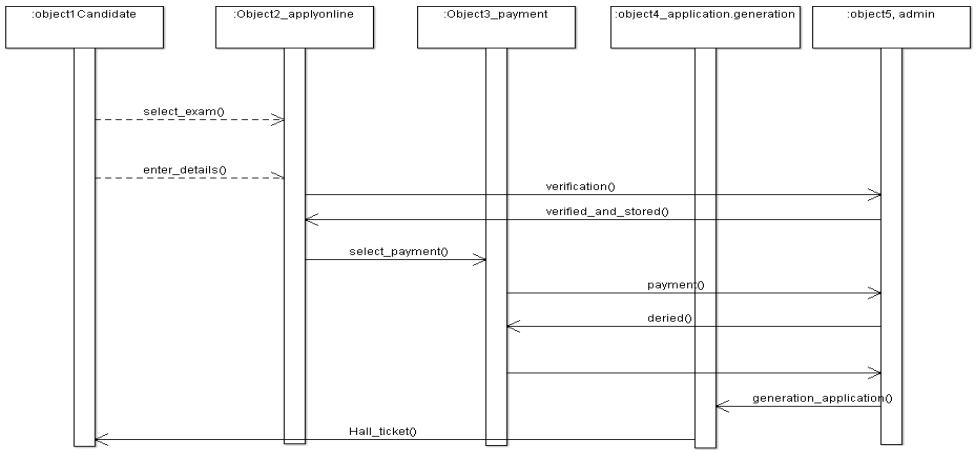
UML CLASS DIAGRAM:



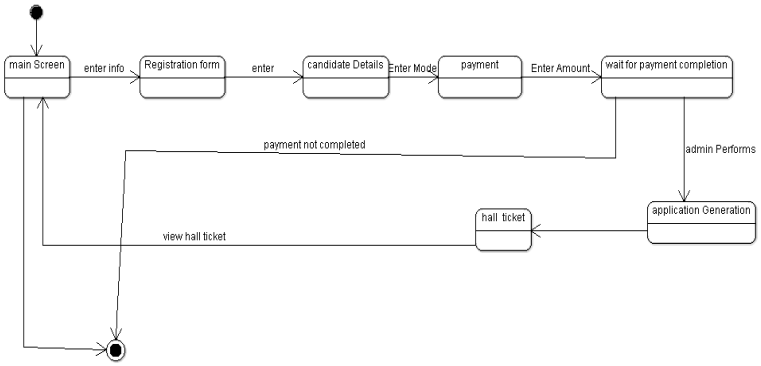
UML INTERACTION DIAGRAM:

The sequence diagram describes the sequence of steps to show:

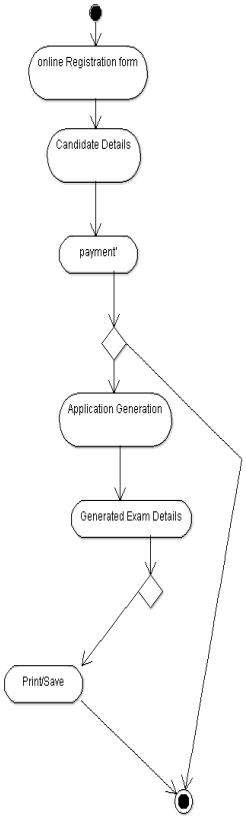
- The candidate selects the exam and enters their detail in the online registration form.
- Then the candidate selects the payment mode like Credit card, Debit card and Net banking.
- If the payment is not completed then it goes for repayment and after completion of the payment the application will be generated.
- The detail of exam is viewed by the candidate.



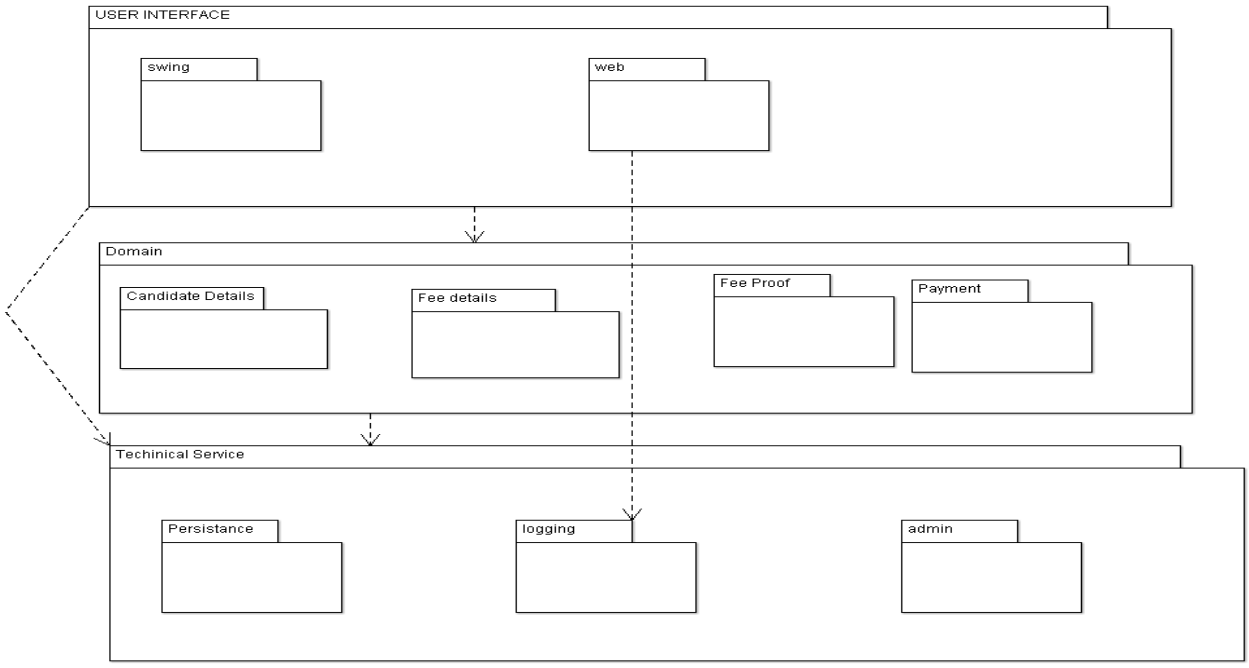
UML STATECHART & ACTIVITY DIAGRAM:



- The state chart diagram starts with the main screen the candidate will enter into the registration form he / she enters the necessary details.
- The candidate moves to the payment option he /she enters the amount and waits for the payment completion.
- The Admin performs the application generation and the hall ticket will be viewed by the candidate through the main screen.
- If the payment doesn't complete it moves to the end state.



UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

Name	Date of Birth	Qualification	Gender	Select Exam	Phone Number
Saravana	01-05-1997	B.E	M	GROUP 1	9087349490
Surya	06-07-1998	B.E	M	GROUP 2	8337849320
Vinoth	09-08-1998	B.E	M	GROUP 3	7334739390

Register No	Name	Application No.	Hall Ticket	Center	Date of Examiner
510615104078	Saravana	1014327	654789	VOC COLLEGE	06-10-2017
501615104088	Surya	1014329	698547	CAHCET	05-11-2107
510615104105	Vinoth	1014320	647895	VIT	05-12-2017

UML DOMAIN OBJECT LAYER:

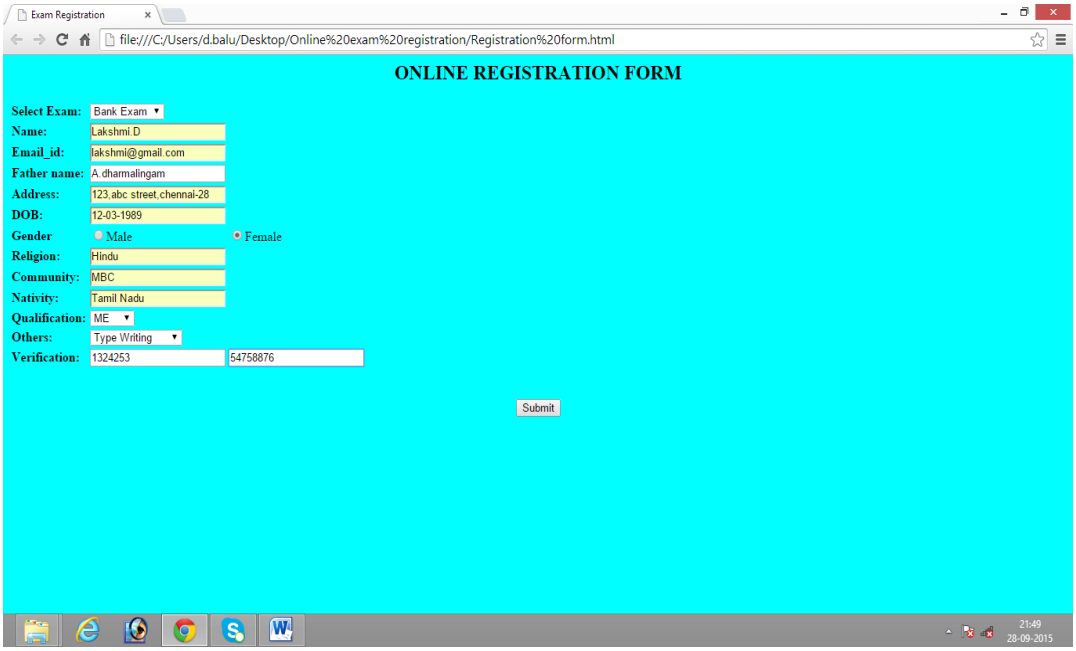
PERSONAL INFO:-

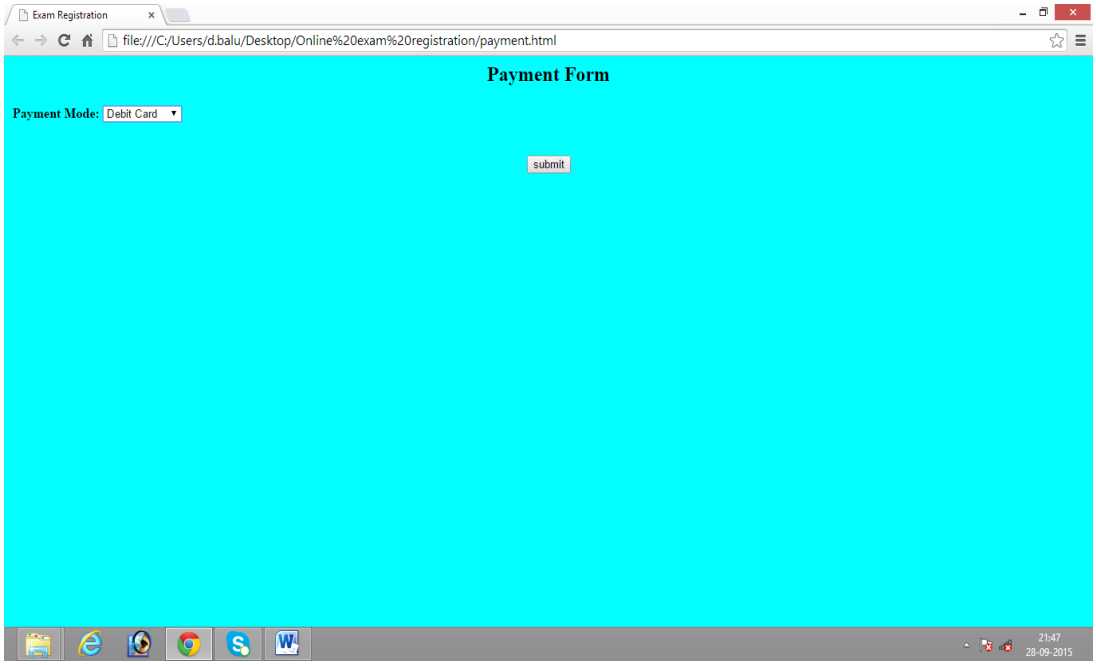
```
import java.util.Vector;

public class personal Info {
  public String persss_der;
  public string verify;
    public Vector  mycandidate;
  public void save() {
  }
  public void modify() {
 }
  public void newOperation() {
  }

}
```

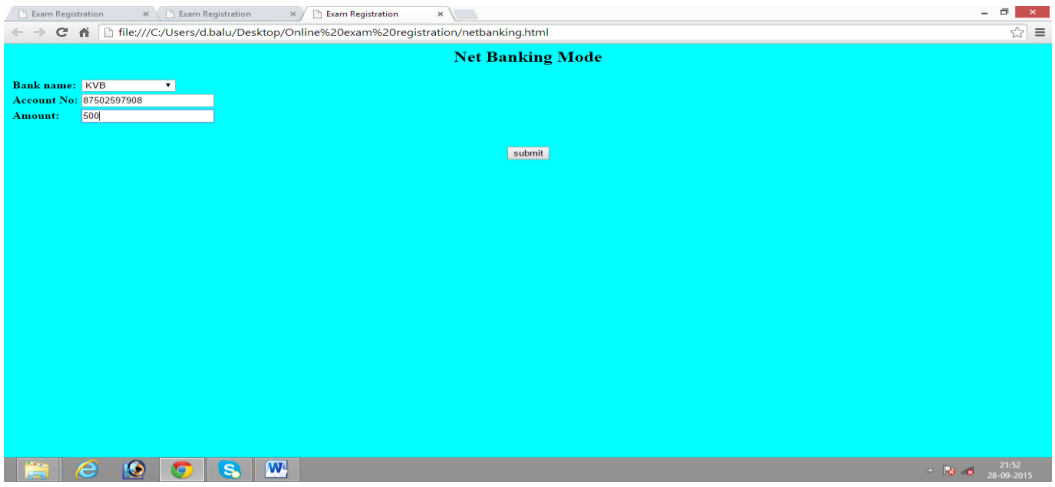
ADMIN:-

```
import java.util.Vector;

public class admin {
  public string candidateInfo;
    public Vector  myFees Details;
      public Vector  myDebit  card;
  public void save() {
  }
  public void Discard() {
  }
}
```

USER INTERFACE LAYER:







Result

Thus the Exam Registration has been done successfully by using Argo-UML.

EX.NO.: 4	STOCK MAINTENANCE SYSTEM

AIM:

To design Stock maintenance System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying stock maintenance system, software has been created by designing through ARGO-UML tool. This software is designed for supporting the computerized Stock Maintenance System. In this system, the customer can place order and purchase items with the aid of the stock dealer and central stock system. This order is verified and the items are delivered to the customer.

The process of Stock Maintenance System is that the customer login to the particular site to place the order for the customer product. The Stock Maintenance System are described sequentially through steps

- The customer login to the particular site.
- They fill the customer details.
- They place the order for their product.
- The vendor login and views the customer details and orders.

Functionalities of Customer:

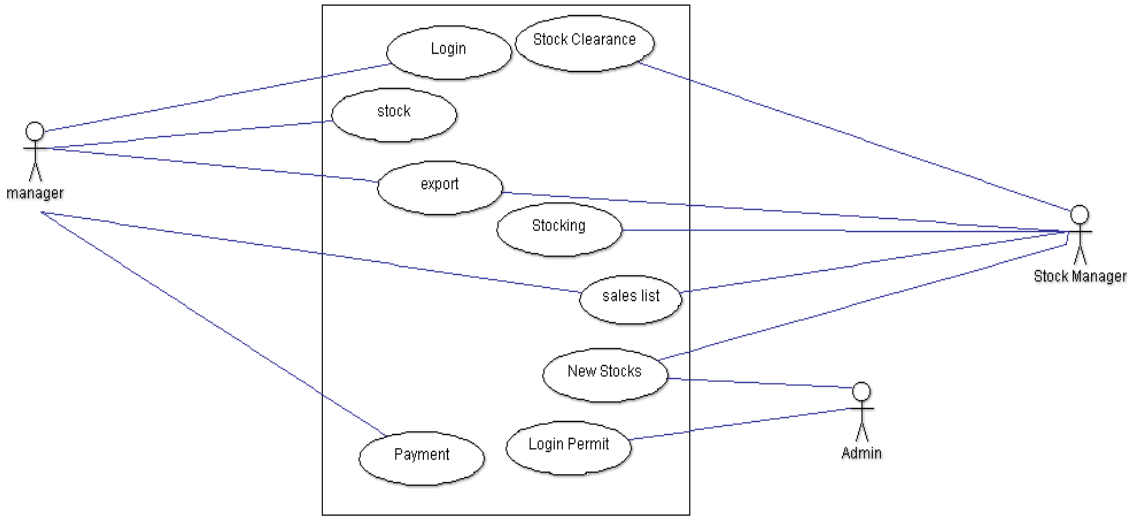
To purchase an item, Constraints checked here includes

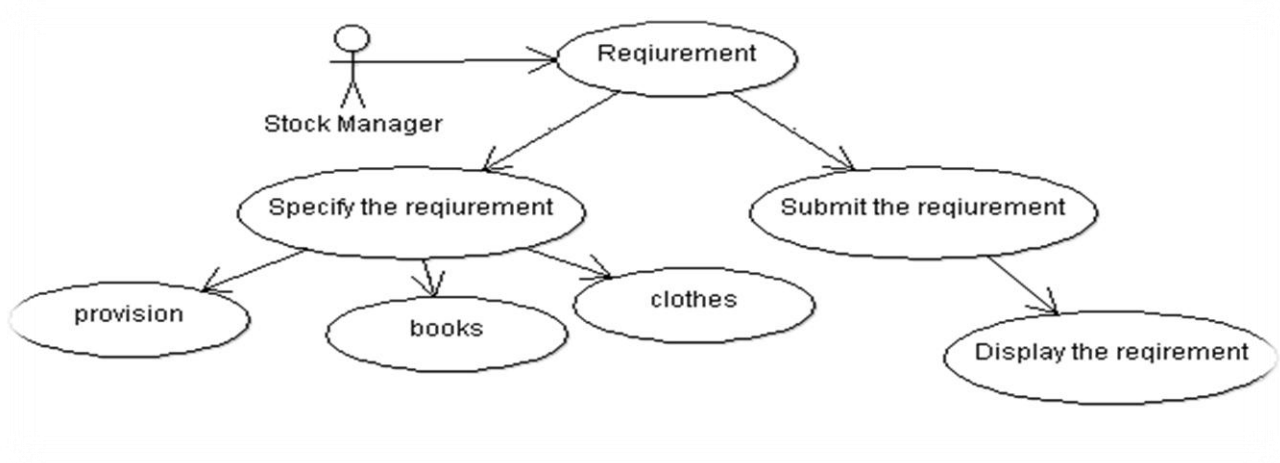
- Maximum price of the item.
- Number of item left.
- Conflicts in timing with other orders of some items.

Functionalities of Stock Manger:

- Can view items currently available.
- Can view information of the entire customer who has order the items.
- Conflicts with the items available at the stock room and the customer order.
- Can edit the items orders such as number of items left or to increase or decrease the number of availability.

UML USECASE DIAGRAM:





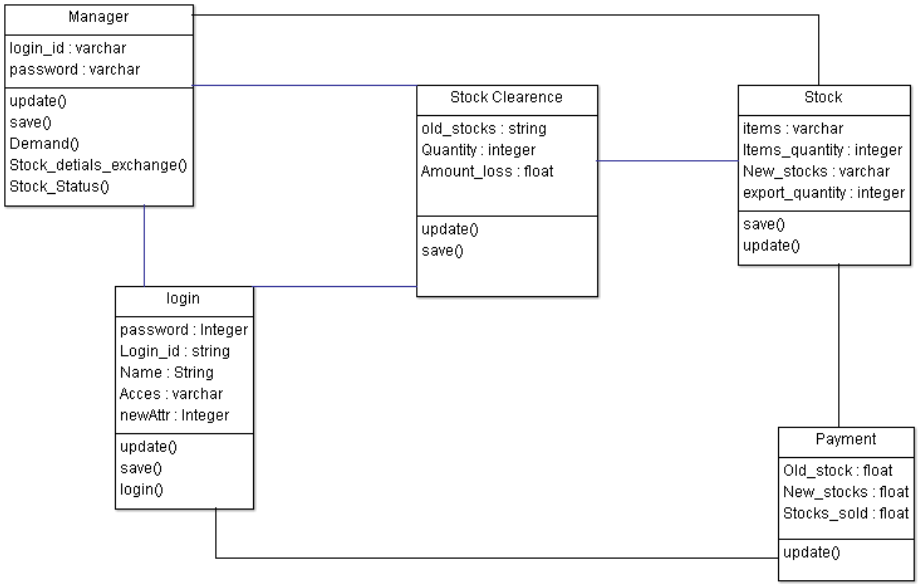
ACTOR: - Stock Manager, Manager, Admin

Use Case: - Stock Clearance, login, Stock, Export, Stock List, New Stock, and Payment.

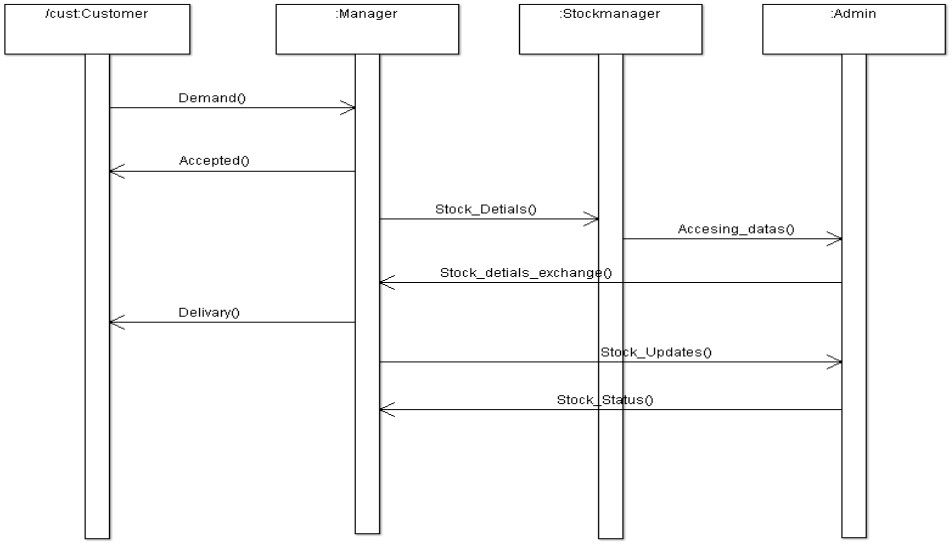
UML CLASS DIAGRAM:

- Login
- Manager
- Stock Clearance

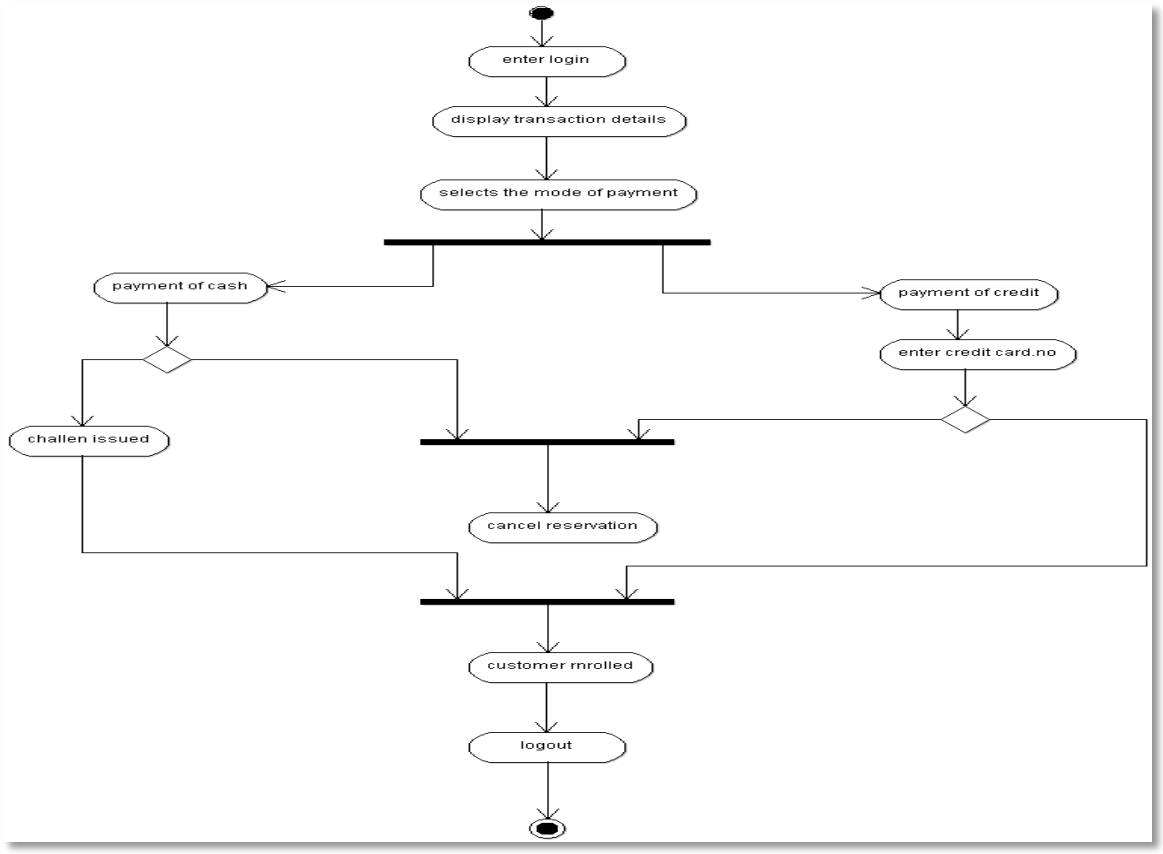
- payment
- Stock



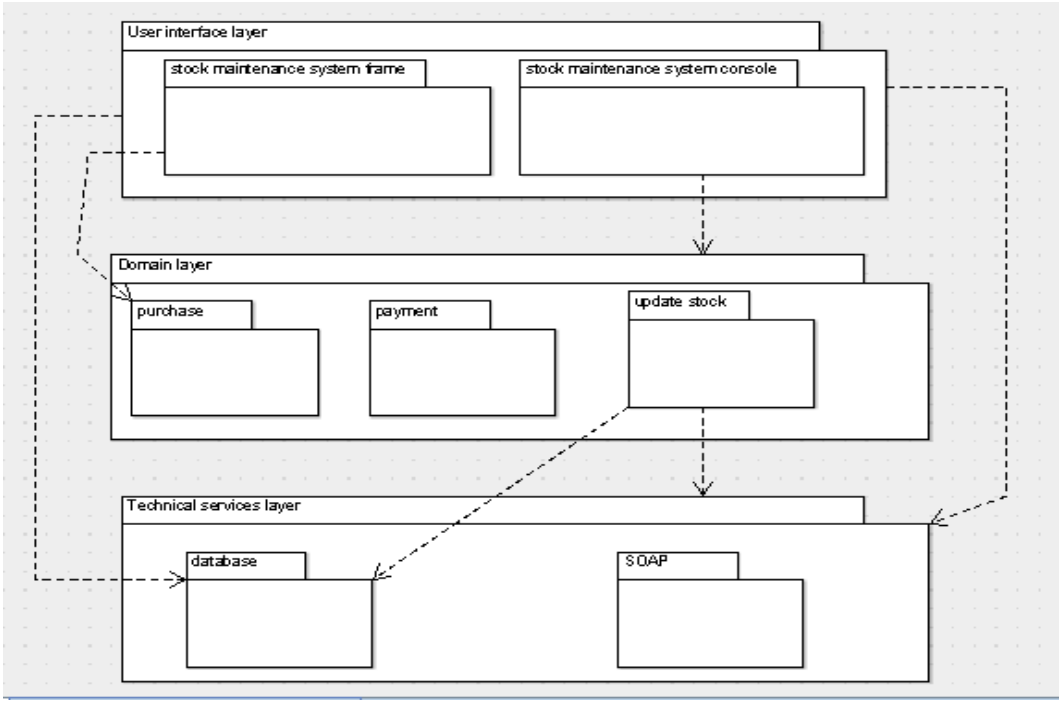
UML INTERACTION DIAGRAM:



UML ACTIVITY DIAGRAM:



UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.No.	USERNAME	PASSWORD
1	Santhosh	54792
2	Vignesh	12345
3	Sharfu	67890

S.No.	Amount	Name of the Bank	Branch of the bank	Contact
1	10,000	City Union	Vellore	9765432198
2	5,000	ICICI	Chennai	9743278367
3	15,000	SBI	Ranipet	8764352789

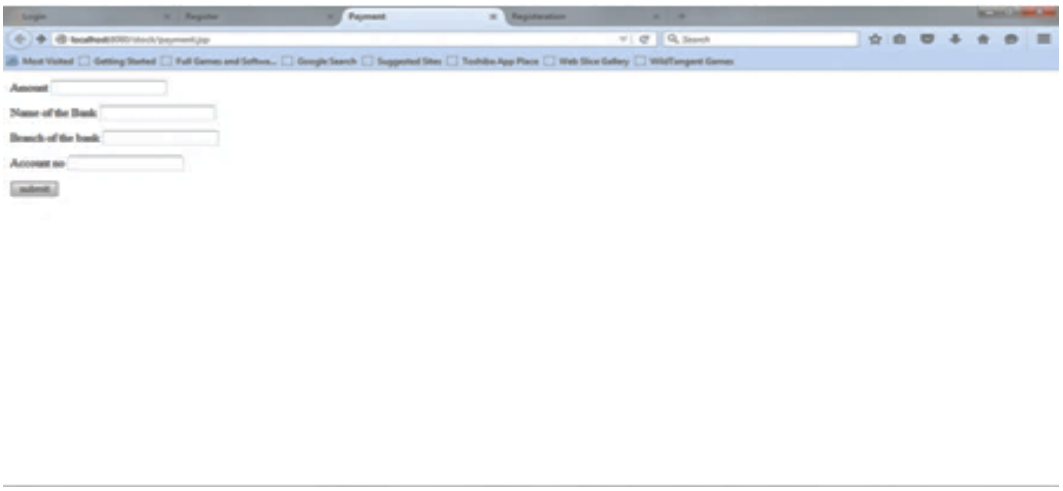
S.no	Name	Email Id	Address	Contact
1	Santhosh	santy22@gmail.com	Zee flat	9765432198
2	Vignesh	Vignesh5@gamil.com	Nethaji road	9743278367
3	Sharfu	sharfu66@gmall.com	Kaspa road	8764352789

UML DOMAIN OBJECT LAYER:

```
import java.util.Vector;

public class login {

  public Integer password;

  public string Login_id;

  public String Name;

  public varchar Acces;

  public Integer newAttr;

    public Vector  myStock Clearence;
    public Vector  myManager;
    public Vector  myPayment;

  public void update() {
  }

  public void save() {
  }
```

```
  public void login() {
  }

}

import java.util.Vector;

public class Stock Clearence {

  public string old_stocks;

  public integer amount;

  public float Amount_loss;

    public Vector  mylogin;
    public Vector  myStock;
    public Vector  myManager;

  public void update() {
  }

  public void save() {
  }

}
```

USER INTERFACE LAYER:



Result:

Thus the Stock Maintenance System has been done successfully by using Argo-UML.

EX.NO.: 5	ONLINE COURSE REGISTRATION

AIM:

To design Online Course Registration System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying Online Course Registration, software has been created by designing through ARGO-UML tool.

The exam registration is an application in which applicant can register themselves for the exam. The details of the students who have registered for the examination will be stored in a database and will be maintained. The registered details can then be verified for any fraudulent or duplication and can be removed if found so.

PROBLEM STATEMENT:

Exam registration system is used in the effective dispatch of registration from to all of the students this system adopts a comprehensive approach to minimize the manual work and schedule resources, time in cogent manner the core of the system is to get the online registration from (with details such as name , reg.no , etc.,) filled by the student whose statement is verified for it is genuineness by the exam registration system with respect to the already existing information in the database.

The process of students accessing the registration application and applying for the examination by filling out the form with proper details and then the authorities verify those details given for truth and correctness are sequenced through steps

a. The students access exam registration application.

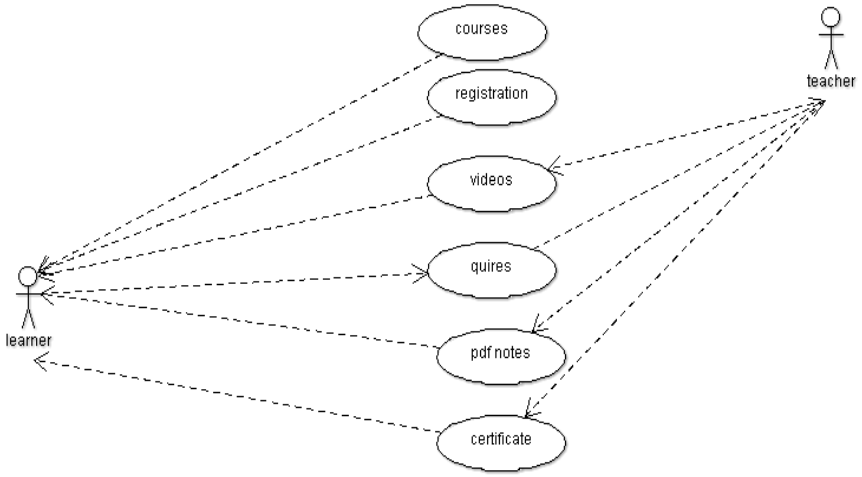
b. They fill out the form with correct and eligible details.

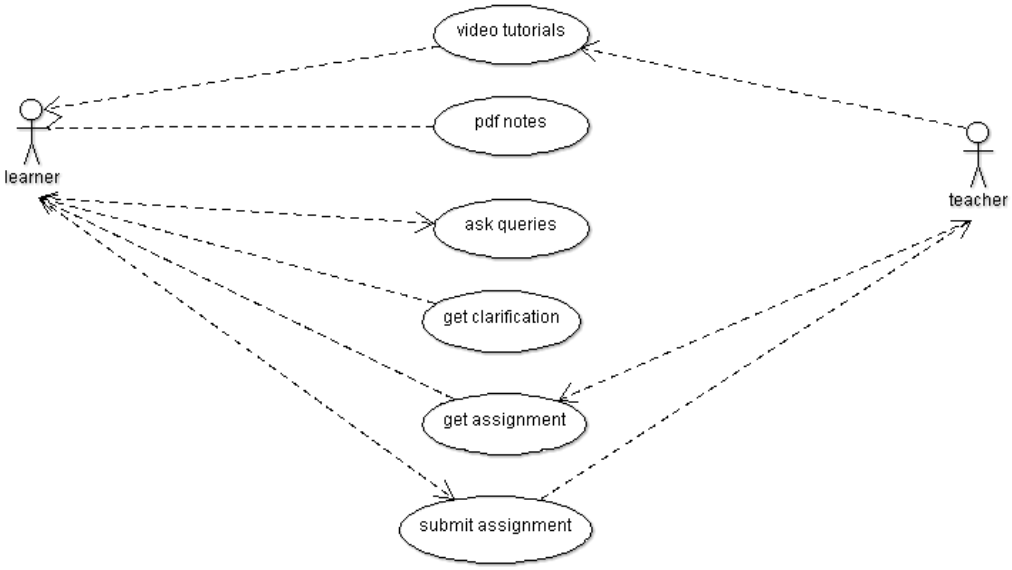
c. They complete the payment process.

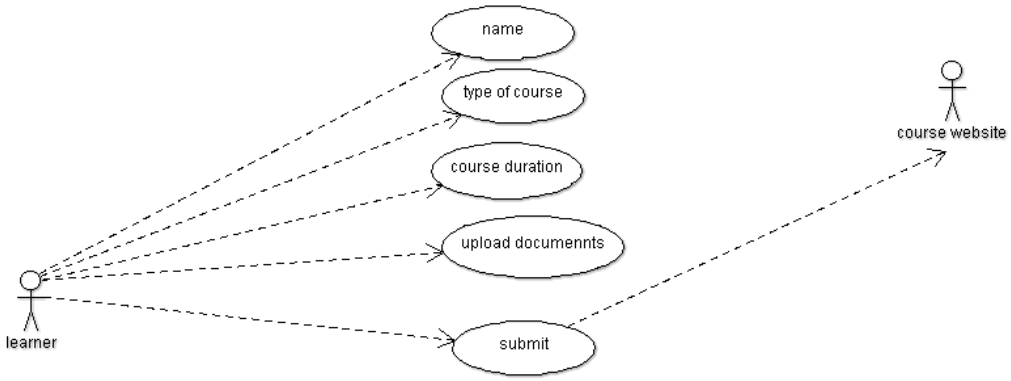
d. The authorities verify or check the details.

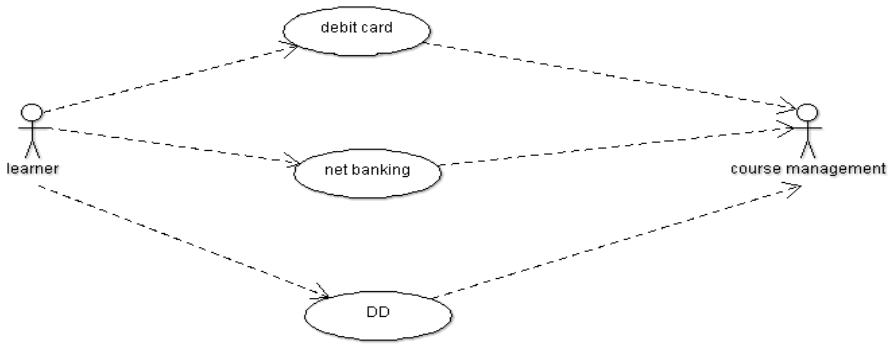
e. After all verification the exam registration database is finalized.

UML USECASE DIAGRAM:









The actors in this use case diagram are Students, Interface and Database. The use cases are the activities performed by actors.

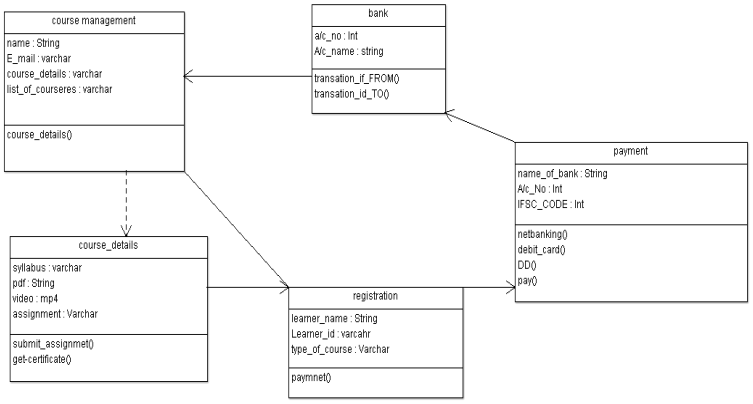
a. Student fills outs the form in the form filling process.

b. The interface checks and validates registered details.

c. Then the database is searched for details and verified.

d. Database stores the details and returns acknowledgement.

UML CLASS DIAGRAM:

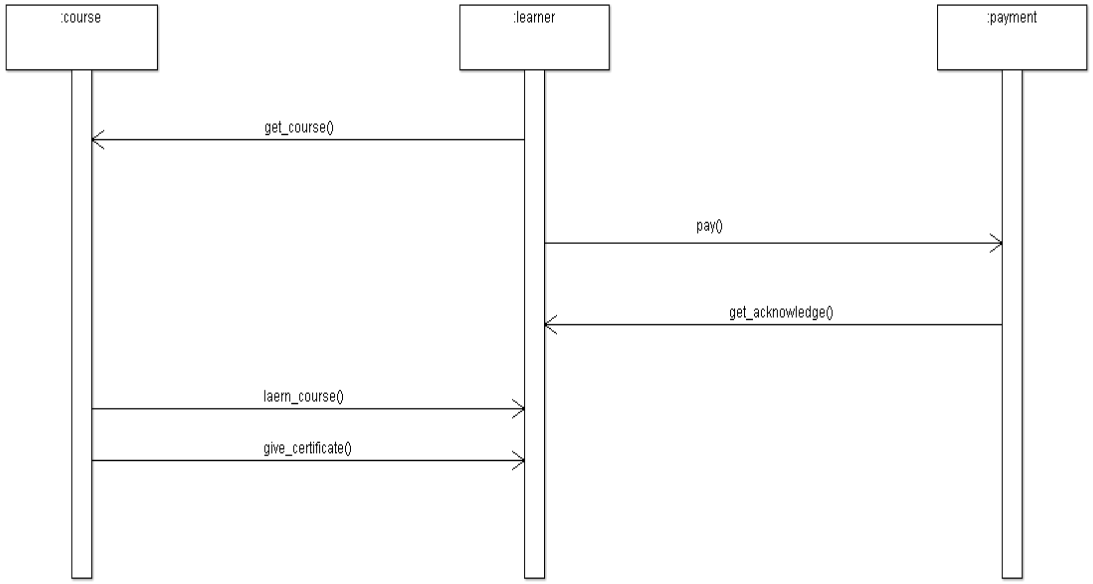


- Course Management
- Course Details
- Registration
- Payment
- Bank

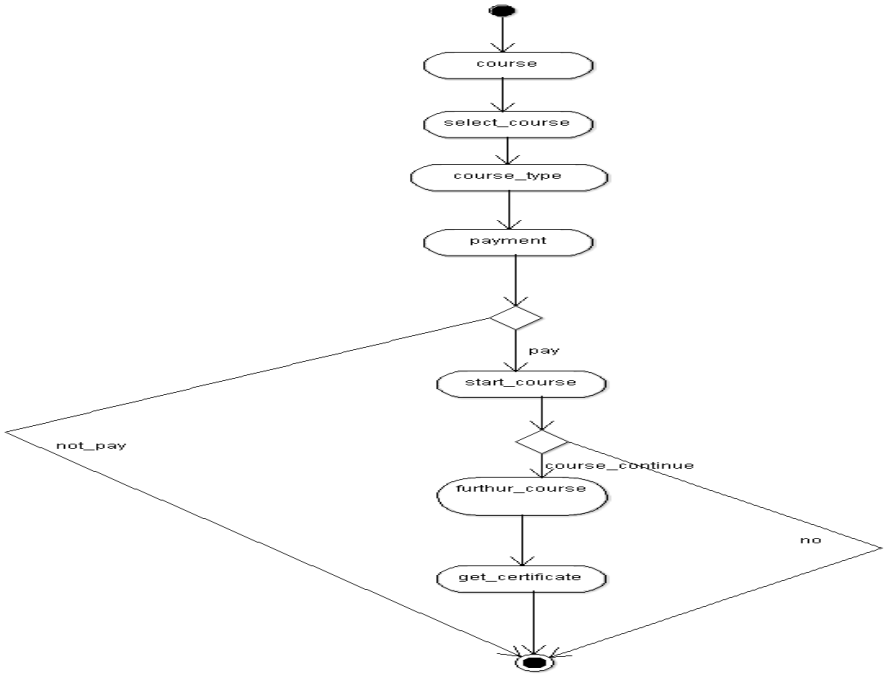
UML INTERACTION DIAGRAM:

The sequence diagram describes the sequence of steps to show:

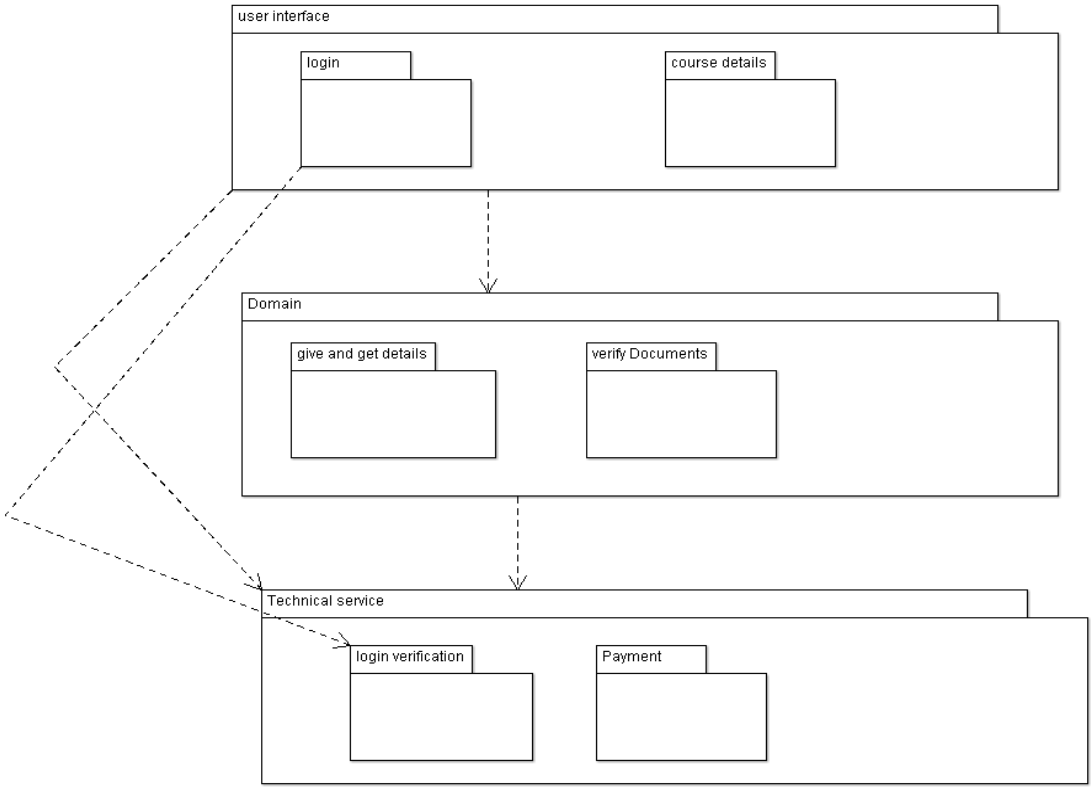
- The candidate selects the exam and enters their detail in the online registration form.
- Then the candidate selects the payment mode like Credit card, Debit card and Net banking.
- If the payment is not completed then it goes for repayment and after completion of the payment the application will be generated.
- The detail of exam is viewed by the candidate.



UML ACTIVITY DIAGRAM:



UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

Name	Date of Birth	Qualification	Gender	Select Course	Phone Number
Saravana	01-05-1997	B.E	M	NEET	9087349490
Surya	06-07-1998	B.E	M	IIT	8337849320
Vinoth	09-08-1998	B.E	M	VIT	7334739390

Register No	Name	Application No	Hall Ticket	Center	Date of Examiner
510615104078	Saravana	1014327	654789	VOC COLLEGE	06-10-2017
501615104088	Surya	1014329	698547	CAHCET	05-11-2107
510615104105	Vinoth	1014320	647895	VIT	05-12-2017

UML DOMAIN OBJECT LAYER:

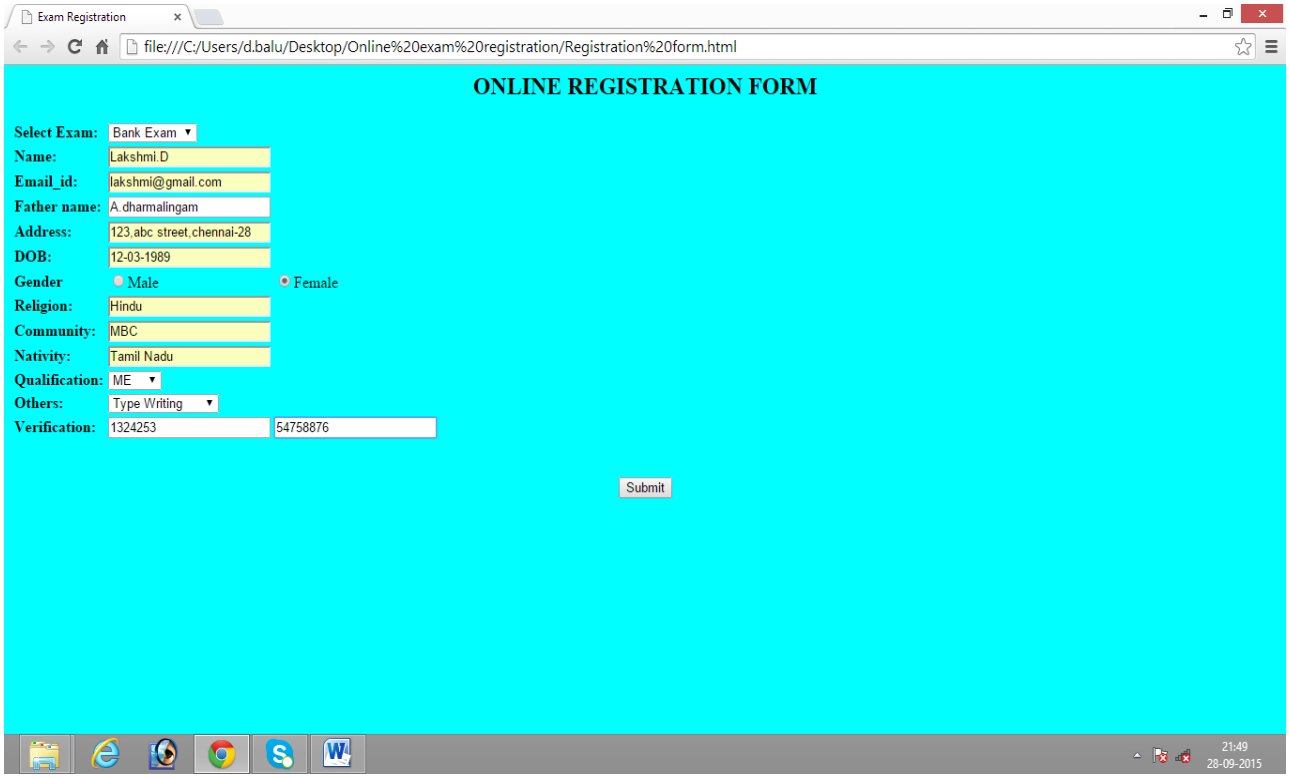
PERSONAL INFO:-

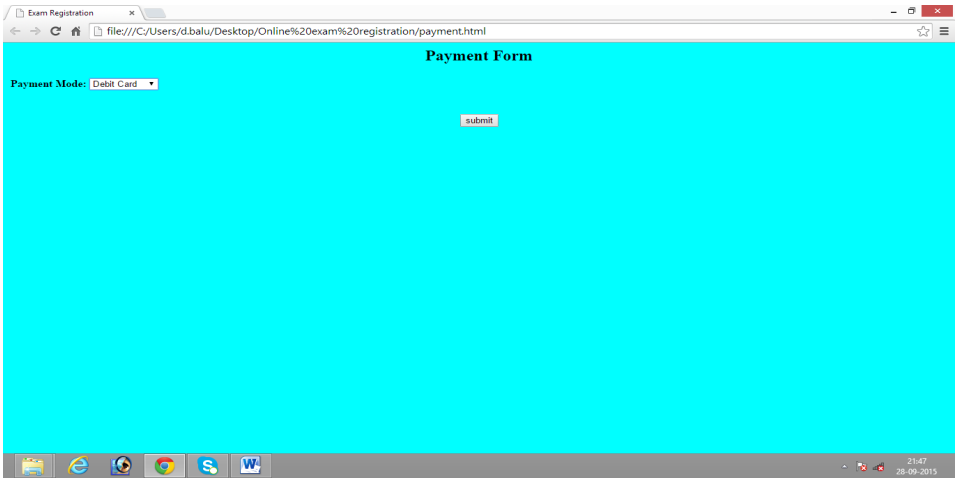
```
import java.util.Vector;

public class personal Info {
  public String persss_der;
  public string verify;
    public Vector  mycandidate;
  public void save() {
  }
  public void modify() {
 }
  public void newOperation() {
  }

}
```

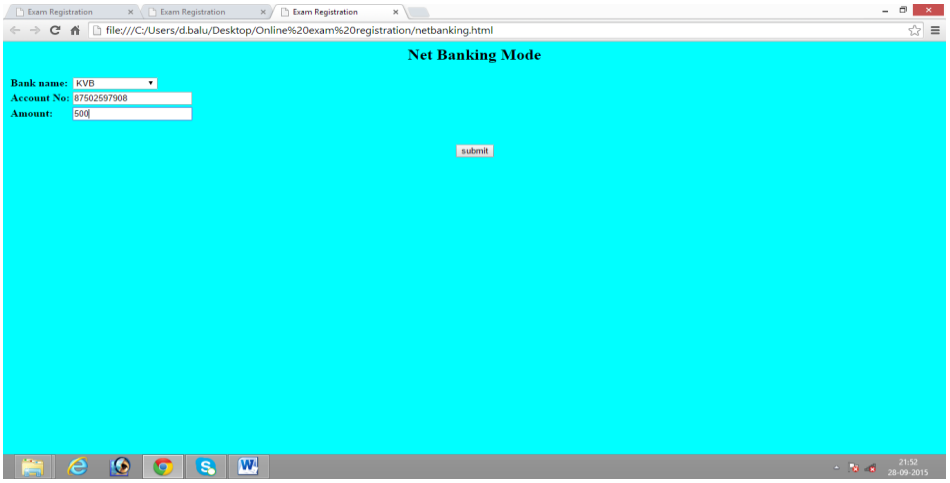
ADMIN:-

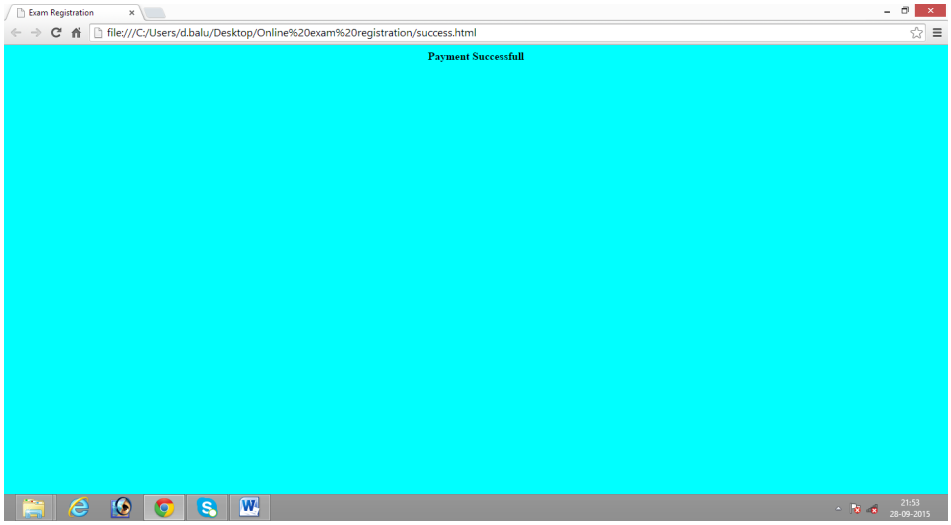
```
import java.util.Vector;

public class admin {
  public string candidateInfo;
    public Vector  myFees Details;
      public Vector  myDebit  card;
  public void save() {
  }
  public void Discard() {
  }
}
```

USER INTERFACE LAYER:









Result:

Thus the Online Course Reservation System has been done successfully by using Argo-UML.

EX.NO.:6	E-TICKETING

AIM:

To design E-Ticketing System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

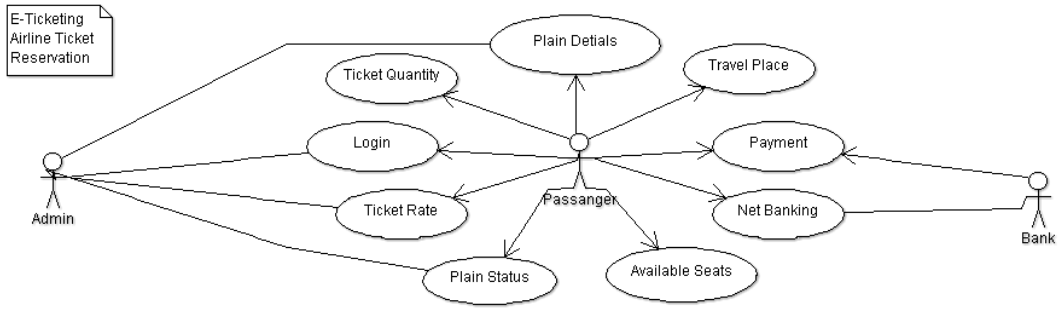
To simplify the process of applying e-ticketing, software has been created by designing through ARGO-UML tool.

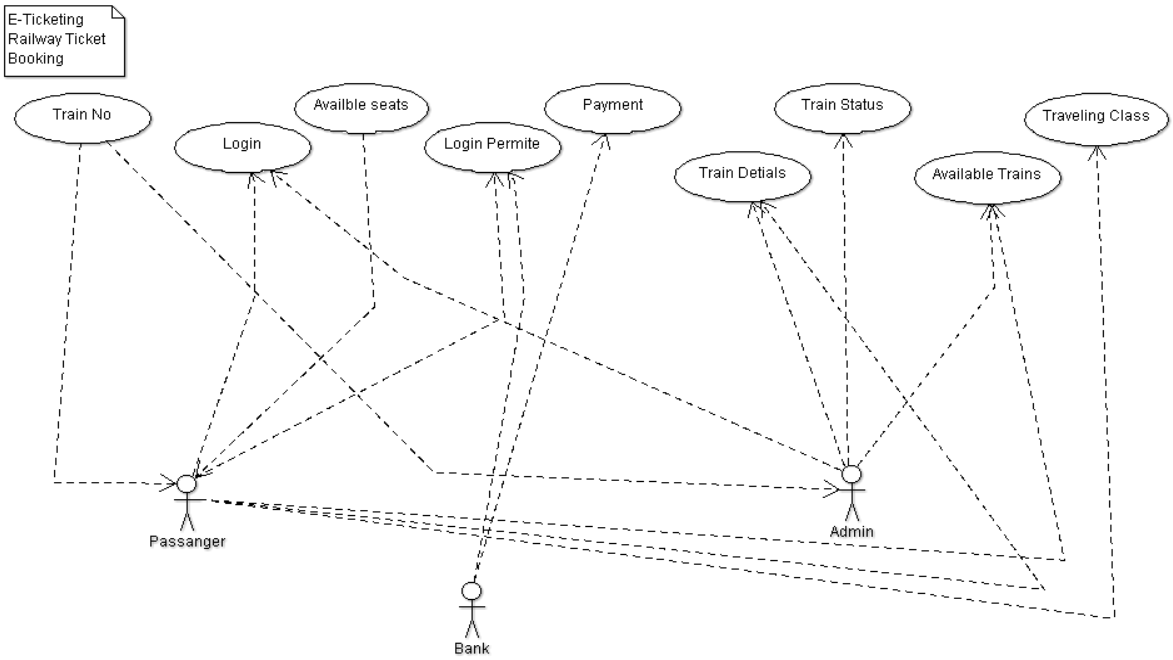
In the E-Ticketing system the main process is a applicant have to login the database then the database verifies that particular username and password then the user must fill the details about their personal details then selecting the flight and the database books the ticket then send it to the applicant then searching the Availability of seats or else cancelling the process.

PROBLEM STATEMENT:

- The E-Ticketing system is the initial requirement to develop the project about the mechanism of the E-ticketing system what the process do at all.
- The requirement are analyzed and refined which enables the end users to efficiently use the E-ticketing system.
- The complete project is developed after the whole project analysis explaining about scope and project statement is prepared.
- The main scope for this project is the applicant should reserve for the flight ticket.
- First the applicant wants to login to the database after that the person wants to fill their details.
- Then the database will search for ticket or else the person will cancel the ticket if he/she is in not need.

UML USECASE DIAGRAM:



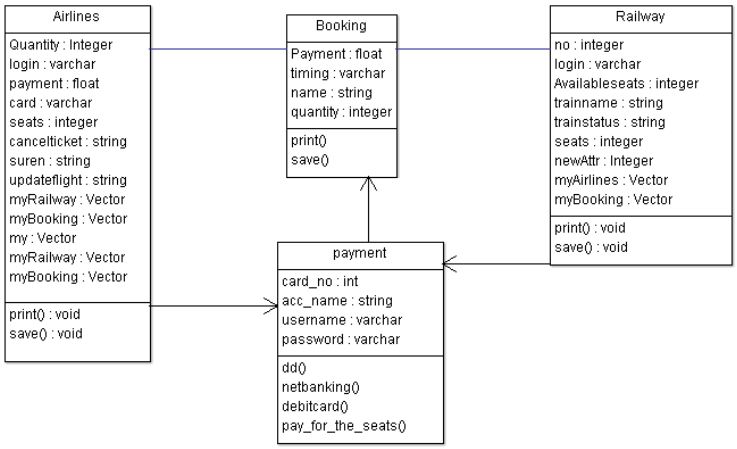


UML CLASS DIAGRAM:

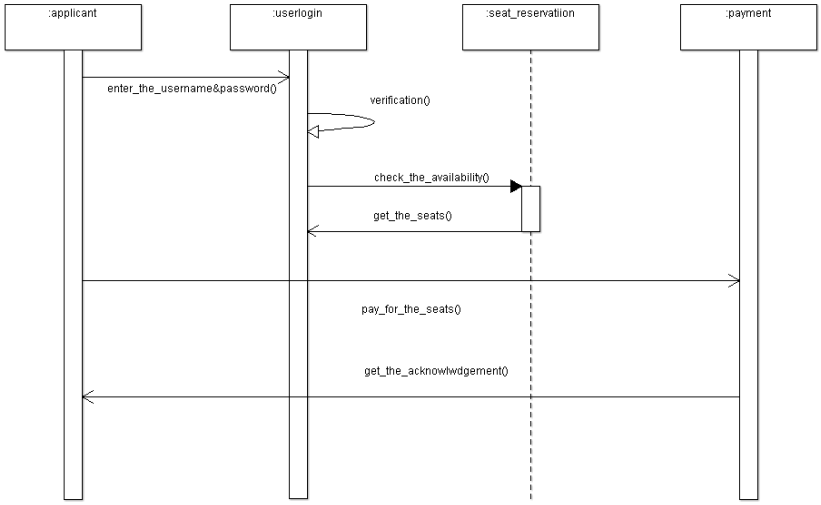
This class diagram has two classes applicant, E-Ticketing Data Base.

Applicant -logins the E-Ticketing and filling the required data fields.

E-Ticketing Data Base-verify the login and filling the details and selected applicant details are stored in it.
This diagram shows the classes associated with this system and the way how they are linked with each other. This diagram also shows the attributes and methods of the class. The first partition shows the name of the class and second shows the attributes and third shows the methods.



UML INTERACTION DIAGRAM:

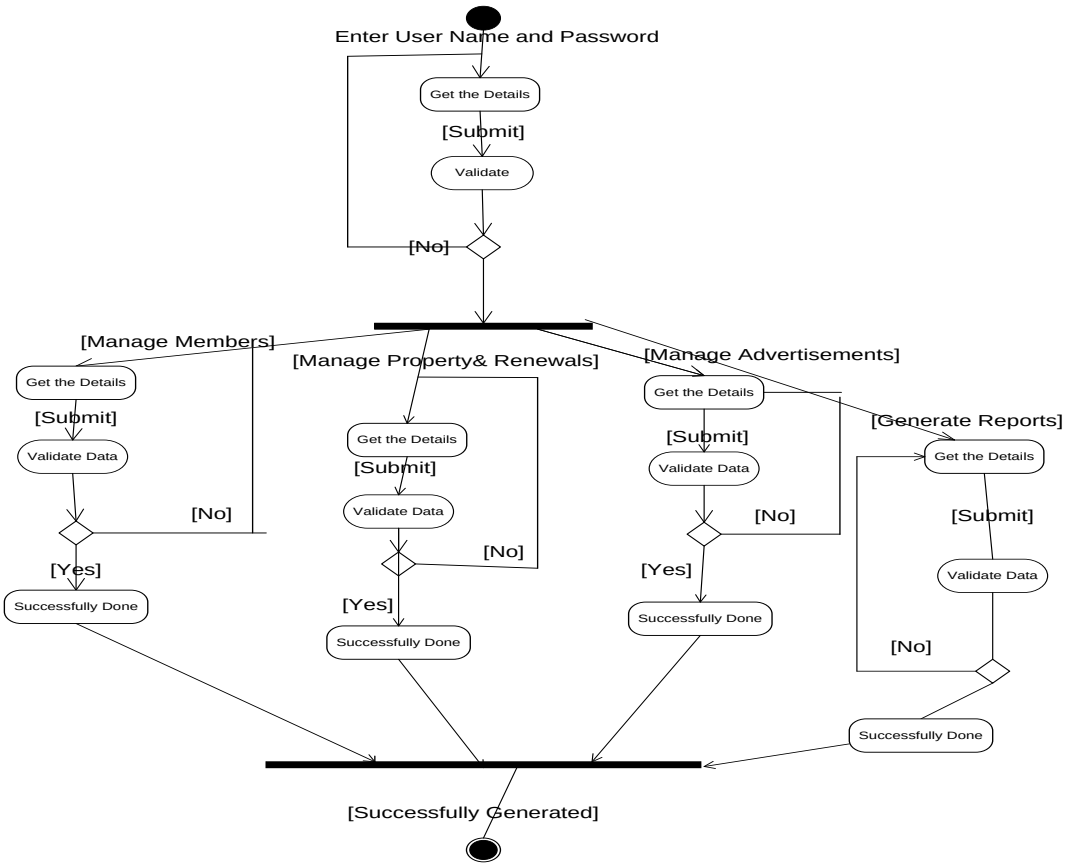


This sequence diagram describes the sequence of steps to show:

- Applicants are used to login the form and then it's verify the username and password.

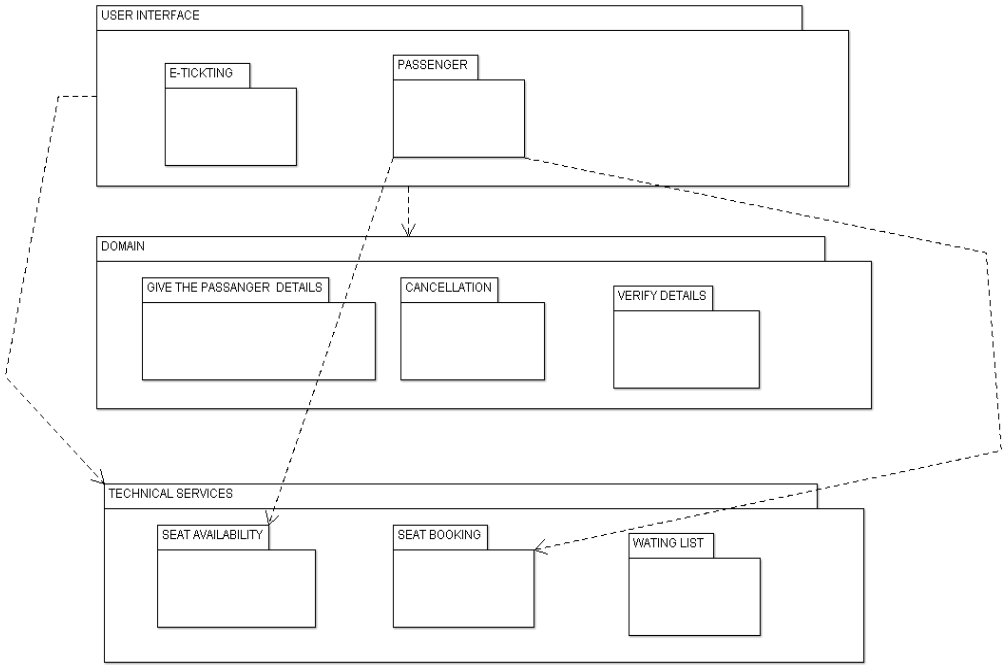
- If the password and username are correct then applicants are used to login the filling details.
- Applicants are used to selecting the Movies and book the tickets.
- Now the E-Ticketing Data Base verify the filling Details.
- And then the E-Ticketing Data Base displays the ticket information.
- In case of any sudden change of the plan, The applicant can cancel the ticket

UML ACTIITY DIAGRAM:



- This activity diagram describes the behavior of the system.
- First state is login where the applicant login to the E-Ticketing system.
- The next state is filling details the applicants are used to fill the form.
- Then applicant used to selecting the flight.
- The applicant appears for book ticket and search details from E-Ticketing Data Base.

UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.No	Passenger	Train_No	From	Destination	Spare
1.	ChanBasha	123782	Vellore	Ooty	300
2.	Thouseef	347832	Ambur	Banglore	400
3.	Syed Owies	434792	Katpadi	Delhi	1200

Applicant_id	Passenger	Train_No	Seat_No	Verify_id
384392034	ChanBasha	123782	S10-30	Aadhar card
324832303	Thouseef	347832	D6-29	Pan card
324839304	Syed Owies	434792	A1-20	Aadhar card

UML DOMAIN OBJECT LAYER:

Booking:

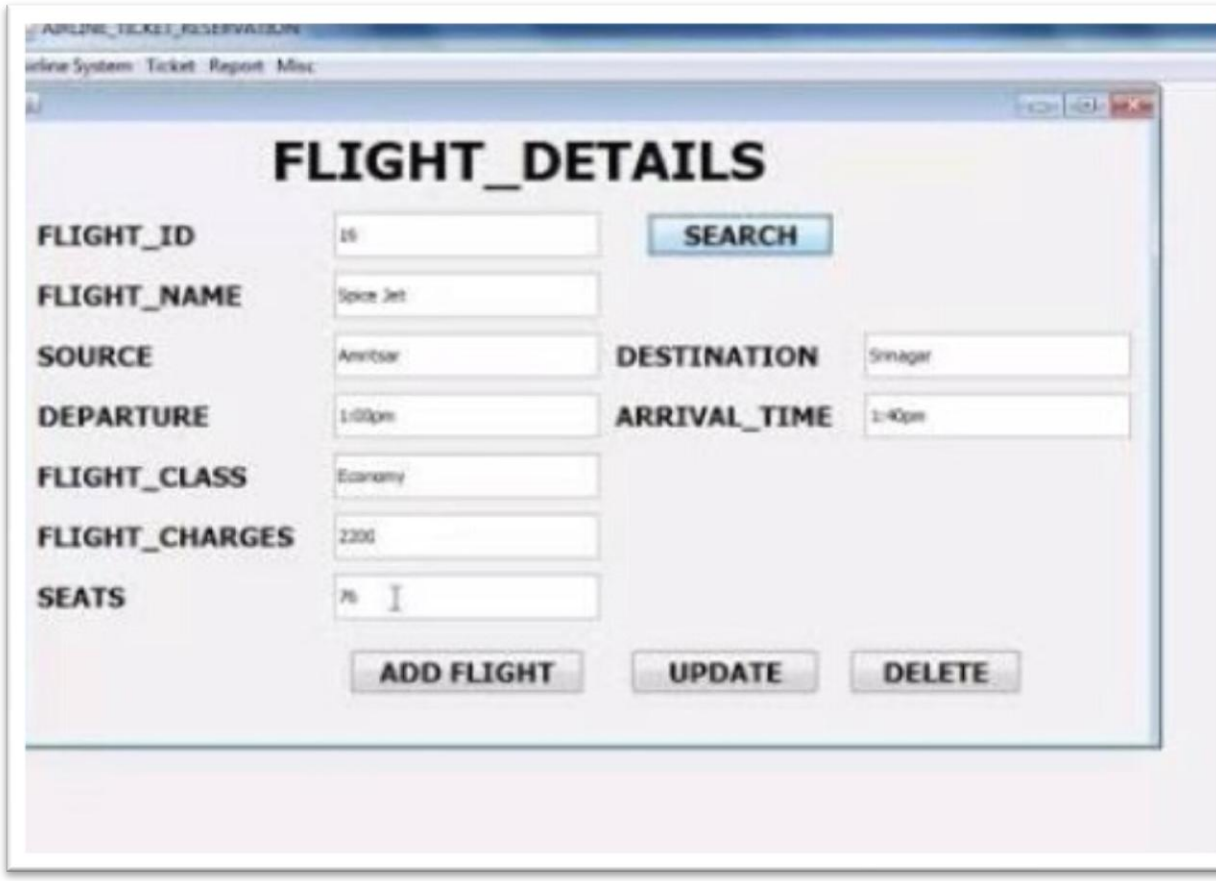
```
import java.util.Vector;

public class Booking
{

  public float Payment;

  public varchar timing;

  public string name;

  public integer quantity;

  public Vector  myAirlines;
  public Vector  myRailway;

  public void print()
  {
  }

  public void save()
  {
  }

}
```

Railways:

```
import java.util.Vector;

public class Railway implements Booking
{
  private integer no;

  public varchar login;

  public integer Availableseats;

  public string trainname;
```

```
public string trainstatus;

private integer seats;

public Integer newAttr;

  public Vector  myAirlines;
  public Vector  myBooking;

public void print() {  }

public void save() {  }
        }
```

USER INTERFACE LAYER:



Result:

Thus the E-Ticketing has been done successfully by using Argo-UML.

EX.NO.:7	SOFTWARE PERSONNEL MANAGEMENT SYSTEM

AIM:

To design Software Personnel Management System by using Argo-UML tool.

PROCEDURE:

- The software to be designed will control a simulated software personnel management system.
- This software is designed for the process of knowing the details of a person works in a software company. The details are being stored in the central management system for the crosschecking the person's details.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying Software Personal Management, software has been created by designing through ARGO-UML tool.

The employee management system is used to manage our personnel things such as maintaining databases in offices etc. this project is easy for the CEO to handle the details. This is personally used for CEO.

PROBLEM STATEMENT:

The CEO must enter the name and password to login the form and select the particular employee to view the details about that employee and maintaining the employee details personally. This process of employee management system are described sequentially through following steps,

- The CEO login to the employee management system.
- He/she search for the list of employees.
- Then select the particular employee.
- Then view the details of that employee.
- After displaying the employee details then logout.

UML USECASE DIAGRAM:

The actors in this use case diagram are Student, Staffs and Library Database. The use cases are the activities performed by actors.

The use case diagram in the employee management system illustrates the sequence of sequencing and describing an interaction between a CEO and a system.

Login:

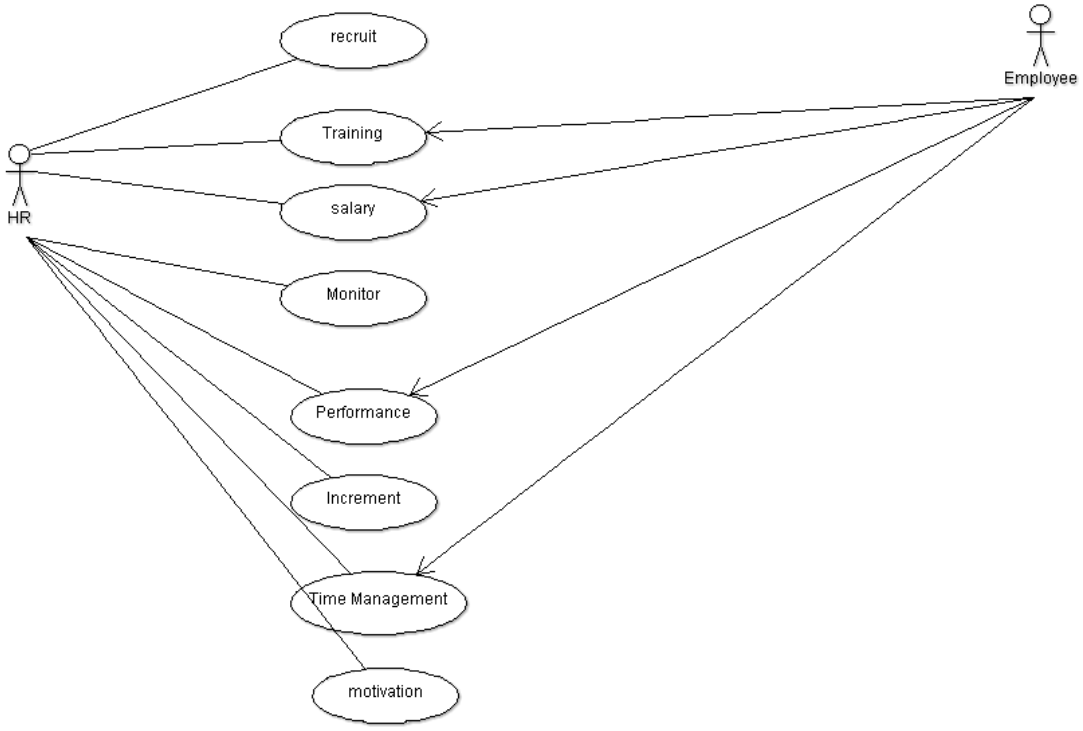
This use case gives as entry to the CEO and the database.

List of employee:

This will create the situation for the CEO to select particular employee from the available list.

Employee details:

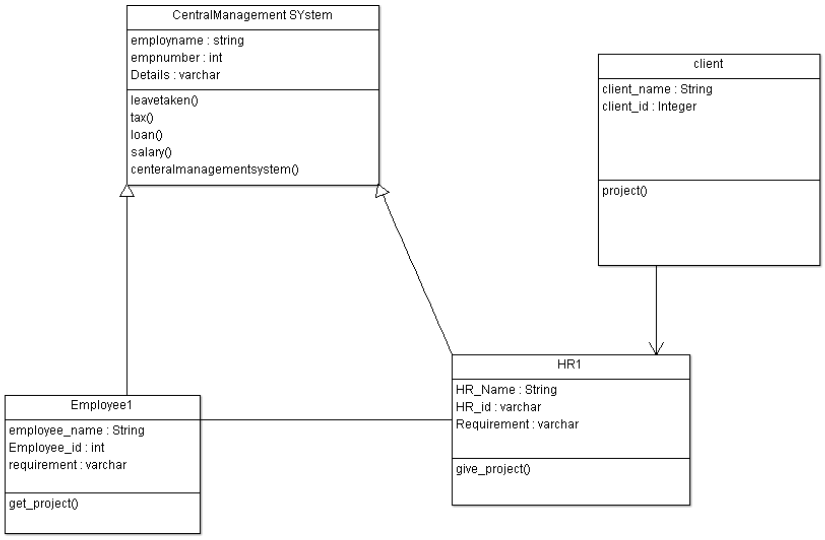
The CEO can able to view the details of the employee using this use case.



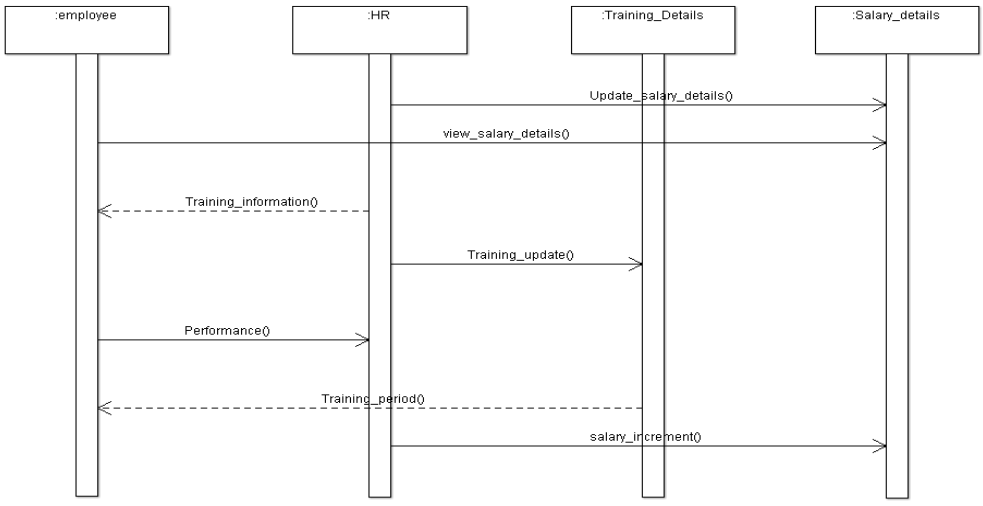
UML CLASS DIAGRAM:

The Classes used in this project are:

- **CEO:** The CEO has to login the form by specifying the name and password of him.
- **Database:** The database checks whether the CEO has given the name and password accordingly if not the error message will be displayed.
- **Available employees:** The database is connects to the list of available employees and the CEO if wants then select the employee from it.



UML INTERACTION DIAGRAM:

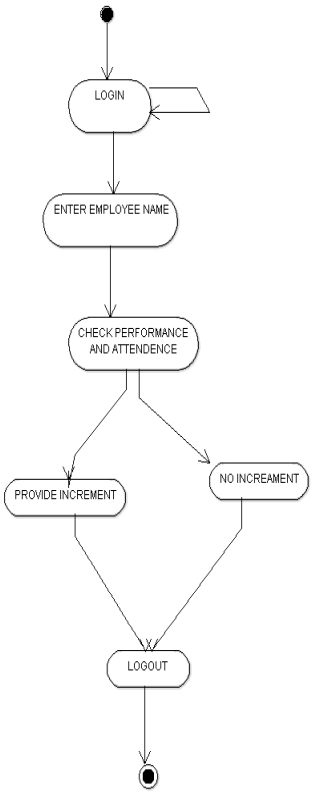


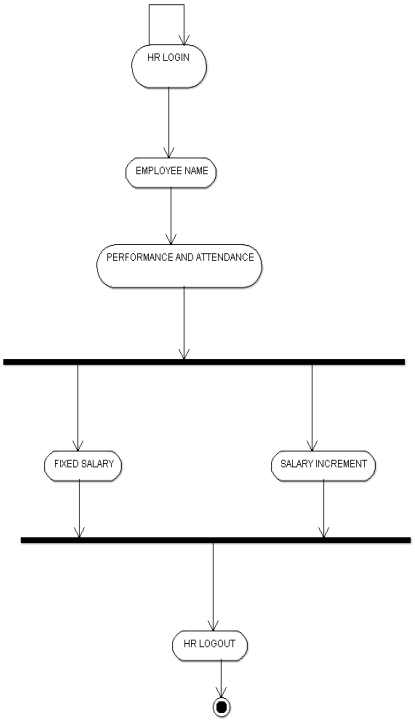
The CEO must enter his name and password to login the employee management system. The verification process is undergone by the database.

If the details are correct he can enter to the system otherwise error is displayed. After login the details of the particular employee is viewed by the CEO.

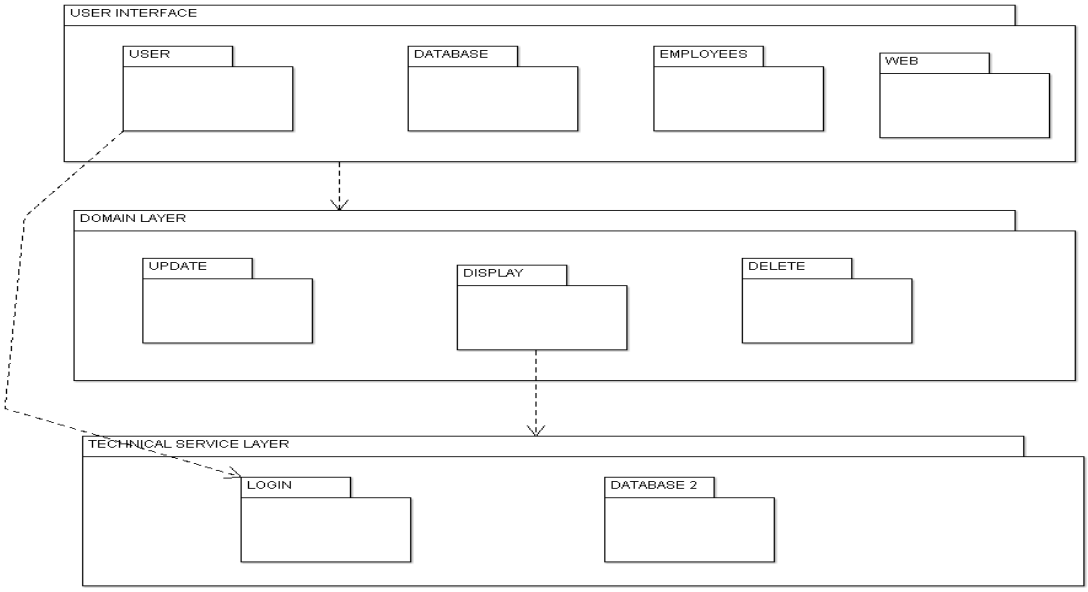
Finally he is logged out from the system.

UML ACTIVITY DIAGRAM:





UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.NO	EMPLOYEE NAME	PASSWORD
1	Surya	54792
2	Saravana	12345
3	Seenu	67890

S.no	Amount	Name of the Bank	Branch of the bank	Contact
1	20,000	City Union	Vellore	9765432198
2	15,000	ICICI	Chennai	9743278367
3	25,000	SBI	Ranipet	8764352789

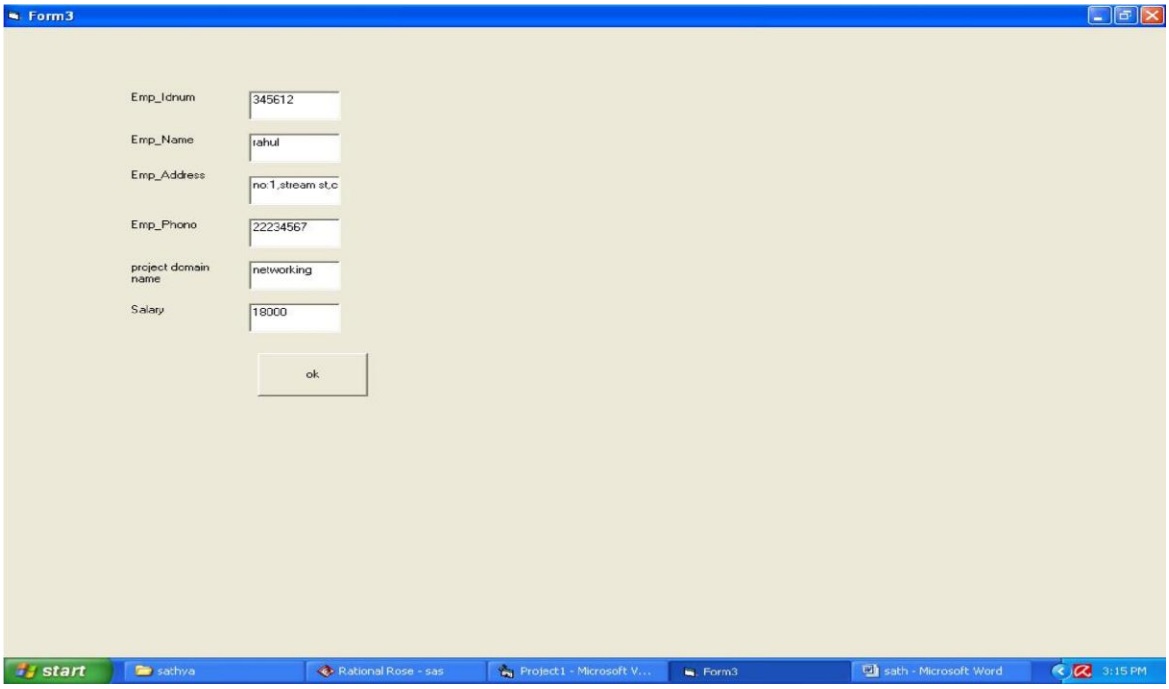
S.no	Employee Name	Email Id	Address	Contact
1	Surya	surya22@gmail.com	Zee flat	9765432198
2	Saravana	sarvanan@gamil.com	Nethaji road	9743278367
3	Seenu	Seenu@gmai.com	Kaspa road	8764352789

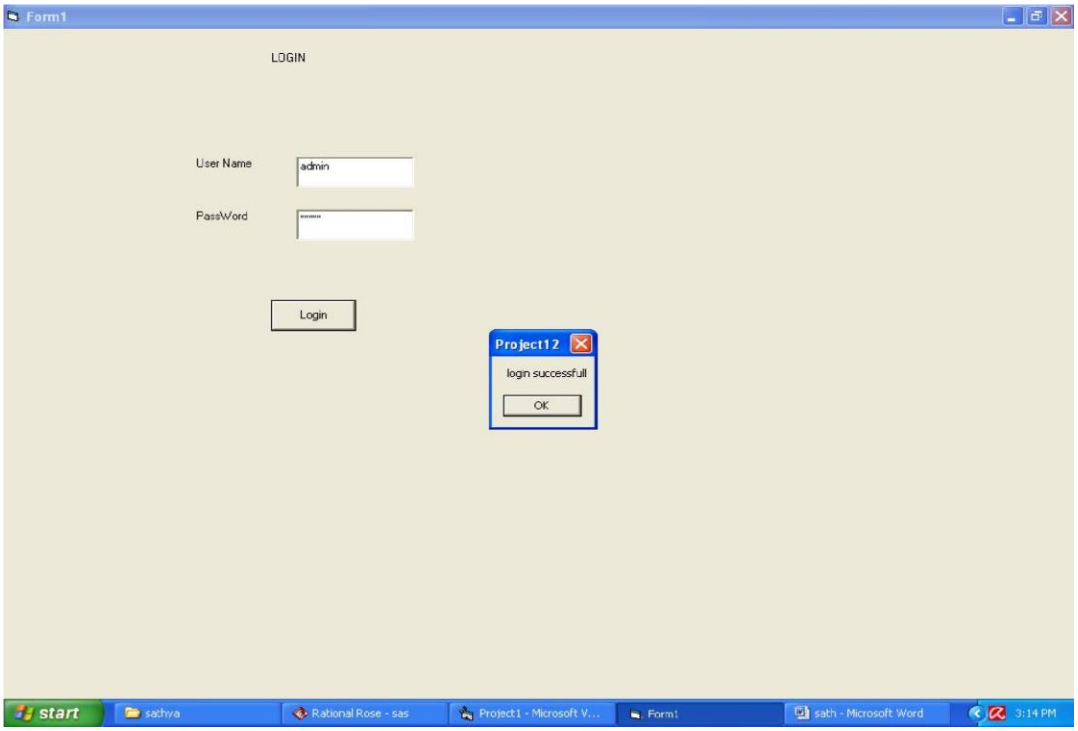
UML DOMAIN OBJECT LAYER:

```
public class CentralManagement System {
  public string employname;
  public int empnumber;
```

```
  public varchar Details;
  public void leavetaken() {
  }
  public void tax() {
  }
  public void loan() {
  }
  public void salary() {
  }
  public void centeralmanagementsystem() {
  }
 }

 public class HR1 extends CentralManagement System,
 CentralManagement {
  public varchar checkdetails;
  public Integer newattr;
     public Vector  myEmployee1;
  public void lossofpay() {
  }
  public void tax() {
  }
  public void project() {
  }
 }
```

USER INTERFACE LAYER:





Result:

Thus the Software Personal Management System has been done successfully by using Argo-UML.

EX.NO.: 8	**CREDIT CARD PROCESSING SYSTEM**

AIM:

To design Credit Card Processing System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying Credit Card Processing, software has been created by designing through ARGO-UML tool.

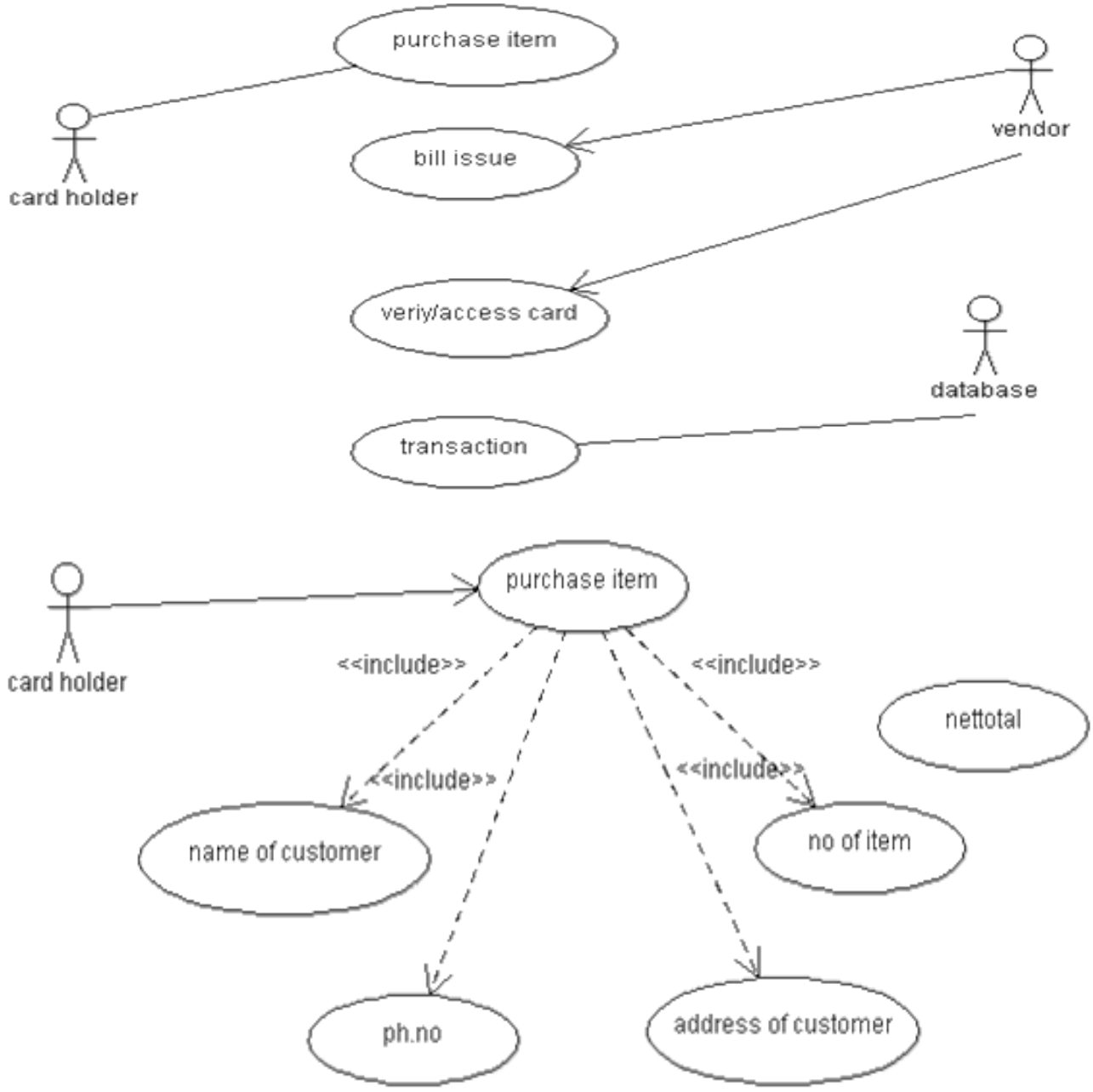
Online payment system can left people who purchase something and make payments using their credit card their bank account through internet. The problem here is to build up a reliable, affordable, secure and scalable online transaction processing systems do that consumers and business merchant bank can allow the business merchant to accept credit cards over the internet.

PROBLEM STATEMENT:

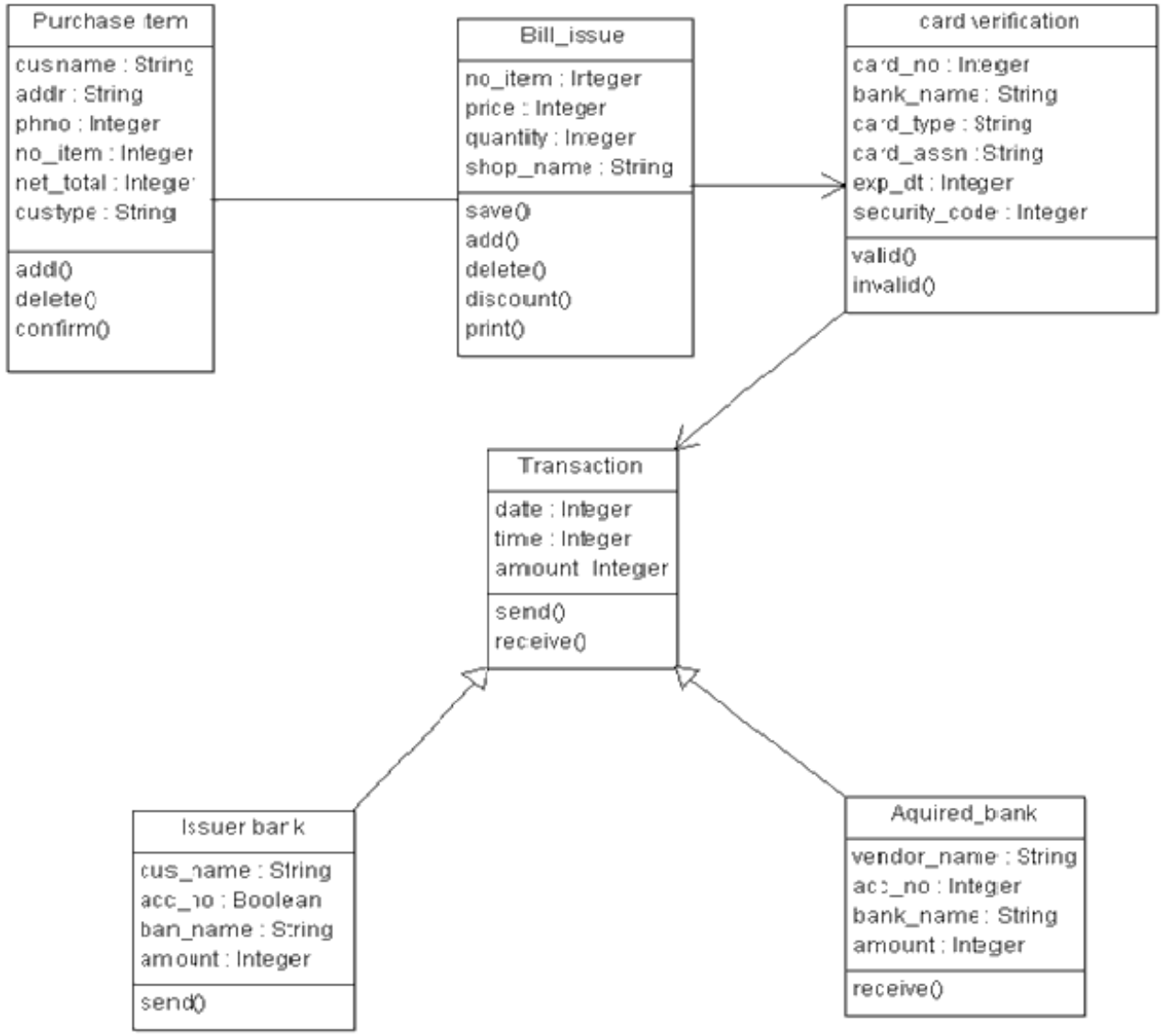
- When the consumer decides to buy something the merchant's commerce application prompts the consumer for credit card information usually along with other information such as a shipping address.
- The consumer enters payment information either into a form secured by secure socket layer (SSL) protocol or into an application such as internet explorer that is complaint with the secure electronic transaction specification. With the secure form the payment information is protected by SSL.
- Using the payment software incorporated in the web server the merchant sends the encrypted transaction to the acquiring
- Processor for authorization.
- The acquiring processor either authorizes a certain amount of money. An authorization reduces the available credit limit but does not actually put a charge on the customer's bill or move money to the merchant.
- If transaction is authorized a "capture" is next step.
- If "cancels" void is generated if consumer returns goods after the transaction has been captured credit given is cancelled.

- Final step is to “settle” the transaction between the merchant and acquiring processor.

UML USECASE DIAGRAM:

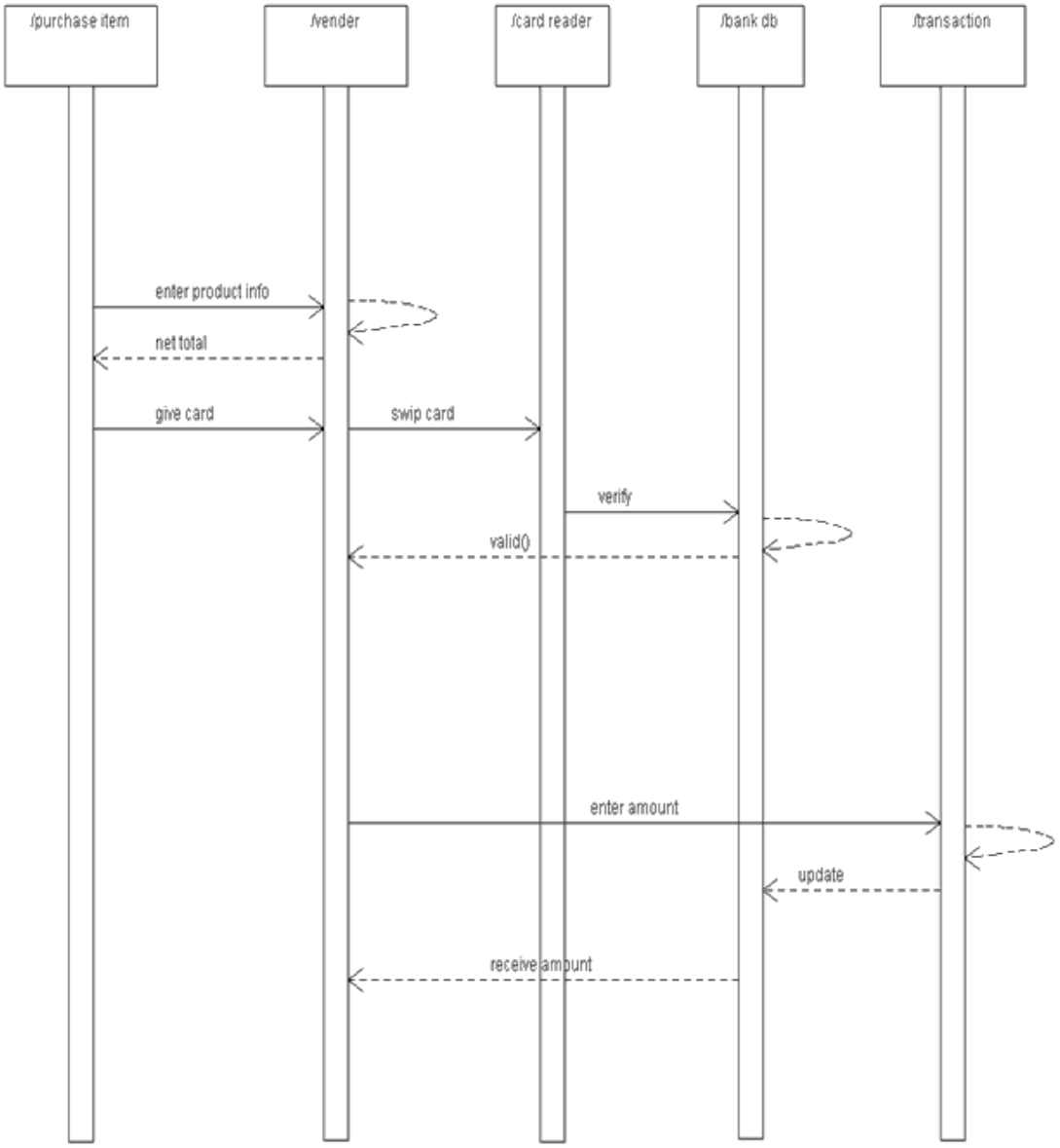


UML CLASS DIAGRAM:



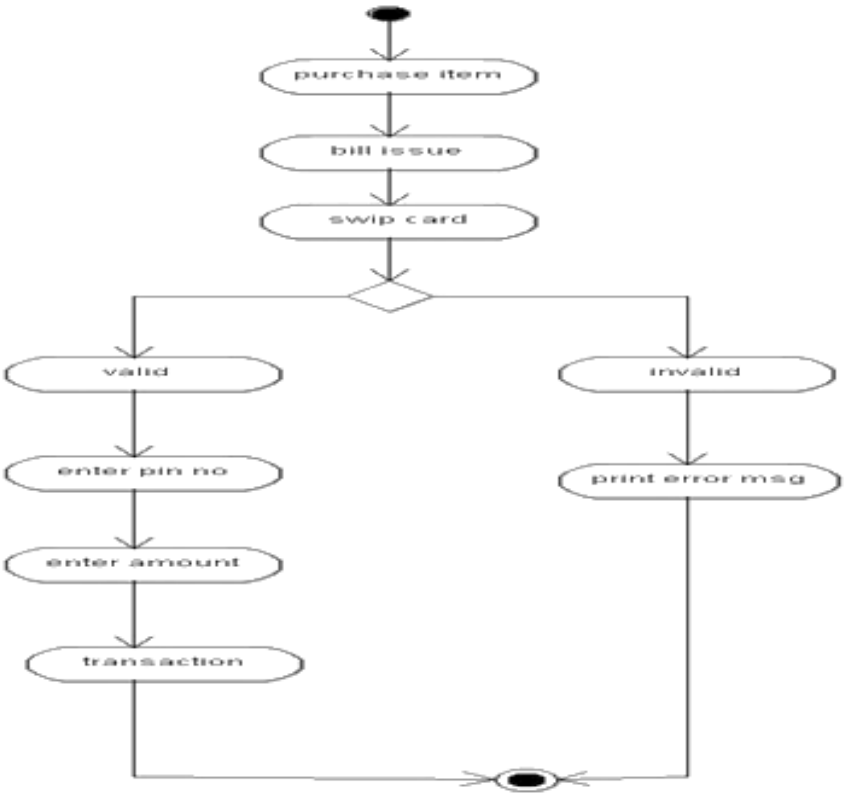
- Purchase item
- Bill issue
- Card verification
- transaction
- Bank database

UML INTERACTION DIAGRAM:

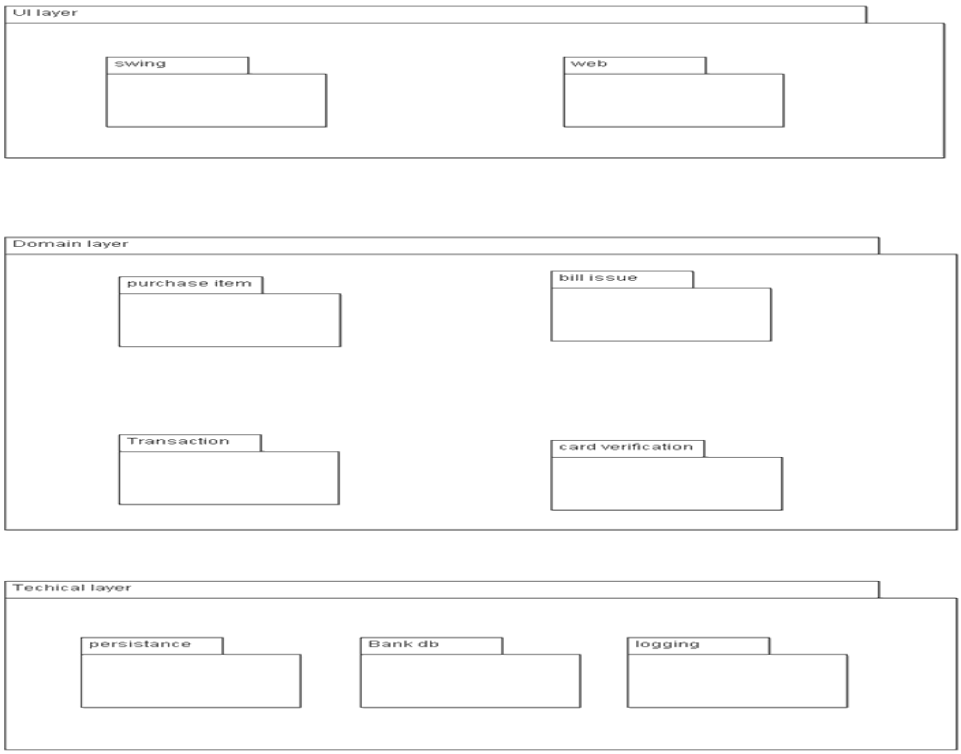


UML ACTIVITY DIAGRAM:

- Purchase item
- Bill issue
- Swipe the card
- Verification
- Transaction



UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.No.	Acc_Holder	Acc_No	Card_No	Card_Name	Validity
1.	Yogesh	2075731634	4383462074632073	VISA	03/33
2.	Charumathi	4385349539	4593498734984959	RUPAY	05/40
3.	Karthikeyan	495734799	5893484474853988	MASTERCARD	03/50

UML DOMAIN OBJECT LAYER:

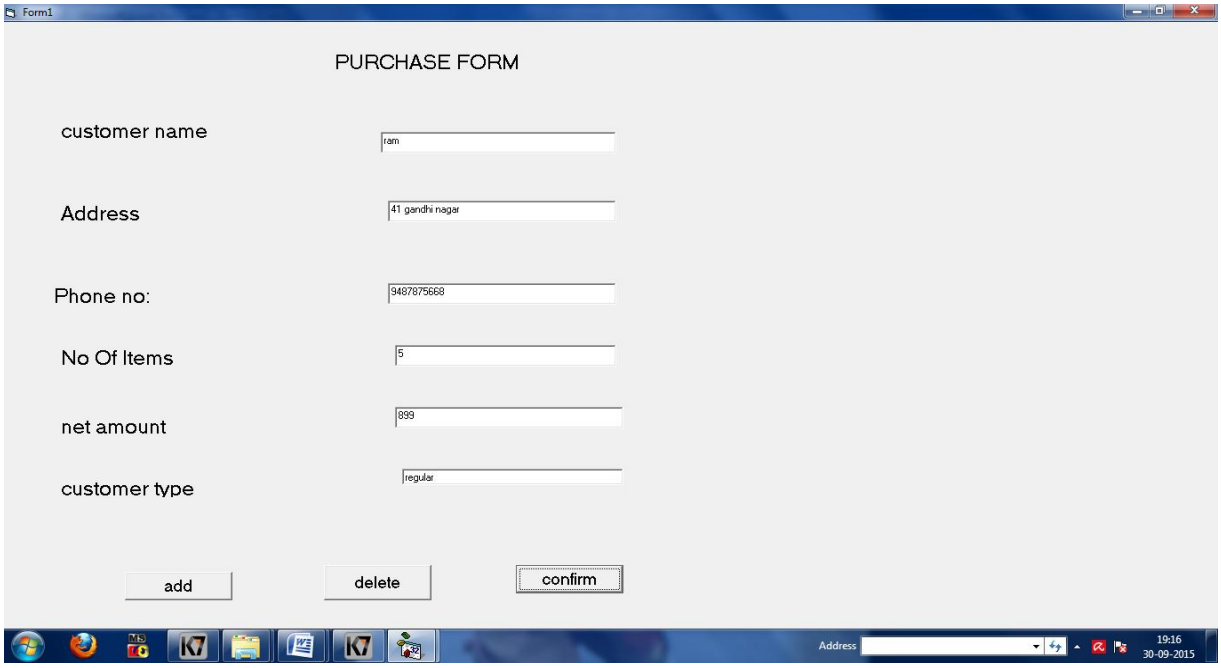
PURCHASE ITEM

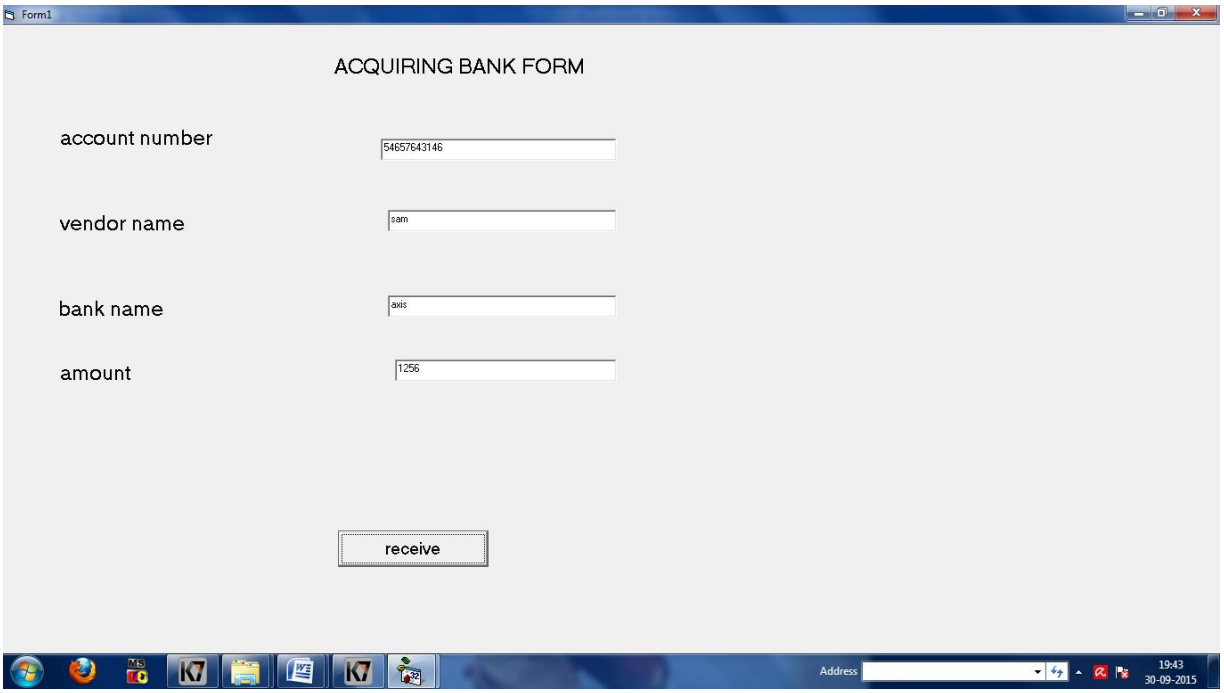
```
import java.util.Vector;
public class Purchase item
{
  /* {src_lang=Java}*/
  public String cusname;
  public String addr;
  public Integer phno;
  public Integer no_item;
  public Integer net_total;
  public String custype;
  public Vector  myBill_issue;
  public Vector  myBill_issue;
  public void add() {  }
  public void delete() {  }
  public void confirm() {  }
}
```

BILL ISSUE

```
import java.util.Vector;
public class Bill_issue
{
  /* {src_lang=Java}*/
  public Integer no_item;
  public Integer price;
  public Integer quantity;
  public String shop_name;
  public Vector  myPurchase item;
  public Vector  mycard verification;
  public void save() {  }
  public void add() {  }
  public void delete() {  }
  public void discount() {  }
  public void print() {  }
}
```

USER INTERFACE LAYER:





Result:

Thus the Credit Card Processing System has been done successfully by using Argo-UML.

EX.NO.:9	E-BOOK MANAGEMENT SYSTEM

AIM:

To design E-book Management System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying E-Book Management, software has been created by designing through ARGO-UML tool.

PROBLEM STATEMENT:

The software to be designed will control an eBook managing website which has eBooks in its database, which is available for registered users to download it by paying the necessary cost; the registered users can login and search for the required books and the users can update their details.

This can serve more than one customer at the same time, initially the customer enters his User ID and password and opens his account the User ID and password combination is checked in the database and only if correct, the account opens. After finishing the user can logout.

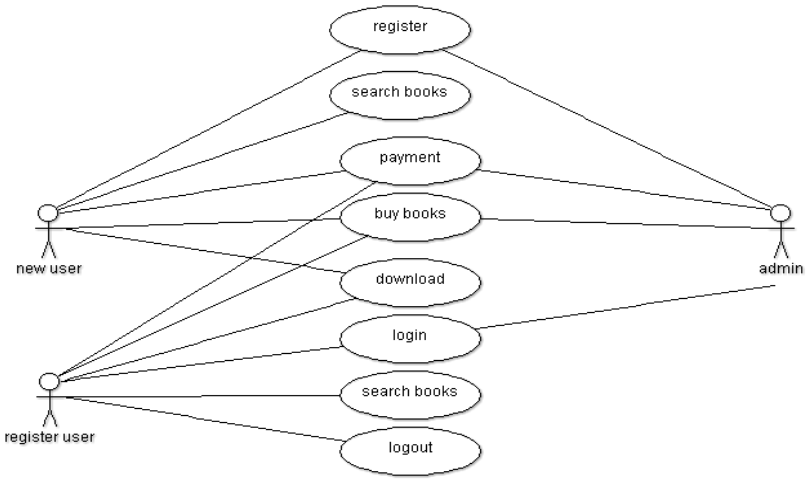
This software must provide the following:

1. A new customer must be able to register himself for using it using his email ID.
2. A registered user must be able to log-in into his account.
3. A user must be able to search for the book which he requires and must be able to get the book if required after paying the required cost.
4. He must be able to pay cost through credit card, debit card and internet banking.
5. He must be able to update his personal details.

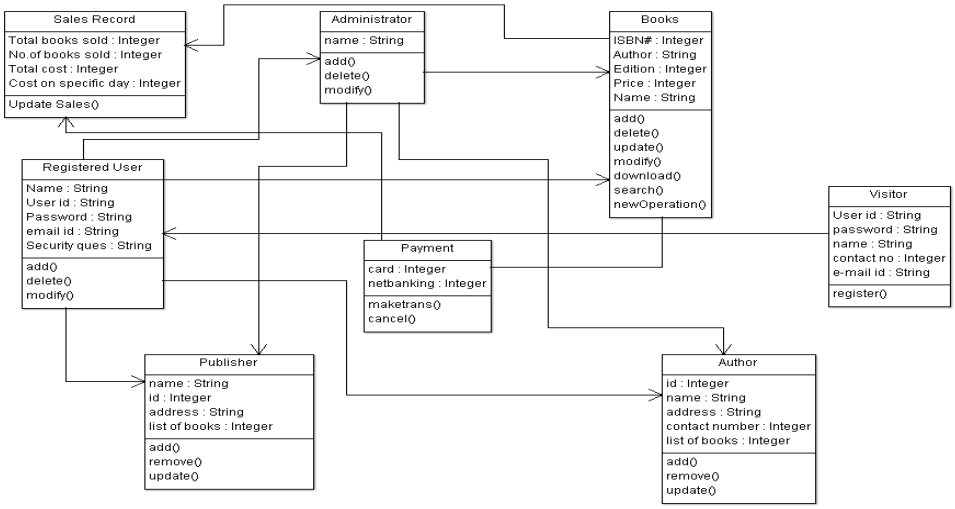
UML USECASE DIAGRAM:

User access his account by providing the correct userid and password.

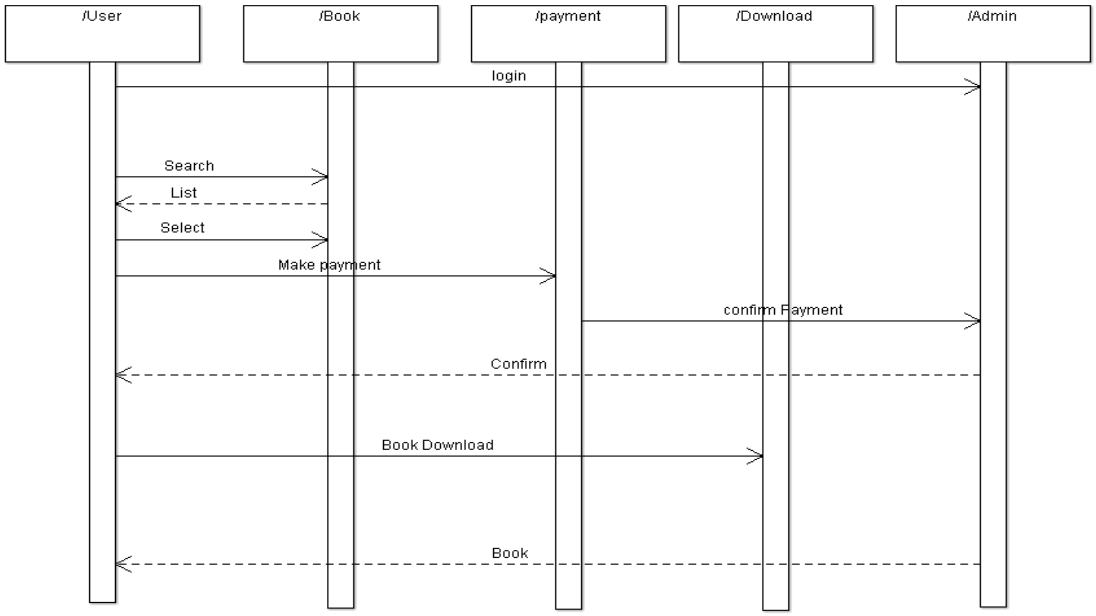
1. The user searches for the required book.
2. The user selects the book which he wants from the listed books.
3. The user confirms which redirects to payment page.
4. In payment page, the user enters the details of netbanking or his card.
5. The details are validated in the Bank database, if correct,
6. An OTP is generated and sent to the user's mobile.
7. If that is entered correctly, transaction is done and updated in the bank database.
8. The book is downloaded to the user and the database is updated.The user can either log out or he can continue to download more books.



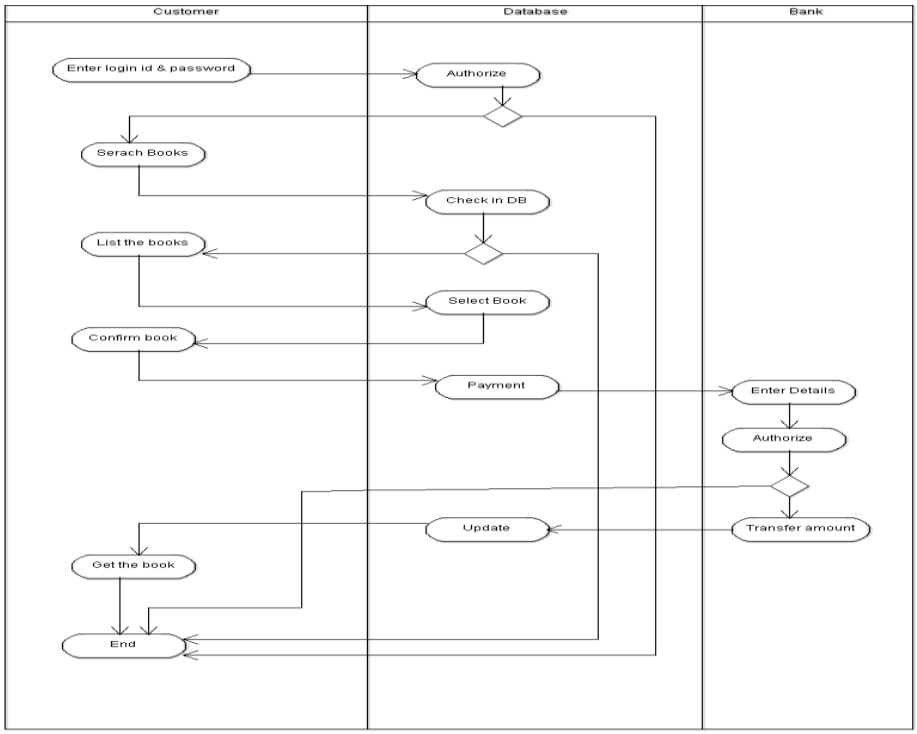
UML CLASS DIAGRAM:



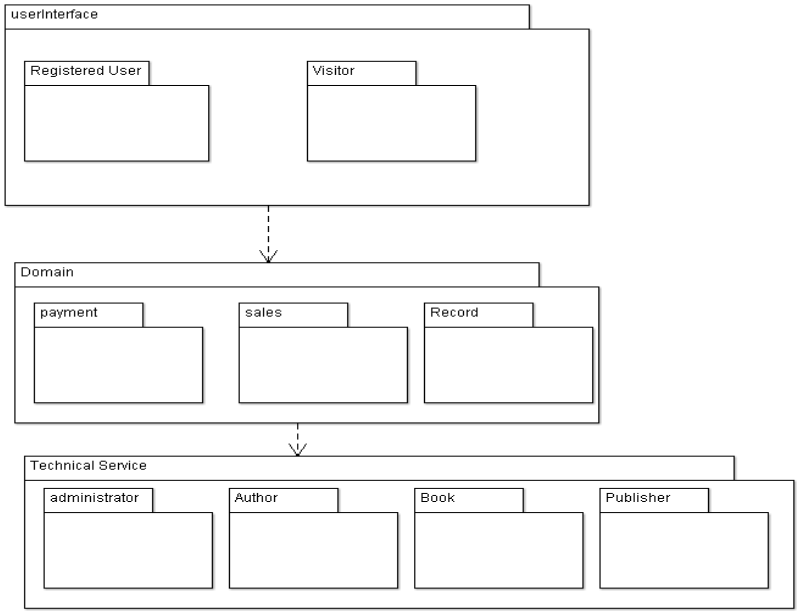
UML INTERACTION DIAGRAM:



UML ACTIVITY DIAGRAM:



UML PACKAGE DIAGRAM



UML TECHNICAL SERVICE LAYER:

S.No	Author Id	Name	Address	Contact Number	Total Books
1.	300001	Mories Mano	Punjab	9832872738	10
2.	300002	AK-Ray	Delhi	9374389392	15
3.	300003	Singaravelu	Chennai	9374239982	30

UML DOMAIN OBJECT LAYER:

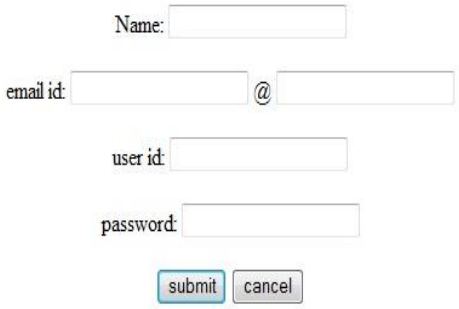
Administrator:

```
public class Administrator {
  /* {src_lang=Java}*/
public String name;
public void add() {  }
public void delete() {  }
public void modify() {  }
}
```

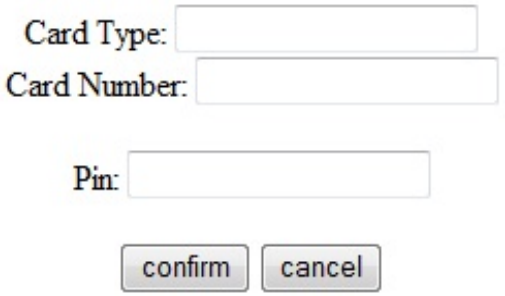
Author:

```
importjava.util.Vector;

public class Author {
  /* {src_lang=Java}*/
public Integer id;
public String name;
public String address;
public Integer contact number;
public Integer list of books;
public Vector  myAdministrator;
public void add() {
  }
public void remove() {
  }
public void update() {
  }
}
```

USER INTERFACE LAYER:



Registration Form



Payment

Result:

Thus the E-Book Management System has been done successfully by using Argo-UML.

EX:NO:10	RECRUITMENT SYSTEM

AIM:

To design Recruitment System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying, software has been created by designing through ARGO-UML tool.

The Online Recruitment System is an online website in which applicant can register themselves and then attend the exam. Examination will be conducted at some venue. The details of the examination, venue & Date of the examination will be made available to them through the website. Based on the outcome of the exam the applicant will be short listed and the best applicant is selected for the job.

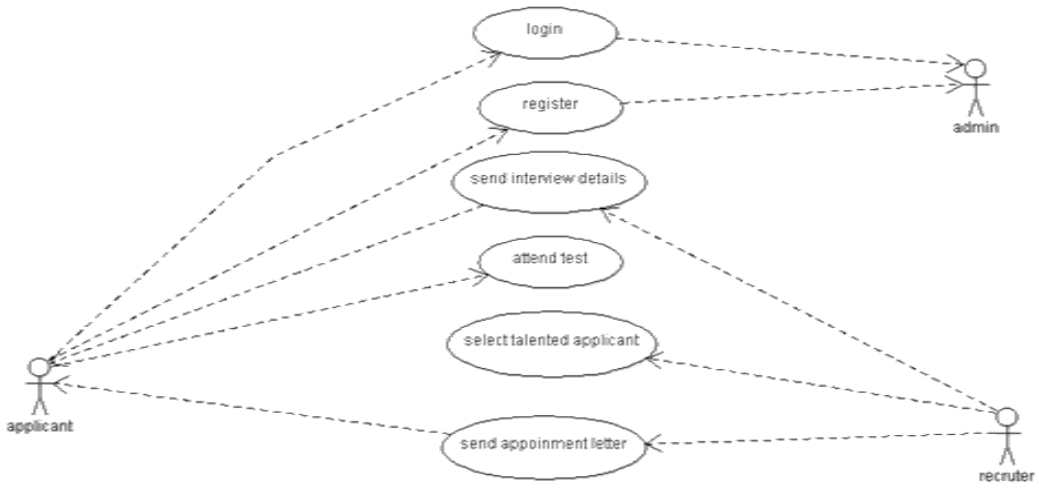
PROBLEM STATEMENT:

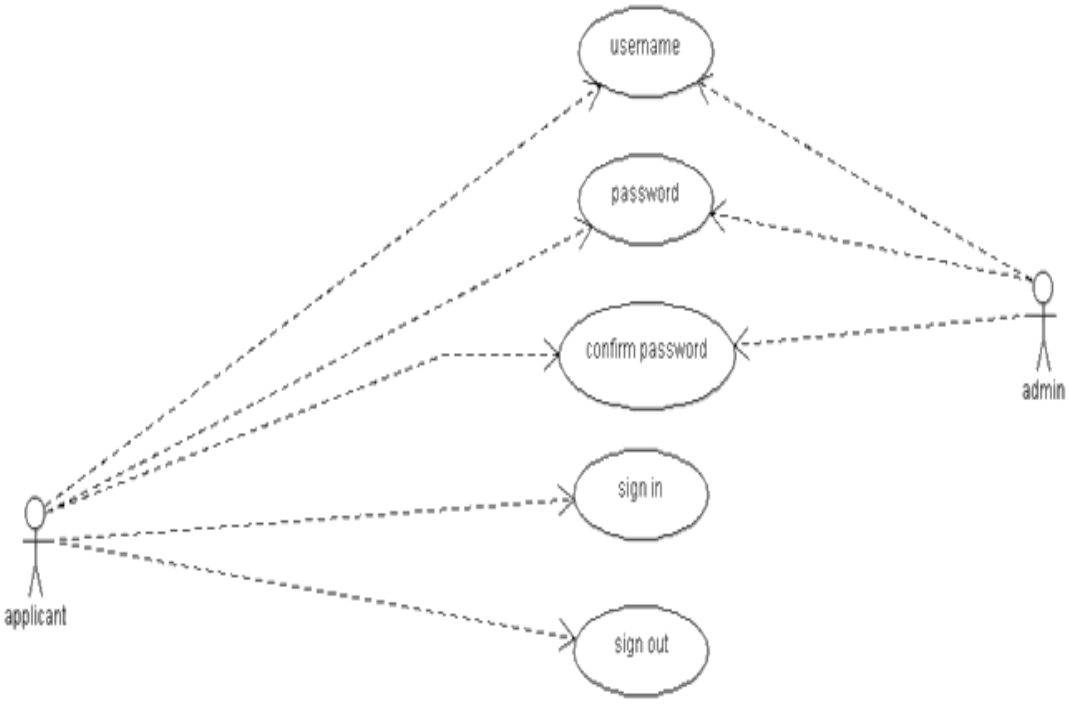
The process of applicant is login to the recruitment system and register for the job through online. The resume is processed by the company and the required applicant is called for the test. On the basis of the test marks, they are called for next level of interview. Finally the best applicant is selected for the job.

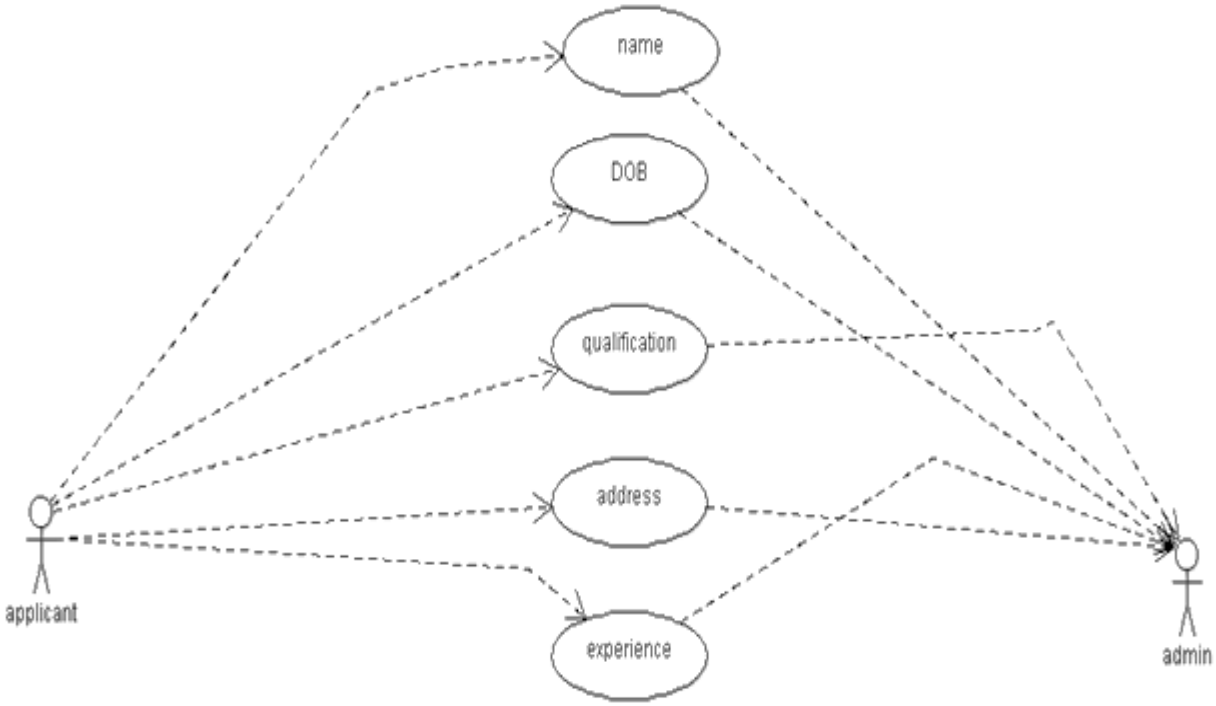
These processes of online recruitment system are described sequentially through following steps:

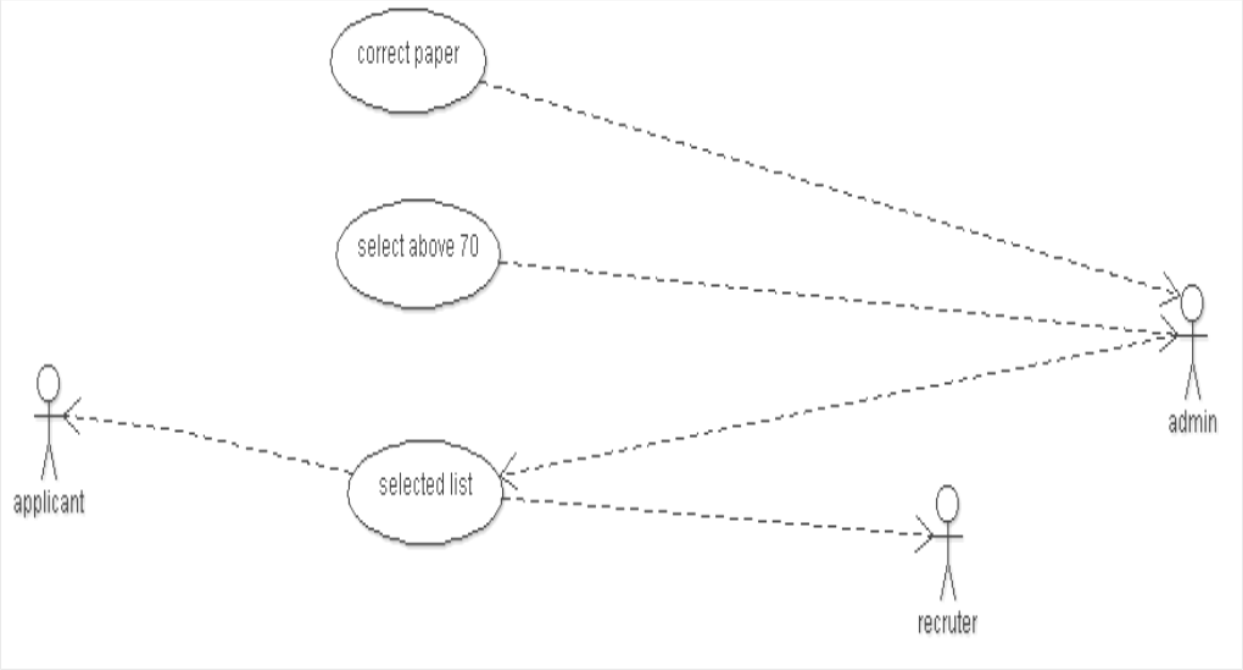
- The applicant login to the online recruitment system.
- They register to the company for the job.
- They appear for examination.
- Based on the outcome of the exam, the best applicant is selected.
- The recruiter informs the applicant about their selection

UML USECASE DIAGRAM:




username
password
confirm password
sign in
sign out
applicant
admin


name
DOB
qualification
address
experience
applicant
admin



[1] Candidate should sign up by giving username and password.

[2] Admin sends the reply to the candidate with register numbers.

[3] Enter into the software to attend the aptitude test. It will show the home page.

[4] Candidate should attend the test which is conducted on online

[5] Admin should correct the answer sheet which has been sent by candidate.

[6] Admin should select the candidate and update the admin tools and company details to respective candidates.

[7] Organization should send the conformation letter to the selected candidate.

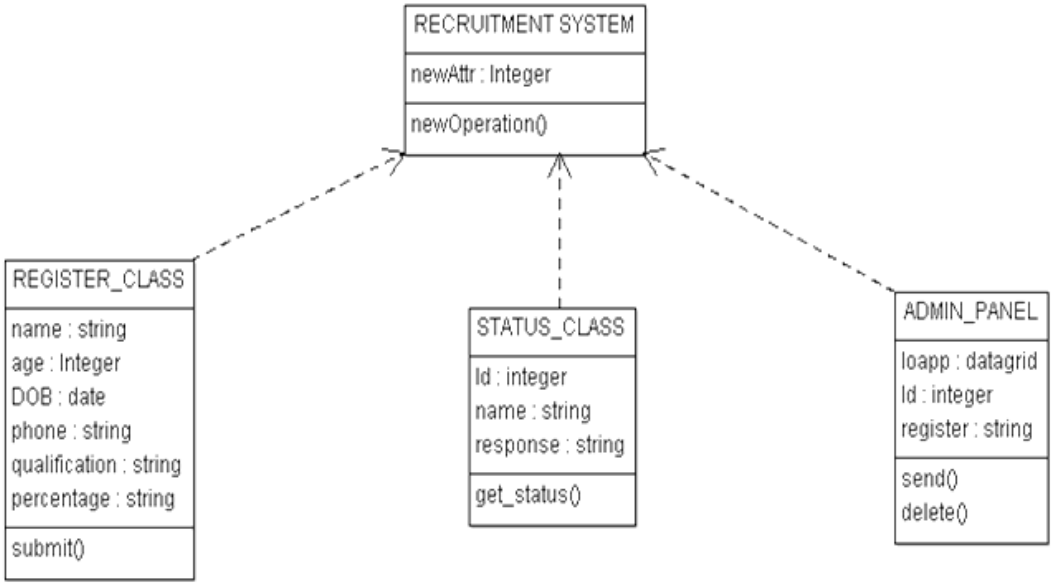
UML CLASS DIAGRAM:

This class diagram has three classes' applicant, recruiter and database.

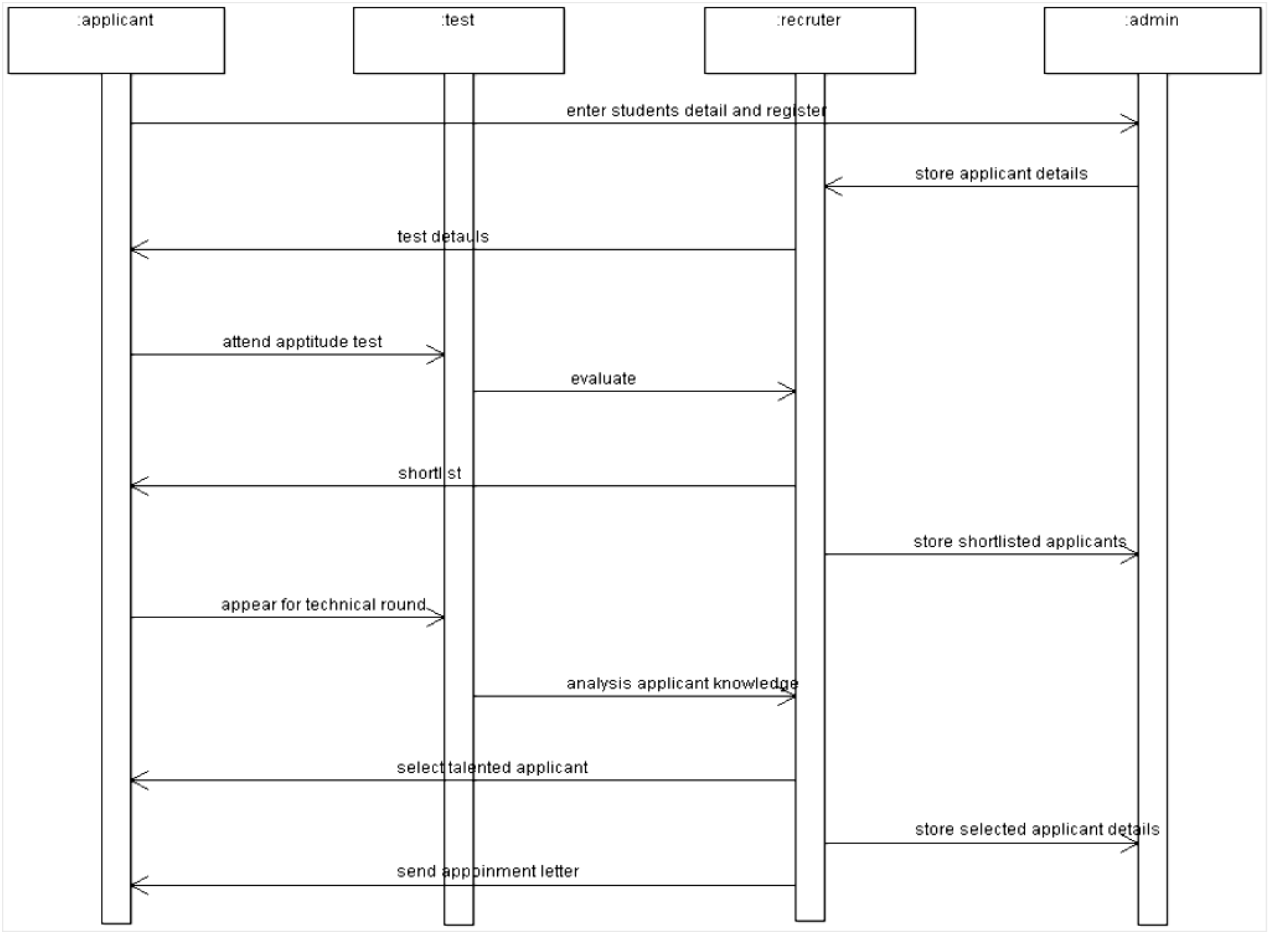
• **Applicant** – is the class name. Its attributes are username, password, name, phone no and address. The operations performed in the applicant class are login, register and giving applicant details.

• **Recruiter** – is the class name. Its attributes are name, designation, phone no, marks in apps and marks in technical. The operations performed are selecting applicants based on apps and technical.

• **Database** – is the class name. The operations performed are storing applicant details, verifying login and storing selected applicant details.



UML INTERACTION DIAGRAM:



REGISTER

This sequence diagram describes the sequence of steps to show:

• The applicant login in to the recruitment system and register for job.

• The verification done in the database and recruiter

• The interview details are sending to the applicant by recruiter.

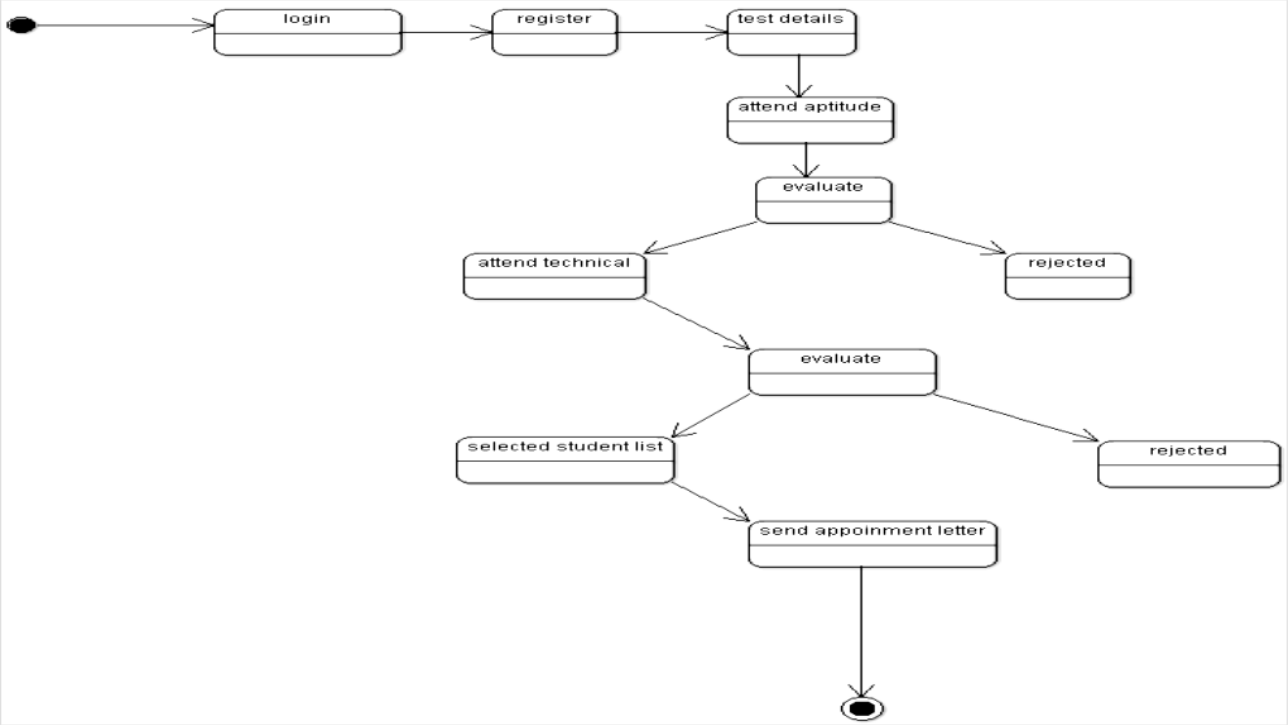
SELECTING APPLICANT

This sequence diagram shows steps to show:

• The applicant attend aptitude test and they are short listed based on evaluation

• The applicant appear for technical round

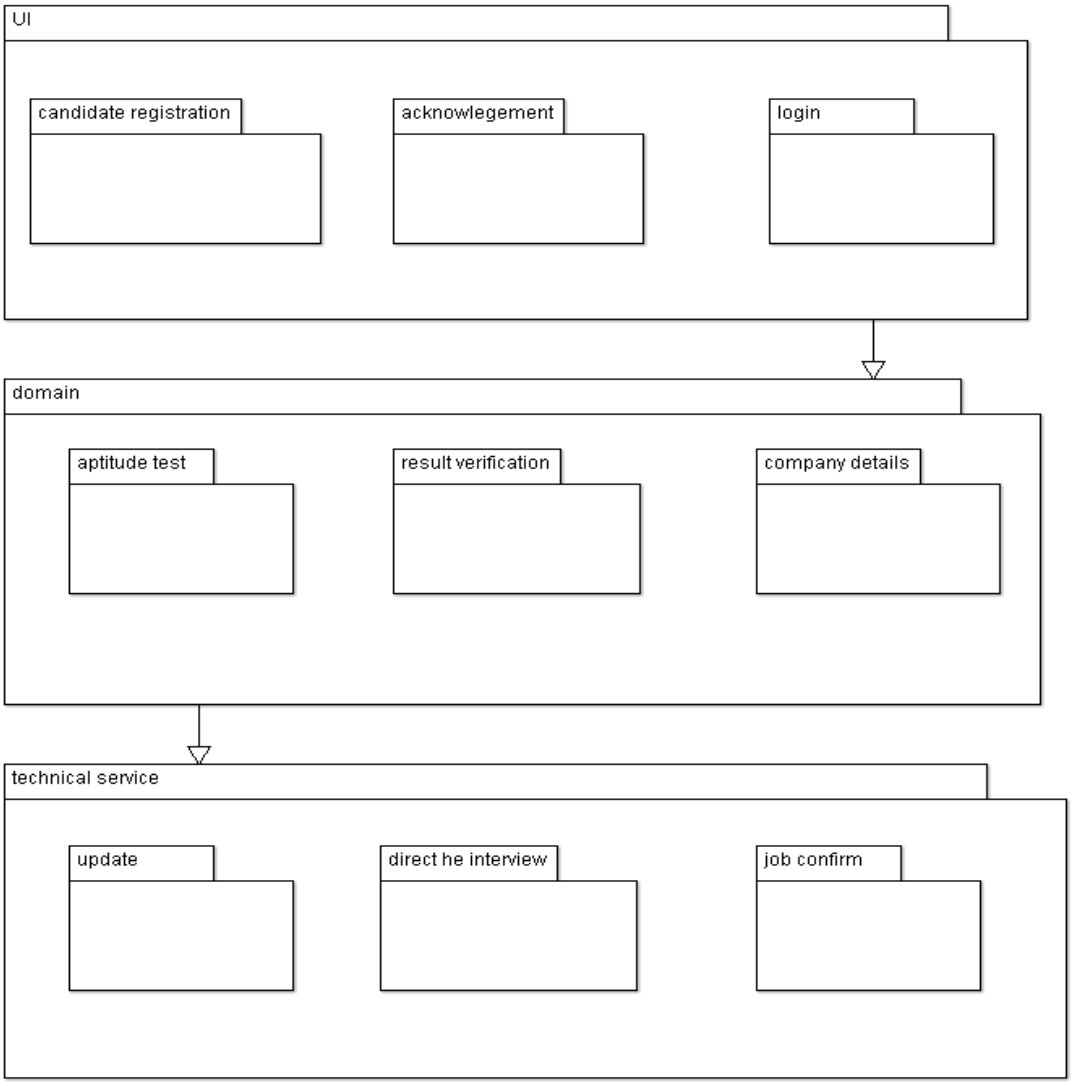
<u>UML STATECHART DIAGRAM</u>:



This state diagram describes the behavior of the system.

- First state is login where the applicant login to the recruitment system.
- The next state is register where the applicant register for job.
- Then verify the applicant details and sent interview details.
- The applicant appears for test.
- Update database with details of selected applicant.

UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER:

S.No.	Name	Tech_Marks
1.	Maha	90
2.	Sowmi	92
3.	Sharmi	88
4.	Divya	70
5.	Tejesh	70
6.	Barath	70
7.	Harish	70

UML DOMAIN OBJECT LAYER

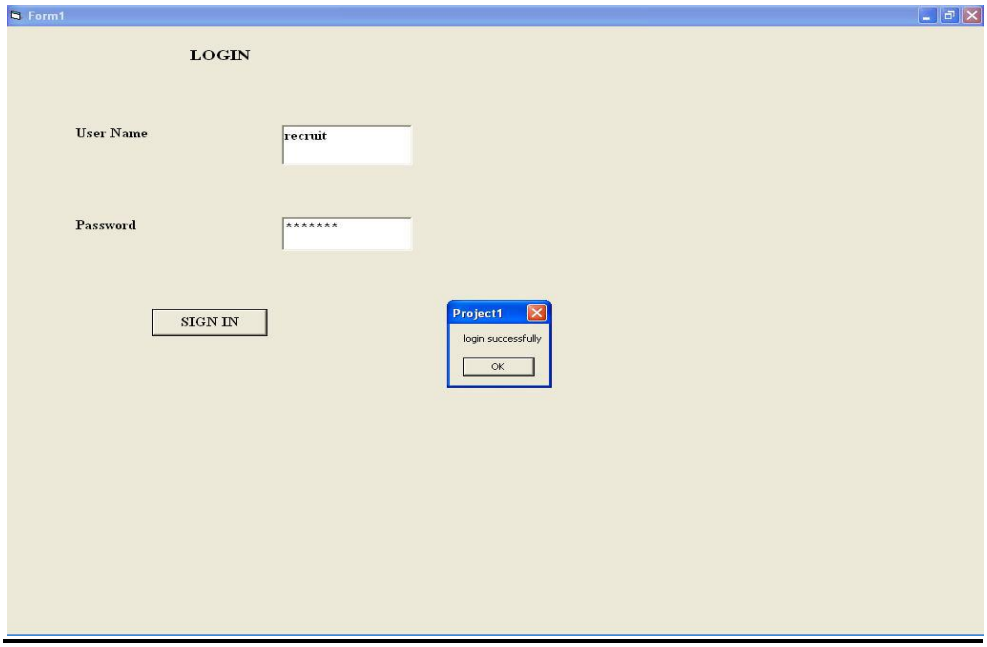
Register Class

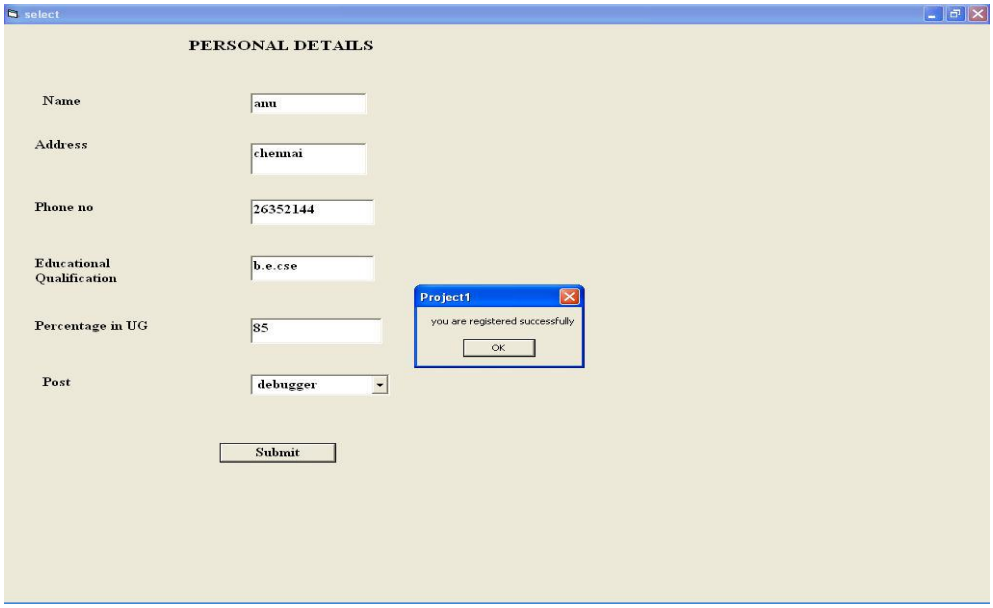
```
import java.util.Vector;

public class Register class {

  public String name;

  public Int age;

  public varchar DOB;

  public int Phone;

  public string Qualification;

  public varchar Percentage;

    public Vector  myRecruitment System;

  public void submit() {
  }

}
```

ADMIN PANEL

```
import java.util.Vector;

public class ADMIN PANEL {

  public INT ID;

  public String register;

    public Vector  myRecruitment System;

  public void send() {
  }

  public void delete() {
  }

}
```

USER INTERFACE LAYER:





Result:

Thus the Recruitment System has been done successfully by using Argo-UML.

EX.NO.:11	FOREIGN TRADIING SYSTEM

AIM:

To design Foreign Trading System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying, software has been created by designing through ARGO-UML tool.

The main activity of Foreign Trading System (FTS) is import and export procedure.

The online FTS is almost entirely a "spot" market. It means that trading is made immediately.

It also involves documentation, procedures, rules and regulations importing and exporting countries. Foreign trading spot transaction is made within two working days.

PROBLEM STATEMENT:

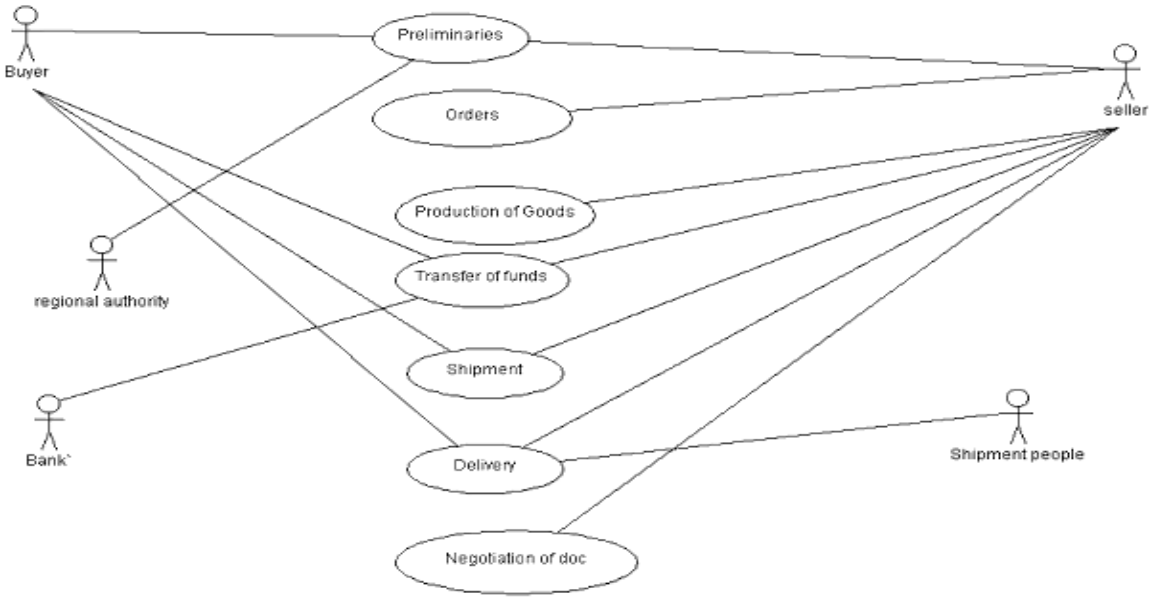
The steps involved in FTS are:

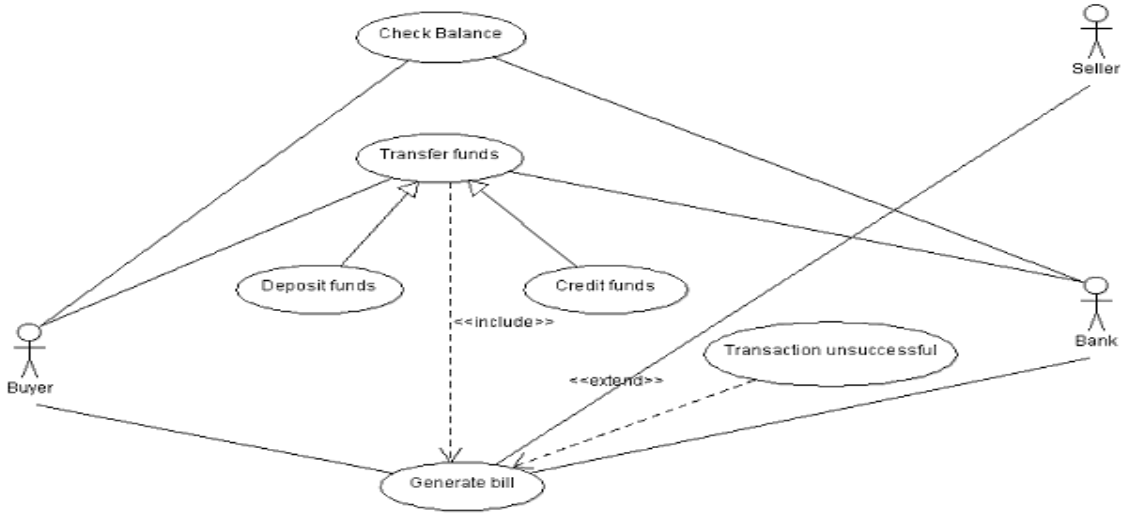
- The process of FTS begins with getting the username and password from the trader by the administrator.
- The administrator gives the authority to the trader to check the details of commodities.
- If the trader is satisfied with the commodities places his order to the administrator.
- The administrator checks for the availability and update the details in the database.
- The traders pass the amount to the administrator.
- The administrator will provide the bill after receiving the amount.
- The shipment people deliver the commodities to the trader.
- The system provides an interface where the buyer can fill in their personnel details and they can also scan and upload necessary documents. By using this shipment of goods can be done easily. Also application can be processed in speedy manner.

UML USECASE DIAGRAM:

The actors in this use case diagram are

- Trader
- Buyer
- Regional authorities
- Bank
- Shipment people

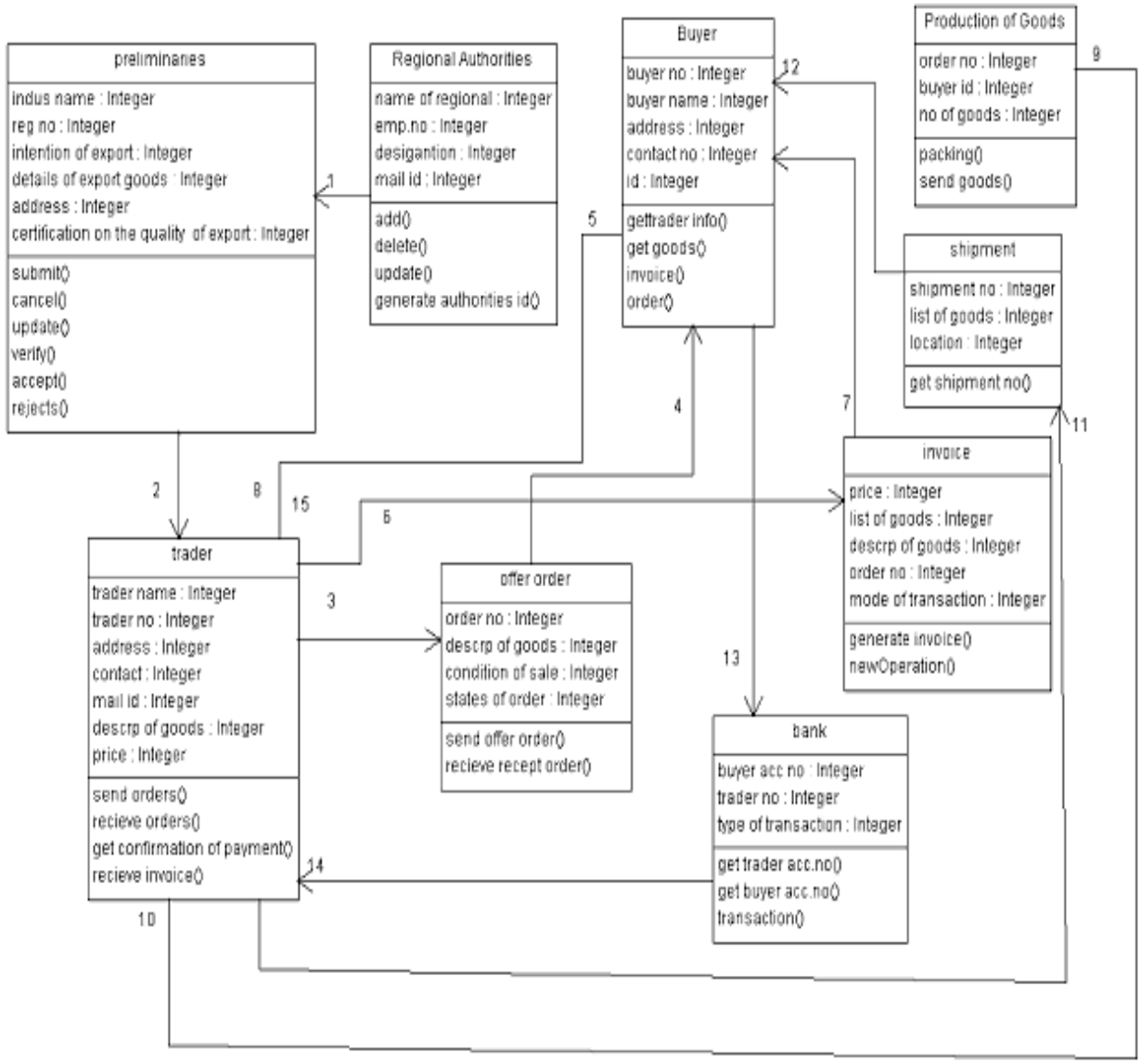




UML CLASS DIAGRAM:

This class diagram consists of nine class.

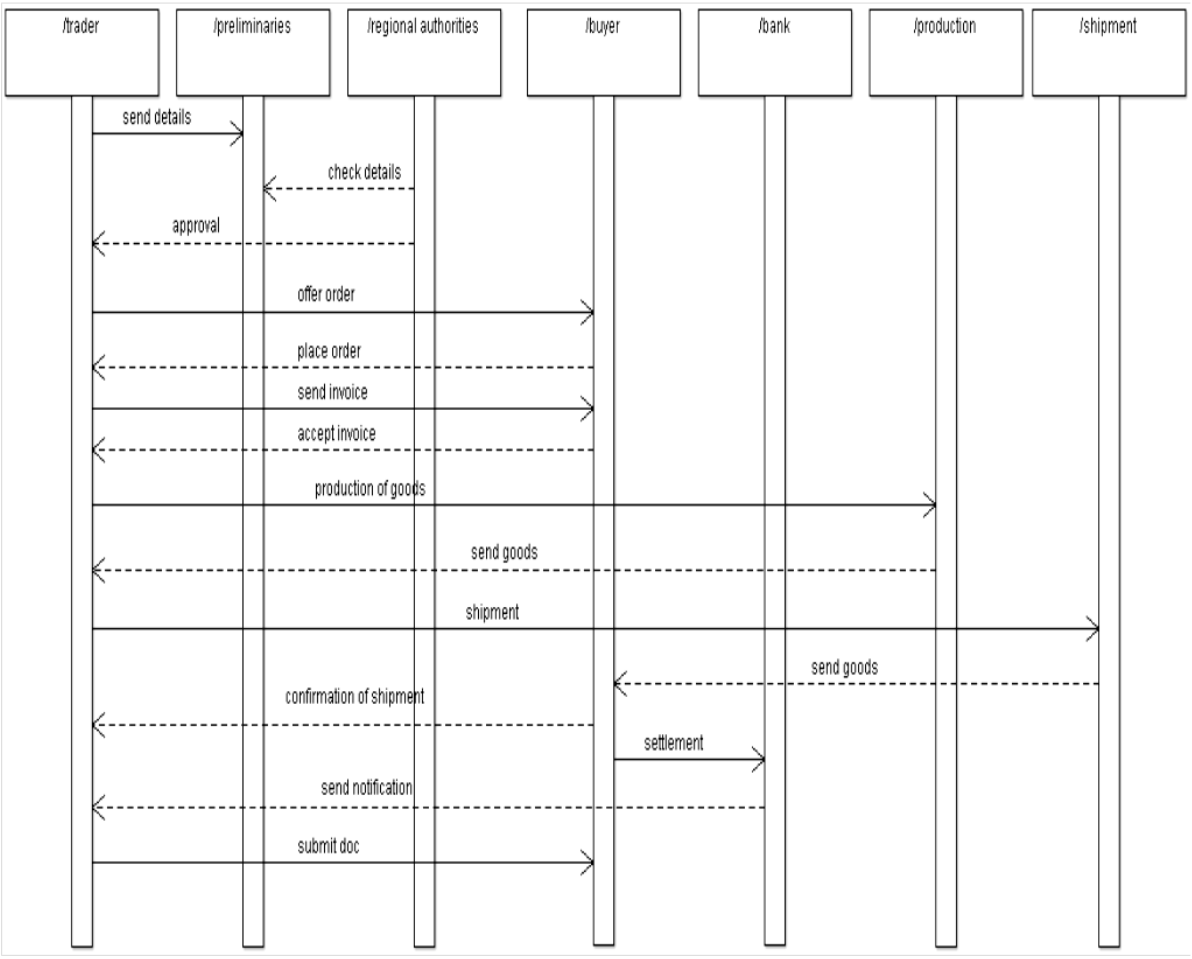
- Preliminaries
- Regional authorities
- Buyer
- Trader
- Offer order
- Invoice
- Production of goods
- Bank
- Shipment



UML INTERACTION DIAGRAM:

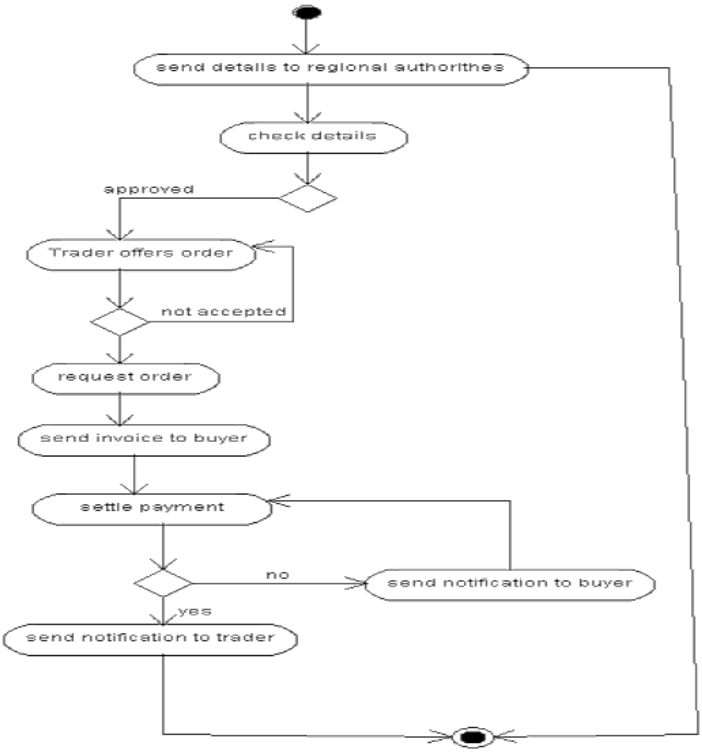
The sequence diagram represents:

- ✓ The trader sends the details to the regional authorities and the regional authority gives approval .
- ✓ The trader offer order to the buyer and the buyer places his order.
- ✓ Then the trader generate the invoice and the buyer accepts it .
- ✓ Goods are produced and its send to the buyer through the shipment people .
- ✓ The payment is done via bank.
- ✓ The notification is send to the trader by bank about the update of payment.
- ✓ At the end the trader submits the document to the buyer.

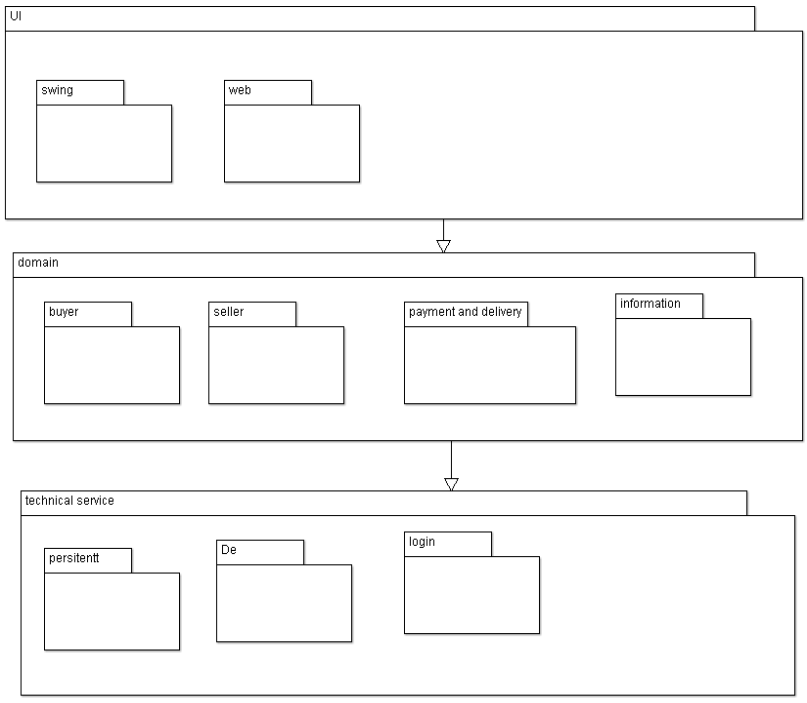


UML ACTIVITY DIAGRAM:

- Perform preliminaries activities i.e., getting IEC number from regional licensing authorities in the first action
- Submit a proposal order to the buyer in the send action
- After obtaining a confirmed order should produce the goods exactly as specifies in the invoice in third and fourth action.
- If the exporting house does not have production facilities, it has to procure the products from others.
- Transport the goods to the buyer in the sixth action.
- The trader submits the relevant documents to his buyer for getting the payment for the goods exported in seventh and eight action.
- The decision state is the state where the trader decides sends the notification to the buyer if the unexpected failure in the payment occurs.



UML PACKAGE DIAGRAM



UML TECHNICAL SERVICE LAYER:

Trade Name	Trade No	Contact	Mail	Price
Siva Agencies	5390	2823992	Siva1@gmail.com	10,00,000
BMPSC	3487	2348930	bmpsc@gmail.com	2,00,000
Vicky Agencies	4589	2394030	Vky02@gmail.com	1,00,000

Buyer Name	Buyer No	Contact	Mail	Price
Priyanka	3348	2938490	priyank@gmail.com	10,00,000
Preetha	2093	2049034	prite@gmail.com	2,00,000
Lubna	2398	2349394	lubu@gmail.com	1,00,000

UML DOMAIN OBJECT LAYER:

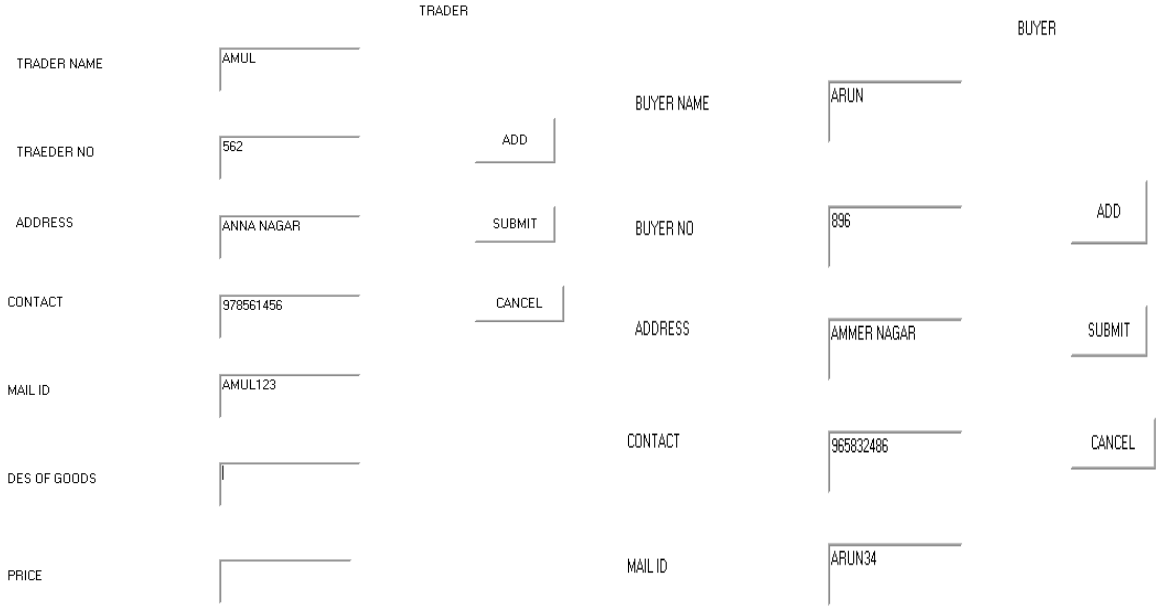
Code for preliminaries:

```
import java.util.Vector;
public class preliminaries
 {
  /* {src_lang=Java}*/
     private Integer indus name;
     public Integer reg no;
     public Integer intention of export;
     public Integer details of export goods;
     public Integer address;
     public Integer certification on the quality
of export;
     public Vector  2;
public void submit(){ }
public void cancel() { }
  public void update() {  }
public void verify(){ }
     public void accept() { }
 public void rejects(){ }
}
```

Coding for regional authorities:

```
public class Regional Authorities
 {
      /* {src_lang=Java}*/
      public Integer name of regional;
      public Integer emp.no;
      public Integer desigantion;
      public Integer mail id;
      public preliminaries mypreliminaries;
      public void add()
     { }
      public void delete()
     { }
     public void update()
     { }
    public void generate authorities id()
     {  }
}
```

USER INTERFACE LAYER:



Result:

Thus the Foreign Trading System has been done successfully by using Argo-UML.

EX.NO.:12	CONFERENCE MANAGEMENT SYSTEM

AIM:

To design Conference Management System by using Argo-UML tool. .

PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying, software has been created by designing through ARGO-UML tool.

The Conference Management System is an online website in which candidate can submit the paper and register themselves and then attend the conference. The paper will be reviewed. The details of the conference, date and time will be made available to them through the website. After getting the confirmation details the candidate should submit the revised and camera ready paper. Then the registration process will be done.

PROBLEM STATEMENT:

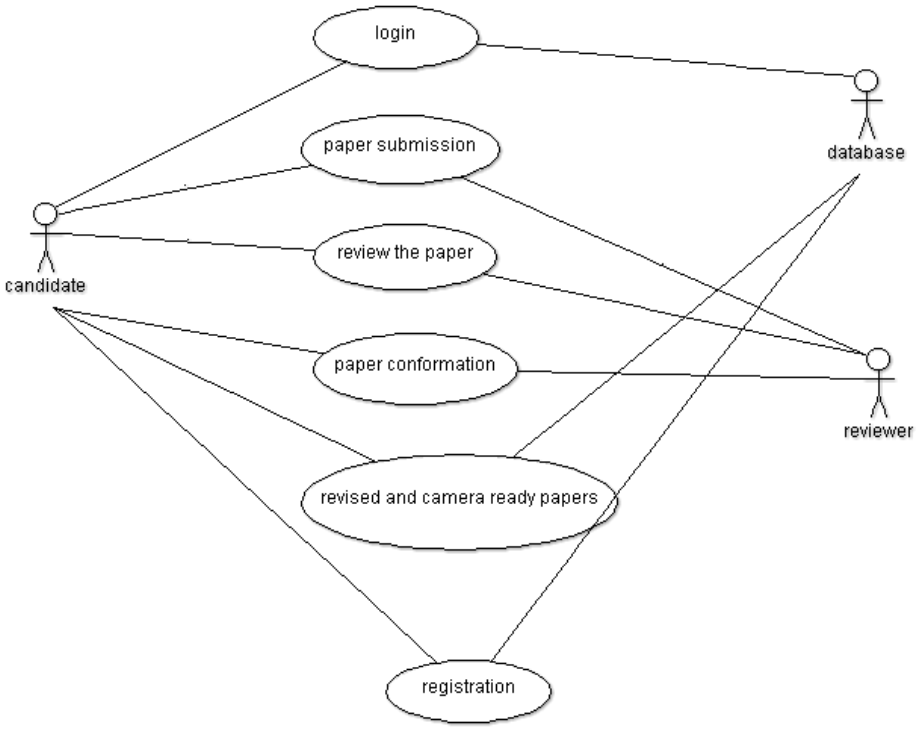
The process of the candidates is to login the conference system and submit the paper through online. Then the reviewer reviews the paper and sends the acknowledgement to the candidate either paper selected or rejected. This process of on conference management system are described sequentially through following steps,

- The candidate login to the conference management system.
- The paper title is submitted.
- The paper is been reviewed by the reviewer.
- The reviewer sends acknowledgement to the candidate.
- Based on the selection, the best candidate is selected.
- Finally the candidate registers all details.

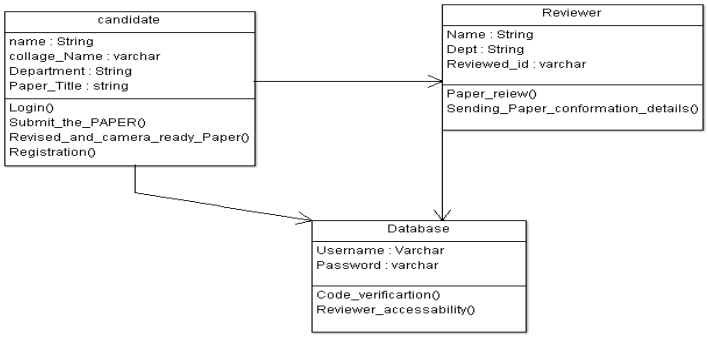
UML USECASE DIAGRAM:

- Candidate
- Reviewer
- Databases

- Login
- Paper submission
- Review the paper
- Paper confirmation details
- Revised and camera ready paper
- Registration



UML CLASS DIAGRAM:



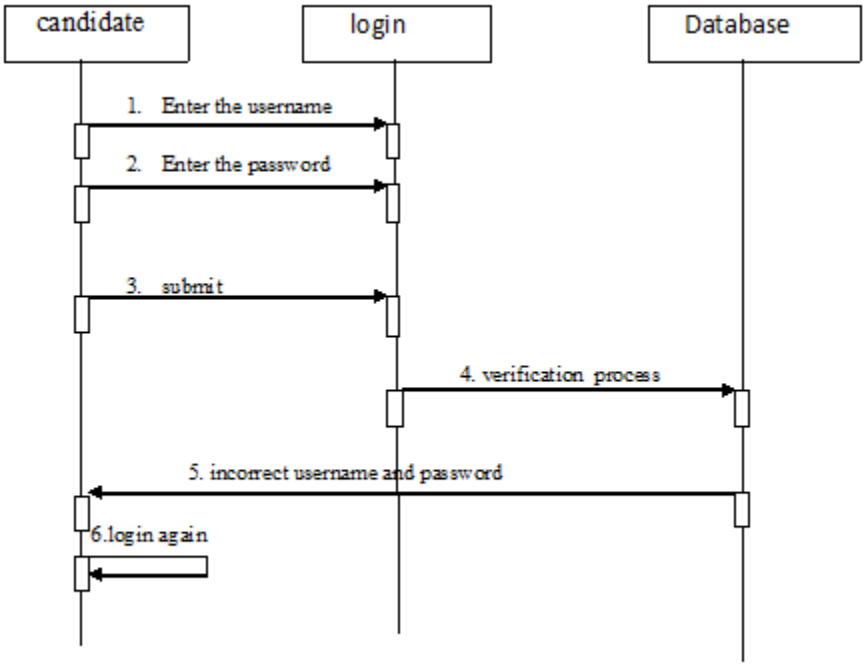
This class diagram has three classes' candidate, reviewer and database.

• **Candidate** – Its attributes are name, college name, department, paper title. The operations performed in the candidate class are login, submit the paper, submit revised and camera ready paper and registration.

• **Reviewer** – Its attributes are name, department, reviewer ID and the operations performed are review the paper and send the paper confirmation details.

• **Database** –The operations performed are storing candidate details and verifying login.

UML INTERACTION DIAGRAM:



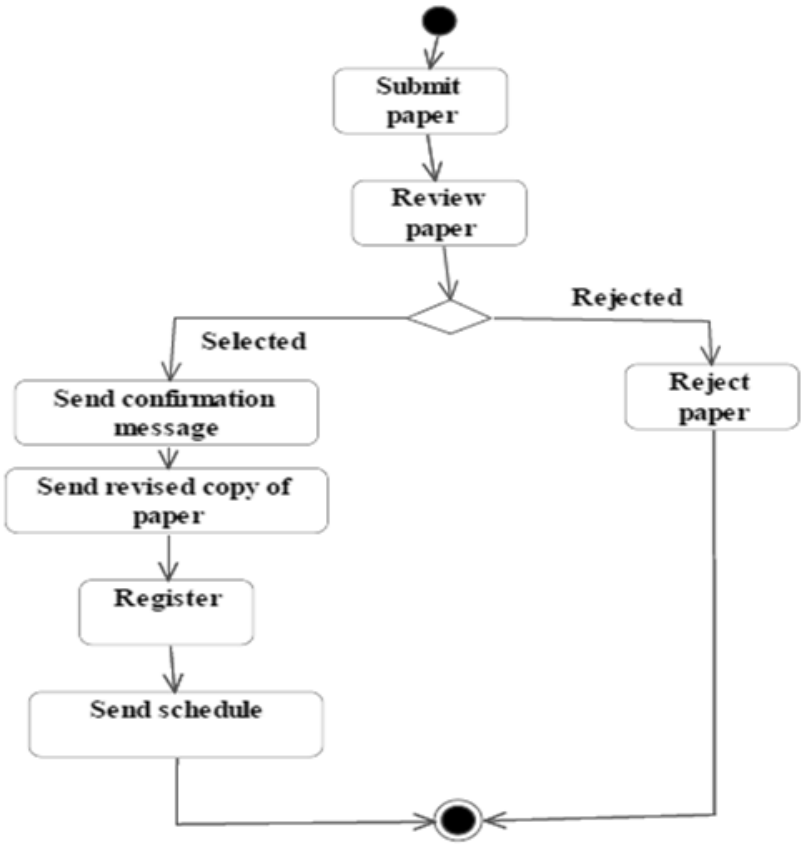
UML ACTIVITY DIAGRAM:

• First the candidate login to the database.

• Then the candidate should submit the paper.

• If it is selected the acknowledgement will send to the candidate.

• After submitting revised paper the registration process will be done.

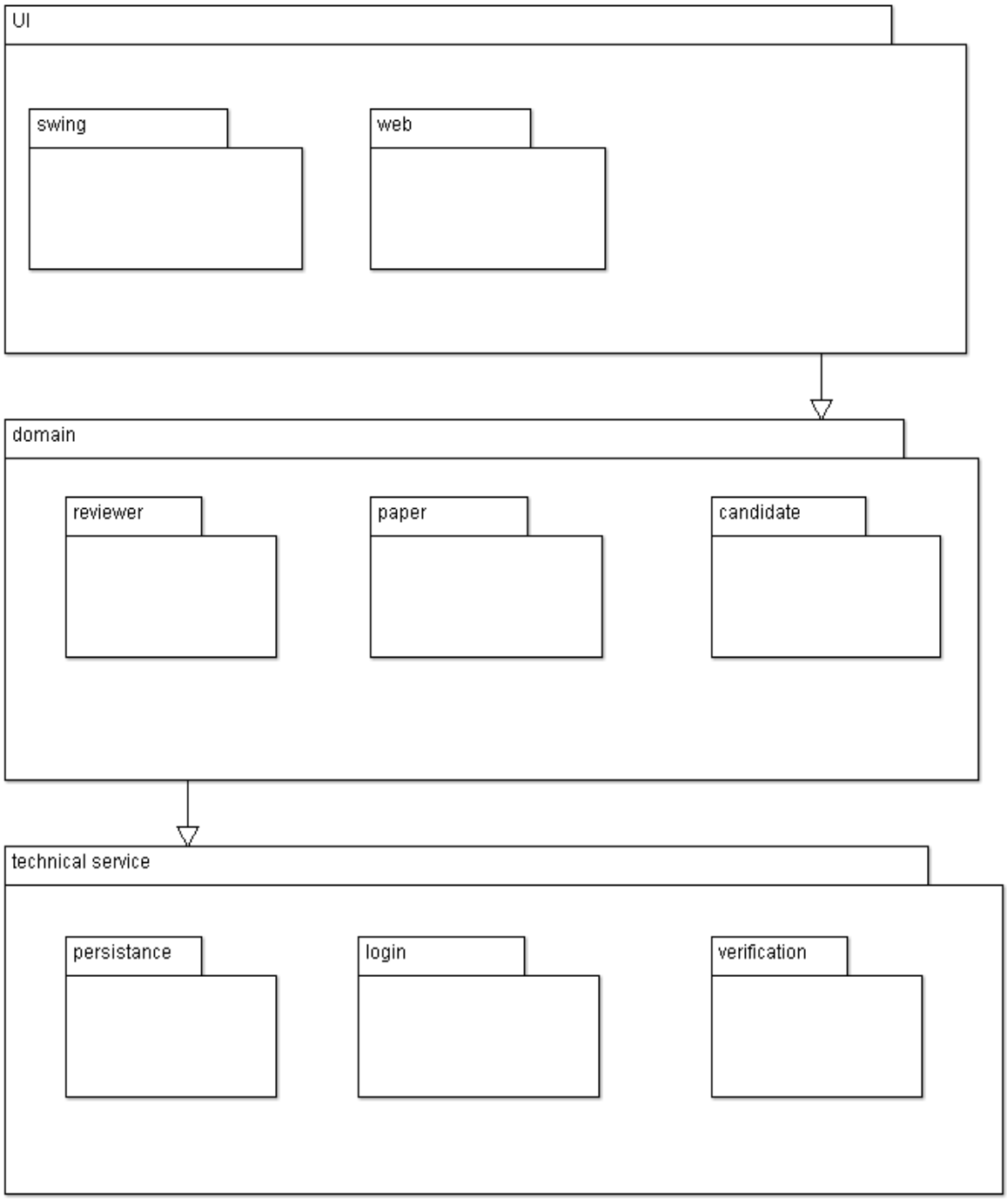


UML TECHNICAL SERVICE LAYER

S.No.	Name	Name of the Paper	Contact	E-mail
1.	Preethi	Cloud Computing	9324389383	preethi@gmail.com
2.	Nathiya	Networking	9348739923	nadhisri@gmail.com
3.	Pavithra	Database	7354863948	pavi@gmail.com

S.No.	Name of the Paper	Date	Time	Place
1.	Cloud Computing	02-09-2017	9.00 A.M	Chennai
2.	Networking	15-09-2017	10.00 A.M	Selam
3.	Database	03-10-2017	9.30 A.M	Bangalore

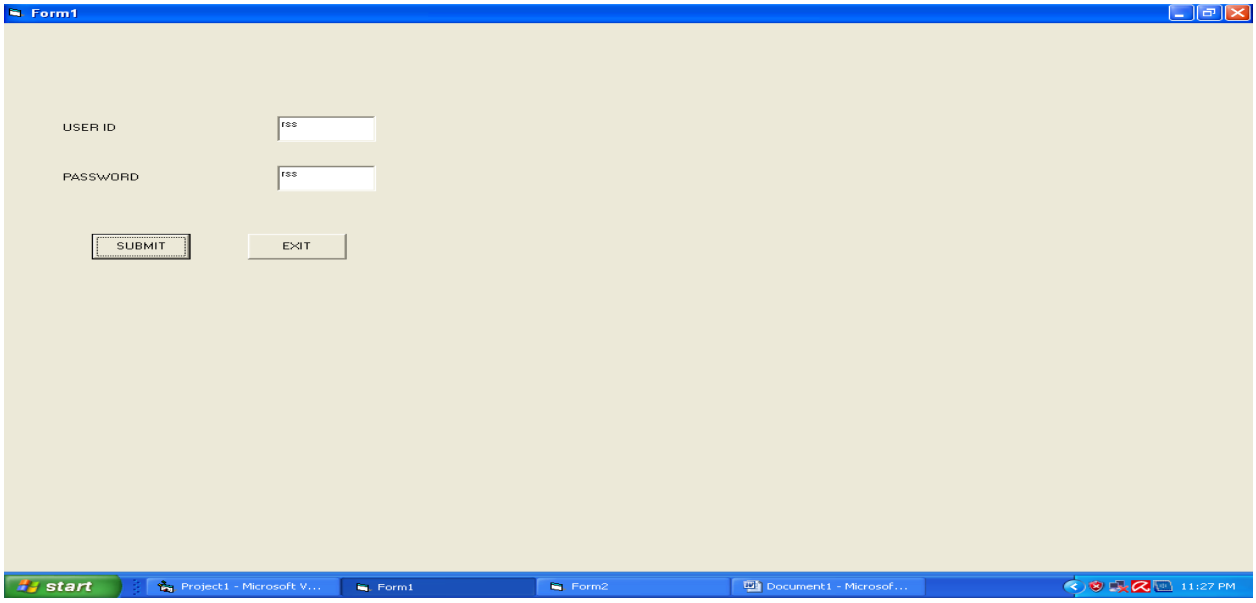
UML PACKAGE DIAGRAM:

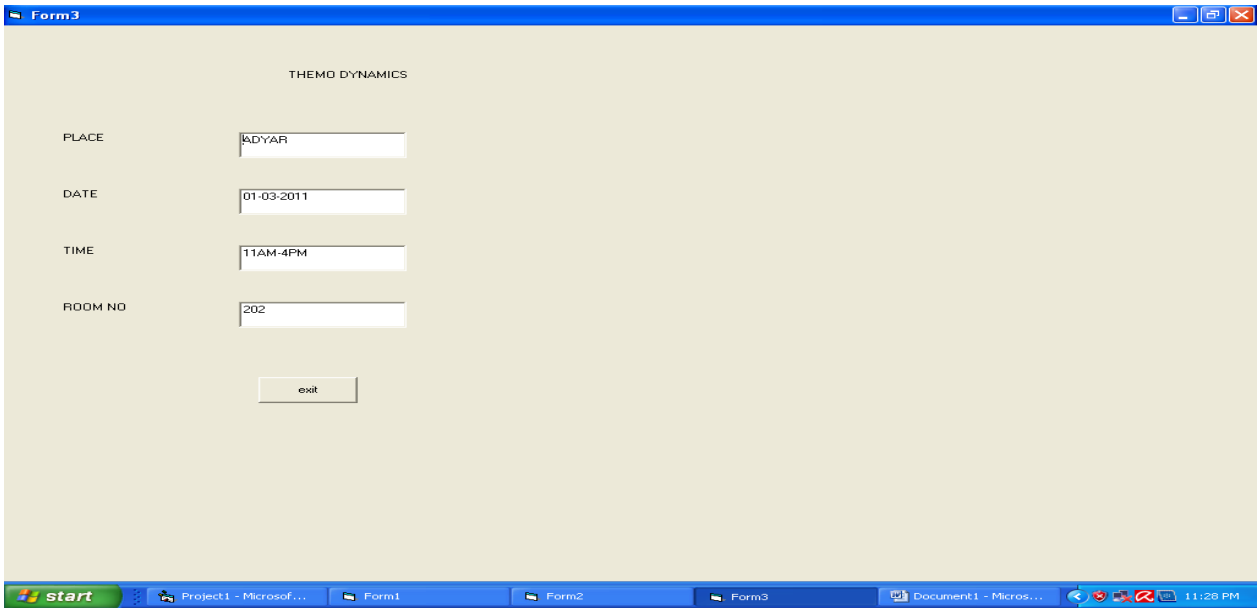


UML DOMAIN OBJECT LAYER

```
import java.util.Vector;
public class application
{
  public char name;
  private char fathername;
  public int Dateofbirth;
  private varchar permanentaddress;
  private varchar Temporary_address;
  public varchar email;
  public int Phonenumber;
  public varchar panNo;
  public varchar ApplicationNO;
  public varchar Username;
  public varchar password;
    public Vector  myDatabase;
  public void login()
  {
  }
  public void submitdetails()
  {
  }
  public void checkingstatus()
  {
  }
}

import java.util.Vector;

public class Database
{
    public char name;
    public Vector  myapplication;
    public Vector
mypassportAdministration;
    public Vector
myregionalAdminstrator;
    public Vector  mypolice;
  public void store()
  {
  }
}
```

USER INTERFACE LAYER:





Result:

Thus the Conference Management System has been done successfully by using Argo-UML.

EX.NO.:13	**BPO MANAGEMENT SYSTEM**

AIM:

To design BPO Management System by using Argo-UML tool.

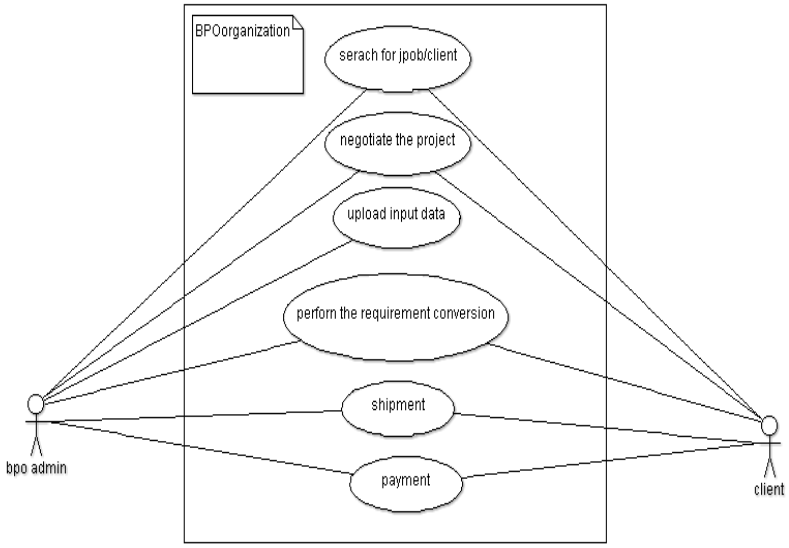
PROBLEM ANALYSIS AND PROJECT PLAN:

To simplify the process of applying, software has been created by designing through ARGO-UML tool. Generally outsourcing can be defined as an organization entering into a contract with another organization to operate and managed one or more of its business process. There are many problems faced by the BPO one among them is meeting their targets and leaving the concern very often and switch to another company. In this project we deal with the inbound system of the BPO. In inbound system the agent calls the customer from his database to sell his product.

PROBLEM STATEMENT:

With the reduction in communication costs and improved bandwidths and associated infrastructure, BPO as a segment is witnessing a massive growth. One of the key challenges that BPO companies that provide data entry/data validation services is an efficient and effective way of getting the source documents from different customers and accurately route the same to different operators for processing.

UML USECASE DIAGRAM:



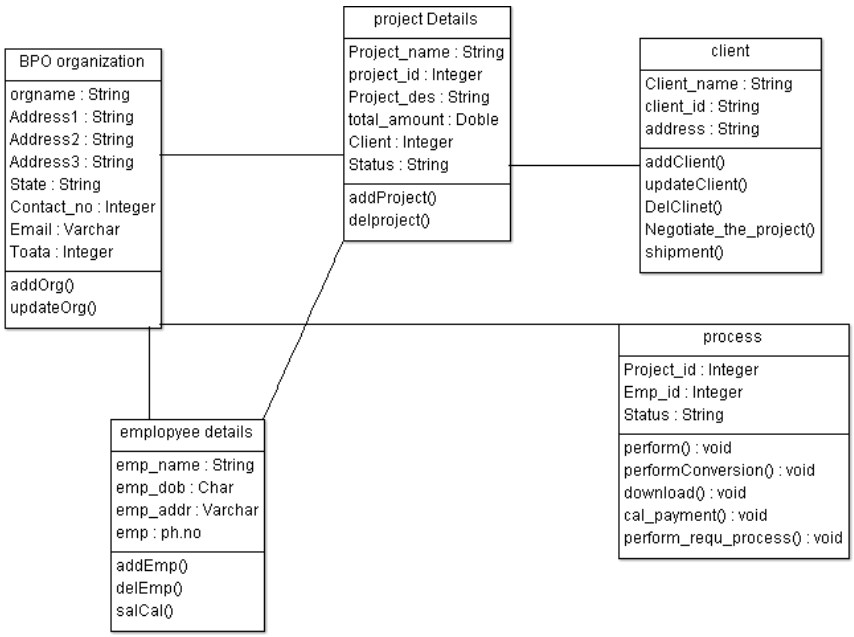
ACTORS

- BPO organization:
- Client

USE-CASE

- Search for client/job
- Negotiate the project
- Upload input data
- Perform required conversion
- Quality Check

UML CLASS DIAGRAM:

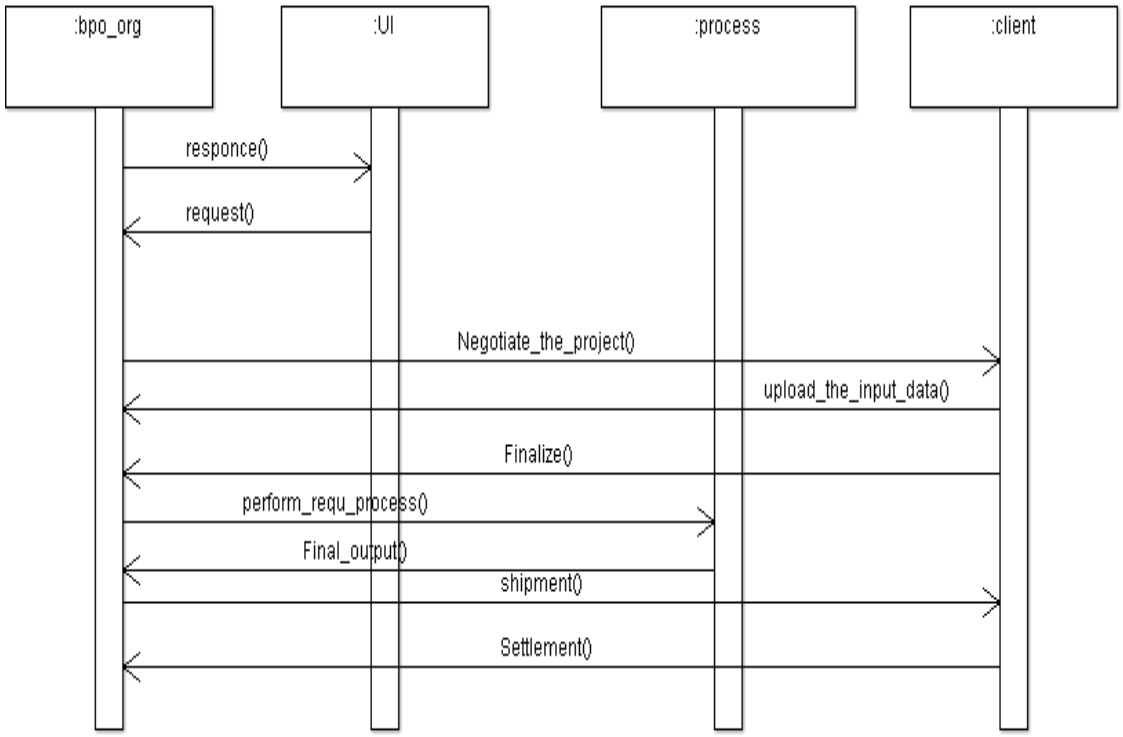


This class diagram has three class process agent, customer and database.

- **Agent –** is the class name. Its attributes are username, password, name, phone no and address. The operations performed by the agent class are login, giving details to customer and selling the product.

- **Customer –** is the class name. Its attributes are name, phone no, address. The operations performed are attending the call, asks about the product.
- **Database –** is the class name. The operations performed are storing customer details, verifying login and updating the customer details.

UML INTERACTION DIAGRAM:

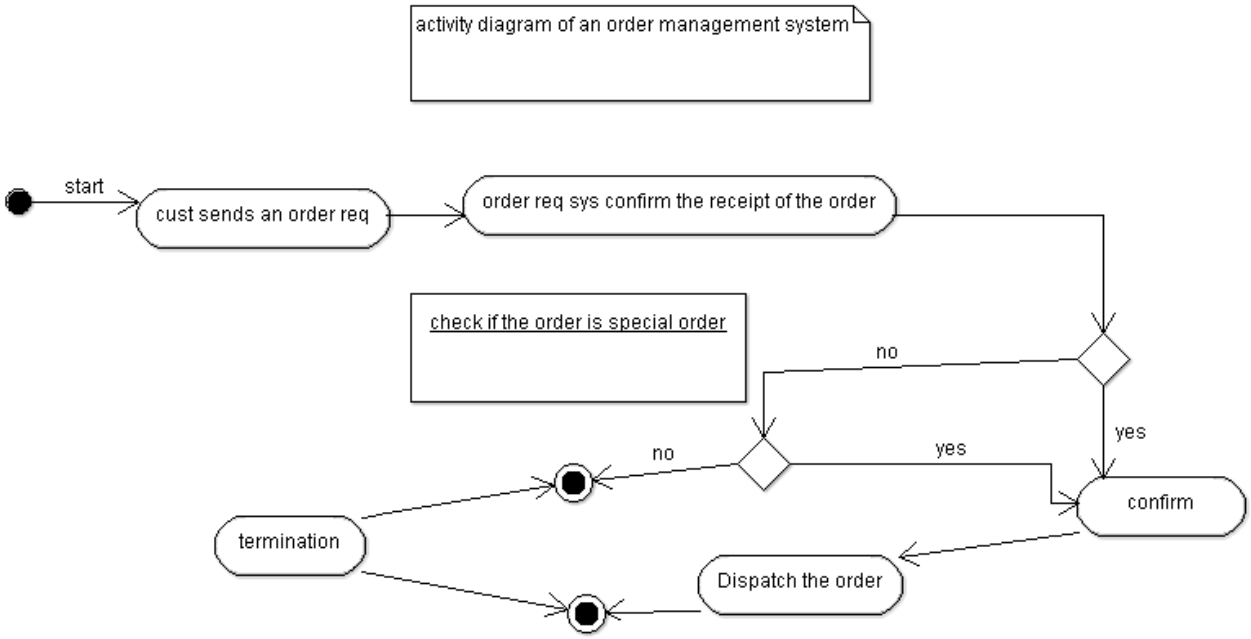


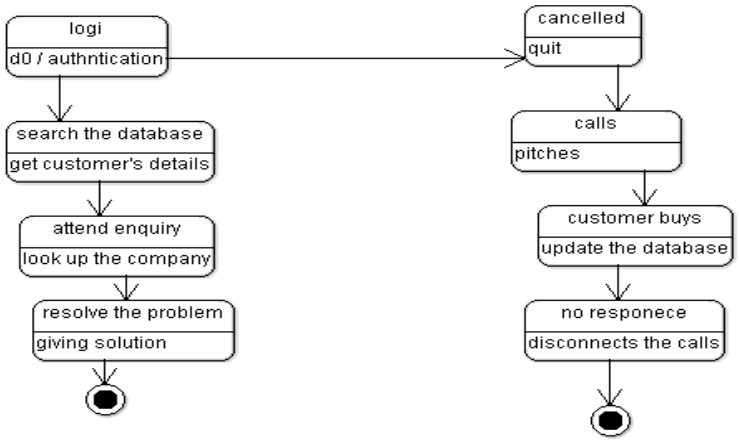
The single use case in BPO management system is taken and sequence of operations followed in the use case.

The BPO has the following sequence of process:

1. Agent fetches the data from the database.
2. Database provides the details of the customer to agent and agent dials to the customer.
3. Customer responds to the agent and agent pitches his/her product.
4. If necessary customer buys else discards.
5. Agent updates the call history.
6. Proceeds with another call.

UML STATECHART AND ACTIVITY DIAGRAM:



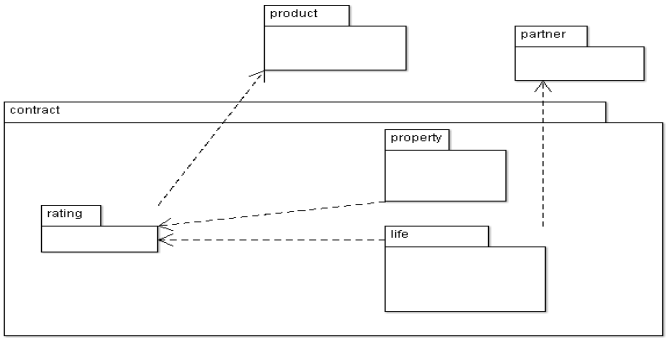


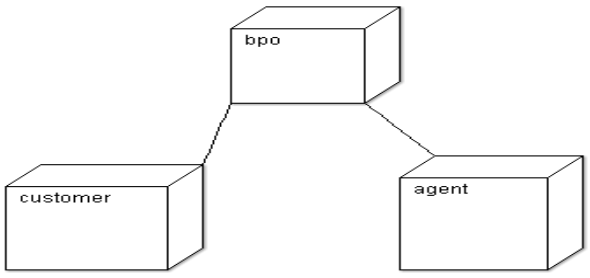
UML PACKAGE DIAGRAM:

The three layers in BPO management systems are

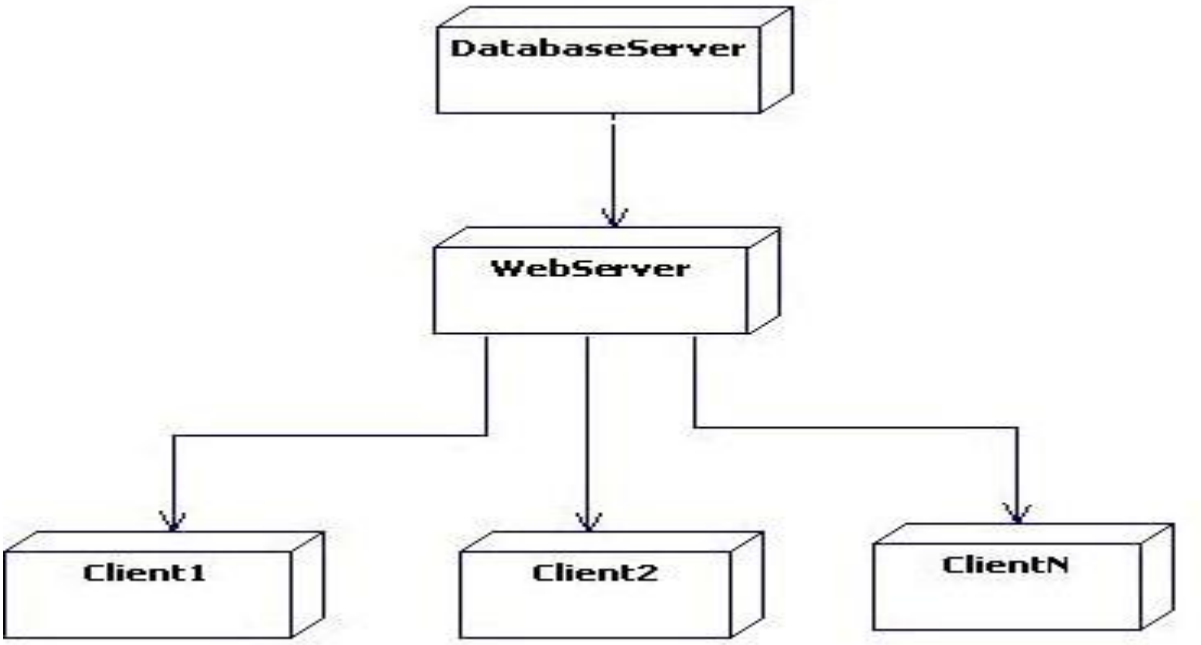
- **The User interface layer** – consists of the web and login. This layer describes how The agent logs on to the website and gets the customer details.

- **The Domain layer –** Shows the activities that are performed in the BPO management system. The agent makes the call and he pitches about the product to customer and makes sale. Finally agent aborts the call and proceeds with another call.
- **The Technical service layer –** The customer details are shown in the database. If the customer buys product it makes the sale entry.
- **Activity final node –** The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes.
- **Activity –** The rounded rectangles represent activities that occur. An activity may be physical, such as Inspect Forms, or electronic, such as display the BPO details.
- **Flow/Edge –** The arrows on the diagram. Although there is a subtle difference between flows and edges I have never seen a practical purpose for the difference although I have no doubt one exists. I'll use the term flow.





UML TECHNICAL SERVICE LAYER:



UML DOMAIN OBJECT LAYER:

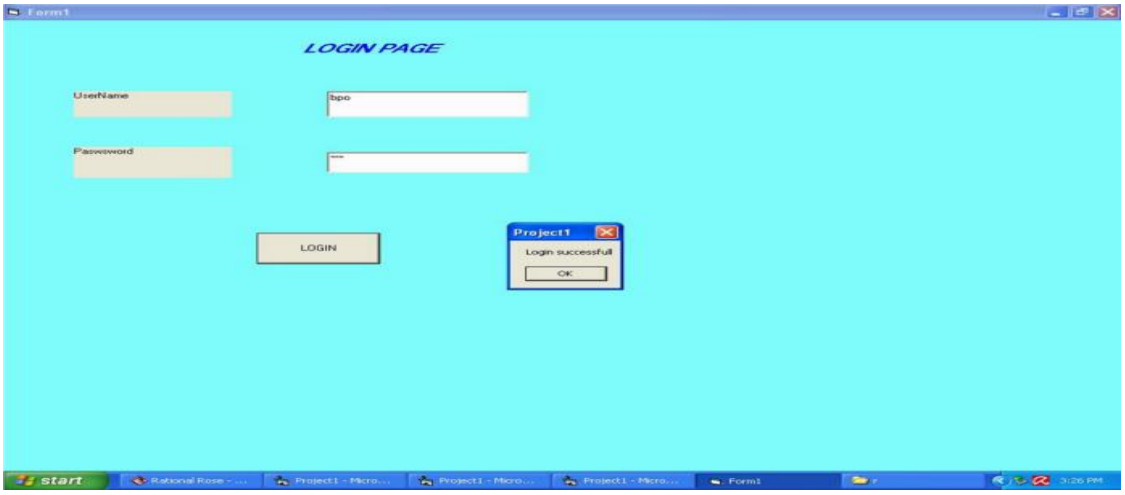
BPO Organization

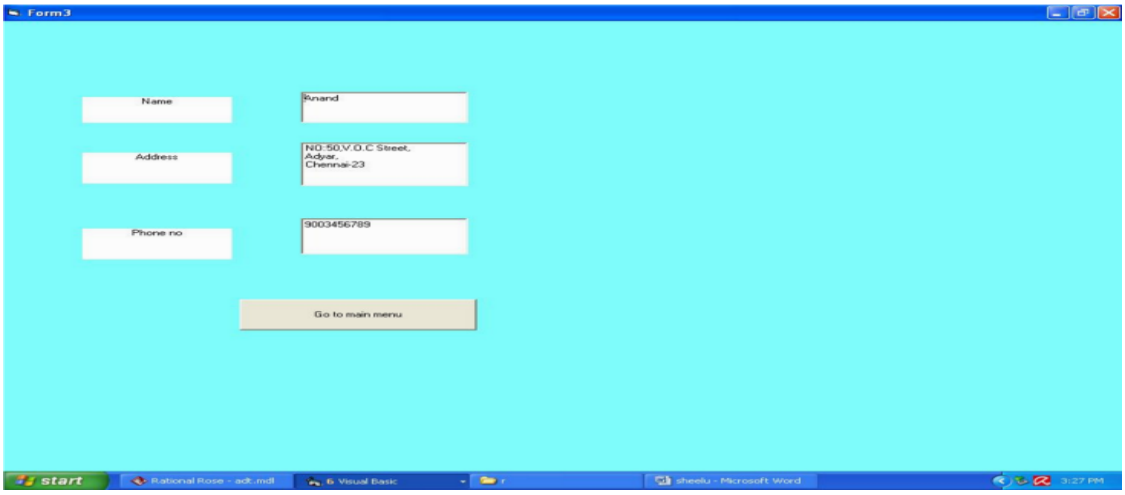
```
import java.util.Vector;

public class BPO organization {
  public String orgname;
  public String Address1;
  public String Address2;
  public String Address3;
  public String State;
  public Integer Contact_no;
  public Varchar Email;
  public Integer Toata;
    public Vector  myproject Details;
    public Vector  myemplopyee details;
    public Vector  myprocess;
  public void addOrg() {
  }
  public void updateOrg() {
  }

}
```

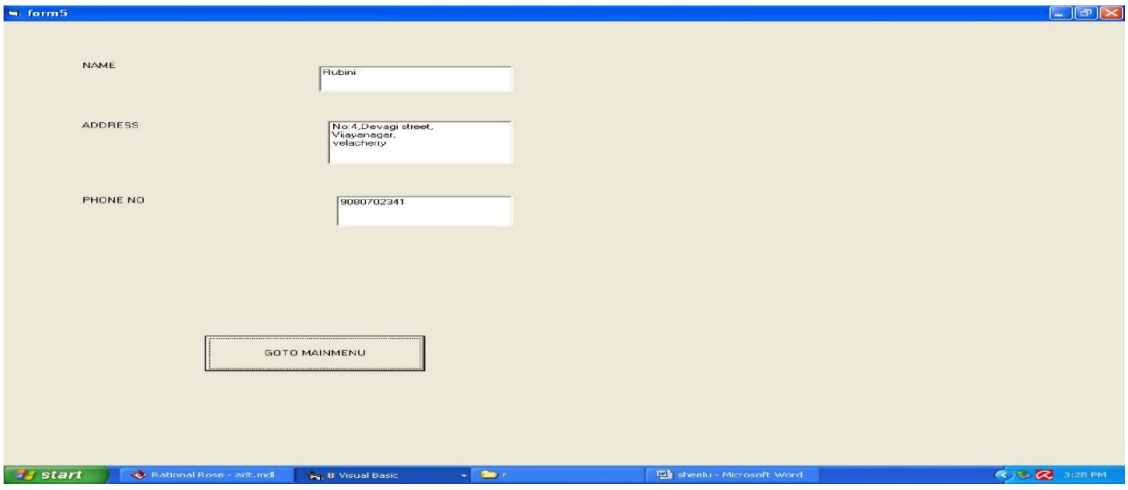
Employee Details:-

```
import java.util.Vector;

public class emplopyee details {
  public String emp_name;
  public Char emp_dob;
  public Varchar emp_addr;
  public ph.no emp;
    public Vector  myBPO organization;
    public Vector  myproject Details;
  public void addEmp() {
  }
  public void delEmp() {
  }
  public void salCal() {
  }

}
```

USER INTERFACE LAYER:







Result:

Thus the BPO Management System has been done successfully by using Argo-UML.

EX.NO.:14	LIBRARY MANAGEMENT SYSTEM

AIM:

To design Library Management System by using Argo-UML tool.

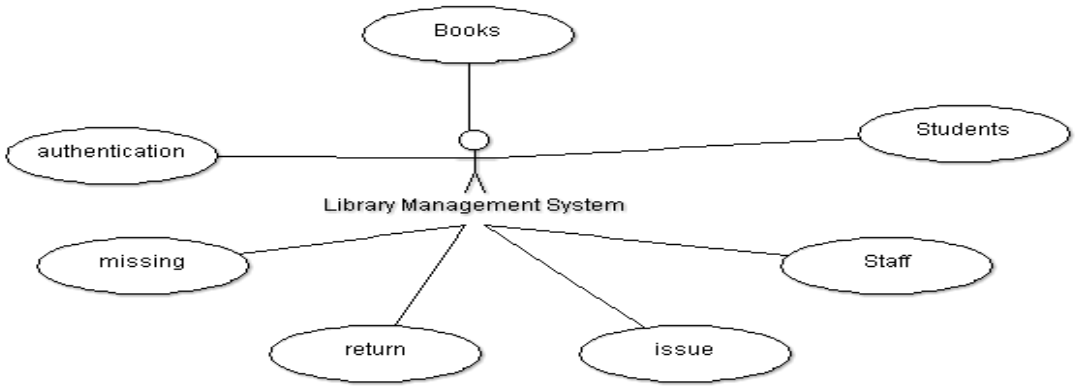
PROBLEM ANALYSIS AND PROJECT PLAN"

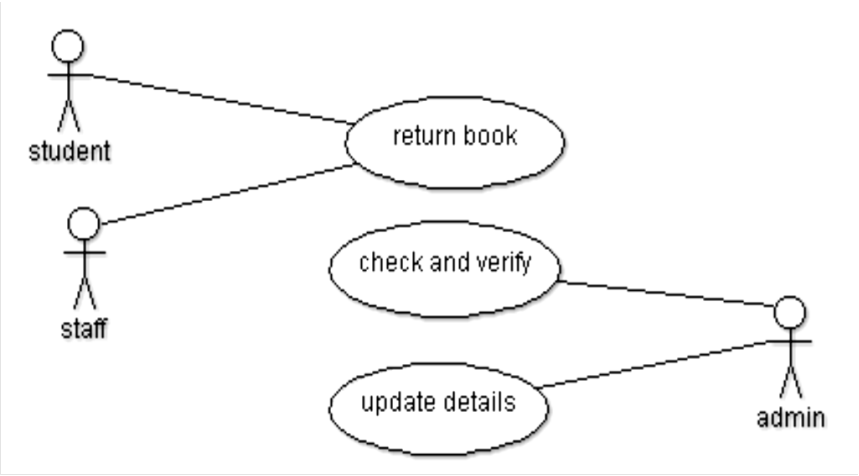
To simplify the process of applying Library Management System, software has been created by designing through ARGO-UML tool.

PROBLEM STATEMENT:

- The software to be designed will control a simulated library management system.
- As the student enters the library students gets a library card.
- The library card is used for borrowing, lending and paying fines for books.
- The librarian is the person who is an intermediate between the student and the database.
- The library will service more than one student
- A student will be required to show his/her library card to the librarian.
- Once the librarian gets to know that the borrowed book is lost .The librarian immediately accesses the database and updates that the book is lost in the login and fine is also calculated.

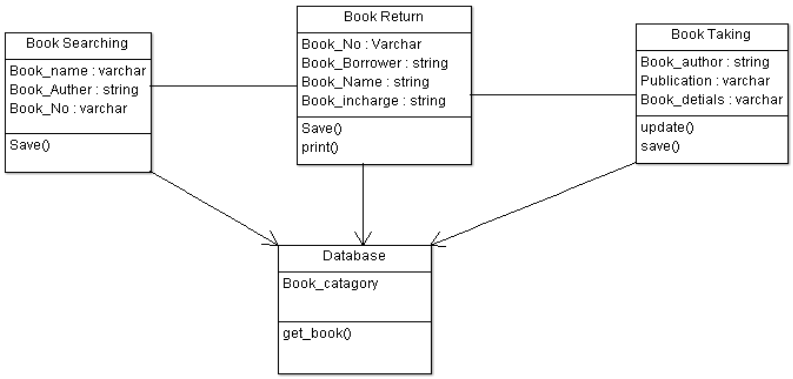
UML USECASE DIAGRAM:





1. Students and Staffs access his account by providing the correct userid and password.
2. The Students & Staffs searches for the Availability of Books.
3. The Students selects the Book which he wants from the listed stocked books.
4. The Book is issued to the Student or Staff and the database is updated.
5. The Student can either can continue to getting another books from library.

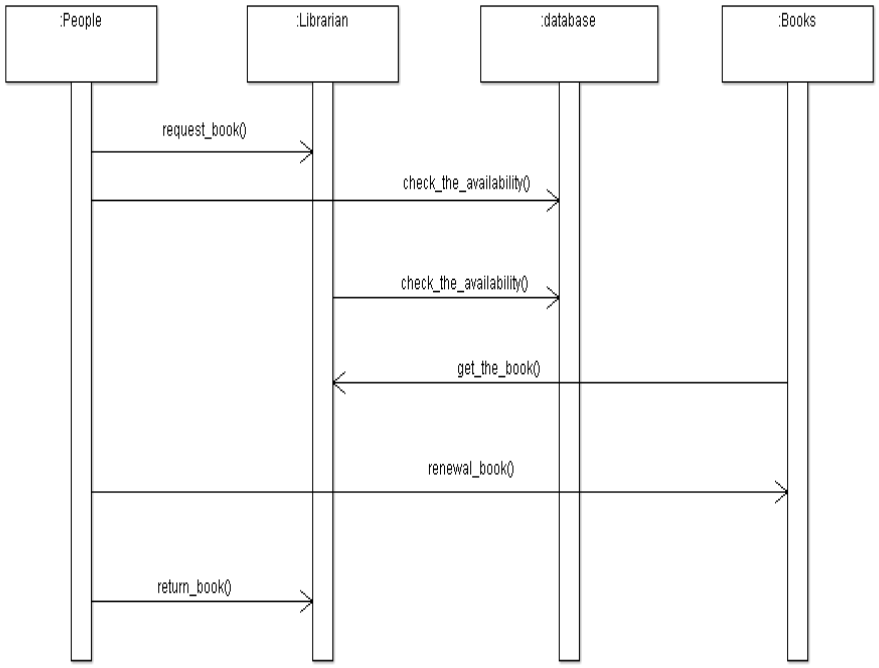
UML CLASS DIAGRAM:



- Administration class
- Get Membership
- Search books
- Issue book
- Return book
- Journals

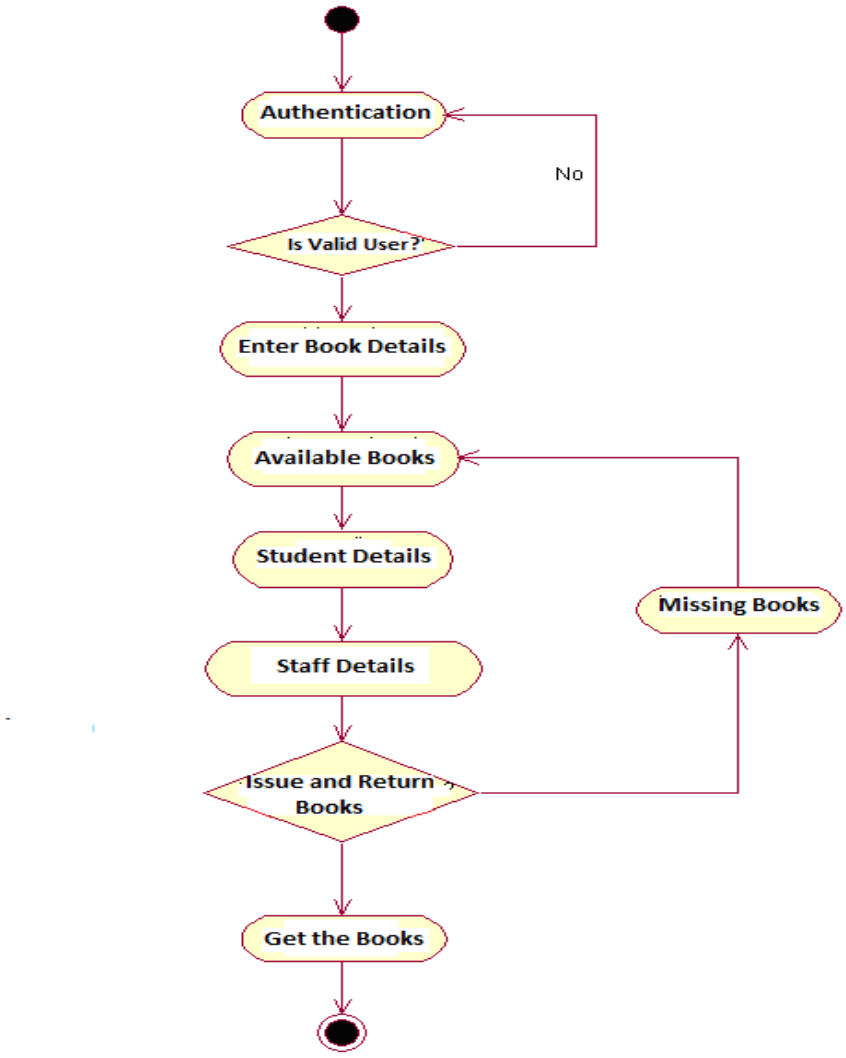
- Here Administration class is related to get membership class.
- Search books are related to journal class.
- Issue book is related to return book.

UML INTERACTION DIAGRAM:



- The student request membership to the admin.
- The admin generates membership to the student.
- The student enters the login id.
- The admin verifies the login id with the database.
- If the admin is correct authentication is granted by the admin. Otherwise the login id must be entered again.
- The student enters the details of the book (searching the book).
- The admin checks for the availability of the particular book.
- If that book is available it is borrowed.
- After borrowing book the admin updates the status in the student database.

UML STATECHART DIAGRAM:

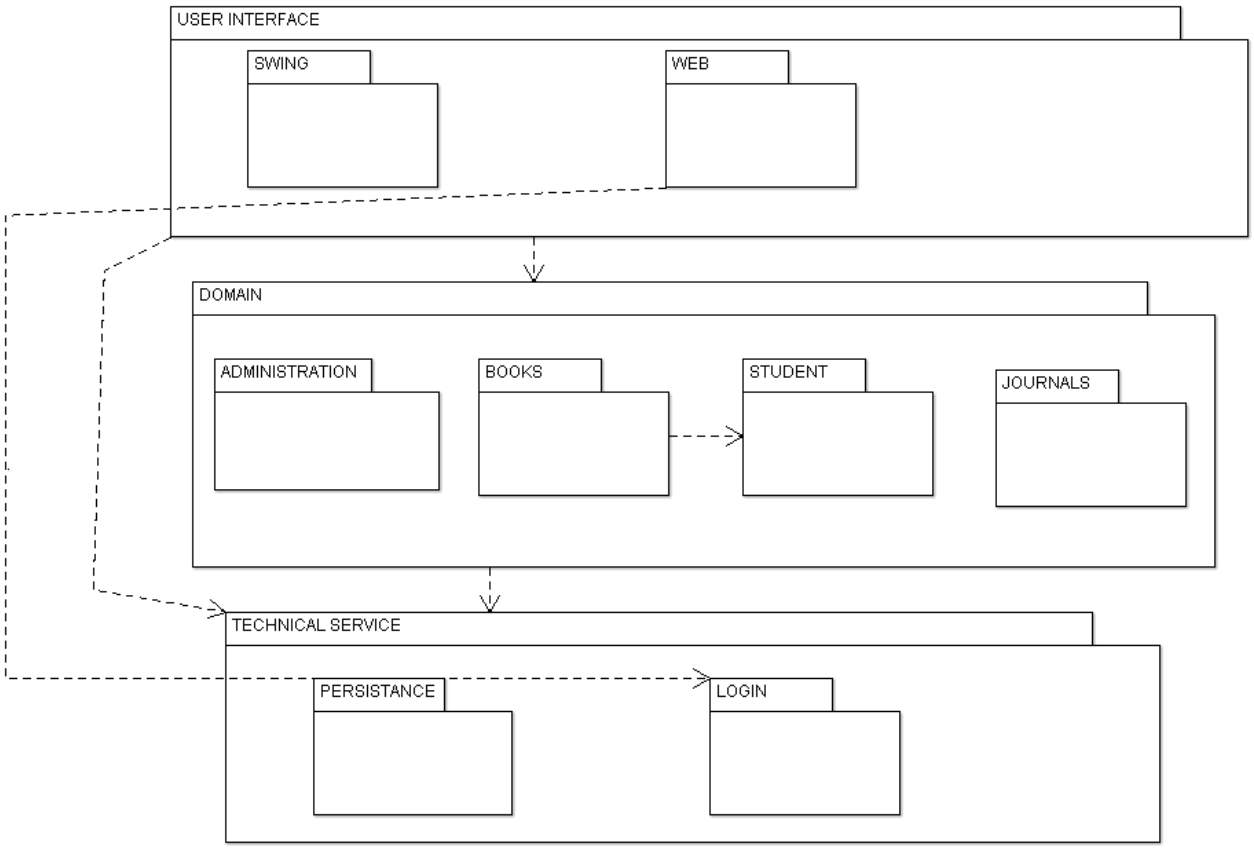


UML TECHNICAL SERVICE LAYER:

S. No.	Name of the user	Book Name	Book No	Date of Issue	Date of Return
1.	Monisha	Java	2426	17-06-2017	26-06-2017
2.	Swetha	Networks	1282	15-07-2017	24-07-2017
3.	Mutta	DPSD	3782	11-08-2017	20-08-2017

S. No.	Book Name	Book Author	Book No	Price
1.	Java	Complete Reference	2426	300/-
2.	Computer Networks	S.Davie	1282	500/-
3.	Software Engineering	R.S.Pressmen	9090	400/-

UML PACKAGE DIAGRAM:

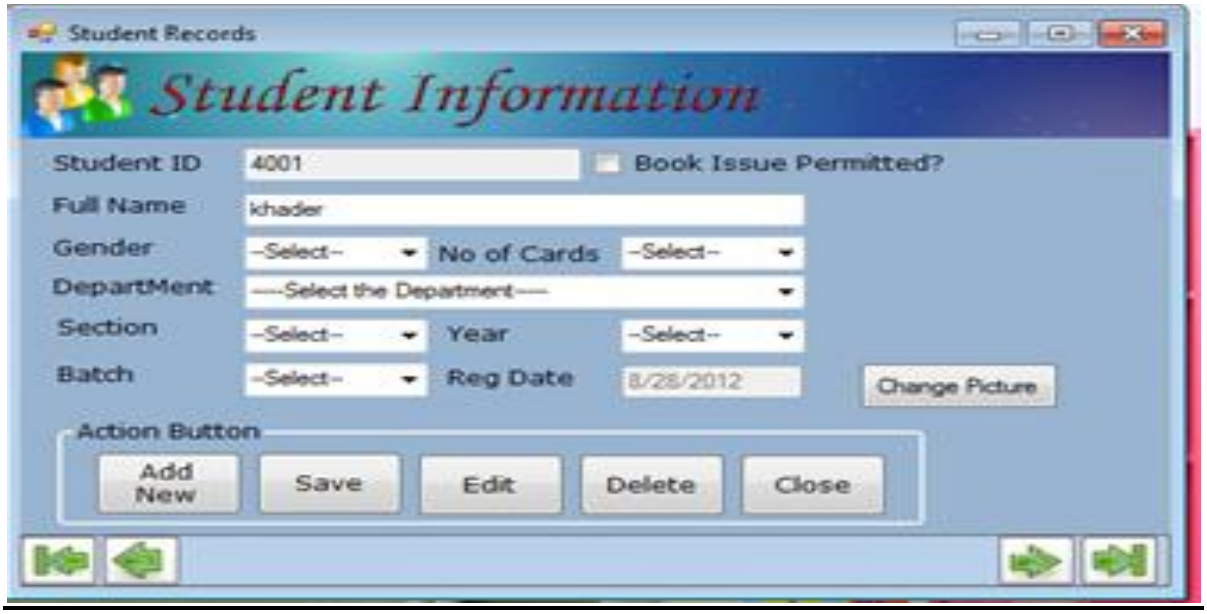


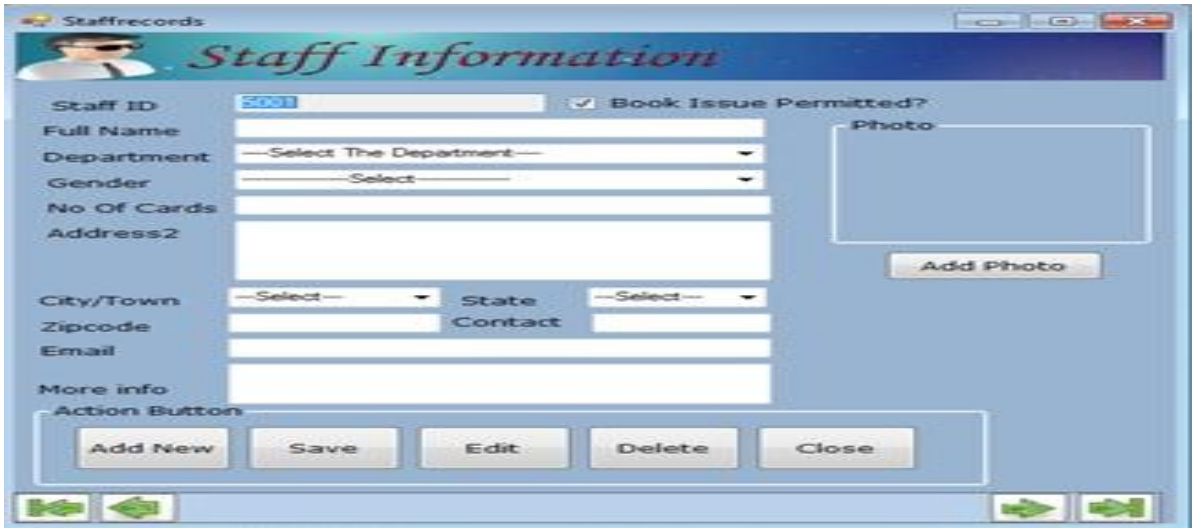
UML DOMAIN OBJECT LAYER:

```
public class Books {
        Private object Name;
        Private object Identify;
        Private object Book Details;
        Private object Edition;
        Public void Book Title () {
        }
        Public void Edition () {
        }
        Public void Price () {
        }
        Public void Remarks () {
        }
        Public void Savings () {
        }
}
```

```
public class Staff {
        Private object Name;
        Private object identity;
        Private object Address;
        Public void Allotment of Books () {
        }
        Public void Return Books () {
        }
        Public void Fine for Missing () {
        }
}
```

USER INTERFACE LAYER:





Result:

Thus the Library Management System has been done successfully by using Argo-UML.

EX.NO.:15	STUDENT INFORMATION SYSTEM

AIM:

To design Students Information System by using Argo-UML tool.

PROBLEM ANALYSIS AND PROJECT PLAN:

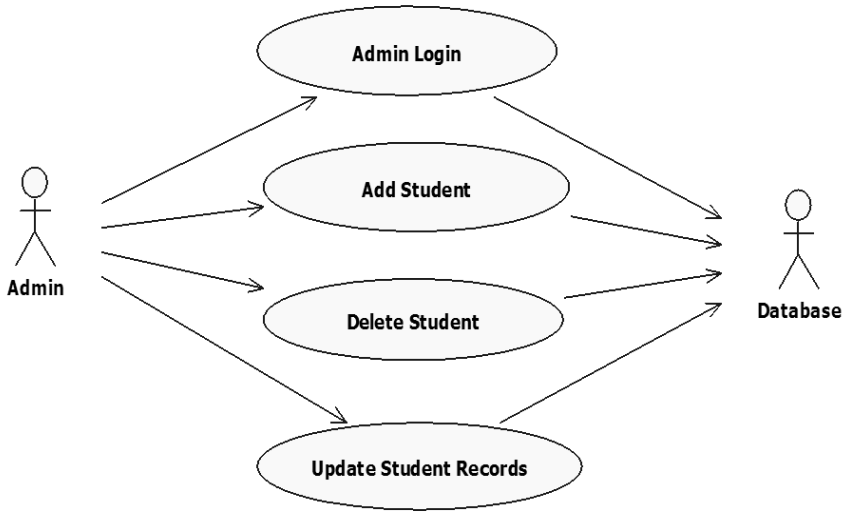
To simplify the process of applying, software has been created by designing through ARGO-UML tool.

A Student Information System (SIS) is a software application for educational establishments to manage student data. Student information systems provide capabilities for entering student test and other assessment scores, building student schedules, tracking student attendance, and managing many other student-related data needs in a school, college or university.

PROBLEM STATEMENT:

- Effective for Administration Purpose
- Cheap
- Better Service

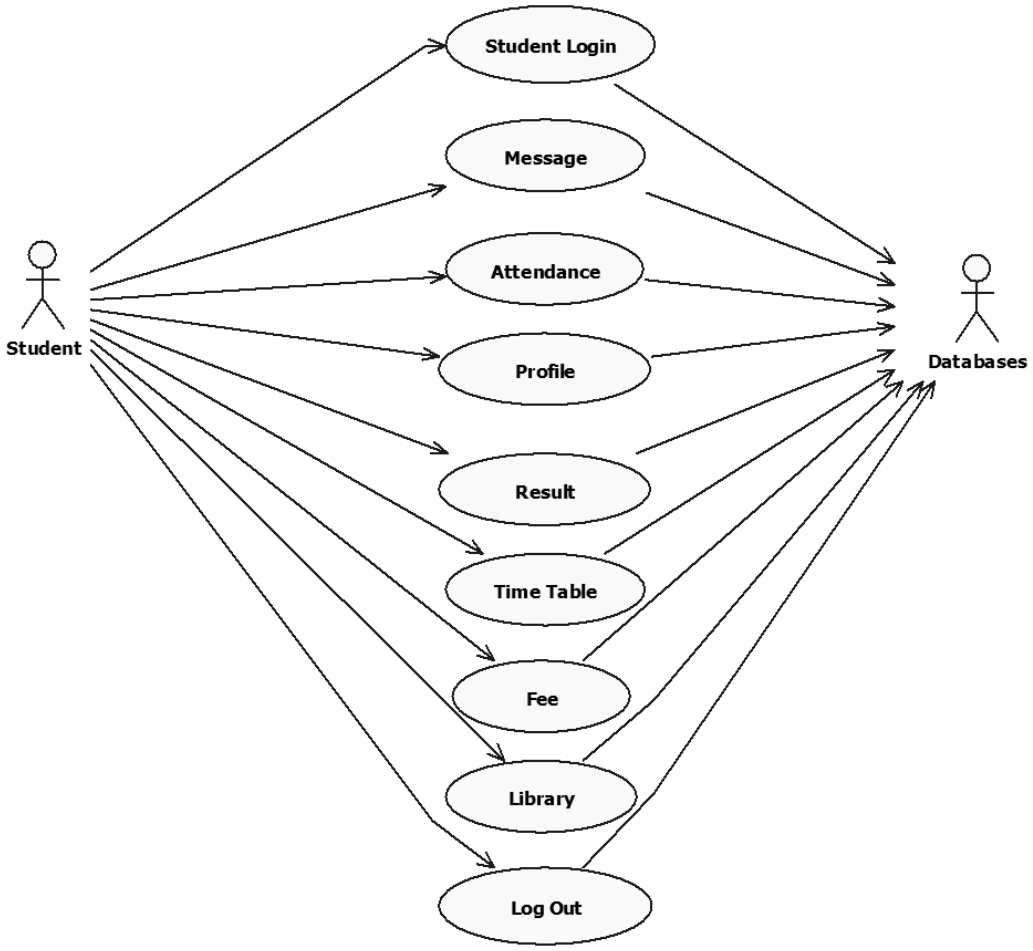
UML USECASE DIAGRAM:



The actors in this use case diagram are Admin, Student, and Database. The use cases are the activities performed by actors.

Admin register login, and store the student records details in database.

Student Register from the Student Login process.
Then the database is searched for details and verified.
Database stores the details and returns acknowledgement.



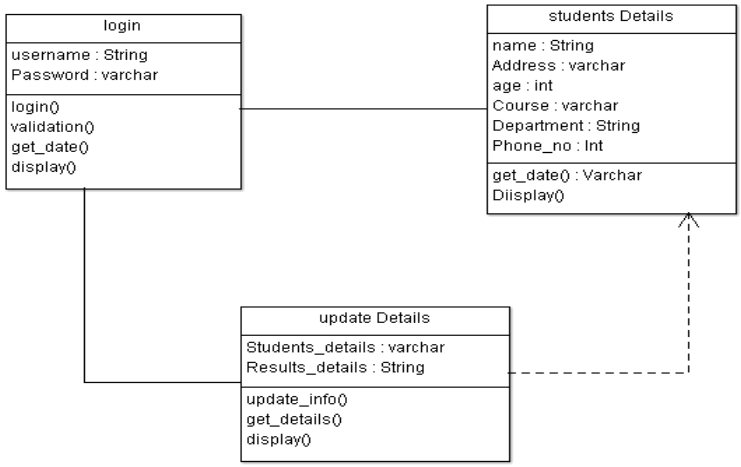
UML CLASS DIAGRAM:

This class diagram has three classes Login, Student details and Update details in database.

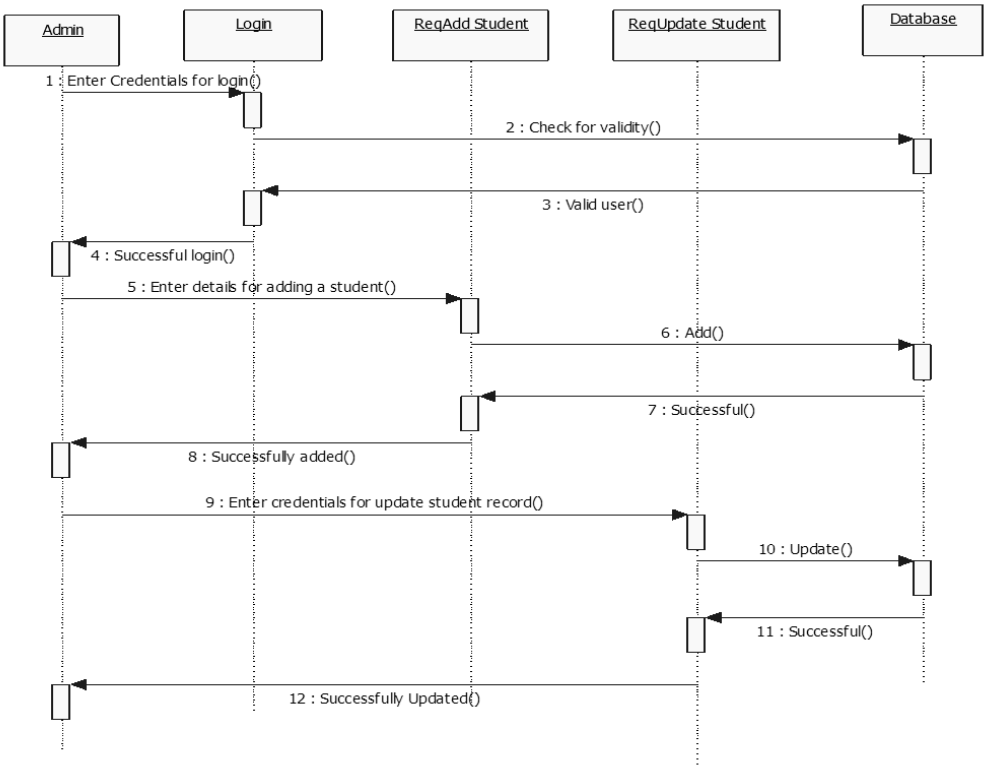
a. **Students** – is the class name. Its attributes are name, Address, DOB, Gender, College, Subjects, Semester, Year, Degree, and Branch. The operations Performed in the students class, Store database and Update.

b. **Administration**– is the class name. Its attributes are Login, Password and database. The operations performed are Student Details store in database and send acknowledgement.

c. **Database** – is the class name. The operations performed are storing Search and storing the values.



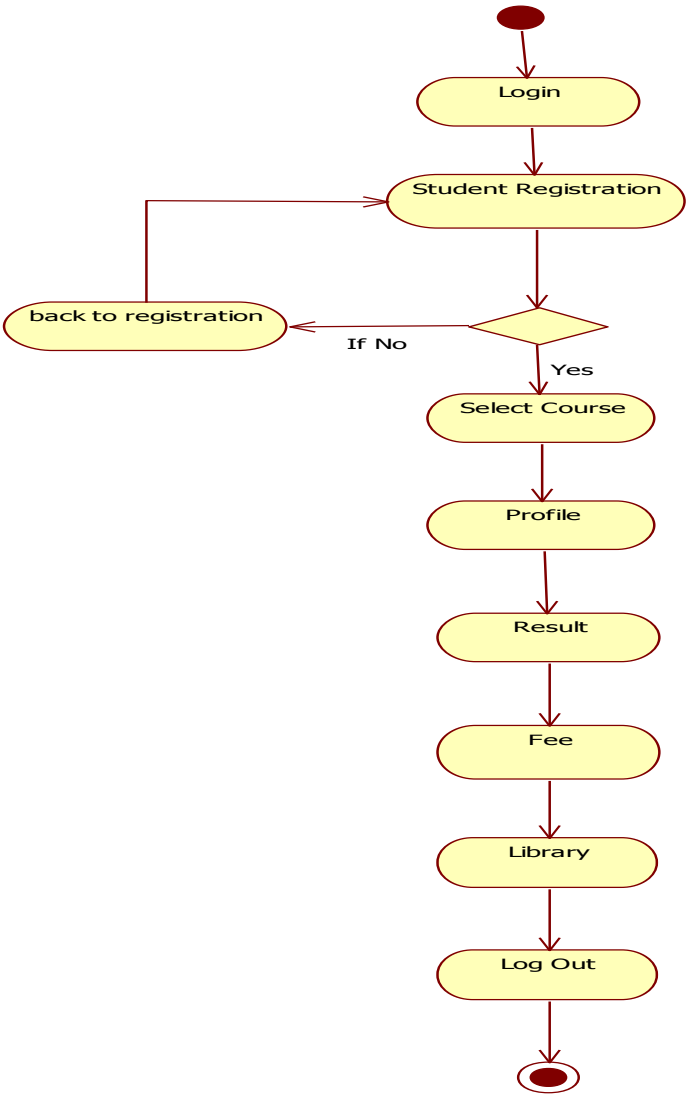
UML INTERACTION DIAGRAM:



The sequence diagram describes the sequence of steps to show:

- The Admin login and registering for Add Student Details.
- The verification done by the interface and sending acknowledgement for registration.
- Searching the database with login and displaying it for maintenance.

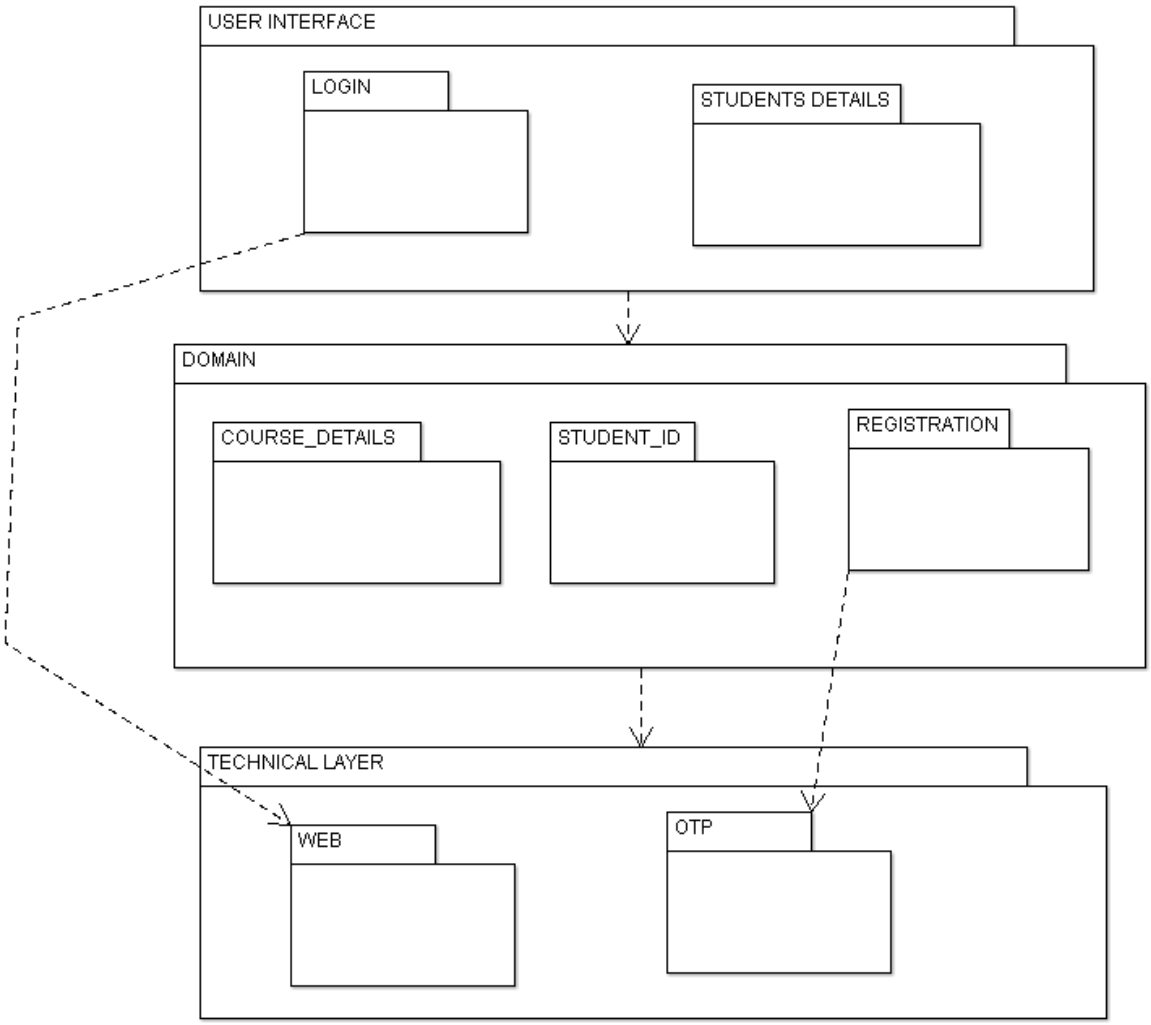
UML ACTIVITY DIAGRAM:



This activity diagram flow of stepwise activities performed in recruitment system.

- The student details are Add and stored in database.
- Select the course from the given Course by student.
- Search Profile and Result with login and if data present in the database.
- The searched data is displayed if available and then Log Out.

UML PACKAGE DIAGRAM:



UML TECHNICAL SERVICE LAYER

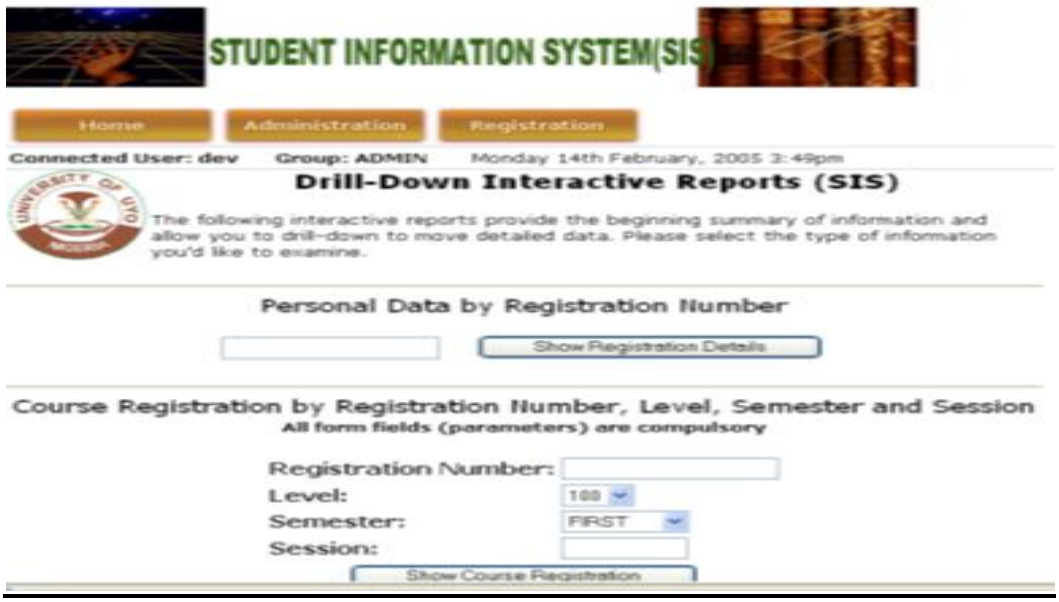
S. No.	Name	Roll No	Department	Year	Percentage
1.	Ismail	15283	CSE	III	70
2.	Vickey	15238	CIVIL	III	80
3.	Suhail	16238	ECE	II	84
4.	Trippy	18998	MECH	IV	96

UML DOMAIN OBJECT LAYER:

```
import java.util.Vector;
public class students Details {
  public String name;
  public varchar Address;
  public int age;
  public varchar Course;
  public String Department;
  public Int Phone_no;
    public Vector  mylogin;
  public Varchar get_date() {
  return null;
  }
  public void Diisplay() {
  }
}
```

```
import java.util.Vector;
public class update Details {
  public varchar Students_details;
  public String Results_details;
    public Vector  mylogin;
  public void update_info() {
  }
  public void get_details() {
 }
  public void display() {
  }
}
```

USER INTERFACE LAYER



Result:

Thus the Student Information System has been done successfully by using Argo-UML.

Additional Experiments

Hospital Management System

AIM:

To model the " **Hospital Management System** " using various UML (Unified Modeling Language) diagrams.

PROBLEM STATEMENT:

The hospital has several specialized departments like Cardiology, Gynecologic, Orthopedics, Pediatrics, ENT etc. OPD is another independent department. A doctor is only associated with one specialized department at a time though he/she can be a member of the OPD(Outside Patients Department) department. Each doctor has a visiting time and day in a week.

At reception the patient details are entered and the fees are also taken and the patient is tracked on the basis of the Id generated.

In routine a patient can visit the doctors either directly selecting a doctor or by getting admitted to the hospital and then a doctor visits the patients.

A doctor can prescribe tests for the patient to perform. The patient visits the lab to get done the tests prescribed by his/her doctor. The reports are given to the patient. The payments pertaining to the tests are done at the reception. Referring the reports, the doctor prescribes the patient medicines or further tests or is asked to get admitted.

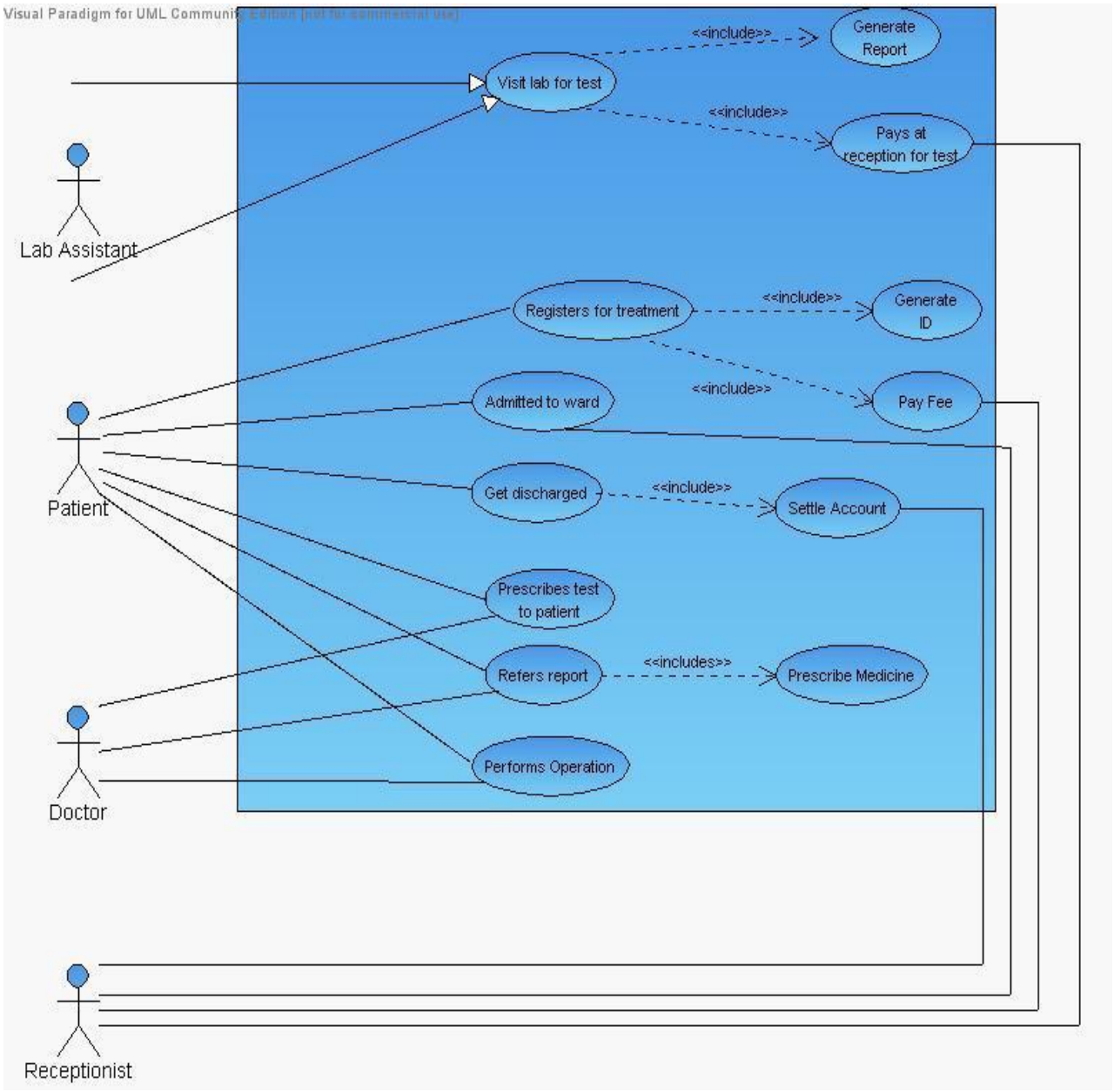
A patient is admitted into a ward of a specialized department (if available) as per the doctor's prescription. The number of wards is limited and if there is no vacant ward the admission of the patient is rescheduled.

As per the prescription of the doctor the patient is operated on a specified date and time as decided by the doctor who is doing the operation.

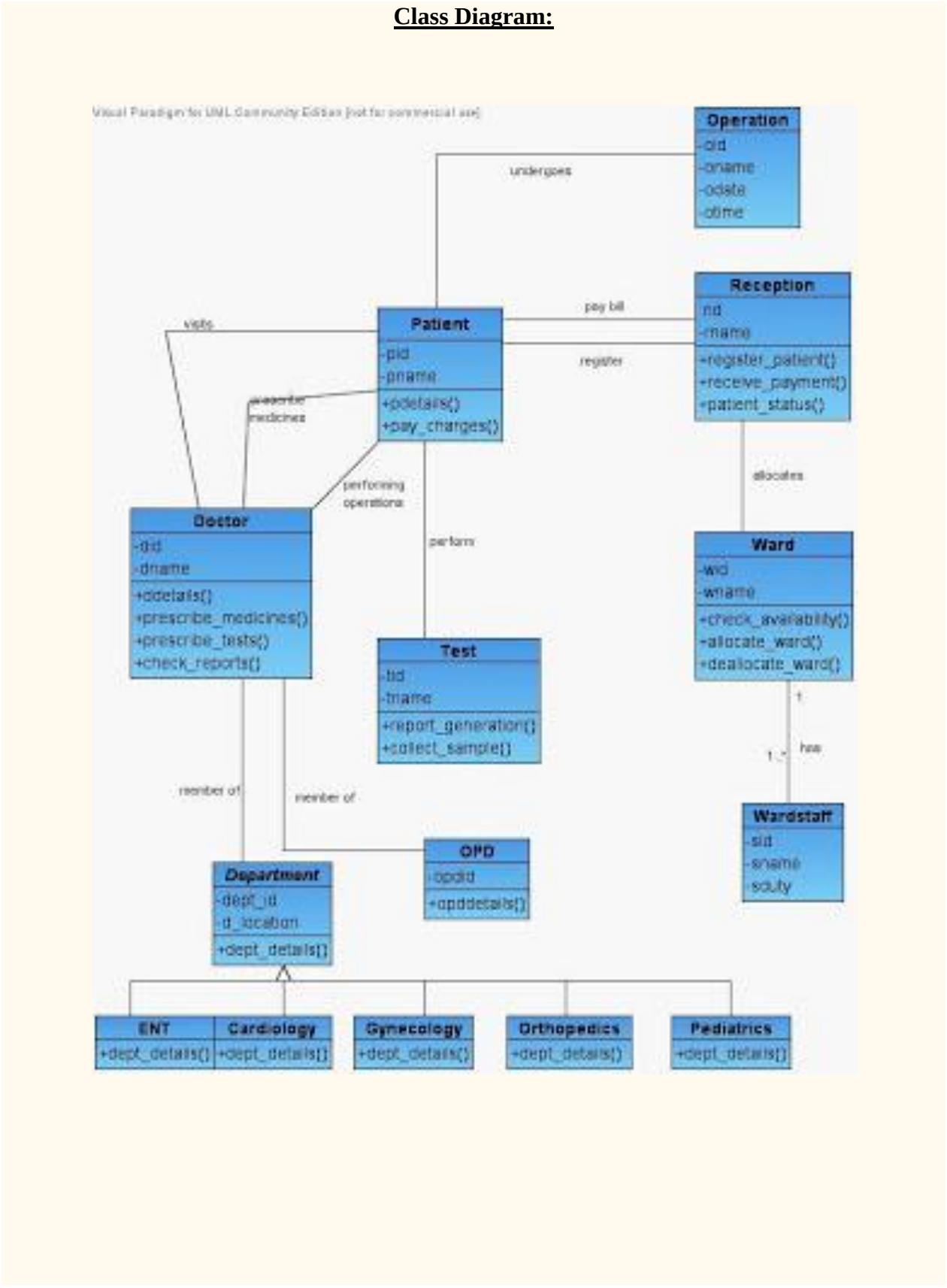
After the completion of the treatment a patient may get discharged on an advice of a doctor and upon the complete payment of all due charges at the reception. On payment of full dues the reception generates a discharge ticket for the patient.

UML Diagrams for Hospital Management System

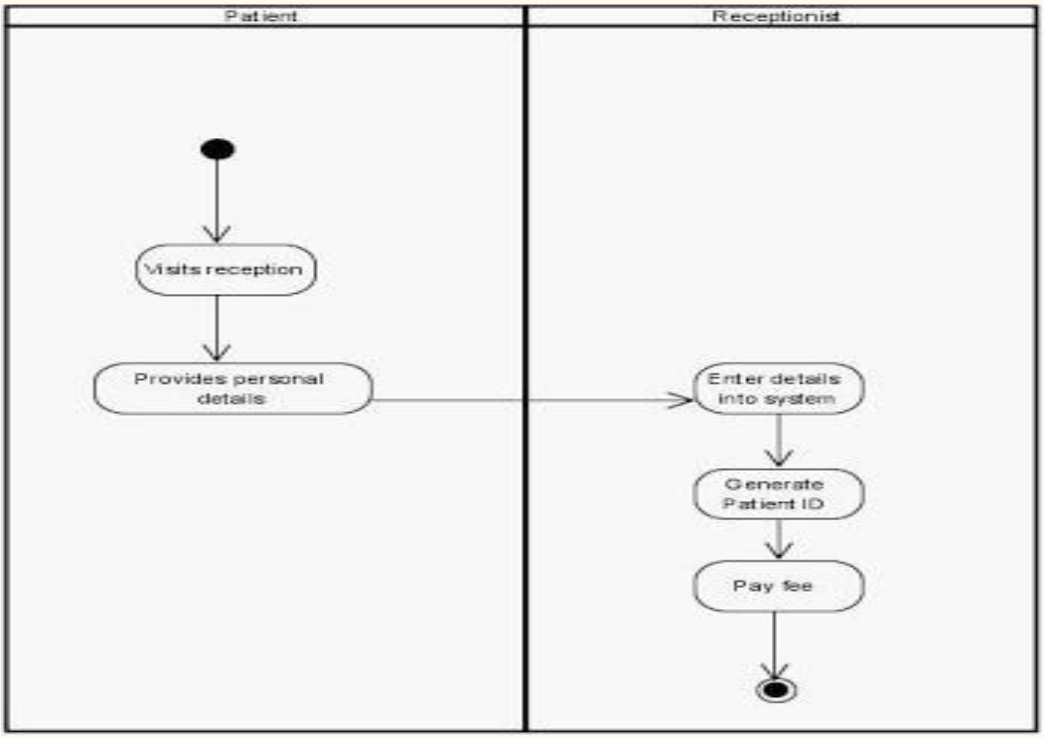
Use Case Diagram:



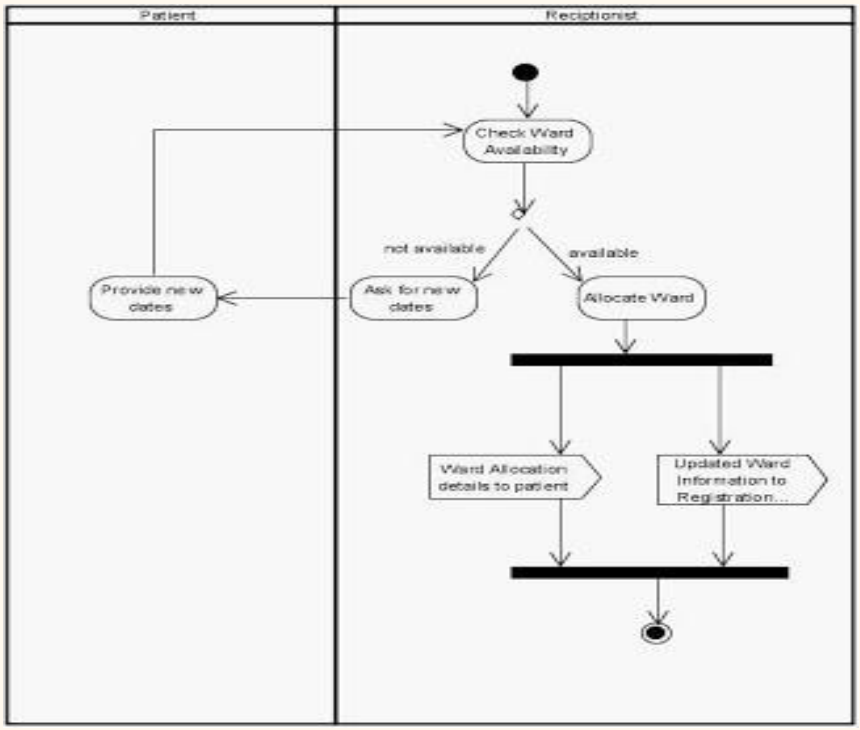
Class Diagram:



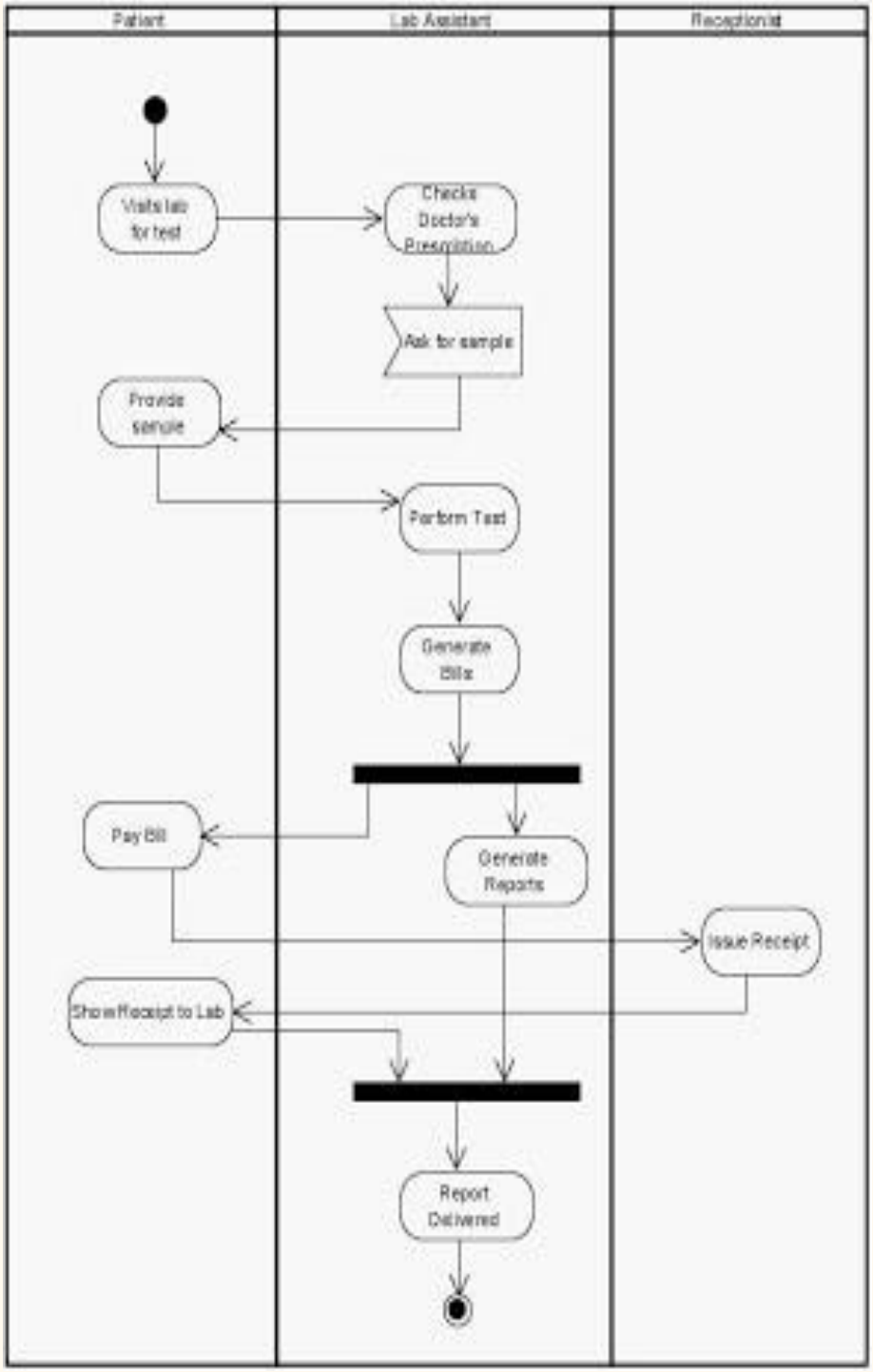
Activity Diagram Registration:



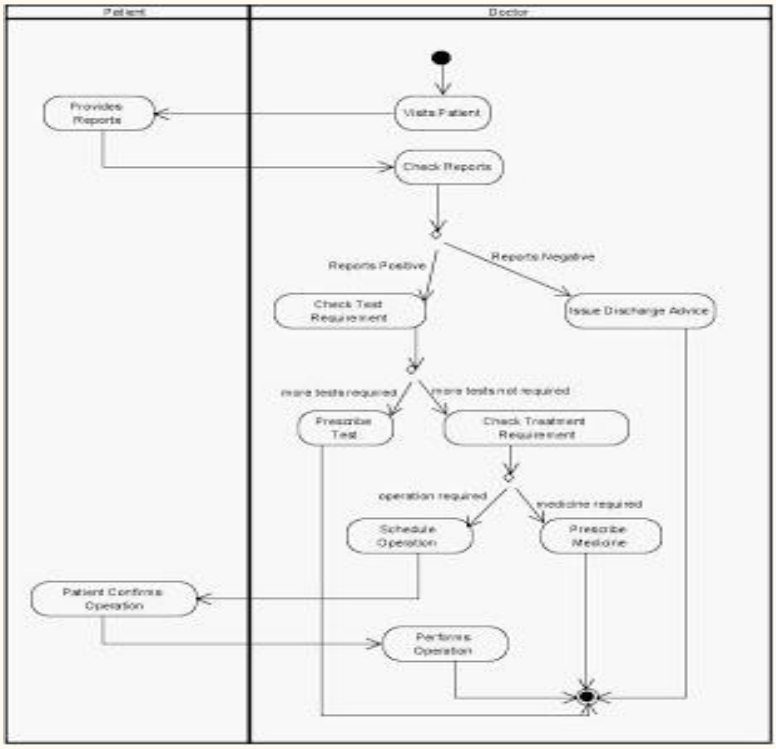
Activity Diagram for Ward Allocation:



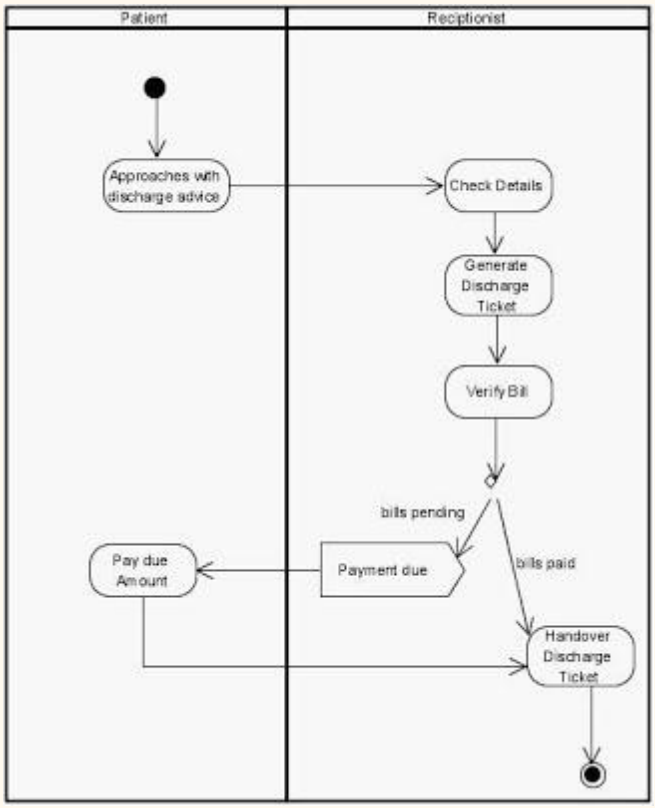
Activity Diagram for Tests to Perform:



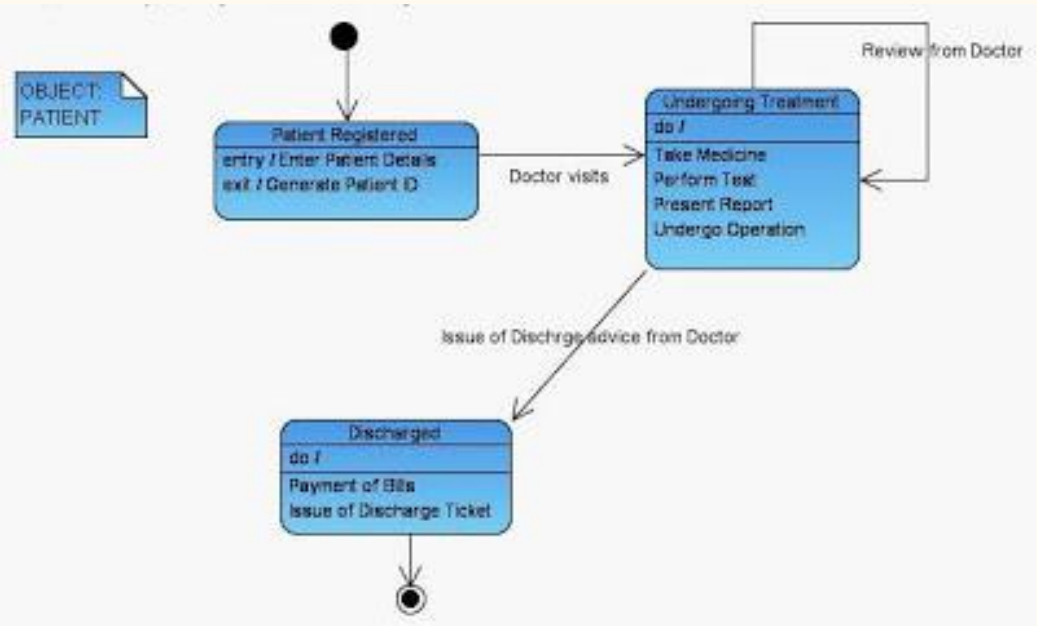
Activity Diagram for Treatment and Operations:



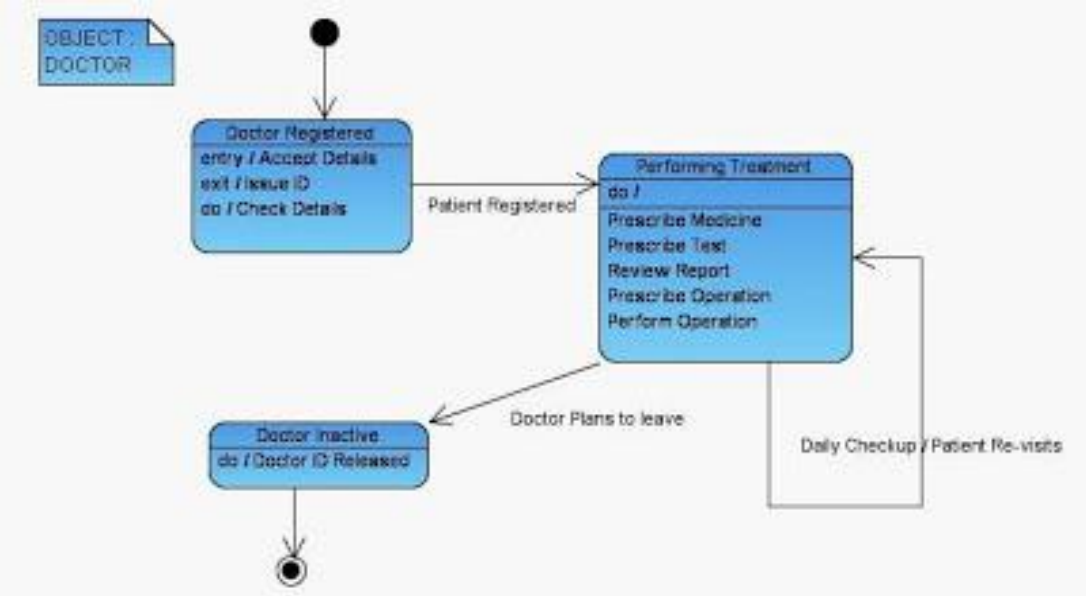
Activity Diagram Discharge:



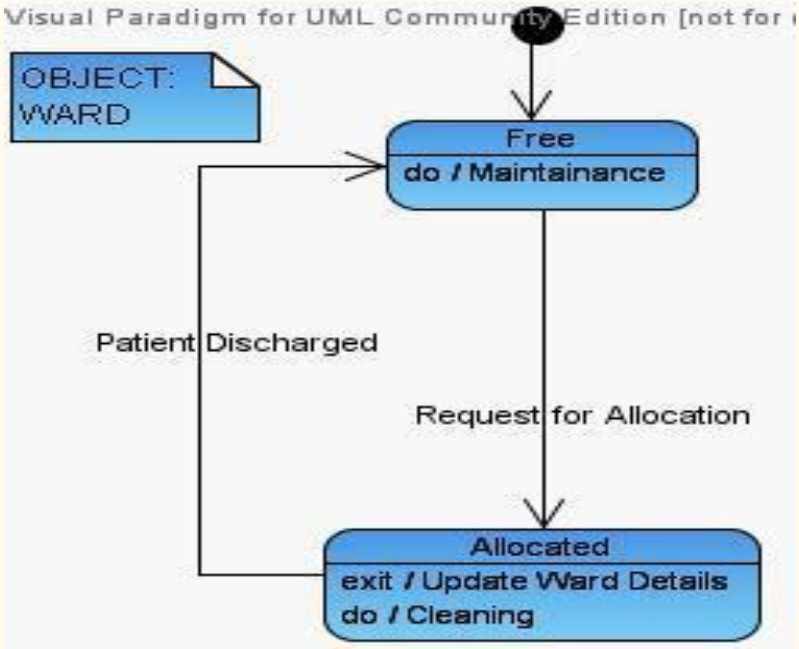
State Chart Diagram for Patient:



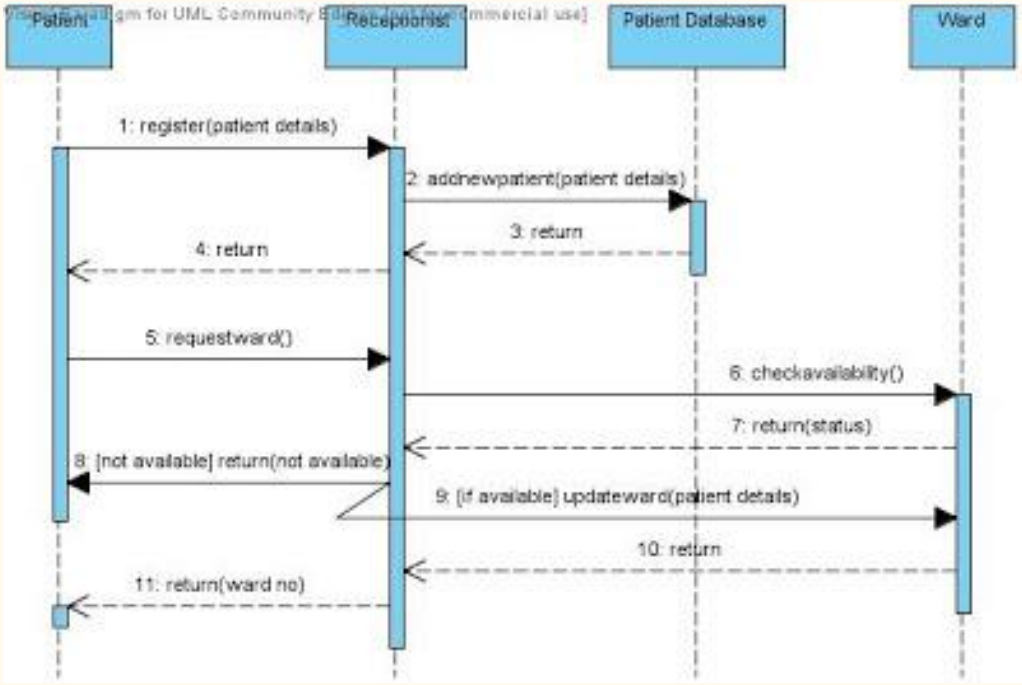
State Diagram for Doctor:



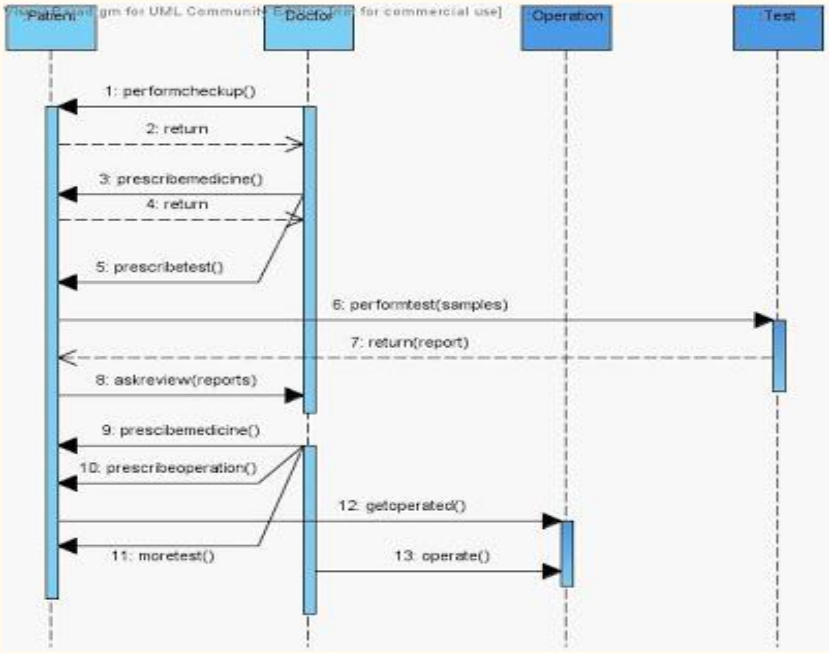
State Diagram for Ward Object:



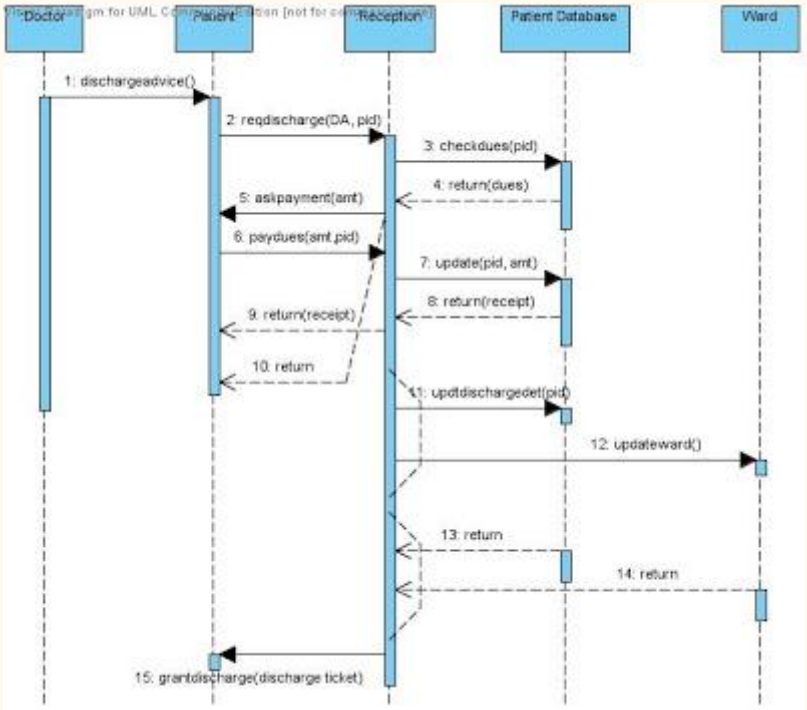
Sequence Diagram for Patient Admit / Registration:



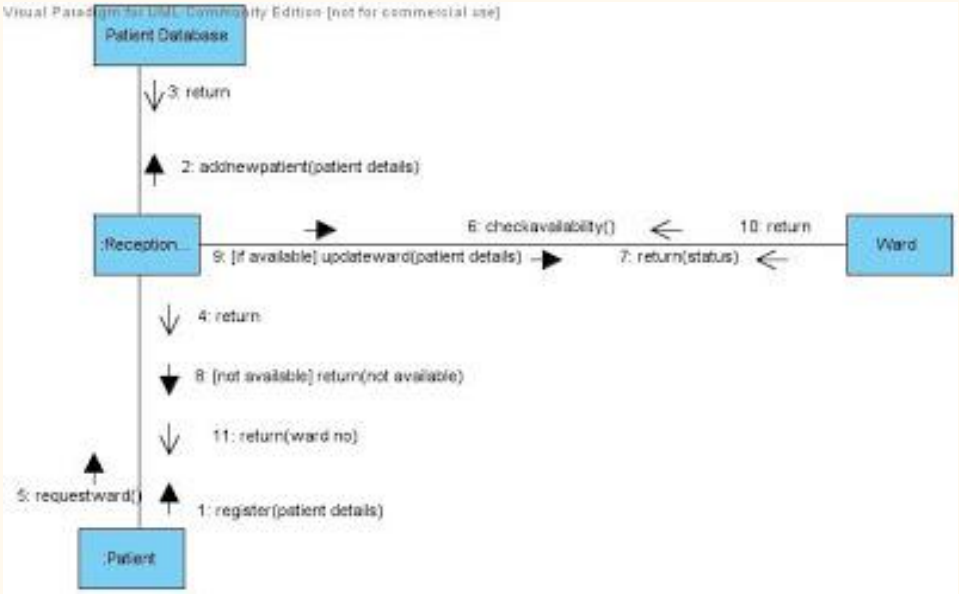
Sequence Diagram Test & Operation:



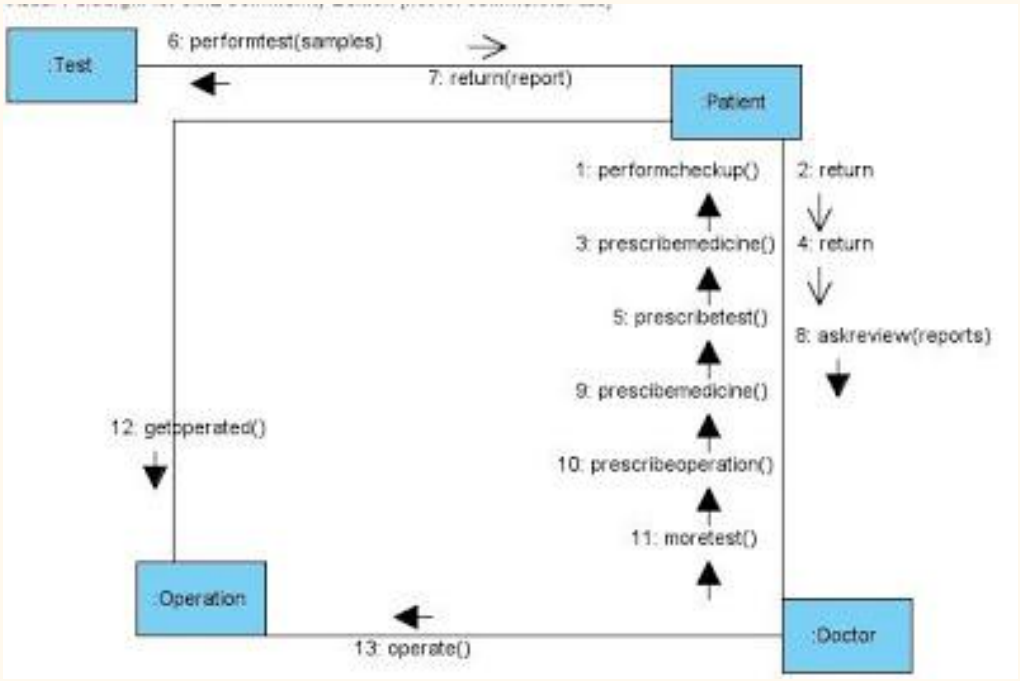
Sequence Diagram Discharge from Hospital:



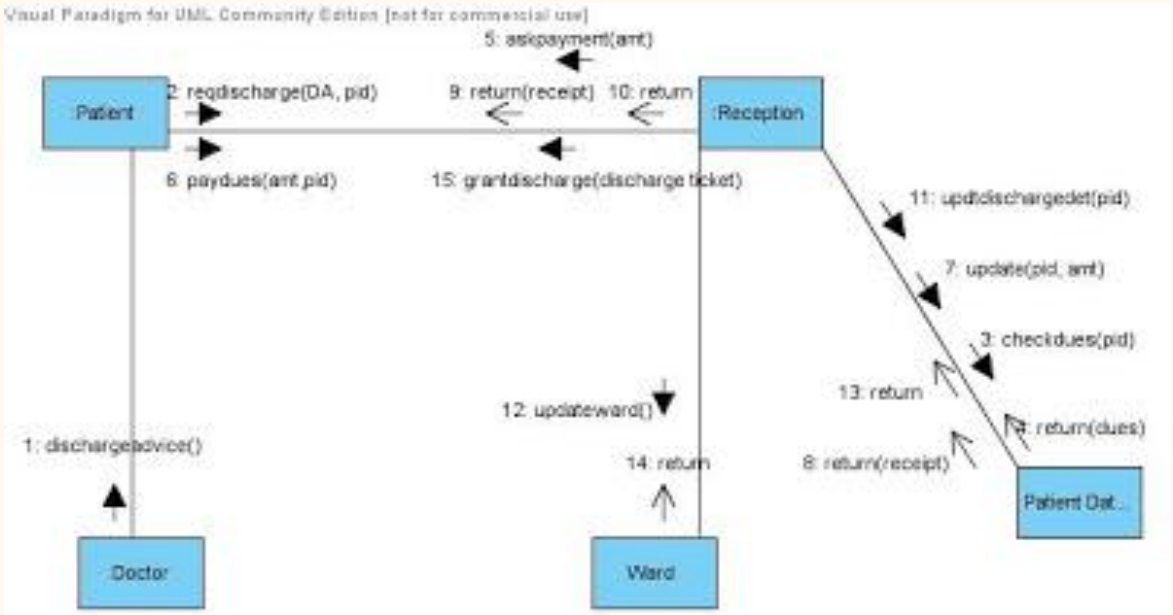
Collaboration Diagram Admit to Hospital:



Collaboration Diagram for Treatment at Hospital:



Collaboration Diagram for Discharge from Hospital:



Result:

Thus the UML diagram for Hospital Management System was drawn and code was generated using ArgoUML.

Banking System

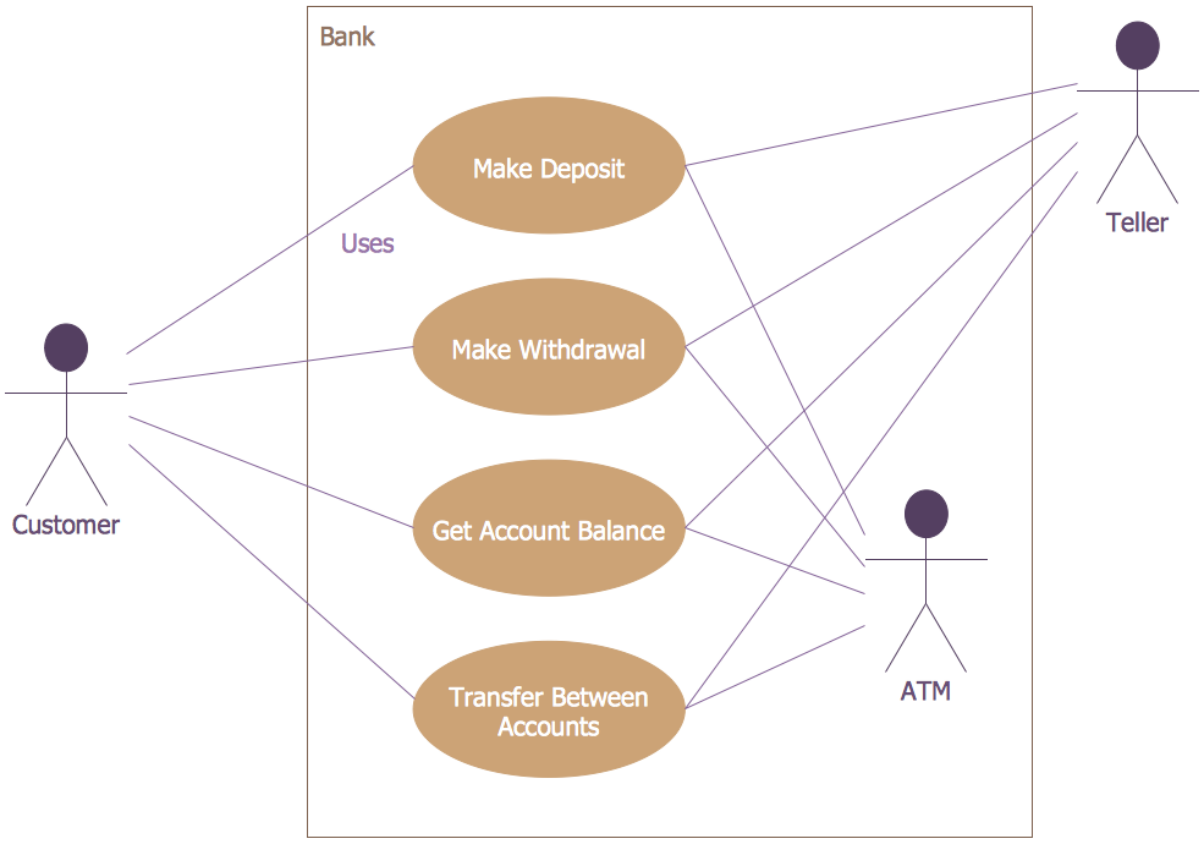
AIM:

To model the "**Banking System**" using various UML (Unified Modeling Language) diagrams.

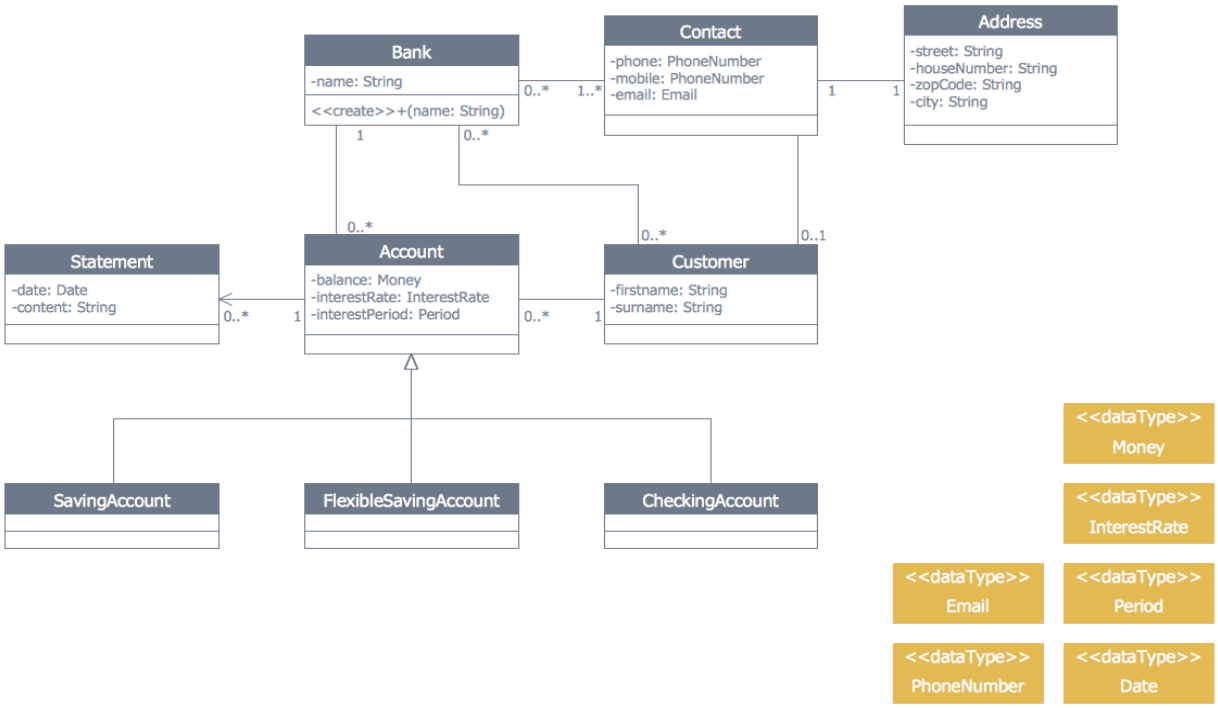
PROBLEM STATEMENT:

This banking process consists of five divisions. There are customer details, creating a new account, withdrawing money, loan details and depositing money. The customer details consist of customer name, address, phone number, account number. To create a new account verifies the rules. Enter the account and then get an account number from a database. To withdraw money checks the balance in our account and then get the money. The loan details consist of loan types like home loans, car loans, education loans etc. To deposit money enter the account number and give the account to be deposited.

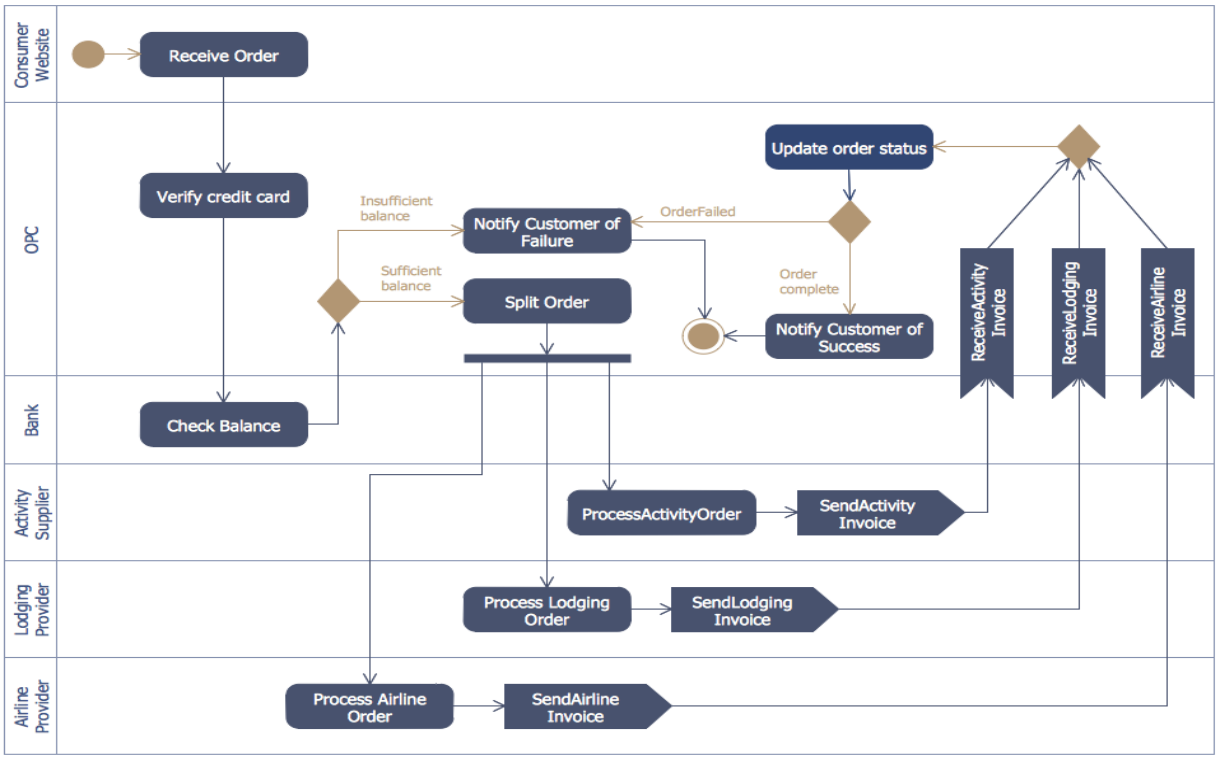
Bank ATM Use Case Diagram



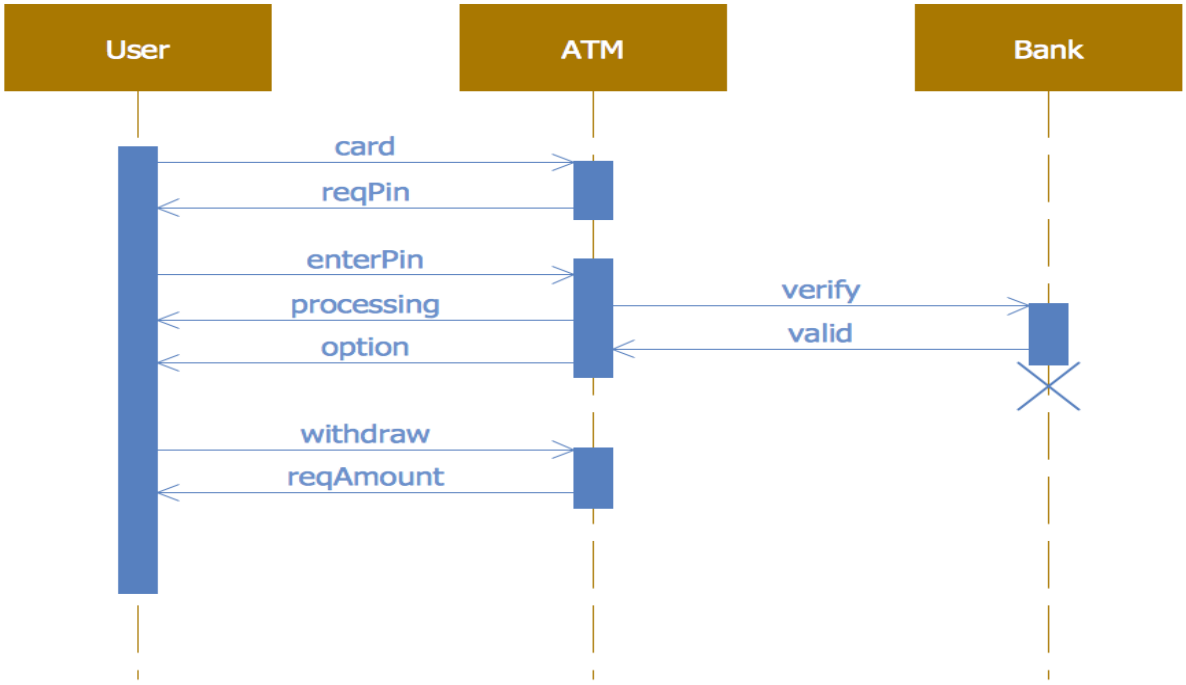
Class UML Diagram for Bank



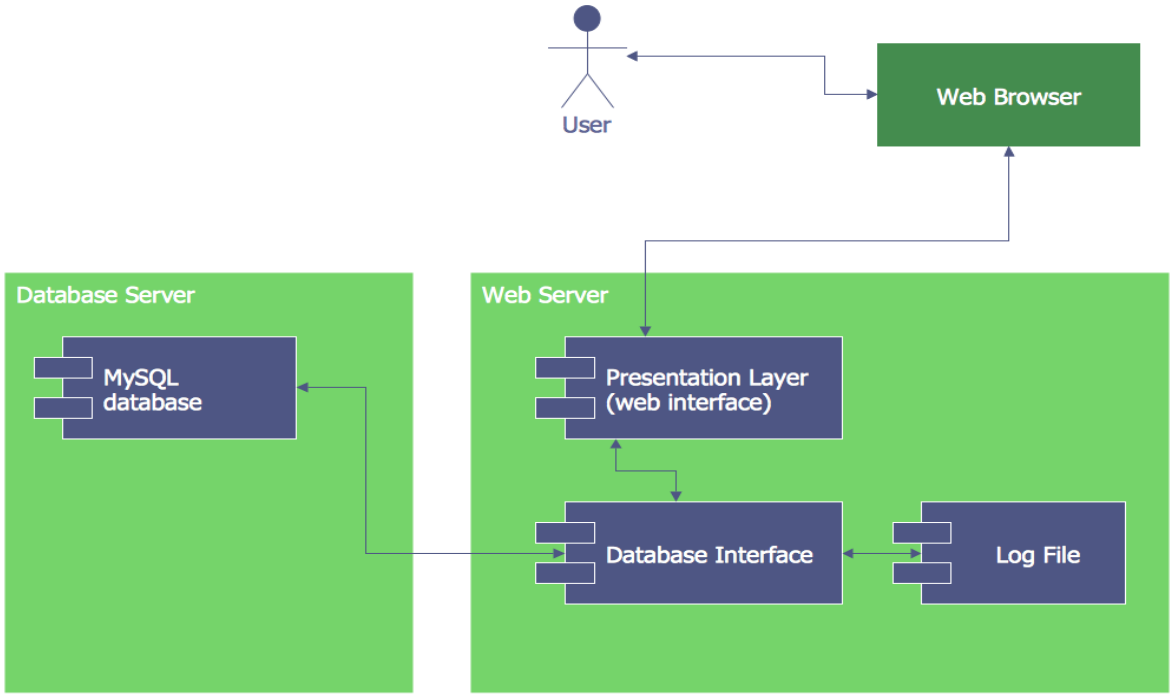
Activity diagram



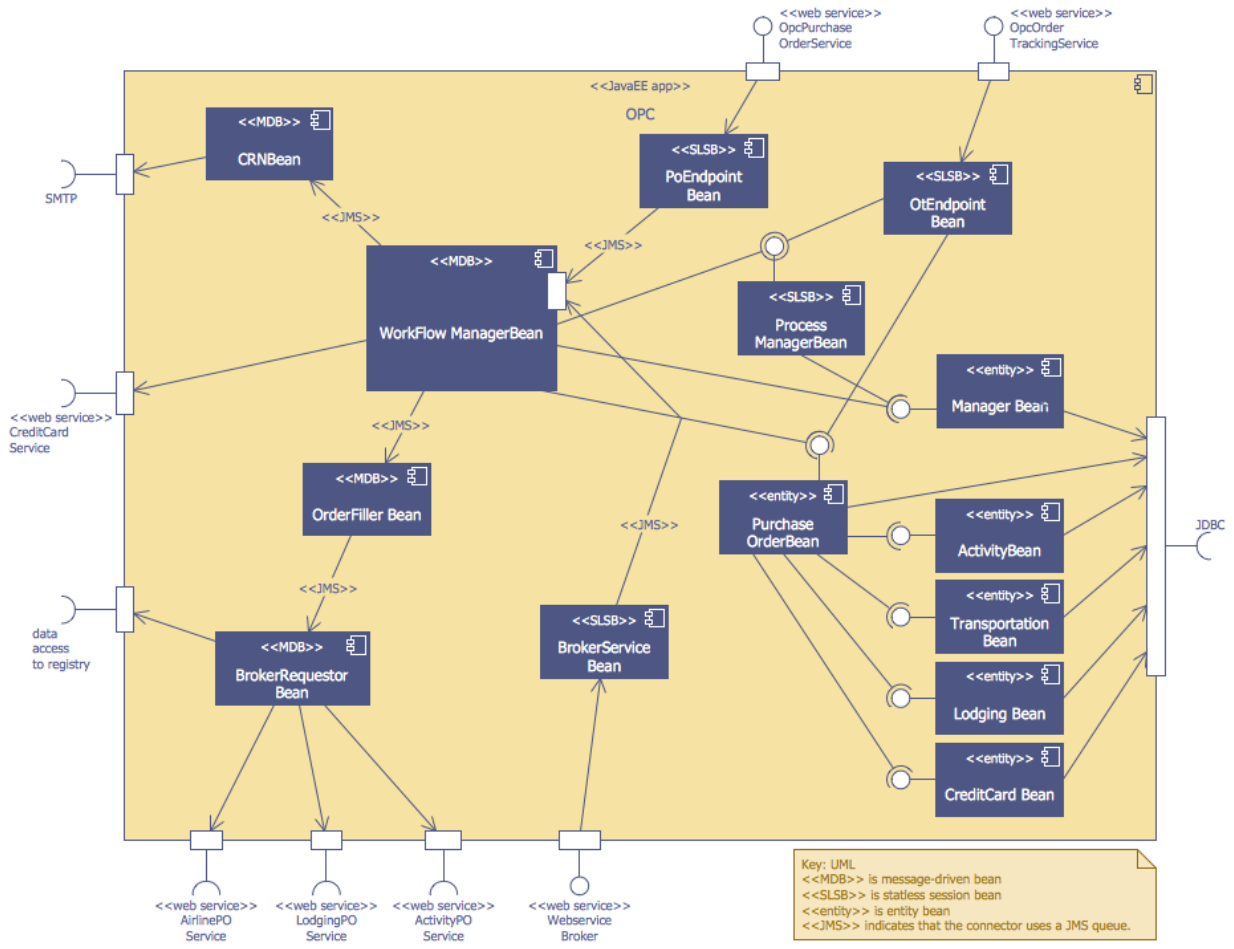
Interaction diagrams



Deployment diagram



Component diagram



Result:

Thus the UML diagram for Banking System was drawn and code was generated using ArgoUML.