# SRM VALLIAMMAI ENGINEERING COLLEGE

SRM NAGAR, KATTANKULATHUR -603203

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# LAB MANUAL

## 1904405 - OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

(**II YEAR CSE/ FOURTH SEMESTER**)

(**REGULATION 2019**)
2022-2023 – Even Sem

**PREPARED BY**

**Ms.R.Anitha** *Assistant Professor (Sr.G)*
**Mr.K.Shanmugam**, *Assistant Professor (Sr.G)*
**Mr.N.Leo Bright Tennisson ,** *Assistant Professor (Sr.G)*

Department of Computer Science and Engineering,
SRM Valliammai Engineering College.

## 1904405 OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

**L T P C0 0 4 2**

### COURSE OBJECTIVES :

The student should be made to:

- To capture the requirements specification for an intended software system
- To draw the UML diagrams for the given specification
- To map the design properly to code
- To test the software system thoroughly for all scenarios
- To improve the design by applying appropriate design patterns.

### LIST OF EXPERIMENTS

### UML DESIGN

1. To develop a problem statement.
2. Identify Use Cases and develop the Use Case model.
3. Identify the conceptual classes and develop a domain model with UML Class diagram.
4. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence diagrams.
5. Draw relevant state charts and activity diagrams.
6. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
7. Develop and test the Technical services layer.
8. Develop and test the Domain objects layer.
9. Develop and test the User interface layer

**Suggested domains for Mini-Project ( any 3 can done):**

1. Passport automation system.
2. Exam Registration
3. Stock maintenance system.
4. Online course reservation system
5. Software personnel management system
6. Recruitment system
7. Conference Management System
8. BPO Management System
9. Library Management System

**TOTAL : 60 PERIODS**

### COURSE OUTCOMES :

**At the end of the course, the student should be able to**

- Perform OO analysis and design for a given problem specification.
- Identify and map basic software requirements in UML mapping.
- Use the UML analysis and design diagrams.
- Improve the software quality using design patterns and to explain the rationale behind applying specific design patterns
- Create code from design.

## LIST OF EQUIPMENTS FOR A BATCH OF 30 STUDENTS:

SOFTWARE: Rational Rose Or Agro UML
HARDWARE: Standalone desktops - 30 Nos. (or) Server supporting 30 terminals or more.

**Ex.No:1**

# PASSPORT AUTOMATIONSYSTEM

**DATE:**

### AIM

To develop a Passport Automation System project using argouml tool.

### PROBLEM STATEMENT

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. The core of the system is to get the registration form 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. The administrator will be provided with an option to display the current status of application to the applicant, which they can view in their online interface. After all the necessary criteria have been met, the original information is added to the database and the passport is sent to the applicant.

## SOFTWARE REQUIREMENT SPECIFICATION

TABLE OF CONTENTS

## 1. INTRODUCTION

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.

### 1.1. PURPOSE

The entire process of Issue of Passport is done in a manual manner then it would take several months for the passport to reach the applicant. Considering the fact that the number of applicants for passport is increasing every year, an automated system becomes essential to meet the demand.

### 1.2. SCOPE

The system provides a communication platform between the applicant and the administrator. This will help both the applicant and issuer by reducing the time and workload.

### 1.3 DOCUMENT CONVENTIONS, ACRONYMS AND ABBREVIATIONS

- Administrator - Refers to the super user who is the Central Authority who has been vested with the privilege to manage the entiresystem.
- Applicant - One who wishes to obtain thePassport.
- PAS - Passport AutomationSystem.

### 1.4.REFERENCES

www.passport.tn.nic.in

www.india.gov.in

## 2.0 OVERALLDESCRIPTION

### 2.1 PRODUCTPERSPECTIVE

The PAS acts as an interface between the applicant and the administrator. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the passport.

### 2.2 PRODUCTFUNCTIONS

Front-End**:** The area of the interface in which the applicants will see their status.Back-End: The area of the interface in which the administrator verifies the details of the applicant.

### 2.3. TOOLS TO BE USED

Visual Basic and Microsoft Access

## 3. EXTERNAL INTERFACEREQUIREMENTS

### 3.1 HARDWAREINTERFACES

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWAREINTERFACE

The system uses ODBC drive to connect and control the database.

## 4 SYSTEMFEATURES

### 4.1 APPLYING FORPASSPORT

#### 4.1.1 DESCRIPTION ANDPRIORITY

This system allows the applicant to apply for the passport through the interface.

#### 4.1.2 STIMULUS/RESPONSESEQUENCE

When the applicant submits all the required details, his status will be updated in his/her account.

#### 4.1.3 FUNCTIONALREQUIREMENTS

REQ 1: The applicant must have a registered account.

REQ 2: The applicant must submit only the valid information.

## 5 OTHER NONFUNCTIONALREQUIREMENTS

### 5.1 PERFORMANCEREQUIREMENTS

To increase performance and to free up database resources for other tasks, the default features are written to cache files on their initial load, and consequent accesses to them result in parsing a flat file for data.

### 5.2 SAFETY REQUIREMENTS

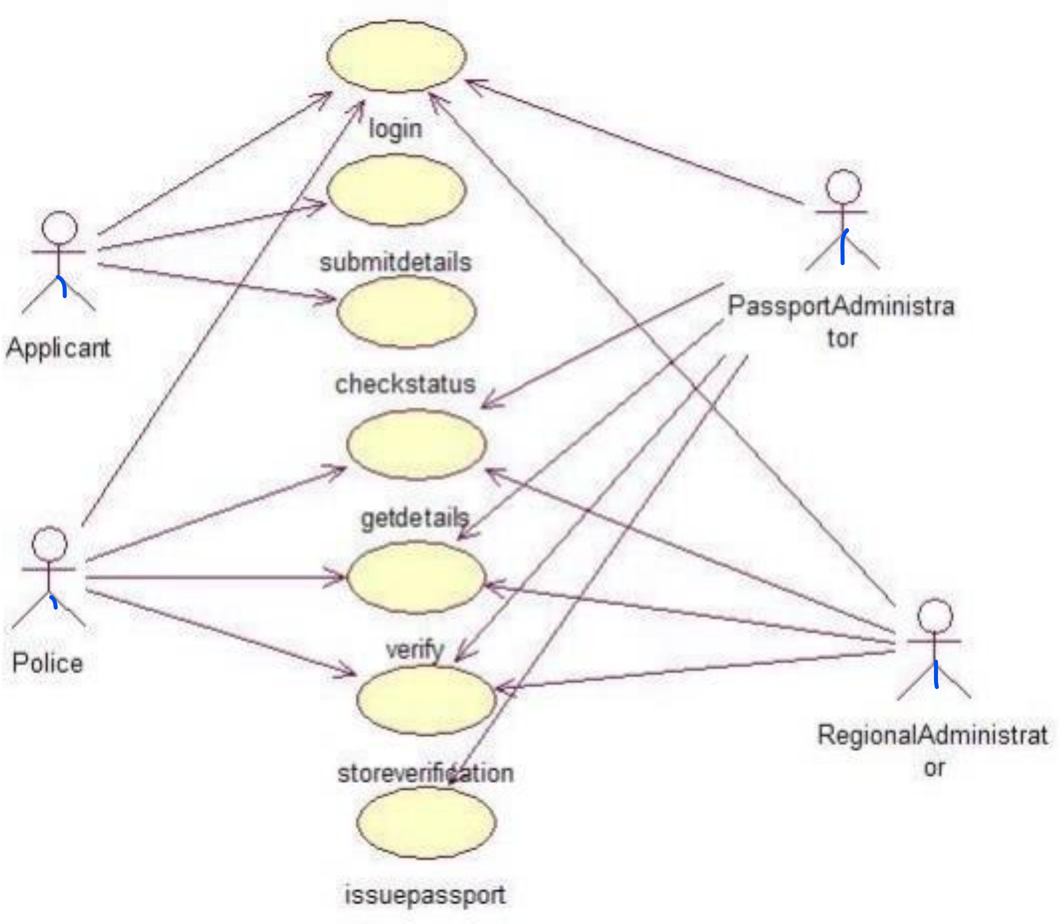
The database must be maintained effectively and the administrator must maintain the interface properly. The user has to be careful while submitting the information.

### 5.3 SECURITY REQUIREMENTS

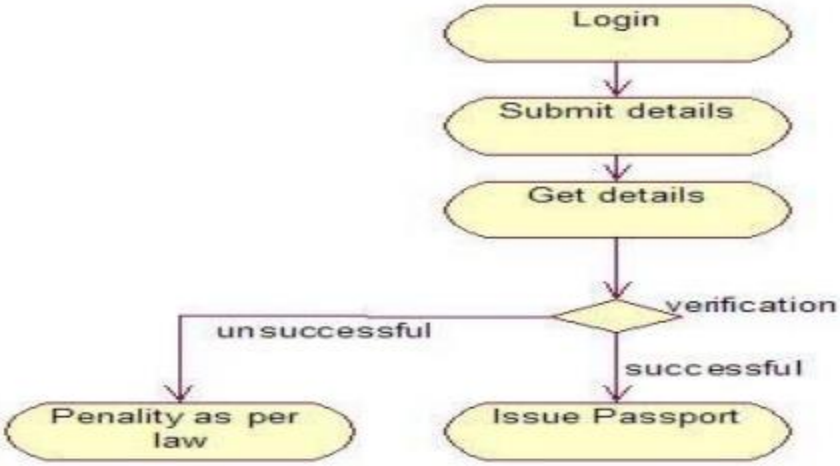
Passwords for registered accounts are stored as a hash in the database. Only the administrator can have access to the main database. The database must be protected from hacking the details of the applicants

## PASSPORT AUTOMATION SYSTEM

**1.Use Case Diagram :**

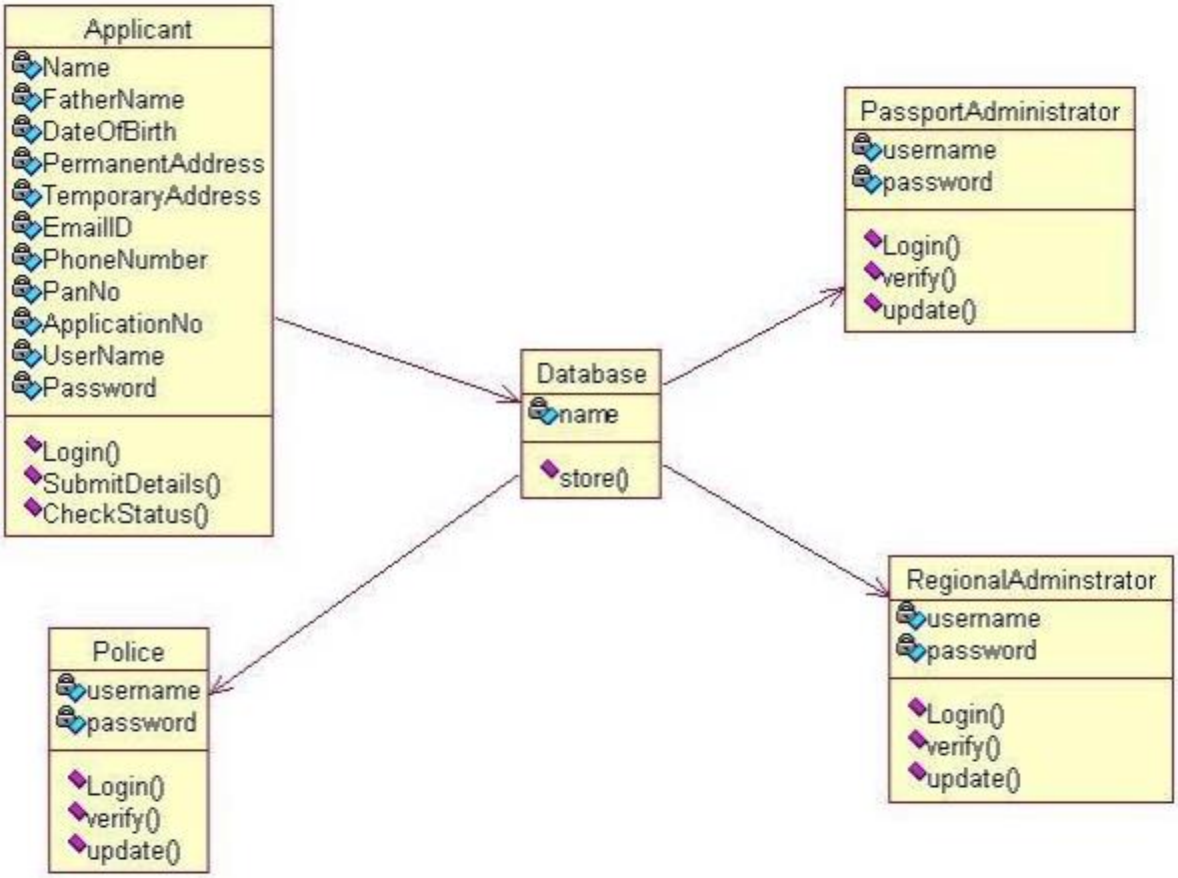


**2.Activity Diagram :**



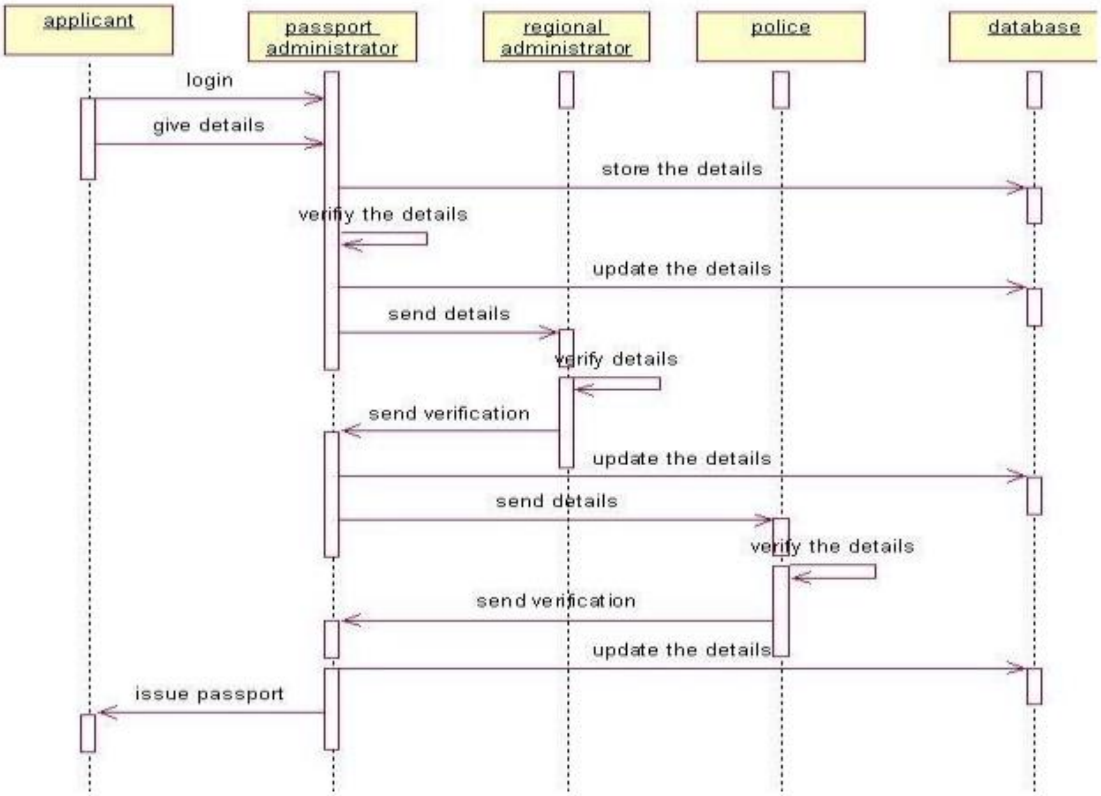
operation

**3.Class Diagram:**



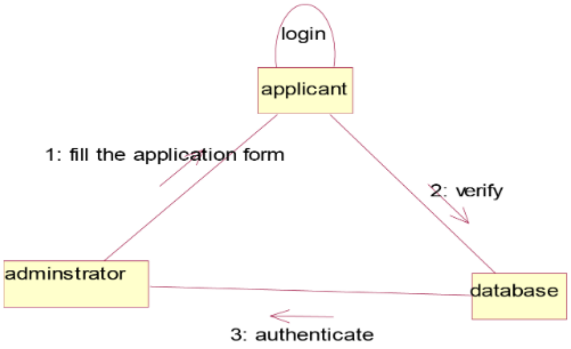
## 4.Interaction Diagram:

### (i)Sequence Diagram

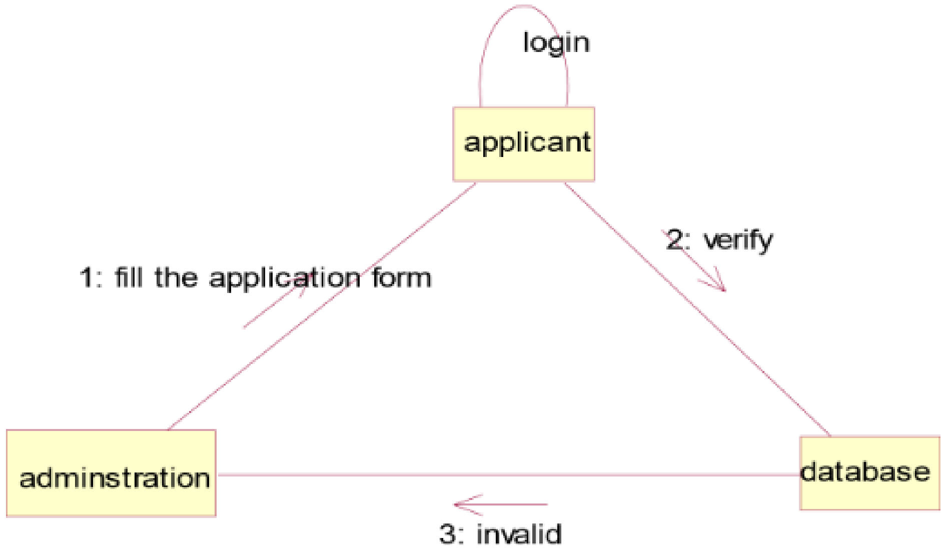


## (ii)Collaboration Diagram :

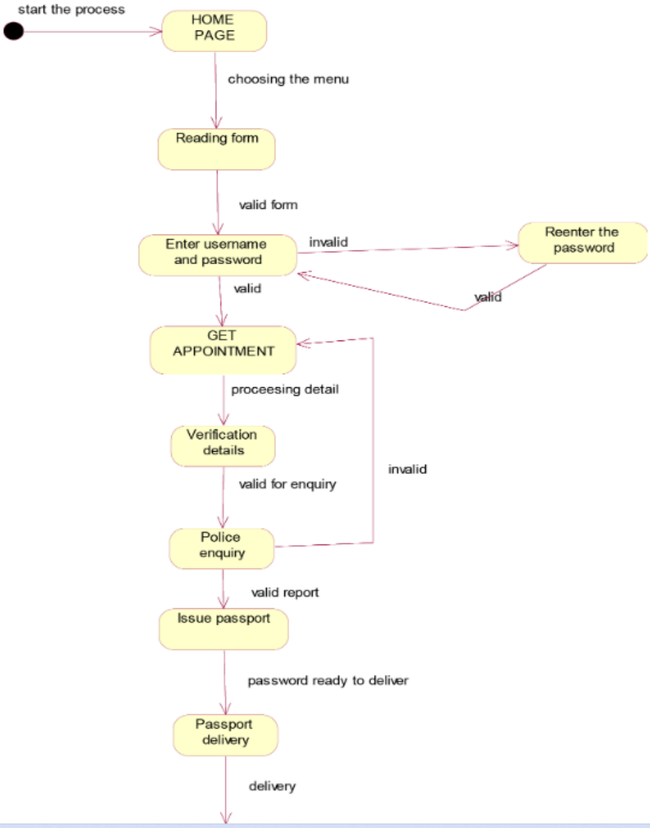
**Valid Passport Pin:**



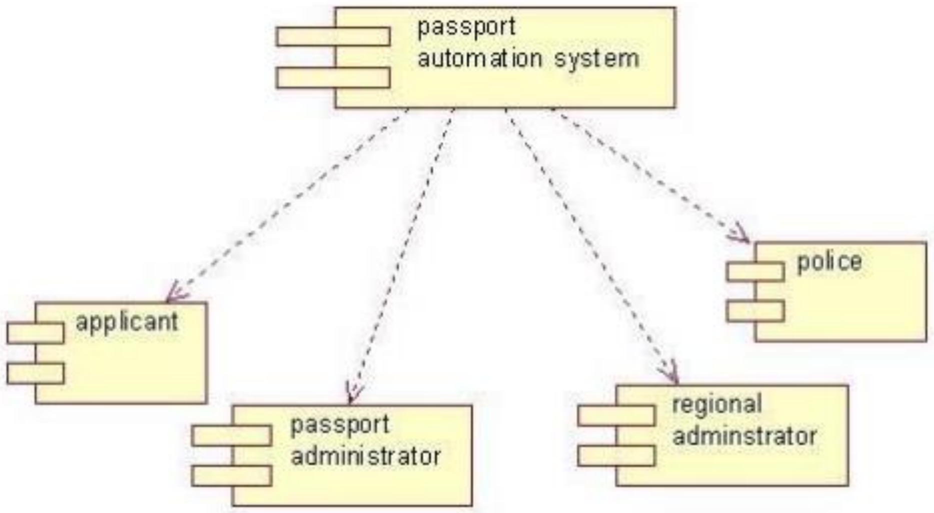
### Invalid Passport Pin :



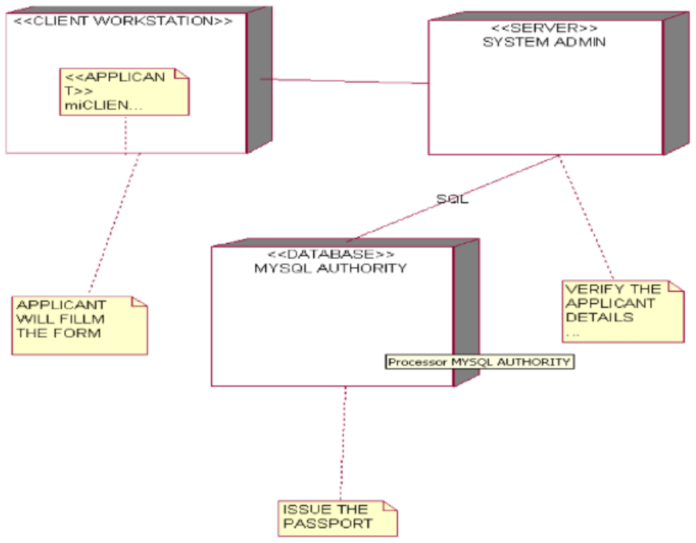
## 5.State Chart Diagram



## 6.Component Diagram:



## 7.Deployment Diagram:



## Result:

Thus the UML diagrams has been successfully completed for Passport automation system.

## Ex.No: 2 EXAM REGISTRATION SYSTEM

**DATE:**

**AIM**

To develop a Exam registration system project using argouml tool.

**PROBLEM STATEMENT:**

Exam registration system is the process of enrolling for exams through online. Enrolling students into the general associate course examination is a very difficult and important process. In case of degree courses, the exam is held two times per annum and students are required to pass in the exam to get their degrees. The main outcome is to computerize everything related to associate course/degree examination. Student can enter their details in a form and information related to examination such as venue, date, time, etc., are displayed in a web page. The registration number and other related information will be sent through e-mail.

**SOFTWARE REQUIREMENT SPECIFICATION**

TABLE OF CONTENTS

### TABLE OF CONTENTS

### 1. INTRODUCTION:

Exam registration system is an interface between the applicant/student and the authority responsible for issuing the hall ticket.

#### 1.1. PURPOSE

If the exam registration is done in conventional method, it would require more time and it would contain lots of error in it. So this system uses online registration system which minimizes error in conventional system.

### 1.2. PRODUCT SCOPE

The scope of this system involves the development and maintenance of the online examination registration that integrates all support modules after suitable evaluation and acceptance and appropriate ASP(Application Service Provider) hosting solutions.

### 1.3. DOCUMENT CONVENTIONS

- Applicant-The person who registers for exam.

### 1.4.REFERENCE

http://en.wikipedia.org/wiki/examregistrationsystem.

## 2. OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

The form acts as an interface between the student and administrator. It makes the registration process simple as it involves the registration system.

### 2.2. PRODUCT FUNCTIONS

System separate registration forms for students. Management of exam from one central location and providing for printing the hall ticket/professional certificate online.

### 2.3 TOOLS TO BE USED

Visual basic and Microsoft Access

## 3. EXTERNAL INTERFACES

### 3.1 HARDWARE INTERFACES

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWARE INTERFACE

The system uses ODBC drive to connect and control the database.

## 4. SYSTEM FEATURES

### 4.1. FILING THE FORM

### 4.1.1. DESCRIPTION AND PRIORITY:

The system allows the user to fill the registration form.

### 4.1.2. STIMULUS/RESPONSE SEQUENCE:

The user must fill the form as per the requirements to get selected for exam.

### 4.1.3. FUNCTIONAL REQUIREMENTS

The member should be authenticated by means of unique login id and password.

## 5. OTHER NON-FUNCTIONAL REQUIREMENTS

### 5.1. PERFORMANCE REQUIREMENTS

To improve the performance, the database resources should be freed frequently.

### 5.2. SAFETY REQUIREMENTS:

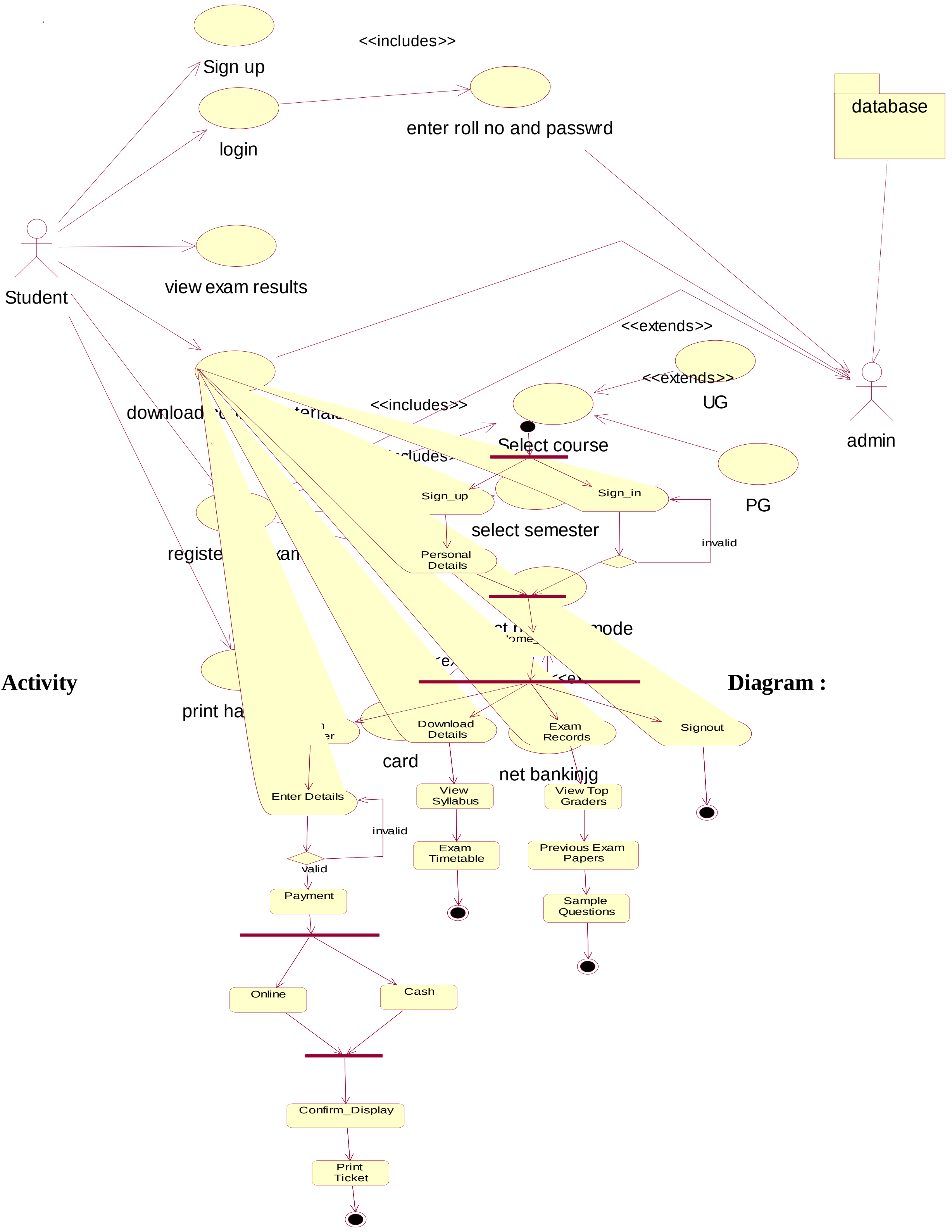
The database must be maintained and monitored frequently to prevent the attack of virus.

### 5.3. SECURITY REQUIREMENTS:

The proof of the applicants must be kept securely in the database and only administrator can access.

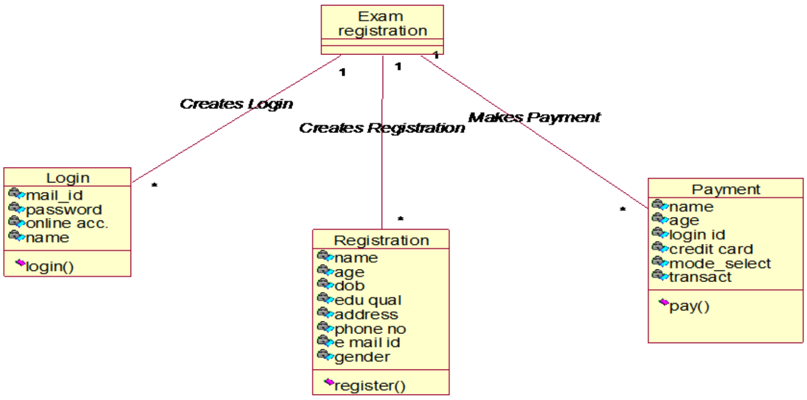
# EXAM REGISTRATION SYSTEM

## 1.Use Case Diagram :



## 2.Activity Diagram :

## 3.Class Diagram:

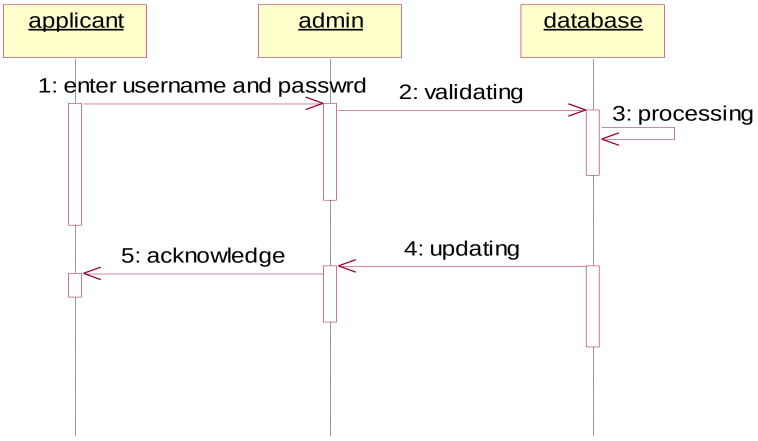


## 4.Interaction Diagram:

### (i)Sequence Diagram :

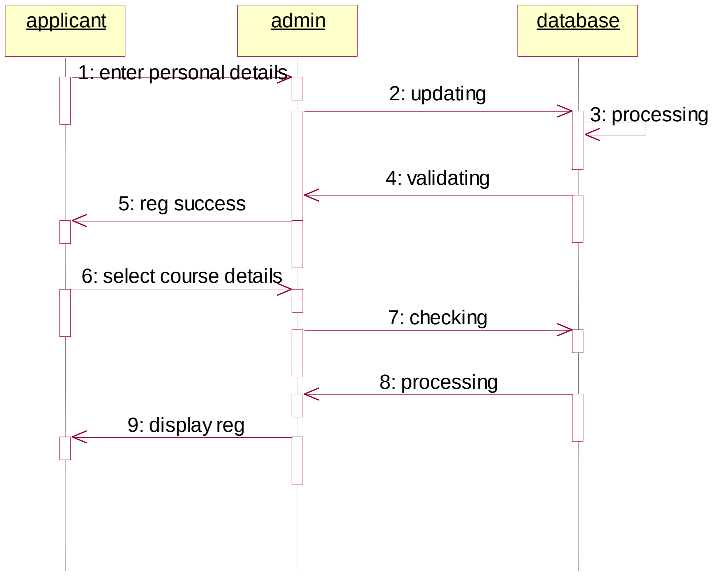
### Login:

***Login sequence***

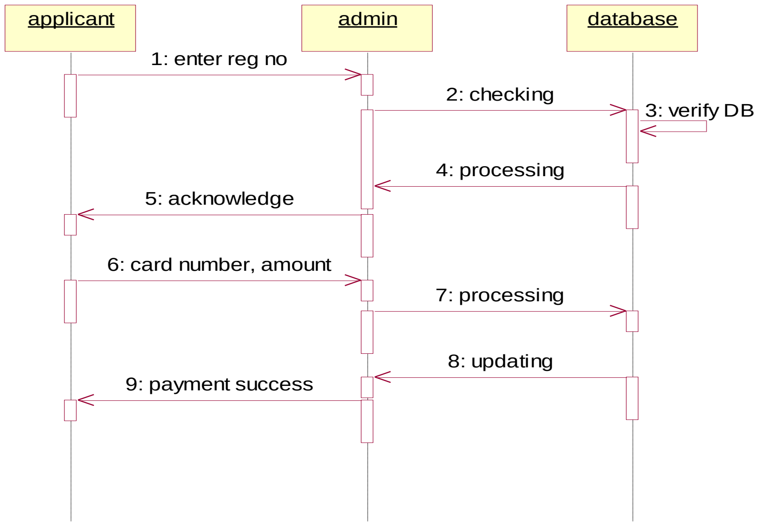


### Registration:

***Registration Routine***



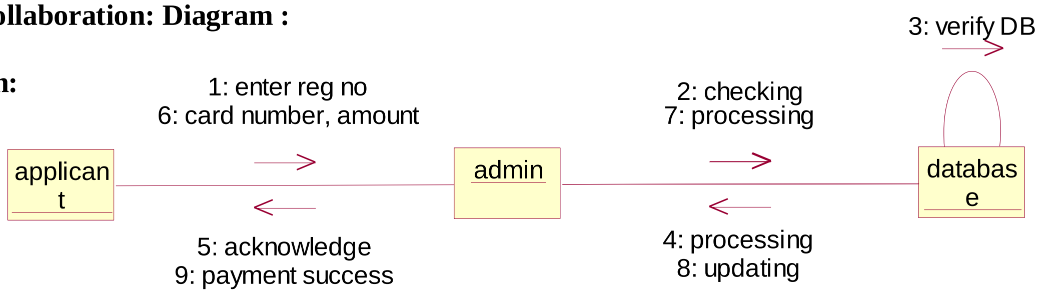
## Payment:



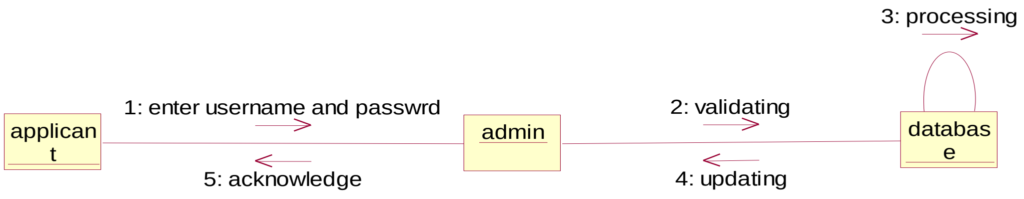
***Payment procedure***

## (ii)Collaboration: Diagram :

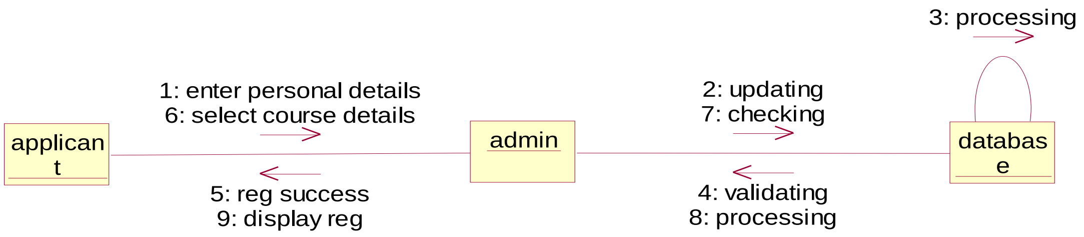
### Login:



## Registration:



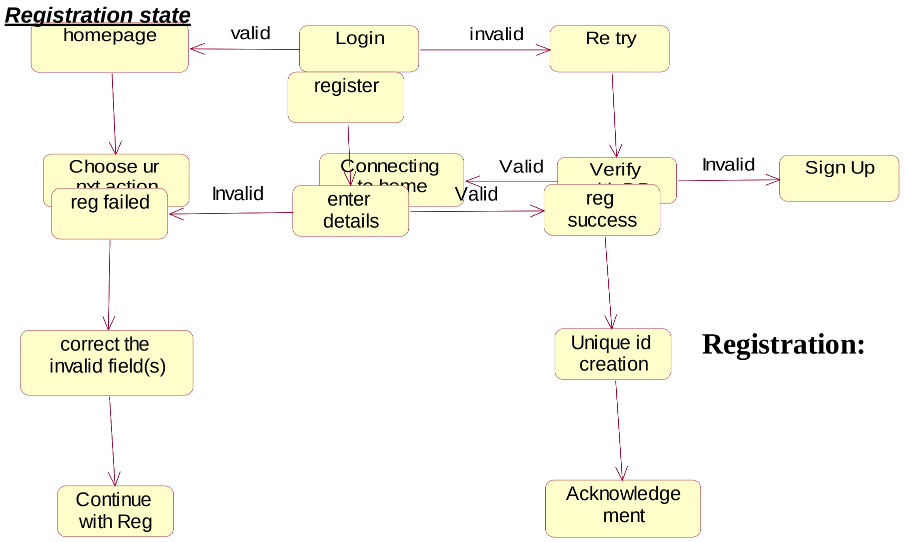
## Payment:



## 5.State chart Diagram :

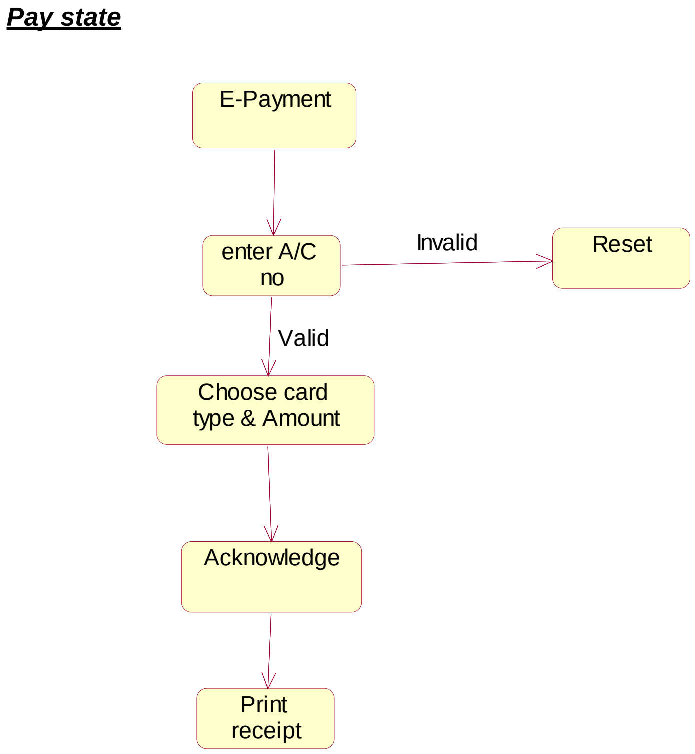
## Login:

***Login state***

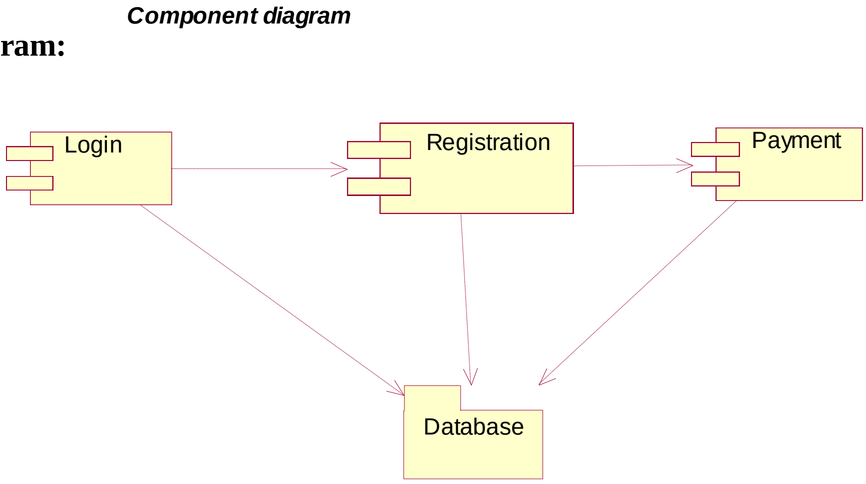


## Registration:

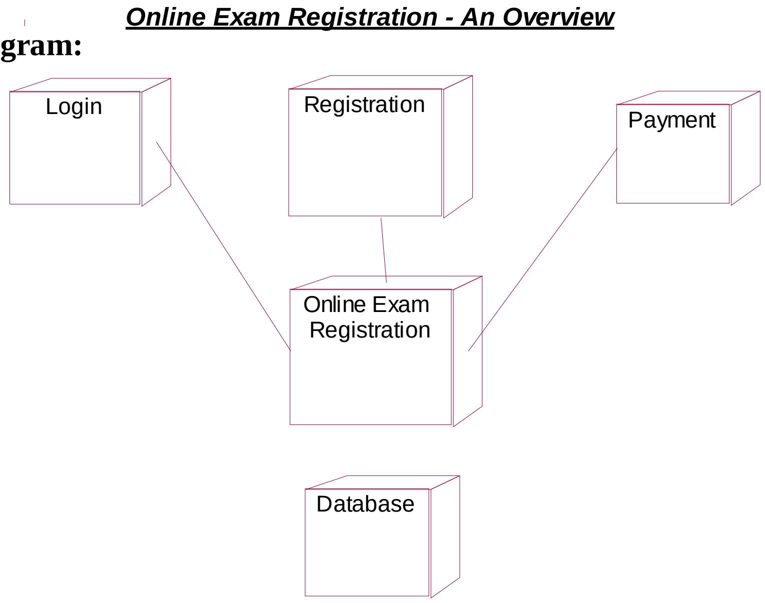
## Payment:



## 6.Component Diagram:



## 7.Deployment Diagram:



**Result:**

Thus the UML diagrams has been successfully completed for Library management system.

## Ex.No: 3 STOCK MAINTENANCE SYSTEM

**DATE:**

**AIM**

To develop a Stock Maintenance system project using argouml tool**.**

**PROBLEM STATEMENT**

In any organization the records about stock are maintained manually. Since these activities are done manually and need regular updates to these records there is much probability of error occurrence, moreover this work is a tedious work process. The various activities involved in these types of systems are adding stock, deleting stock, updating stock, and searching stock.The system access is given to the accountant, stock manager.

## SOFTWARE REQUIREMENT SPECIFICATION

### TABLE OF CONTENTS

## 1. INTRODUCTION:

In order to overcome the major flaws of the manual record maintenance,t he shift towards computerization of the process is mandatory. The primary objective of this project is to implement the software which is capable of maintenance of the records with simplicity and in well structured manner. The main aim is to overcome the work load and time consumption which makes the maintenance of in an organization as a tedious process.

### 1.1 PURPOSE:

The main purpose of this project is to maintain the stock list details, company's address and inform the stock manager when the action is required to be done. The project avoids excess ordering of the stock and helps to order necessary stock when the stock goes beneath the minimum level specified.When the particular product reaches minimum level then it brings a notice to the stock manager.

### 1.2 Product scope:

The scope of the product is to maintain stock in a stationary stock company to calculate the available quantity of a particular product present in the stock. It's easy to add to the stock list, when there is an arrival of a new product to the super market.When a particular product reaches a minimum level then automatically it types letter with the required quantity of product to the product company.

### 1.3. DOCUMENT CONVENTIONS

- Administrator-who supervises the system and maintains the database.
- Customer-who purchases the product.

### 1.4. REFERENCES

www.scribd.com/doc/23362865/srs-of-stock-maintenance

## 2. OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

The system is updated for every change in the stock exchange. It contains the exact rate, date and time of the stock being purchased and sold.

### 2.2 PRODUCT FUNCTIONS:

It secures the information of the stock exchanges and records the date and time of the exchanges. It maintains the details of purchased stocks and sold stocks separately for the purpose of better clarity and to avoid clashes between data.

### 2.3. TOOLS TO BE USED:

Visual Basic 6.0 and Microsoft Access 2003

## 3. EXTERNAL INTERFACE

### 3.1 HARDWARE INTERFACES

The stock details entered through keyboard

### 3.2 SOFTWARE INTERFACES

The system uses ODBC drive to connect and control the database.

## 4. SYSTEM FEATURES

### 4.1 SYSTEM DESCRIPTION AND PRIORITY

This system allows the employee to record all the details that occur during the delivery of product.

### 4.2 STIMULUS/RESPONSE SEQUENCE

The cost of delivered products is calculated.

### 4.3 FUNCTIONAL REQUIREMENT

Req-1: The cost of each product must be calculated.

Req-2: Each product must be given an unique id.

## 5. OTHER NON-FUNCTIONAL REQUIREMENTS

### 5.1 SECURITY REQUIREMENTS

This system is secured and it can only be accessed using password.

### 5.2 PERFORMANCE REQUIREMENTS

Once the details are furnished, the immediate notification of data must be provided.

### 5.3 SAFETY REQUIREMENTS

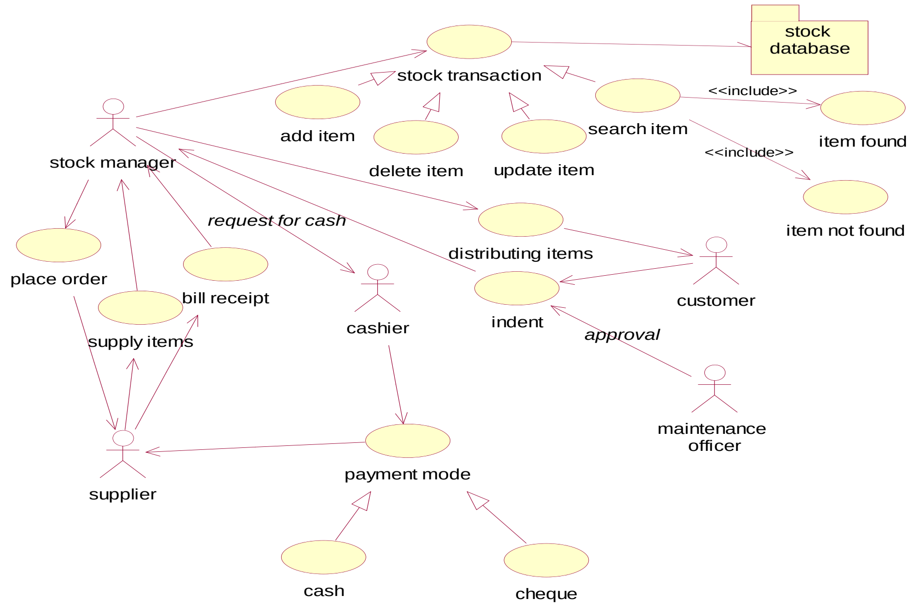
The database must be maintained and monitored effectively to avoid crashing or hacking.

### 5.4 AVAILABILITY REQUIREMENTS

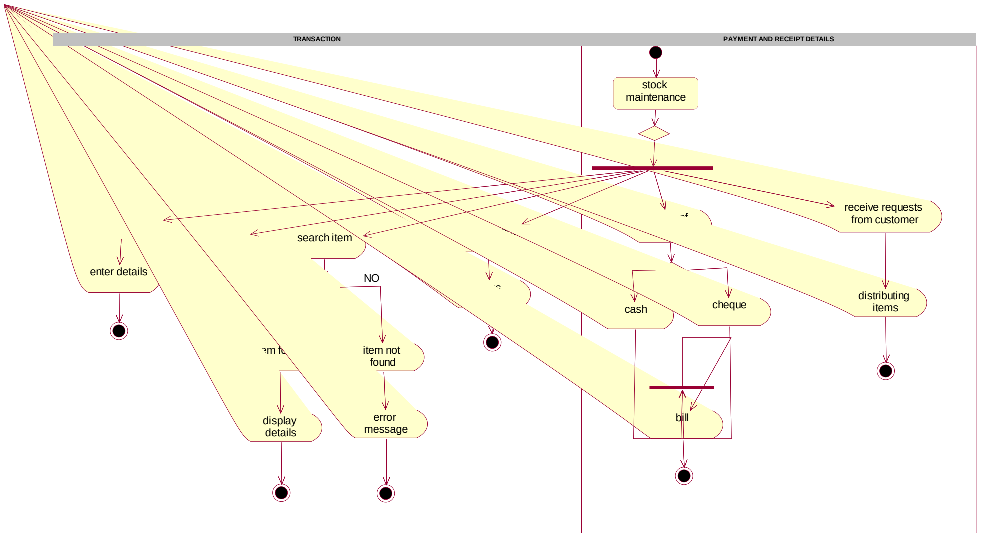
The information provided should be retrieved at any time. Therefore availability of data is more important.

# STOCK MAINTENANCE SYSTEM

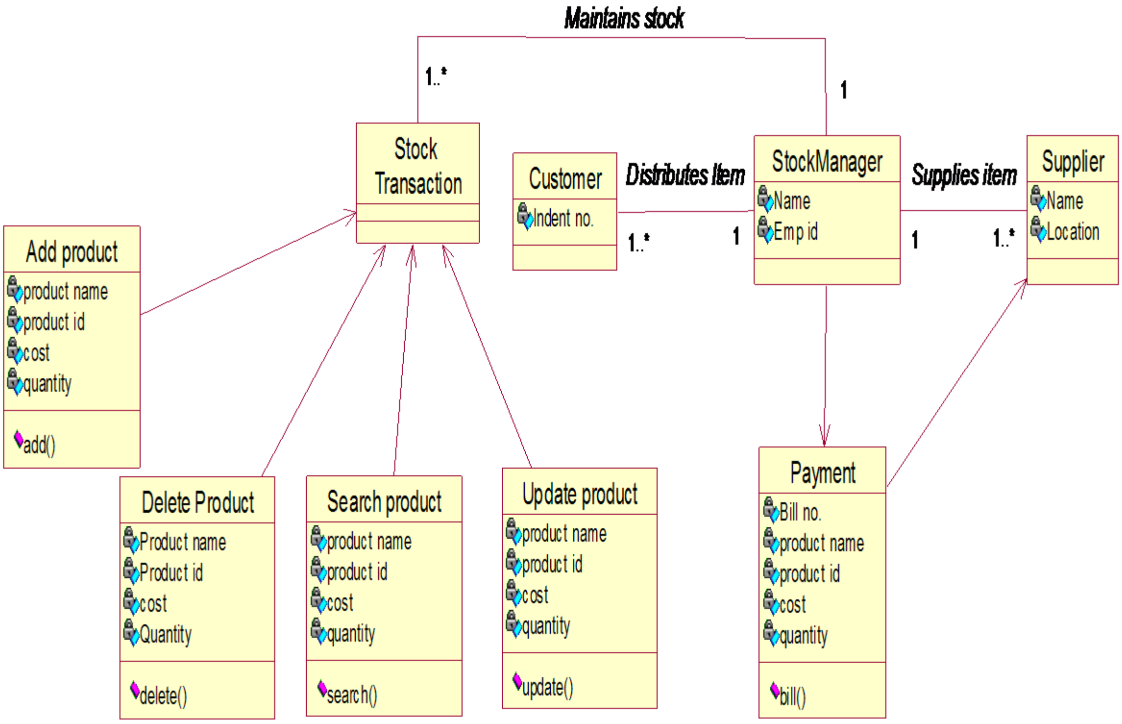
## 1.Use Case Diagram :



## 2.Activity Diagram :

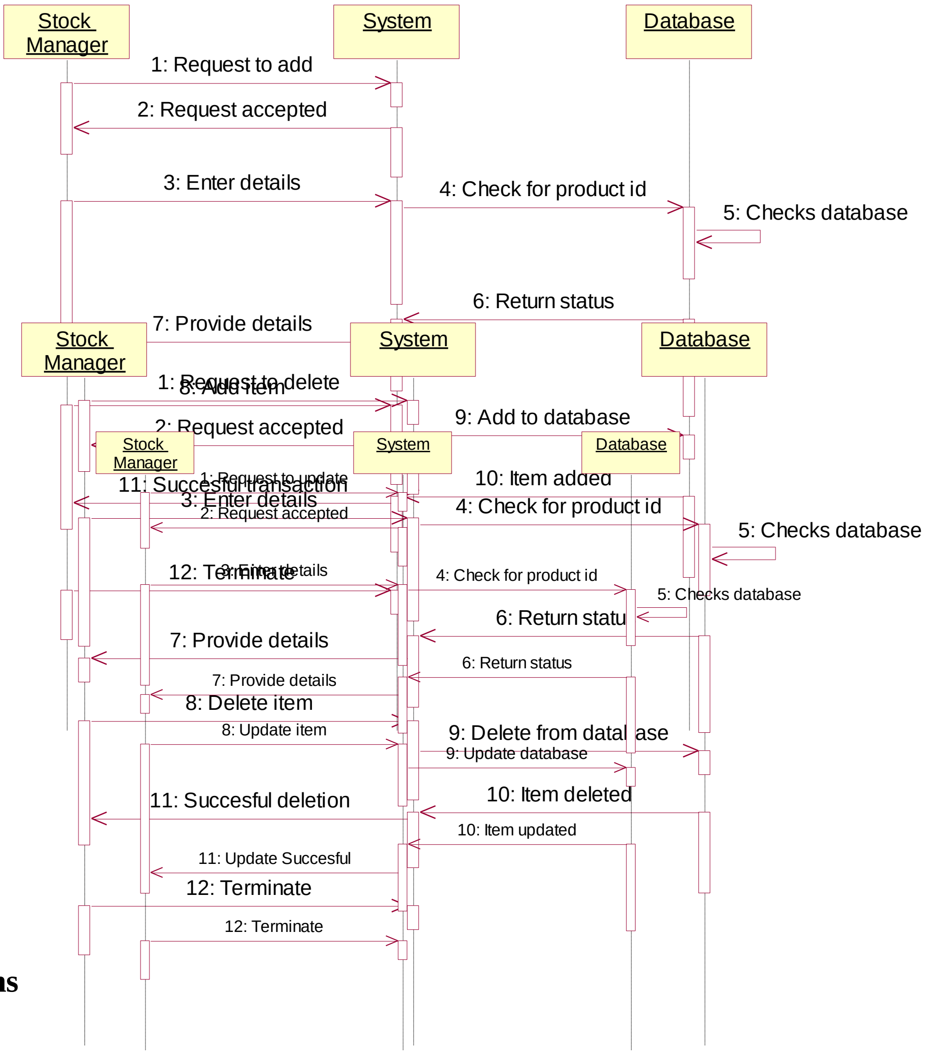


## 3.Class Diagram



## 4.Sequence Diagram :

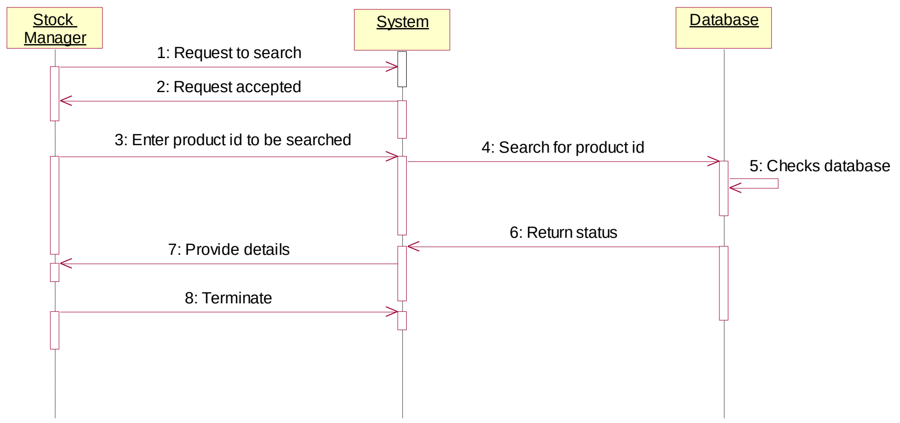
## Add items:



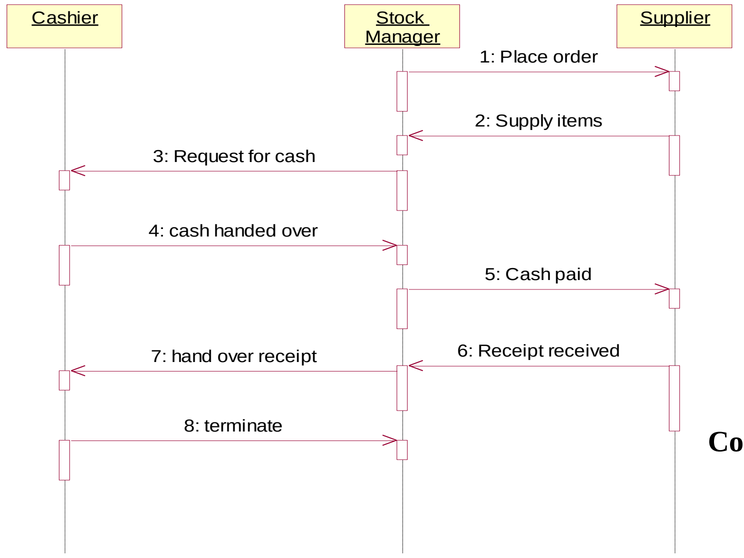
## Delete Items

## Update Items

## Search Items

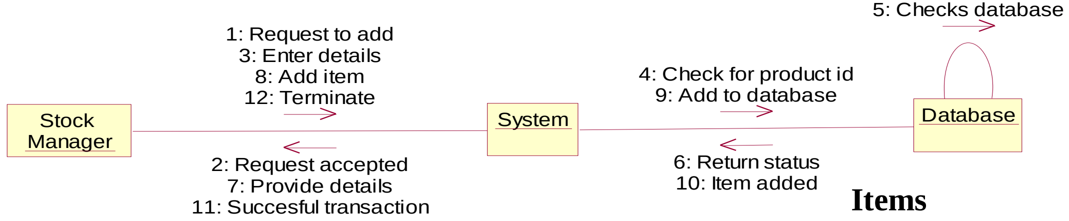


## Payment

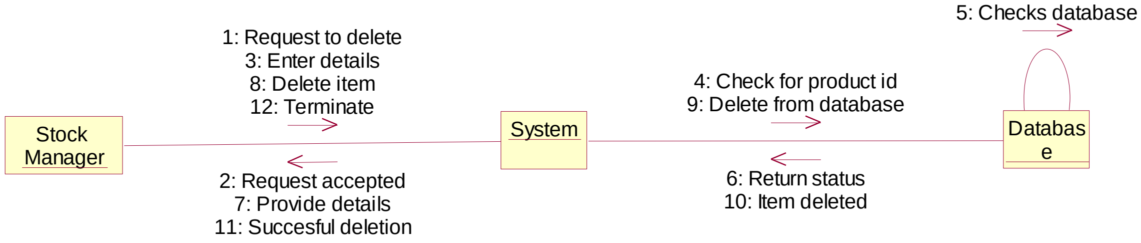


## Collaboration Diagram :

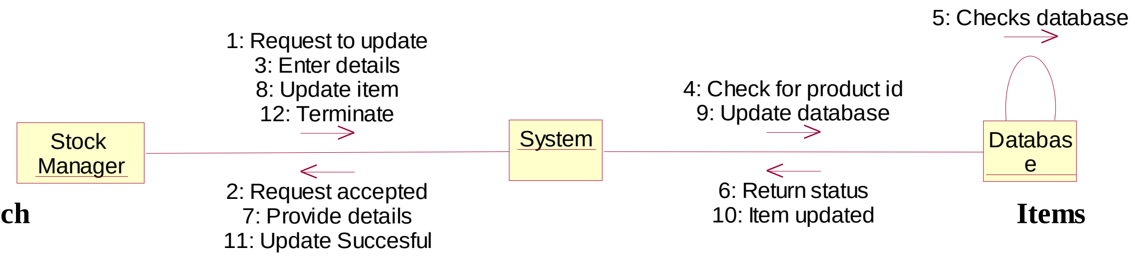
## Add Items



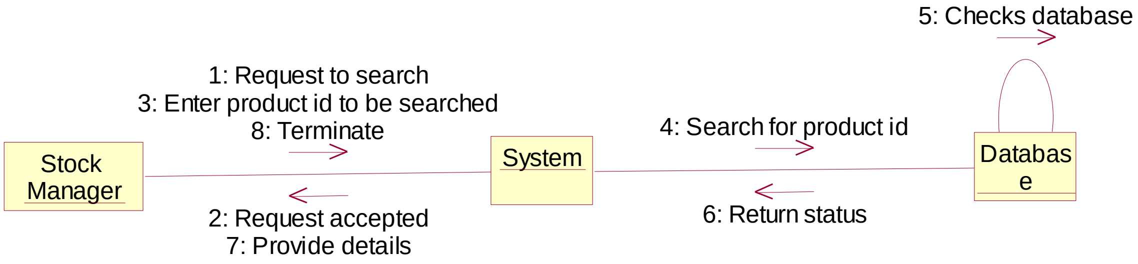
## Delete Items



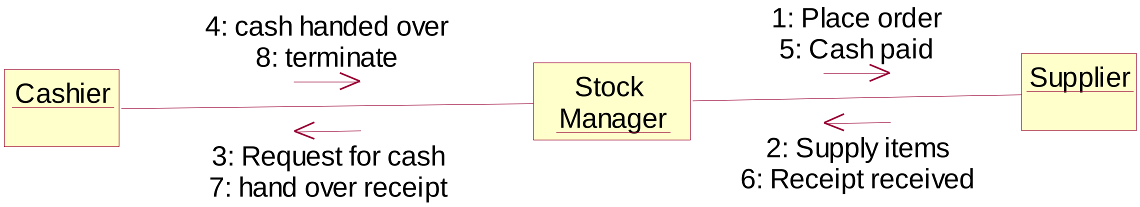
## Update Items



## Search Items

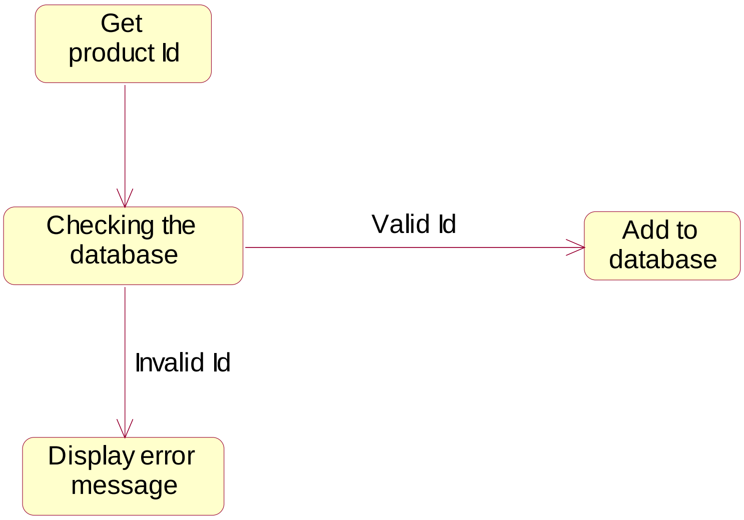


## Payment

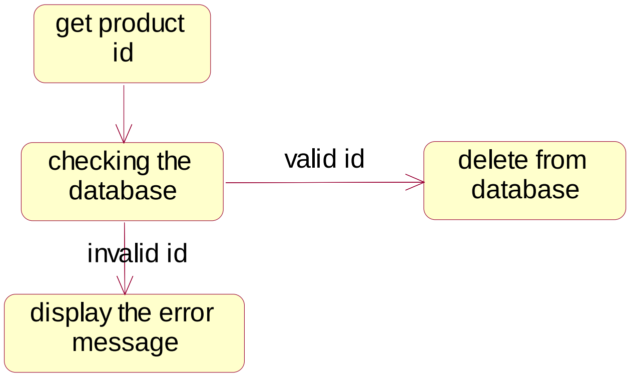


## 5.State chart Diagram :

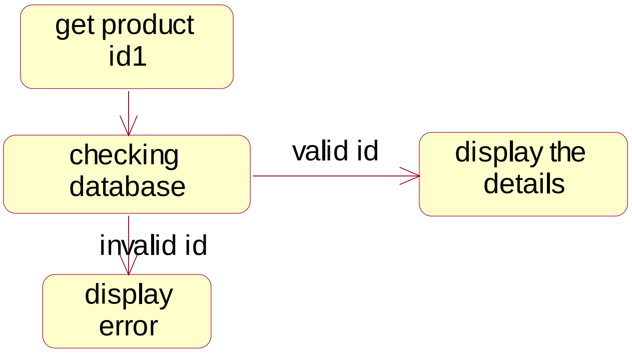
ADD ITEMS



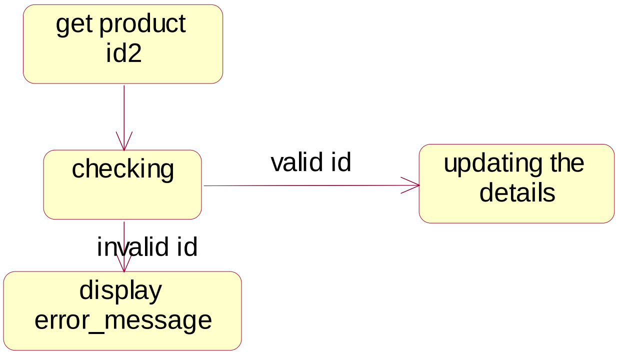
DELETE ITEMS



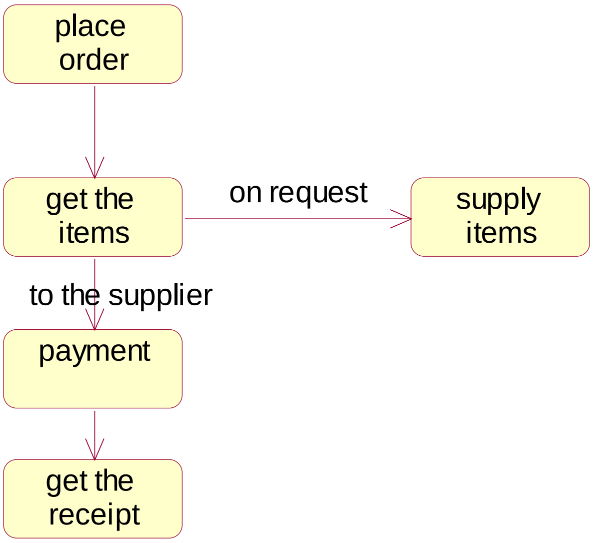
SEARCH ITEMS



UPDATE ITEMS

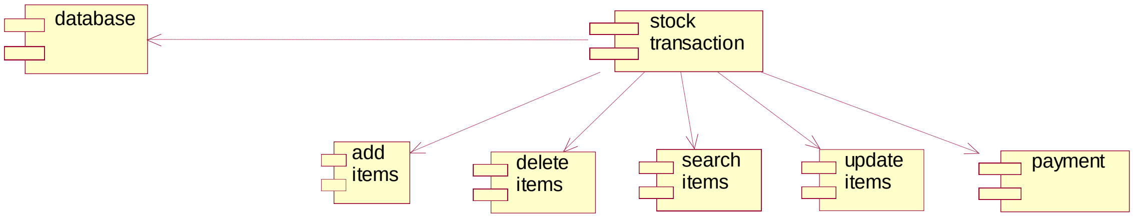


PAYMENT

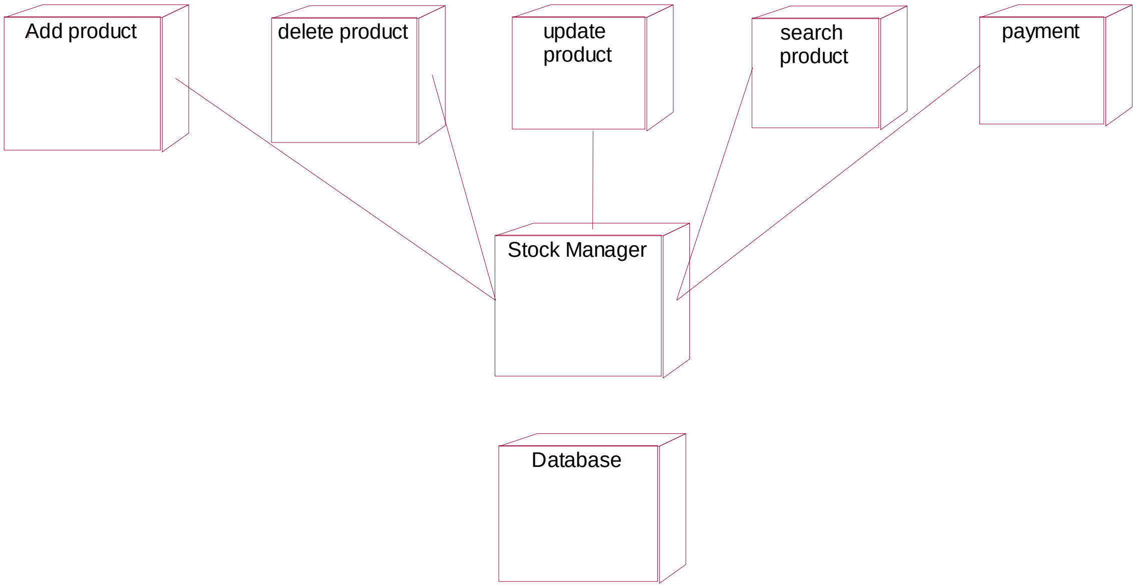


## 6.Component diagram

COMPONENT DIAGRAM



## 7.Deployment Diagram



## Result:

Thus the UML diagrams has been successfully completed for Stock maintenance system.

**Ex.No: 4** **ONLINE COURSE RESERVATION SYSTEM**

**DATE:**

**AIM**

To develop a online course reservationsystem project using argouml tool**.**

**PROBLEM STATEMENT:**

Gives the description about the course and the colleges. The system gives the overall status of the course available in all the college and also it helps to check the availability of the course in a particular college. If the user wants to register the course, an account is required. Depending on the availability of the course, the user can reserve their required course. For reservation, the user details like name,percentage,mail id,phone number,address are needed. And also the system checks for the eligibility which depends on the percentage obtained by the student. It follows some criteria. If the criteria is not satisfied, the reservation is not possible and its displays them as a non-eligible candidate. After reserving the course register number is displayed to confirm the details of the student.

## SOFTWARE REQUIREMENT SPECIFICATION:

### Table of contents

## 1. INTRODUCTION:

Student admissions are a vital part of any university’s running because students are what keep a University alive. The student admission is one of the most important activities within a university as one cannot survive without students. A poor admissions system can mean fewer students being admitted into a university because of mistakes or an overly slow response time.

The process begins with a potential student completing an application form through the Universities and Colleges Admissions Service, the first step for students is to apply directly to the university through a custom online form.

The next step is for the Admissions service center has to review the application and ensure that all of the required information has been provided, from the form itself to the supplementary documentation, such as language and degree certificates. If any of the required information is missing, it is the secretary for the department to which the application concerns that contacts the potential student and arranges for the delivery of the outstanding data

The application in its entirety is then forwarded, complete with a recommendation, to the respective department's Admissions Tutor, who has the final say as to whether each potential student is accepted or rejected. Before making a decision, the Admissions Tutor reviews the application and the additional documentation, comparing the academic credentials to a list of university rankings and previous, similar applications.

### 1.1. PURPOSE:

The purpose of this SRS document is to specify software requirements of the Online Admission for the university. It is intended to be a complete specification of what functionality the admission provides. The main purpose of the system is to automate the task carried out by different peoples in the organization to perform the student admission. Specific design and implementation details will be specified in a future document.

### 1.2. SCOPE:

This project's aim is to automate the system, pre-checking the inclusion of all required material and automatically ranking each student's application based on a number of criteria. These criteria include the ranking of their university, their grade at said university and their language grade certificate. The data used by the system is stored in a database that will be the centre of all information held about students and the base for the remainder of the process after the initial application has been made. This enables things to be simplified and considerably quickened making the jobs of the people involved easier. It supports the current process but centralizes it and makes it possible for decisions to be made earlier and easier way.

### 1.3. DOCUMENT CONVENTIONS:

Student – The person who books the ticket.

### 1.4. REFERENCES:

www.annauniv.edu

## 2. OVERALL DESCRIPTION:

### 2.1. PRODUCT PRESPECTIVE:

- The basic page layout is created using visual basic is the user interface for the student side.
- The Client Software is to provide the user interface on system user client side and for the server a dedicated database is used to store and retrieve the information.
- On the server side ODBC server is used for storing the information.

### 2.2 PRODUCT FUNCTIONS:

This system functions with a database at the back end, for reserving seats.

### 2.3 TOOLS TO BE USED:

Visual basic and Microsoft Access

## 3. EXTERNAL INTERFACES:

### 3.1 HARDWARE INTERFACES:

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWARE INTERFACES:

The system uses ODBC drive to connect and control the database.

## 4. SYSTEM FEATURES:

### 4.1. COURSE RESERVATION:

#### 4.1.1.1 DESCRIPTION AND PRIORITY:

Allow the student to select or block the seats..

#### 4.1.1.2 STIMULUS / RESPONSE SEQUENCE:

The seat is blocked as per the student’s requirement.

#### 4.1.2 FUNCTIONAL REQUIREMENT

REQ 1: student can create its own account.

REQ 2: The system will update the details about the seats available.

## 5. OTHERNON-FUNCTIONAL REQUIREMENT:

### 5.1. PERFORMANCE REQUIREMENT:

To increase the performance, free up database resource for other tasks.

### 5.2. SAFETY REQUIREMENT:

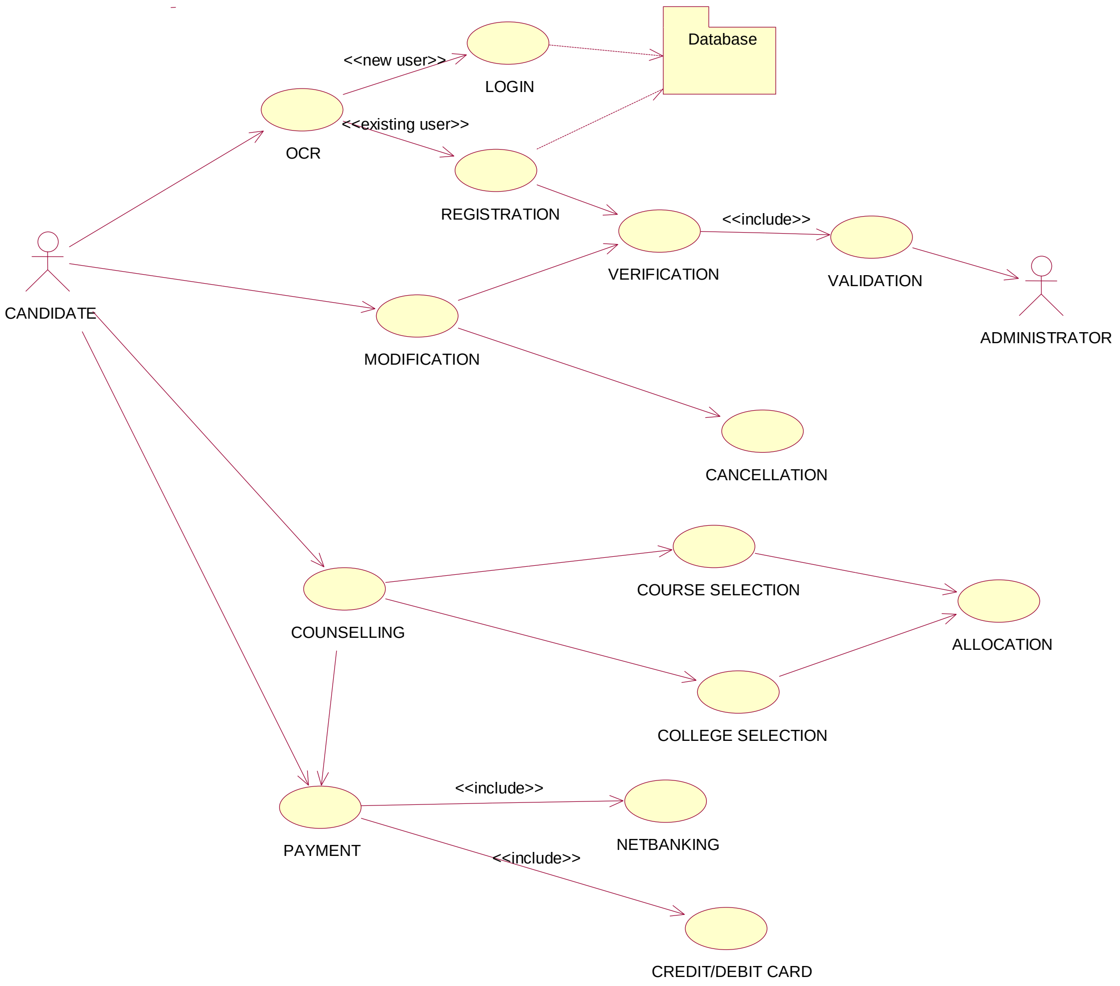
The requirement where datbase main effectively and administartor must

### 5.3. SECURITY REQUIREMENT:

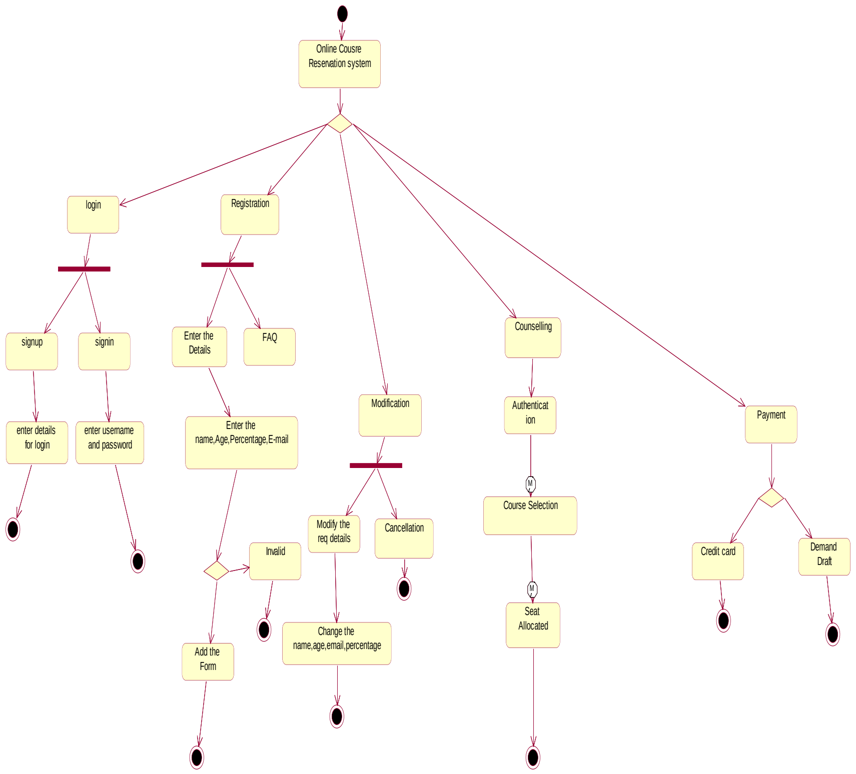
Passwords for registered accounts are stored as a hash in the database only the staff can access the main database.

# ONLINE COURSE RESERVATION SYSTEM

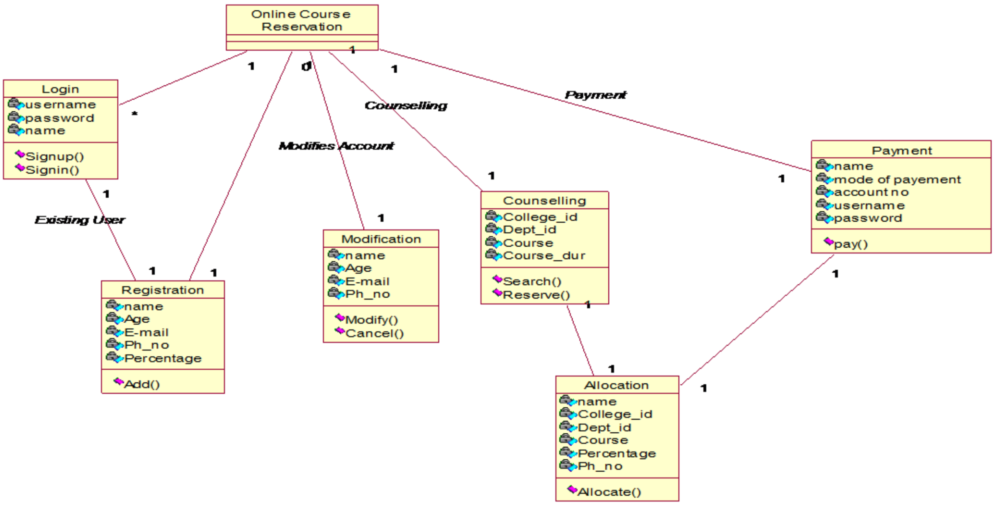
## 1.Use Case Diagram :



## 2.Activity Diagram :

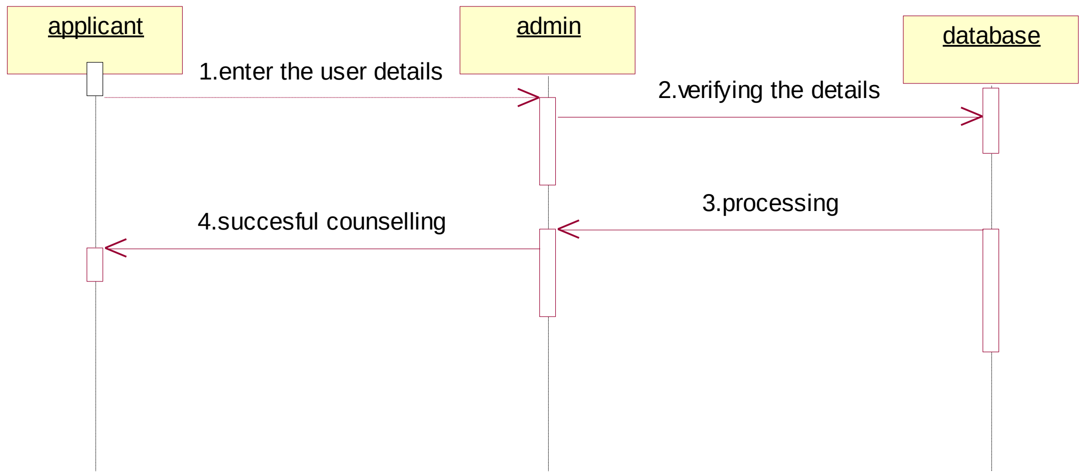


## 3.Class Diagram :

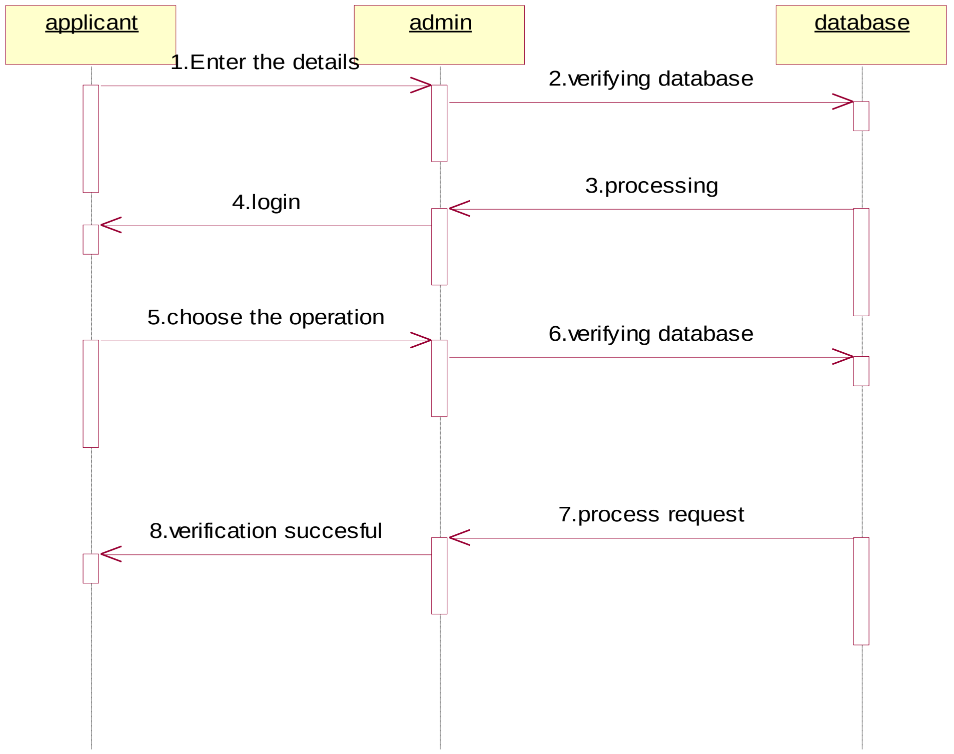


## 4.INTERACTION DIAGRAM:
## (i)SEQUENCE DIAGRAM :

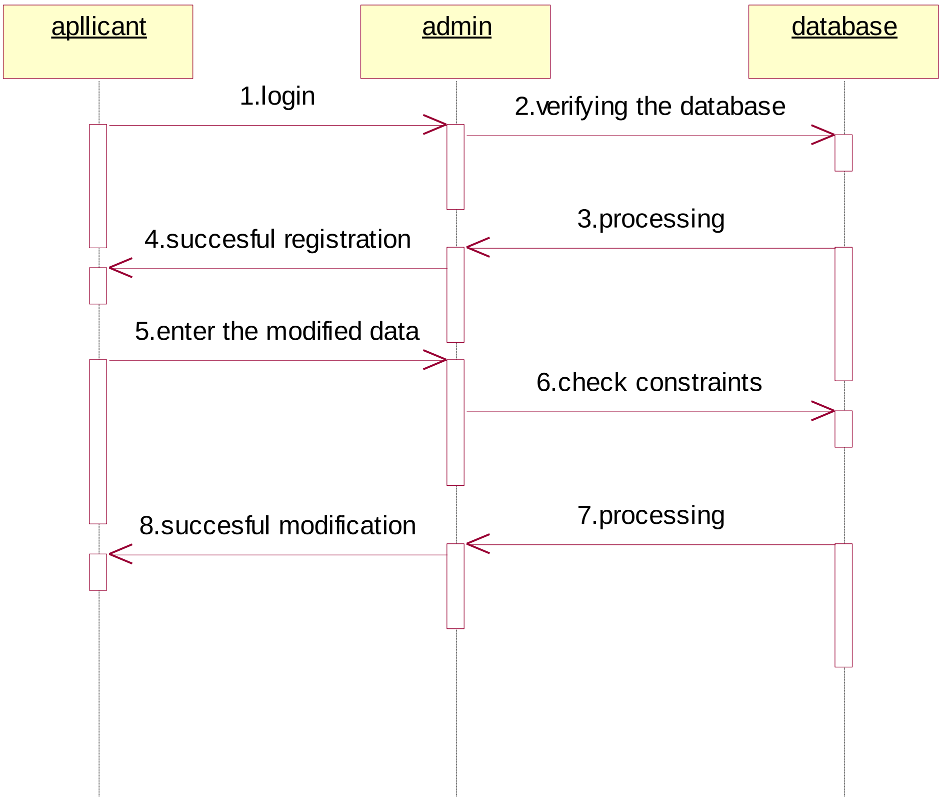
## COUNSELLING:



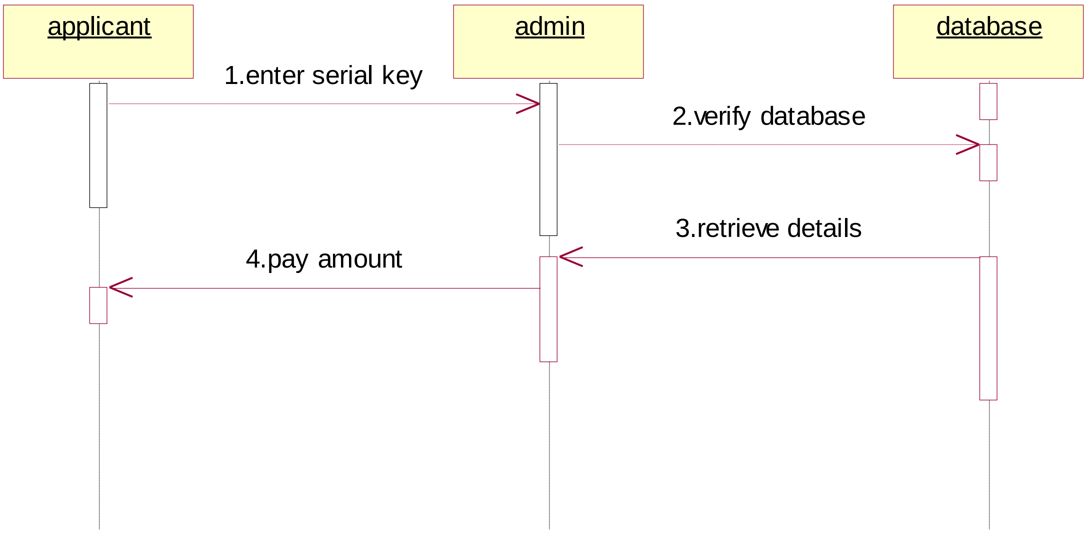
## LOGIN:



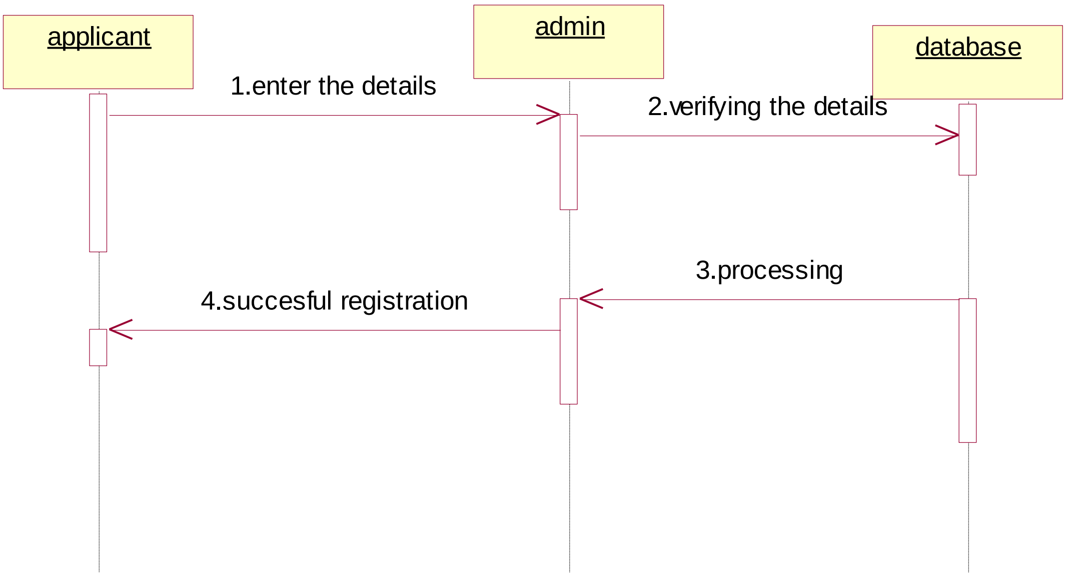
## MODIFICATION:



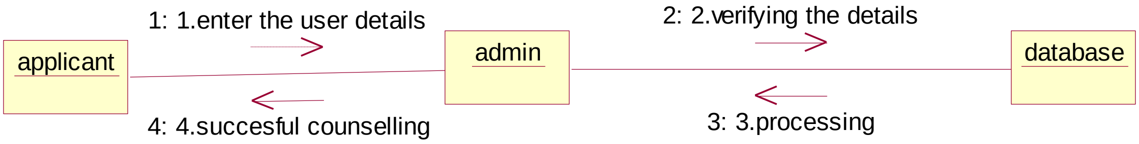
## PAYMENT:



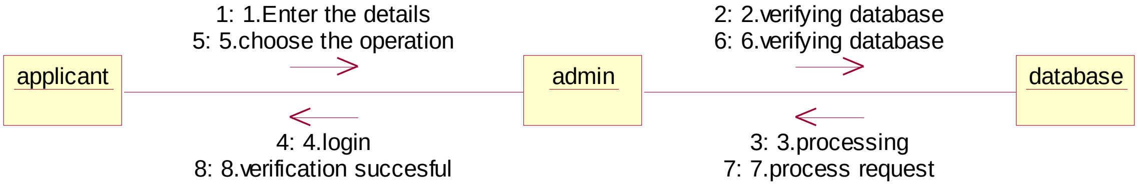
## REGISTRATION:



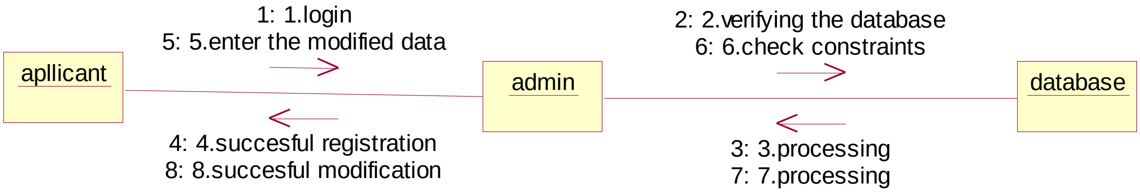
## (ii)COLLABORATION DIAGRAM :
## COUNSELLING:


1: 1.enter the user details
2: 2.verifying the details
applicant
admin
database
4: 4.succesful counselling
3: 3.processing

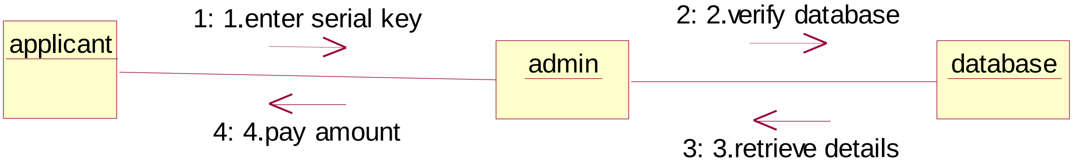
## LOGIN:



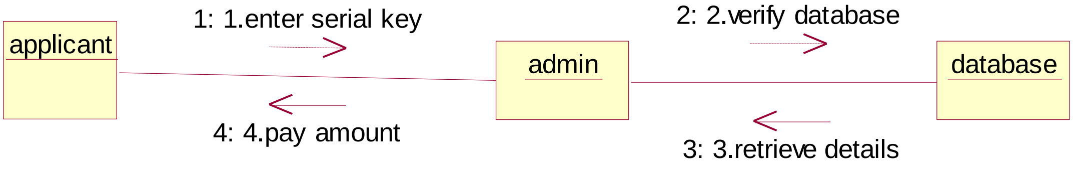
## MODIFICATION:


1: 1.login
5: 5.enter the modified data
2: 2.verifying the database
6: 6.check constraints
apllicant
admin
database
4: 4.succesful registration
8: 8.succesful modification
3: 3.processing
7: 7.processing

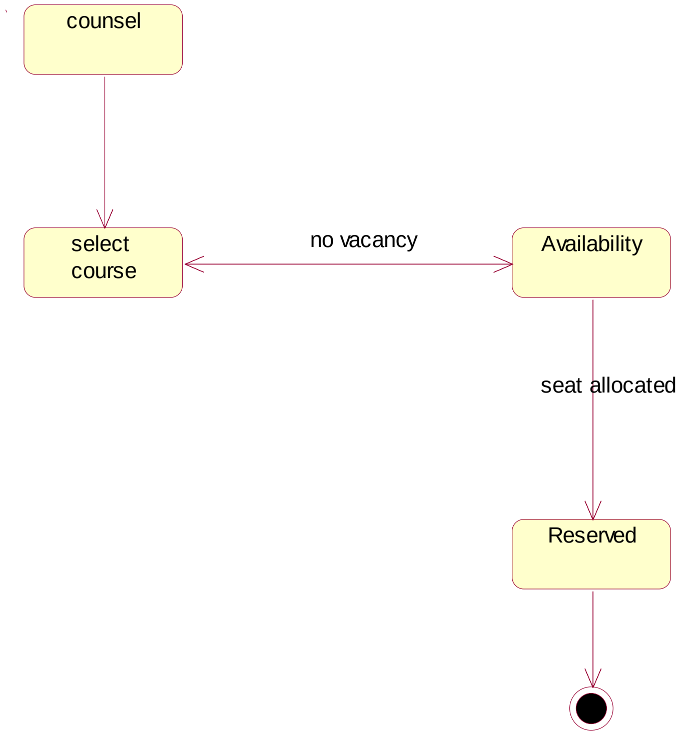
## PAYMENT:



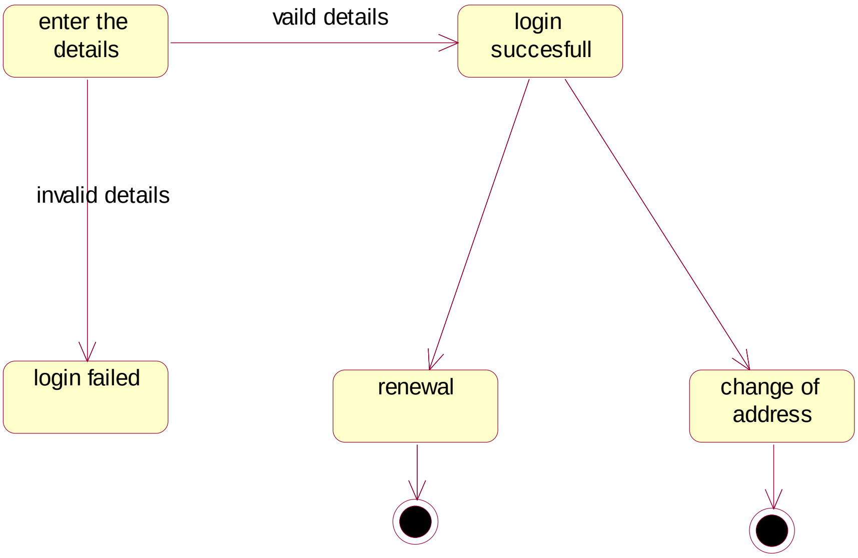
## REGISTRATION:



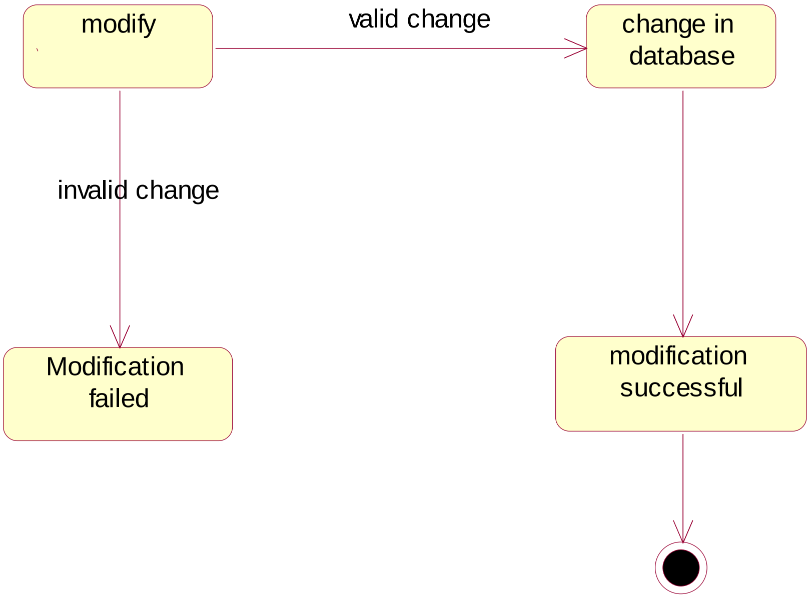
## 5.STATE CHART DIAGRAM:
## COUNSELLING:



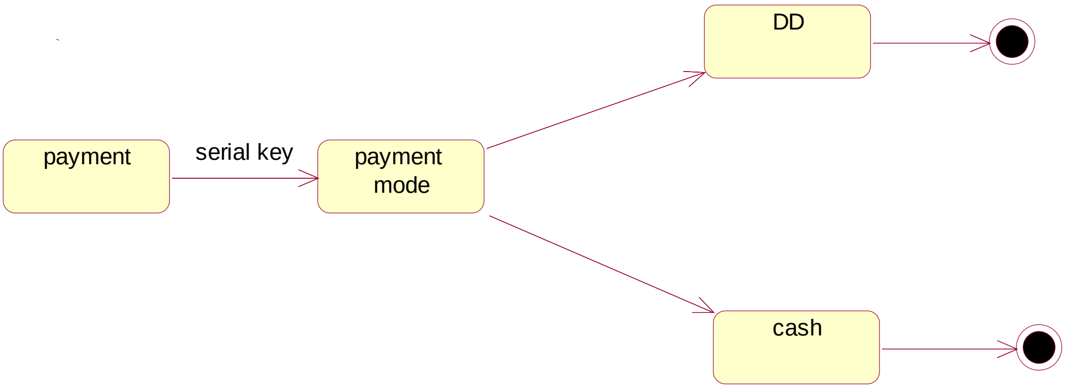
## LOGIN:



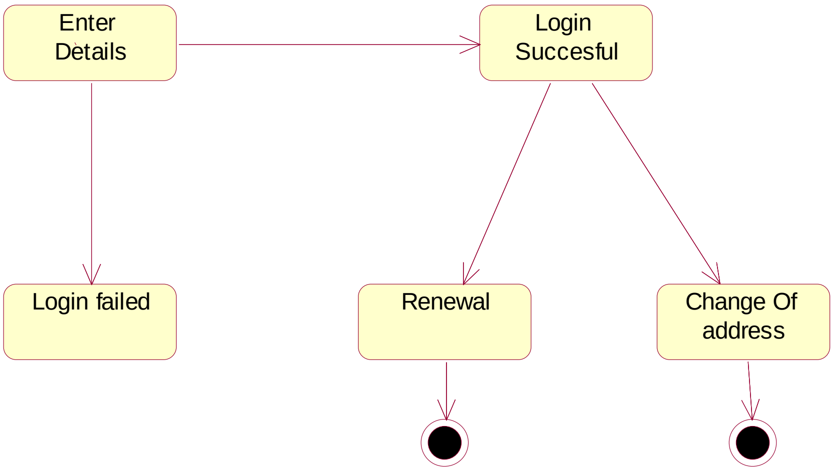
## MODIFICATION:



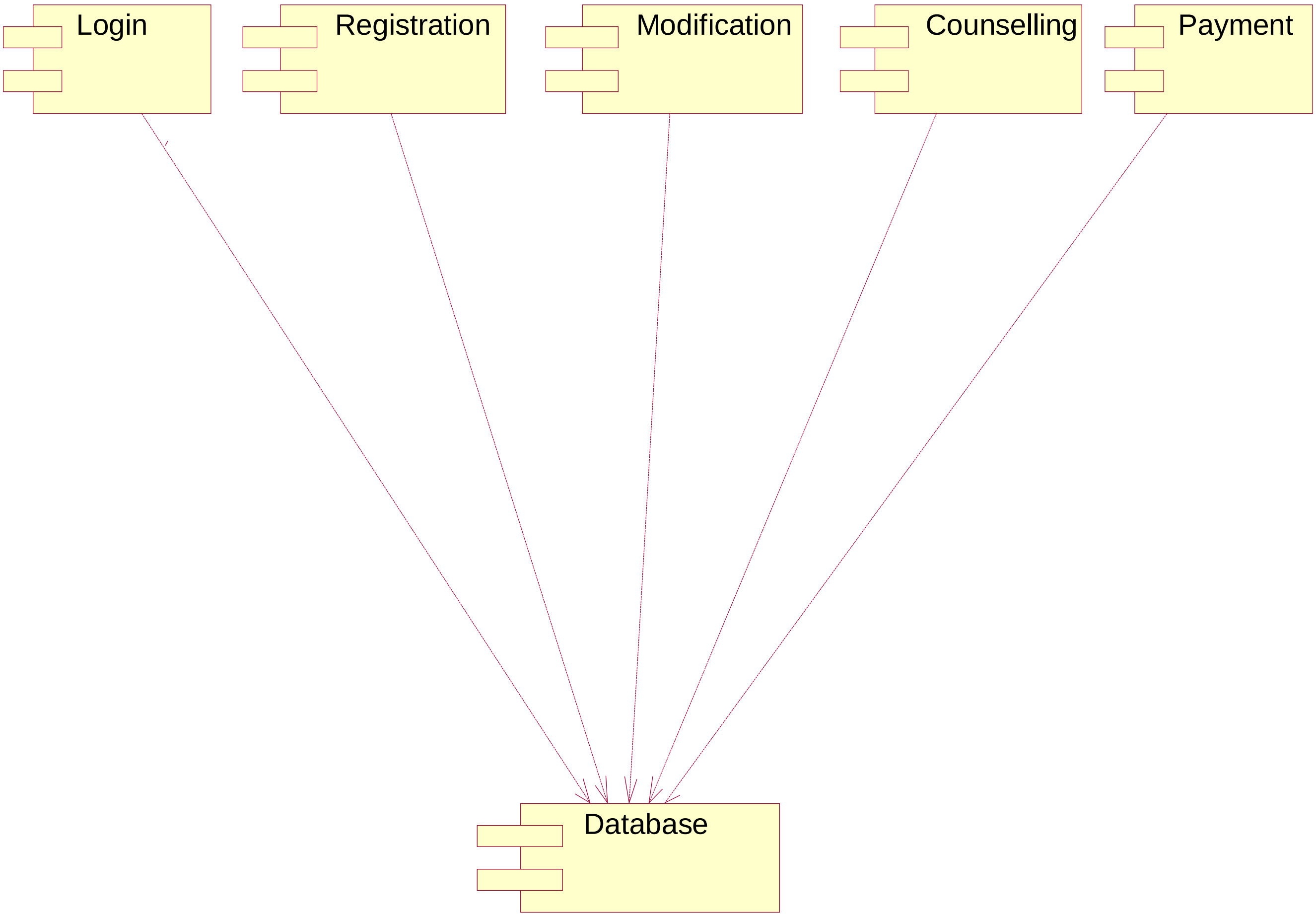
## PAYMENT:



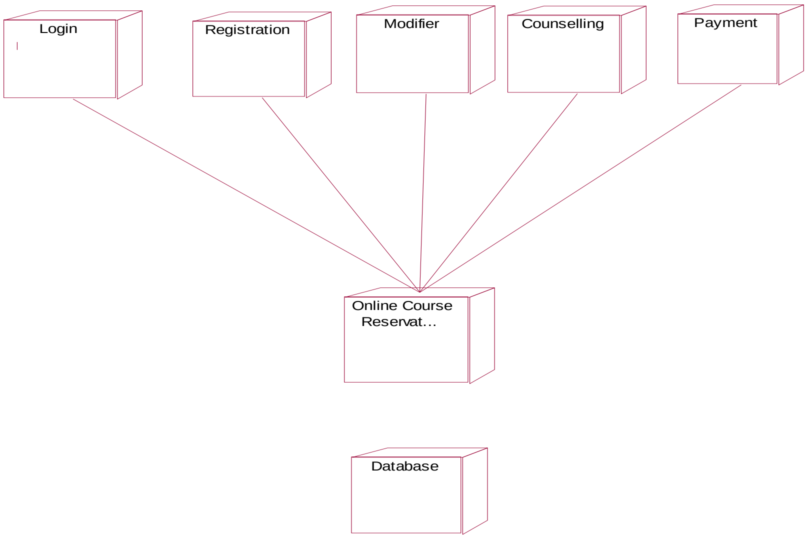
## REGISTRATION:



## 6.COMPONENT DIAGRAM



## 7.DEPLOYMENT DIAGRAM:



## Result:

Thus the UML diagrams has been successfully completed for Online Course Reservation system.

**EX:NO: 5** **SOFTWARE PERSONAL MANAGEMENT SYSTEM**

**DATE:**

**AIM:**

To develop a Software Personal Management System Project using argouml tool.

**PROBLEM STATEMENT**

The software personal management system is system to keep track of the employee details of particular concern. The recruitment status of the company can be maintained. Generally human resource deals with the details of the employee where once the name of employee is typed; the entire history of person is displayed. Multiple accesses to this system allow the authorized manager to update the audit employee files. There are certain folder to maintain the training serious that were given to employee and number of person attending the training. New certification page are provided to log individual employee accreditation. Once there is a change in the employee details, it is updated.

**SOFTWARE REQUIREMENT SPECIFICATION**

TABLE OF CONTENTS

## 1. INTRODUCTION

The software personal management is a system to maintain the records about each employee in a firm. This helps in easy storage and retrieval of detail whenever it is necessary.

### 1.1. PURPOSE

The personal management system is a human resource management system. The system was developed to track employee's attendance and personal management. Employees can be tracks for previous employment history, skill, training, education.

### 1.2. SCOPE

This system tracks the employee attention and personal history. It also helps human resource department to develop and maintain an integrated organisation.

### 1.3. DOCUMENT CONVENTION

- Managers – The person who maintains the employee detail record
- Employee- The person who works in a firm and whose detail are to be kept track.

### 1.4. REFERENCES:

www.scribe.com/software personal maintenance system.

## 2. OVERALL DESCRIPTION

### 2.1. PRODUCT PERSPECTIVE:

This project is a self contained project designed for effective maintanence of an organization.

### 2.2. PRODUCT FUNCTIONS:

There are two main contents,

i) Front end – Java net beans, This is the area that all the employee can see

ii) Back end – Database, This area is designed and maintained by the HR.

### 2.3. OPERATING ENVIRONMENT:

Visual Basic 6.0 and Microsoft Access 2003.

## 3. EXTERNAL INTERFACE

### 3.1. HARDWARE INTERFACE

The system requires extensive database to connect with the hardware computer.

### 3.2. SOFTWARE INTERFACE

The system uses ODBC drive to connect and control the database.

## 4. SYSTEM FEATURES

### 4.1. EMPLOYEE TRACKING

#### 4.1.1. SYSTEM DESCRIPTION AND PRIORITY

The system allows to track the details and history of a employee.

#### 4.1.2. STIMULUS/RESPONSE SEQUENCE

The tracking is done in order to inverse the performance of organization.

#### 4.1.3. FUNCTIONAL REQUIREMENT:

Req1: The employee id and name must be entered.

Req2: The system must be updated.

## 5. OTHER NON FUNCTIONAL REQUIREMENT:

### 5.1. PERFORMANCE MANAGEMENT:

The detail of employee must be uploaded to increase the performance of the organization.

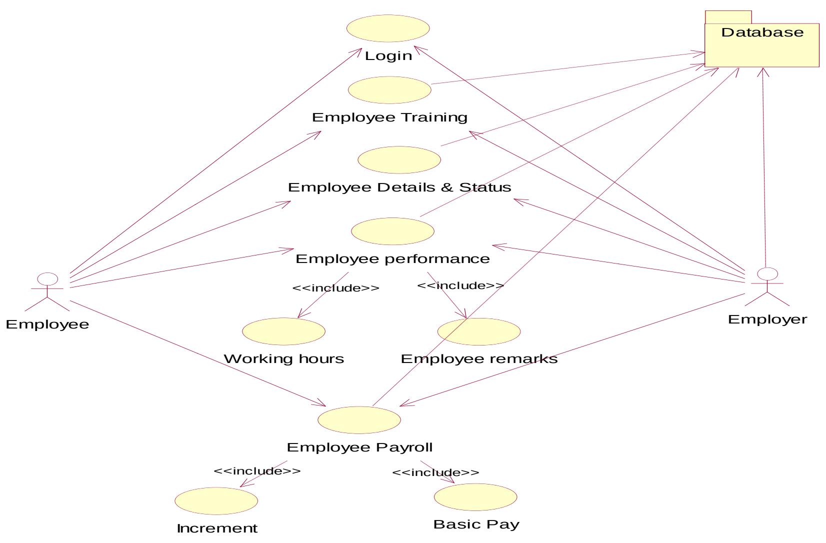
### 5.2. SECURITY REQUIREMENT:

The detail of the employee must be recorded recursively and only the human resource manager must have the authority to access them.

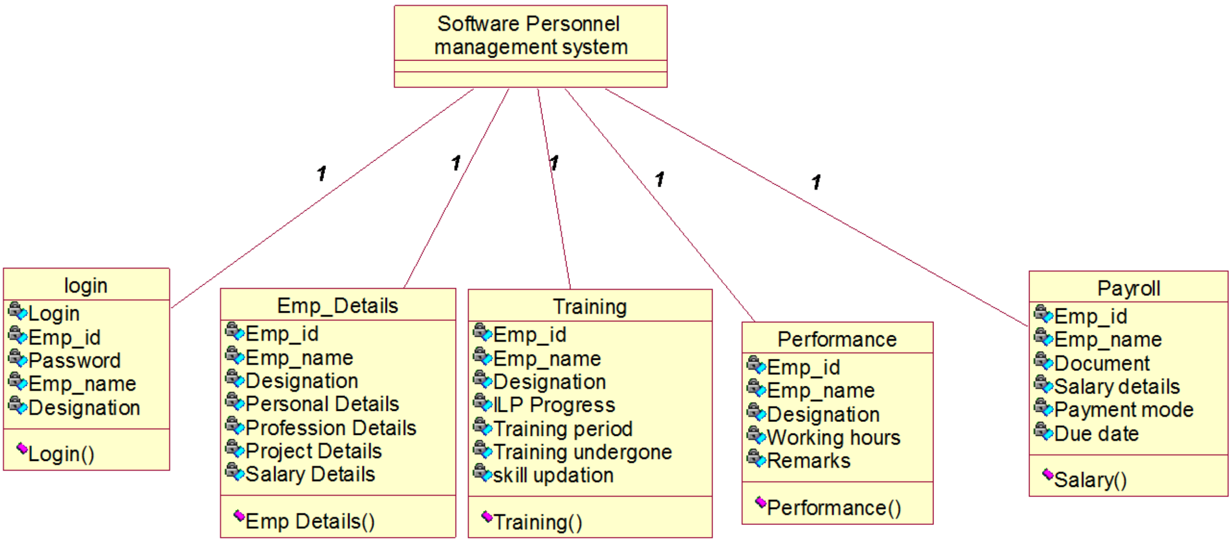
### 5.3. SAFETY REQUIREMENT:

The database must be maintained efficiently without crushing or hacking.

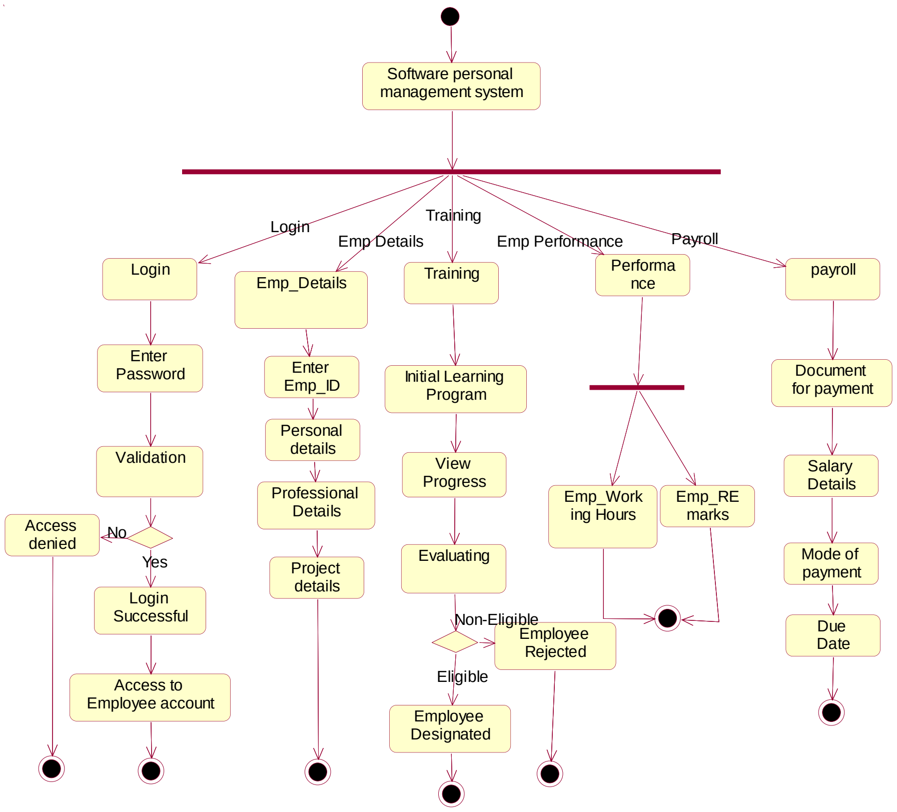
## 1.Use case Diagram:



## 2.Class diagram:



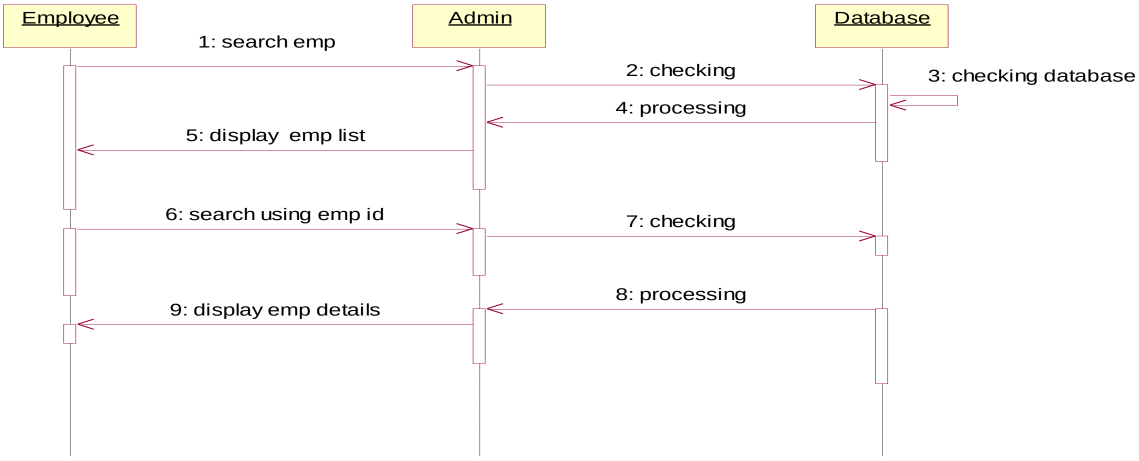
## 3.Activity diagram:



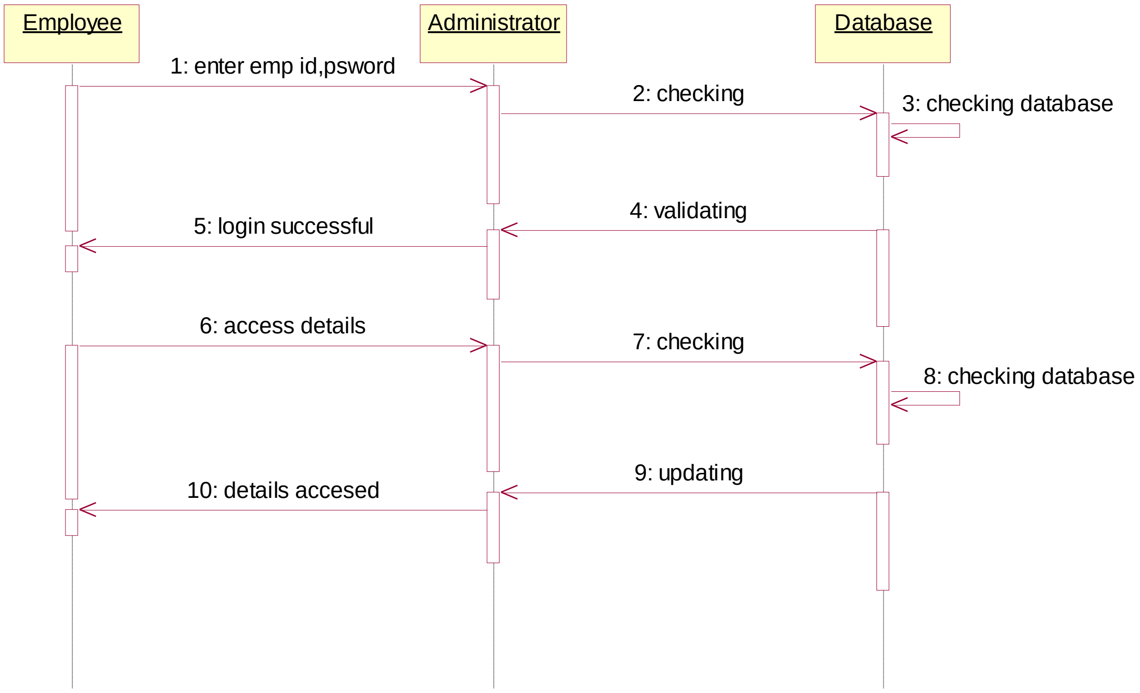
## 4.Interaction Diagram:

## (i)Sequential Diagram:

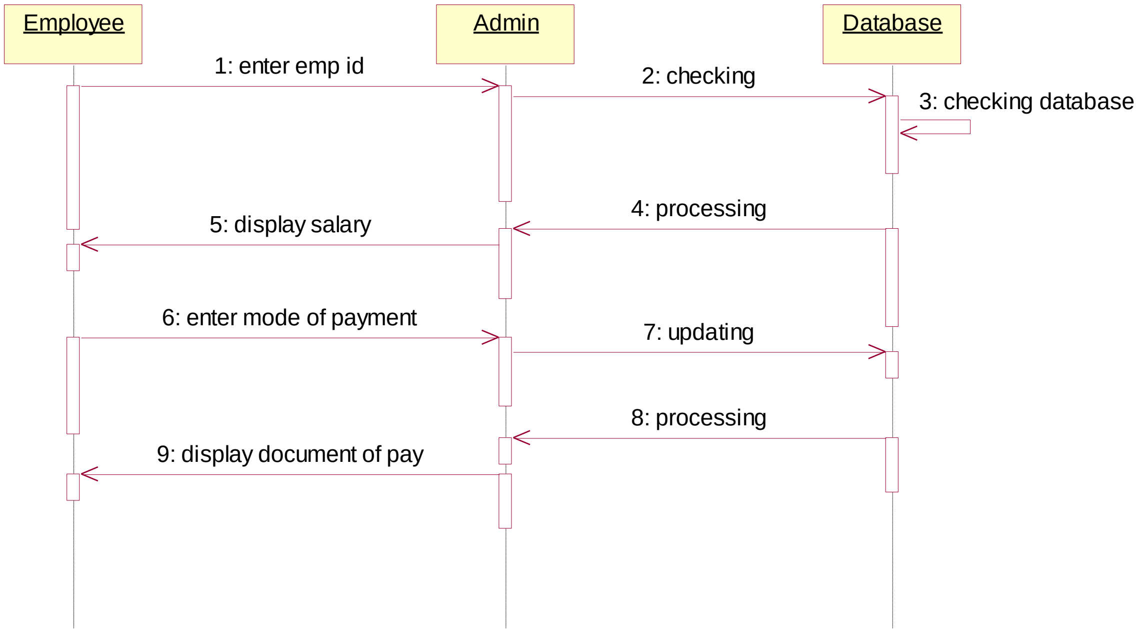
Emp_Details:



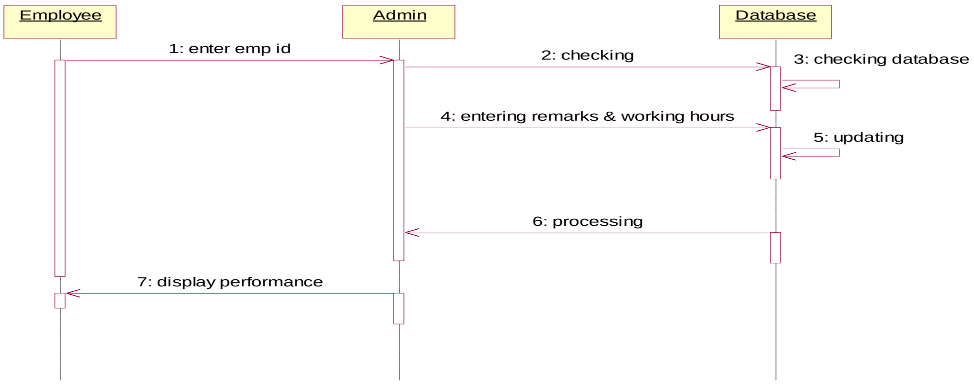
Emp login:



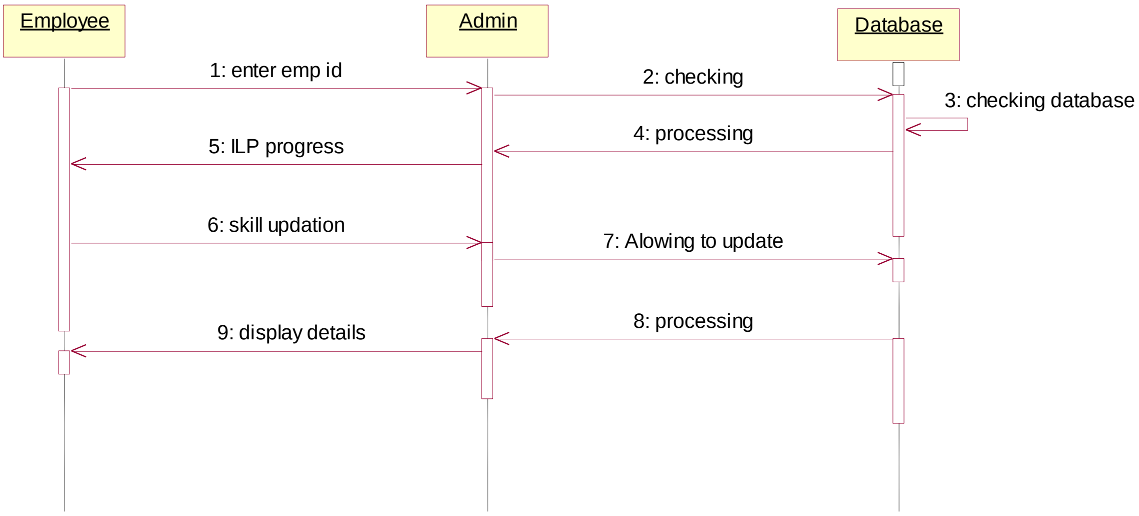
Payroll:



Emp_Performance:

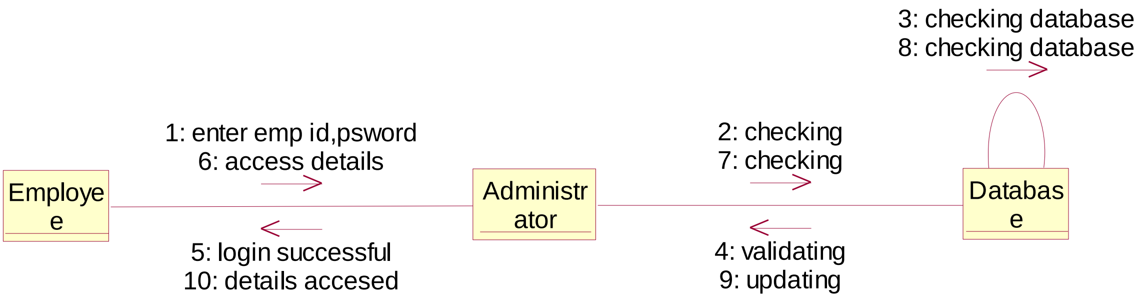


Training:

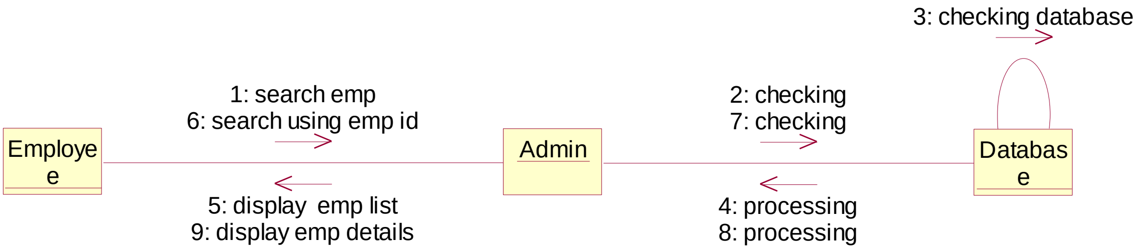


## (ii)Collabaration Diagram:

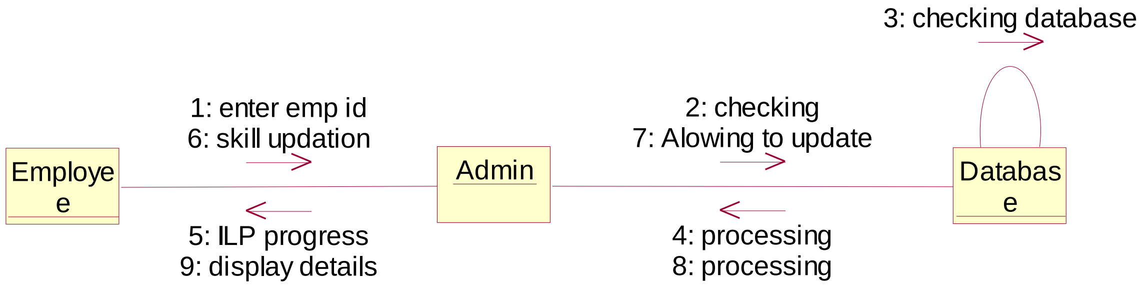
Login:



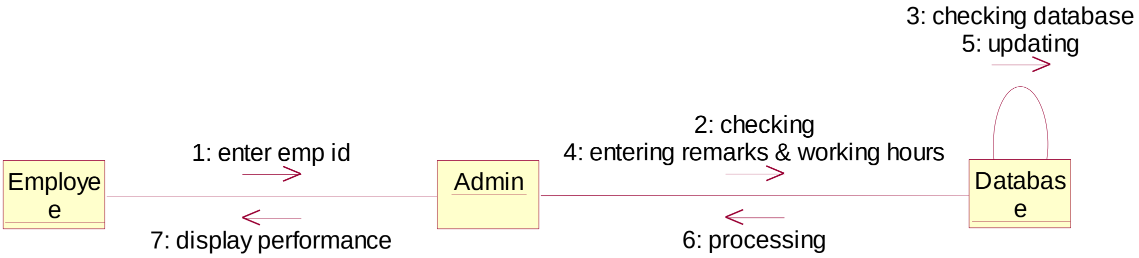
Details:



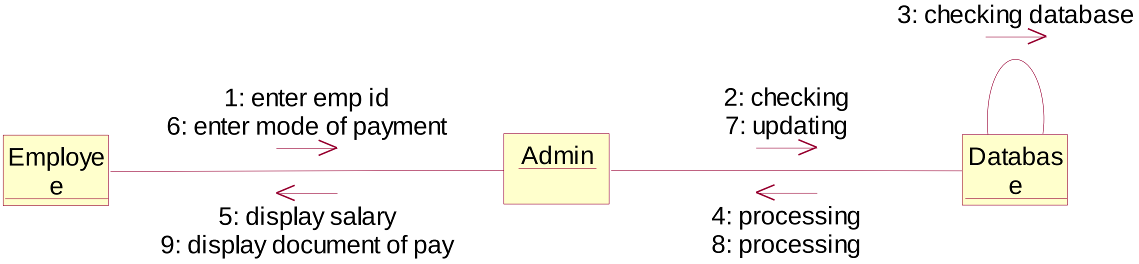
Training:



Performance:

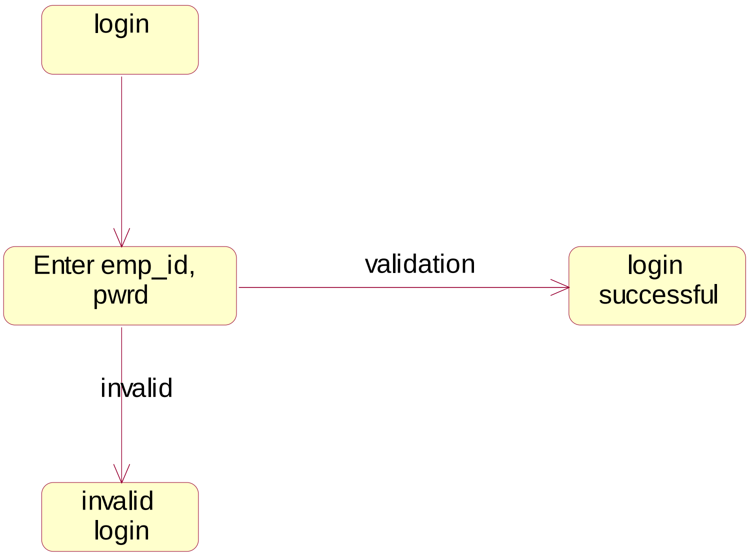


Payroll:

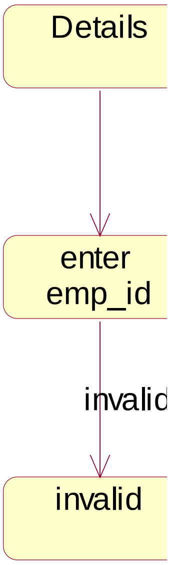


## 5.StateChart Diagram:

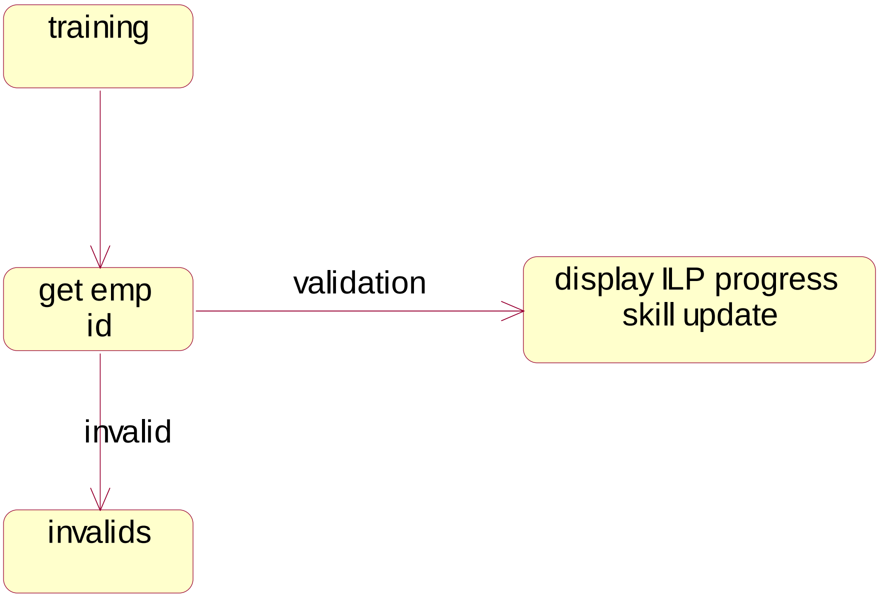
Login:



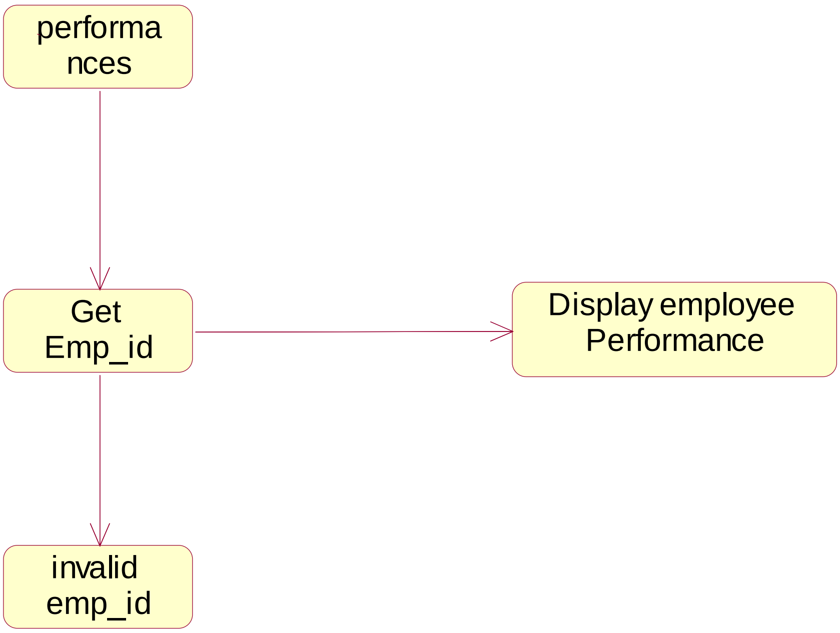
Details:



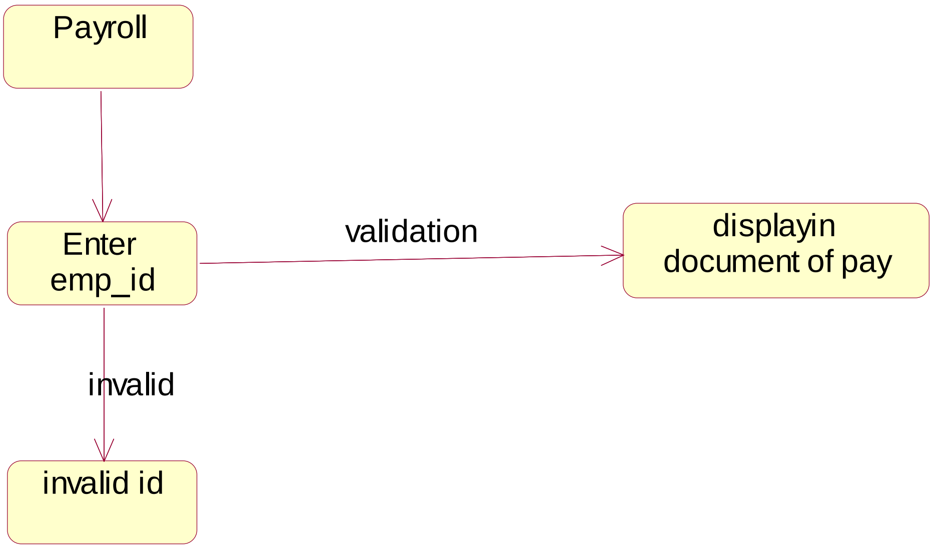
Training:



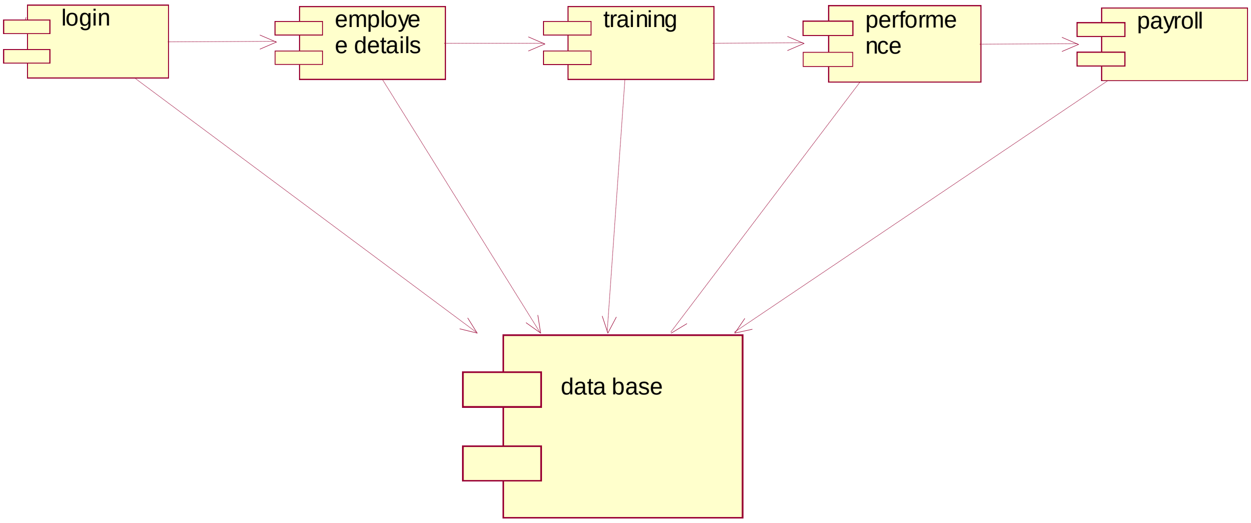
Performance:



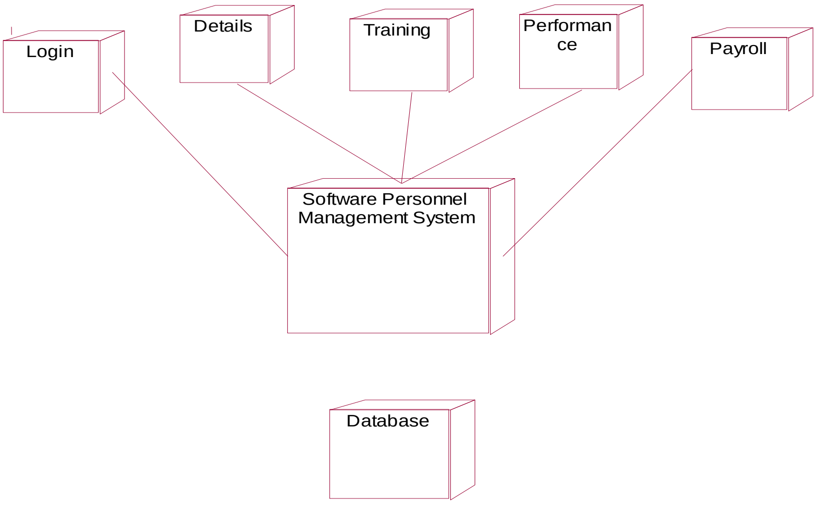
Payroll:



## 6.Component diagram:



## 7.Deployment diagram:



## Result:

Thus the UML diagrams has been successfully completed for Software Personnel

Management System.

## EX.NO : 6 RECRUITMENT SYSTEM

**DATE:**

**AIM:**

To develop recruitment system project using argouml tool**.**

**PROBLEM STATEMENT**

The recruitment system allows the job seekers to enroll their names through the process of registration. The employee also can get the list of available candidates and shortlist for their company requirement. Once the applicant enrolls he receives an id, which helps him in further correspondence. A fees amount is received from the job seekers for enrollment. This system makes the task of the job seeker easier rather than waiting in queue for enrollment. This also reduces the time consumption for both for the job seeker and employee.

**SOFTWARE REQUIREMENT SPECIFICATION**

### TABLE OF CONTENTS

### 1. **INTRODUCTION:**

This project aimed at developing a web-based recruitment system by creating details about vacancies, storing application data and interview process initiation.

### 1.1 PURPOSE:

Manual recruitment is a tedious processing to the increasing number of job seekers. People around the world can apply for the online recruitment system and it is very convenient too. This system is specially designed for those seek the most demanding job and challenging positions in their own field.

### 1.2. SCOPE:

Recruitment system allows the job seekers to find their dream job in the chosen field. This system helps the companies to recruit the right candidate for the job. This system serves as a common meeting ground for job seekers and employees, both locally and globally.

### 1.3. DOCUMENT CONVENTIONS:

Administrator - The person who has the full authority over the system. He can view all registered users.

Job Seekers - the person who registers in the system in search of jobs.

Company - A company can register itself, select the required employees and provide information about the result.

### 1.4. REFERENCES:

www.timesjob.com:

## 2 OVERALL DESCRIPTIONS:

### 2.1 PRODUCT PERSPECTIVE:

This project is a self-contained project for convenient recruitment process.

### 2.2 PRODUCT FUNCTIONS:

The front end of the project is where the visitors or job seeker's do their enrollment. The back end is managed by the administrator effectively

### 2.3 TOOLS TO BE USED:

Visual basic and Microsoft Access

## 3. EXTERNAL INTERFACES:

### 3.1 HARDWARE INTERFACES:

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWARE INTERFACE:

The system uses ODBC drive to connect and control the database.

## 4. SYSTEM FEATURES:

### 4.1 APPLYING FOR JOB:

#### 4.1.1 SYSTEM DESCRIPTION AND PRIORITY:

This feature allows the job seeker to apply for job.

### 4.1.2 STIMULUS AND RESPONSE SEQUENCE:

When the job seeker uploads his resume, the list of jobs matching the resume will be posted.

### 4.1.3 FUNCTIONAL REQUIREMENTS:

REQ1: Only registered users can apply for the job.

REQ2: The Company has the right to recruit the candidates, not the system.

## 5 OTHER NON-FUNCTIONAL REQUIREMENTS:

### 5.1 PERFORMANCE MANAGEMENT:

To increase the performance and free up database resources, the default features are written to cache files on their initial load.

### 5.2 SAFETY REQUIREMENTS:

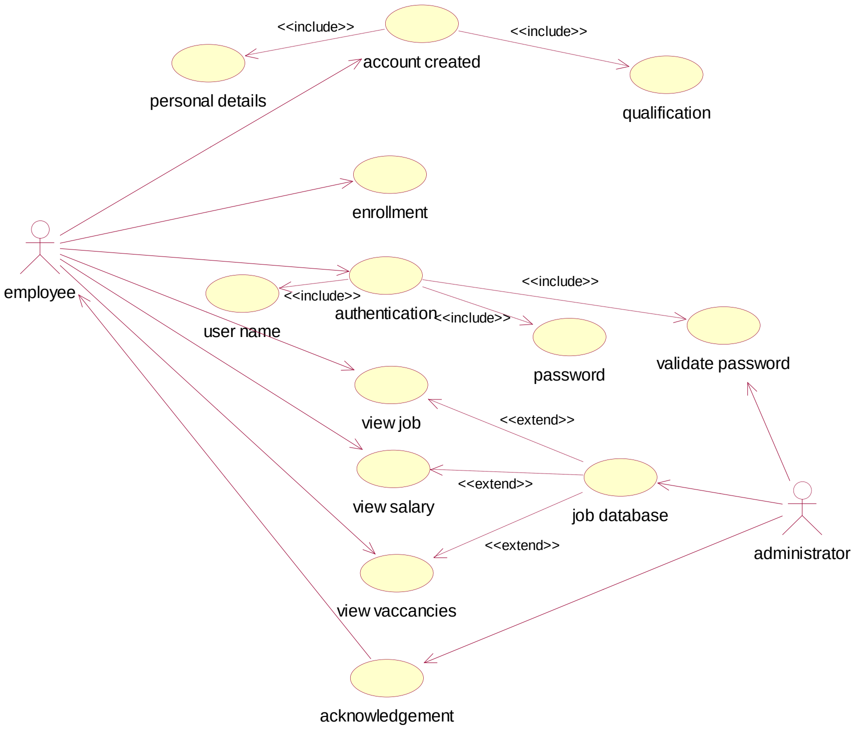
The database must be maintained effective from hacking.

### 5.3SECURITY REQUIREMENTS:

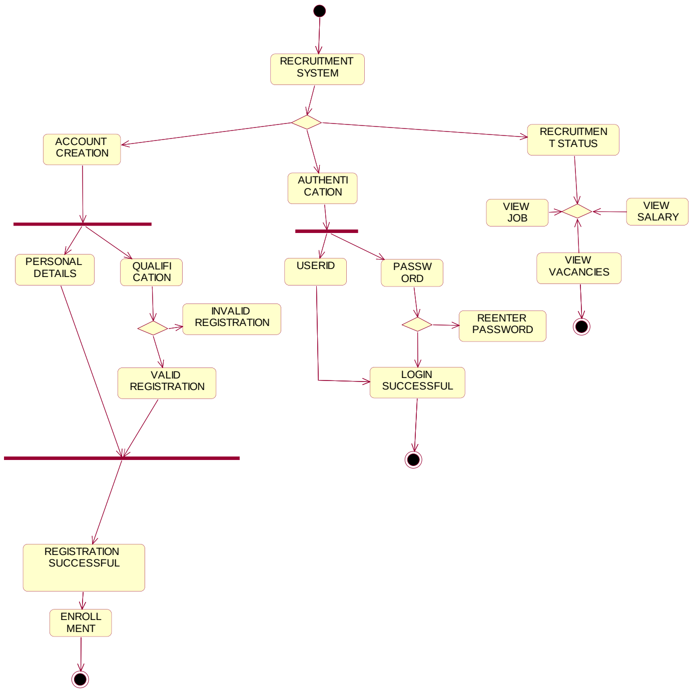
Password for registered accounts are stored securely in a hash, the uploads made by the user must be checked and then accepted.

## RECRUITMENT SYSTEM

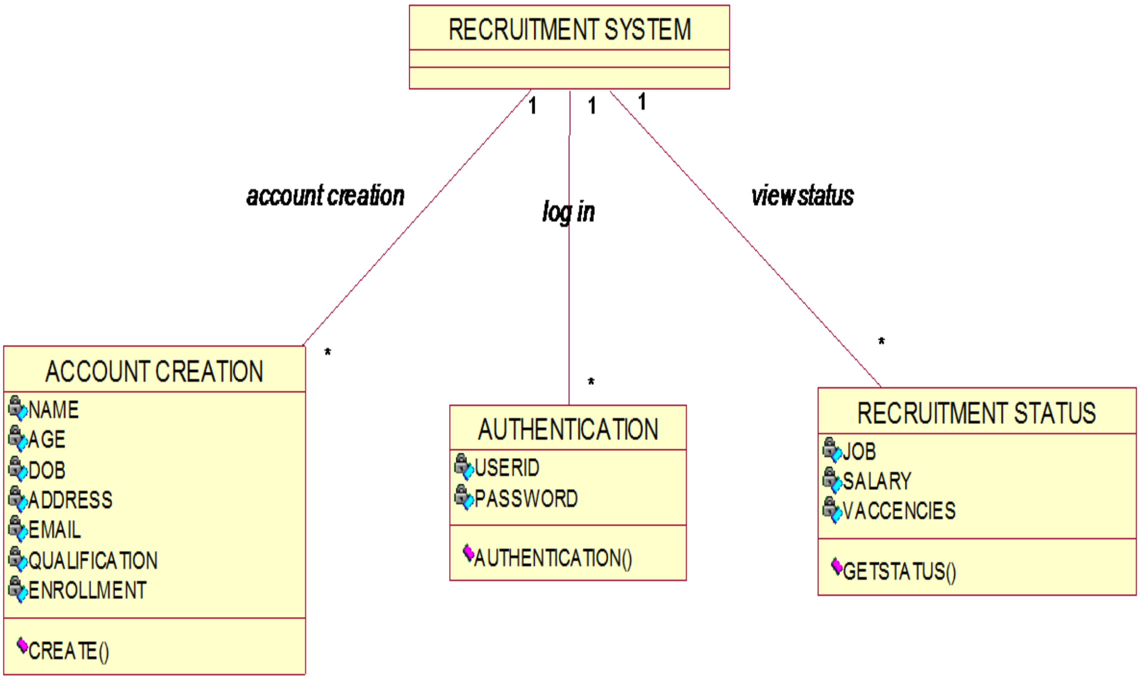
## 1.USECASE DIAGRAM:



## 2.ACTIVITY DIAGRAM:



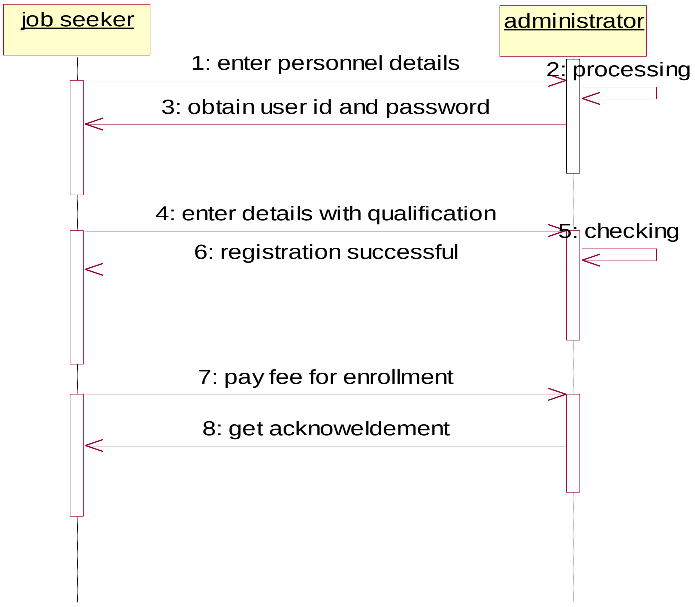
## 3.CLASS DIAGRAM:



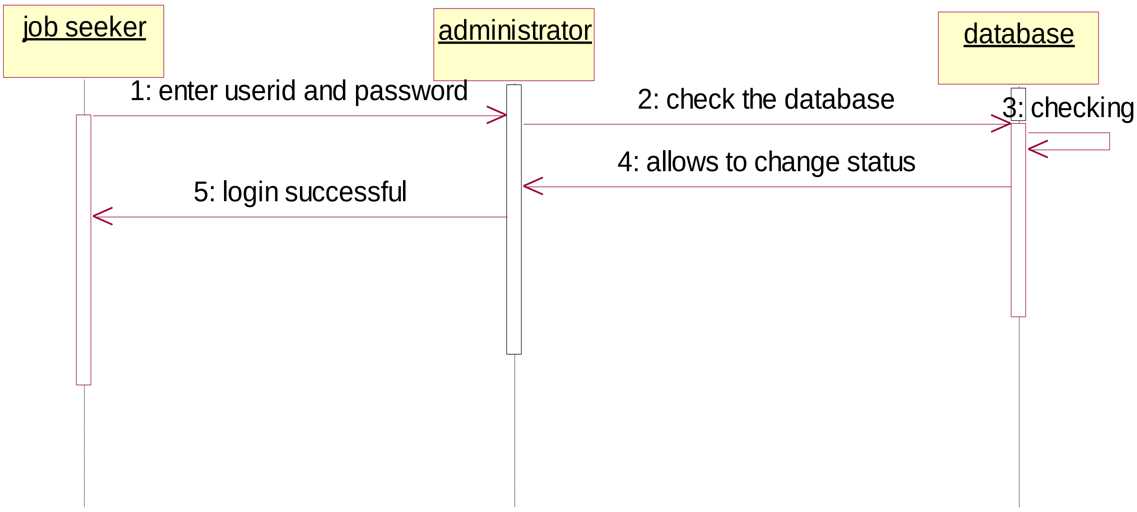
## 4.INTERACTION DIAGRAM:

## (i)SEQUENCE DIAGRAM:

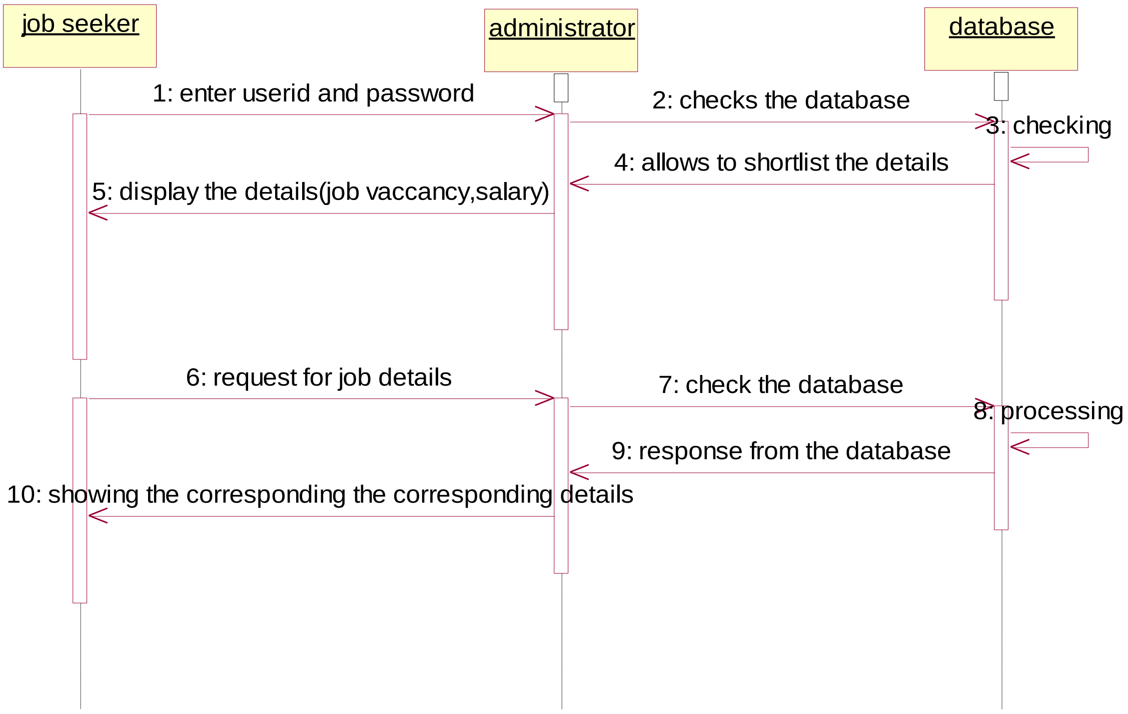
## ACCOUNT CREATION:


job seeker
administrator
1: enter personnel details
2: processing
3: obtain user id and password
4: enter details with qualification
5: checking
6: registration successful
7: pay fee for enrollment
8: get acknoweldement

## AUTHENTICATION:

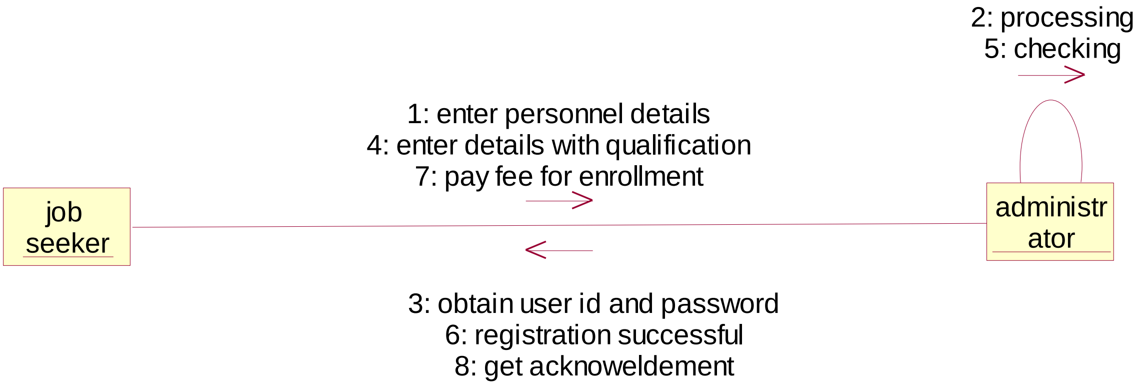


## RECRUITMENT STATUS:

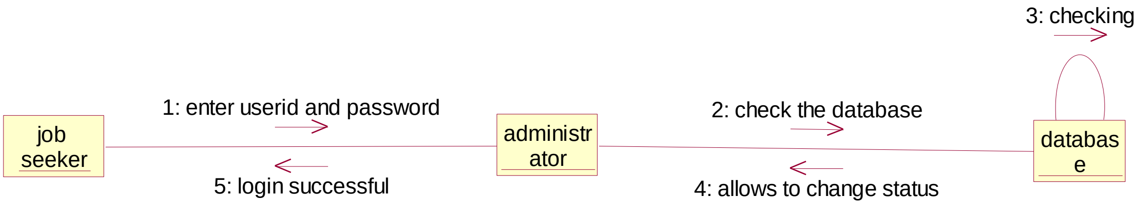

job seeker
administrator
database
1: enter userid and password
2: checks the database
3: checking
4: allows to shortlist the details
5: display the details(job vaccancy,salary)
6: request for job details
7: check the database
8: processing
9: response from the database
10: showing the corresponding the corresponding details

## (ii)COLLABORATION DIAGRAM:

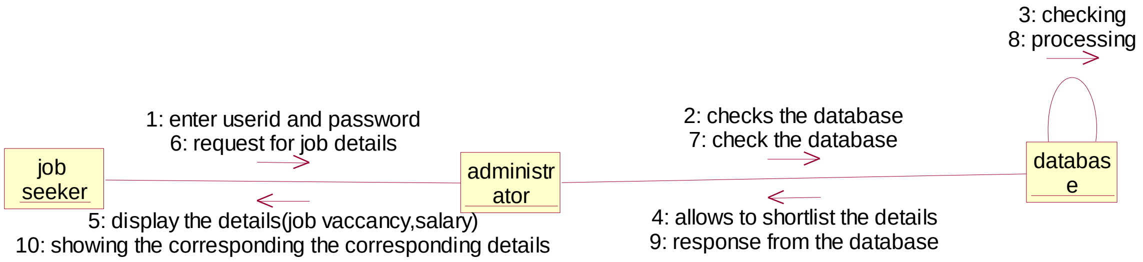
## ACCOUNT CREATION:


2: processing
5: checking
1: enter personnel details
4: enter details with qualification
7: pay fee for enrollment
job seeker
administr ator
3: obtain user id and password
6: registration successful
8: get acknoweldement

## AUTHENTICATION:

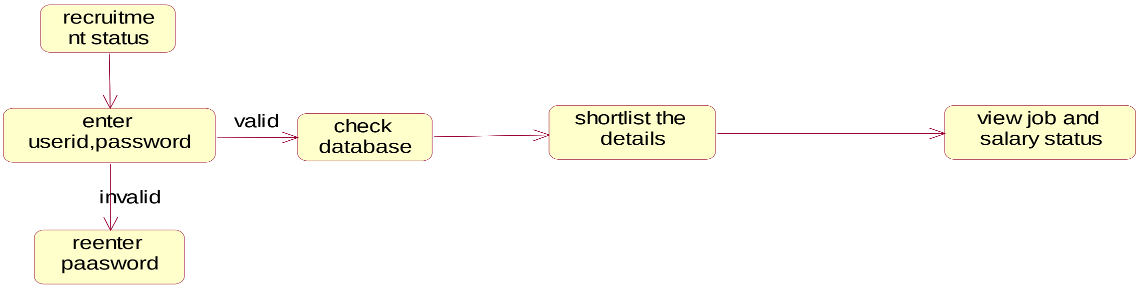


## RECRUITMENT STATUS:

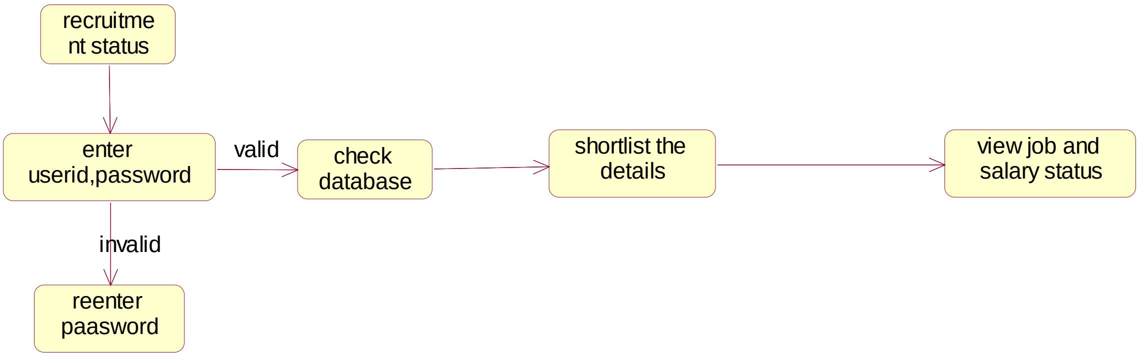

3: checking
8: processing
1: enter userid and password
6: request for job details
2: checks the database
7: check the database
job seeker
administr ator
databas e
5: display the details(job vaccancy,salary)
10: showing the corresponding the corresponding details
4: allows to shortlist the details
9: response from the database

## 5.STATE CHART DIAGRAM:

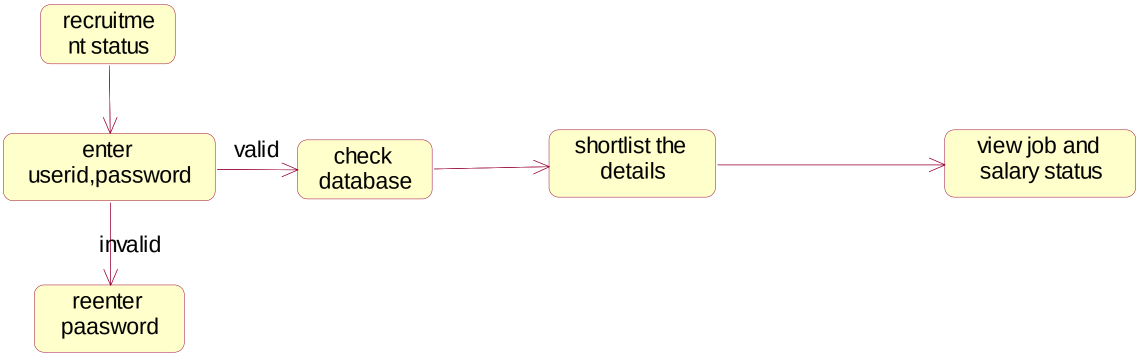
## ACCOUNT CREATION:



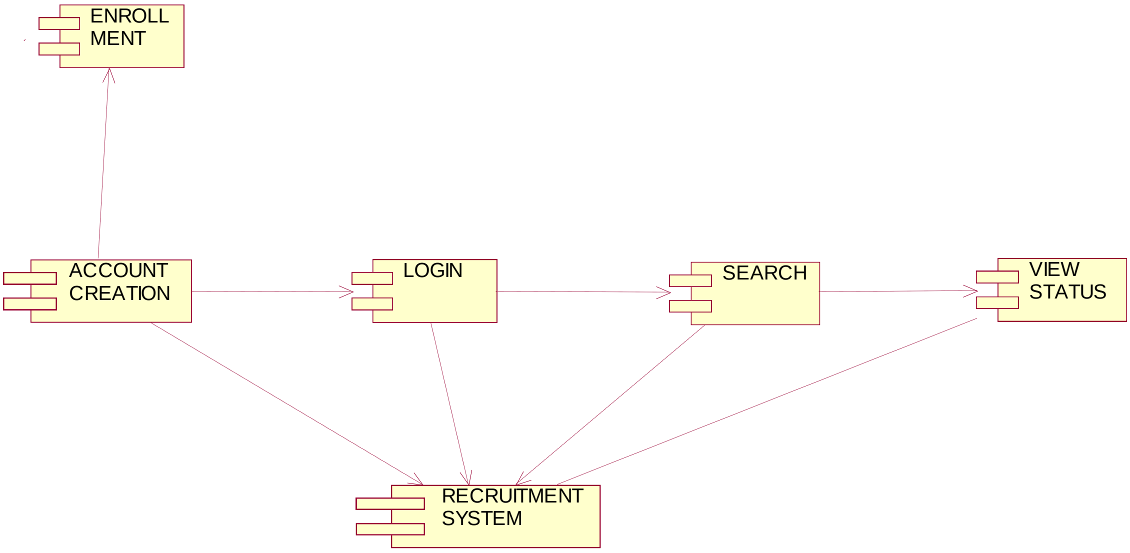
## AUTHENTICATION:



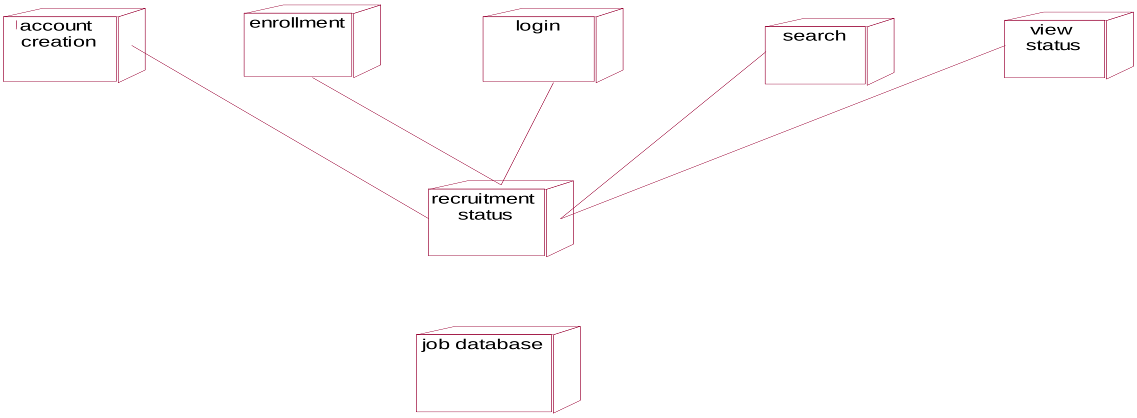
## RECRUITMENT STATUS:



## 6.COMPONENT DIAGRAM:



## 7.DEPLOYMENT DIAGRAM:



## Result:

Thus the UML diagrams has been successfully completed for Recruitment system.

# Ex.No: 7 CONFERENCE MANAGEMENT SYSTEM

**DATE:**

**AIM**

To develop a Conference Management System project using argouml tool.

**PROBLEM STATEMENT**

Conference management system is used to register for a conference by filling all the details. It maintains the active database about information by conference in an efficient way in exercising the system there are problems such as time consumption, lot of paper work and space consumption. So to overcome these problems we move onto conference management .it can be viewed and analyzed in a quick and easy way. It shares time and maintenance cost, it provides online submission of paper in a simple way for participants

**SOFTWARE REQUIREMENT SPECIFICATION**

TABLE OF CONTENTS

## 1. INTRODUCTION

A conference management system can be regarded as domain specific content management system.

### 1.1.PURPOSE

The purpose of this SRS is secure as the responsibility for all requirements that conference management should process

### 1.2. SCOPE

The system provides a communication platform between the user and the reviewer. This will help both the user and reviewer by reducing the time and workload. It is a web based system that supports the entire information about conference in an efficient way.

### 1.3.DOCUMENT CONVENTIONS, ACRONYMS AND ABBREVIATIONS

- Reviewer-Refers to the user who is the Central Authority who has been vested with the privilege to manage the entire system and select the papers.
- Applicant - One who wishes to submit the paper for the conference

- CMS – Conference Management System.

## 1.4.REFERENCES

www.papers.in,www.india.in

## 2.0 OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

The CMS acts as an interface between the applicant and the administrator. It uses external relational database management system as a data. Every case of the system uses only a web browser as an event to connect the system.

### 2.2 PRODUCT FUNCTIONS

This system functions with a database at the backend,for keeping track of its registered students and also its available resources.

### 2.3. TOOLS TO BE USED

Visual Basic and Microsoft Access

## 3. EXTERNAL INTERFACE REQUIREMENTS

### 3.1 HARDWARE INTERFACES

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWARE INTERFACE

The system uses ODBC drive to connect and control the database.

## 4 SYSTEM FEATURES

## 4.1 APPLYING FOR CONFERENCE

### 4.1.1 DESCRIPTION AND PRIORITY

This system allows the student who wishes to participate at the conference has to enroll their names and make their registration easy way .

### 4.1.2 STIMULUS/RESPONSE SEQUENCE

When the applicant submits all the required details, his status will be updated in their database and after he submits the paper, it will be reviewed by the reviewer and he gives the rating. After few days, the applicant can check whether his paper is selected for the conference.

### 4.1.3 FUNCTIONAL REQUIREMENTS

REQ 1: The applicant must have a registered account.

REQ 2: The applicant must submit only the valid information, papers.

## 5 OTHER NONFUNCTIONAL REQUIREMENTS

### 5.1 PERFORMANCE REQUIREMENTS

To increase performance and to free up database resources for other tasks, the default features are written to cache files on their initial load, and consequent accesses to them result in parsing a flat file for data.

### 5.2 SAFETY REQUIREMENTS

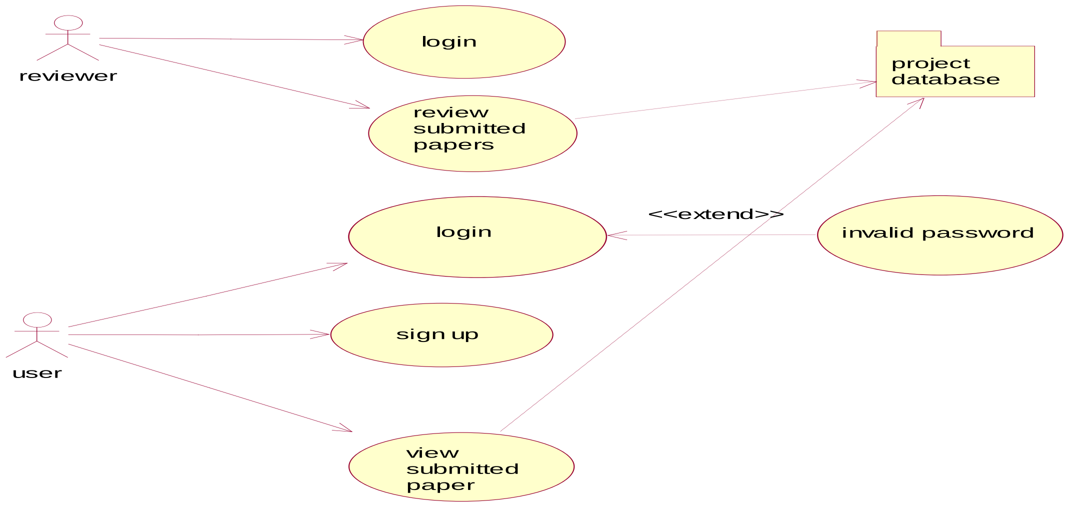
The database must be maintained effectively and the administrator must maintain the interface properly. The user has to be careful while submitting the information.

### 5.3 SECURITY REQUIREMENTS

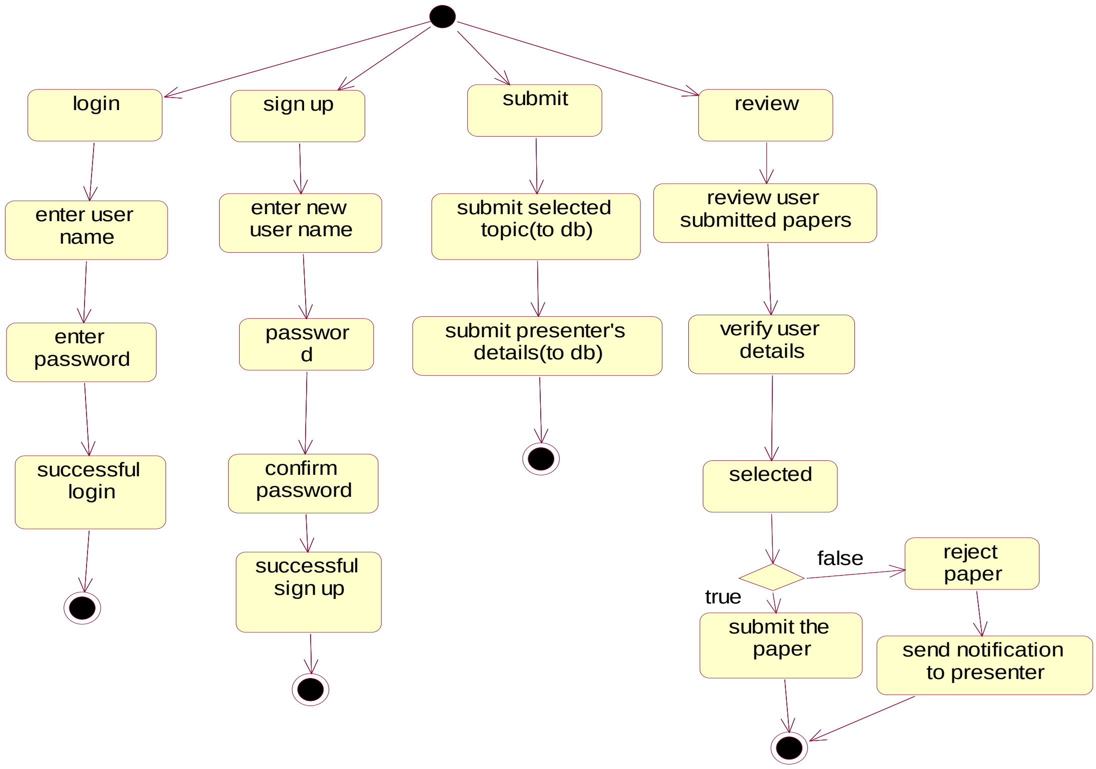
details for registered users are stored in the database. Only the database administrator can have access to the main database. The database must be protected from hacking the details of the applicants.

## CONFERENCE MANAGEMENT SYSTEM

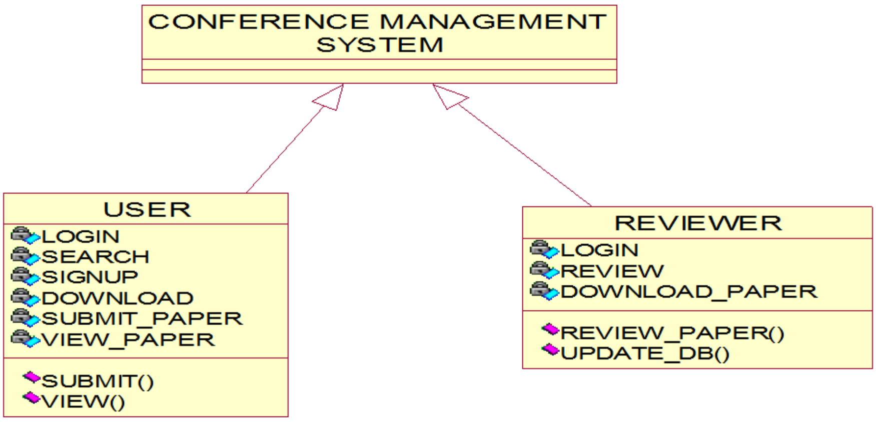
## 1.USE CASE DIAGRAM



## 2.ACTIVITY DIAGRAM

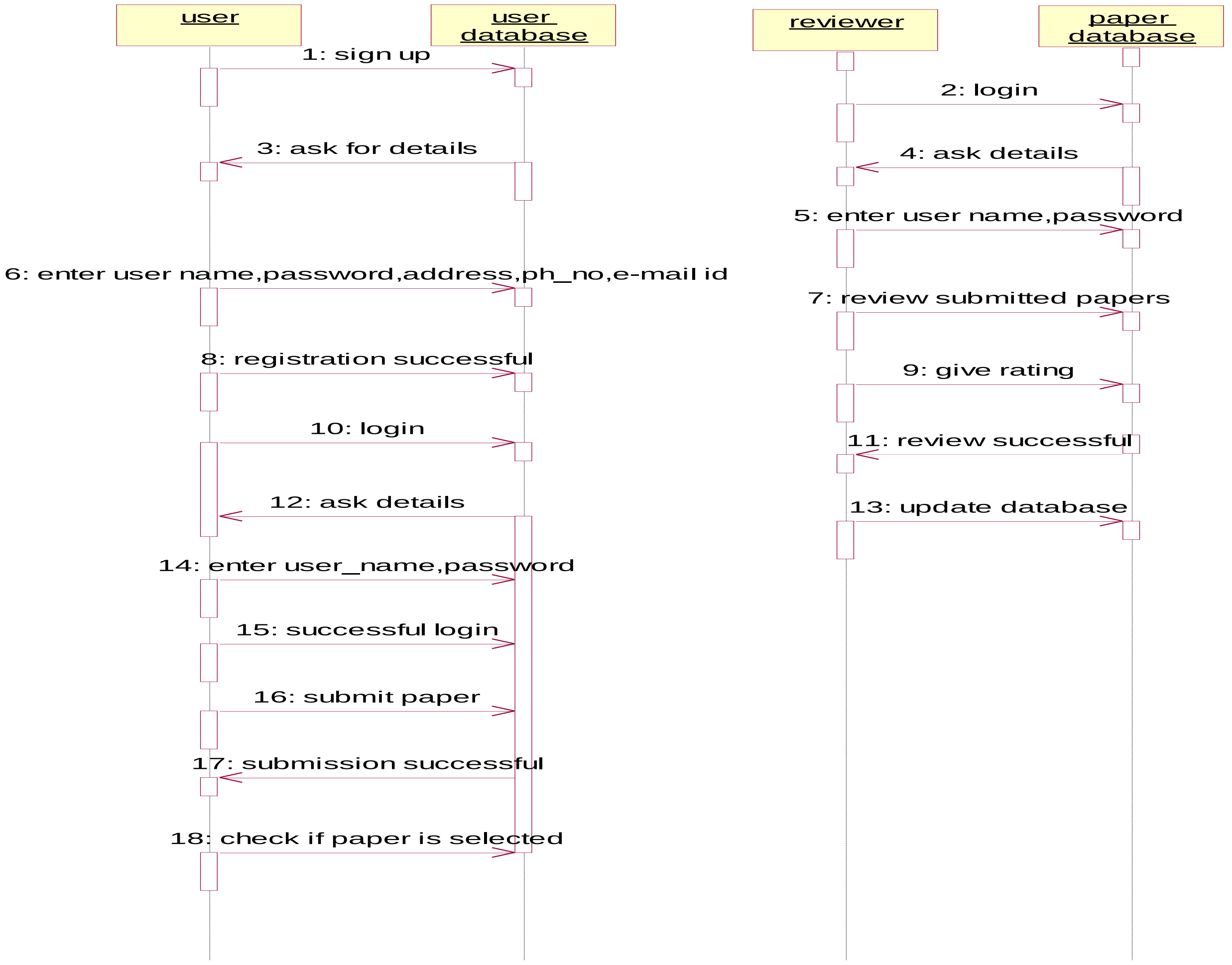


## 3.CLASS DIAGRAM

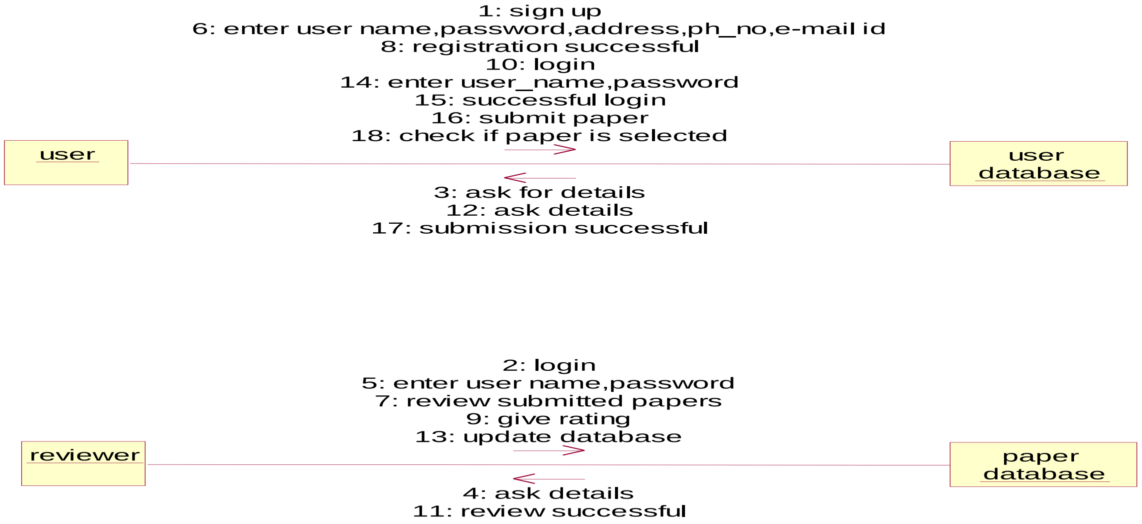


## 4.INTERACTION DIAGRAM:

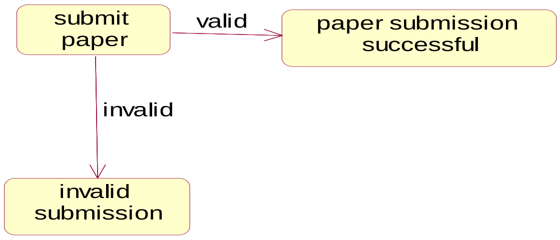
## (i) Sequence Diagram


user
user database
reviewer
paper database
1: sign up
2: login
3: ask for details
4: ask details
5: enter user name,password
6: enter user name,password,address,ph_no,e-mail id
7: review submitted papers
8: registration successful
9: give rating
10: login
11: review successful
12: ask details
13: update database
14: enter user_name,password
15: successful login
16: submit paper
17: submission successful
18: check if paper is selected

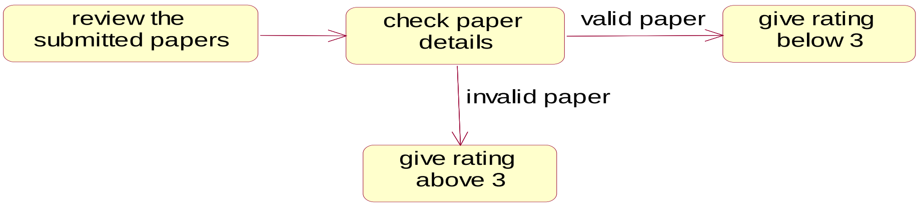
## (ii)COLLABORATION DIAGRAM


1: sign up
6: enter user name,password,address,ph_no,e-mail id
8: registration successful
10: login
14: enter user_name,password
15: successful login
16: submit paper
18: check if paper is selected
user
user database
3: ask for details
12: ask details
17: submission successful
2: login
5: enter user name,password
7: review submitted papers
9: give rating
13: update database
reviewer
paper database
4: ask details
11: review successful

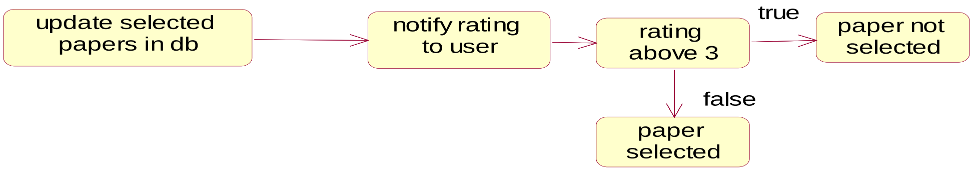
## 5.STATECHART DIAGRAM



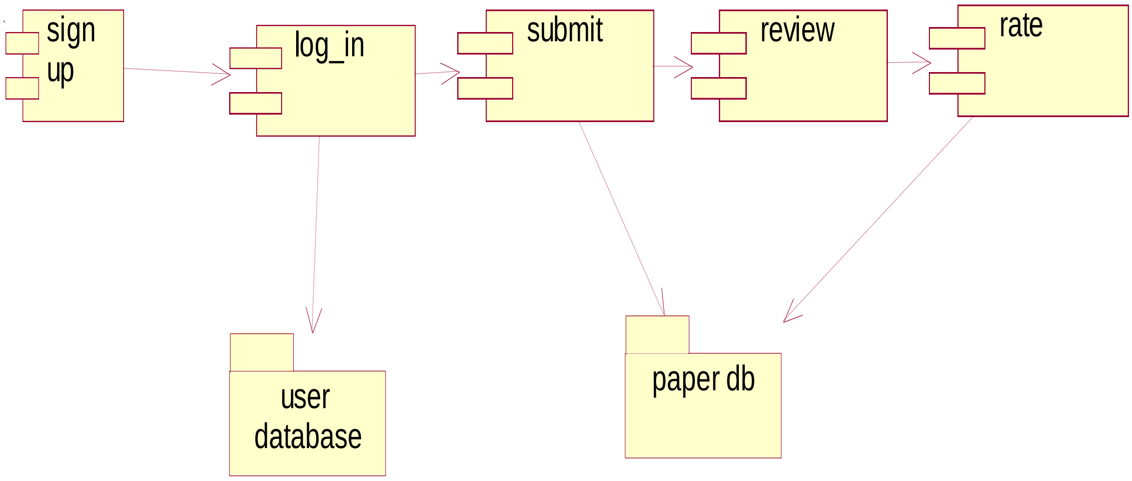
REVIEW PAPER



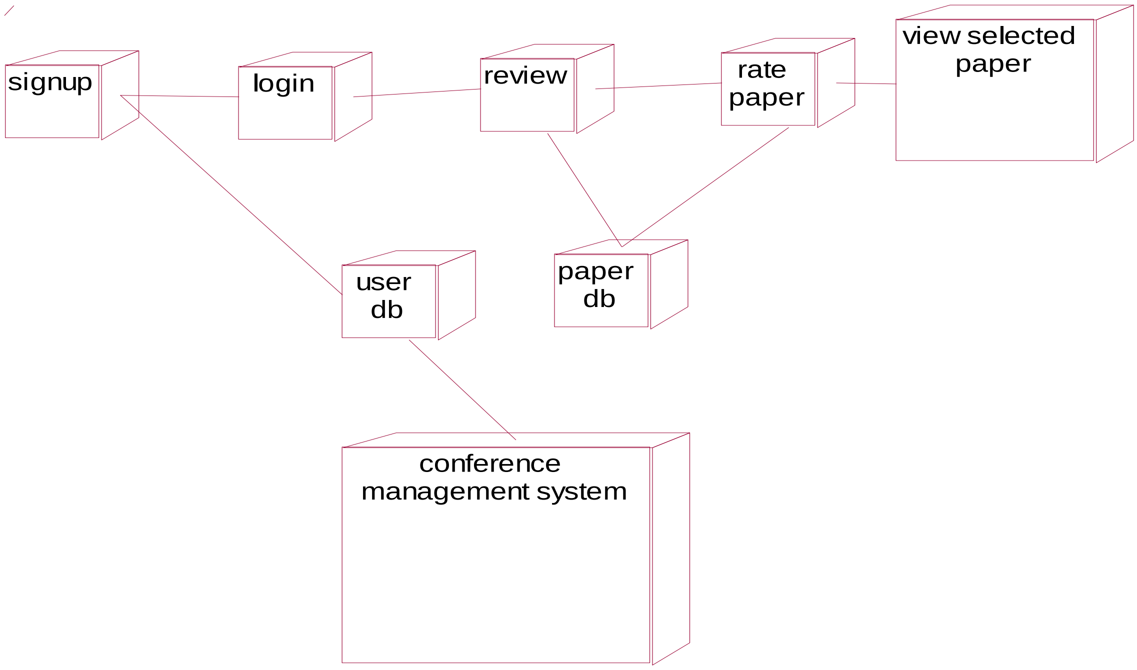
CONFIRM PAPER



## 6.COMPONENT DIAGRAM



## 7.DEPLOYMENT DIAGRAM



**Result:**

Thus the UML diagrams has been successfully completed for Conference management system.

**EX.NO:8** **BPO MANAGEMENT SYSTEM**

**DATE:**

**AIM:**

To development bpo management system project using argouml tool.

**PROBLEM STATEMENT**

Business Process Outsourcing (BPO) Management System is International Management of our business. The customer needs to register them in the system. Each customer has his own username and password. This helps the customer to login in the system, get product information they need, register their complaints about product or management etc. The system works and their process are similar to all call centers. Managing foreign (International) companies and responding their needs and complaints are main process involved in BPO system.

**SOFTWARE REQUIREMENT SPECIFICATION**

TABLE OF CONTENTS

## 1. INTRODUCTION

BPO Management Services (BPOMS) offers strategic new business product for the BPO marketplace. BPO management service is both a BPO direct services and provides an expert BPO project manager.

### 1.1.PURPOSE

BPOMS provide a broad portfolio of business and technology solution to help its client world-wide improve their business performance.

### 1.2.SCOPE

It provides a good improvement in business performance. It improves the existing business technology and improves direct services.

### 1.3 DOCUMENT CONVENTIONS

BPO – Business Process Outsourcing

ITO – IT Outsourcing specifies in mainframe, issues, mid-range and Inter service manage services as well as remote desktop and network services.

### 1.4 REFERENCE

http://www.bporms.com

## 2. OVERALL DESCRIPTION

### 2.1. PRODUCT PERSPECTIVE

BPOMS highly experienced experts have skills to design, implements and support the right solution for each customer need.

### 2.2. PRODUCT FUNCTIONS

Our use of quality delivery method and ISO certified process enables us to deliver service and computer engagement is highly efficient.

### 2.3 TOOLS TO BE USED

Visual basic and Microsoft Access

## 3. EXTERNAL INTERFACES

### 3.1 HARDWARE INTERFACES

The system should have good hardware support. The processor should have high speed and must be of high efficiency.

### 3.2 SOFTWARE INTERFACE

The system uses extensive database. Hence it requires JDBC drivers for connecting to the database.

## 4. SYSTEM FEATURES

### 4.1 CONTRAST DEVELOPMENT

#### 4.1.1. SYSTEM DESCRIPTION AND PRIORITY

This feature allows the BPO to management the constraints.

#### 4.1.2 STIMULUS/RESPONSE SEQUENCE

When the customers call for query the employee must satisfy them with meaningful solution.

#### 4.1.3 FUNCTIONAL REQUIREMENTS

REQ 1: The employee must be hard working.

REQ 2: The Company must have details about the contract.

## 5. OTHER NON-FUNCTIONAL REQUIREMENTS

### 5.1 PERFORMANCE REQUIREMENTS

This system makes use of extensive database hence the initial factor must be loaded in the cache during the initial loading process.

### 5.2 SAFETY REQUIREMENTS

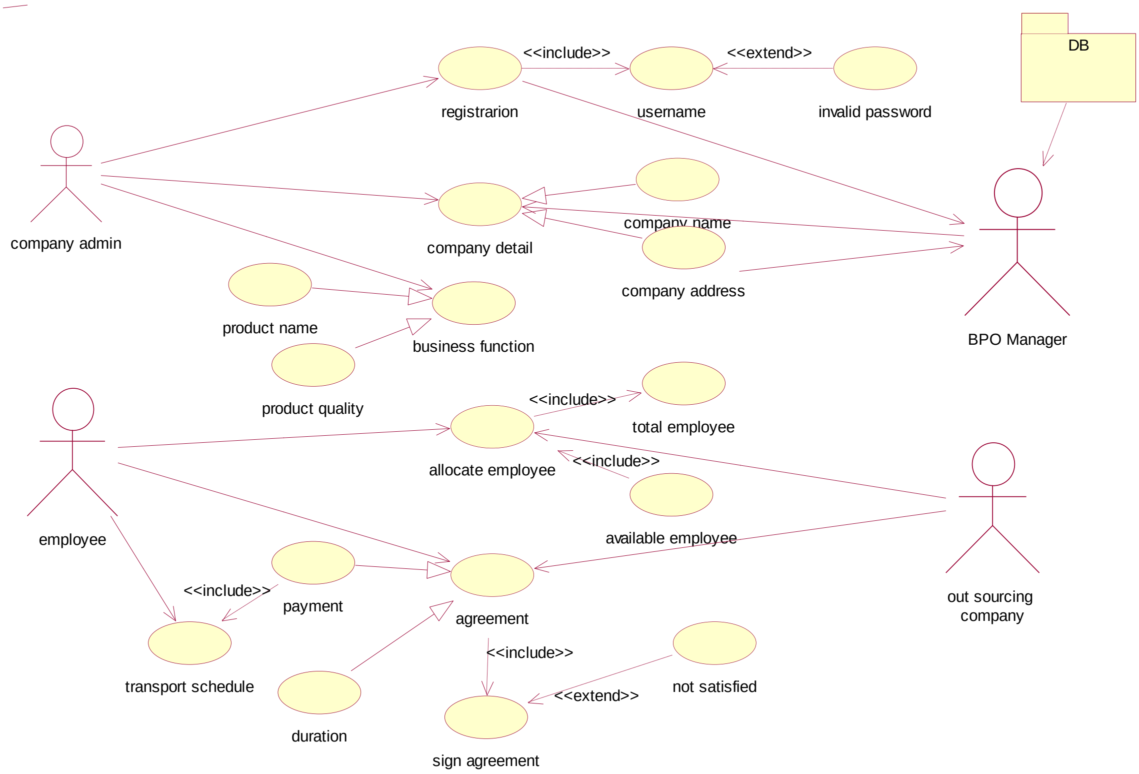
The database must be managed effectively and protected from hacking and virus attack.

### 5.3 SECURITY REQUIREMENTS

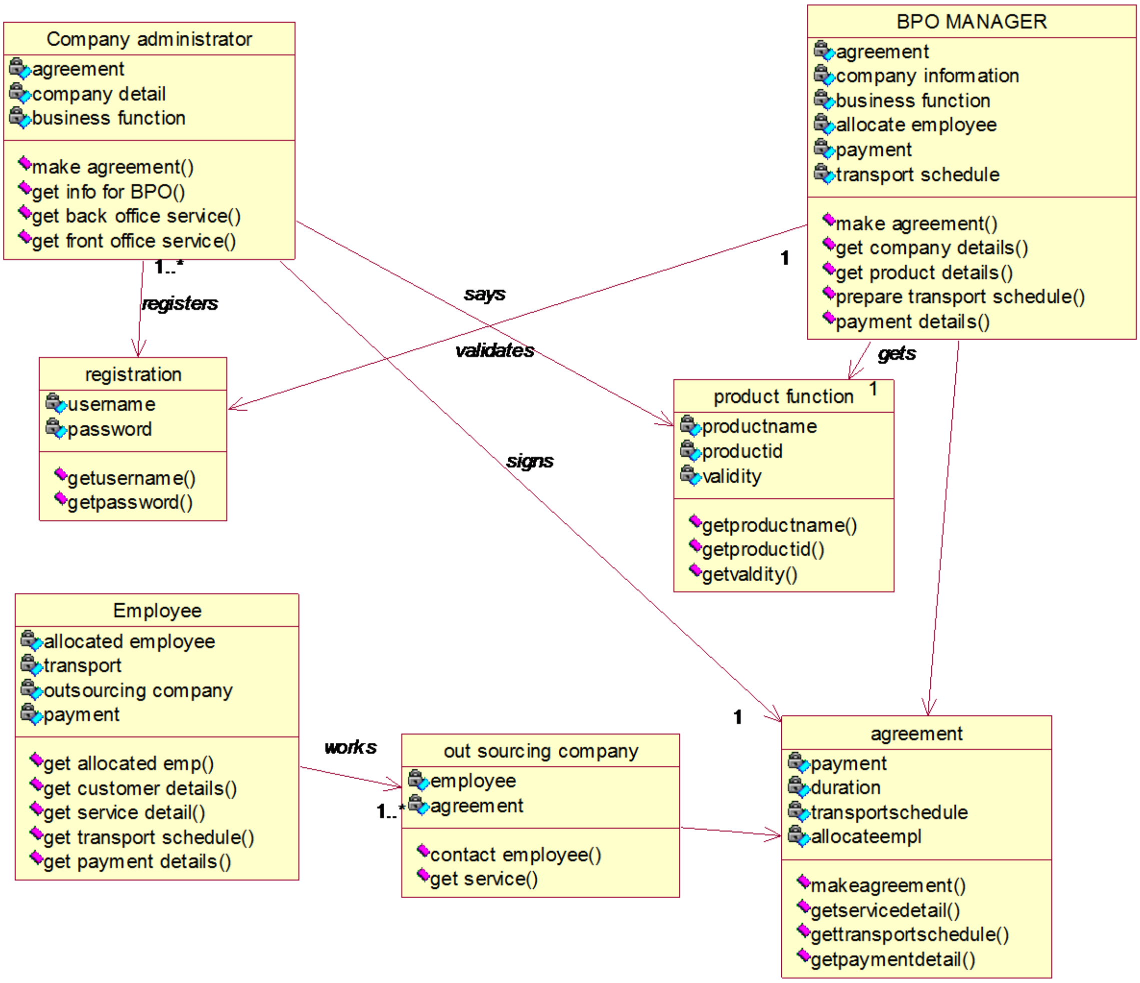
The HR or the administrator can access the profiles of the company and employee details.

## BPO Management System

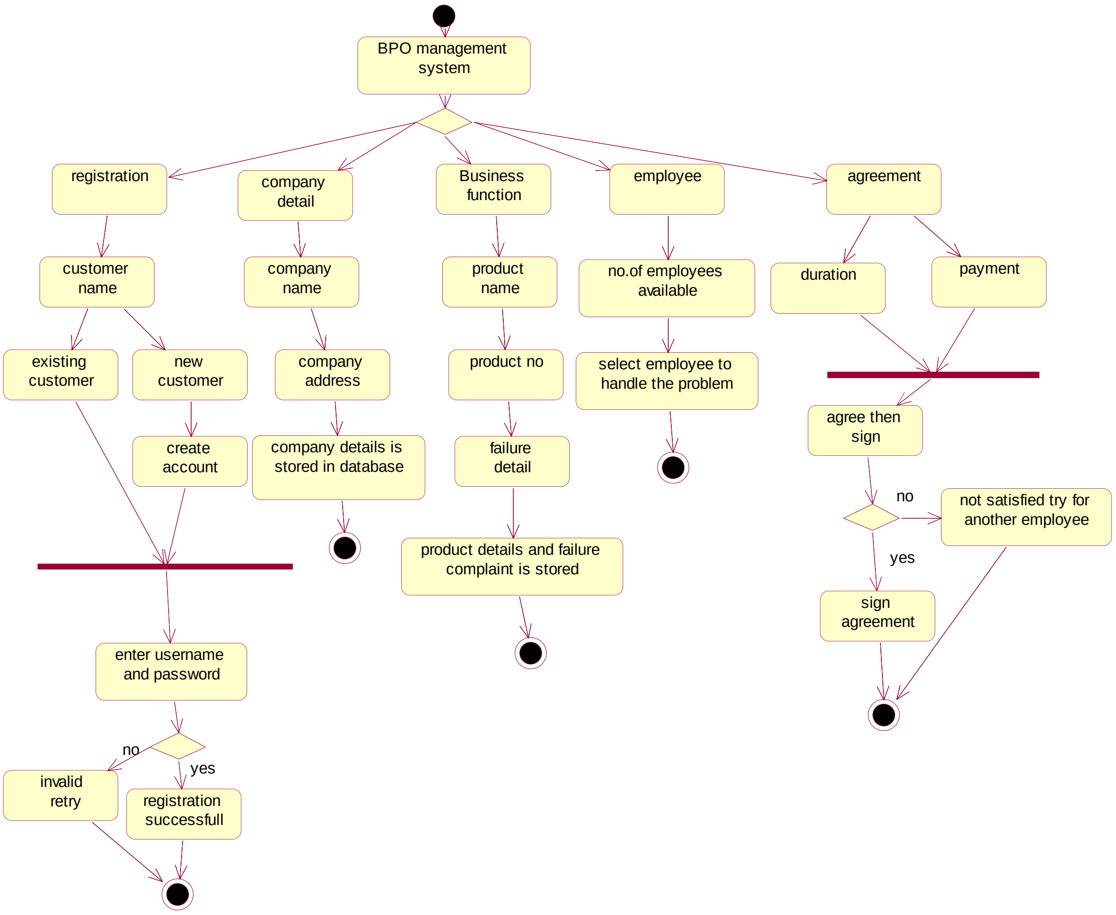
### 1.Use case Diagram:



## 2.Class diagram



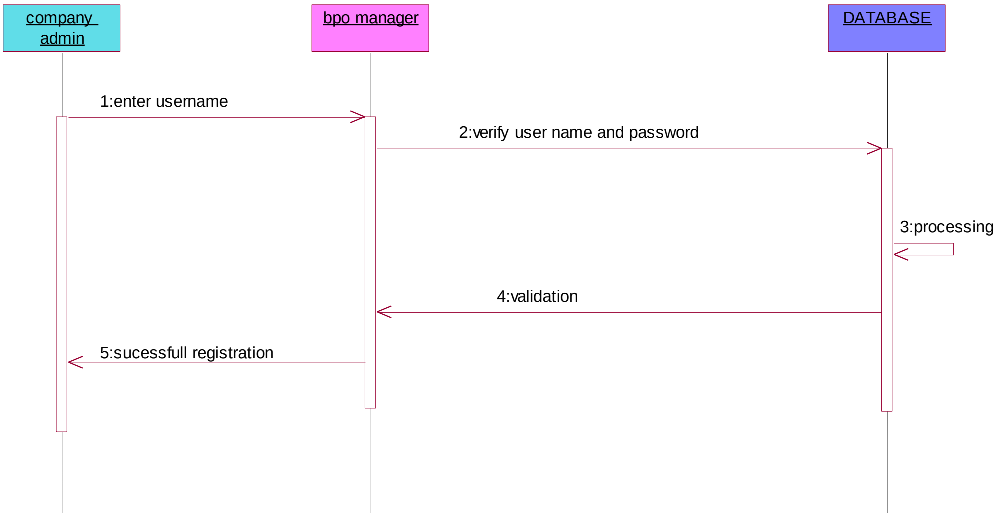
## 3.Activity diagram



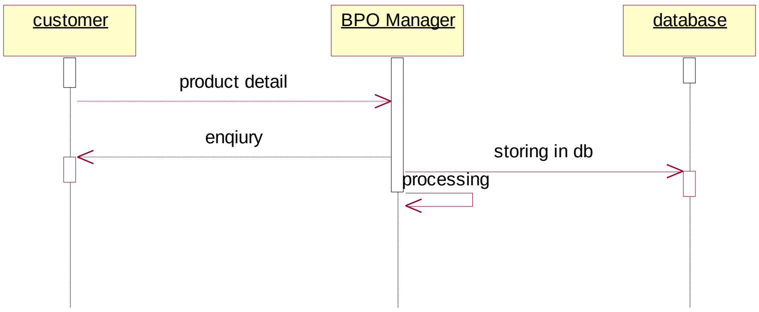
## 4.INTERACTION DIAGRAM:

## (i)Sequence diagram

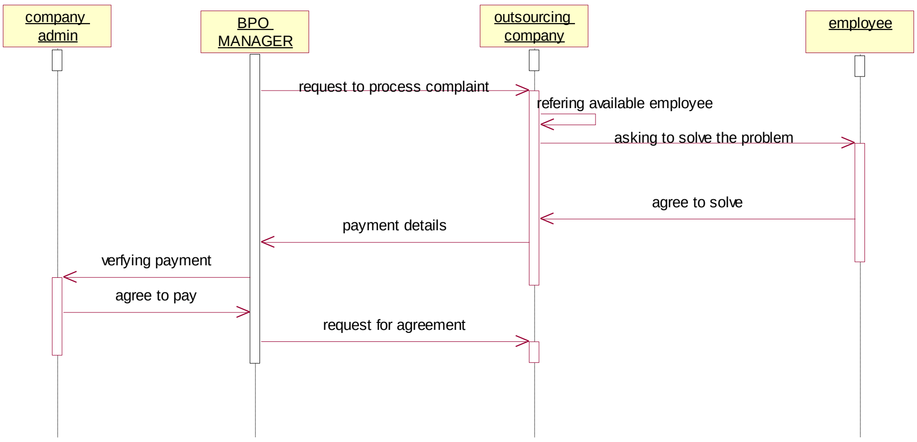
Registration



Product details

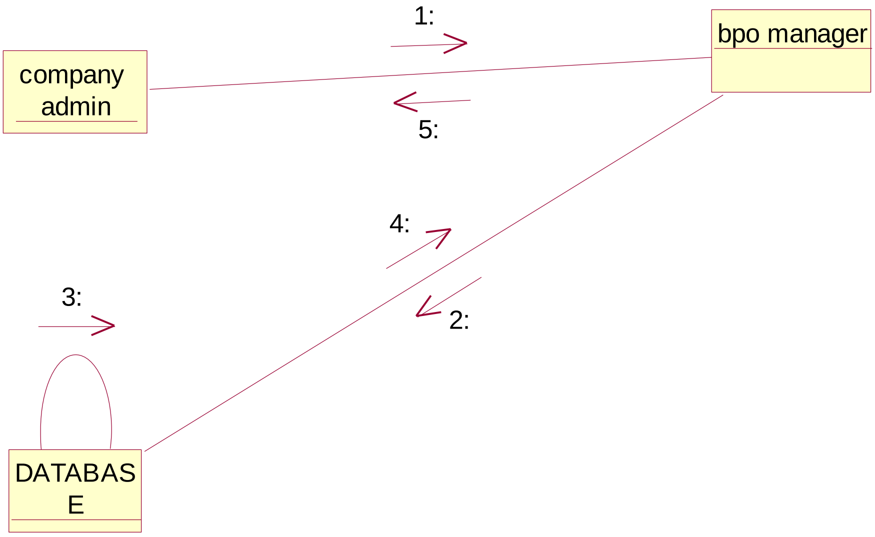


Agreement

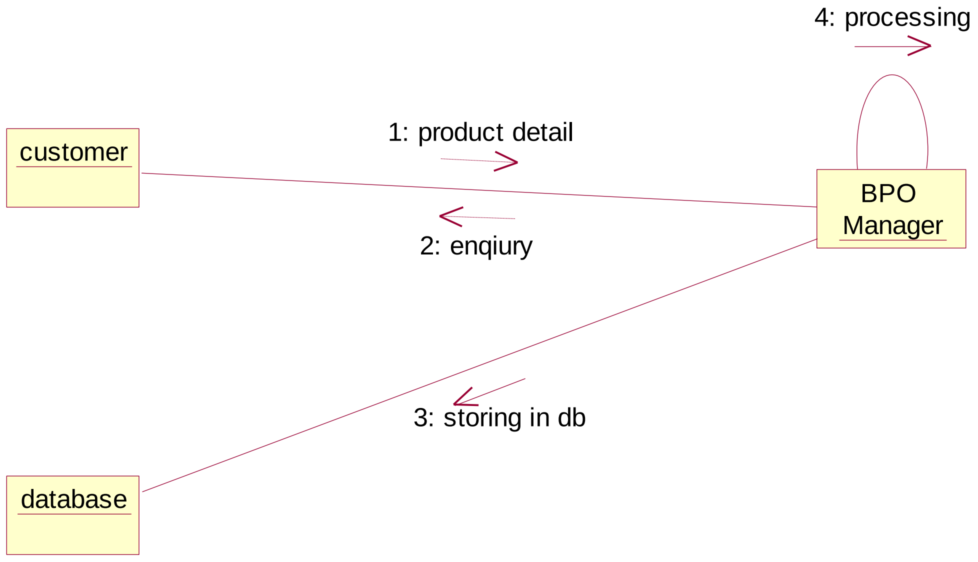


## (ii)Collaboration diagram

Registration

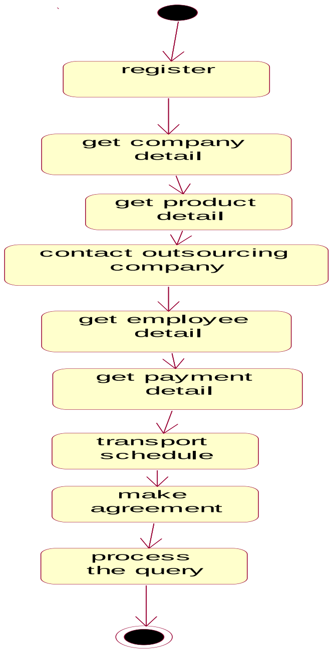


Product detail

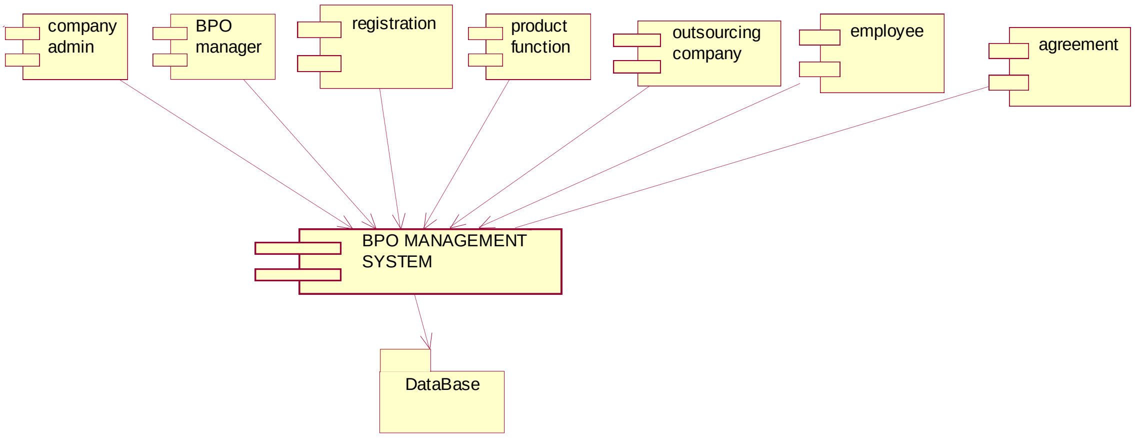


Agreement

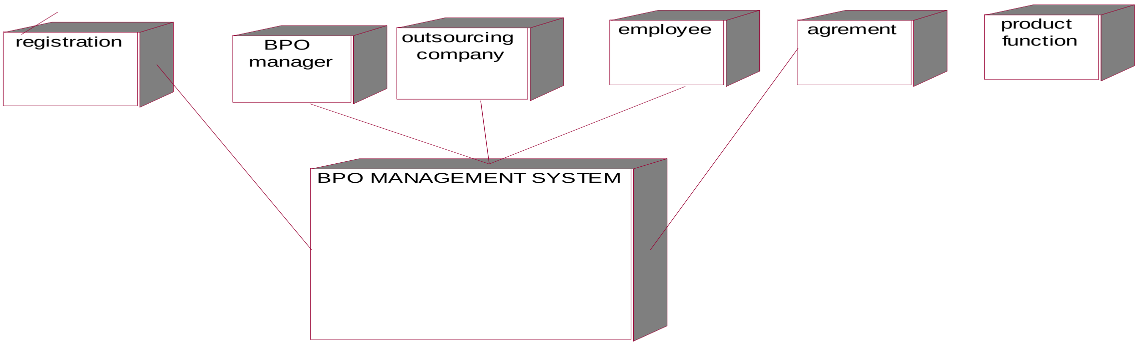
## 5.State Chart Diagram



## 6.Component diagram



## 7.Deployment  diagram



**Result:**

Thus the UML diagrams has been successfully completed for BPO management system.

**EX.NO:9** **LIBRARY MANAGEMENT SYSTEM**

**DATE:**

**AIM:**

To development library management system project using argouml tool.

**PROBLEM STATEMENT:**

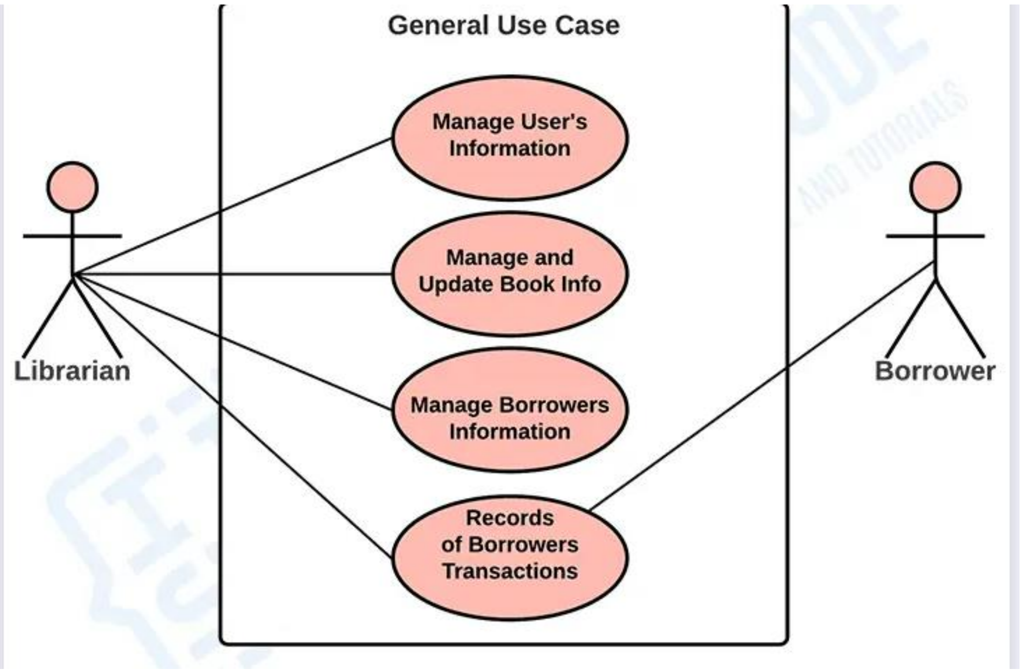
The library management system is a software system that issues books and magazines to registered students only. The student has to login after getting registered to the system. The borrower of the book can perform various functions such as searching for desired book, get the issued book and return the book

**SOFTWARE REQUIREMENT SPECIFICATION**

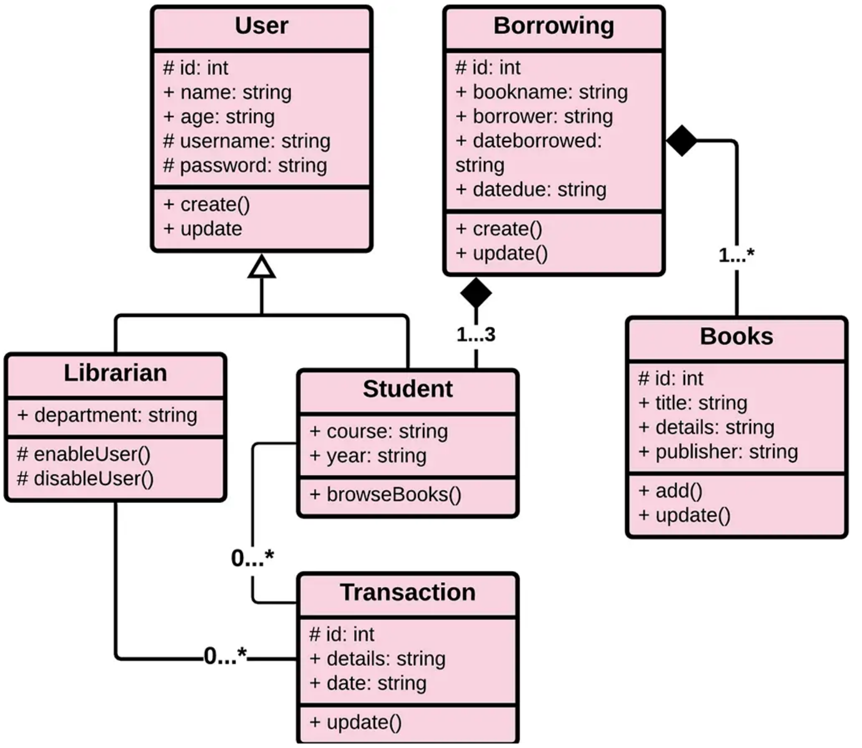
**TABLE OF CONTENTS**

**LIBRARY MANAGEMENT SYSTEM**

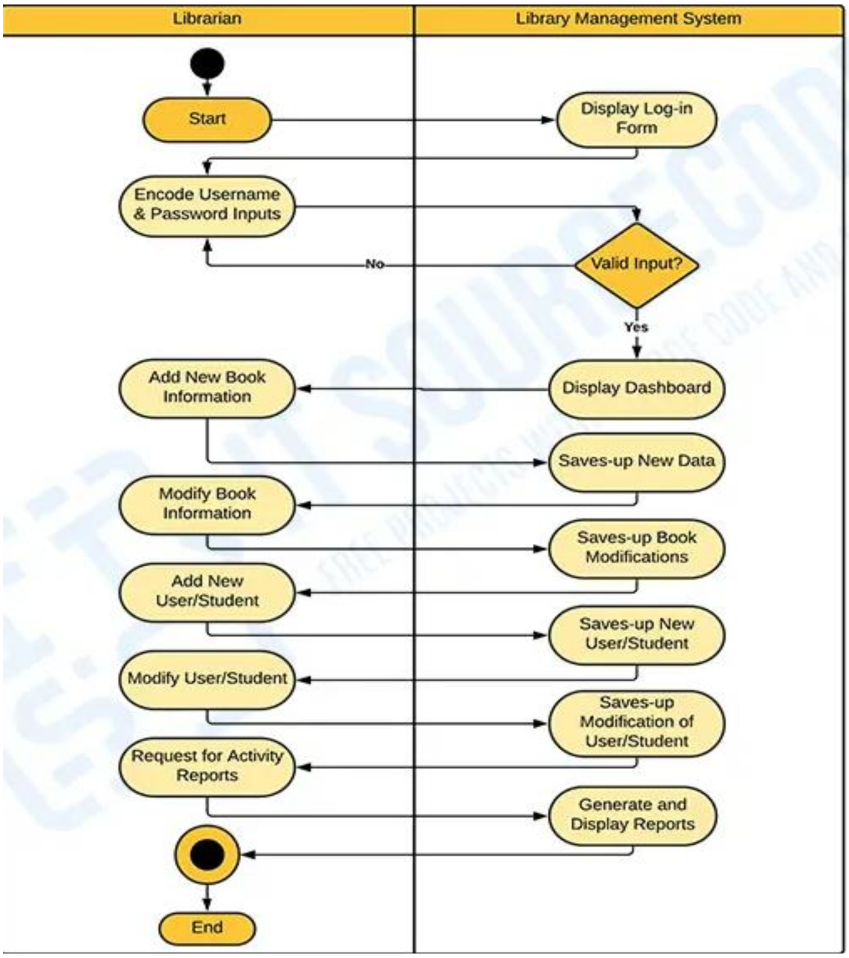
**1.USE CASE DIAGRAM**


General Use Case
Manage User's Information
Manage and Update Book Info
Manage Borrowers Information
Records of Borrowers Transactions
Librarian
Borrower

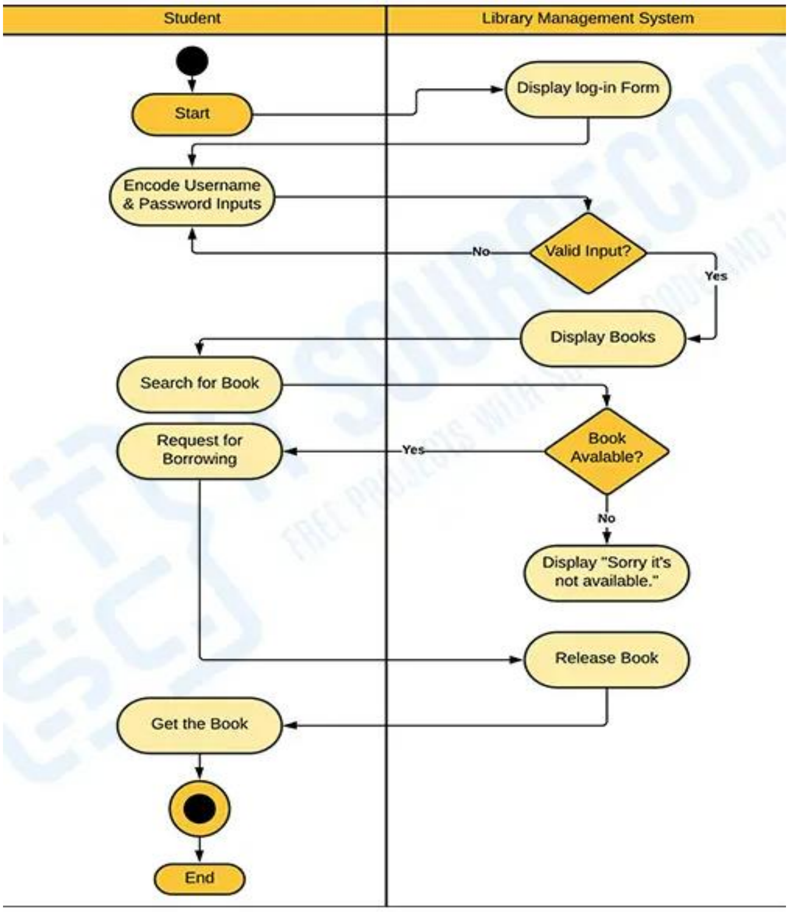
## 2.Class Diagram



## 3.Activity diagram – Librarian Vs Library Management System

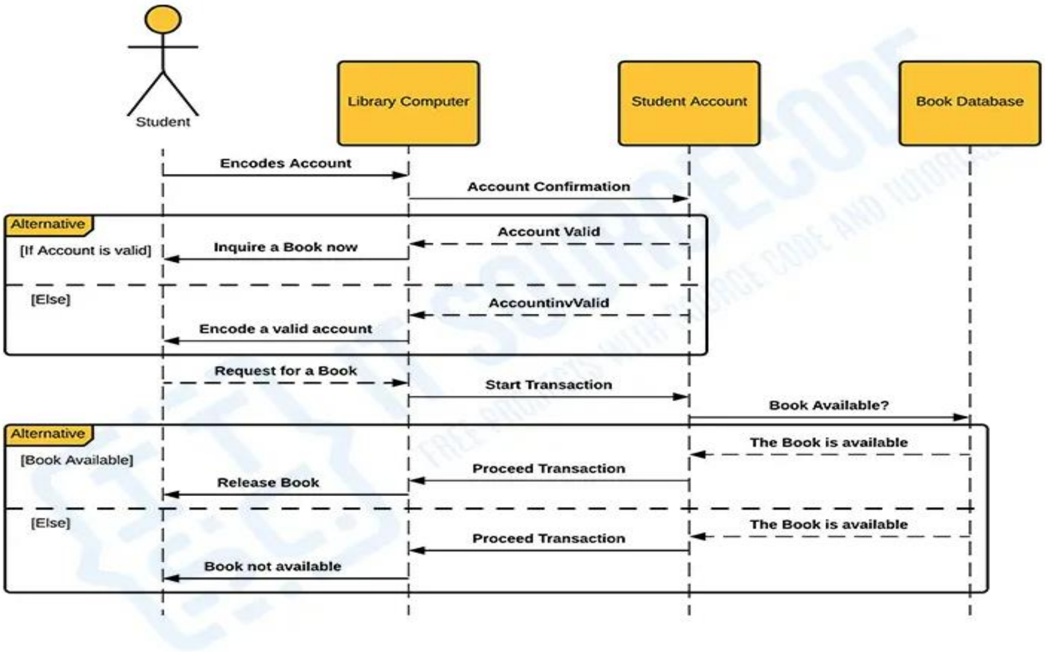

Librarian
Library Management System
Start
Display Log-in Form
Encode Username & Password Inputs
Valid Input?
No
Yes
Add New Book Information
Display Dashboard
Saves-up New Data
Modify Book Information
Saves-up Book Modifications
Add New User/Student
Saves-up New User/Student
Modify User/Student
Saves-up Modification of User/Student
Request for Activity Reports
Generate and Display Reports
End

## Student Vs Library Management System

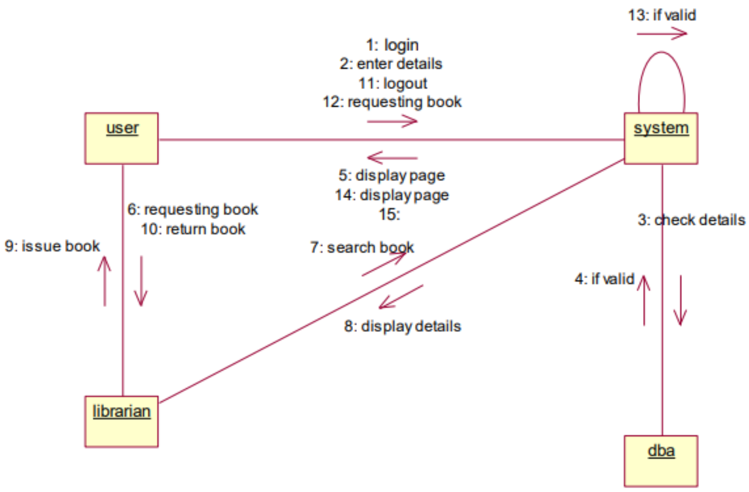


## 4.Interaction Diagram

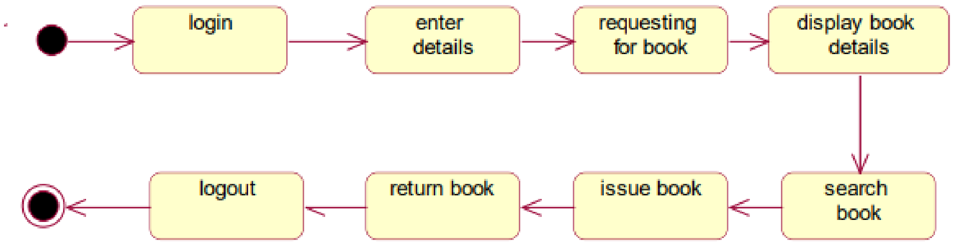
### (i)Sequence Diagram



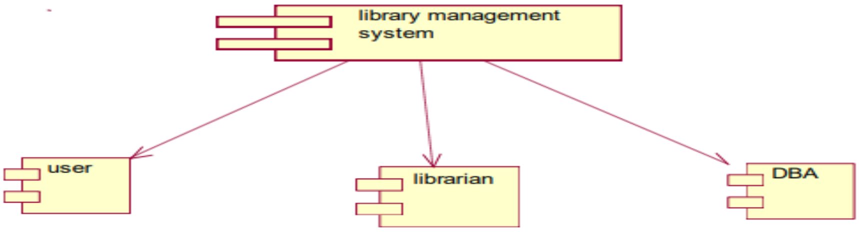
### (ii)Collaboration diagram



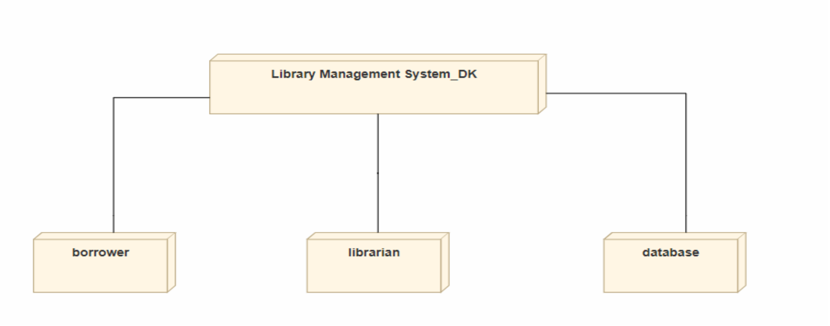
## 5.State Chart Diagram



## 6. Component diagram



## 7.Deployment Diagram



## Result:

Thus the UML diagrams has been successfully completed for Library management system.

# Mini-Projects

**Ex.No:1** **Passport Automation System:**

```
packagepassport;
import
java.util.logging.Level;impo
rt
java.util.logging.Logger;imp
ortjava.sql.*;




/**
 *
 *@authorSRMVEC
 */
publicclassLoginextendsjavax.swing.JFrame {




  /**
   *CreatesnewformLogin
   */
  public Login()
    {initComponent
    s();
  }



  /**
   * Thismethodis calledfromwithintheconstructortoinitialize theform.
   * WARNING:DoNOTmodifythis code. The content ofthismethodisalways
   * regeneratedbytheFormEditor.
   */@SuppressWarnings("unchec
  ked")
  // <editor-fold defaultstate="collapsed" desc="Generated
  Code">privatevoidinitComponents(){

    jPanel1=newjavax.swing.JPanel();
    jLabel1 = new
    javax.swing.JLabel();jButton1 = new
    javax.swing.JButton();jButton2 = new
    javax.swing.JButton();jLabel2 = new
    javax.swing.JLabel();jLabel3 = new
    javax.swing.JLabel();username=newja
    vax.swing.JTextField();
    password = new
    javax.swing.JPasswordField();err=newjava
    x.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


    jLabel1.setFont(new java.awt.Font("Tahoma", 1, 24)); //
    NOI18NjLabel1.setText("Passport
    Automation");jLabel1.setToolTipText("");

    jButton1.setText("Login");
    jButton1.addActionListener(new
      java.awt.event.ActionListener()
      {publicvoidactionPerformed(java.awt.event.ActionEvente
      vt){
        jButton1ActionPerformed(evt);
      }
   });


    jButton2.setText("Register");
    jButton2.addActionListener(new
      java.awt.event.ActionListener()
      {publicvoidactionPerformed(java.awt.event.ActionEvente
      vt){
        jButton2ActionPerformed(evt);
      }
   });


    jLabel2.setText("Password");
  jLabel3.setText("Username");


    username.addActionListener(new
      java.awt.event.ActionListener()
      {publicvoidactionPerformed(java.awt.event.ActionEventevt
      ){
        usernameActionPerformed(ev
                              t);
      }
   });


  password.setToolTipText("");


    javax.swing.GroupLayoutjPanel1Layout=newjavax.swing.GroupLayout(jPanel1);jPanel1.
    setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(jPanel1Layout.createParallelGroup(javax.swing.Group
      Layout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILI
NG)
           .addGroup(jPanel1Layout.createSequentialGroup()
             .addGap(158,158,158)
             .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE,
88,javax.swing.GroupLayout.PREFERRED_SIZE)
             .addGap(88,88,88)
             .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE,
98,javax.swing.GroupLayout.PREFERRED_SIZE)
             .addGap(0,71,Short.MAX_VALUE))
           .addGroup(jPanel1Layout.createSequentialGroup()
             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,Short.MAX_VAL
             UE)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
               .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
335,javax.swing.GroupLayout.PREFERRED_SIZE)
               .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
                   .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE,
55,javax.swing.GroupLayout.PREFERRED_SIZE)
                   .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
55,javax.swing.GroupLayout.PREFERRED_SIZE))
                 .addGap(41,41,41)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G,false)
                   .addComponent(username,javax.swing.GroupLayout.DEFAULT_SIZE,111
                   ,
Short.MAX_VALUE)
                   .addComponent(password))))))
         .addGap(101,101,101))
       .addGroup(jPanel1Layout.createSequentialGroup()
         .addGap(192,192,192)
         .addComponent(err, javax.swing.GroupLayout.PREFERRED_SIZE,
175,javax.swing.GroupLayout.PREFERRED_SIZE)
         .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,Short.MAX_VALUE)
         )
     );
     jPanel1Layout.setVerticalGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLa
       yout.Alignment.LEADING)
       .addGroup(jPanel1Layout.createSequentialGroup()
         .addGap(54,54,54)
         .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
33,javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addGap(52,52,52)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
               .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE,
22,javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(username,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
41,Short.MAX_VALUE)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
               .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE,
23,javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(password,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(14,14,14)
            .addComponent(err)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
NE)
               .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
34,javax.swing.GroupLayout.PREFERRED_SIZE)
               .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
37,javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(94,94,94))
      );


      javax.swing.GroupLayoutlayout=newjavax.swing.GroupLayout(getContentPane());getCo
      ntentPane().setLayout(layout);
      layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
         nt.LEADING)
         .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE)
            .addContainerGap())
      );
      layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
         LEADING)
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,layout.createSequentialGro
         up()
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,Short.MAX_VALUE)
            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap())

      );
      pack();
   }//</editor-fold>



   private void jButton1ActionPerformed(java.awt.event.ActionEvent
     evt) {Connectioncon= null;
     Statement statement =
     null;try{
       Class.forName("com.mysql.jdbc.Driver");
       con
=DriverManager.getConnection("jdbc:mysql://localhost:3306/summa?characterEncoding=latin
1&use Configs=maxPerformance","root", "admin");
       Statementstmt=con.createStatement();
         ResultSet rs=stmt.executeQuery("select * from
      passport");Stringun= username.getText();
      String pw =
      password.getText();while(rs.
      next())
{
   if (un.equals(rs.getString(1)) &&
     pw.equals(rs.getString(2))){passport.NewJFrame1 j =
     new passport.NewJFrame1();j.username= un;
     j.setinfo();this.sh
     ow(false);j.show(
     true);

   }}
   err.setText("invalid username or
password!");con.close();

     }
     catch (Exception e)
       {e.printStackTrac
       e();
     }
   }


   private void jButton2ActionPerformed(java.awt.event.ActionEvent
     evt) {passport.NewJFrame i=newpassport.NewJFrame();
     this.show(false);i.show(t
     rue);

   }


   privatevoidusernameActionPerformed(java.awt.event.ActionEventevt){
     //TODOaddyourhandlingcode here:
   }


   /**
    * @param argsthecommandline arguments
    */
   publicstaticvoidmain(Stringargs[]){
     /*Setthe Nimbus lookandfeel*/


//<editor-folddefaultstate="collapsed"desc="Lookandfeelsettingcode(optional)">
     /*IfNimbus(introducedinJavaSE6)isnotavailable,staywiththedefaultlook andfeel.
      *Fordetailsseehttp://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
      */t
     ry{
       for (javax.swing.UIManager.LookAndFeelInfo
info
:javax.swing.UIManager.getInstalledLookAndFeels(
)){
         if ("Nimbus".equals(info.getName()))
           {javax.swing.UIManager.setLookAndFeel(info.getClassNa
           me()); break;
         }

       }
     } catch (ClassNotFoundException ex)
       {java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.
       SEVERE,
null,ex);
     } catch (InstantiationException ex)
       {java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.
       SEVERE,
null,ex);
     } catch (IllegalAccessException ex)
       {java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.
       SEVERE,
null,ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex)
          {java.util.logging.Logger.getLogger(Login.class.getName()).log(java.util.logging.Level.
          SEVERE,
null,ex);
        }
        //</editor-fold>


        /*Createanddisplaytheform*/


        java.awt.EventQueue.invokeLater(new
          Runnable() {publicvoidrun() {
            newLogin().setVisible(true);
          }
        });


      }
```

```
   // Variables declaration - do not
   modifyprivatejavax.swing.JLabelerr;
   private javax.swing.JButton
   jButton1;private
   javax.swing.JButton
   jButton2;private
   javax.swing.JLabel
   jLabel1;private
   javax.swing.JLabel
   jLabel2;private
   javax.swing.JLabel
   jLabel3;privatejavax.swing.JPanel
   jPanel1;
   private javax.swing.JPasswordField
   password;privatejavax.swing.JTextFieldus
   ername;
   // End of variables
   declarationprivatejavax.swi
   ng.JPanels;
}
packagepassport;


/**
 *
 *@authorSRMVEC
 */
publicclassdummyextendsjavax.swing.JFrame{


   /**
    *Createsnewform dummy
    */
   public dummy()
      {initComponent
      s();
   }


   /**
    * Thismethodis calledfromwithintheconstructortoinitialize theform.
    * WARNING:DoNOTmodifythiscode. The content ofthismethodisalways
    * regeneratedbytheFormEditor.
    */@SuppressWarnings("unchec
   ked")
   // <editor-fold defaultstate="collapsed" desc="Generated
   Code">privatevoidinitComponents(){

      jLabel1 = new
      javax.swing.JLabel();jLabel2 =
      new javax.swing.JLabel();jButton1
      = new
      javax.swing.JButton();jButton2 =
      new
```

```
javax.swing.JButton();name=newj
avax.swing.JTextField();
password=newjavax.swing.JPasswordField();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


jLabel1.setText("name");


jLabel2.setText("pass");


jButton1.setText("jButton1");
jButton1.addActionListener(new
   java.awt.event.ActionListener()
   {publicvoidactionPerformed(java.awt.event.ActionEvente
   vt){
      jButton1ActionPerformed(evt);
   }
});


jButton2.setText("jButton2");


password.addActionListener(new
   java.awt.event.ActionListener()
   {publicvoidactionPerformed(java.awt.event.ActionEventevt
   ){
      passwordActionPerformed(evt);
   }
});


javax.swing.GroupLayoutlayout=newjavax.swing.GroupLayout(getContentPane());getCo
ntentPane().setLayout(layout);
layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
   nt.LEADING)
   .addGroup(layout.createSequentialGroup()
      .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
      G)
         .addGroup(layout.createSequentialGroup()
            .addGap(165,165,165)
            .addComponent(jButton1)
            .addGap(110,110,110)
            .addComponent(jButton2))
         .addGroup(layout.createSequentialGroup()
            .addGap(120,120,120)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRA
            ILING)
               .addComponent(jLabel1)
```

```
              .addComponent(jLabel2))
            .addGap(157,157,157)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
            DING,
false)
              .addComponent(name)
              .addComponent(password,
javax.swing.GroupLayout.DEFAULT_SIZE, 135,Short.MAX_VALUE))))
        .addContainerGap(123,Short.MAX_VALUE))
    );
    layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
      LEADING)
      .addGroup(layout.createSequentialGroup()
        .addGap(68,68,68)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
        NE).addComponent(jLabel1)
          .addComponent(name,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(23,23,23)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
        NE)
          .addComponent(jLabel2)
          .addComponent(password,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
131,Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
        NE)
          .addComponent(jButton1)
          .addComponent(jButton2))
        .addGap(107,107,107))
    );


    pack();
  }//</editor-fold>


  privatevoidpasswordActionPerformed(java.awt.event.ActionEventevt){
    //TODOaddyourhandlingcode here:
  }


  privatevoidjButton1ActionPerformed(java.awt.event.ActionEventevt){
    //TODOaddyourhandlingcode here:
  }


  /**
   * @param argsthecommandline arguments
   */
  publicstaticvoidmain(Stringargs[]){
    /*Setthe Nimbus lookandfeel*/
    //<editor-folddefaultstate="collapsed"desc="Lookandfeelsettingcode(optional)">
    /*IfNimbus(introducedinJavaSE6)isnotavailable,staywiththedefaultlook andfeel.
     *Fordetailsseehttp://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */t
    ry{
      for (javax.swing.UIManager.LookAndFeelInfo
info
:javax.swing.UIManager.getInstalledLookAndFeels(
)){
        if ("Nimbus".equals(info.getName()))
          {javax.swing.UIManager.setLookAndFeel(info.getClassNa
          me()); break;
        }
      }
    } catch (ClassNotFoundException ex)
      {java.util.logging.Logger.getLogger(dummy.class.getName()).log(java.util.logging.Leve
      l.SEVERE,
null,ex);
    } catch (InstantiationException ex)
      {java.util.logging.Logger.getLogger(dummy.class.getName()).log(java.util.logging.Leve
      l.SEVERE,
null,ex);
    } catch (IllegalAccessException ex)
      {java.util.logging.Logger.getLogger(dummy.class.getName()).log(java.util.logging.Leve
      l.SEVERE,
null,ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex)
      {java.util.logging.Logger.getLogger(dummy.class.getName()).log(java.util.logging.Leve
      l.SEVERE,
null,ex);
    }
    //</editor-fold>


    /* Create and display the form
    */java.awt.EventQueue.invokeLater(newRunna
    ble(){
      publicvoidrun() {
        newdummy().setVisible(true);
      }
    });
  }

  // Variables declaration - do not
  modifyprivate javax.swing.JButton
  jButton1;private
  javax.swing.JButton
  jButton2;private javax.swing.JLabel
  jLabel1;private javax.swing.JLabel
  jLabel2;privatejavax.swing.JTextFie
  ldname;
  privatejavax.swing.JPasswordFieldpassword;
```

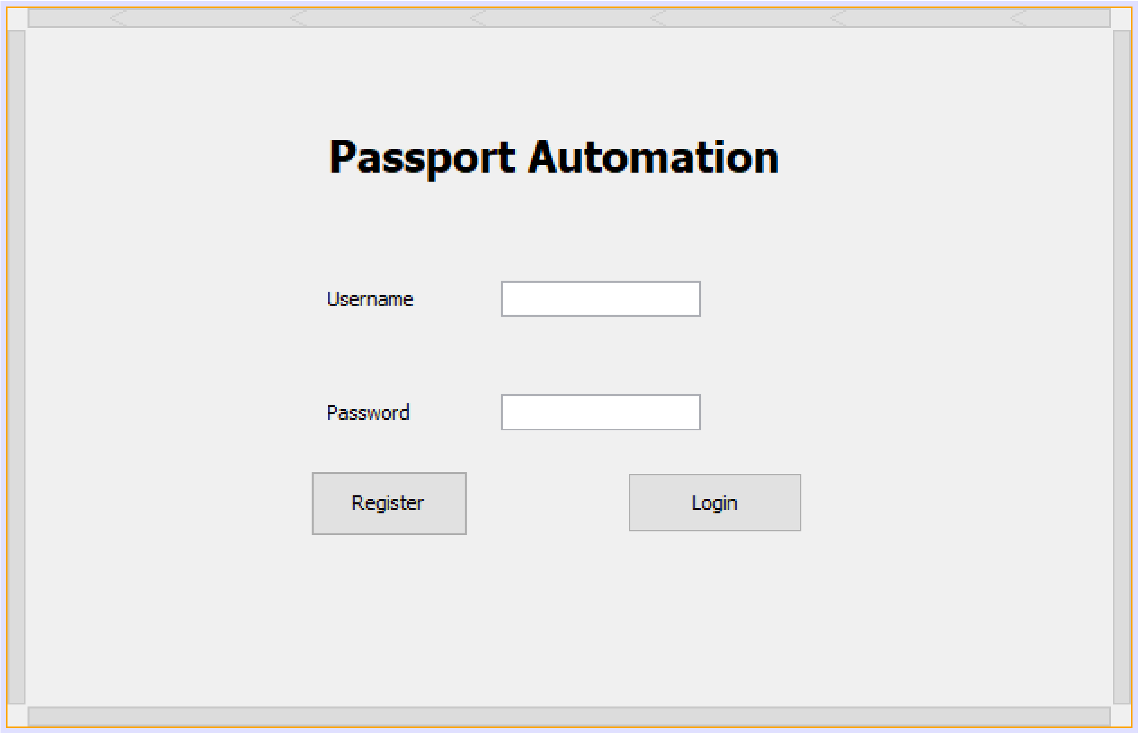
```
   //Endofvariablesdeclaration
}
packagepassport;


import
java.sql.Connection;import
java.sql.DriverManager;imp
ort
java.sql.ResultSet;importjav
a.sql.Statement;

/**
 *
 *@authorSRMVEC
 */
publicclassNewJFrameextendsjavax.swing.JFrame{


   /**
    *Createsnewform NewJFrame
    */
   public
      NewJFrame()
      {initComponents
      ();
   }


   /**
    * Thismethodis calledfromwithintheconstructortoinitialize theform.
    * WARNING:DoNOTmodifythiscode. The content ofthismethodisalways
    * regeneratedbytheFormEditor.
    */@SuppressWarnings("unchec
   ked")
   // <editor-fold defaultstate="collapsed" desc="Generated
   Code">privatevoidinitComponents(){

      lname = new javax.swing.JLabel();
      jLabel2 = new javax.swing.JLabel();
      jLabel3 = new javax.swing.JLabel();
      jLabel1 = new javax.swing.JLabel();
      jLabel4 = new javax.swing.JLabel();
      jLabel5 = new javax.swing.JLabel();
      jLabel6 = new javax.swing.JLabel();
      name = new javax.swing.JTextField();
      gender = new javax.swing.JTextField();
      age = new javax.swing.JTextField();
      address=newjavax.swing.JTextField();
      username = new javax.swing.JTextField();
      password = new javax.swing.JTextField();
      jButton1=newjavax.swing.JButton();

      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
      lname.setText("name");


      jLabel3.setText("gender");


      jLabel1.setText("age");

      jLabel4.setText("address");


      jLabel5.setText("Username");


      jLabel6.setText("password");


      jButton1.setText("submit");
      jButton1.addActionListener(new
        java.awt.event.ActionListener()
        {publicvoidactionPerformed(java.awt.event.ActionEvente
        vt) {
          jButton1ActionPerformed(evt);
        }
      });


      javax.swing.GroupLayoutlayout=newjavax.swing.GroupLayout(getContentPane());getCo
      ntentPane().setLayout(layout);
      layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
        nt.LEADING)
        .addGroup(layout.createSequentialGroup()
          .addGap(48,48,48)
          .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
          G)
            .addComponent(jLabel2)
            .addGroup(layout.createSequentialGroup()
              .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
              DING)
                .addGroup(layout.createSequentialGroup()
                  .addGap(11,11,11)
                  .addComponent(lname,
javax.swing.GroupLayout.PREFERRED_SIZE,
47,javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(layout.createSequentialGroup()
                  .addGap(10,10,10)


.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                  .addComponent(jLabel1)
                  .addComponent(jLabel3)
                  .addComponent(jLabel4)
```

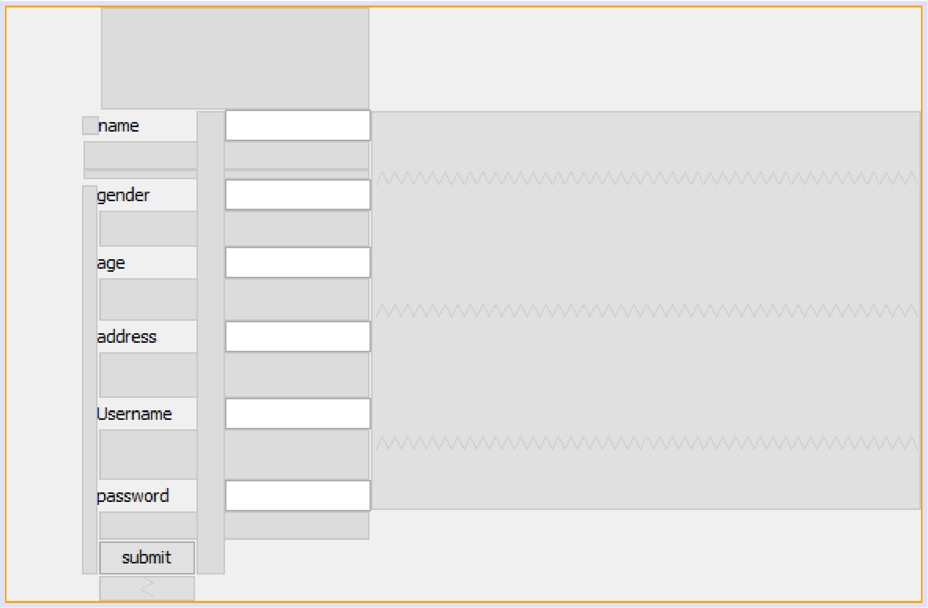
```
                      .addComponent(jLabel5)
                      .addComponent(jLabel6)
                      .addComponent(jButton1))))
                .addGap(18,18,18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
                DING,
false)
                  .addComponent(name)
                  .addComponent(gender)
                  .addComponent(age)
                  .addComponent(address)
                  .addComponent(username)
                  .addComponent(password,
javax.swing.GroupLayout.DEFAULT_SIZE, 93,Short.MAX_VALUE))))
           .addContainerGap(350,Short.MAX_VALUE))
      );
      layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
        LEADING)
        .addGroup(layout.createSequentialGroup()
           .addGap(65,65,65)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
             .addComponent(lname)
             .addComponent(name,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(18,18,18)
           .addComponent(jLabel2)
           .addGap(6,6,6)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
             .addComponent(jLabel3)
             .addComponent(gender,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(23,23,23)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
           .addComponent(jLabel1)
             .addComponent(age,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(27,27,27)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
           .addComponent(jLabel4)
             .addComponent(address,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(29,29,29)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
           .addComponent(jLabel5)
             .addComponent(username,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(32,32,32)
           .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELI
           NE)
           .addComponent(jLabel6)
              .addComponent(password,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,ja
vax.swing.GroupLayout.PREFERRED_SIZE))
           .addGap(18,18,18)
         .addComponent(jButton1)
           .addContainerGap(16,Short.MAX_VALUE))
      );


      pack();
   }//</editor-fold>


   private void jButton1ActionPerformed(java.awt.event.ActionEvent
     evt) {Connectioncon=null;
    Statementstatement=null;
    try{
      Class.forName("com.mysql.jdbc.Driver
      ");con=
DriverManager.getConnection("jdbc:mysql://localhost:3306/summa?characterEncoding=latin1
&use Configs=maxPerformance","root", "admin");
      Statement
      stmt=con.createStatement();String cmd
      = "insert into passport
      values(";cmd+="'"+username.getText()
      +"',";
      cmd+="'"+password.getText()+"',";
      cmd +=
      "'"+name.getText()+"',
      ";cmd+=age.getText()+", ";
      cmd+="'"+gender.getText()+"', ";
      cmd +=
     "'"+address.getText()+"')";Syste
     m.out.println(cmd);stmt.execute
     (cmd);
     passport.Login i = new
     passport.Login();this.show(false);
     i.show(true);
}
    catch (Exception e)
      {e.printStackTrac
      e();
    }
   }


   /**
    * @param argsthecommandline arguments
    */
   publicstaticvoidmain(Stringargs[]){
      /*Setthe Nimbus lookandfeel*/
      //<editor-folddefaultstate="collapsed"desc="Lookandfeelsettingcode(optional)">
      /*IfNimbus(introducedinJavaSE6)isnotavailable,staywiththedefaultlook andfeel.
       *Fordetailsseehttp://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
       */t
      ry{
         for (javax.swing.UIManager.LookAndFeelInfo
info
:javax.swing.UIManager.getInstalledLookAndFeels(
)){
            if ("Nimbus".equals(info.getName()))
               {javax.swing.UIManager.setLookAndFeel(info.getClassNa
               me()); break;
            }
         }
      }catch(ClassNotFoundExceptionex){


java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
      }catch(InstantiationExceptionex){


java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
      }catch(IllegalAccessExceptionex){


java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
      }catch(javax.swing.UnsupportedLookAndFeelExceptionex){


java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.S
EVERE, null, ex);
      }
      //</editor-fold>



      /* Create and display the form
      */java.awt.EventQueue.invokeLater(newRunna
      ble(){
         publicvoidrun() {
            newNewJFrame().setVisible(true);
         }
      });



   }


   // Variables declaration - do not modifyprivate javax.swing.JTextField address;private
   javax.swing.JTextField age;private javax.swing.JTextField gender;private
   javax.swing.JButton jButton1;private javax.swing.JLabel jLabel1;private
```

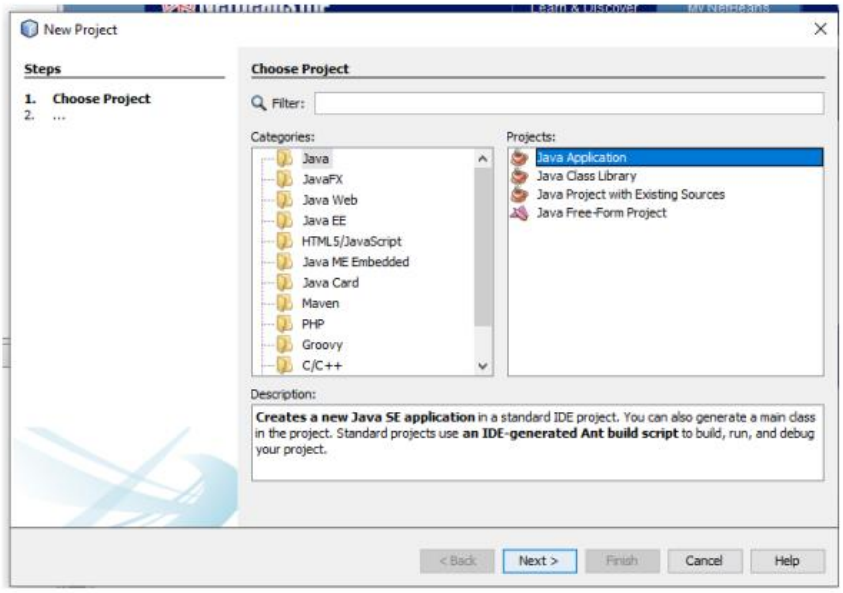
```
    javax.swing.JLabel jLabel2;private javax.swing.JLabel jLabel3;private javax.swing.JLabel
    jLabel4;private javax.swing.JLabel jLabel5;private javax.swing.JLabel jLabel6;private
    javax.swing.JLabellname;private javax.swing.JTextField name;private
    javax.swing.JTextField password;privatejavax.swing.JTextFieldusername;
    //Endofvariablesdeclaration
  }
```

**Result:**

Thus the implementation of passport management system has been succesfully completed and verified.







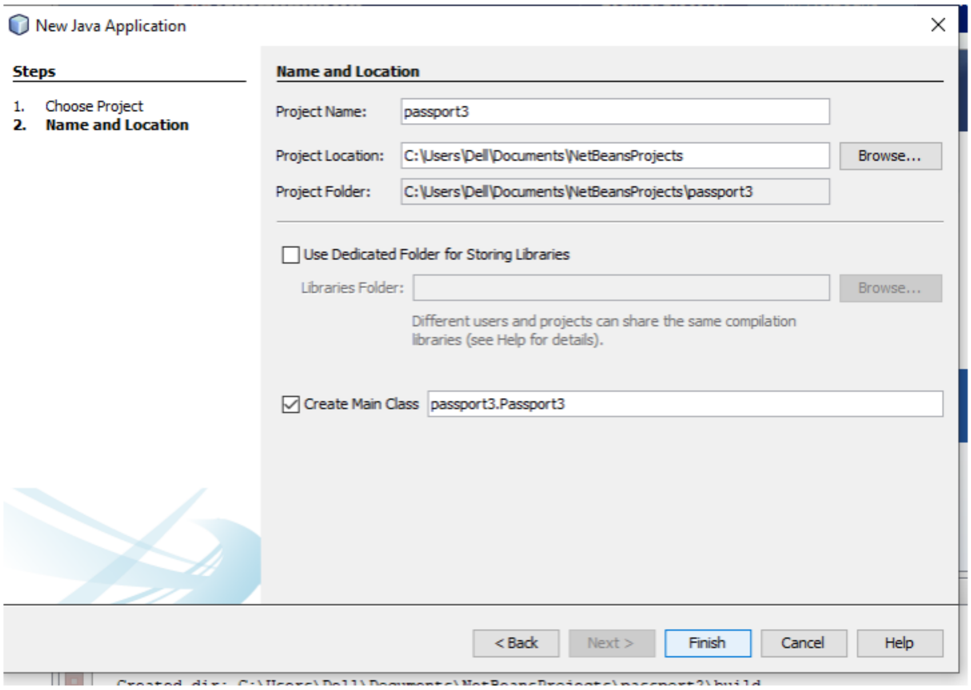
**Creates a new Java SE application** in a standard IDE project. You can also generate a main class in the project. Standard projects use **an IDE-generated Ant build script** to build, run, and debug your project.

# Ex.No:2 Exam Registration

Step 1: Create new project:

• Choose java application.

- Enter project name



Project Name: passport3

Project Location: C:\Users\Dell\Documents\NetBeansProjects

Project Folder: C:\Users\Dell\Documents\NetBeansProjects\passport3

□ Use Dedicated Folder for Storing Libraries

Libraries Folder:

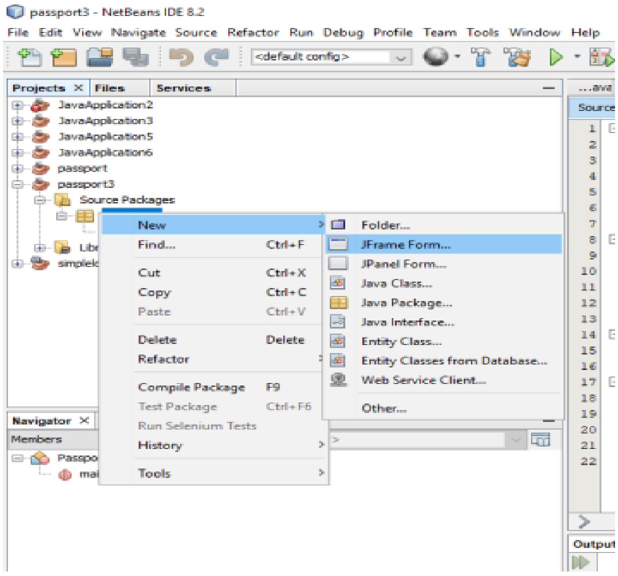
Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Main Class passport3.Passport3

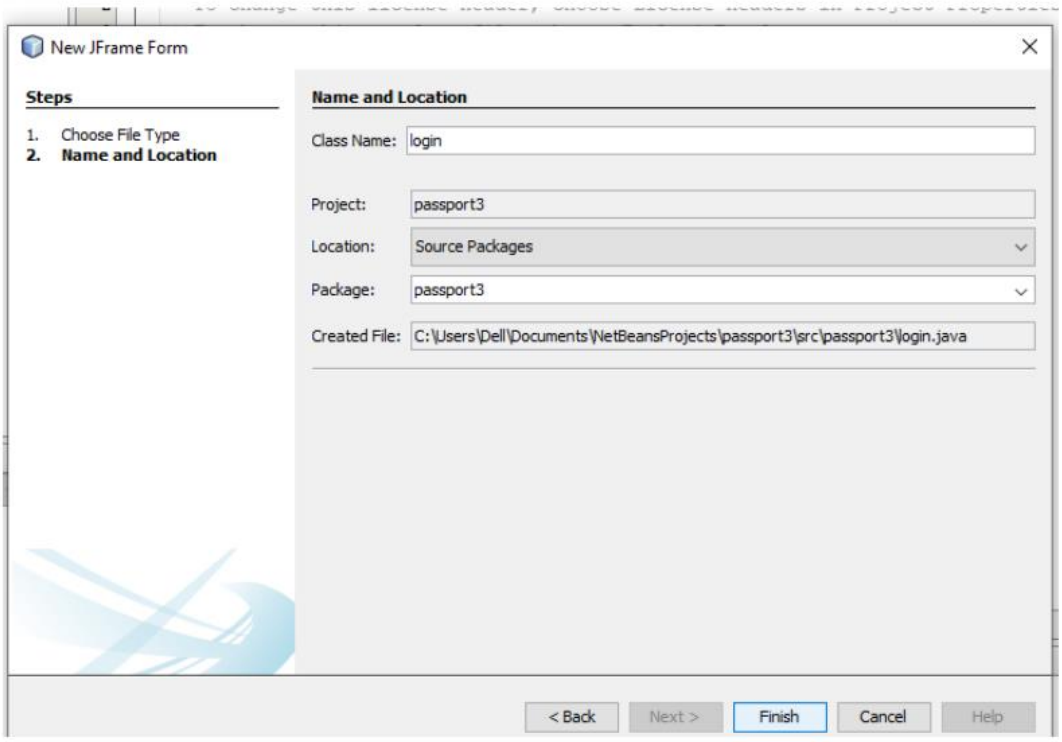
```
Created dir: C:\Users\Dell\Documents\NetBeansProjects\passport3\build
```

Create jFrame:

- To create jFrame right click project > New > JFrame Form



- Set suitable jFrame name.



Class Name: login

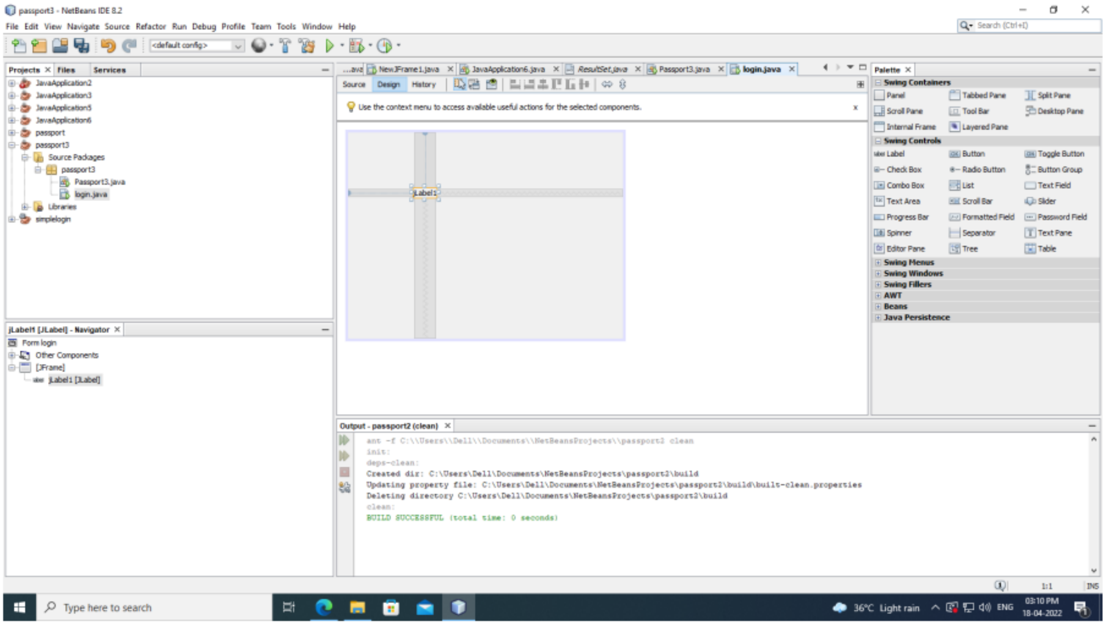
Project: passport3

Location: Source Packages

Package: passport3

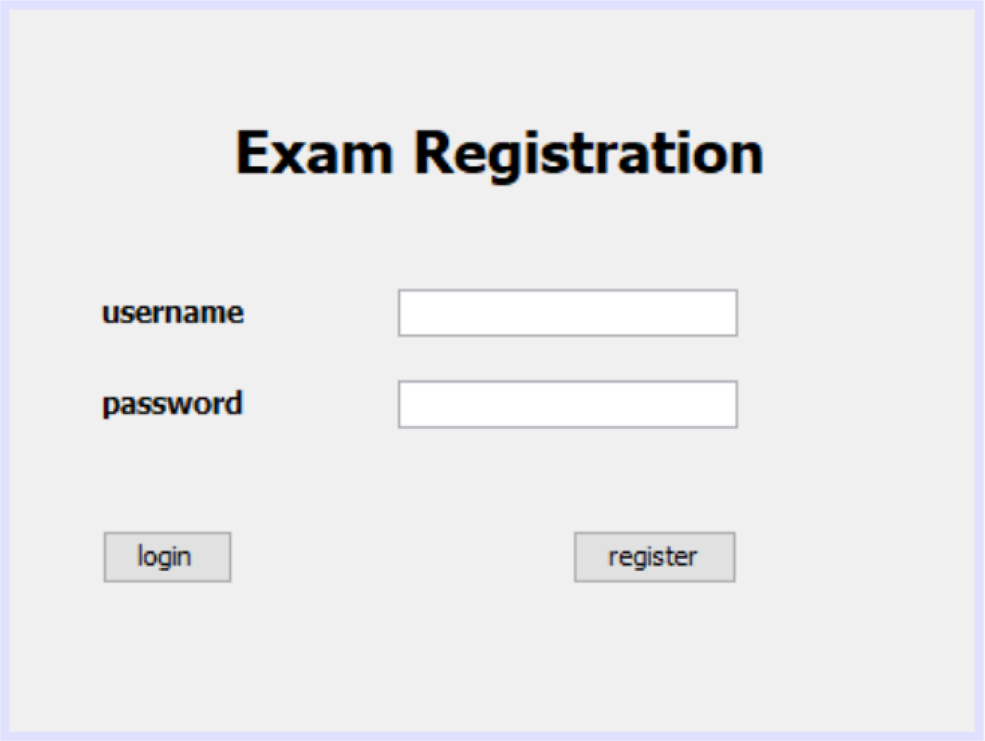
Created File: C:\Users\Dell\Documents\NetBeansProjects\passport3\src\passport3\login.java

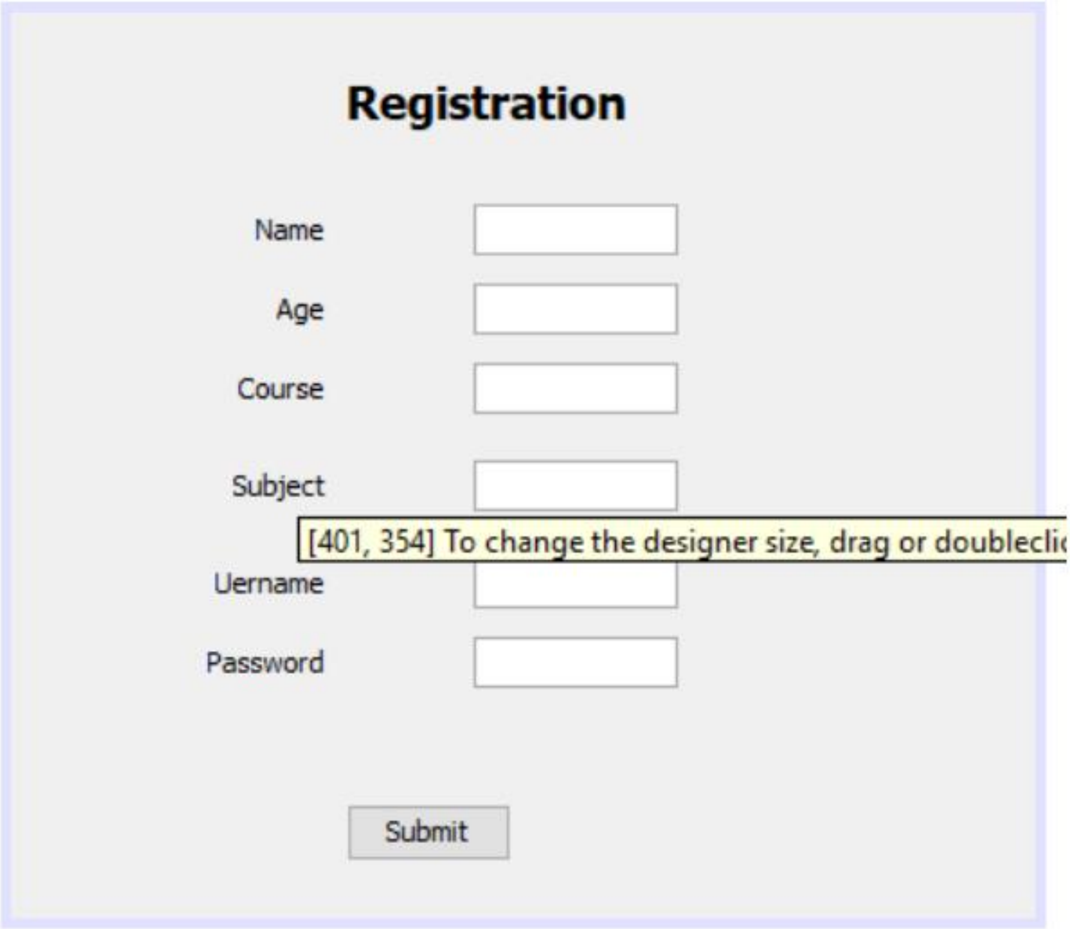
- Place suitable Attributes in the jFrame using Palette.



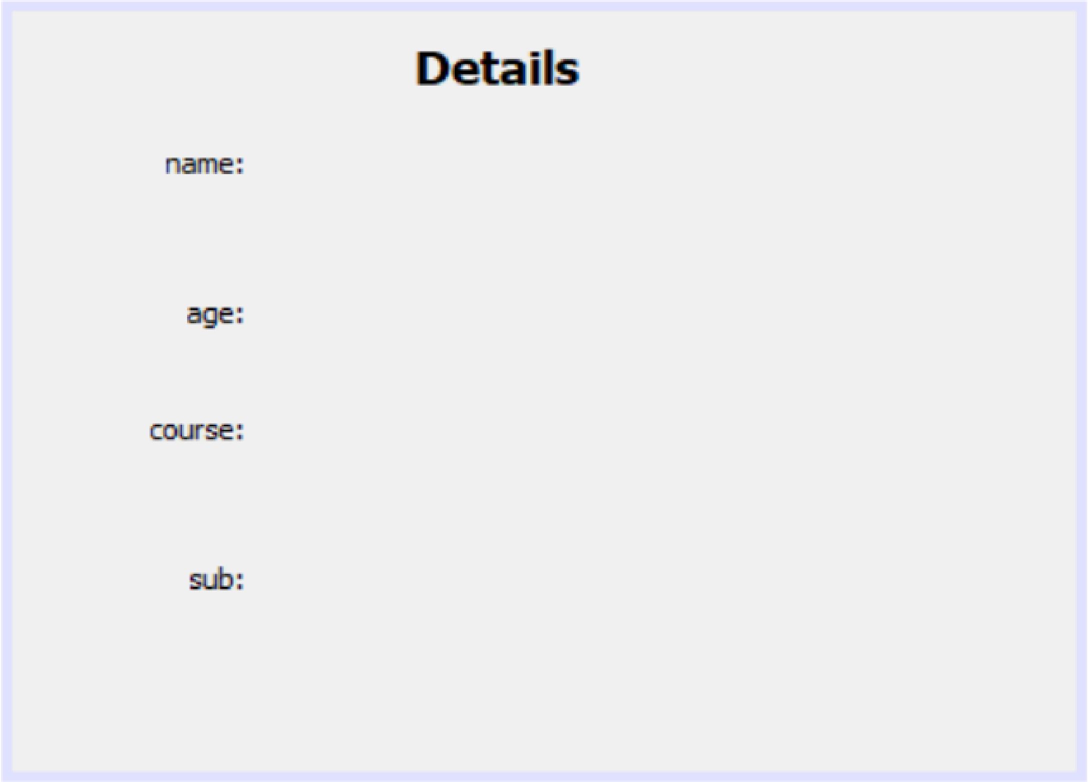
```
ant -f C:\\Users\\Dell\\Documents\\NetBeansProjects\\passport2 clean
init:
deps-clean:
Created dir: C:\Users\Dell\Documents\NetBeansProjects\passport2\build
Updating property file: C:\Users\Dell\Documents\NetBeansProjects\passport2\build\built-clean.properties
Deleting directory C:\Users\Dell\Documents\NetBeansProjects\passport2\build
clean:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Create following jFrames:



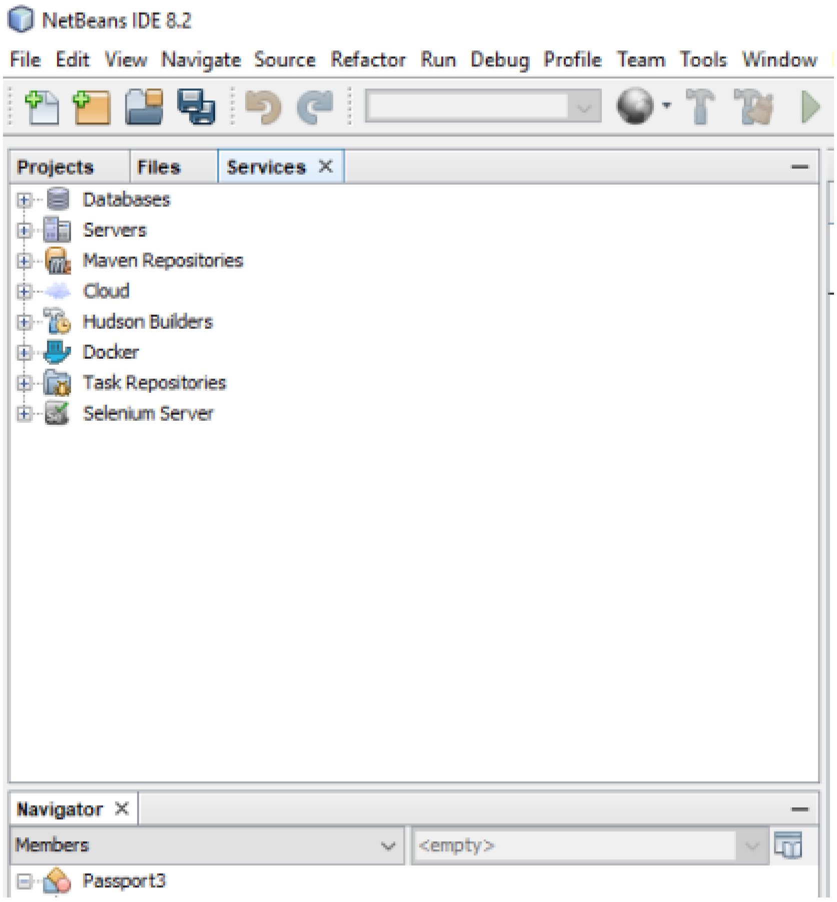


[401, 354] To change the designer size, drag or doublecli

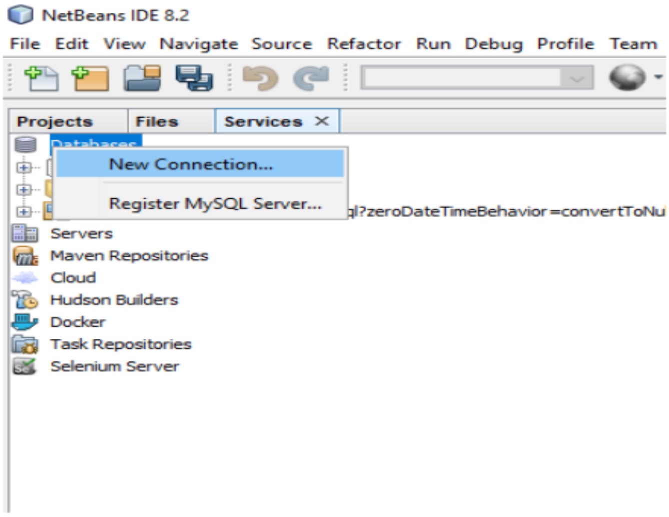


Connect Database:

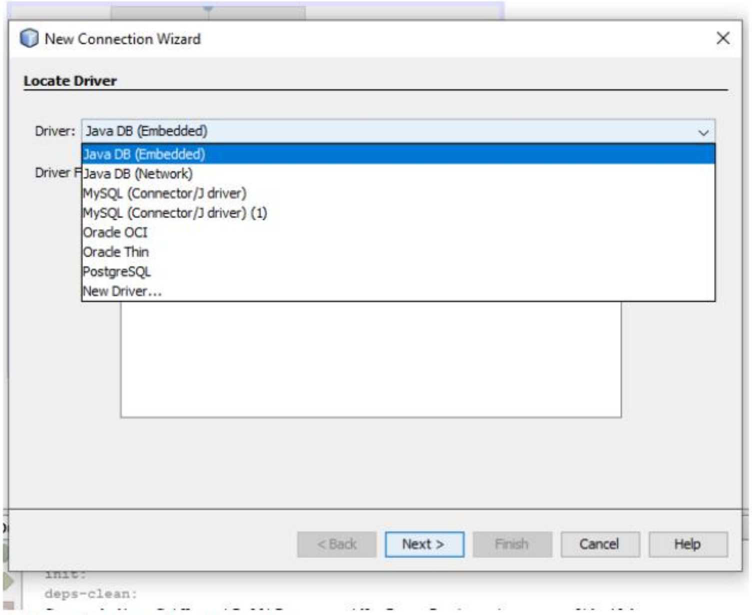
- Goto Services:



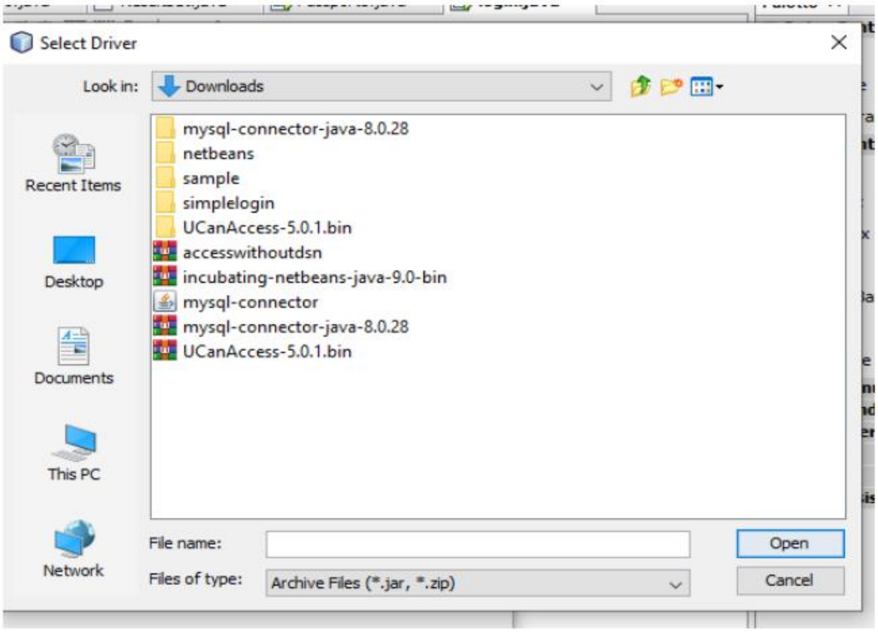
- Right click on database>New Connection.



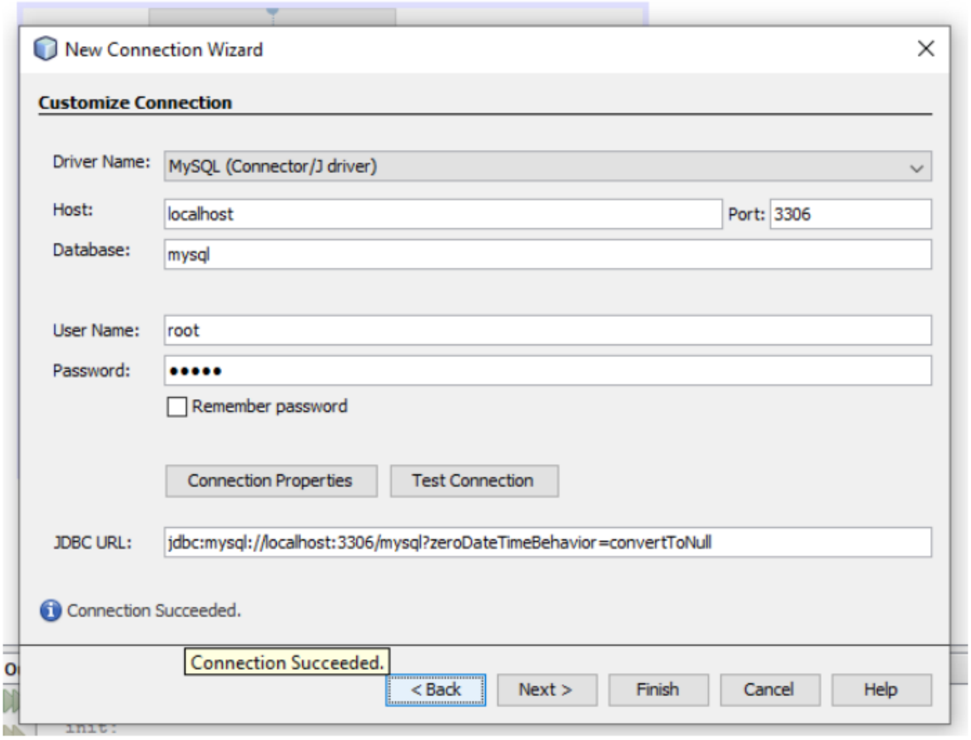
- Select MySQL



- Select the mysql-connector



- Enter your MySQL password and check connection with Test Connection.



New Connection Wizard

Customize Connection

Driver Name: MySQL (Connector/J driver)

Host: localhost Port: 3306

Database: mysql

User Name: root

Password: •••••

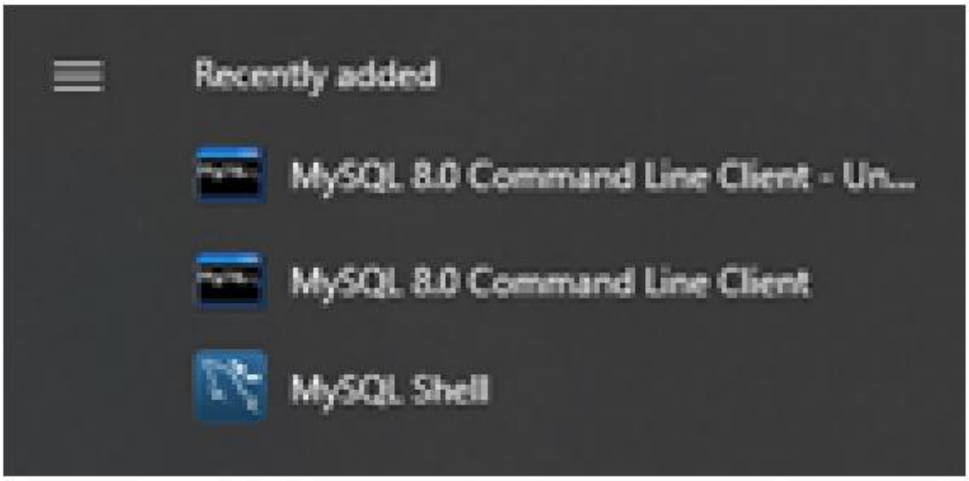
☐ Remember password

JDBC URL: jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertToNull

Connection Succeeded.

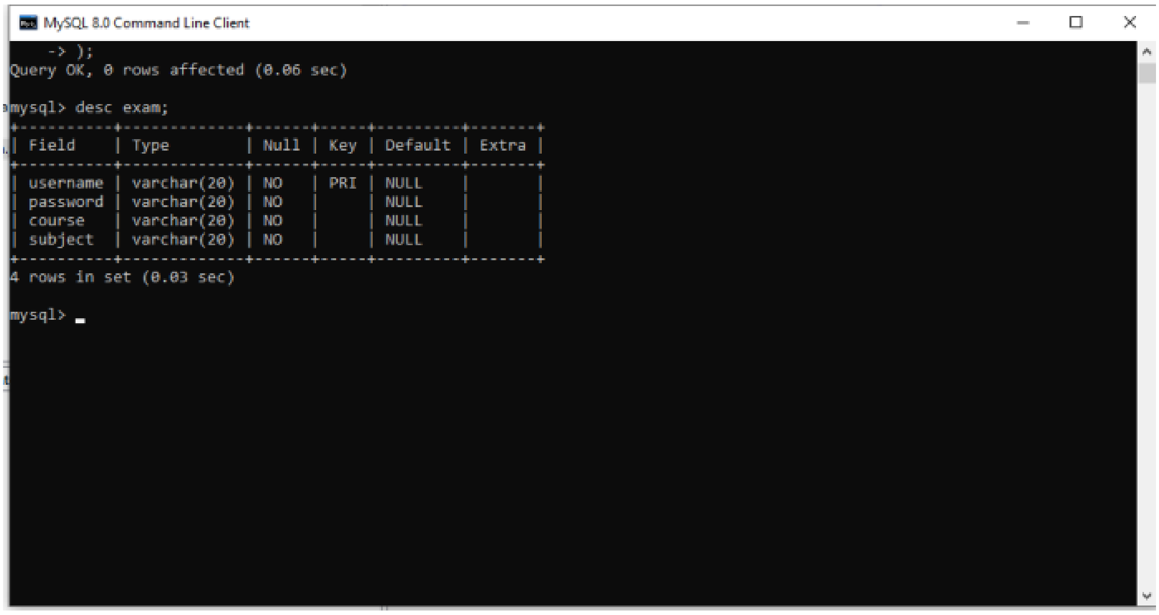
## Create table in database:

- Open MySQL Command line client


Recently added
MySQL 8.0 Command Line Client - Un...
MySQL 8.0 Command Line Client
MySQL Shell

• Create database using:
    Create database database_name;
• Change to database using:
    Use database_name;
• Create suitable table for your project.
    Using your knowledge from DBMS lab.

For example:

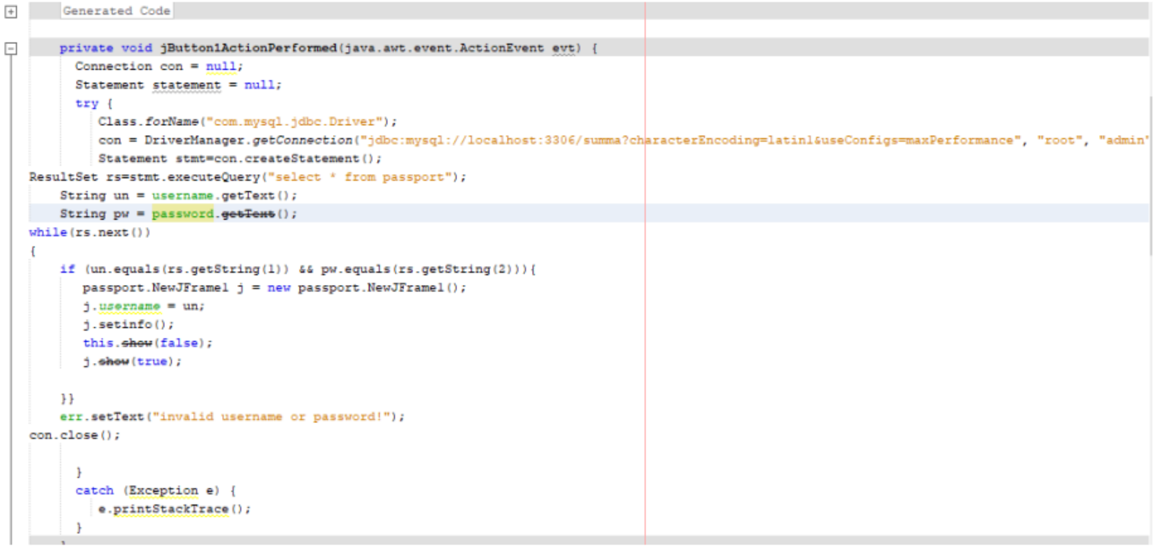


MySQL 8.0 Command Line Client

```
    -> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc exam;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(20) | NO   | PRI | NULL    |       |
| password | varchar(20) | NO   |     | NULL    |       |
| course   | varchar(20) | NO   |     | NULL    |       |
| subject  | varchar(20) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
4 rows in set (0.03 sec)

mysql> _
```

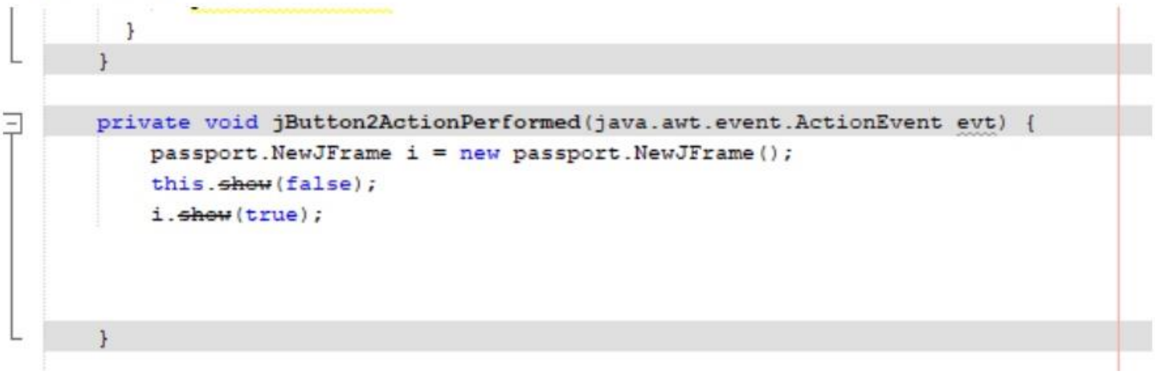
Programs:

Login>login(button):



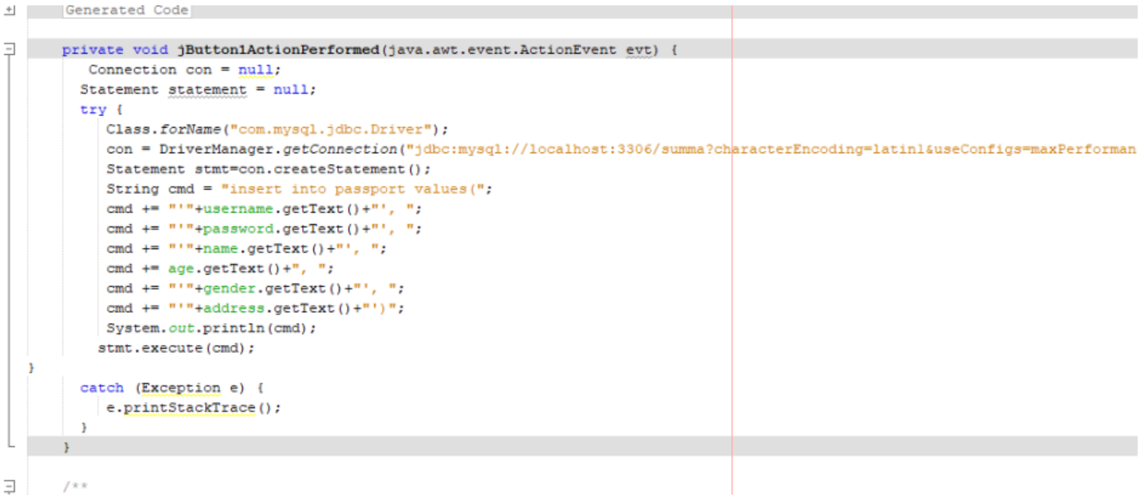
```
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      Connection con = null;
      Statement statement = null;
      try {
         Class.forName("com.mysql.jdbc.Driver");
         con = DriverManager.getConnection("jdbc:mysql://localhost:3306/summa?characterEncoding=latin1&useConfigs=maxPerformance", "root", "admin"
         Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from passport");
    String un = username.getText();
    String pw = password.~~getText~~();
while(rs.next())
{
    if (un.equals(rs.getString(1)) && pw.equals(rs.getString(2))){
       passport.NewJFrame1 j = new passport.NewJFrame1();
       j.username = un;
       j.setinfo();
       this.~~show~~(false);
       j.~~show~~(true);

    }}
    err.setText("invalid username or password!");
con.close();

      }
      catch (Exception e) {
         e.printStackTrace();
      }
```

Login>Register(button):



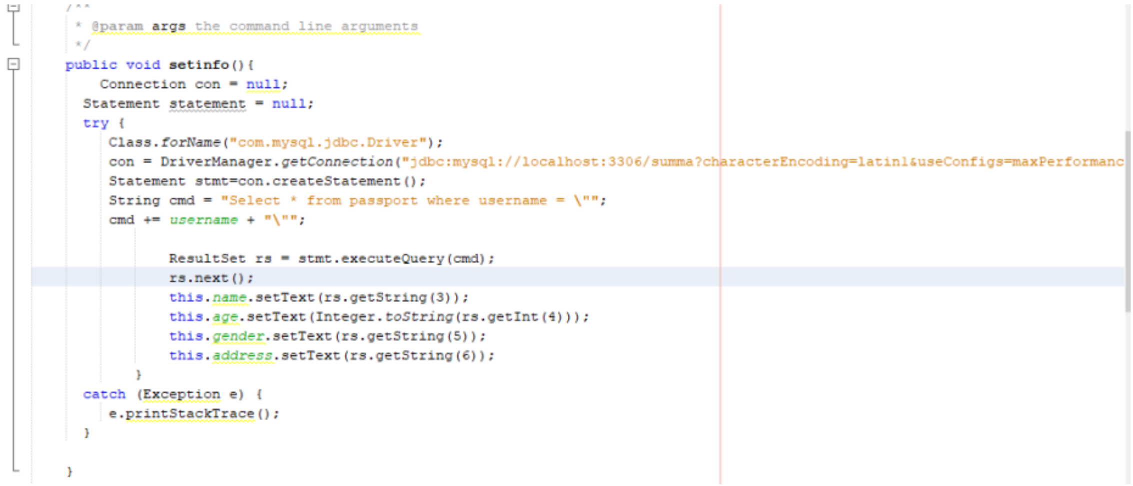
```
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        passport.NewJFrame i = new passport.NewJFrame();
        this.~~show~~(false);
        i.~~show~~(true);



    }
```

Register(jFrame)>submit:



```
    Generated Code

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
         Connection con = null;
        Statement statement = null;
        try {
           Class.forName("com.mysql.jdbc.Driver");
           con = DriverManager.getConnection("jdbc:mysql://localhost:3306/summa?characterEncoding=latin1&useConfigs=maxPerforman
           Statement stmt=con.createStatement();
           String cmd = "insert into passport values(";
           cmd += "'"+username.getText()+"', ";
           cmd += "'"+password.getText()+"', ";
           cmd += "'"+name.getText()+"', ";
           cmd += age.getText()+", ";
           cmd += "'"+gender.getText()+"', ";
           cmd += "'"+address.getText()+"')";
           System.out.println(cmd);
          stmt.execute(cmd);
}
        catch (Exception e) {
           e.printStackTrace();
        }
    }

    /**
```

Create a method setinfo in Display info(jFrame):



```
    /**
     * @param args the command line arguments
     */
    public void setinfo(){
         Connection con = null;
     Statement statement = null;
     try {
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/summa?characterEncoding=latin1&useConfigs=maxPerformanc
        Statement stmt=con.createStatement();
        String cmd = "Select * from passport where username = \"";
        cmd += username + "\"";

               ResultSet rs = stmt.executeQuery(cmd);
               rs.next();
               this.name.setText(rs.getString(3));
               this.age.setText(Integer.toString(rs.getInt(4)));
               this.gender.setText(rs.getString(5));
               this.address.setText(rs.getString(6));
           }
     catch (Exception e) {
        e.printStackTrace();
     }

    }
```

**Result:**

Thus the implementation of Exam registration system has been succesfully completed and verified.

**Ex.No:3** **Stock maintenance system**

## Implementation of Stock Maintenance System:

```
importjava.awt.Dimension;imp
ortjava.awt.Toolkit;
import
java.awt.event.ActionEvent;import
java.awt.event.ActionListener;imp
ortjava.sql.ResultSet;
importjava.sql.Statement;
import
java.text.SimpleDateFormat;imp
ortjava.util.Date;
import
javax.swing.JOptionPane;imp
ortjavax.swing.Timer;
importjavax.swing.table.DefaultTableModel;


/*
 * Tochangethislicenseheader,choose LicenseHeadersinProjectProperties.
 * Tochangethistemplatefile, chooseTools|Templates
 * andopen thetemplate intheeditor.
 */


/**
 *
 * @authoracer
 */
public class Home extends
        javax.swing.JFrame {pop_uppop=
        newpop_up();
    public Home()
        {initComponent
        s();Time();
        setLocationRelativeTo(nu
        ll);Toolkittoolkit
        =getToolkit();
```

```
  Dimensionsize=toolkit.getScreenSize();
  setLocation(size.width/ 2-getWidth()/2,size.height/2-getHeight()/2);
}
   publicvoidTime(){


  new Timer(0, new
     ActionListener() {@Override
     public void actionPerformed(ActionEvent
        e) {Date d=newDate();
        SimpleDateFormat s = new
        SimpleDateFormat("hh:mm:ss
        a");time.setText(s.format(d));
     }
  }
  ).start();
}
/**
 * Thismethodis calledfromwithintheconstructortoinitialize theform.
 * WARNING:DoNOTmodifythiscode. The content ofthismethodisalways
 * regeneratedbythe FormEditor.
 */@SuppressWarnings("unchec
ked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">privatevoidinitComponents(){

  jPanel1 = new javax.swing.JPanel();
  jPanel2 = new javax.swing.JPanel();
  jLabel1 = new javax.swing.JLabel();
  jLabel3 = new javax.swing.JLabel();
  jLabel5 = new javax.swing.JLabel();
  jLabel6 = new javax.swing.JLabel();
  time=newjavax.swing.JLabel();
  jPanel3 = new javax.swing.JPanel();
  HOME = new javax.swing.JPanel();
  jLabel2 = new javax.swing.JLabel();
  STOCKS=newjavax.swing.JPanel();
  jScrollPane1 = new javax.swing.JScrollPane();
  jTable1=newjavax.swing.JTable();
  jButton1 = new javax.swing.JButton();
  jButton2 = new javax.swing.JButton();
  jButton3 = new javax.swing.JButton();
  searchBox = new javax.swing.JTextField();
  jLabel4 = new javax.swing.JLabel();
  jButton4=newjavax.swing.JButton();

  setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLO
  SE); setUndecorated(true);

  jPanel1.setBackground(new java.awt.Color(255, 255,
  255));jPanel1.setLayout(neworg.netbeans.lib.awtextra.AbsoluteLayout());

  jPanel2.setBackground(new java.awt.Color(85, 143,
    245));jPanel2.setLayout(neworg.netbeans.lib.awtextra.AbsoluteLayout());

    jLabel1.setForeground(newjava.awt.Color(153,153,153));
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/icons8_shutdown_30px.png")));//NOI18
N
    jLabel1.setBorder(javax.swing.BorderFactory.createLineBorder(newjava.awt.Color(153,15
    3,
153)));
    jLabel1.addMouseListener(new
      java.awt.event.MouseAdapter()
      {publicvoidmouseClicked(java.awt.event.MouseEventevt)
      {
        jLabel1MouseClicked(evt);
      }
    });
    jPanel2.add(jLabel1,neworg.netbeans.lib.awtextra.AbsoluteConstraints(830,0,57,50));

    jLabel3.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/icons8_home_40px.png")));//NOI18N
    jLabel3.setBorder(javax.swing.BorderFactory.createMatteBorder(1,1,1,1,new
java.awt.Color(153,153,153)));
    jLabel3.addMouseListener(new
      java.awt.event.MouseAdapter()
      {publicvoidmouseClicked(java.awt.event.MouseEventevt)
      {
        jLabel3MouseClicked(evt);
      }
    });
    jLabel3.addKeyListener(new
      java.awt.event.KeyAdapter()
      {publicvoidkeyPressed(java.awt.event.KeyEventevt
      ){
        jLabel3KeyPressed(evt);
      }
   });
    jPanel2.add(jLabel3,neworg.netbeans.lib.awtextra.AbsoluteConstraints(0,0,50,50));

    jLabel5.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel5.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/icons8_out_of_stock_30px_1.pn
g"))); //NOI18N
    jLabel5.setBorder(javax.swing.BorderFactory.createMatteBorder(1,1,1,1,new
java.awt.Color(153,153,153)));
    jLabel5.addMouseListener(new
      java.awt.event.MouseAdapter()
      {publicvoidmouseClicked(java.awt.event.MouseEventevt)
      {
        jLabel5MouseClicked(evt);
      }
    });
    jLabel5.addKeyListener(new
      java.awt.event.KeyAdapter()
      {publicvoidkeyPressed(java.awt.event.KeyEventevt
      ){
        jLabel5KeyPressed(evt);
      }
    });
    jPanel2.add(jLabel5,neworg.netbeans.lib.awtextra.AbsoluteConstraints(50,0,60,50));


    jLabel6.setFont(newjava.awt.Font("Perpetua",0,24));//NOI18N
    jLabel6.setForeground(new java.awt.Color(255, 255,
    255));jLabel6.setText("StockManagementSystem");
    jPanel2.add(jLabel6,neworg.netbeans.lib.awtextra.AbsoluteConstraints(300,10,-1,30));


    time.setBackground(new java.awt.Color(255, 255,
    255));time.setFont(new java.awt.Font("Trebuchet MS", 2,
    14)); // NOI18Ntime.setForeground(new java.awt.Color(255,
    255, 255));time.setText("jLabel7");
    jPanel2.add(time,neworg.netbeans.lib.awtextra.AbsoluteConstraints(620,17,160,20));


    jPanel1.add(jPanel2,neworg.netbeans.lib.awtextra.AbsoluteConstraints(0,0,890,50));


    jPanel3.setBackground(newjava.awt.Color(255,255,255));
    jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(newjava.awt.Color(204,20
    4,
204)));
    jPanel3.setLayout(newjava.awt.CardLayout());


    HOME.setBackground(newjava.awt.Color(255,255,255));


    jLabel2.setIcon(newjavax.swing.ImageIcon(getClass().getResource("/icon/161001
4526305.jpg")));//NOI18N


    javax.swing.GroupLayoutHOMELayout=newjavax.swing.GroupLayout(HOME);HOME.s
    etLayout(HOMELayout);
    HOMELayout.setHorizontalGroup(HOMELayout.createParallelGroup(javax.swing.Group
      Layout.Alignment.LEADING)
      .addComponent(jLabel2,javax.swing.GroupLayout.DEFAULT_SIZE,888,Short.MAX_
      VALUE)
    );
    HOMELayout.setVerticalGroup(HOMELayout.createParallelGroup(javax.swing.GroupLa
      yout.Alignment.LEADING)
      .addComponent(jLabel2,javax.swing.GroupLayout.DEFAULT_SIZE,518,Short.MAX_
      VALUE)
    );
```

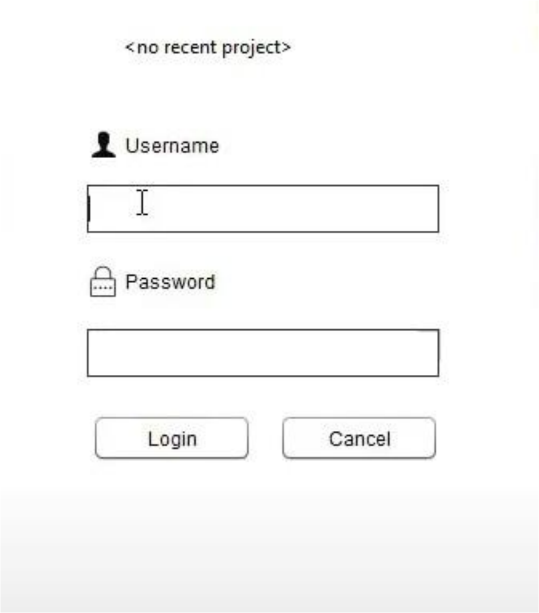
```
jPanel3.add(HOME,"card2");


STOCKS.setBackground(newjava.awt.Color(255,255,255));


jTable1.setModel(new
  javax.swing.table.DefaultTableModel(newObject[][]{

  },
  newString[]{
    "Stock#","Stockname","Case","StockDate","PersoninCharge"
  }
){
  boolean[] canEdit = new boolean
    [] {false,false, false,true,true
  };


  public boolean isCellEditable(int rowIndex, int
    columnIndex) {returncanEdit [columnIndex];
  }
});
jTable1.setGridColor(newjava.awt.Color(0,255,0));
jTable1.setSelectionBackground(new java.awt.Color(102, 102,
255));jTable1.addMouseListener(newjava.awt.event.MouseAd
apter(){
  publicvoidmouseClicked(java.awt.event.MouseEventevt){
    jTable1MouseClicked(evt);
  }
  public void mouseEntered(java.awt.event.MouseEvent
    evt) {jTable1MouseEntered(evt);
  }
});
jScrollPane1.setViewportView(jTable1);


jButton1.setBackground(newjava.awt.Color(255,255,255));j
Button1.setText("CREATE");
jButton1.addActionListener(new
  java.awt.event.ActionListener()
  {publicvoidactionPerformed(java.awt.event.ActionEvente
  vt){
    jButton1ActionPerformed(evt);
  }
});


jButton2.setBackground(newjava.awt.Color(255,255,255));j
Button2.setText("UPDATE");
jButton2.addActionListener(new
  java.awt.event.ActionListener()
  {publicvoidactionPerformed(java.awt.event.ActionEvente
         vt){
            jButton2ActionPerformed(evt);
         }
      });


      jButton3.setBackground(newjava.awt.Color(255,255,255));j
      Button3.setText("DELETE");
      jButton3.addActionListener(new
         java.awt.event.ActionListener()
         {publicvoidactionPerformed(java.awt.event.ActionEvente
         vt){
            jButton3ActionPerformed(evt);
         }
      });

      searchBox.setBorder(javax.swing.BorderFactory.createLineBorder(newjava.awt.Color(20
      4,204,
204)));
      searchBox.addKeyListener(new
         java.awt.event.KeyAdapter()
         {publicvoidkeyReleased(java.awt.event.KeyEventevt){
            searchBoxKeyReleased(evt);
         }
      });


      jLabel4.setIcon(newjavax.swing.ImageIcon(getClass().getResource("/icon/icons8_search_
20px_1.png")));//NOI18N


      jButton4.setBackground(new java.awt.Color(255, 255,
      255));jButton4.setText("REFRESH");
      jButton4.addActionListener(new
         java.awt.event.ActionListener()
         {publicvoidactionPerformed(java.awt.event.ActionEvente
         vt){
            jButton4ActionPerformed(evt);
         }
      });


      javax.swing.GroupLayoutSTOCKSLayout=newjavax.swing.GroupLayout(STOCKS);ST
      OCKS.setLayout(STOCKSLayout);
      STOCKSLayout.setHorizontalGroup(STOCKSLayout.createParallelGroup(javax.swing.Gr
         oupLayout.Alignment.LEADING)
         .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 888,Short.MAX_VALUE)
         .addGroup(STOCKSLayout.createSequentialGroup()
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE,
129,javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
132,javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE,
121,javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE,
109,javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0,0,Short.MAX_VALUE))
      .addGroup(STOCKSLayout.createSequentialGroup()
        .addGap(24,24,24)
        .addComponent(jLabel4)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(searchBox, javax.swing.GroupLayout.PREFERRED_SIZE,
225,javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,Short.MAX_VALUE)
        )
    );
    STOCKSLayout.setVerticalGroup(STOCKSLayout.createParallelGroup(javax.swing.Grou
      pLayout.Alignment.LEADING)
      .addGroup(STOCKSLayout.createSequentialGroup()
        .addContainerGap()


.addGroup(STOCKSLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
                                    LEADING)
          .addComponent(searchBox, javax.swing.GroupLayout.PREFERRED_SIZE,
35,javax.swing.GroupLayout.PREFERRED_SIZE)
          .addComponent(jLabel4,
javax.swing.GroupLayout.Alignment.TRAILING,javax.swing.GroupLayout.PREFERRED_SI
ZE, 35,javax.swing.GroupLayout.PREFERRED_SIZE))
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
391,javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)


.addGroup(STOCKSLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG,false)
          .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE, 39,Short.MAX_VALUE)
          .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE)
          .addComponent(jButton3,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE)
          .addComponent(jButton4,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE))
        .addGap(0,30,Short.MAX_VALUE))
    );


    jPanel3.add(STOCKS,"card3");
    jPanel1.add(jPanel3,neworg.netbeans.lib.awtextra.AbsoluteConstraints(0,50,890,520));


    javax.swing.GroupLayoutlayout=newjavax.swing.GroupLayout(getContentPane());getCo
    ntentPane().setLayout(layout);
    layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignme
      nt.LEADING)
      .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE)
    );
    layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
      LEADING)
      .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,Sho
rt.MAX_VALUE)
    );


    pack();
  }//</editor-fold>


  privatevoidjLabel3KeyPressed(java.awt.event.KeyEventevt){
    // TODO add your handling code
    here:HOME.setVisible(true);STOC
    KS.setVisible(false);
  }

  privatevoidjLabel5KeyPressed(java.awt.event.KeyEventevt){
    //TODOaddyourhandlingcode here:
    HOME.setVisible(false);STOCKS.
    setVisible(true);
  }


  privatevoidjLabel3MouseClicked(java.awt.event.MouseEventevt){
    //TODOaddyourhandlingcode here:
     HOME.setVisible(true);STOCKS.
     setVisible(false);
  }


  privatevoidjLabel5MouseClicked(java.awt.event.MouseEventevt){
    //TODOaddyourhandlingcode here:
    HOME.setVisible(fals
    e);STOCKS.setVisibl
    e(true);try{
        DefaultTableModel table = (DefaultTableModel)
        jTable1.getModel();table.setRowCount(0);
        Statementstate=Source.mycon().createStatement();ResultS
        etrs=state.executeQuery("SELECT*FROM`tblstock`");

        while(rs.next()){
              Object o[] = {rs.getString("id"), rs.getString("stockname"),
rs.getString("quantity"),rs.getString("date"), rs.getString("person")};
              table.addRow(o);
            }
        } catch (Exception ex)
            {System.out.println(ex.getMessage(
            ));}

  }



  privatevoidjButton1ActionPerformed(java.awt.event.ActionEventevt){
     // TODO add your handling code
     here:pop.setVisible(true);pop.stock
     .setText("AddStock");

  }



  privatevoidjButton2ActionPerformed(java.awt.event.ActionEventevt){
     // TODO add your handling code
      here:if(searchBox.getText().equals
      ("")){

     JOptionPane.showMessageDialog(null,"Clickoneformthetable");



     }else{
       DefaultTableModel table = (DefaultTableModel)
       jTable1.getModel();intr= jTable1.getSelectedRow();

        String[] Array = {table.getValueAt(r,
        0).toString(),table.getValueAt(r,
        1).toString(),table.getValueAt(r,
        2).toString(),table.getValueAt(r,
        3).toString(),table.getValueAt(r,4).toString()
        };

        pop_uppop=newpop_up();

     pop.stockid.setText(Array[0]);pop.
     stockname.setText(Array[1]);
```

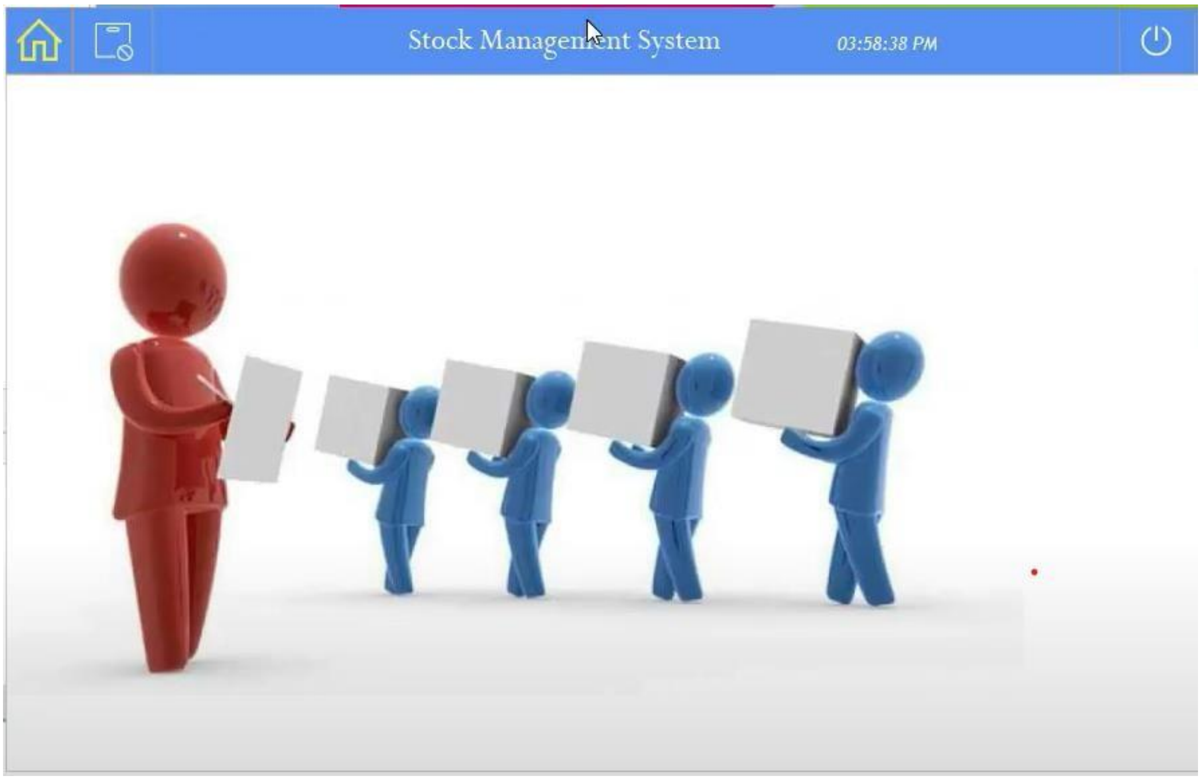
```
      pop.stockqty.setText(Array[2]);pop.stock
      date.setText(Array[3]);pop.stockperson.s
      etSelectedItem(Array[4]);pop.setVisible(t
      rue);pop.stock.setText("UpdateStock");
      }
    }


    privatevoidjButton3ActionPerformed(java.awt.event.ActionEventevt){
      // TODO add your handling code
      here:if(searchBox.getText().equals(
      "")){

      JOptionPane.showMessageDialog(null,"Clickoneformthetable");



      }else{
        DefaultTableModel table = (DefaultTableModel)
        jTable1.getModel();intr= jTable1.getSelectedRow();

          String[] Array = {table.getValueAt(r,
          0).toString(),table.getValueAt(r,
          1).toString(),table.getValueAt(r,
          2).toString(),table.getValueAt(r,
          3).toString(),table.getValueAt(r,4).toString()
          };

          pop_uppop=newpop_up();

      pop.stockid.setText(Array[0]);pop.
      stockname.setText(Array[1]);pop.st
      ockqty.setText(Array[2]);pop.stock
      date.setText(Array[3]);
      pop.stockperson.setSelectedItem(Array[4]
      );pop.setVisible(true);pop.stock.setText("
      Delete Stock");
      }
    }

    privatevoidsearchBoxKeyReleased(java.awt.event.KeyEventevt){
      //TODOaddyourhandlingcode here:
      try {
          DefaultTableModel table = (DefaultTableModel)
          jTable1.getModel();table.setRowCount(0);
          Statementstate=Source.mycon().createStatement();
          ResultSet rs = state.executeQuery("select * from tblstock wherestockname like '%"
  +searchBox.getText()+"%' orpersonlike'%"+searchBox.getText()+"%'");


            while(rs.next()){



              Object o[] = {rs.getString("id"), rs.getString("stockname"),
  rs.getString("quantity"),rs.getString("date"), rs.getString("person")};
              table.addRow(o);
          }
      } catch (Exception ex)
          {System.out.println(ex.getMessage());
      }
    }


    privatevoidjTable1MouseClicked(java.awt.event.MouseEventevt){
      // TODO add your handling code
       here:intt=jTable1.getSelectedRow(
       );
       Stringstockid=jTable1.getValueAt(t,0).toStrin
       g();searchBox.setText(stockid);jButton3.setEn
       abled(true);
       jButton2.setEnabled(true);

    }


    privatevoidjTable1MouseEntered(java.awt.event.MouseEventevt){
      //TODOaddyourhandlingcode here:



    }


    privatevoidjButton4ActionPerformed(java.awt.event.ActionEventevt){
      //TODOaddyourhandlingcode here:
        searchBox.setText("
        ");try{
            DefaultTableModel table = (DefaultTableModel)
            jTable1.getModel();table.setRowCount(0);
            Statementstate=Source.mycon().createStatement();ResultS
            etrs=state.executeQuery("SELECT*FROM`tblstock`");
            while(rs.next()){

              Object o[] = {rs.getString("id"), rs.getString("stockname"),
  rs.getString("quantity"),rs.getString("date"), rs.getString("person")};
              table.addRow(o);
            }
          } catch (Exception ex)
            {System.out.println(ex.getMessage(
            ));
          }
    }


    privatevoidjLabel1MouseClicked(java.awt.event.MouseEventevt){
      // TODO add your handling code
      here:dispose();
      Login log = new
      Login();log.setVisi
      ble(true);
    }



    /**
     * @param argsthecommandline arguments
     */
    publicstaticvoidmain(Stringargs[]){
      /*Setthe Nimbus lookandfeel*/
      //<editor-folddefaultstate="collapsed"desc="Lookandfeelsettingcode(optional)">
      /*IfNimbus(introducedinJavaSE6)isnotavailable,staywiththedefaultlook andfeel.
        *Fordetailsseehttp://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */try{
    for (javax.swing.UIManager.LookAndFeelInfo info
  :javax.swing.UIManager.getInstalledLookAndFeels()){
            if ("Nimbus".equals(info.getName()))
              {javax.swing.UIManager.setLookAndFeel(info.getClas
              sName()); break;
            }
          }
      } catch (ClassNotFoundException ex)
          {java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Leve
          l.SEVERE,
  null,ex);
      } catch (InstantiationException ex)
          {java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Leve
          l.SEVERE,
  null,ex);
      } catch (IllegalAccessException ex)
          {java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Leve
          l.SEVERE,
  null,ex);
      } catch (javax.swing.UnsupportedLookAndFeelException ex)
          {java.util.logging.Logger.getLogger(Home.class.getName()).log(java.util.logging.Leve
          l.SEVERE,
  null,ex);
      }
      //</editor-fold>



      /* Create and display the form
      */java.awt.EventQueue.invokeLater(newRun
      nable(){
          publicvoidrun() {
            newHome().setVisible(true);
      }
    });
```

```
    }


    // Variables declaration - do not
    modifyprivate javax.swing.JPanel
    HOME;private javax.swing.JPanel
    STOCKS;private
    javax.swing.JButton
    jButton1;private
    javax.swing.JButton
    jButton2;private
    javax.swing.JButton

    jButton3;private
    javax.swing.JButton
    jButton4;private
    javax.swing.JLabel
    jLabel1;private
    javax.swing.JLabel
    jLabel2;private
    javax.swing.JLabel
    jLabel3;private
    javax.swing.JLabel
    jLabel4;private
    javax.swing.JLabel
    jLabel5;private
    javax.swing.JLabel
    jLabel6;private
    javax.swing.JPanel
    jPanel1;private
    javax.swing.JPanel
    jPanel2;privatejavax.swing.JPanel
    jPanel3;
    private javax.swing.JScrollPane
    jScrollPane1;privatejavax.swing.JTabl
    e jTable1;
    privatejavax.swing.JTextFieldsearchBo
    x;privatejavax.swing.JLabeltime;
    //Endofvariablesdeclaration
  }
```

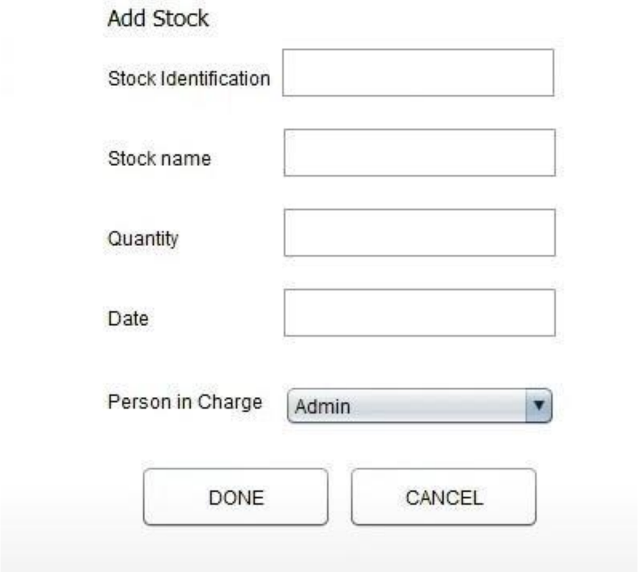
## LOGIN



**DASHBOARD**



## ADDSTOCK



Add Stock

Stock Identification

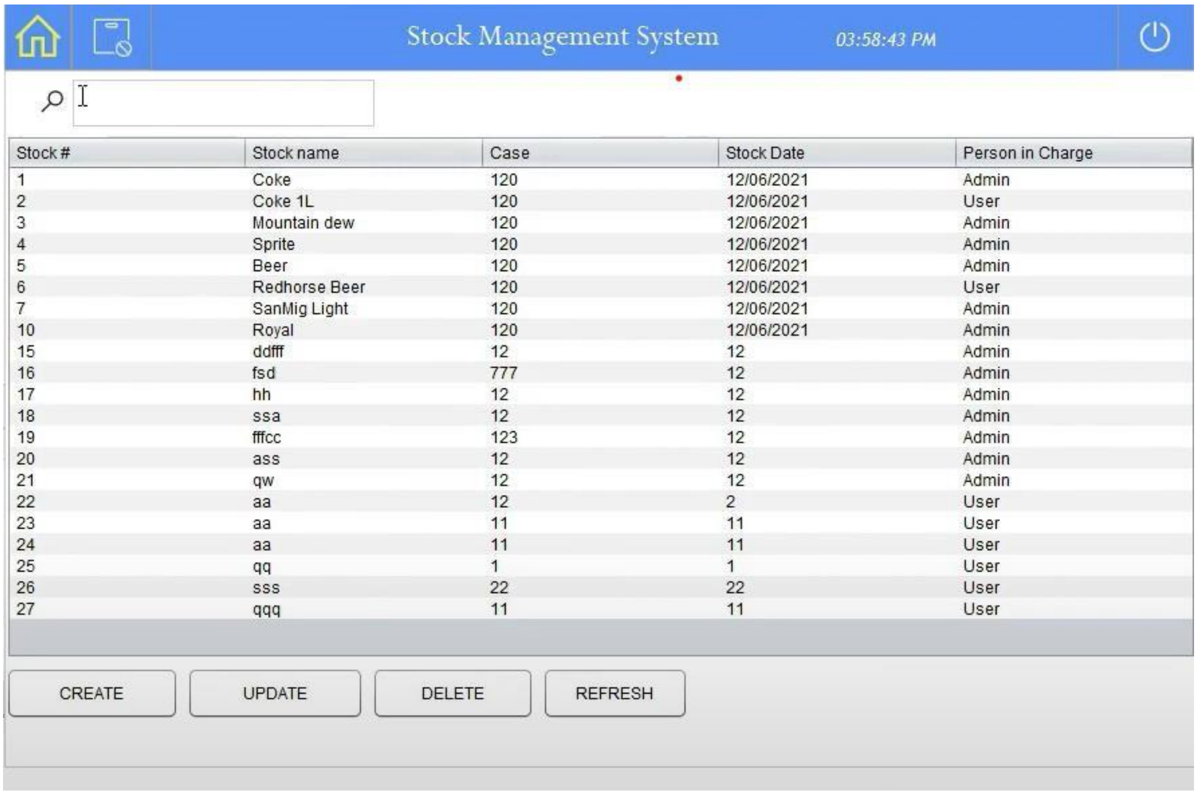
Stock name

Quantity

Date

Person in Charge: Admin

**STOCKMAINTENANCE**



Stock Management System

03:58:43 PM

| Stock # | Stock name | Case | Stock Date | Person in Charge |
|---|---|---|---|---|
| 1 | Coke | 120 | 12/06/2021 | Admin |
| 2 | Coke 1L | 120 | 12/06/2021 | User |
| 3 | Mountain dew | 120 | 12/06/2021 | Admin |
| 4 | Sprite | 120 | 12/06/2021 | Admin |
| 5 | Beer | 120 | 12/06/2021 | Admin |
| 6 | Redhorse Beer | 120 | 12/06/2021 | User |
| 7 | SanMig Light | 120 | 12/06/2021 | Admin |
| 10 | Royal | 120 | 12/06/2021 | Admin |
| 15 | ddfff | 12 | 12 | Admin |
| 16 | fsd | 777 | 12 | Admin |
| 17 | hh | 12 | 12 | Admin |
| 18 | ssa | 12 | 12 | Admin |
| 19 | fffcc | 123 | 12 | Admin |
| 20 | ass | 12 | 12 | Admin |
| 21 | qw | 12 | 12 | Admin |
| 22 | aa | 12 | 2 | User |
| 23 | aa | 11 | 11 | User |
| 24 | aa | 11 | 11 | User |
| 25 | qq | 1 | 1 | User |
| 26 | sss | 22 | 22 | User |
| 27 | qqq | 11 | 11 | User |

**Result:**

Thus the implementation of Stock maintenance system has been succesfully completed and verified.

## Content beyond Syllabus

**Ex No :**

**QUIZ SYSTEM**

**AIM:**

To implement the Quiz System using Rational Software.

ALGORITHM:

Step 1:

Identify the Actors and Use cases.

In this system, the actors and use cases are:

Actors:

User ,System and Database

.

Use cases:

Login,Option,answering questions,calculating marks and Display score card.

Step 2:

Build the relationship between actors and use cases.

2. ALGORITHM

FOR SEQUENCE DIAGRAM:

Step 1:

Identify the objects.

In this system, the objects are:

User, Systemand Database.

Step 2:

Identify the sequence of events

Step 3:

Login to the system to check the validity.

Step 4:

Select the option in the system.

Step 5:

Answer the questions for the selected options

Step 6:

Evaluate the answer and calculate the mark.

Step 7:

Save the score in the database

Step 8:

Display the score.

ALGORITHM

FOR CLASS DIAGRAM:

Step 1:

Create class diagram for each identified objects under sequence diagram.

Identified objects are: User, Systemand Database.

Step 2:

Class diagram for each object is divided into three parts.

Top portion represents the Class Name.

Middle portion represents the Attributes.

Bottom portion represents the Methods.

Procedure to write the software for the Quiz System using Rational Software tool
**Selecting the Software**:

1 Click start-> Rational Software->Rational Rose Enterprise Edition.
**Create the project**:
2. In the Create New Model window select VB6 Standard Framework->A screen appears; select the untitled project from the browser window and save it
**To draw the diagram:**
3. Right Click the Use case view->New->Use case diagram->Sequence diagram->Class diagram (name and save the files)
4. Click Use case diagram->using the tool box build the use case diagram.
5. Click Sequence diagram->using the tool box build the sequence diagram.
6.Click Class diagram-> using the tool box build the class diagram.
**Generate the coding form:**
7.Right click component view -> component diagram-> open specification -> select the stereotype as DLL-> select the language as visual basic->switch from general tab to realizes Tab-> select the classes you created in class diagram-> right click ->Assign->Ok.
8. Right click component view -> update code form model-> the tool window for converting Diagrams to code appears->next->finish->the skeleton code in VB is generated automatically for the class diagram designed.

## Viva questions

1. What is Object Oriented analysis & Design?
2. List the 4 phases in UP.
3. Compose your views on iterative Development and write it benefits.
4. Define UML.
5. Define Class Diagram? When to use Class Diagrams?
6. Define Use Case.
7. Point out the relationship used in Use case.
8. Discover the major Difference between Component and Deployment Diagram.
9. Classify the 3 kinds of actors in usecase.
10. Define State Chart Diagram? When to use State Diagram?
11. Compare Activity and state chart diagram? Mention the Elements of an Activity Diagram.
12. Define Aggregation and Composition.
13. Differentiate between method and message in object.
14. Formulate the purpose of Interaction Diagram.
15. Analyze the need of State chart Diagram.

16. Discuss the Strength and Weakness of the Use case Diagram.
17. Interpret the meaning of event, state and Transition.
18. Give the use of Package Diagram.
19. Compare and Contrast of Sequence and Collaboration diagram.
20. Illustrate the means of representing a node in a Deployment Diagram.