**EX.NO: 1 ARGO –UML NOTATIONS**

**DATE:**

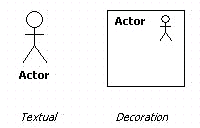
**USE CASE DIAGRAM SPECIFIC TOOLS**

Several tools are provided specific to UML model elements on use case diagrams. The detailed properties of these model elements are described in the section on use case diagram model elements

**Actor**

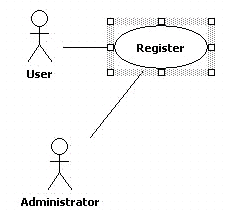
. Add an actor to the diagram. For convenience, when the mouse is over a selected actor

it displays two handles to left and right which may be dragged to form association relationships.

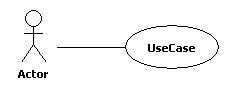


**Use Case**

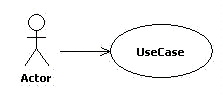
Add a use case to the diagram. For convenience, when the mouse is over a selected use case it displays two handles to left and right which may be dragged to form association relationships and two handles top and bottom which may be dragged to form generalization and specialization relationships respectively.



**Association**

Add an association between two model elements selected using button 1 motion.

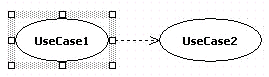
**Directed association**



**Dependency**

Add a dependency between two model elements selected using button 1 motion

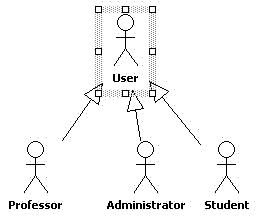
(from the dependent model element).



**Generalization**

Add a generalization between two model elements selected using button

1 motion (from the child to the parent).



**Extend**

Add an extend relationship between two model elements selected using button 1 motion (from the extended to the extending use case).

**Include**

Add an include relationship between two model elements selected using button 1

motion (from the including to the included use case).

CLASS DIAGRAM SPECIFIC TOOLS

Several tools are provided specific to UML model elements on class diagrams. The detailed properties of these model elements are described in the section on class diagram model elements

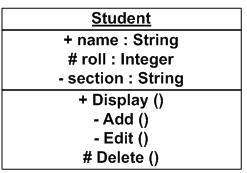
**Package**

Add a package to the diagram.

**Class**

Add a class to the diagram. For convenience, when the mouse is over a selected class it

displays two handles to left and right which may be clicked or dragged to form association relationships (or composition in case SHIFT has been pressed) and two handles top and bottom which may be dragged or clicked to form generalization and specialization relationships respectively.



**Association**

Add an association between two model elements selected using button 1 motion

**Aggregation**

Add an aggregation between two model elements selected using button 1 motion

**Composition**

Add an composition between two model elements selected using button 1 motion

**Generalization**

Add a generalization between two model elements selected using button1 motion

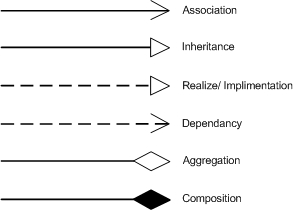
**Realization**

Add a realization between a class and an interface selected using button 1 motion

**Dependency**

Add a dependency between two model elements selected using button 1 motion

(from the dependent model element).

.

**Attribute**

Add a new attribute to the currently selected class. The attribute is given the default name

newAttr of type int and may be edited by button 1 double click and using the key board, or by selecting with button 1 click (after the class has been selected) and using the property tab.

**Operation**

Add a new operation to the currently selected class or interface. The operation is

given the default namenewOperation with no arguments and return type void and may be edited by button 1 double click and using the keyboard, or by selecting with button 1 click (after the class has been selected) and using the property tab.

**Association Class**

Add a new association class between two model elements selected using button 1 motion (from the first model element to the second).

SEQUENCE DIAGRAM SPECIFIC TOOLS

Seven tools are provided specific to UML model elements on sequence diagrams. The detailed properties of these model elements are described in the section on sequence diagram model elements.

**ClassifierRole**

Add a classifierrole to the diagram.

**Message with Call Action**

Add a call message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).

**Message with Return Action**

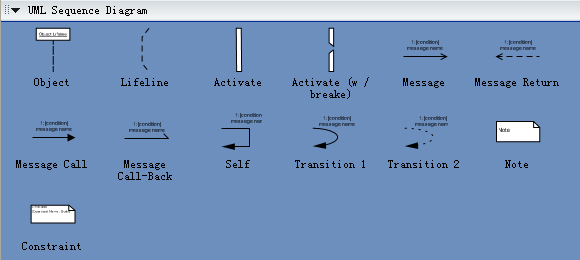
Add a return message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).

**Message with Create Action**

Add a create message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).

**Message with Destroy Action**

Add a destroy message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).



STATE CHART DIAGRAM SPECIFIC TOOLS

Eleven tools are provided specific to UML model elements on statechart diagrams. The detailed properties of these model elements are described in the section on statechart diagram model elements

**Simple State**

Add a simple state to the diagram.

**Composite State**

Add a composite state to the diagram. All model elements that are subsequently placed on the diagram on top of the composite state will form part of that composite state.

**Transition**

Add a transition between two states selected using button 1 motion (from the originating state to the receiving state).

**Synch State**

Add a synchstate to the diagram.

**Submachine State**

Add a submachinestate to the diagram.

**Stub State**

Add a stubstate to the diagram.

**Initial**

Add an initial pseudostate to the diagram.

**Final State**

Add a final state to the diagram.

**Junction**

Add a junction pseudostate to the diagram.

**Choice**

.Add a choice pseudostate to the diagram.

**Fork**

Add a fork pseudostate to the diagram.

**Join**

Add a join pseudostate to the diagram.

**Shallow History**

Add a shallow history pseudostate to the diagram.

**Deep History**

Add a deep history pseudostate to the diagram.

**Call Event**

Add a Call Event as trigger to a transition. There are 4 types of events offered

here:Call Event,Change Event,Signal Event and Time Event

**Guard**

Add a guard to a transition.

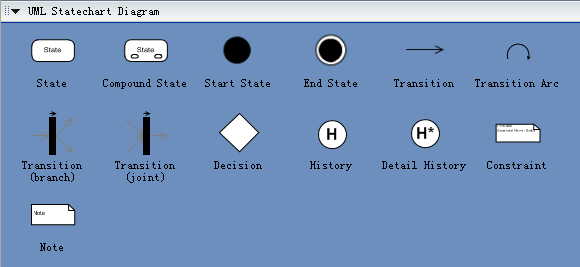
**Call Action**

. Add a call action (i.e. the effect) to a transition. There are 7 types of actions

offered here:

Call Action,Create Action,Destroy Action, Return Action,SendAction,Terminate Action,Uninterpreted Action and

Action Sequence



ACTIVITY DIAGRAM SPECIFIC TOOLS

**Action State**

Add an action state to the diagram.

**Transition**

Add a transition between two action states selected using button 1 motion

(from the originating action state to the receiving action state).

**Initial**

. Add an initial pseudostate to the diagram.

**Final State**

Add a final state to the diagram.

**Junction**

Add a junction (decision) pseudostate to the diagram.

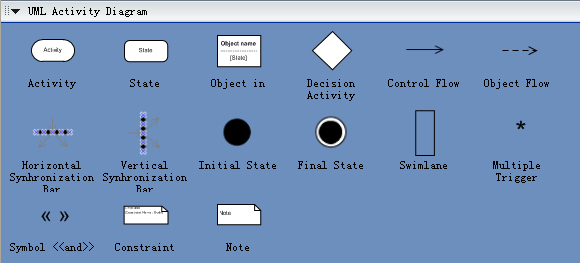
ArgoUML does not enforce this, but an ArgoUML critic will complain about any junction does not follow this rule.

**Fork**

Add a fork pseudostate to the diagram.

**Join**

Add a join pseudostate to the diagram.



**EXNO: 2 PASPORT AUTOMATION SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Passport Automation System is an interface between the Applicant and the

Authority responsible for the Issue of Passport. It aims at improving the efficiency

in the Issue of Passport and reduces the complexities involved in it to the

maximum possible extent

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The System provides an online interface to the user where they can fill in

their personal details. The authority concerned with the issue of passport can

use this system to reduce his workload and process the application in a

speedy manner.Provide a communication platform between the applicant

and the administrator Transfer of data between the Passport Issuing

Authority and the Local Police for verification of applicant's information.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate the workload and the process involved in the issue of passport.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

|  |  |
| --- | --- |
| Applying passport | Applicant must give the details of them for applying passport |
| Verification | Checking the details given by the applicant |
| Checking | The officers should the check personal details of applicant. |

**3. ARGO UML DIAGRAMS:**

**3.1 USECASE DIAGRAM:-**

****

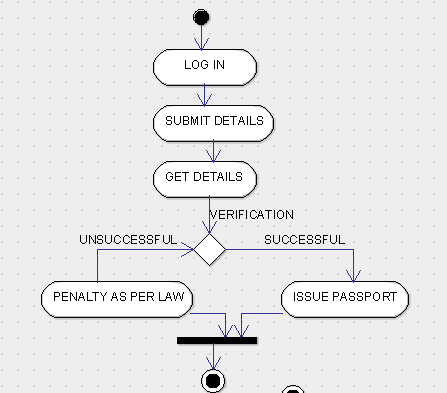
**3.2 SEQUENCE DIAGRAM:**



**3.3 CLASS DIAGRAM:**

****

**3.4 ACTITVITY DIAGRAM:**



**3.5 STATE CHART DIAGRAM:**



**Implementation**

To close a form

this.Close();

To exit an application

Application.Exit();

To go to a new form (Form2 is the name of the target form. It may vary for your project)

Form2 myForm = new Form2();

myForm.Show();

To get the value from textbox (textBox1 is the name of the target textbox. It may vary for your project)

this.textBox1.Text

To display a prompt

MessageBox.Show("HELLO");

For a login page

if (this.textBox1.Text.Equals("myadmin") && this.textBox2.Text.Equals("mypass"))

{

Form4 myForm = new Form4();

myForm.Show();

this.Close();

}

else

{

MessageBox.Show("Wrong credentials", "Error");

}

Database connectivity

---------------------

add "using MySql.Data.MySqlClient;" after choosing MySql.Data from references

This is the general code. Change the query to perform different operations.

This code must be inside button\_Click(object sender, EventArgs e)

NOTE:passport is the database name and details is the table name

try

{

//This is my connection string i have assigned the database file address path

string MyConnection2 = "datasource=localhost;port=3306;username=root;password=";

//This is my insert query in which i am taking input from the user through windows forms

string Query = "insert into passport.details(NAME,FATHERNAME,DOB,ADDRESS,PHONE,EMAIL,PAN) values('" +this.textBox1.Text+ "','" +this.textBox2.Text+ "','" +this.textBox3.Text+ "','" +this.textBox4.Text+ "'," +this.textBox5.Text+ ", ' " +this.textBox6.Text + "' , "+this.textBox7.Text+ ");";

//This is MySqlConnection here i have created the object and pass my connection string.

MySqlConnection MyConn2 = new MySqlConnection(MyConnection2);

//This is command class which will handle the query and connection object.

MySqlCommand MyCommand2 = new MySqlCommand(Query, MyConn2);

MySqlDataReader MyReader2;

MyConn2.Open();

MyReader2 = MyCommand2.ExecuteReader(); // Here our query will be executed and data saved into the database.

MessageBox.Show("DATA SUBMITTED");

while (MyReader2.Read())

{

}

MyConn2.Close();

}

catch (Exception ex)

{

//Will be executed if there are some errors

MessageBox.Show("ERROR");

}

To display some data fetched from the database drag a "DataGridView" from toolbox and after clicking a display button,

try

{

string MyConnection2 = "datasource=localhost;port=3306;username=root;password=";

//Display query

string Query = "select \* from passport.details;";

MySqlConnection MyConn2 = new MySqlConnection(MyConnection2);

MySqlCommand MyCommand2 = new MySqlCommand(Query, MyConn2);

// MyConn2.Open();

//For offline connection we weill use MySqlDataAdapter class.

MySqlDataAdapter MyAdapter = new MySqlDataAdapter();

MyAdapter.SelectCommand = MyCommand2;

DataTable dTable = new DataTable();

MyAdapter.Fill(dTable);

dataGridView1.DataSource = dTable; // here i have assign dTable object to the dataGridView1 object to display data.

// MyConn2.Close();

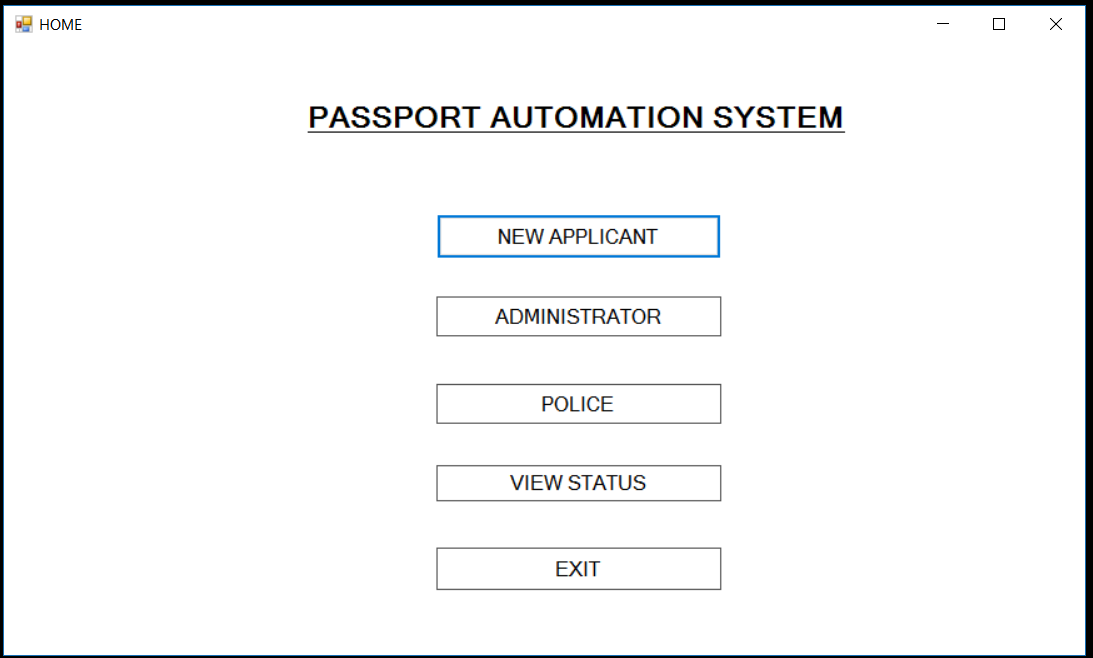
}

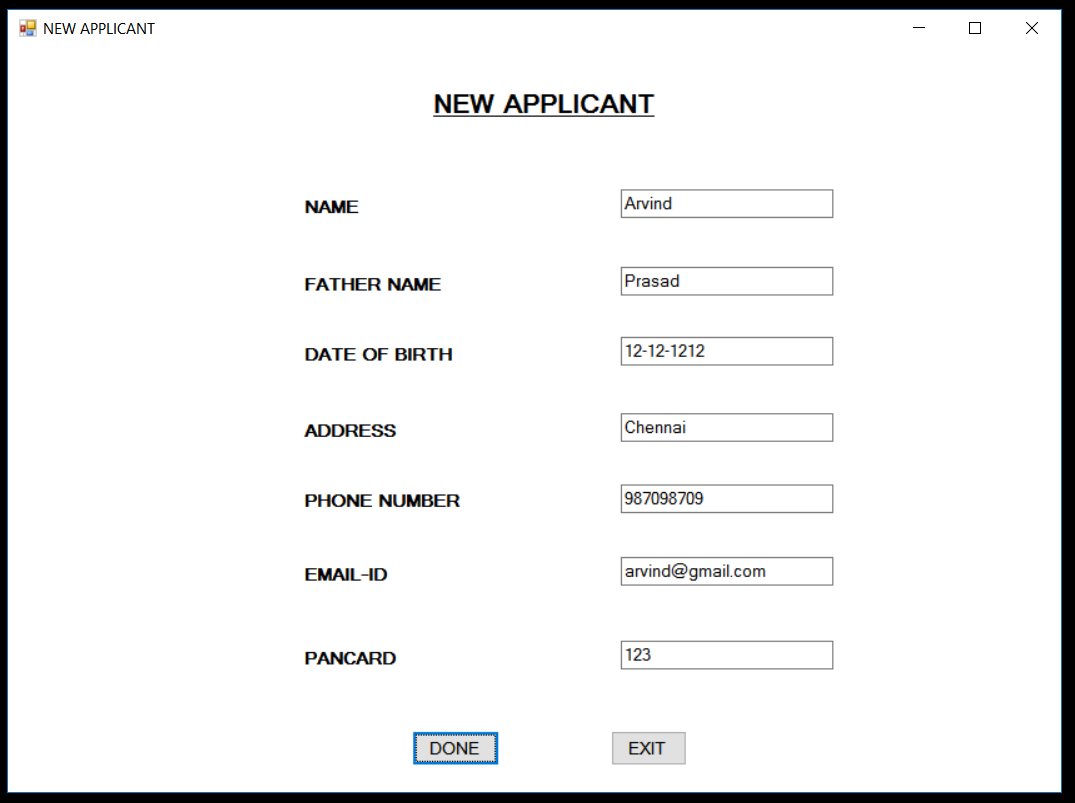
catch (Exception ex)

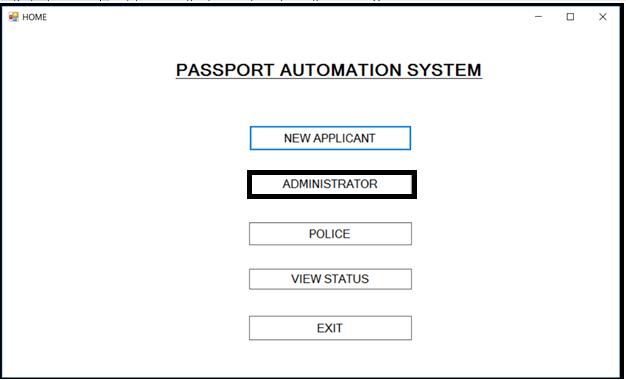
{

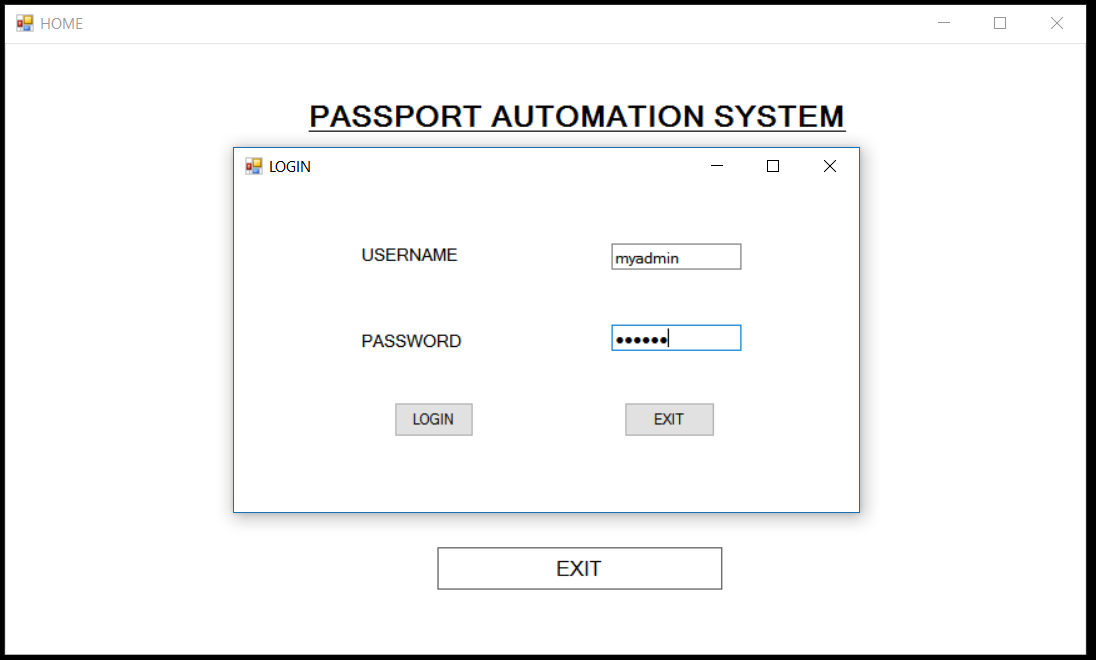
MessageBox.Show(ex.Message);

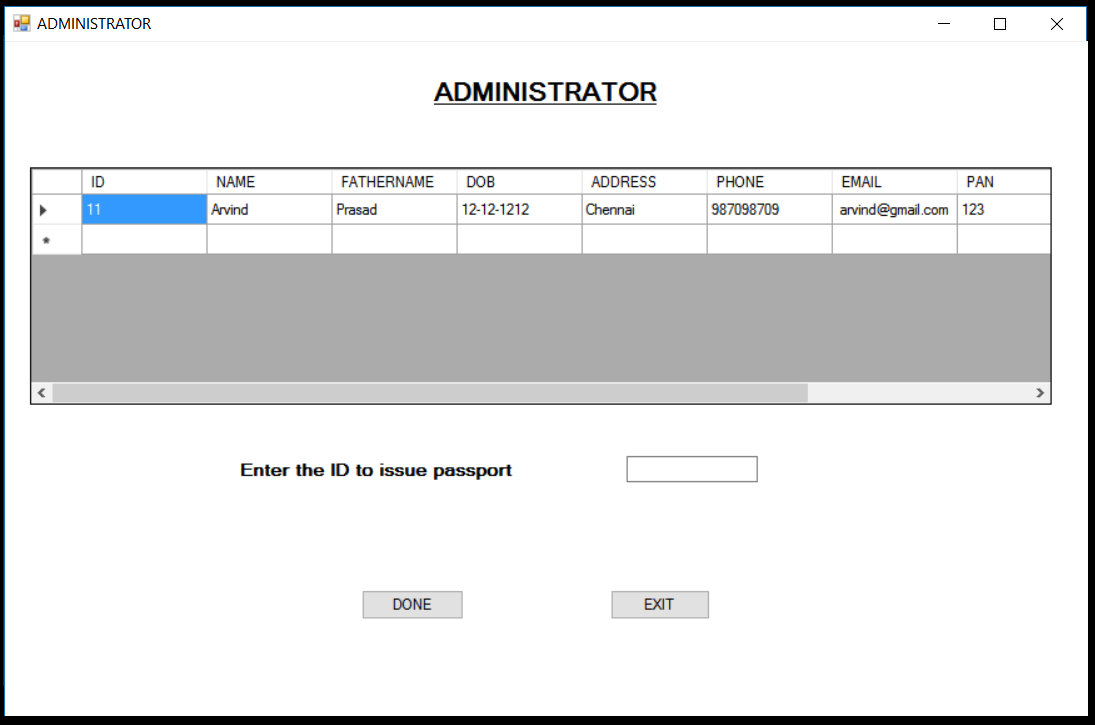
}

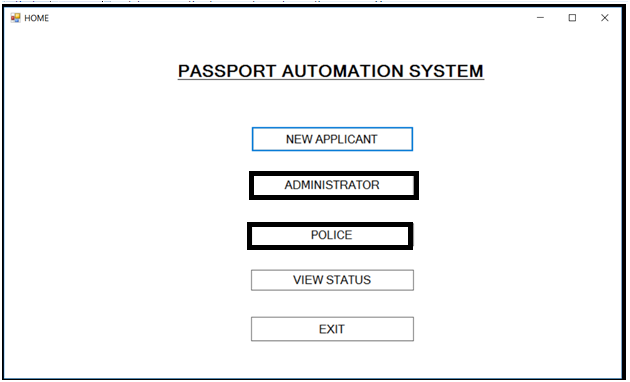


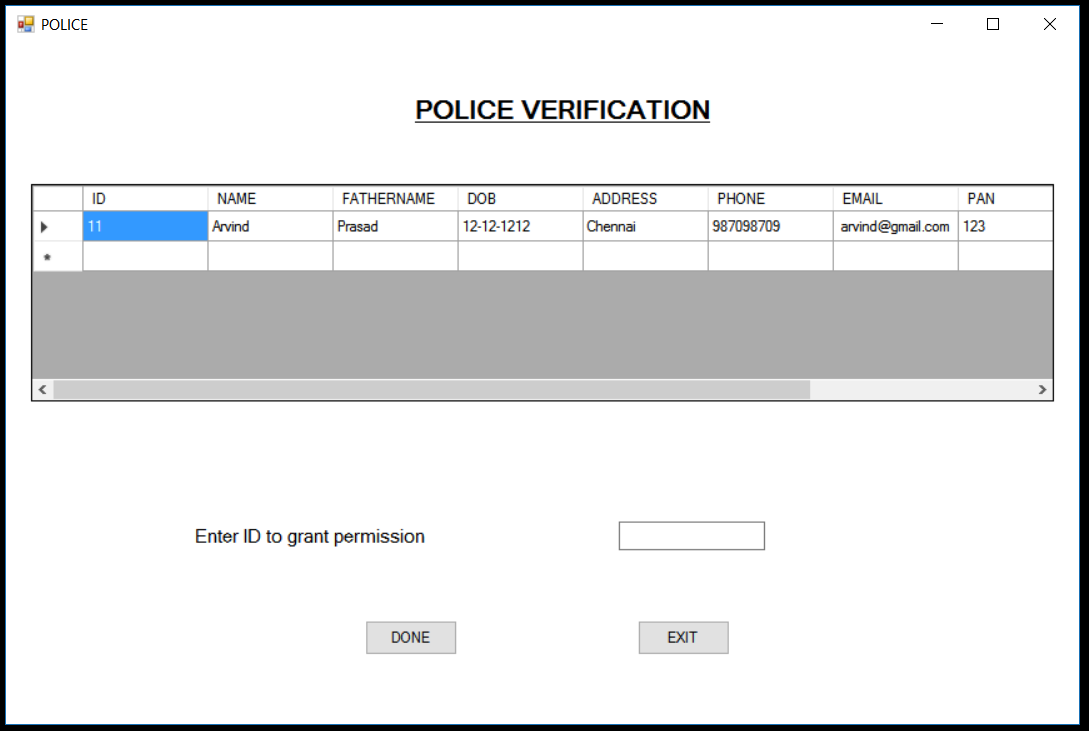


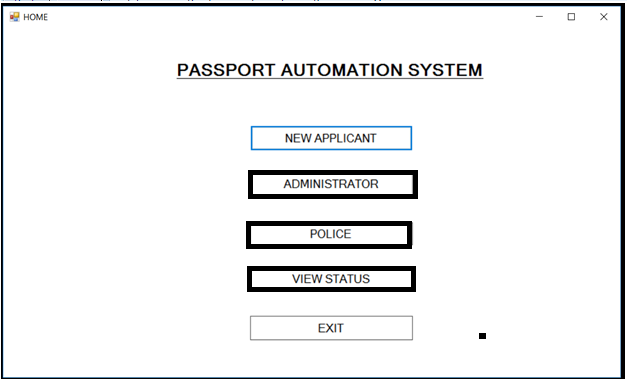


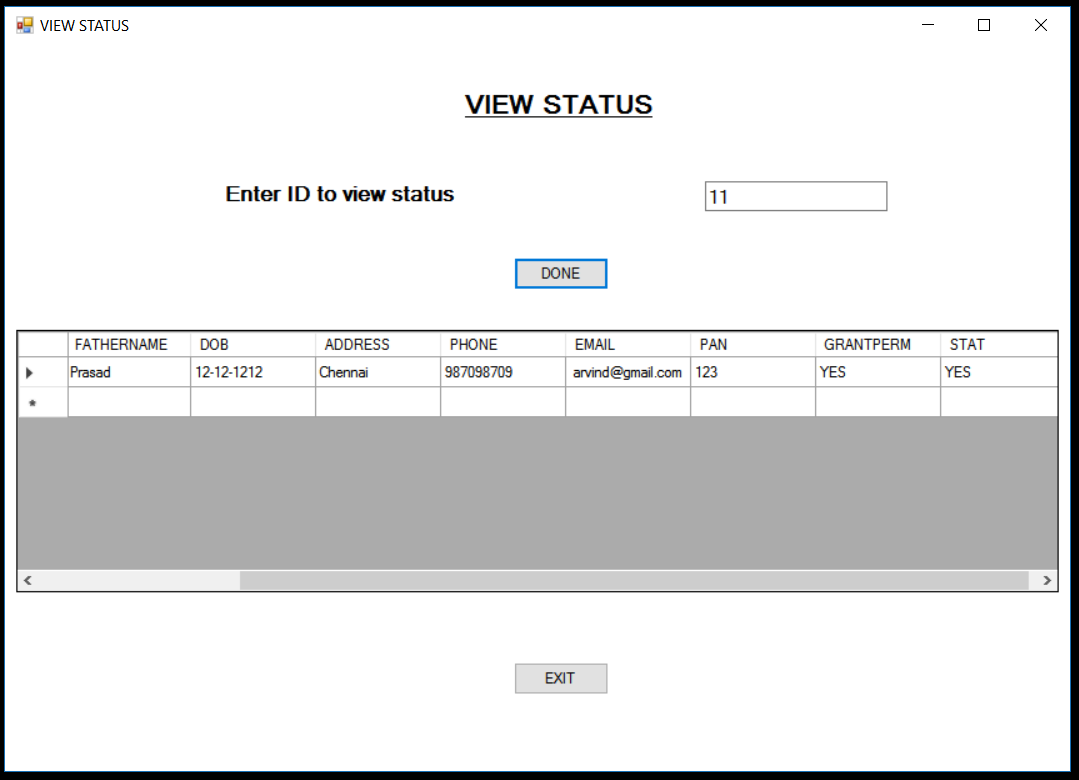


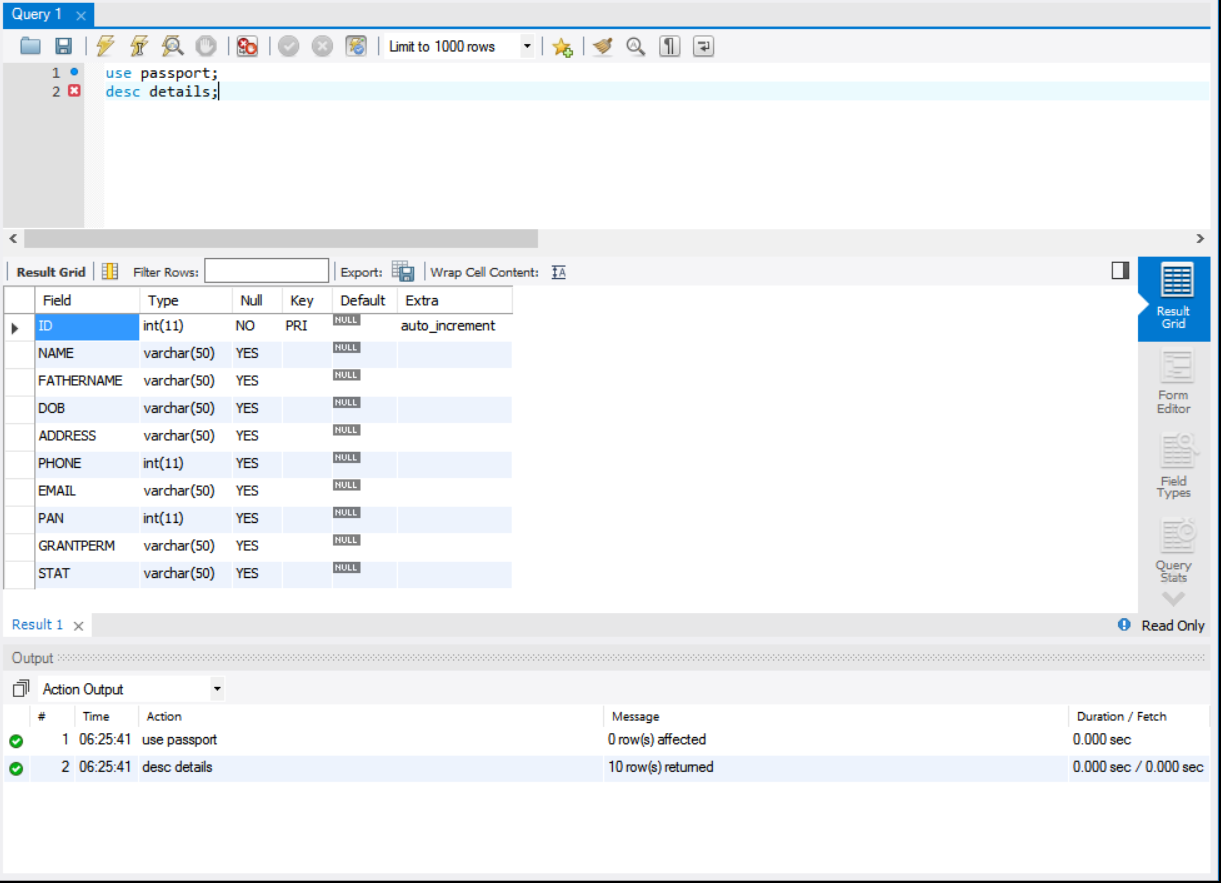












**RESULT:** Thus passport automation system has been implemented successfully.

**EXNO: 3 BOOK BANK SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

The bookbank system is used to issue books to students in a ordered manner.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a book bank system. Which seems to be useful to students.the students should register their id to the admin.and they order books at the respective timings.and the books has dispatched to students in a respective time.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

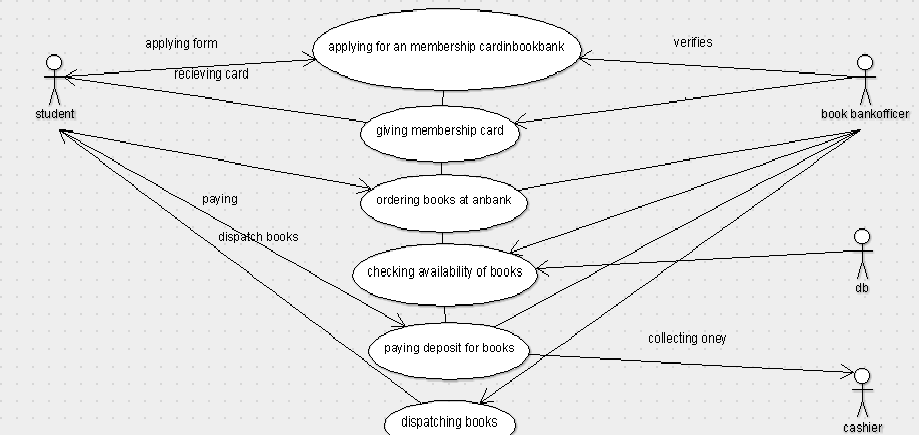
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

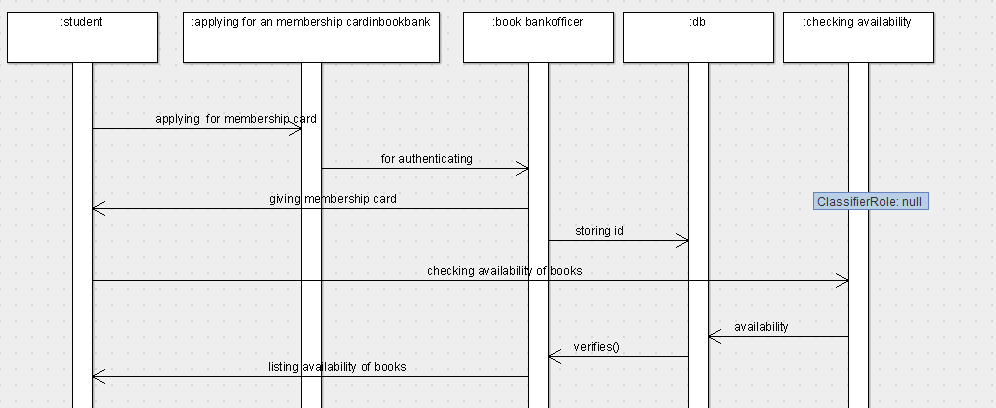
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the books in the bookbank. |
| Select | Select the books if available |

3. ARGO-UML DIAGRAMS:

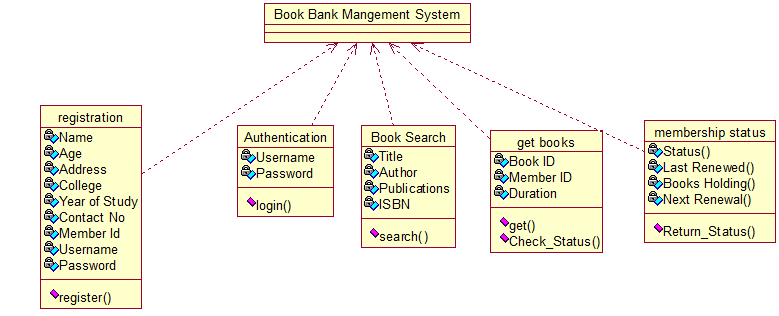
3.1 USECASE DIAGRAM:



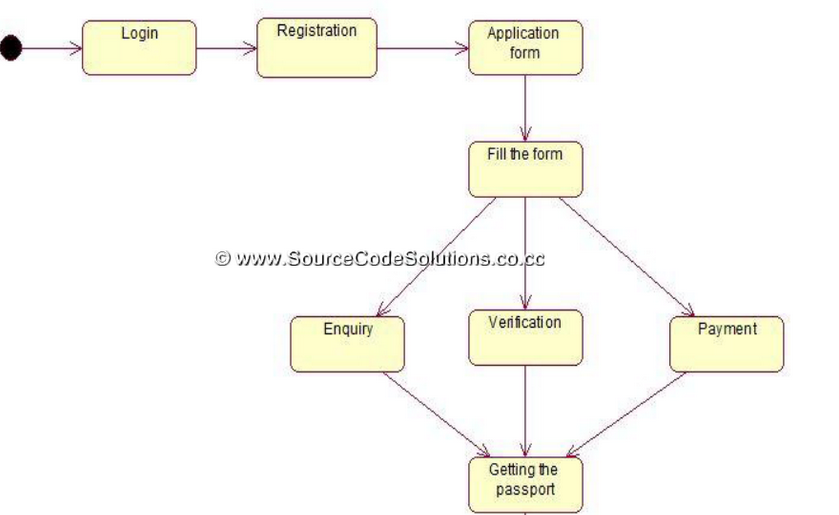
**3.2 SEQUENCE DIAGRAM.**

****

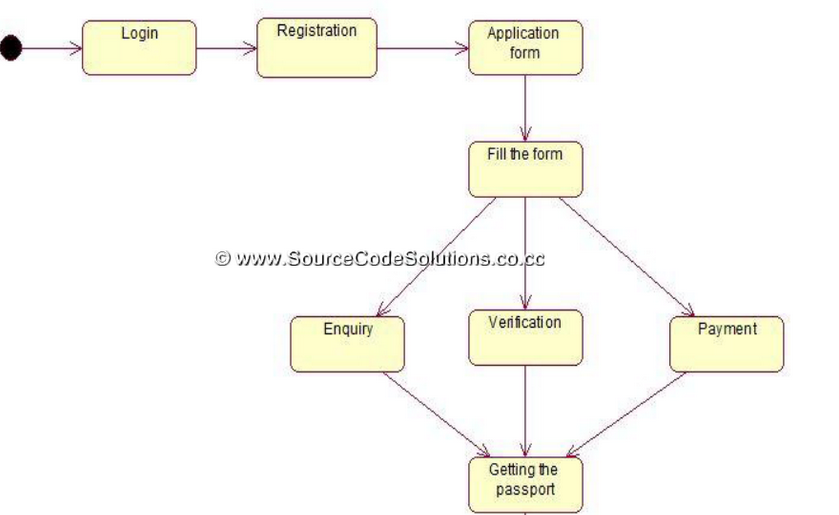
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**



**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus book bank system has been implemented successfully.

**EXNO: 4 EXAM REGISTRATION SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Online EXAM reservation system is used to choose the exam through online by the students. They are provided with a catalog they can choose the exam. The catalog contains the detailed description about each exam and to decide on their own.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a exam Registration System, to register the required exam using registration model, if the candidate want to cancel the exam using cancellation model.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

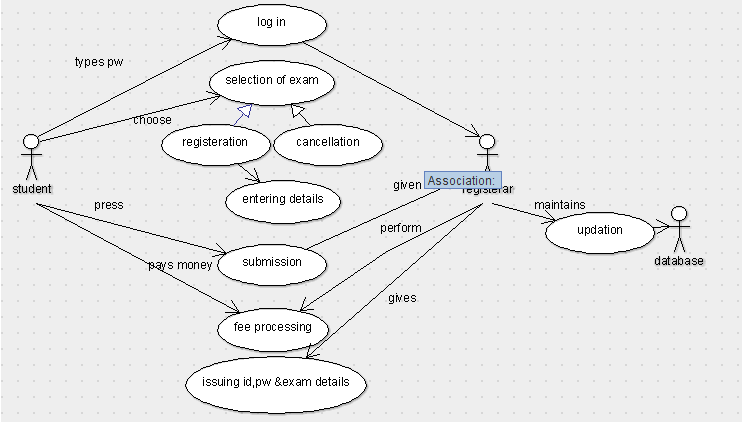
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

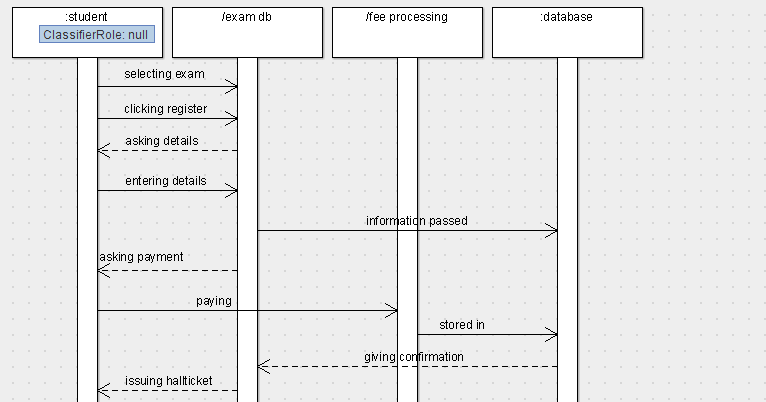
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| VERIFICATION | Check the student details for the exam. |
| Select | Select the eligible students. |

**3. ARGO UML DIAGRAMS:**

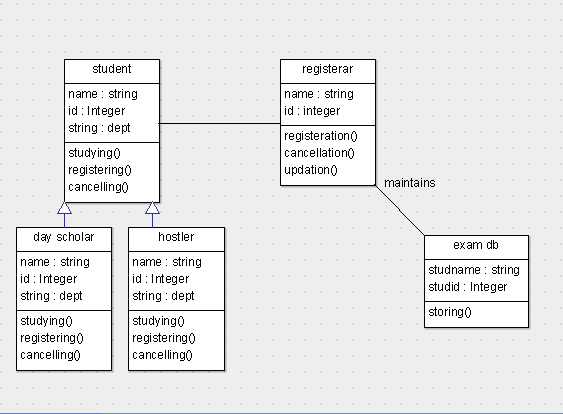
**3.1 USECASE DIAGRAM:**

****

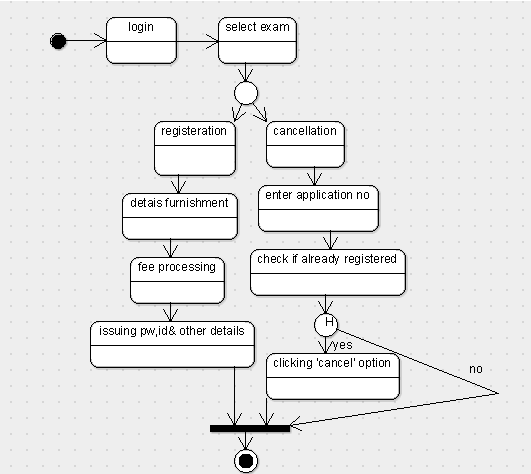
**3.2 SEQUENCE DIAGRAM:**

****

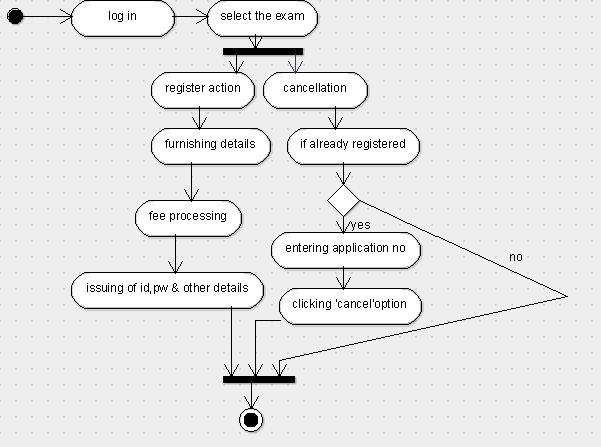
**3.3 CLASS DIAGRAM:**



**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus exam registration system has been implemented successfully.

**EXNO: 5 STOCK MAINTENANCE SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

The purpose of “stock maintenance system” is to maintain the product and its stock details. It creates software which provides automation for indication of product details to the owner. The main categories of users provided are

Stock-holder: one who monitor the software to check stock availability

Maintenance system: system that maintains stock details.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

Stock maintenance system maintains a database where product details like product code, product name and prices are stored and updated. It enables a fully automated system to provide easier way to indicate the availability of product to the stock-holder.

**2.2 PROJECT OBJECTIVE:**

The project objective will be focused on developing a system that contains essential details about the product and its sales details in a database which makes it easier to view the stock details between two dates.

**2.3 SOFTWARE REQUIREMENTS:**

Software: ARGO UML ,visual basic

Operating system: windows

Database: oracle

**2.4 HARDWARE REQUIREMENTS:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

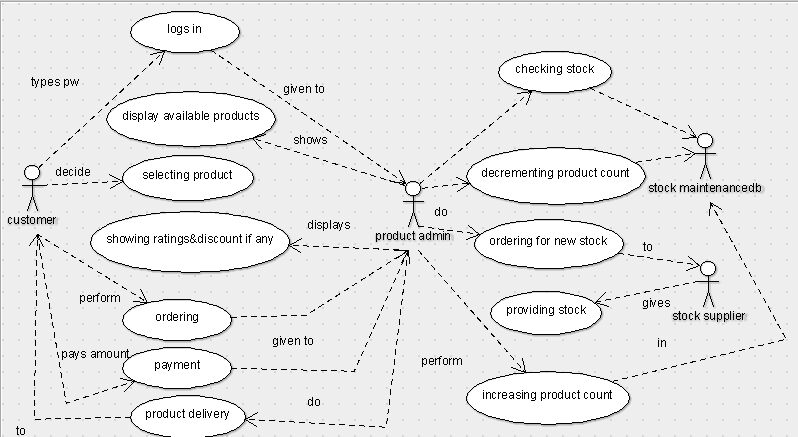
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUCTIONAL REQUIREMENTS:**

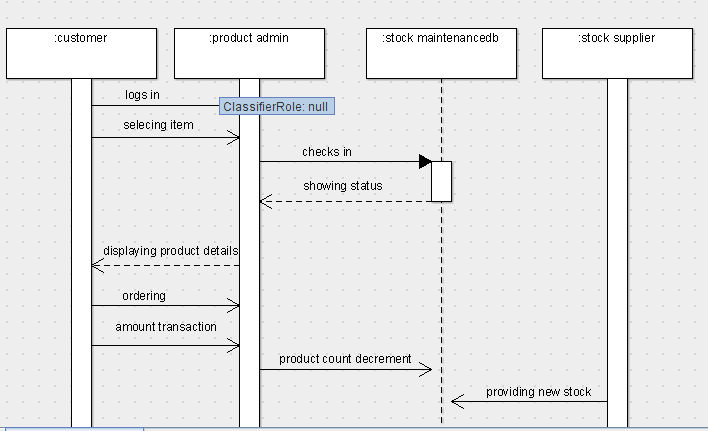
|  |  |
| --- | --- |
| Ordering | Customer orders a product from a stockholder |
| Checking stock | Stock holder verifies stock database for the ordered item |
| Replying | Stockholder replies to customer about the product availability |
| Amount transaction | Customer pays the amount to stock holder |
| Updating | Stockholder updates the product details in the database |

**3. ARGO UML DIAGRAMS:**

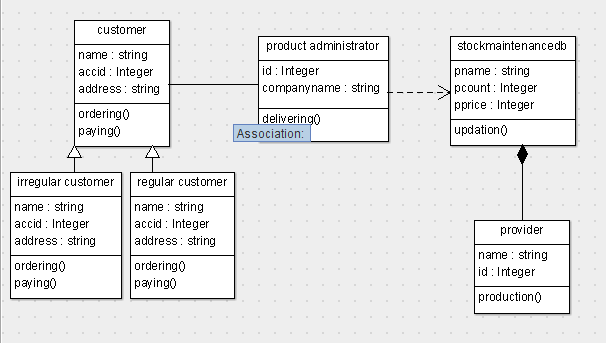
**3.1 USECASE DIAGRAMS:**

****

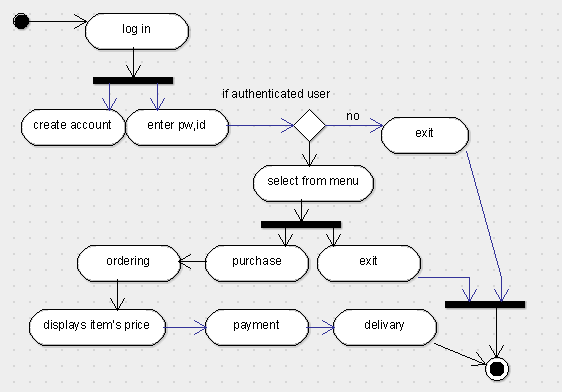
**3.2 SEQUENCE DIAGRAM:**

****

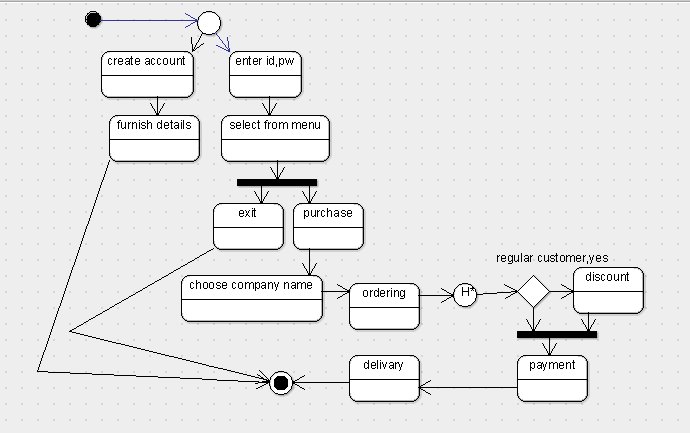
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus stock maintenance system has been implemented successfully.

**EXNO: 6 ONLINE COURSE REGISTRATION SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Online course reservation system is used to choose the course through online by the students. They are provided with a catalog they can choose the course. The catalog contains the detailed description about each course and also the availability of the course which helps the students to decide on their own.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

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

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

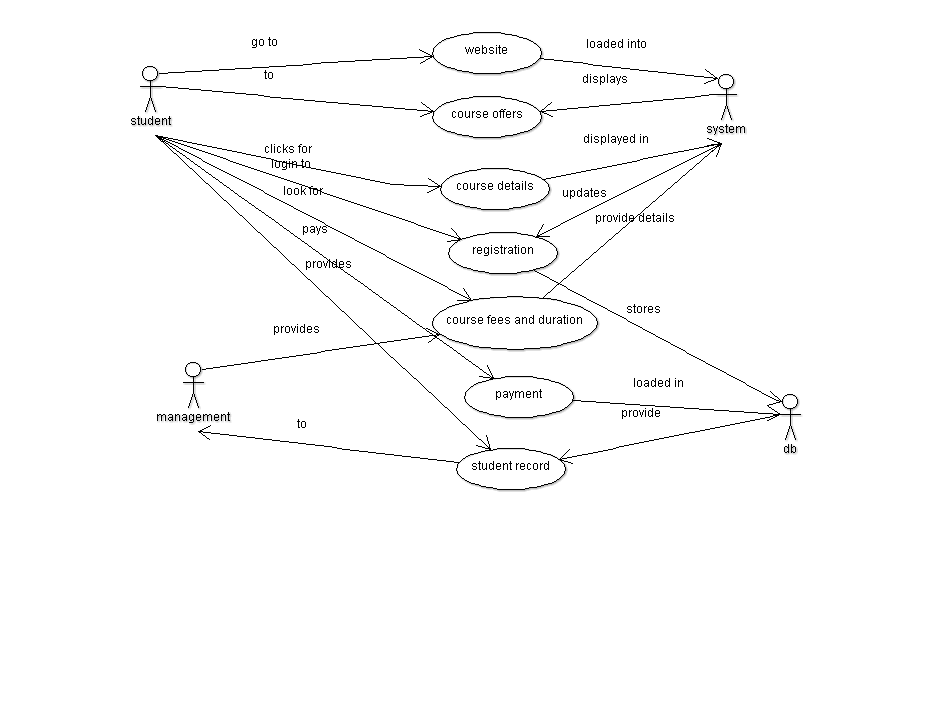
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

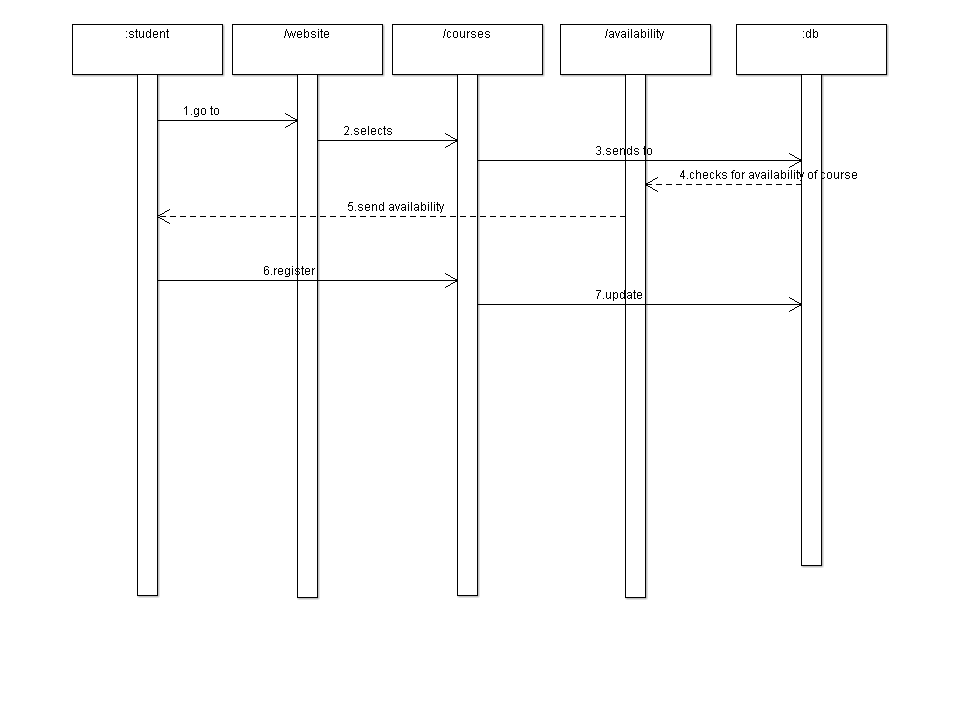
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the course in the college |
| Select | Select the course if available |

**3. ARGO UML DIAGRAMS:**

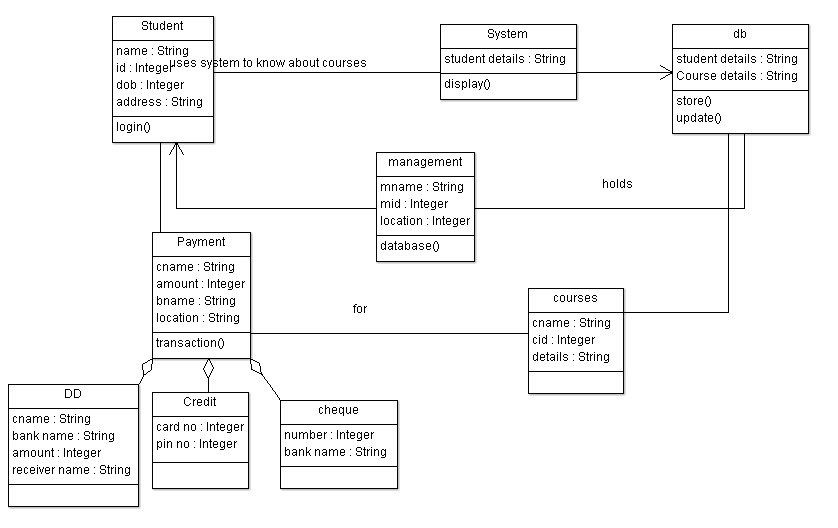
**3.1 USECASE DIAGRAM:**

****

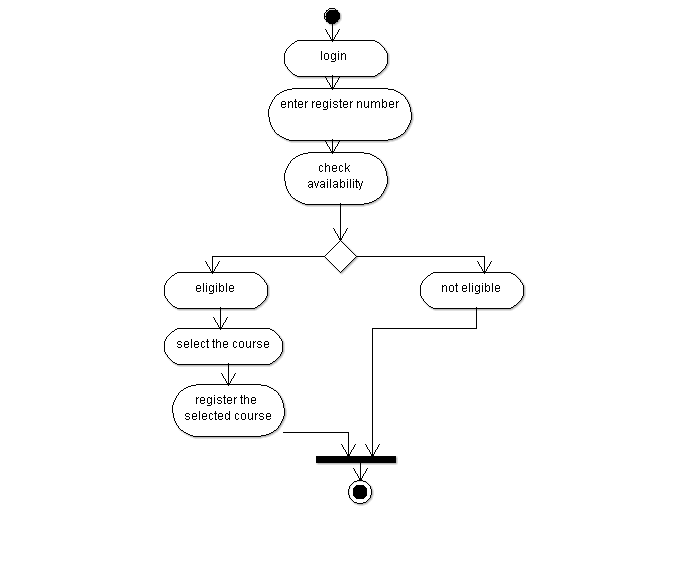
**3.2 SEQUENCE DIAGRAM:**

****

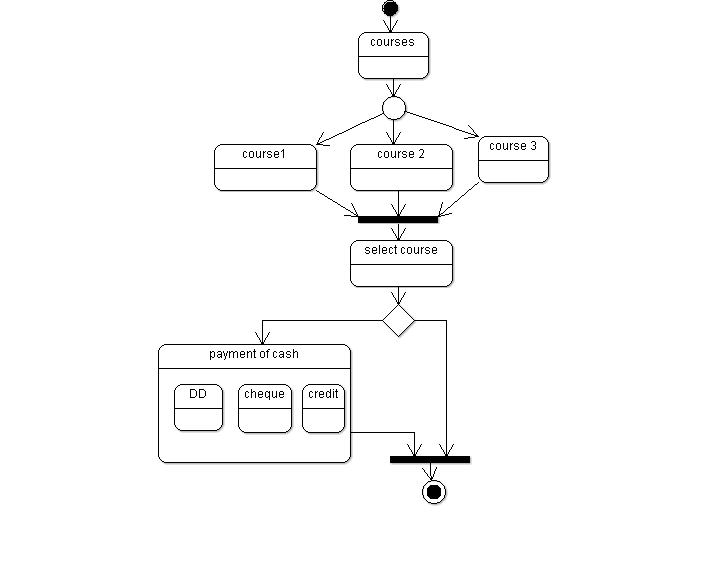
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:** Thus online course registration system has been implemented successfully.

**EXNO: 7 E-TICKETING SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

E-ticketing system is used to reserve the ticket in various fields using the software with argo uml.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a e-ticketing System, to enquire the available tickets using enquiry model and then register the required tickets using registration model, if the candidate want to cancel the tickets using cancellation model, if the candidate wants to modify or change the tickets, a modification model has been developed.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the candidate overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

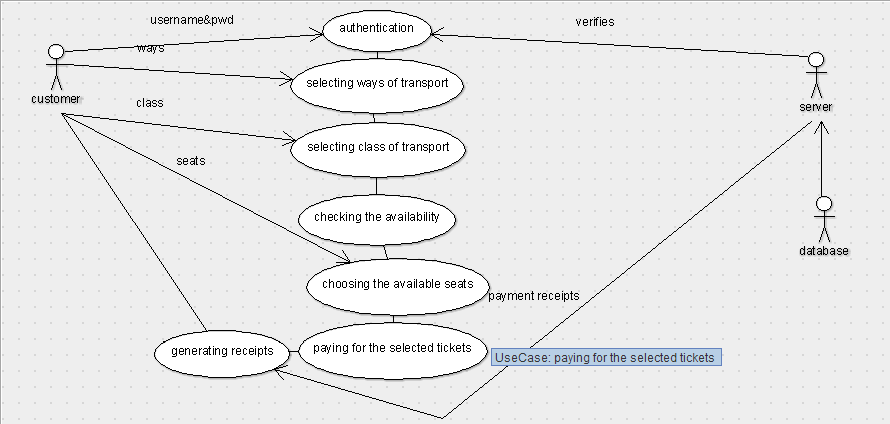
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

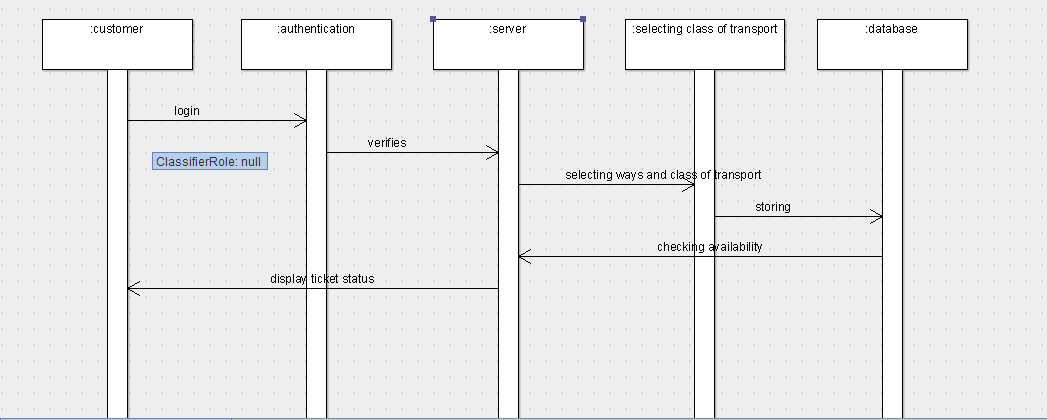
|  |  |
| --- | --- |
| Add details | Add the detail of the candidate to the database |
| Availability | Check the availability of the tickets in the system. |
| Select | Select the tickets if available |

**3. ARGO-UML DIAGRAMS:**

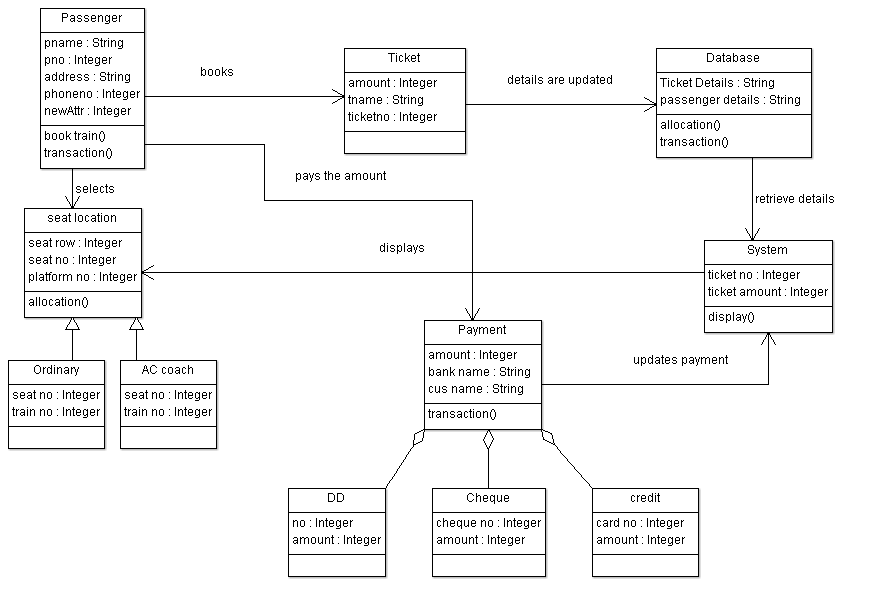
**3.1 USECASE DIAGRAM:**

****

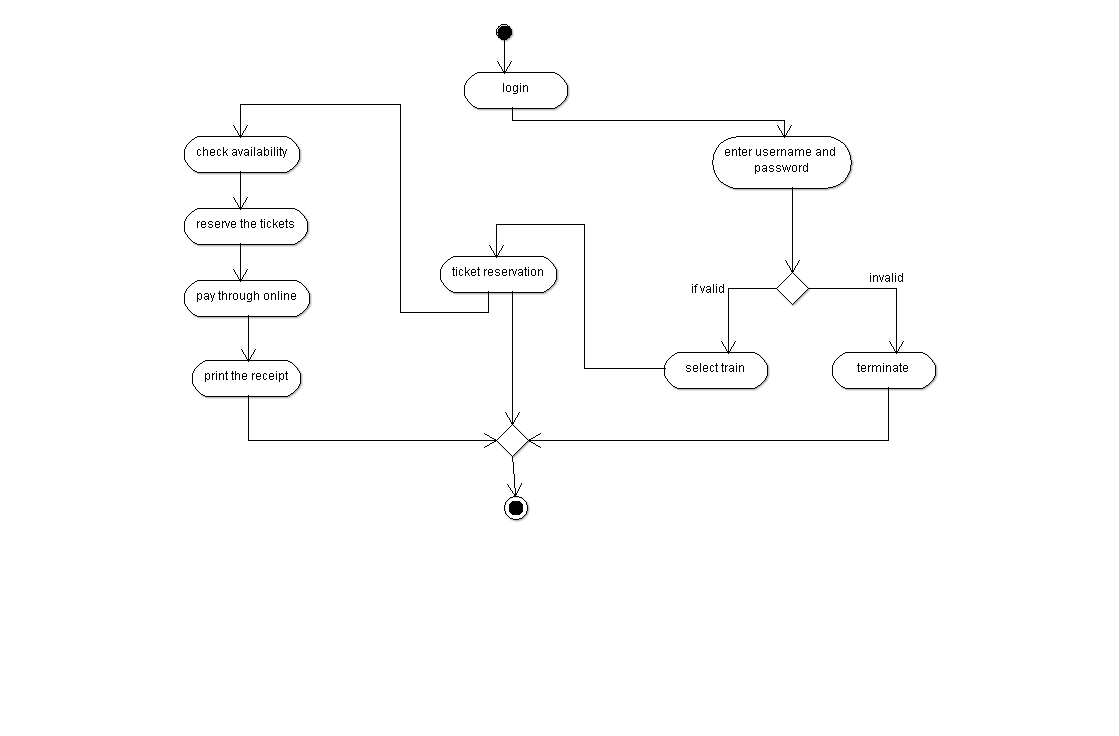
**3.2 SEQUENCE DIAGRAM:**

****

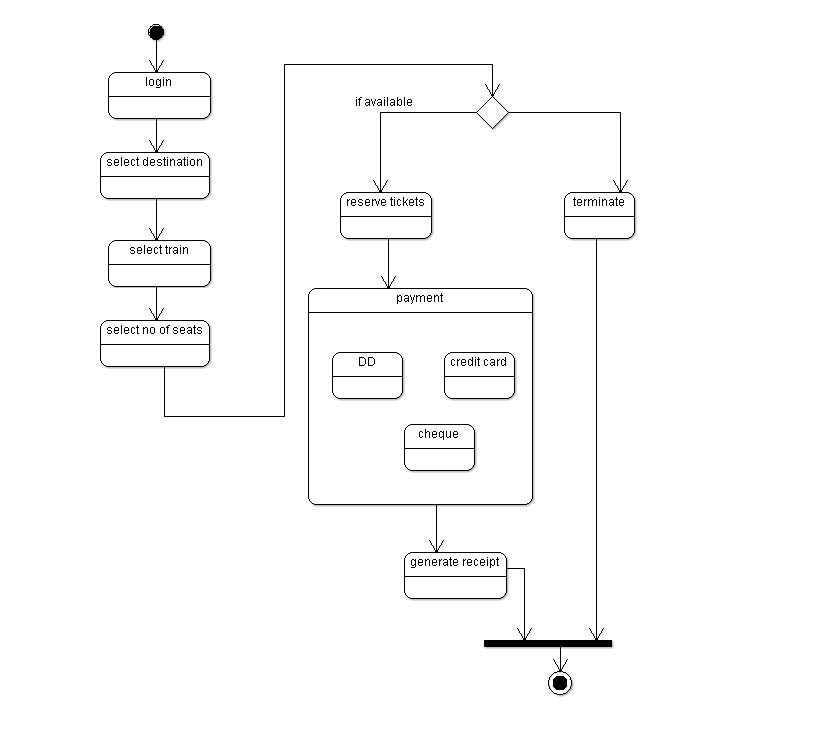
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus e-ticketing system has been implemented successfully.

**EXNO: 8 SOFTWARE PERSONNEL MANAGEMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

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

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: Star UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

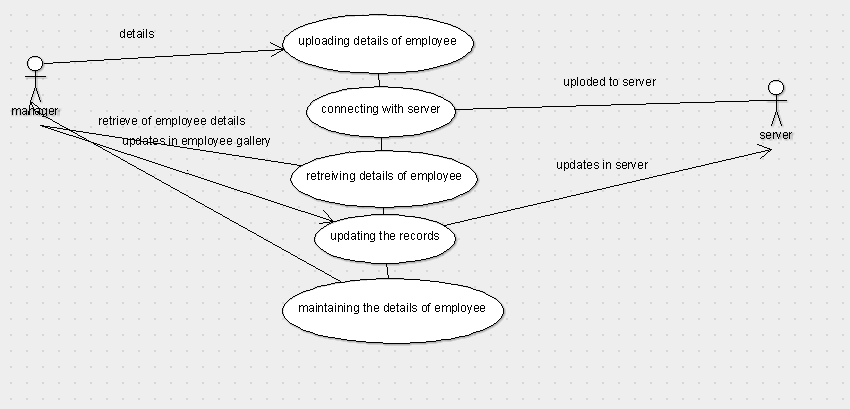
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

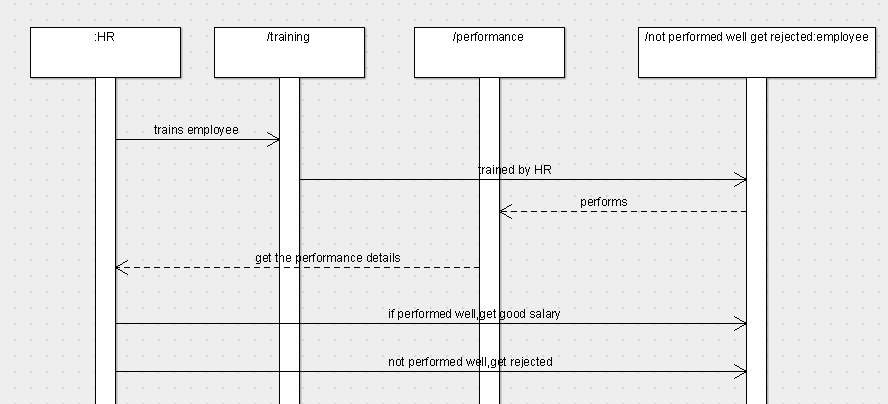
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the course in the college |
| Select | Select the course if available |

**3. ARGO UML DIAGRAMS:**

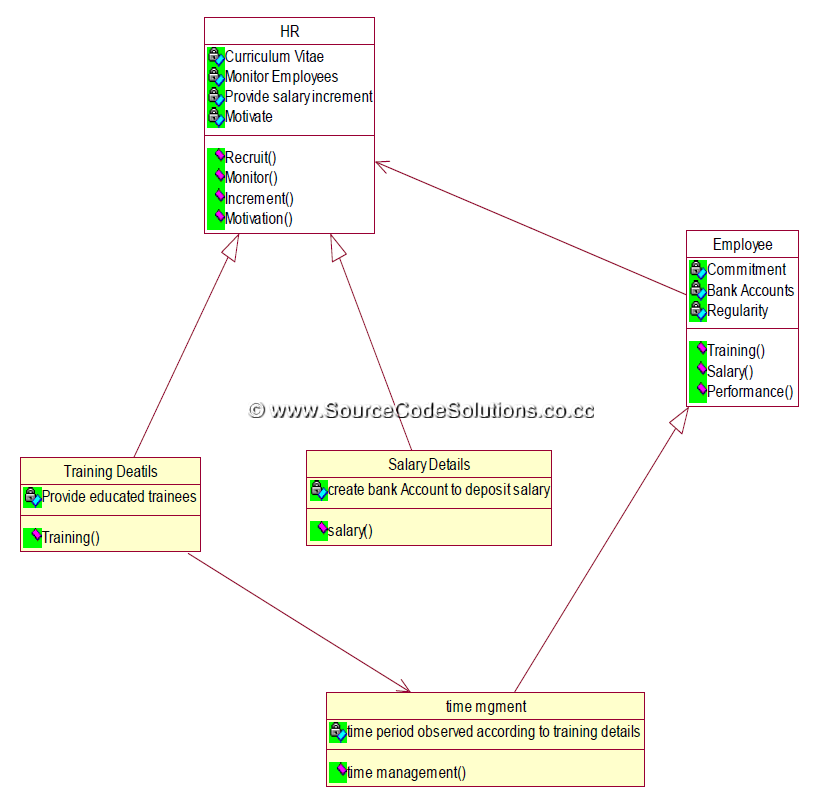
**3.1 USECASE DIAGRAMS:**

****

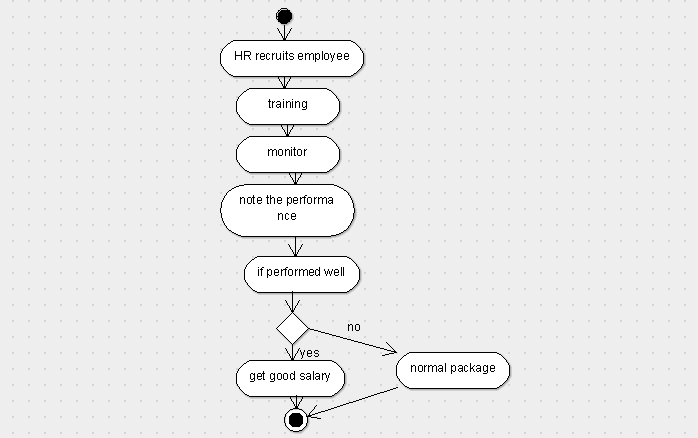
**3.2 SEQUENCE DIAGRAMS:**

****

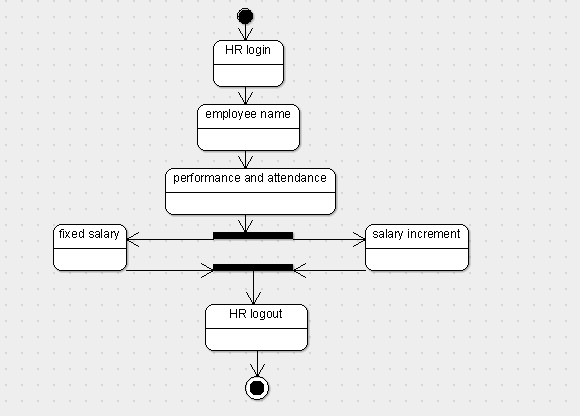
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus software personnel management system has been implemented successfully.

**EXNO: 9 CREDIT CARD SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Credit card system is used to choose the course through online by the students. They are provided with a catalog they can choose the course. The catalog contains the detailed description about each course and also the availability of the course which helps the students to decide on their own.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

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

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: Star UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

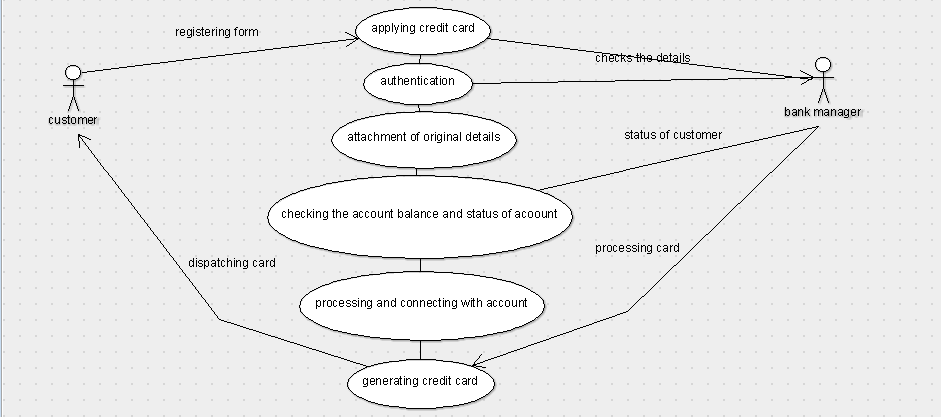
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

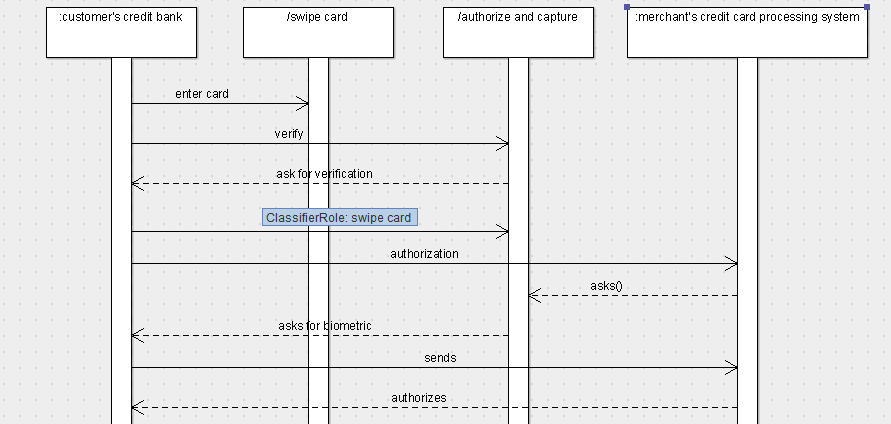
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the course in the college |
| Select | Select the course if available |

**3. ARGO UML DIAGRAMS:**

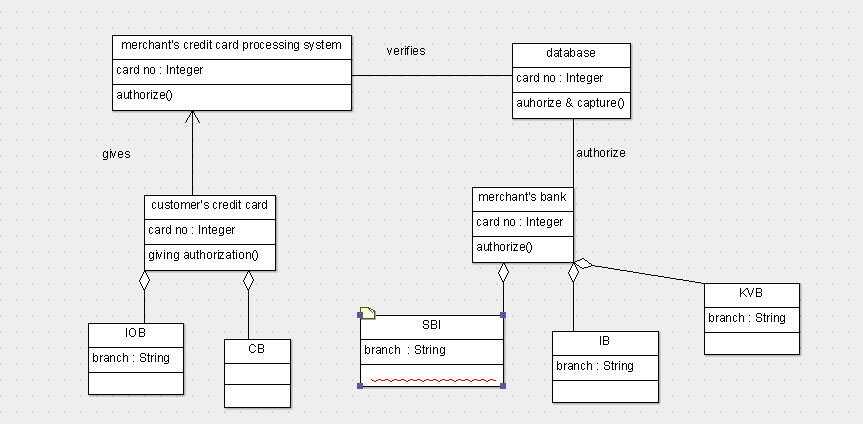
**3.1 USECASE DIAGRAMS:**

****

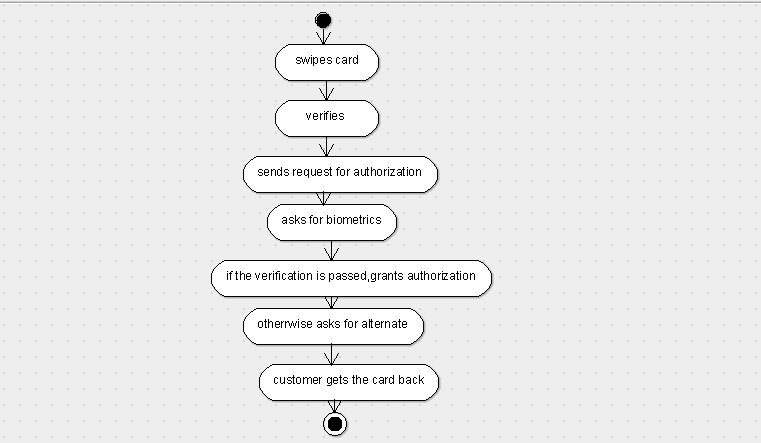
**3.2 SEQUENCE DIAGRAMS:**

****

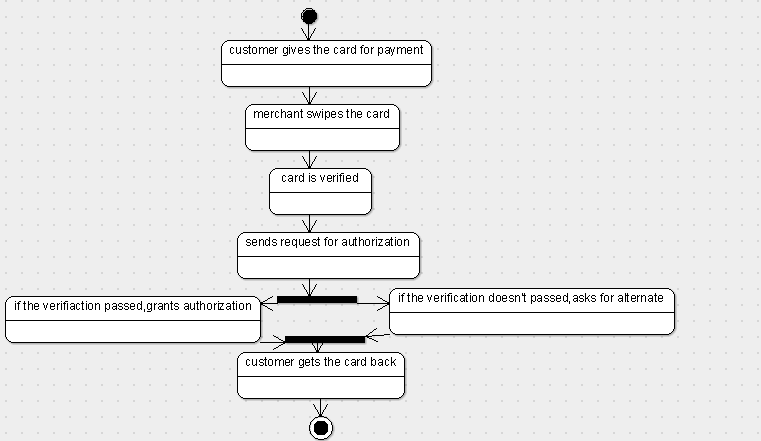
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus credit card system has been implemented successfully.

**EXNO: 10 RECRUITMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

recruitment system is used to select the candidate according to their knowledge through software.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a recruitment System, to enquire the candidates details using enquiry model and then select the candidates with testing methods through software.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the candidate overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

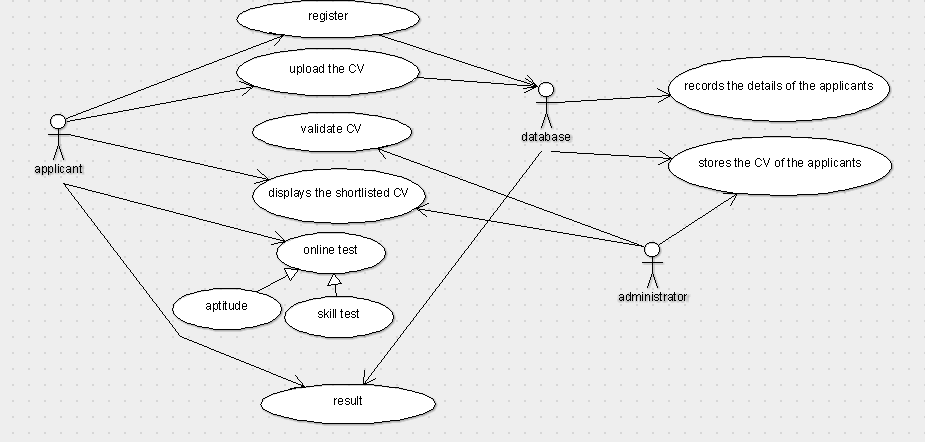
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

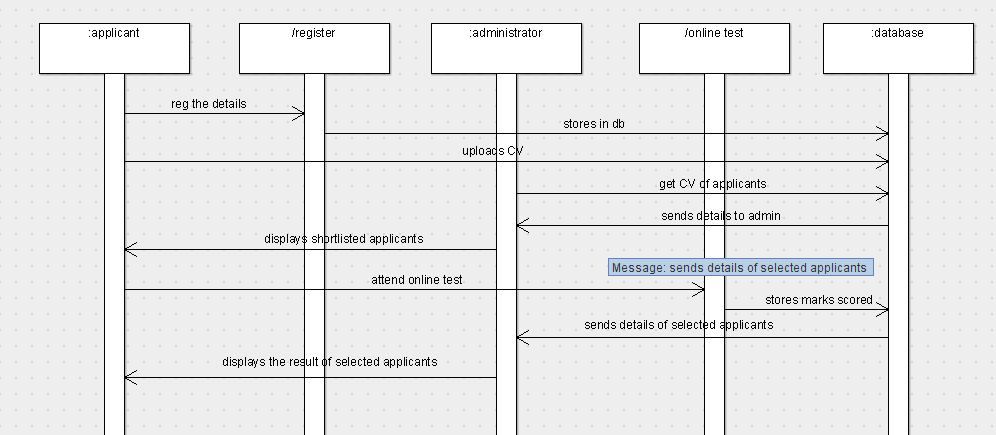
|  |  |
| --- | --- |
| Add details | Add the detail of the client to the database |
| Availability | Check the vacancy of the employment |
| Select | Select the candiadate if any vacancy. |

**3. ARGO UML DIAGRAMS:**

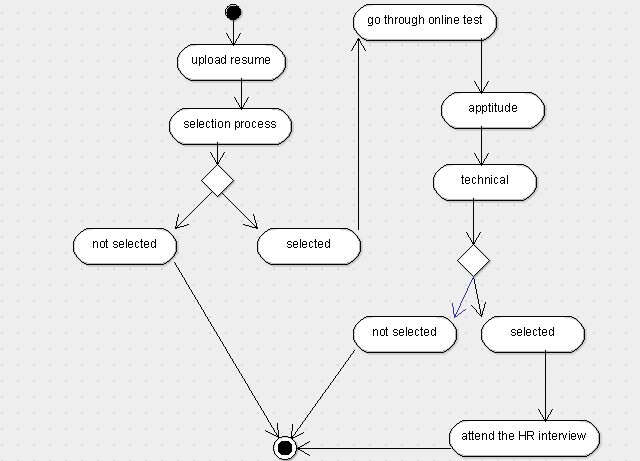
**3.1 USECASE DIAGRAM:**

****

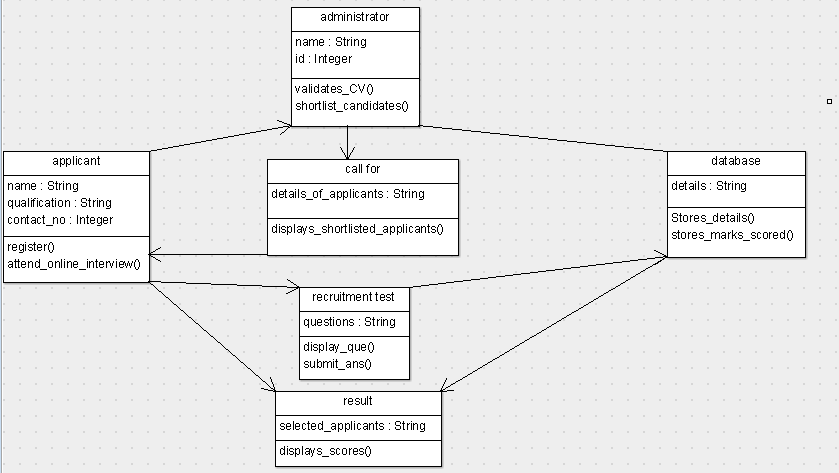
**3.2 SEQUENCE DIAGRAM:**

****

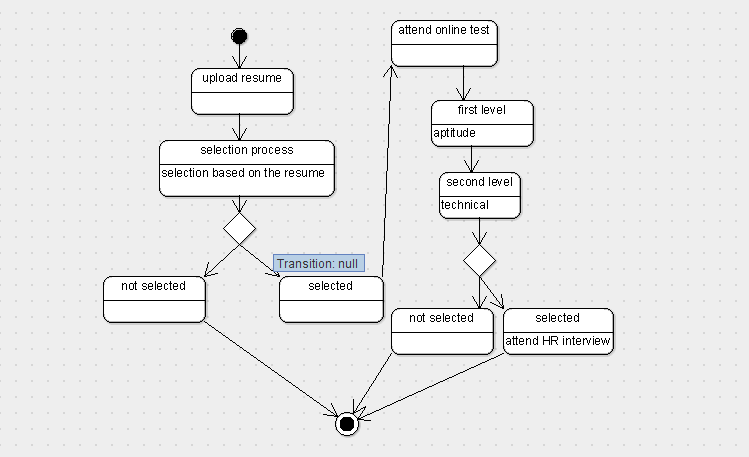
**3.3 ACTIVITY DIAGRAM:**

****

**3.4 CLASS DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus recruitment system has been implemented successfully.

**EXNO: 11 E-BOOK MANAGEMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

e-book management system is used to choose the book and purchase the book through software.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a e-book management System, to enquire the available books using enquiry model and then register the required books using registration model, if the candidate want to cancel the books using cancellation model, if the candidate wants to modify or change the books , a modification model has been developed.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

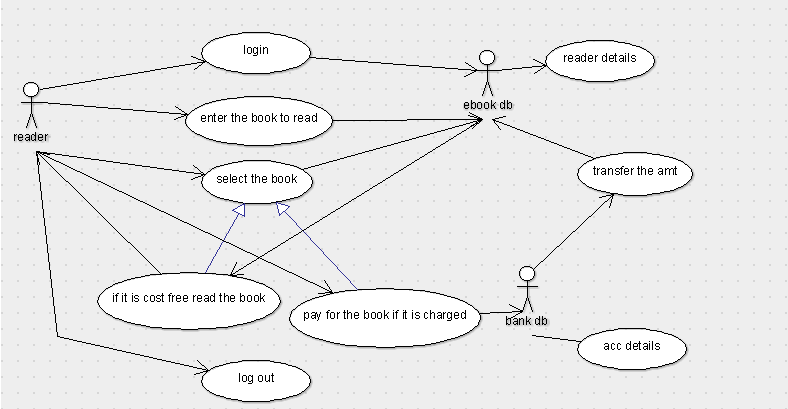
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

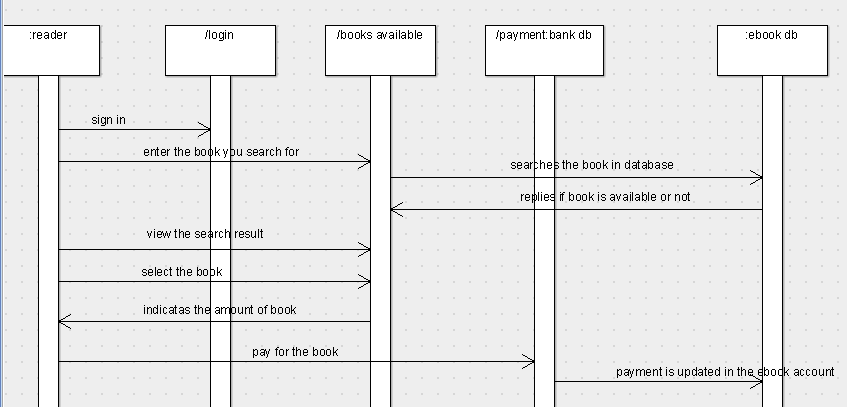
|  |  |
| --- | --- |
| Add details | Add the detail of the booksto the database |
| Availability | Check the availability of the books in system. |
| Select | Select the books if available |

**3. ARGO UML DIAGRAMS:**

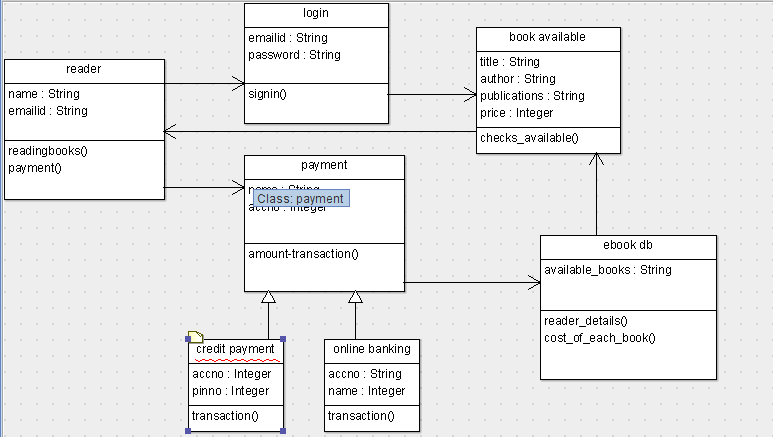
**3.1 USECASE DIAGRAM:**

****

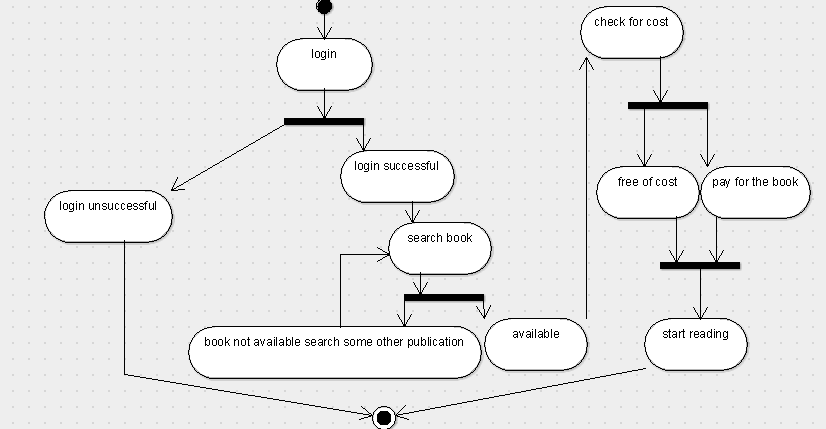
**3.2 SEQUENCE DIAGRAM:**

****

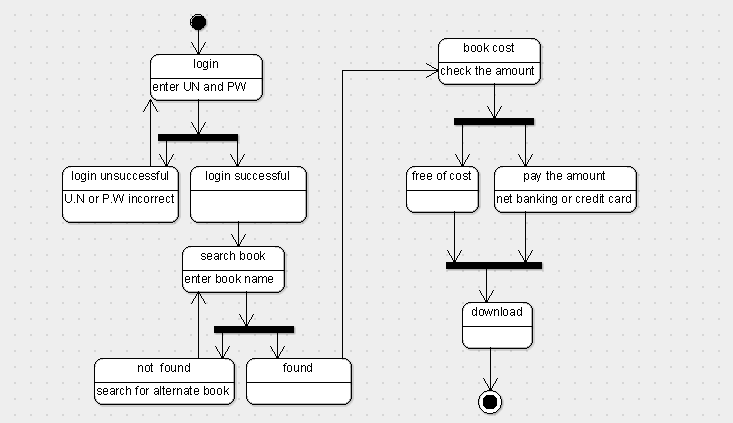
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT**:

Thus e-book management system has been implemented successfully

**EXNO: 12 FOREIGN TRADING SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Foreign trading system is used for marketing in the field of buying and selling products using the software.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a foreign trading System, buying and selling the products through online software system and the detail description of the process is following.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassless of the business dealings in the market. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

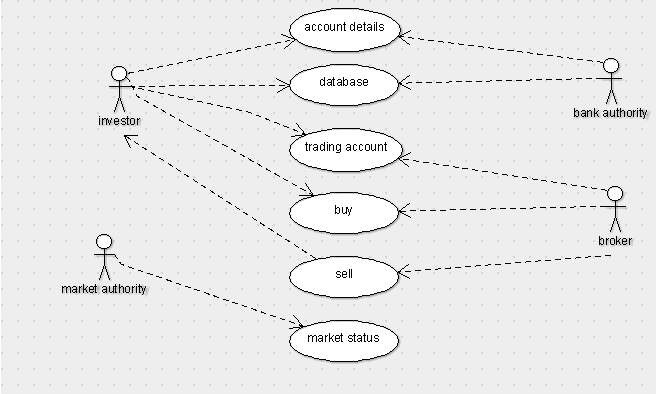
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

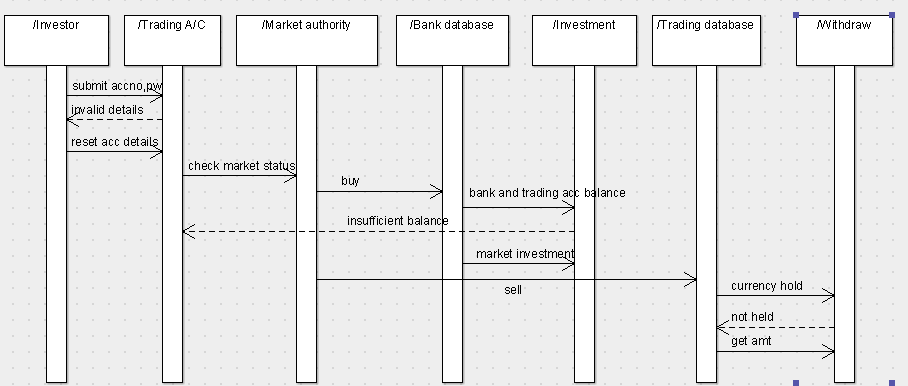
|  |  |
| --- | --- |
| Add details | Add the detail of the investor to the database |
| Availability | Check the availability of the products to the market. |
| Select | Select the available products. |

**3. ARGO-UML DIAGRAMS:**

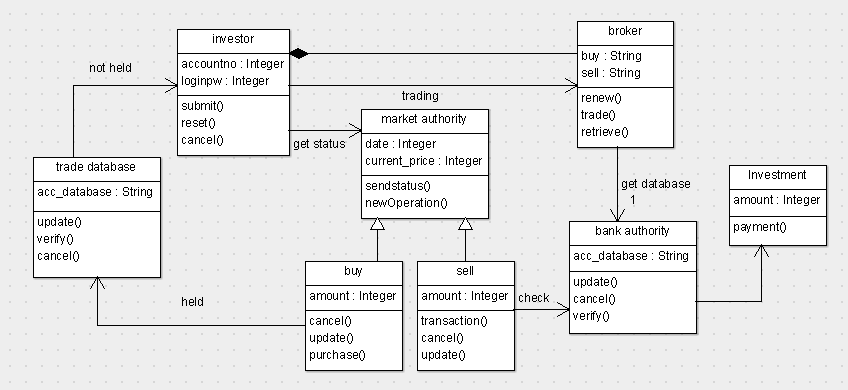
**3.1 USECASE DIAGRAM:**

****

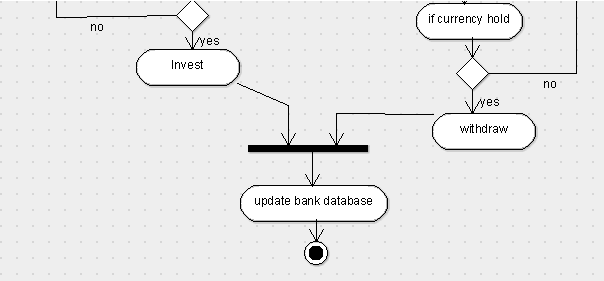
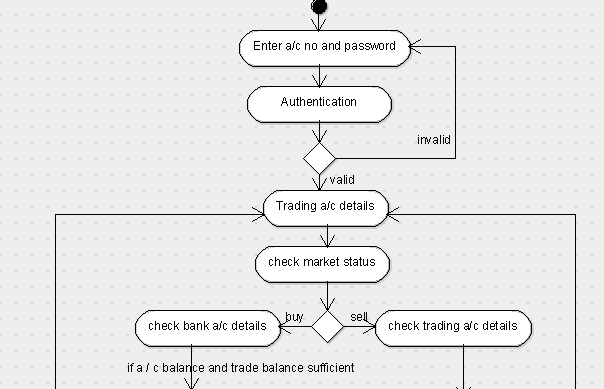
**3.2 SEQUENCE DIAGRAM:**

****

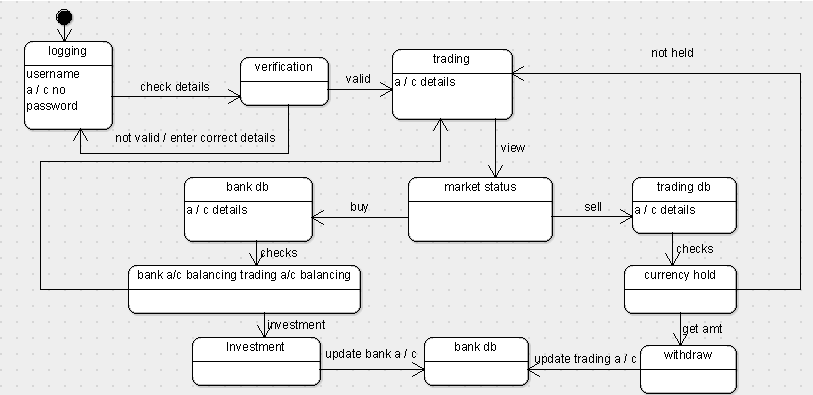
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**



**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus foreign trading system has been implemented successfully.

**EXNO: 13 CONFERENCE MANAGEMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Conference management system is used to interact with the neighbours through software.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

The main scope of the project is to develop a conference management system. Which is used to interact with multiple persons at anytime anywhere through software internet based system.

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to connect the people to interact at anytime anywhere in the universe.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: ARGO UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

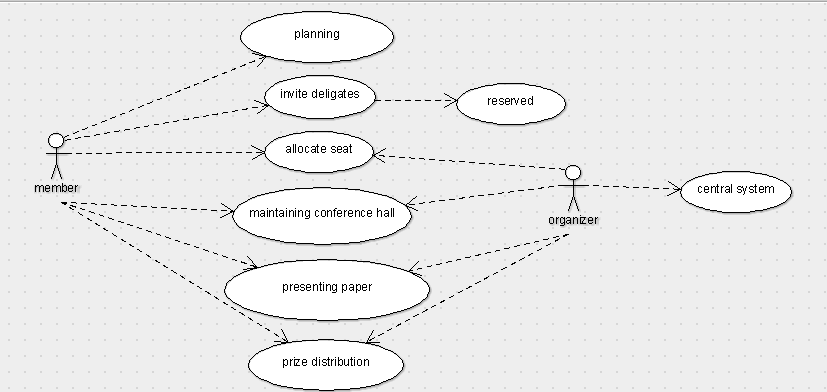
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

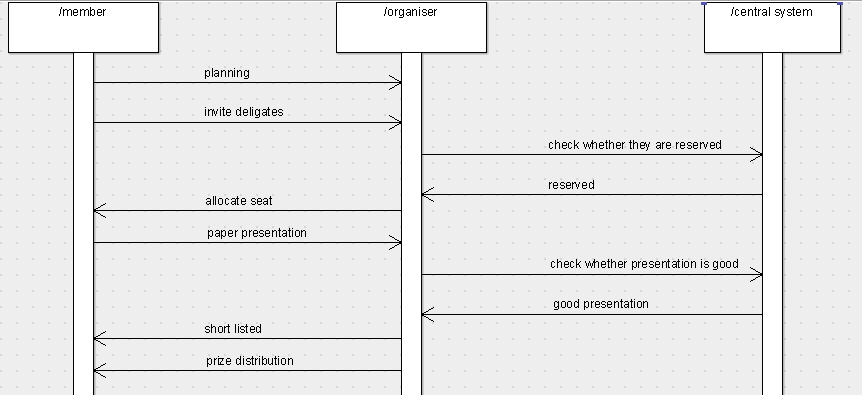
|  |  |
| --- | --- |
| Add details | Add the detail of the neighbours to the database |
| checking | Check the connection whether it is connected or not. |
| Select | Select the neighbour to interact. |

**3. ARGO UML DIAGRAMS:**

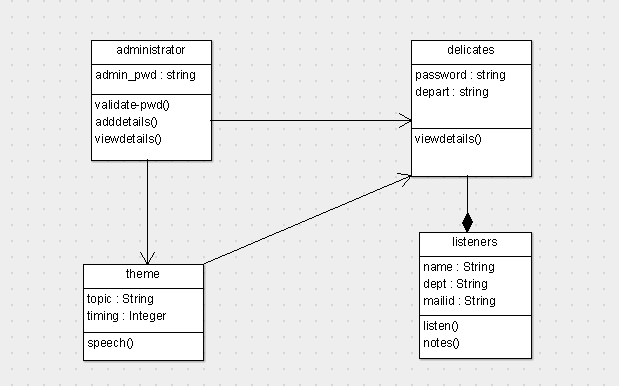
**3.1 USECASE DIAGRAM:**

****

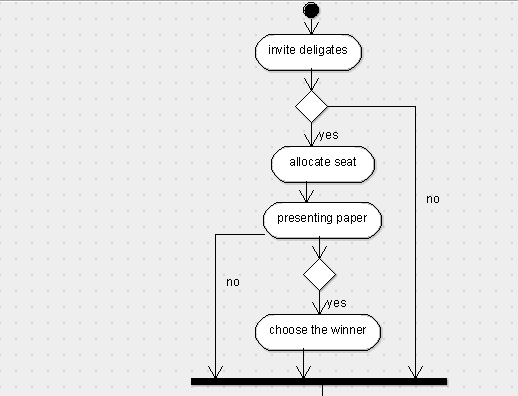
**3.2 SEQUENCE DIAGRAM:**

****

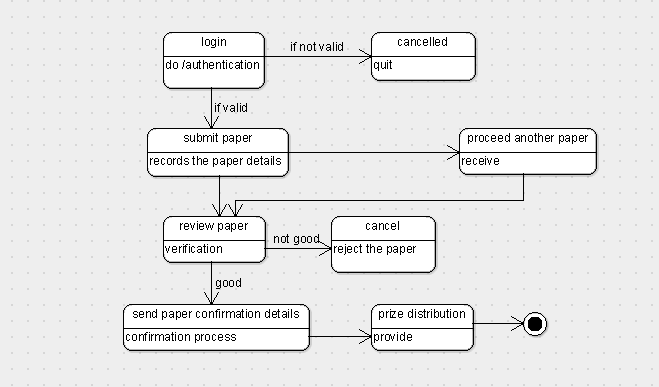
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**



**3.5 STATE CHART DIAGRAM:**

****

**RESULT:**

Thus conference management system has been implemented successfully.

**EXNO: 14 BPO MANAGEMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Online course reservation system is used to choose the course through online by the students. They are provided with a catalog they can choose the course. The catalog contains the detailed description about each course and also the availability of the course which helps the students to decide on their own.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

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

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: Star UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

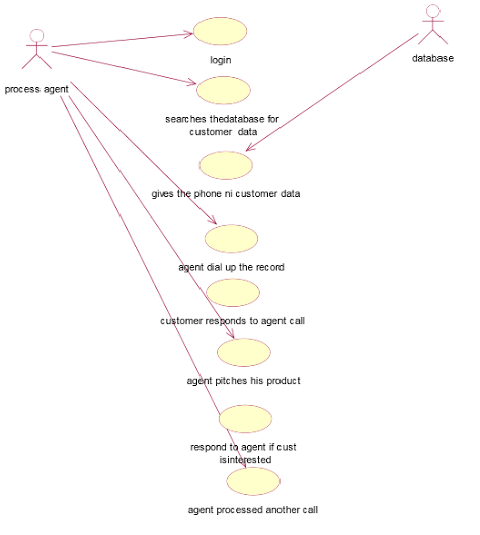
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

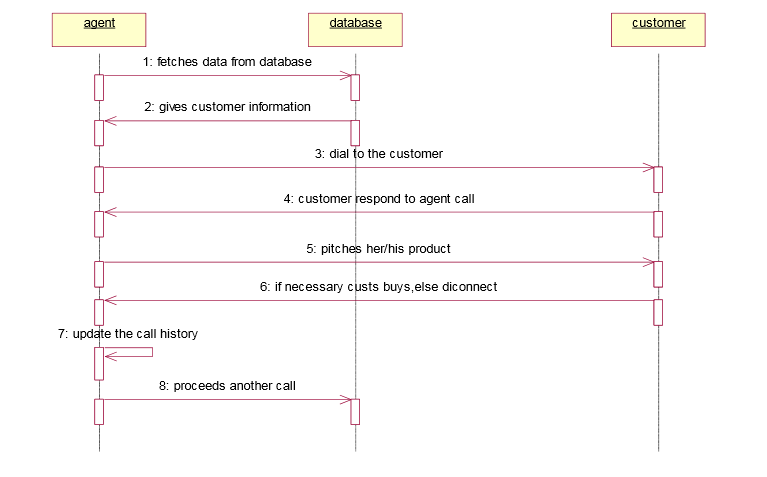
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the course in the college |
| Select | Select the course if available |

**3. ARGO UML DIAGRAMS:**

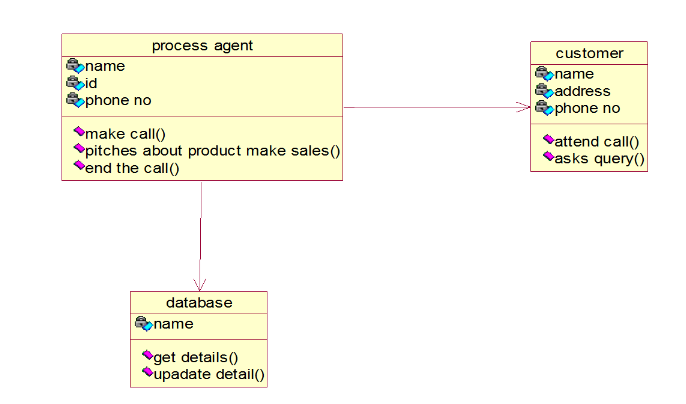
**3.1 USECASE DIAGRAM:**

****

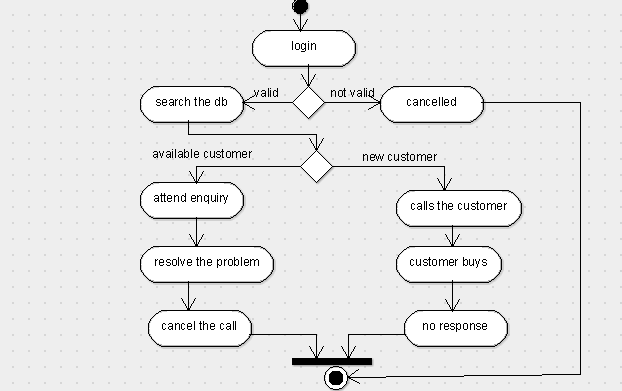
**3.2 SEQUENCE DIAGRAM:**

****

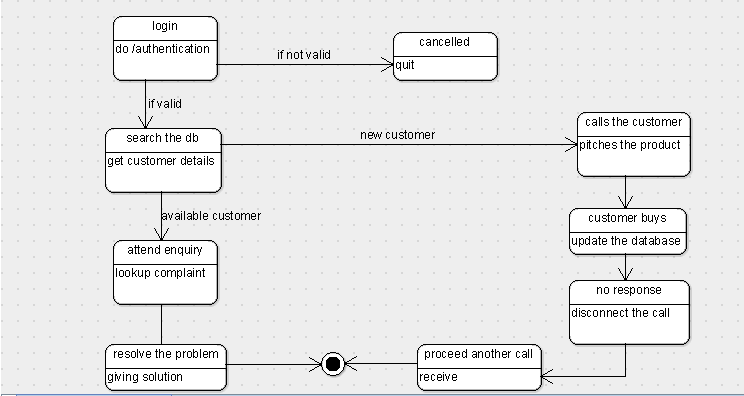
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT**:

Thus BPO management system has been implemented successfully.

**EXNO: 15 LIBRARY MANAGEMENT SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

Online course reservation system is used to choose the course through online by the students. They are provided with a catalog they can choose the course. The catalog contains the detailed description about each course and also the availability of the course which helps the students to decide on their own.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

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

**2.2 PROJECT OBJECTIVE:**

The ultimate objective of this software is to eliminate hassles that the student overcomes while registering him. This software reduces the paper work. This also reduces the time delay.

**2.3 SOFWTARE REQUIREMENTS:**

1. Software: Star UML, visual basic.

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENT:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

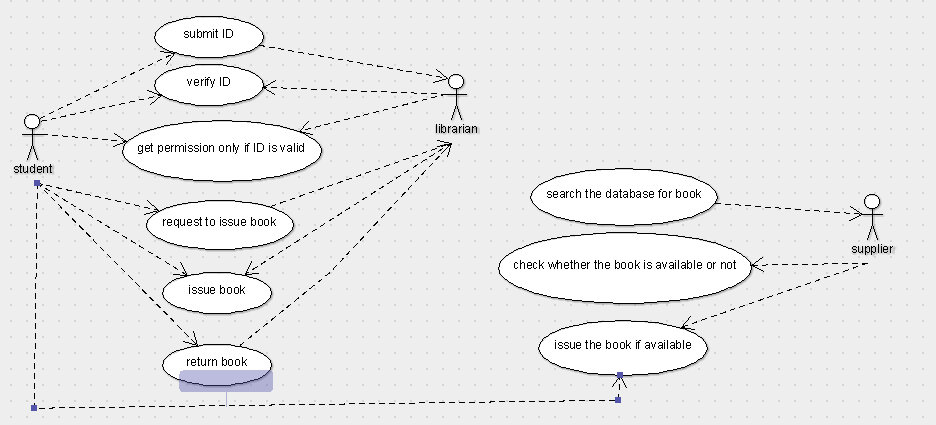
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

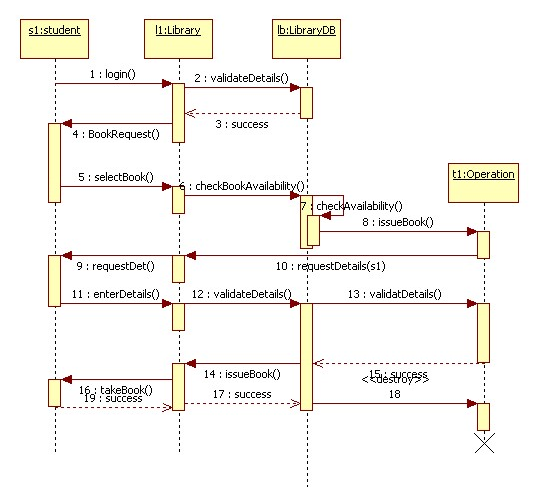
|  |  |
| --- | --- |
| Add details | Add the detail of the student to the database |
| Availability | Check the availability of the course in the college |
| Select | Select the course if available |

**3. ARGO UML DIAGRAMS:**

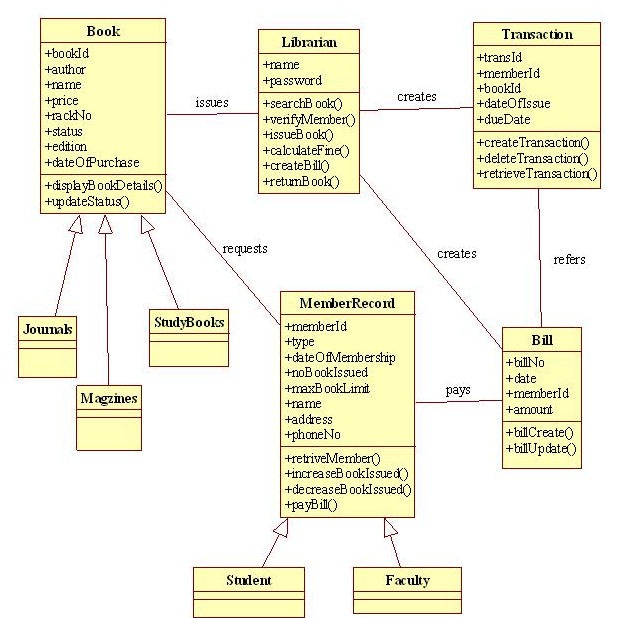
**3.1 USECASE DIAGRAM:**

****

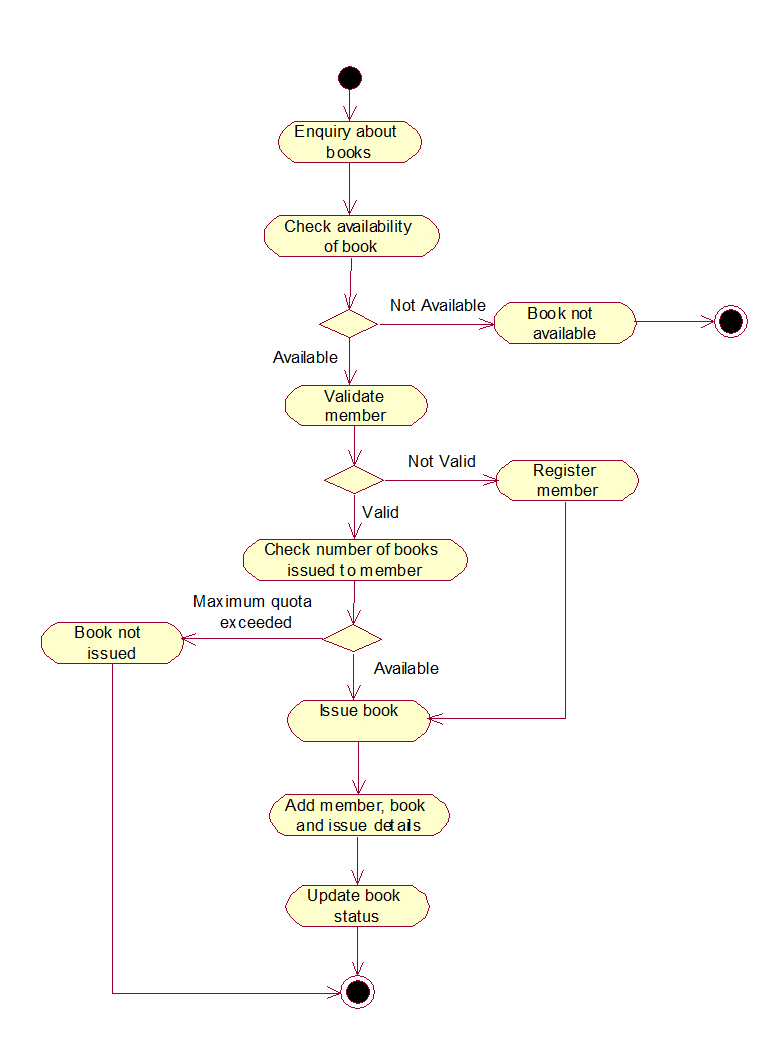
**3.2 SEQUENCE DIAGRAM:**

****

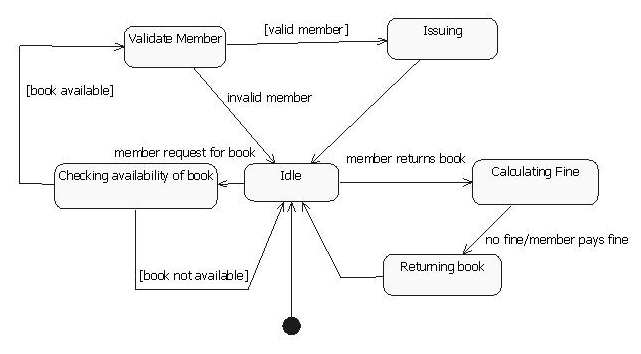
**3.3 CLASS DIAGRAM:**

****

**3.4 ACTIVITY DIAGRAM:**

****

**3.5 STATE CHART DIAGRAM:**

****

**RESULT**:

Thus library management system has been implemented successfully.

**EXNO: 16 STUDENT INFORMATION SYSTEM**

**DATE:**

**1.PROJECT STATEMENT:**

The student information system is a system used for conducting the exam, analyzing the answers, calculating the grade and displaying the result. In this system the student details were present already in the database. The student appears for the exam and enters the answers as required. During the course of the exam, the supervisor monitors the student who writes the exam. The analyzer checks the paper and assigns the marks. Then the system is used for analyzing the percentage and the grade of the student. The administrator and students can view their results by logging into the system.

**2. SOFTWARE REQUIREMENT SPECIFICATIONS:**

**2.1 PROJECT SCOPE:**

•The examination can be conducted in a secure environment.

•Once the papers have been submitted the analyzer can enter the marks into the

database.

•The system assigns the grade automatically based on the marks secured by the

student.

•The marks can be easily updated.

•The students can easily check their scores. Access to database can be setup on

username / password basis with multiple users and privileges.

**2.2 PROJECT OBJECTIVE:**

The ultimate goal of this project is to develop a database to integrate the process of the writing of the exam and assigning the scores to the student and also a way of displaying the result.

**2.3 SOFTWARE REQUIREMENTS:**

1. Software: ARGO UML,visual basic

2. Operating system: Windows XP

3. Database: oracle

**2.4 HARDWARE REQUIREMENTS:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

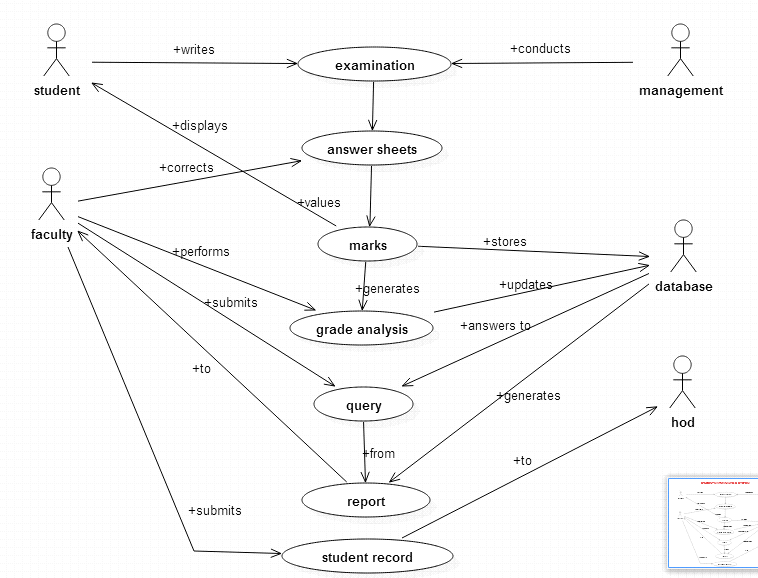
|  |  |
| --- | --- |
| **Content** | **Description** |
| HDD | 20 GB Min  40 GB Recommended |
| RAM | 1 GB Min  2 GB Recommended |

**2.5 FUNCTIONAL REQUIREMENT:**

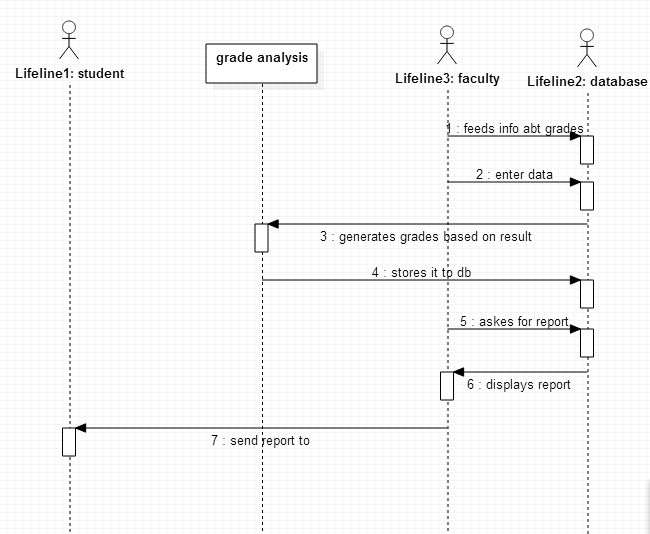
|  |  |
| --- | --- |
| Verifying details | Supervisor verifies the student details before the exam. |
| Answering | The students should answer the questions within the given time limit. |
| Evaluation | The analyzer evaluates the answers and awards marks for the correct answers. |
| Calculating grade | The system generates the grade for the students based on the percentage they have scored. |
| Store the results | The administrator stores the results in the database |
| Display the results | The students view their result by logging into the system. |

**3. ARGO UML – DIAGRAMS:**

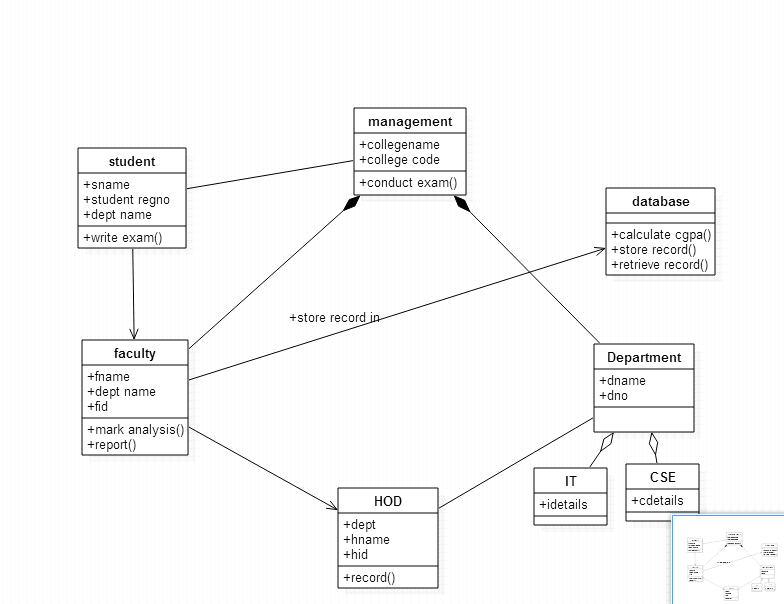
**3.1 USE CASE DIAGRAM:**

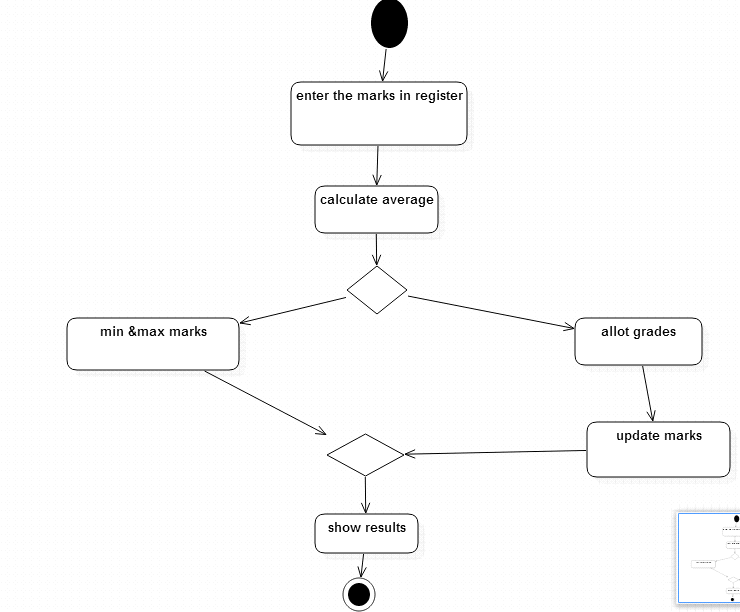


**3.2 SEQUENCE DIAGRAM**

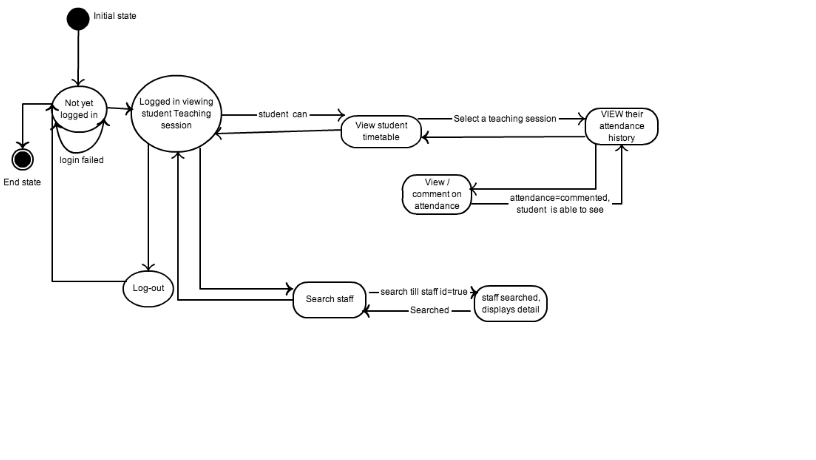


**3.3CLASSDIAGRAM**

**3.4 ACTIVITY DIAGRAM:**



**5.STATE CHART DIAGRAM:**

****

**RESULT**:

Thus student information system has been implemented successfully.

**EX.NO: 17 MINIPROJECT DEVELOPMENT**

**DATE :**

**FRONT END:** JAVA

**BACKEND:**  MYSQL SERVER

**SOFTWARE REQUIRED:**

* ECLIPSCE
* XAMPP CONTROL PANEL
* APACHE TOMCAT8.5 SERVER.

**ALGORITHM:**

* IN **ECLIPSCE:** 🡪FILE🡪NEW🡪OTHERS🡪WEB🡪DYNAMIC WEB PROJECT 🡪 give Project Name 🡪 Click OK.
* Store the APACHE SOFTWARE FILES in the local disk.
* The same local disk should be used throughout the procedures.
* **ECLIPSCE**:IN THE SERVER TAB🡪ClicktheBLUELINK🡪APACHE🡪TOMCAT8.5 🡪NEXT🡪BROWSE🡪APACHE🡪FINISH.
* WEB CONTENT🡪RIGHT CLICK🡪WEB INFO🡪RIGHT CLICK🡪NEW🡪JSP FILE🡪NAME OF JSP FILES.
* **ECLIPSCE:** RIGHT CLICK ON PROJECT NAME (seen on left side)🡪BUILD PATH🡪 CONFIGURE🡪LIBRARIES🡪CLICK ADD EXTERNAL JAR FILE BUTTON 🡪In that select local disk where APACHE SOFTWARE is stored🡪click APACHE 🡪NEXT🡪 LIB🡪SERVELET API🡪APPLY🡪APPLY AND CLOSE.
* Write the two codes given below in two different files with the names insert and retrieve, save in the same local disk as mentioned above.
* **ECLIPSCE:** RIGHT CLICK ON WEB CONTENT 🡪 NEXT🡪CREATE JSP FILE with the names insert.jsp and retrieve.jsp.
* Download the MY SQL CONNECTOR in the same local disk.
* **ECLIPSCE:** Copy and paste the MY SQL CONNECTOR into the LIBRARIES (seen on left side)
* **ECLIPSCE:** Copy and paste the codes in the jsp files from local disk.
* If it shows error ECLIPSCE🡪 SERVER🡪TOMCAT ICON🡪change the port numbers as: 8007,8045,8058.
* save and run.
* goto all programs
* click XAMP 🡪XAMP CONTROL PANEL
* start apache and mysql server
* double click on mysql server admin button, available in the xamp control panel application

OR

* Goto browser🡪 tpye **localhost/phpmyadmin**
* u will get a php myadmin page, click create database and give name for the database
* it will ask u to create table, give table name with no of fields
* then insert field values and do browse, you will get the table with values
* **ECLIPSCE:** run the retrieve file, output will be shown with values retrieved.

**To insert into database**

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<%@page import="java.sql.\*,java.util.\*"%>

<%

String name=request.getParameter("name");

String reg=request.getParameter("register\_no");

String depart=request.getParameter("department");

String year\_field=request.getParameter("year");

int year=Integer.parseInt(year\_field);

try

{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/student\_db", "root", "");

Statement st=conn.createStatement();

int i=st.executeUpdate("insert into student(s\_name,s\_reg,s\_department,s\_year)values('"+name+"','"+reg+"','"+depart+"',"+year+")");

out.println("Data is successfully inserted!");

out.println("<a href='page1.jsp'> Back to Main Page </a>");

}

catch(Exception e)

{

System.out.print(e);

out.println("Error Occured !");

out.println("<a href='page1.jsp'> Back to Main Page </a>");

e.printStackTrace();

}

%>

**To retrieve the data from the table**

<%@ page import="java.sql.\*" %>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html>

<html>

<head>

<title>Student\_Lib</title>

<style type="text/css">

body{

background-image: url("clg.jpg");

background-repeat: no-repeat;

background-size: 100%,100%;

background-color: grey;

}

table{

margin-left:20%;

width:60%;

margin-top:10%;

}

table, th, td {

border: 1px solid black;

background-color: white;

}

td{

text-align:center;

}

.input\_fields{

position: relative;

width:100%;

margin-bottom: 20px;

padding-left: 10px;

padding-top: 3px;

padding-bottom: 3px;

size: 12px;

}

#form\_div{

position: relative;

width:40%;

height: 100%;

left: 25%;

margin-top: 10%;

padding-left: 5%;

padding-right: 5%;

padding-top: 3%;

padding-bottom: 3%;

border-radius: 5%;

background-color:#C0C0C0;

}

#summit\_button{

width:20%;

height:30px;

background-color: grey;

color: white;

margin-left: 40%;

}

</style>

</head>

<body>

<div id="form\_div">

<form method="post" action="insert\_into\_db.jsp">

Student Name : <input class="input\_fields" type="text" name="name">

Reg No : <input class="input\_fields" type="text" name="register\_no">

Department : <input class="input\_fields" type="text" name="department">

Year : <input class="input\_fields" type="number" name="year">

<input id="summit\_button" type="submit" name="button" value="Submit">

</form>

</div>

</body>

<%

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/student\_db","root","");

Statement st=con.createStatement();

ResultSet result=st.executeQuery("select \* from student");

int c=0;

while(result.next()){

if(c==0){

if(result.next()){

out.println("<table><tr><th>Name</th><th>Reg No</th> <th>Deprt</th> <th>Year</th> </tr>");

c=1;

}

}

String name=result.getString("s\_name");

String reg=result.getString("s\_reg");

String deprt=result.getString("s\_department");

String year=result.getString("s\_year");

out.println("<tr> <td>"+name+"</td><td>"+reg+"</td><td>"+deprt+"</td><td>"+year+"</td></tr>");

}

out.println("<table>");

}

catch(Exception e){

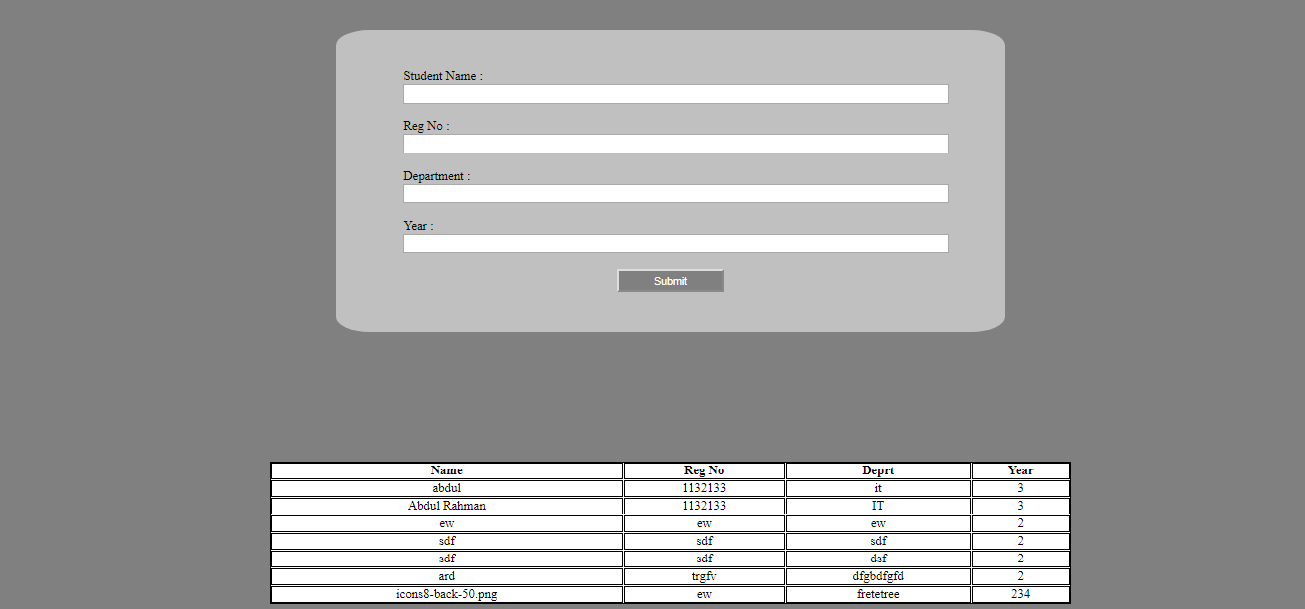
out.println(e);

}

%>

</html>

**OUTPUT:**

****

**RESULT**

Thus the mini project for payroll is done successfully