MAHENDRA COLLEGE OF ENGINEERING

Approved by AICTE and Affiliated by Anna University, Chennai

NAAC Accredited – Recognized U/S 2 (F) & 12 (B) of UGC Act 1956

Salem-Chennai Highway NH 79, Minnampalli, Salem-636106

DEPARTMENT OF INFORMATION TECHNOLOGY

CCS356 – OBJECT ORIENTED SOFTWARE ENGINEERING

LABORATORY

RECORD NOTE BOOK

CLASS: III YEAR / VI SEMESTER

MAHENDRA

COLLEGE OF ENGINEERING

NAAC Accredited

Approved by AICTE and Affiliated by Anna University, Chennai

Salem-Chennai Highway NH 79, Minnampalli, Salem-636106

Department of

_____

_____

LABORATORY RECORD

Certified to be the bonafide of work done by

Name: _____Register No: _____

Class: _____Branch: _____

Laboratory Name: _____

HEAD OF THE DEPARTMENT
DATE:

STAFF-IN-CHARGE

Submitted for the University Practical Examination on

_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

S.NO	DATE	TITLE	PAGE NO	MARK	SIGN

INTERNAL MARK

STAFF INCHARGE

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.

INSTITUTION VISION AND MISSION

VISION

Mahendra College of Engineering is committed to be a leader in Higher Education achieving excellence through world class learning environment for Science and Technology with a blend of advanced research to create ethical and competent professionals.

MISSION

- To provide a conductive atmosphere to impart innovative knowledge and commendable skills through quality education by continuous improvement and customization of teaching.
- To nurture research attitude and bring about tangible developments with dynamic Industry - Institute Interaction.
- To create society oriented citizens with professional ethics.

MAHENDRA COLLEGE OF ENGINEERING

SALEM-CAMPUS, ATTUR MAIN ROAD, MINNAMPALLI, SALEM -636 106.
DEPARTMENT OF INFORMATION TECHNOLOGY

DEPARTMENT VISION AND MISSION

VISION

To become a department, producing graduates with good technical skills in emerging areas of Information Technology, through value based education and research.

MISSION

- To provide exposure to students to the emerging technologies in Hardware and Software.
- To inculcate students with sound application knowledge.
- To establish strong Industry- Institute Interaction.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

To ensure graduates

- Have proficiency in programming skills to design, develop and apply appropriate techniques, to solve complex engineering problems.
- Have knowledge to build, automate and manage business solutions using cutting edge technologies.
- Have excitement towards research in applied computer technologies.

PROGRAM OUTCOMES (POs)

1. Engineering knowledge:

Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis:

Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions:

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal,

and environmental considerations.

4. Conduct investigations of complex problems:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usages:

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7.Environment and sustainability:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics:

Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work:

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communications:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12.Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CCS356 –OBJECT ORIENTED SOFTWARE ENGINEERING LABORATORY

COURSE OBJECTIVES:

- To understand Software Engineering Lifecycle Models
- To Perform software requirements analysis
- To gain knowledge of the System Analysis and Design concepts using UML.
- To understand software testing and maintenance approaches
- To work on project management scheduling using DevOps

LIST OF EXPERIMENTS:

1. Identify a software system that needs to be developed.
2. Document the Software Requirements Specification (SRS) for the identified system.
3. Identify use cases and develop the Use Case model.
4. Identify the conceptual classes and develop a Domain Model and also derive a Class Diagram from that.
5. Using the identified scenarios, find the interaction between objects and represent them using UML Sequence and Collaboration Diagrams
6. Draw relevant State Chart and Activity Diagrams for the same system.
7. Implement the system as per the detailed design
8. Test the software system for all the scenarios identified as per the usecase diagram
9. Improve the reusability and maintainability of the software system by applying appropriate design patterns.
10. Implement the modified system and test it for various scenarios

TOTAL: 30 PERIODS

SOFTWARE AND HARDWARE REQUIREMENTS:

Software Requirements	Argo UML, Star UML- Opensource
Hardware Requirements	Desktop Computer

COURSE OUTCOMES:

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

CO1: Compare various Software Development Lifecycle Models

CO2: Evaluate project management approaches as well as cost and schedule estimation strategies.

CO3: Perform formal analysis on specifications.

CO4: Use UML diagrams for analysis and design.

CO5: Architect and design using architectural styles and design patterns, and test the system

CO's-PO & PSO's MAPPING:

CO's	PO's												PSO's		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
1	2	2	1	2	2	-	-	-	-	1	1	2	2	2	1
2	2	3	2	3	2	-	-	-	2	2	3	2	3	2	1
3	2	3	2	1	1	-	-	-	2	2	3	2	2	3	1
4	2	3	2	2	3	-	-	-	2	2	3	2	2	3	1
5	2	3	1	2	2	-	-	-	-	-	-	1	3	2	2
AVg.	2	2	1	2	2	-	-	-	-	1	1	2	2	2	1

1-low,2-medium,3-high,'-'-nocorrelation

Study of ArgoUML & UML Diagrams

Aim:

To study about case tools and ArgoUML

ArgoUML

ArgoUML is an open source Unified Modeling Language (UML) modeling tool that includes support for all standard UML 1.4 diagrams. It runs on any Java platform and is available in ten languages.

FEATURES

- Support open standards extensively: UML, XMI, SVG, OCL and others.
- 100% Platform independent thanks to the exclusive use of Java
- Open Source, which allows extending or customizing.
- Cognitive features like
  - reflection-in-action
    - Design Critics
    - Corrective Automations (partially implemented)
    - "To Do" List
    - User model (partially implemented)
  - opportunistic design
    - "To Do" List
    - Checklists
  - Comprehension and Problem Solving
    - Explorer Perspectives
    - Multiple, Overlapping Views
    - Alternative Design Representations: Graphs, Text, or Table

Supported Diagrams by ArgoUML

- Use Case Diagrams
- Class Diagrams
- Behavior Diagrams
  - State chart Diagrams
  - Activity Diagrams
  - Interaction Diagrams
    - » Sequence Diagrams
    - » Collaboration Diagrams
- Implementation Diagrams
  - Component Diagrams
  - Deployment Diagrams

Use Case Diagrams

- Present a high-level view of system usage
- These diagrams show the functionality of a system or a class and howthe system interacts with the outside world.
- During analysis to capture the system requirements and to understand how the system should work.

During the design phase, use-case diagrams specify the behavior of the systemas implemented.

Class Diagram

- Helps you visualize the structural or static view of a system.
- Class diagrams show the relationships among class.
- Foundation for component and deployment diagrams.

Sequence Diagram

- Illustrates object interacts arranged in a time sequence
- This shows step-by-step what has to happen to accomplish something in theuse case and emphasize the sequence of events.

Collaboration Diagram

- Provides a view of the interactions or structured relationships between objects in current model.
- Emphasizes relation between objects.
- Used as the primary vehicle to describe interactions that express decision about system behavior.

State Chart Diagram

- To model the dynamic behaviors of individual classes or objects
- Used to model the discrete stages of an objects lifetime.

Activity Diagram

- Model the workflow of a business process and the sequence of activities in a process.
- Similar to a flowchart, l a workflow from activity to activity or from activity to state.
- It is help to understand the overall process.

Component Diagram

- A physical view of the current model and Show the organization and dependencies of software components, including source code, binary code, and executable components

Deployment Diagram

- Each model contains a single deployment diagram that shows the mapping of processes to hardware.

THE ArgoUML USER INTERFACE

Overview of the Window

The title bar of the window shows the following 4 parts of information,separated from each other by a dash.

- The current filename.
- The name of the currently active diagram.
- The name "ArgoUML".
- An asterisk (*). This item is only present if the current project file is "dirty", i.e. it is altered, but not yet saved. In other words, if the asterisk is absent, then the current file has not been altered.

The window comprises four sub-windows or panes namely,

- The Explorer pane,
- The Editing pane
- The Details Pane and
- The To-Do Pane.

THE EDITING PANE

The Toolbar

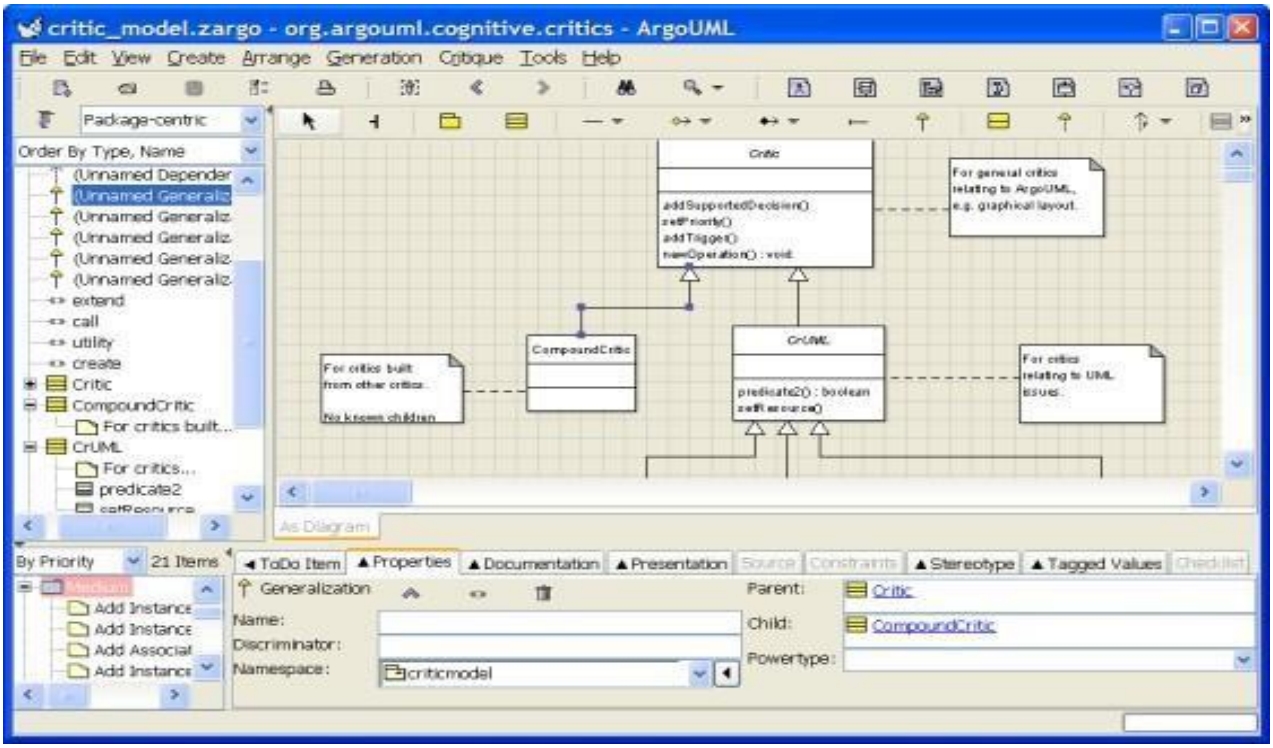
The toolbar at the top of the editing pane provides the main functions of thepane.

THE ArgoUML USER INTERFACE

Overview of the Window

The title bar of the window shows the following 4 parts of information, separated from each other by a dash.

- The current filename.
- The name of the currently active diagram.
- The name "ArgoUML".
- An asterisk (*). This item is only present if the current project file is "dirty", i.e. it is altered, but not yet saved. In other words, if the asterisk is absent, then the current file has not been altered.



The Toolbar consists of the following menus:

- File operations
- Edit operations
- View operations
- Create operations

File operations

-  New
-  Open Project...
-  Save Project
-  Project Properties
-  Print

Edit operations

-  Remove From Diagram

gram ".

-  Delete From Model
-  Configure Perspectives
-  Settings

View operations

-  Find...
-  Zoom

Create operations

- New Use Case Diagram
- New Class Diagram
- New Sequence Diagram
- New Collaboration Diagram
- New Statechart Diagram
- New Activity Diagram
- New Deployment Diagram

The tools fall into four categories.

- ***Layout tools.*** Provide assistance in laying out model elements on the diagram.
- ***Annotation tools.*** Used to annotate model elements on the diagram.
- ***Drawing tools.*** Used to add general graphic objects to diagrams.
- ***Diagram specific tools.*** Used to add UML model elements specific to a particular diagram type to the diagram.

- ***Layout Tools***

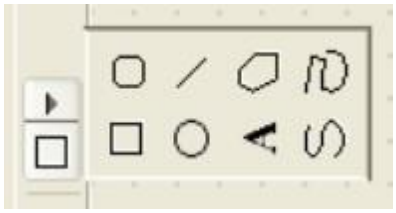
↑ **Select** - This tool provides for general selection of model elements on thediagram.

┤**Broom Tool** – This tool is used to sweep all model elements along.

- ***Annotation Tools***

Annotation tool - It is used to add a comment to a selected UML model element.

- ***Drawing Tools***



- `Rectangle`. Provides a rectangle.
- `Rounded Rectangle`. Provides a rectangle with rounded corners. There is no control over the degree of rounding.
- `Circle`. Provides a circle.
- `Line`. Provides a line.
- `Text`. Provides a text box. The text is entered by selecting the box and typing. Text is centered horizontally and after typing, the box will shrink to the size of the text. However it can be re-sized by dragging on the corners.
- `Polygon`. Provides a polygon. The points of the polygon are selected by button 1 click and the polygon closed with button 1 double click (which will link the final point to the first point).
- `Spline`. Provide an open spline. The control points of the spline are selected with button 1 and the last point selected with button 1 double click.
- `Ink`. Provide a polyline. The points are provided by button 1 motion.

USE CASE DIAGRAM SPECIFIC TOOLS

Actor - Add an actor to the diagram.

Use Case – Add a use case to the diagram.

Association - Add an association between two model elements selectedusing button 1 motion (from the first model element to the second).

The association tool selector

-  `Dependency`. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).
-  `Generalization`. Add a generalization between two model elements selected using button 1 motion (from the child to the parent).
-  `Extend`. Add an extend relationship between two model elements selected using button 1 motion (from the extended to the extending use case).
-  `Include`. Add an include relationship between two model elements selected using button 1 motion (from the including to the included use case).
-  `Add Extension Point`. Add an extension point to a selected use case.

CLASS DIAGRAM SPECIFIC TOOLS

Class diagrams are used for only one of the UML static structure diagrams, the class diagram itself. Object diagrams are represented on the ArgoUML deployment diagram.

ArgoUML uses the class diagram to show model structure through the use of packages.

-  `Package`. Add a package to the diagram.
-  `Class`. Add a class to the diagram. For convenience, when the mouse is over a selected class it displays two handles to left and right which may be clicked or dragged to form association relationships (or composition in case SHIFT has been pressed) and two handles top and bottom which may be dragged or clicked to form generalization and specialization relationships respectively.
- `Association`. Add an association between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of association offered here, `bi-directional` or `unidirectional`.
-  `Aggregation`. Add an aggregation between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of aggregation offered here, `bi-directional` or `unidirectional`.
-  `Composition`. Add an composition between two model elements selected using button 1 motion (from the first model element to the second). There are 2 types of composition offered here, `bidirectional` or `unidirectional`.

- Association-end. Add another end to an already existing association using button 1 (from the association middle to a class, or vice versa). This is the way to create so-calld N-ary associations.
- Generalization. Add a generalization between two model elements selected using button 1 (from the child to the parent).
- Interface. Add an interface to the diagram. For convenience, when the mouse is over a selected interface it displays a handle at the bottom which may be dragged to form a realization relationship (the target being the realizing class).
- Realization. Add a realization between a class and an interface selected using button 1 motion (from the realizing class to the realized interface).
- Dependency. Add a dependency between two model elements selected using button 1 motion (from the dependent model element). There are also 2 special types of dependency offered here, Permission (P) and Usage (U). A Permission is created by default with stereotype Import, and is used to import elements from one package into another.
- Attribute. Add a new attribute to the currently selected class. The attribute is given the default name newAttr of type int and may be edited by button 1 double click and using the keyboard, or by selecting with button 1 click (after the class has been selected) and using the property tab.

SEQUENCE DIAGRAM SPECIFIC TOOLS

- ClassifierRole. Add a classifierrole to the diagram.
- Message with Call Action. Add a call message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- Message with Return Action. Add a return message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- Message with Create Action. Add a create message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
- Message with Destroy Action. Add a destroy message between two classifierroles selected using button 1 motion (from the originating classifierrole to the receiving classifierrole).
-

 Add Vertical Space to Diagram. Add vertical space to a diagram by moving all messages below this down. Click the mouse at the point where you want the space to be added and drag down the screen vertically the distance which matches the height of the space you'd like to have added.

-  Remove Vertical Space in Diagram. Remove vertical space from diagram and move all elements below up vertically. Click and drag the mouse vertically over the space that you want deleted.

COLLABORATION DIAGRAM SPECIFIC TOOLS

-  Classifier Role. Add a classifier role to the diagram.
- Association Role. Add an association role between two classifier roles selected using button 1 motion (from the originating classifier role to the receiving classifier role). There are 6 types of association roles offered here, see Figure 12.4, " The association tool selector. ": association, aggregation and composition, and all these three can be bidirectional or unidirectional.
-  Generalization. Add a generalization between two model elements selected using button 1 (from the child to the parent).
-  Dependency. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).
-  Add Message. Add a message to the selected association role.

STATE CHART DIAGRAM SPECIFIC TOOLS

-  Simple State. Add a simple state to the diagram.
-  Composite State. Add a composite state to the diagram.
-  Transition. Add a transition between two states selected using button 1 motion
-  Synch State. Add a synchstate to the diagram.
-  Submachine State. Add a submachinestate to the diagram.
-  Stub State. Add a stubstate to the diagram.
-  Initial. Add an initial pseudostate to the diagram.

- Shallow History. Add a shallow history pseudostate to the diagram.
- Deep History. Add a deep history pseudostate to the diagram.
- Call Event. Add a Call Event as trigger to a transition. There are 4 types of events offered here: Call Event, Change Event, Signal Event and Time Event.
- Guard. Add a guard to a transition.
- Call Action. Add a call action (i.e. the effect) to a transition.

ACTIVITY DIAGRAM SPECIFIC TOOLS

- Action State. Add an action state to the diagram.
- Transition. Add a transition between two action states selected using button 1 motion (from the originating action state to the receiving action state).
- Initial. Add an initial pseudostate to the diagram.
- Final State. Add a final state to the diagram.
- Junction. Add a junction (decision) pseudostate to the diagram.
- Fork. Add a fork pseudostate to the diagram.
- Join. Add a join pseudostate to the diagram.
- CallState. Add a callstate to the diagram. A call state is an action state that calls a single operation. Hence, the name of the operation being called is put in the symbol, along with the name of the classifier that hosts the operation in parentheses under it.
- ObjectFlowState. Add a objectflowstate to the diagram. An objectflowstate is an object that is input to or output from an action.

DEPLOYMENT DIAGRAM SPECIFIC TOOLS

-  `Node`. Add a node to the diagram. For convenience, when the mouse is over a selected node it displays four handles to left, right, top and bottom which may be dragged to form association relationships.

-  `Node Instance`. Add a node instance to the diagram. For convenience, when the mouse is over a selected node instance it displays four handles to left, right, top and bottom which may be dragged to form link relationships.

 `Component`. Add a component to the diagram. For convenience, when the mouse is over a selected component it displays four handles to left, right, top and bottom which may be dragged to form dependency relationships.

-  `Component Instance`. Add a component instance to the diagram. For convenience, when the mouse is over a selected component instance it displays four handles to left, right, top and bottom which may be dragged to form dependency relationships.

-  `Generalization`. Add a generalization between two model elements selected using button 1 (from the child to the parent).

-  `Realization`. Add a realization between a class and an interface selected using button 1 motion (from the realizing class to the realized interface).

-  `Dependency`. Add a dependency between two model elements selected using button 1 motion (from the dependent model element).

- `Association`. Add an association between two model elements

Exp. No:1 Date:	PASSPORT AUTOMATION SYSTEM	

Aim

To Design and Implement the project of Passport Automation System.

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 online registration form (with details such as name, address etc.,) filled by the applicant whose testament is verified for its genuineness by the Passport Automation System with respect to the already existing information in the database. This forms the first and foremost step in the processing of passport application. After the first round of verification done by the system, the information is in turn forwarded to the regional administrator's (Ministry of External Affairs) office. The application is then processed manually based on the report given by the system, and any forfeiting identified can make the applicant liable to penalty as per the law. The system also provides the applicant the list of available dates for appointment to 'document verification' in the administrator's office, from which they can select one. The system forwards the necessary details to the police for its separate verification whose report is then presented to the administrator. 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.

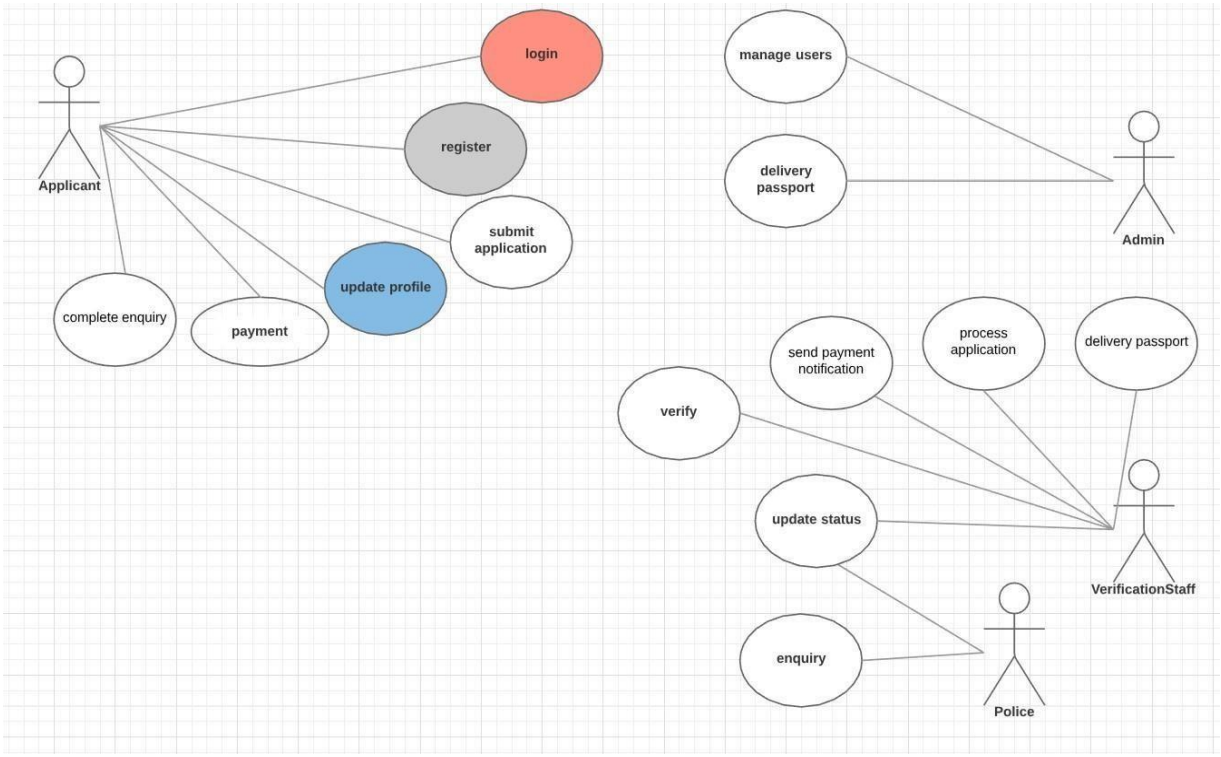
Passport Automation System is an interface between the Applicant and the Authority responsible for the Issue of Passport. It aims at improving the efficiency in the Issue of Passport and reduces the complexities involved in it to the maximum possible extent.

If the entire process of 'Issue of Passport' is done in a manual manner then it would takes 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. So this system uses several programming and database techniques to elucidate the work involved in this process. As this is a matter of National Security, the system has been carefully verified and validated in order to satisfy it.

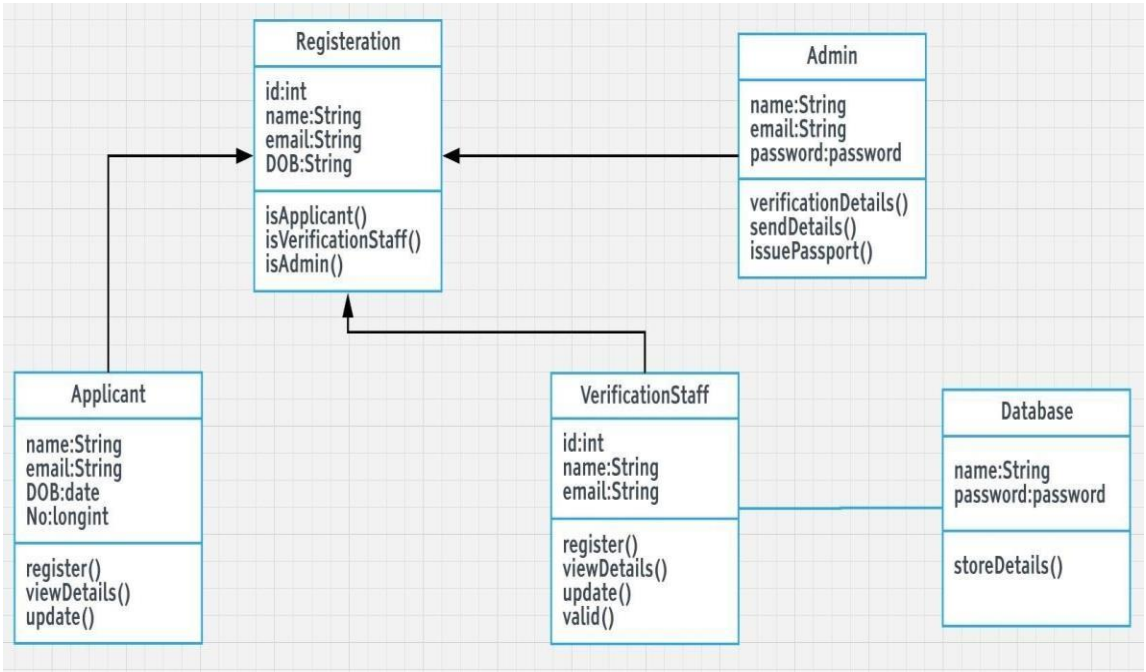
The System provides an online interface to the user where they can fill in their personal details and submit the necessary documents (may be by scanning).

UML Diagrams

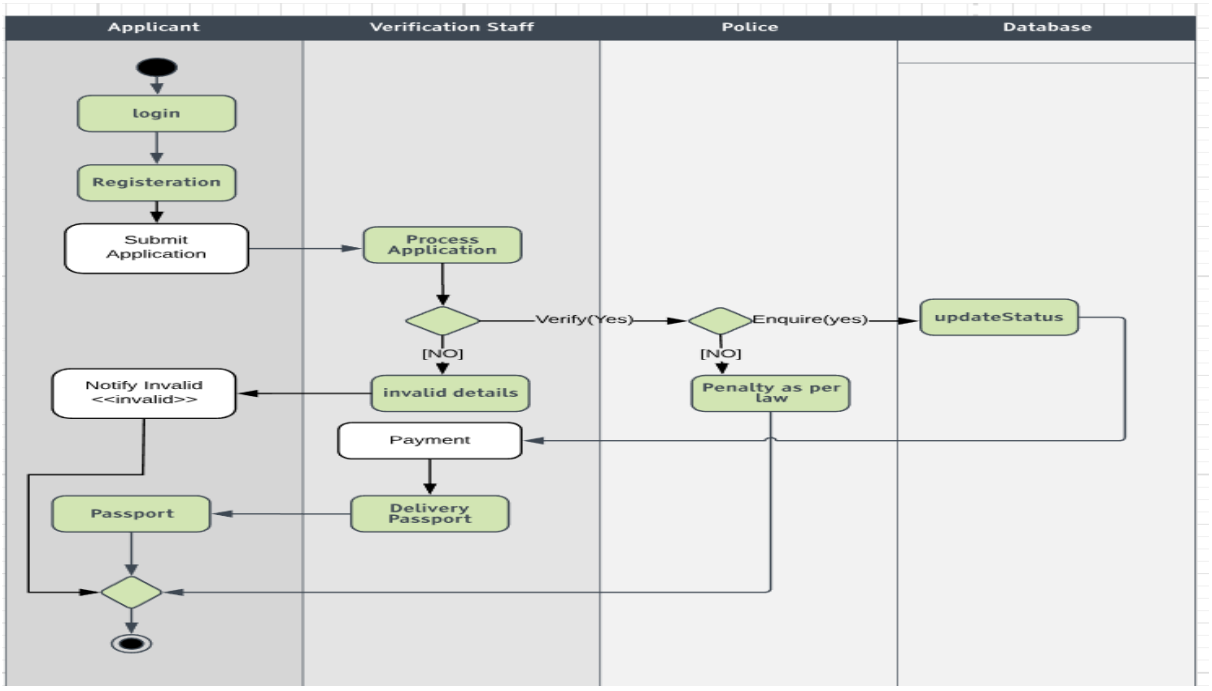
USE CASE DIAGRAM



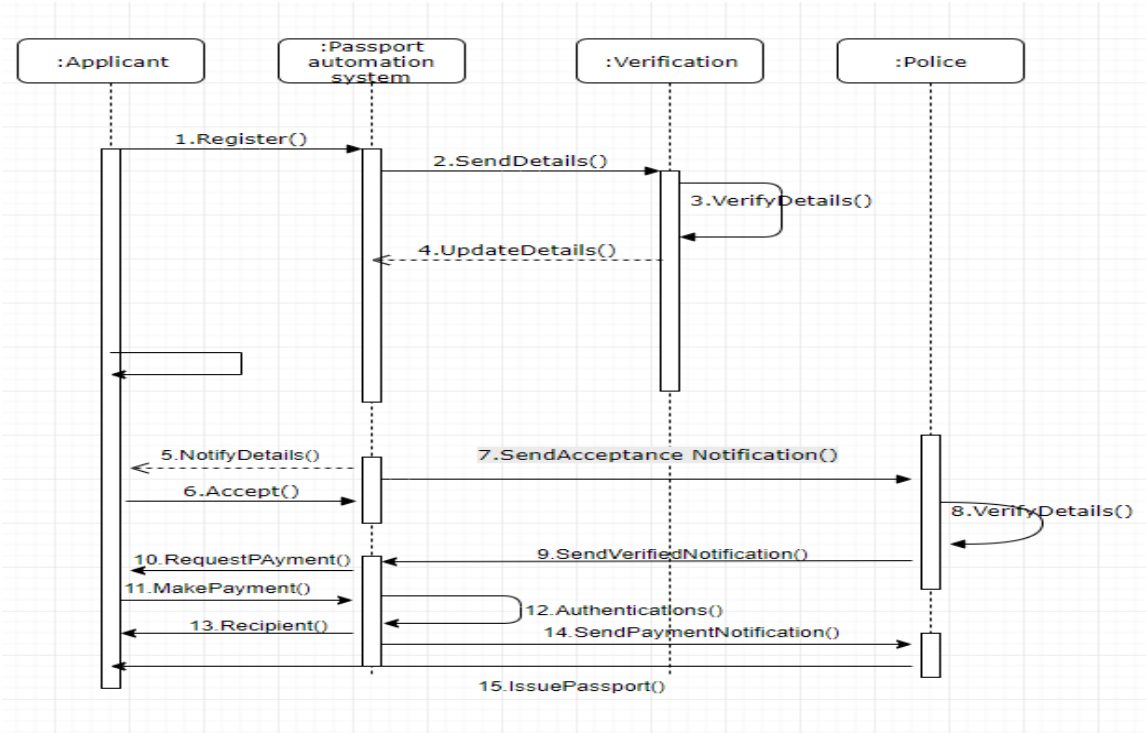
CLASS DIAGRAM



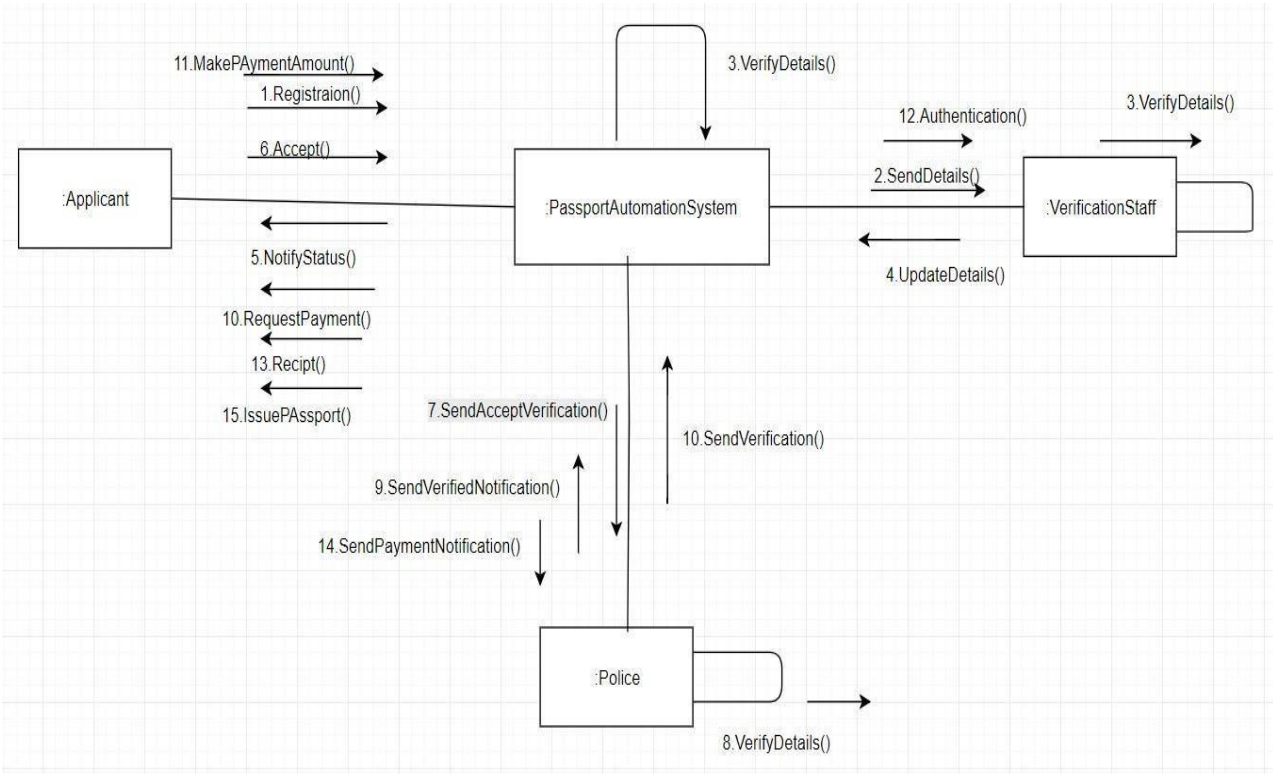
ACTIVITY DIAGRAM



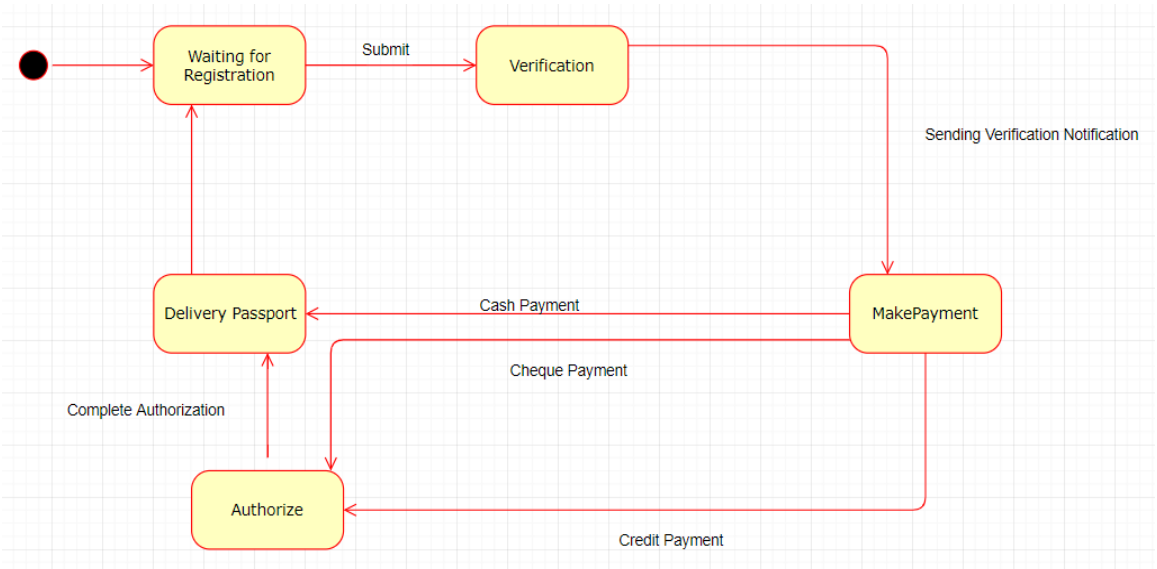
SEQUENCE DIAGRAM



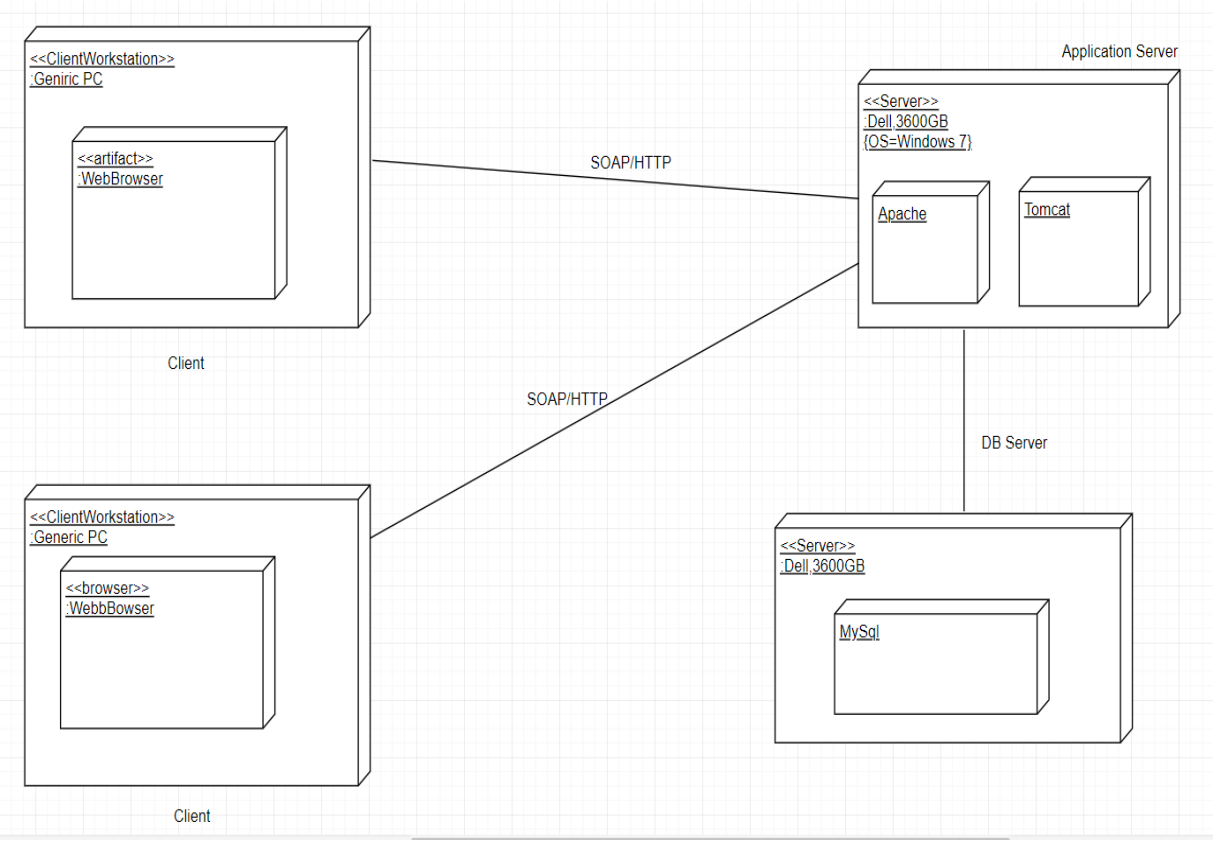
COLLABORATION DIAGRAM



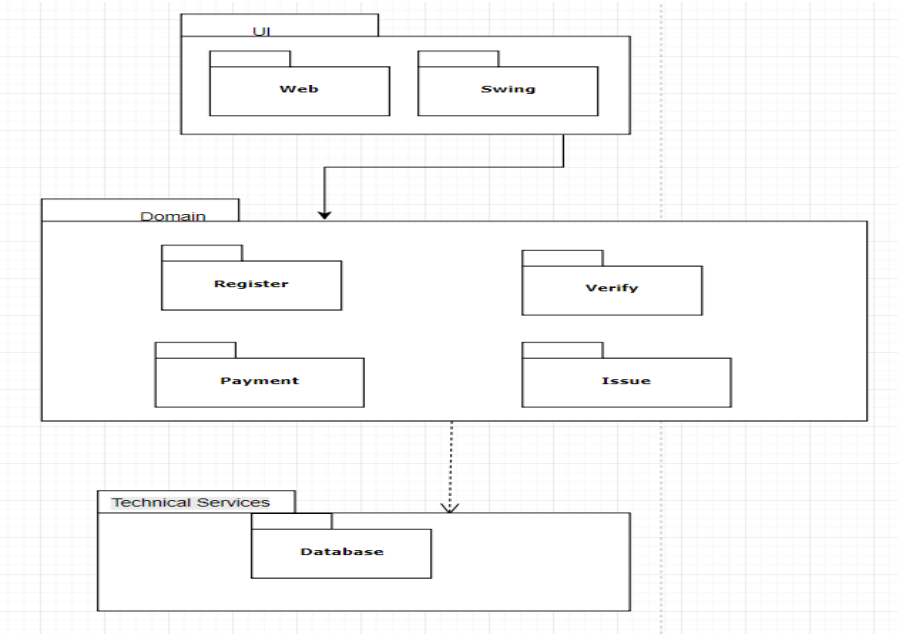
STATE DIAGRAM



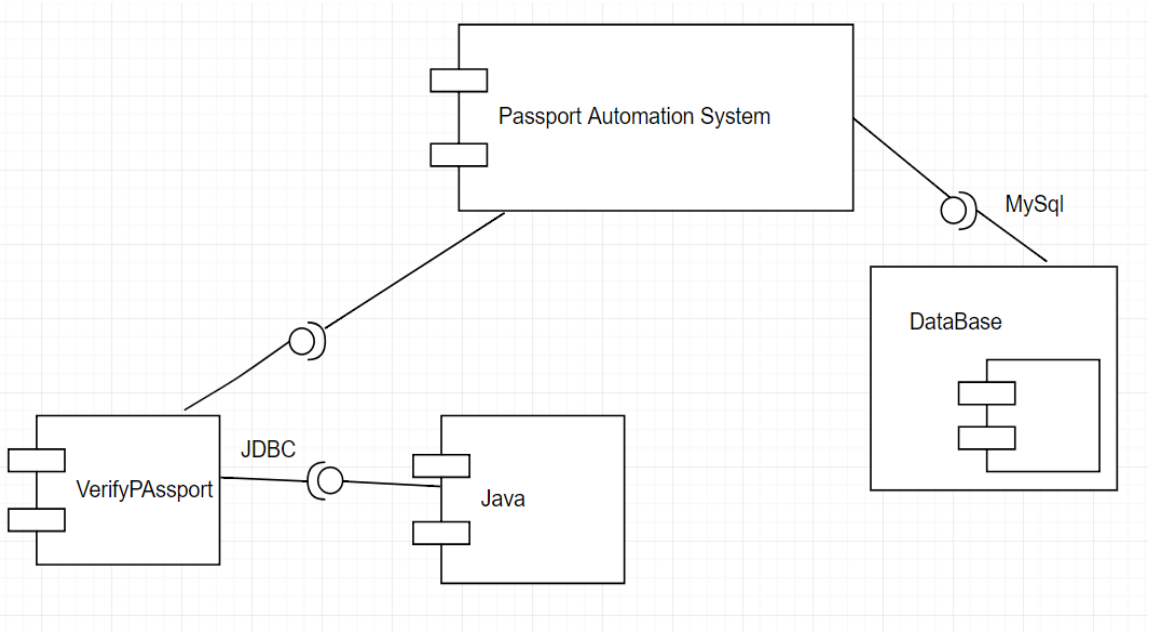
DEPLOYMENT DIAGRAM



PACKAGE DIAGRAM



COMPONENT DIAGRAM



Result

Thus the design and Implementation of passport automation system was completed and verified successfully.

Exp. No:2 Date:	BOOK BANK MANAGEMENT SYSTEM	

Aim

To design and implement the object oriented model of Book Bank Management System.

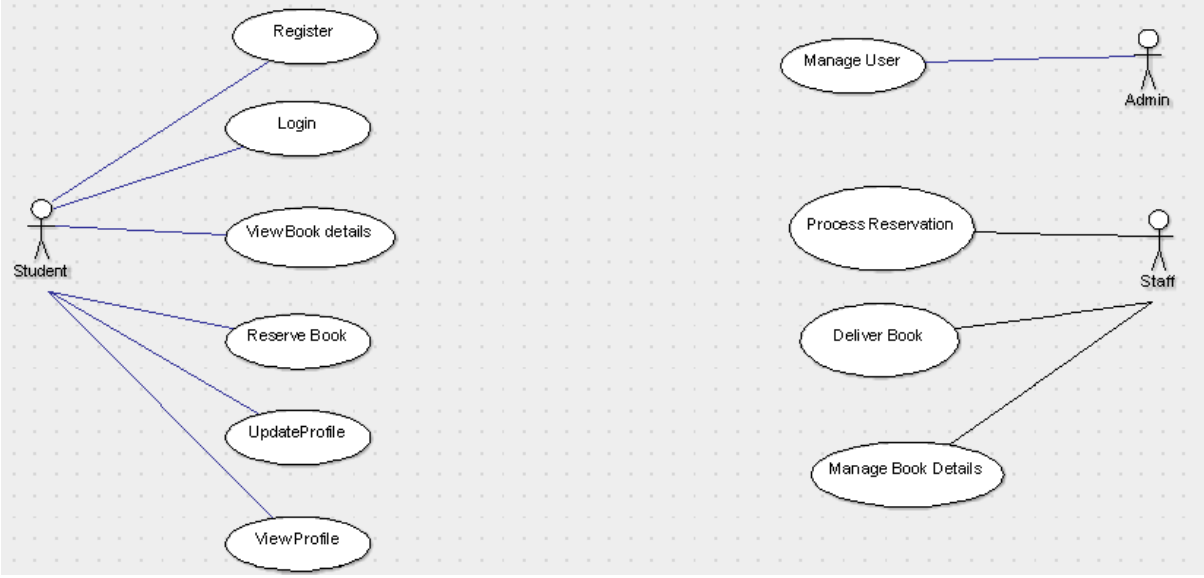
Problem Statement

Book Bank is used by any student, who wishes to borrow the books. The student must first register himself to the system and then can login. The student can perform look ups for getting the desired book. If the book is available then he can borrow the book. The admin issues the book to the student. The DBA maintains the book. The admin update the details to the database. Allows a student, who becomes a member to login using a unique id issued at the time of registering as a member, and after logging in, the member can browse through available books and make requests accordingly. The books will be issued provided there is no due, regarding returning of previous books.

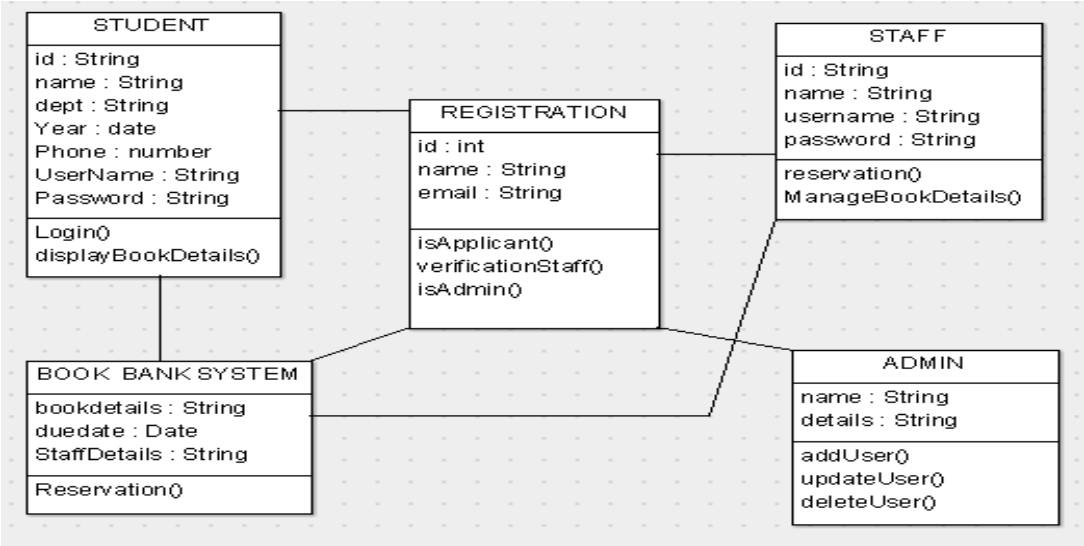
Whenever the student wishes to get books, he/she checks for the availability by logging in. When the request is made, the director of the book bank decides on granting the request of book(s) after checking the member details for due in returning previous books. The member should be authenticated by means of unique login id and password. The availability of books requested must be prompted to the user through e-mail or sms notifications. The web interface should be able to support multiple users trying to log in simultaneously. The student details should be made available in the database and must be updated every time a book is issued or returned or some kind of payment takes place to prevent errors. The member can only access certain details from the database. He/she should not be able to modify the database nor has any of its information corrupted. Only the DBA must be bestowed with the privileges of handling any kind of modifications.

UML Diagrams

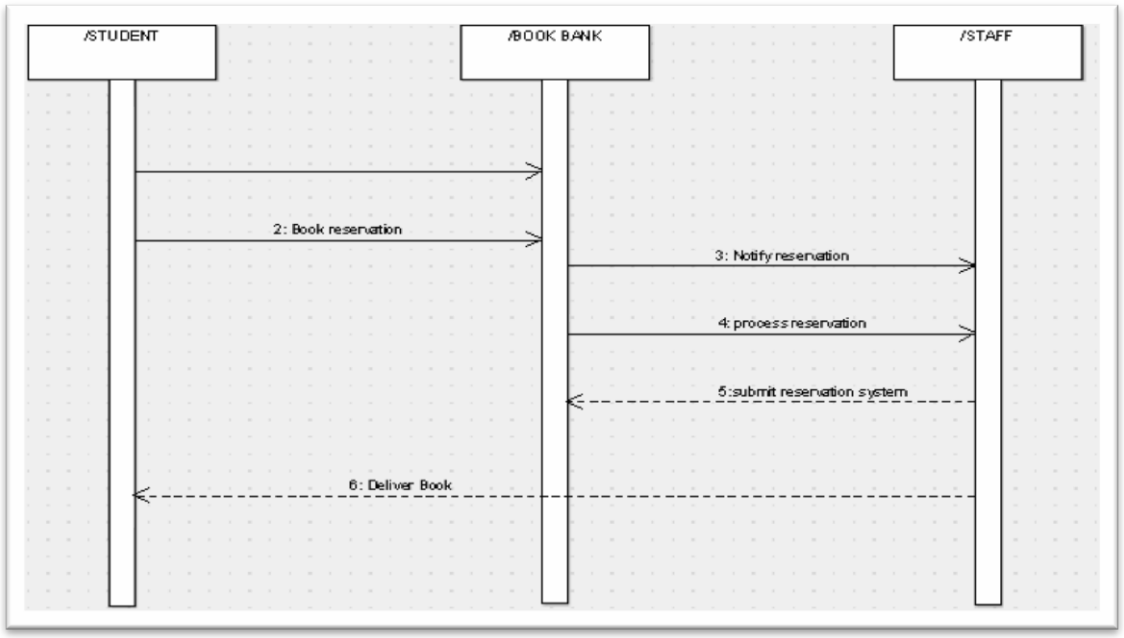
USE CASE DIAGRAM



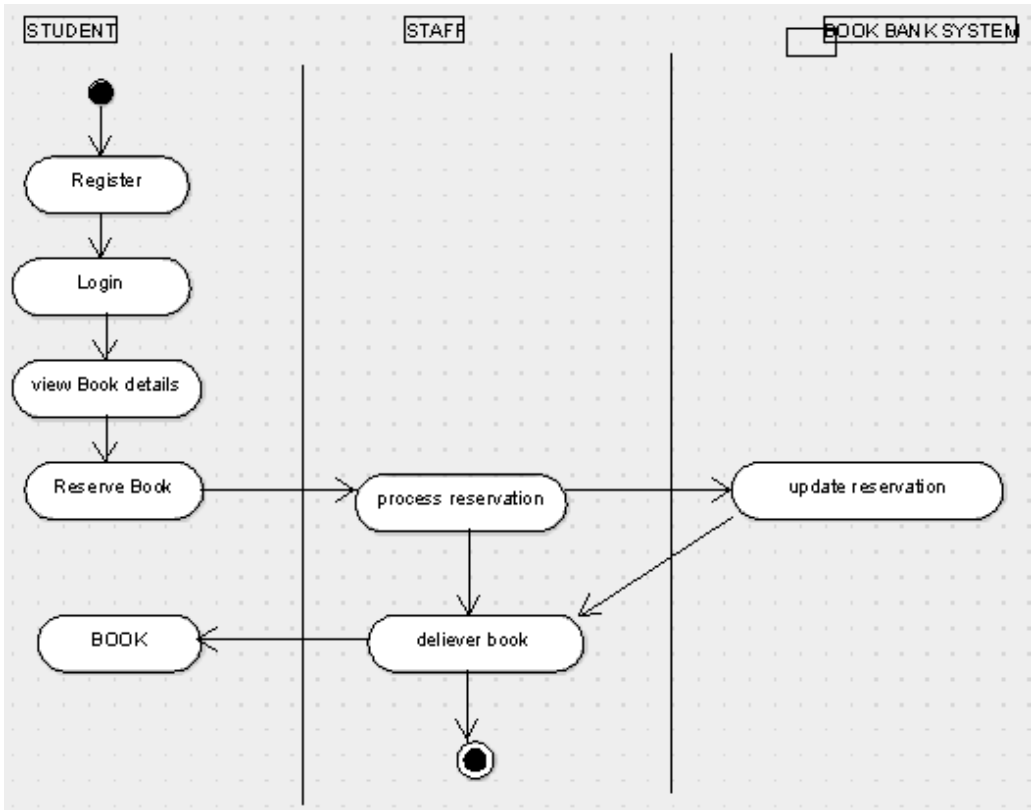
CLASS DIAGRAM



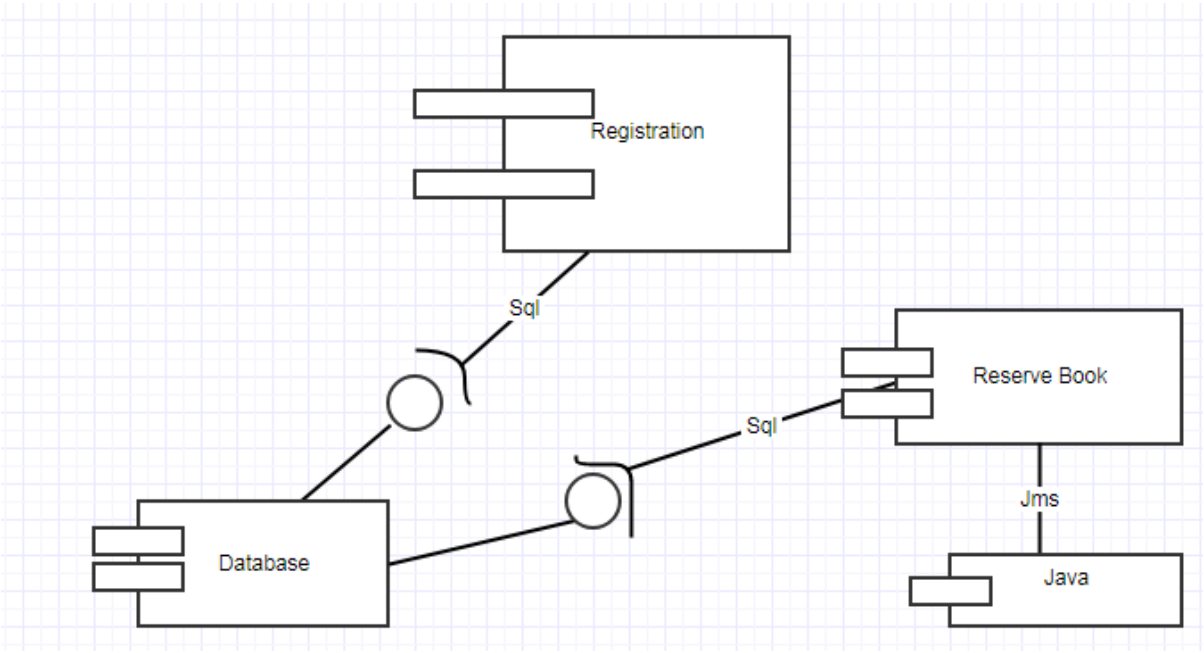
SEQUENCE DIAGRAM



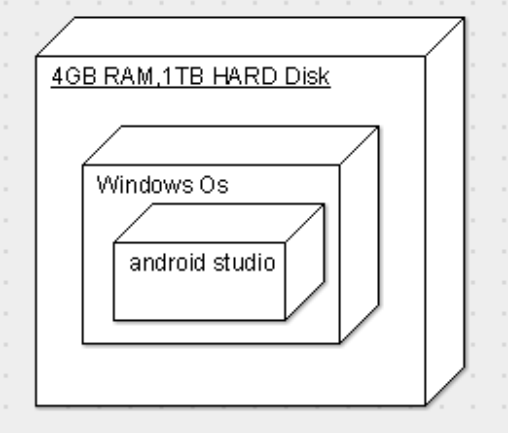
ACTIVITY DIAGRAM



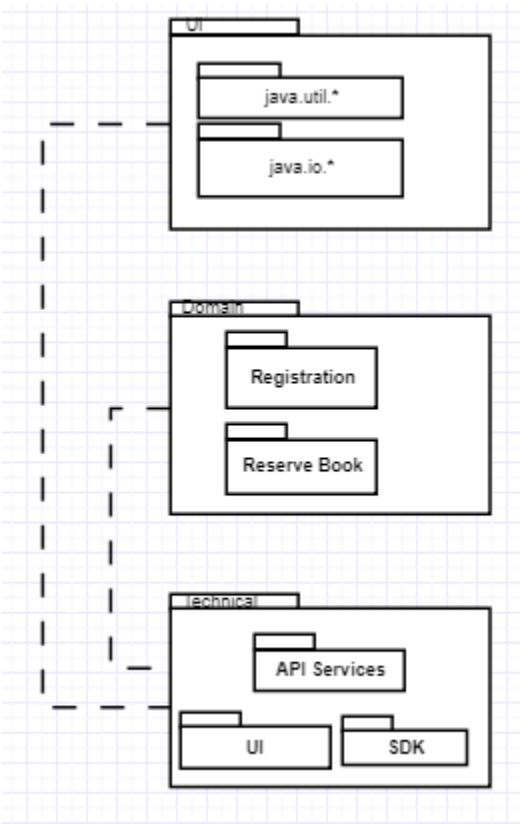
COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM



PACKAGE DIAGRAM



Result

Thus design and implenetation of the project for book bank system was completed and verified succesfully.

Ex no:3 **EXAM REGISTRATION SYSTEM**

Date:

AIM: To create a system to perform the Exam Registration system

(I) PROBLEM STATEMENT:

Exam Registration system.is used in the effective dispatch of registration form to all of the students. 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 online registration form (with details such as name, reg.no etc.,) filled by the student whose testament is verified for its genuineness by the Exam Registration System with respect to the already existing information in the database.

(II) USECASE DIAGRAM:

The Exam Registration use cases in our system are:

1. Login
2. View exam details
3. View fees details
4. Pay fee
5. Display details
6. Logout

USECASE DIAGRAM :

Fig. 3.USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(I) ACTIVITY DIAGRAM:

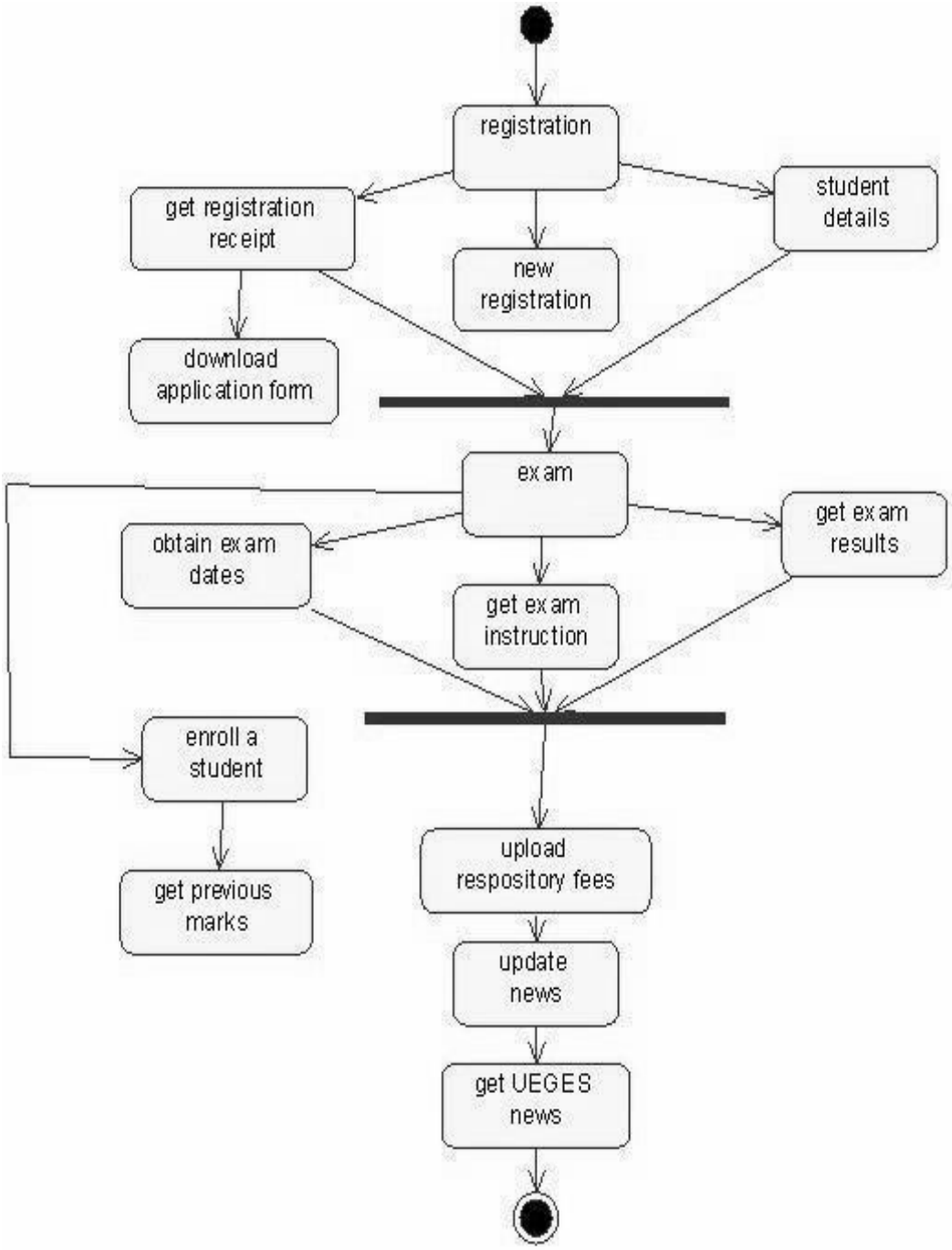


Fig. 4.USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(II)CLASS DIAGRAM:

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

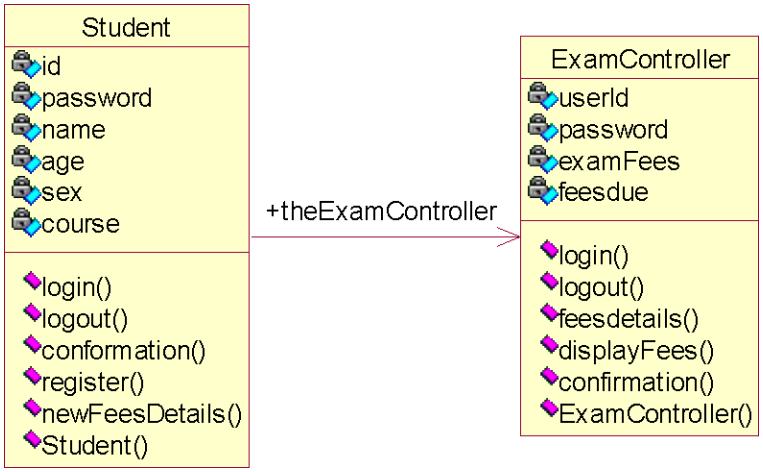


Fig.5. CLASS DIAGRAM FOR EXAM REGISTRATION SYSTEM

(III)INTERACTION DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another

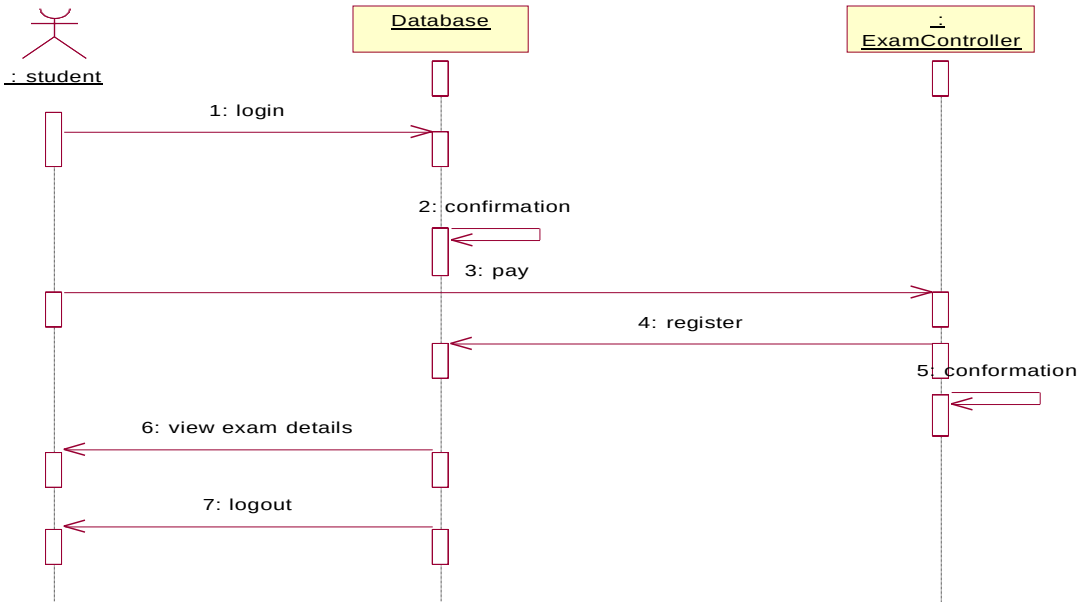


Fig. 6.1. SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

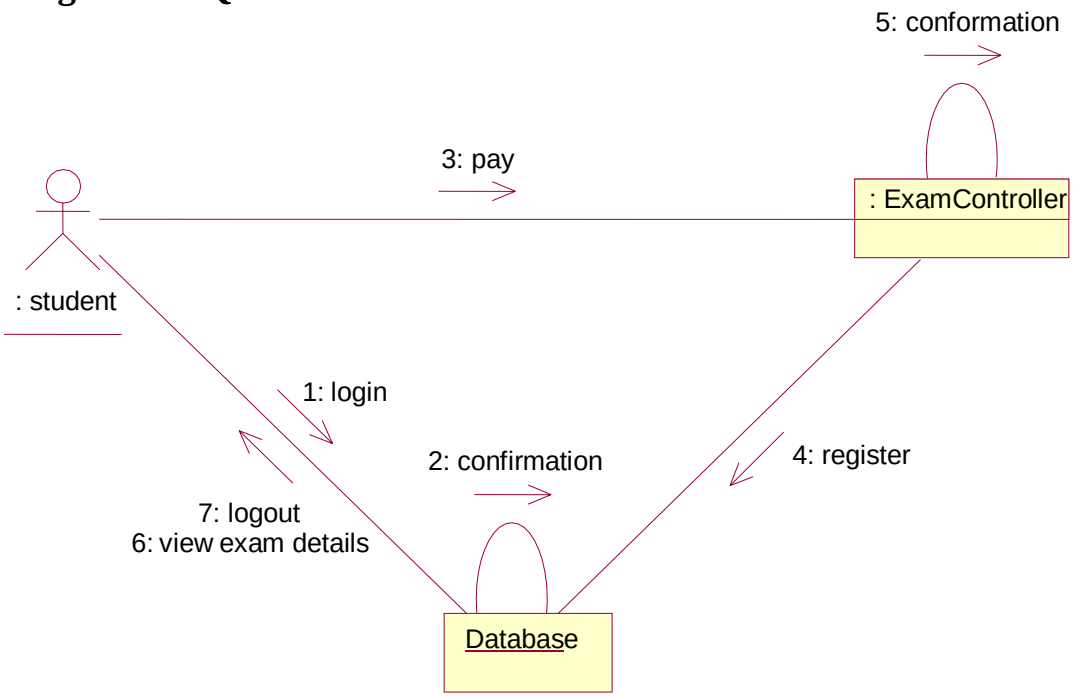
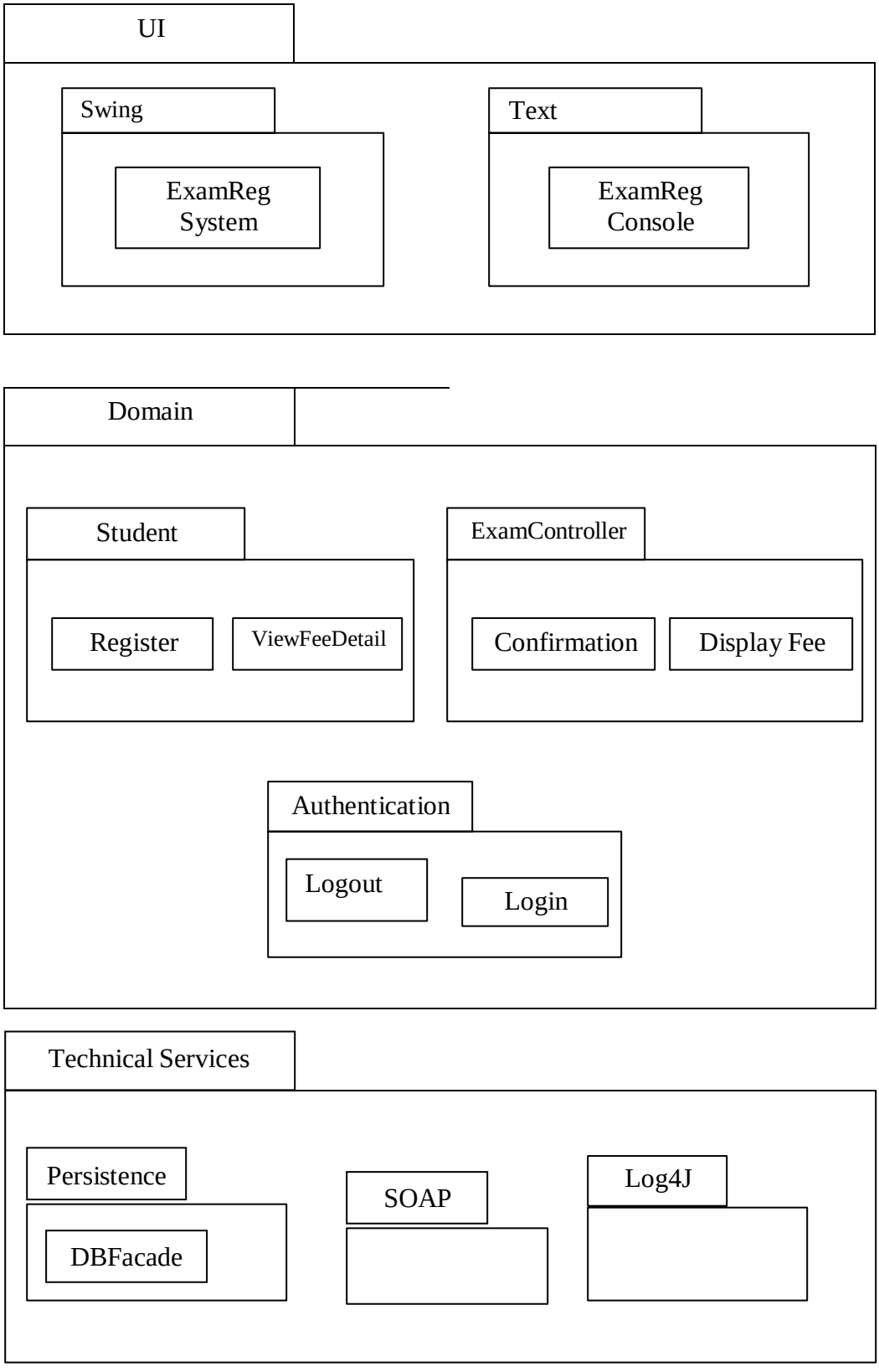


Fig. 6.2. COLLABORATION DIAGRAM FOR REGISTRATION SYSTEM

(IV) PARTIAL LAYERED LOGICAL ARCHITECTURE DIAGRAM:



(V) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

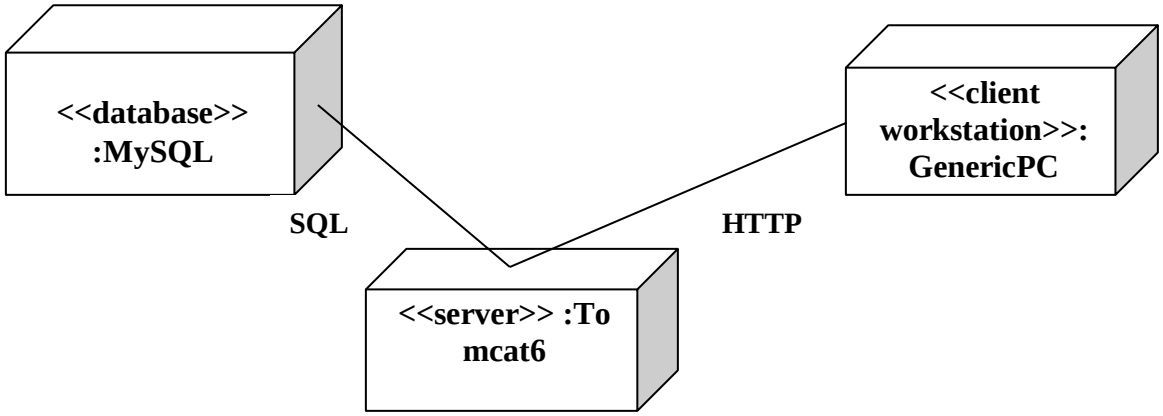


Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.

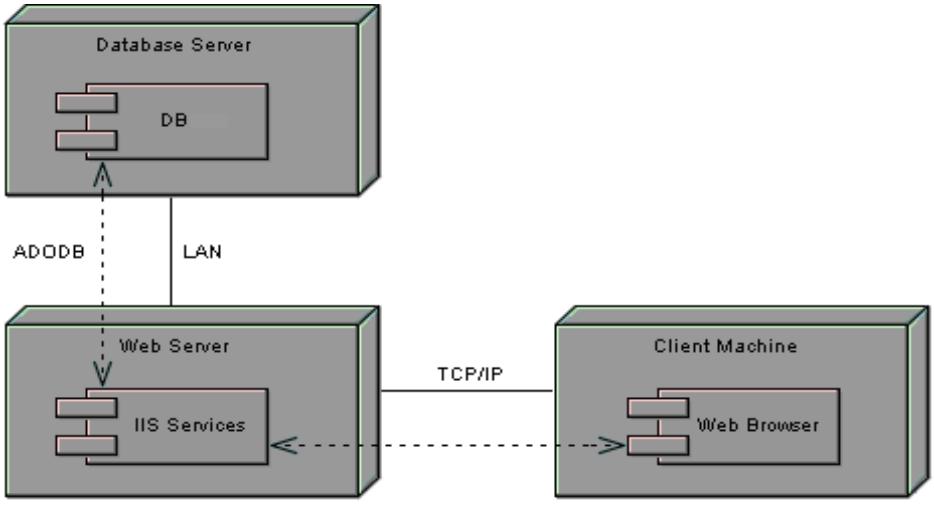


Fig.7.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for Exam Registration system has been successfully executed and codes are generated.

Ex no:4

STOCK MAINTENANCE

Date:

AIM:

To create a system to perform the Stock maintenance

(I) PROBLEM STATEMENT

The stock maintenance system must take care of sales information of the company and must analyze the potential of the trade. It maintains the number of items that are added or removed.The sales person initiates this Use case. The sales person is allowed to update information and view the database.

(II) SOFTWARE REQUIREMENT SPECIFICATION

PURPOSE

The entire process of Stock maintenanceis done in a manual manner Considering the fact that the number of customers for purchase is increasing every year, a maintenance system is essential to meet the demand. So this system uses several programming and database techniques to elucidate the work involved in this process.

SCOPE

• The System provides an interface to the customer where they can fill in orders for the item needed.

• The sales person is concerned with the issue of items and can use this system.

• Provide a communication platform between the customer and the sales person.

(III) USE CASE DIAGRAM

The functionality of a system can be described in a number of different use-cases, each of which represents a specific flow of events in a system. It is a graph of actors, a set of use-cases enclosed in a boundary, communication, associations between the actors and the use-cases, and generalizationamong the use-cases

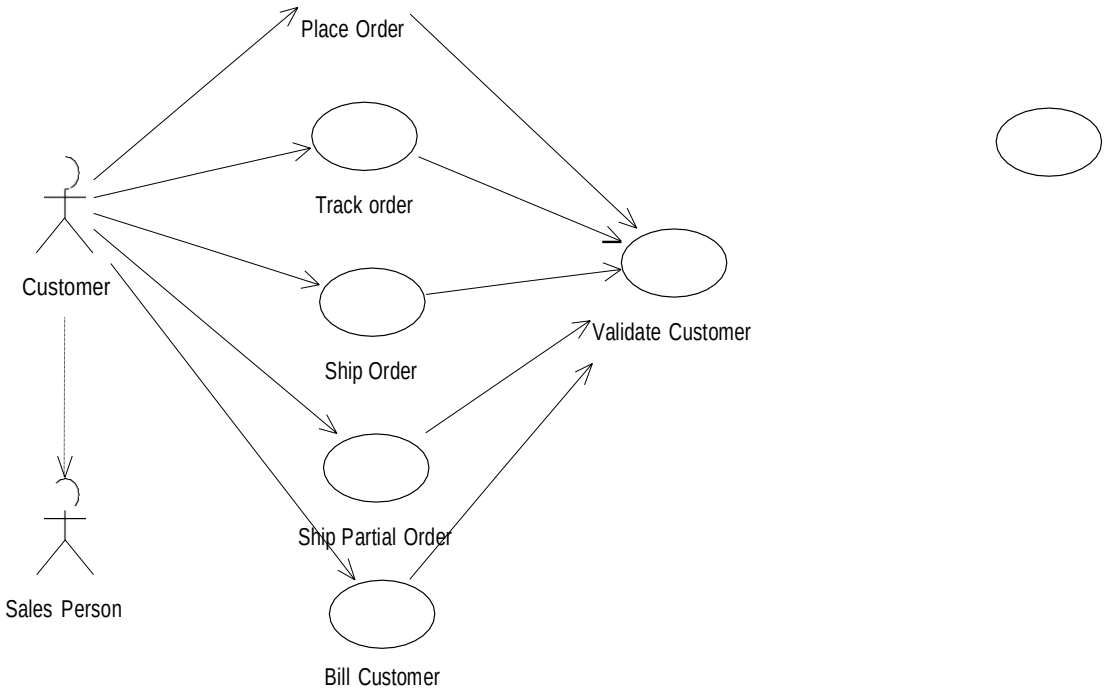


Fig.3. USE CASE DIAGRAM

(IV) ACTIVITY DIAGRAM

It shows organization and their dependence among the set of components. These diagrams are particularly useful in connection with workflow and in describing behavior that has a lot of parallel processing. An activity is a state of doing something: either a real-world process, or the execution of a software routine.

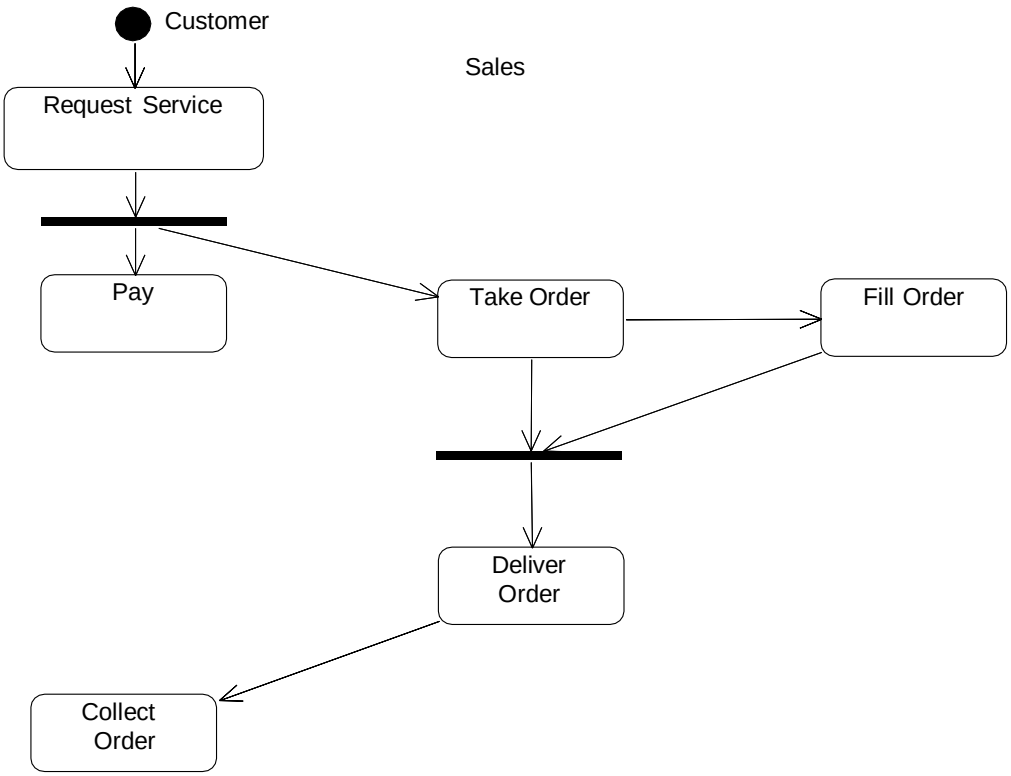


Fig.4. ACTIVITY DIAGRAM

(V) CLASS DIAGRAM

Description:

- A class diagram describes the type of objects in system and various kinds of relationships that exists among them.
- Class diagrams and collaboration diagrams are alternate representations of object models.

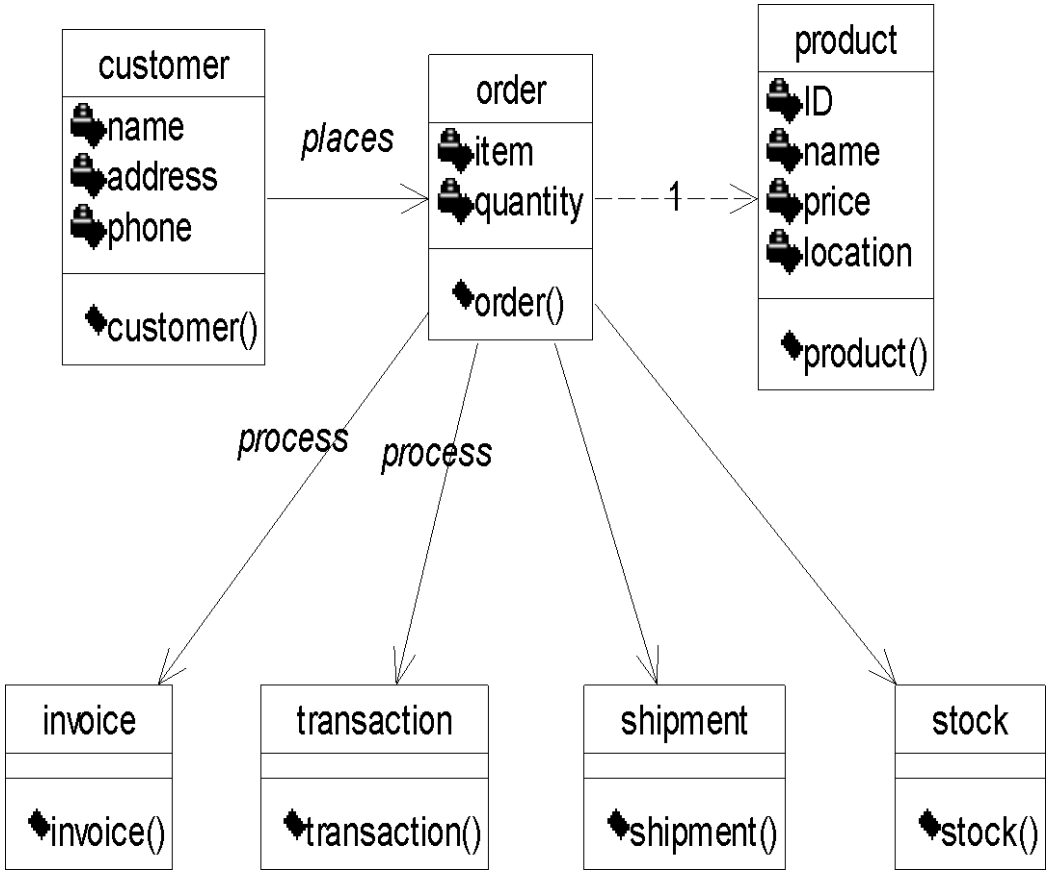


Fig.5. CLASS DIAGRAM

(VI) UML INTERACTION DIAGRAMS

It is the combination of sequence and collaboration diagram. It is used to depict the flow of events in the system over a timeline. The interaction diagram is a dynamic model which shows how the system behaves during dynamic execution.

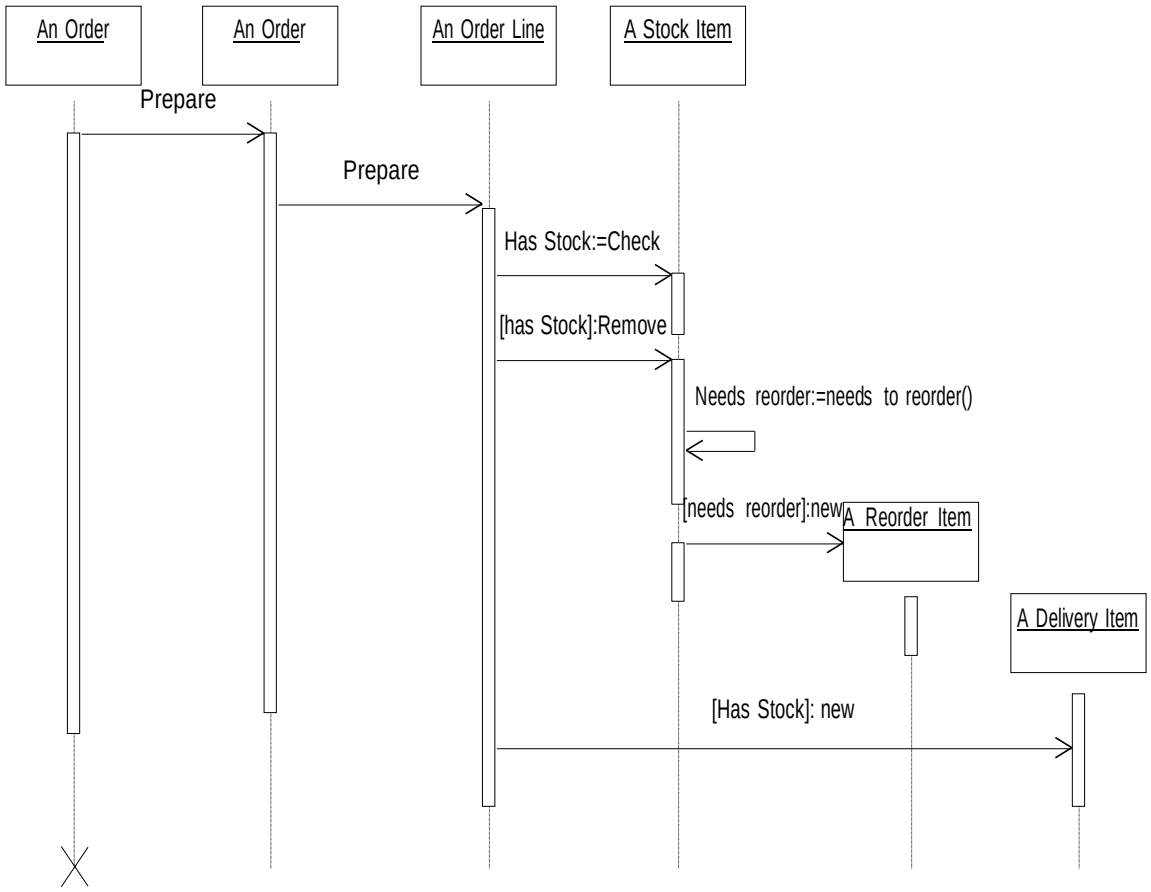


Fig.6.1 SEQUENCE DIAGRAM

COLLABORATION DIAGRAM

Collaboration diagram and sequence diagrams are alternate representations of an interaction. A collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction.

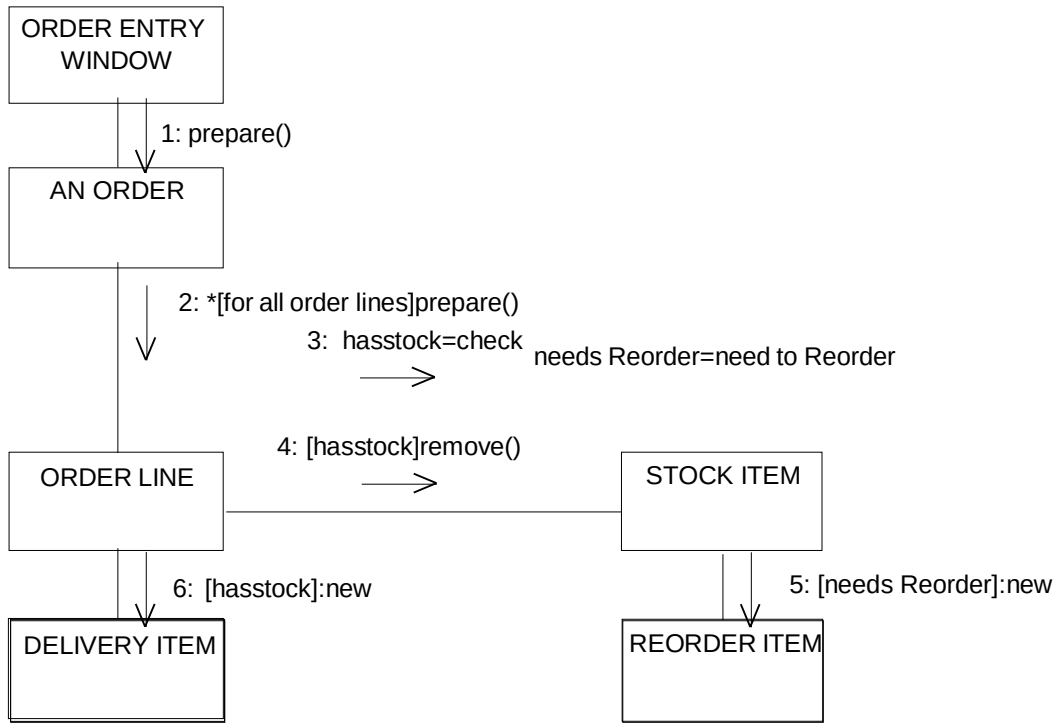
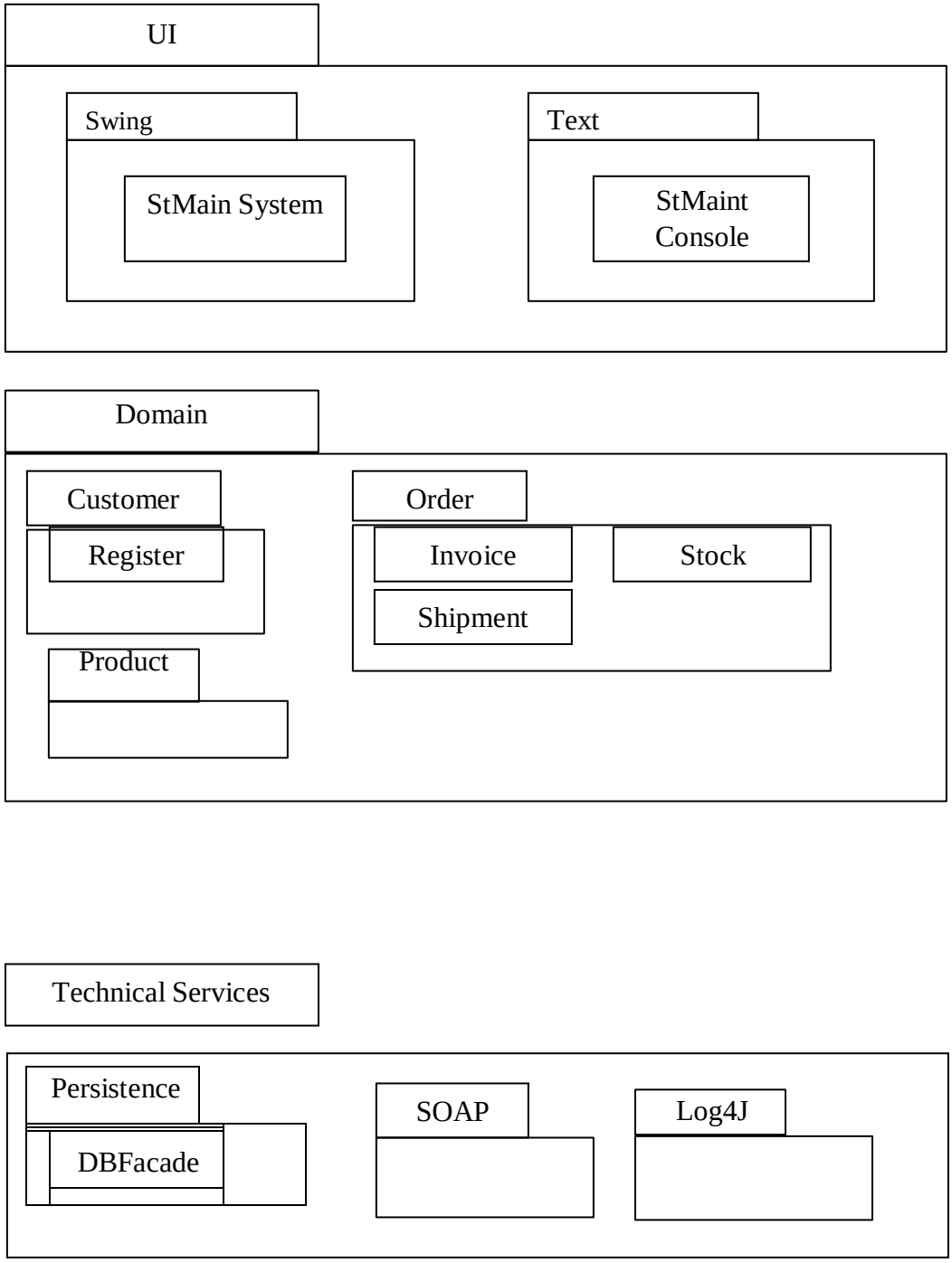


Fig.6.2 COLLABORATION DIAGRAM

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

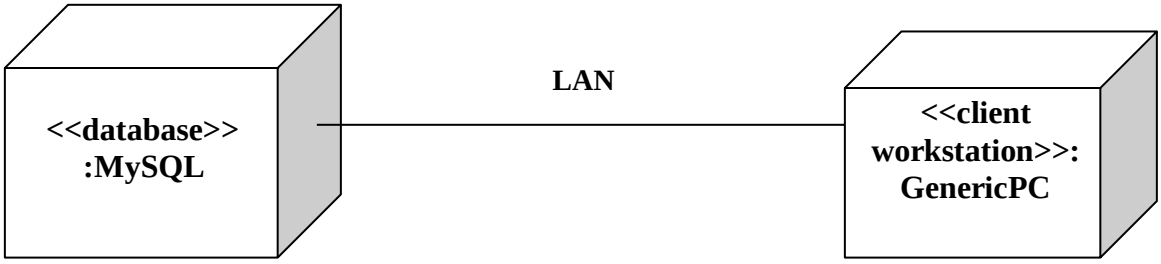


Fig.8.1.DEPLOYMENT DIAGRAM

Component Diagram

Component diagrams are used to visualize the organization and relationships among components in a system.

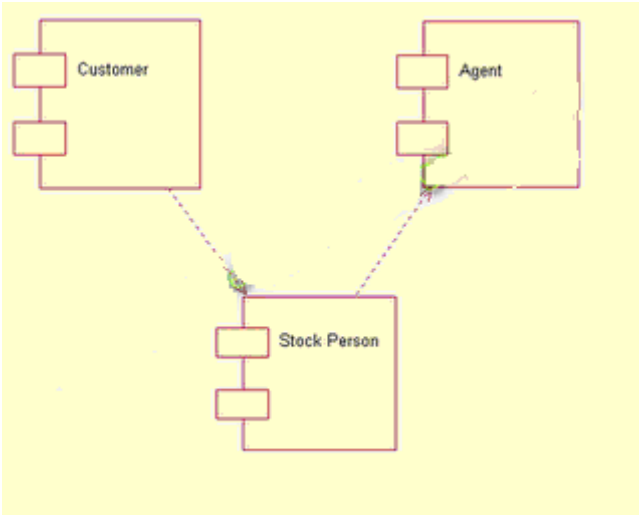


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for stock maintenance system has been successfully executed and codes are generated.

EX NO: 5 ONLINE COURSE RESERVATION SYSTEM

Date:

AIM

To design an object oriented model for course reservation system.

(I) PROBLEM STATEMENT

a. Whenever the student comes to join the course he/she should be provided with the list of course available in the college.

b. The system should maintain a list of professor who is teaching the course. At the end of the course the student must be provided with the certificate for the completion of the course.

(II) SYSTEM REQUIEMENT SPECIFICATION

OBJECTIVES

a. The main purpose of creating the document about the software is to know about the list of the requirement in the software project part of the project to be developed.

b. It specifies the requirement to develop a processing software part that completes the set of requirement.

SCOPE

a. In this specification, we define about the system requirements that are about from the functionality of the system.

b. It tells the users about the reliability defined in usecase specification

FUNCTIONALITY

Many members of the process line to check for its occurrences and transaction, we are have to carry over at sometimes

USABILITY

The user interface to make the transaction should be effectively

PERFORMANCE

It is the capability about which it can performed function for many user at sometimes efficiently (ie) without any ever occurrences

RELIABILITY

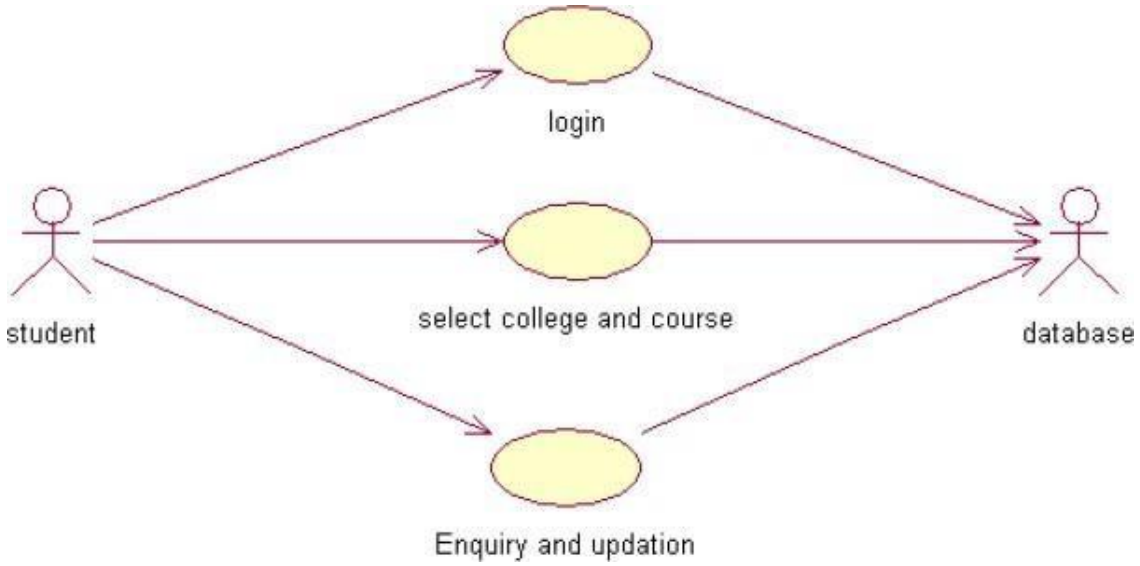
The system should be able to the user through the day to day transaction

(III) USERCASE DIAGRAM

a. Use case is a sequence of transaction in a system whose task is to yield result of measurable value to individual author of the system

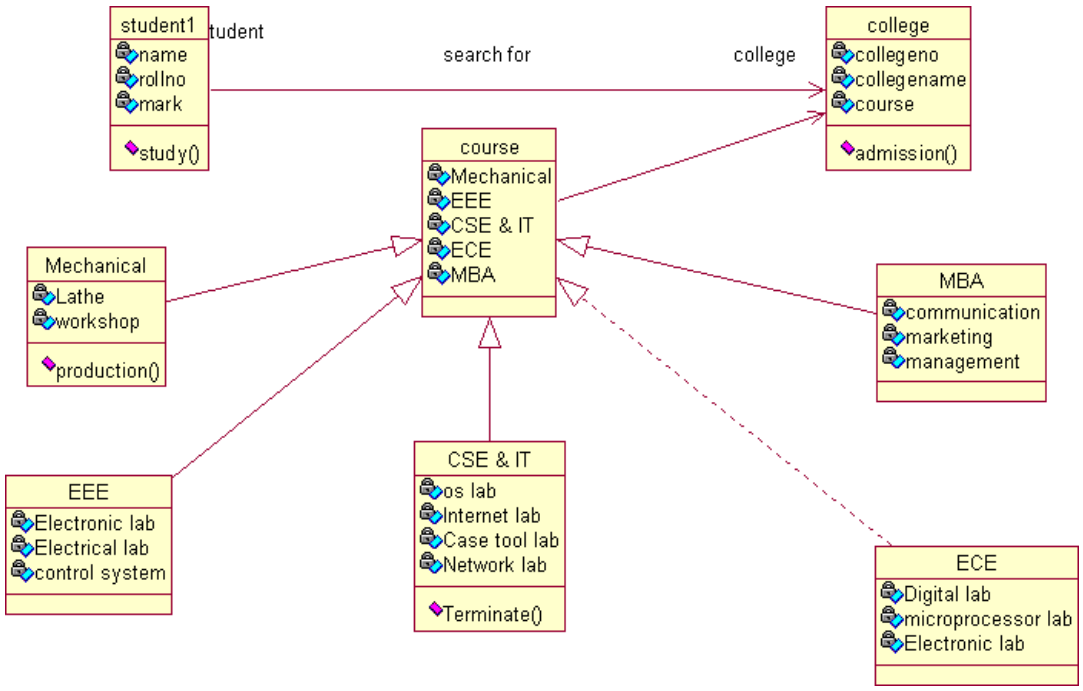
b. Use case is a set of scenarios together by a common user goal

c. A scenario is a sequence of step describing as interaction between a user and a system



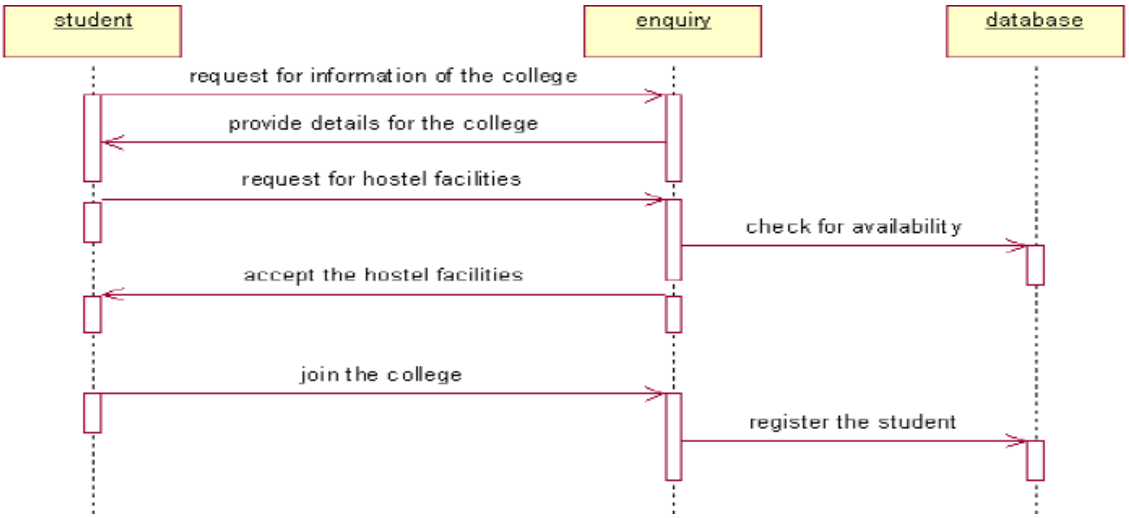
CLASS DIAGRAM:

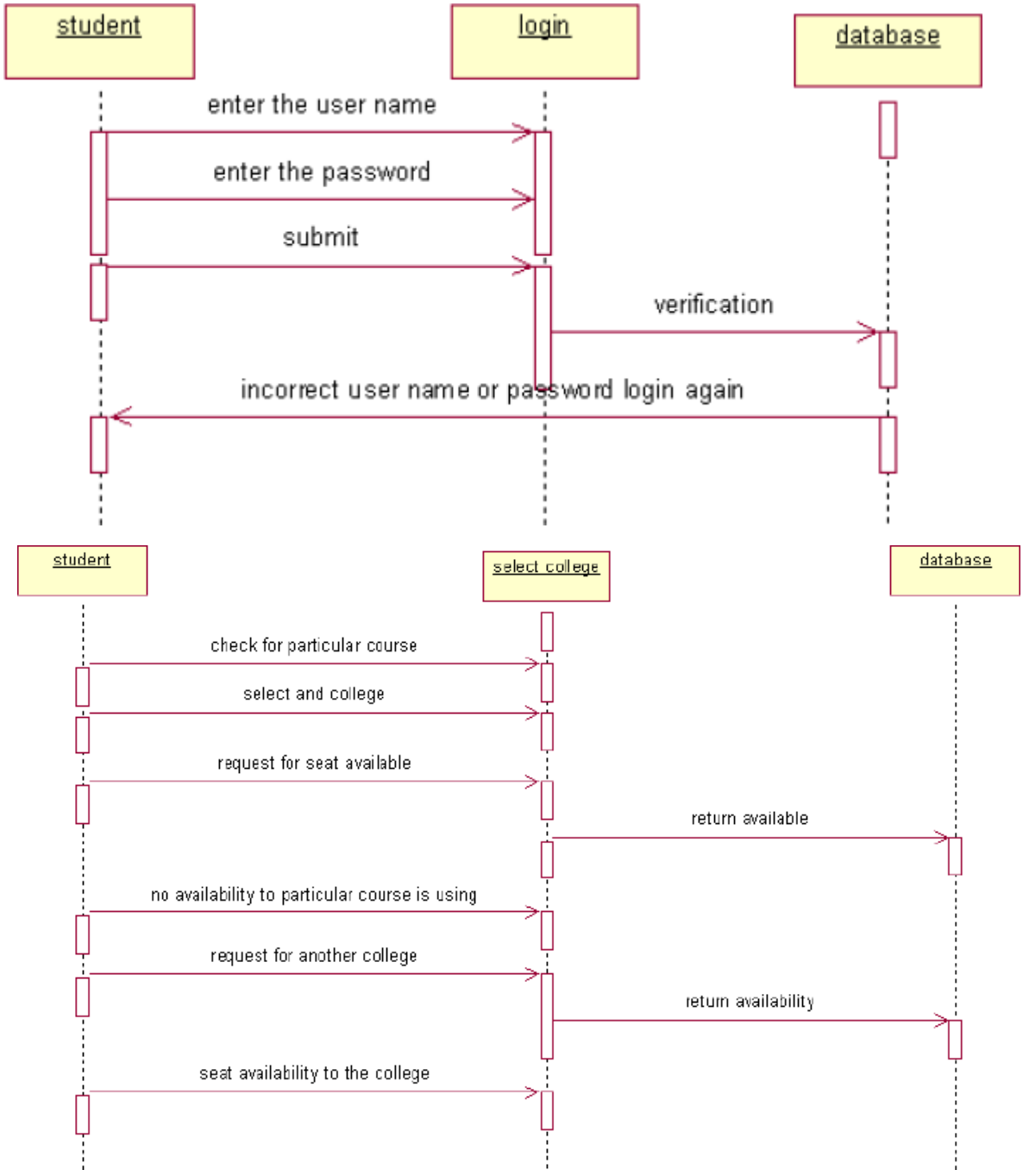
A class diagram describes the type of objectors in the system the various kinds of static relationship that exist among them.



SEQUENCE DIAGRAM

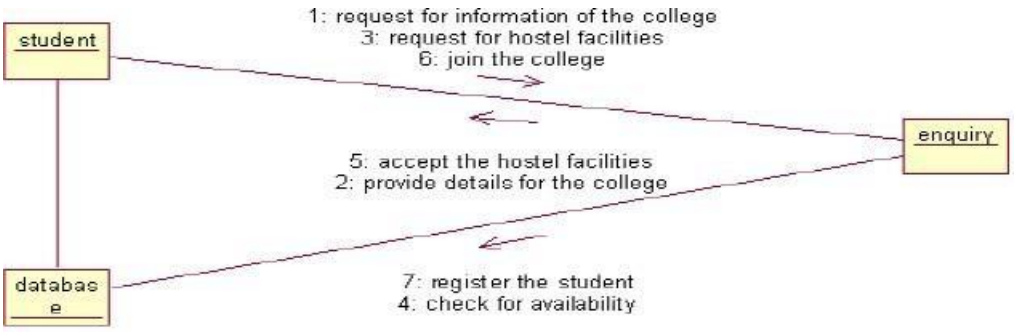
A sequence diagram is one that includes the object of the projects and tells the lifetimes and also various action performed between objects.

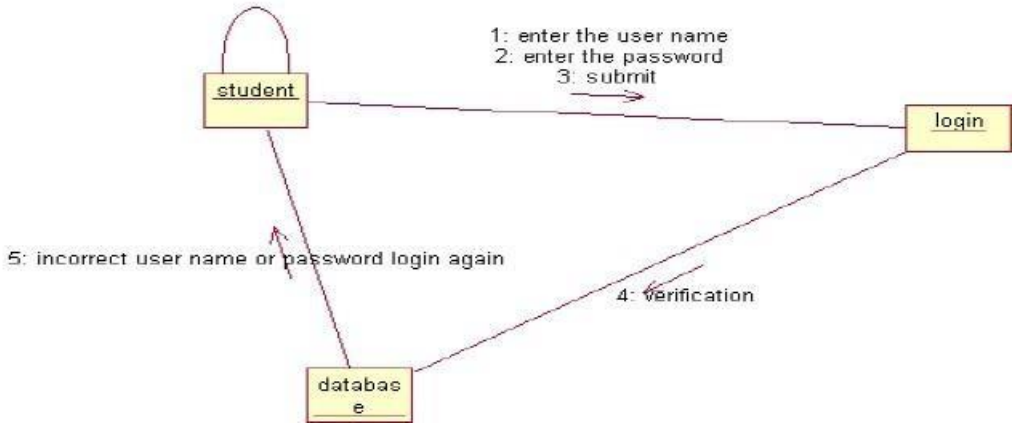


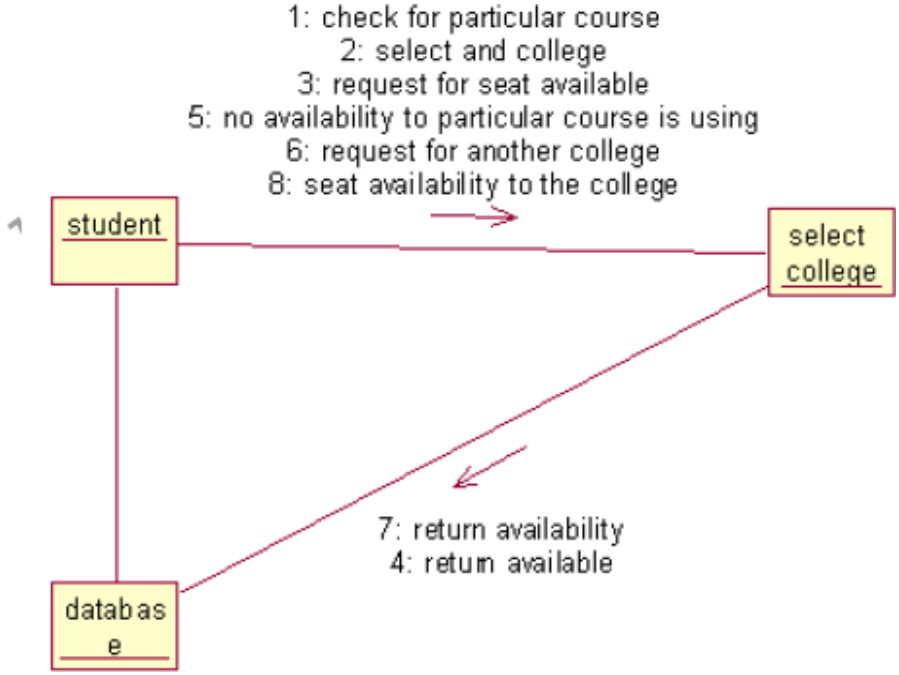


COLLOBORATIION DIAGRAM

It is same as the sequence diagram that involved the project with the only difference that we give the project with the only difference that we give sequence number to each process.

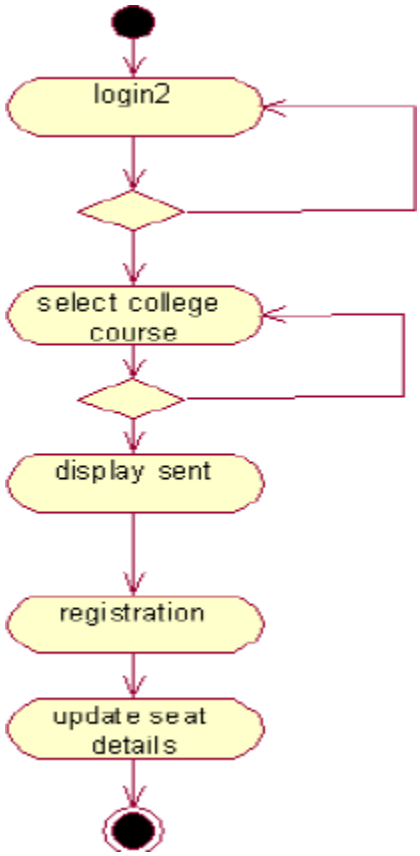

student
1: request for information of the college
3: request for hostel facilities
6: join the college
enquiry
5: accept the hostel facilities
2: provide details for the college
databas
e
7: register the student
4: check for availability


student
1: enter the user name
2: enter the password
3: submit
login
5: incorrect user name or password login again
4: verification
databas
e


1: check for particular course
2: select and college
3: request for seat available
5: no availability to particular course is using
6: request for another college
8: seat availability to the college
student
select
college
7: return availability
4: return available
databas
e

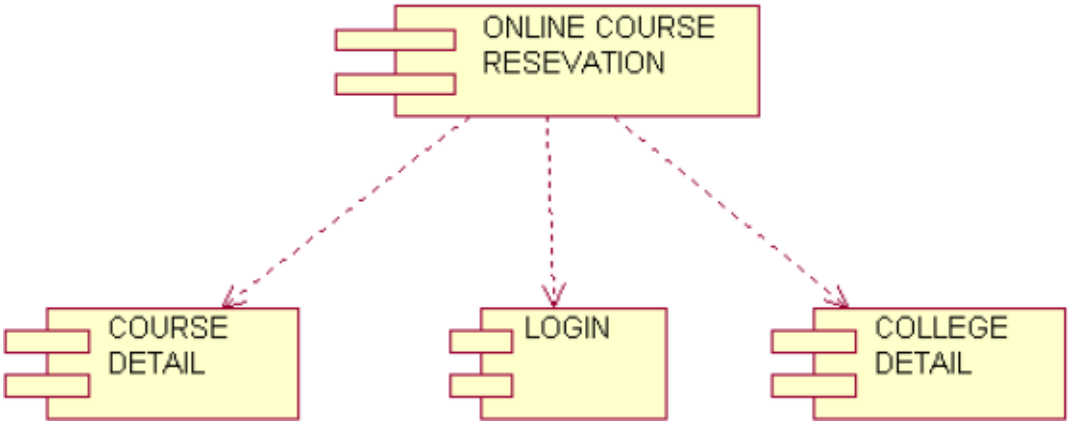
ACTIVIY DIAGRAM

It includes all the activities of particular project and various steps using join and forks



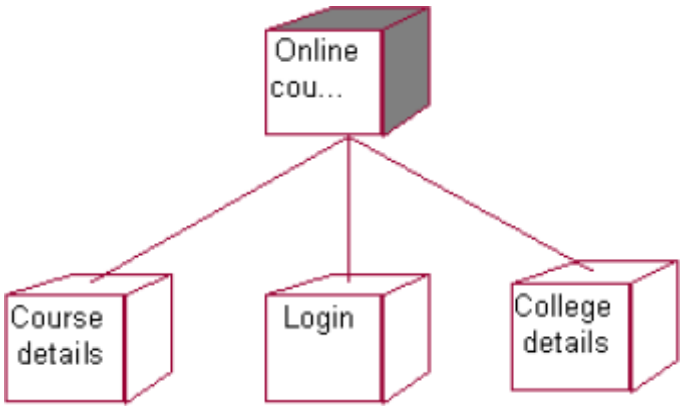
COMPONENT DIAGRAM

The component diagram is represented by figure dependency and it is a graph of design of figure dependency. The component diagram's main purpose is to show the structural relationships between the components of a systems. It is represented by boxed figure. Dependencies are represented by communication association



DEPLOYMENT DIAGRAM

It is a graph of nodes connected by communication association. It is represented by a three dimensional box. A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimentional box. Dependencies are represented by communication association. The basic element of a deployment diagram is a node of two types



PACKAGE DIAGRAM

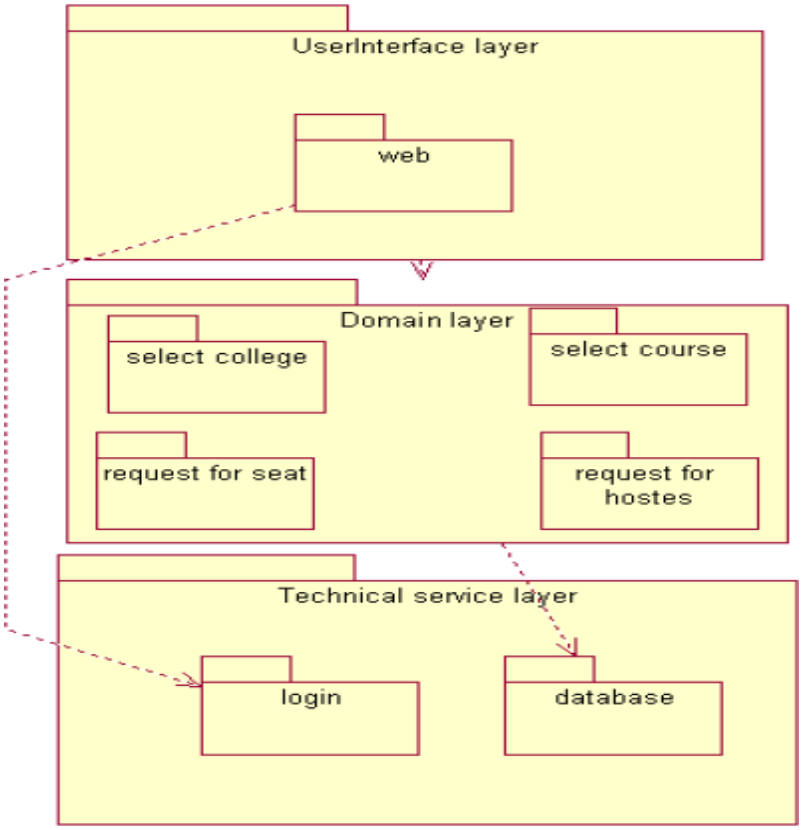
A package diagram is represented as a folder shown as a large rectangle with a top attached to its upper left corner. A package may contain both sub ordinate package and ordinary model elements. All uml models and

diagrams are organized into package. A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).
There are three types of layer. They are

a. User interface layer

b. Domain layer

c. Technical services layer



RESULT

Thus the mini project for online course reservation system has been successfully executed and codes are generated.

EX NO:6 AIRLINE/RAILWAY RESERVATION SYSTEM

Date:

AIM

To develop the Airline/Railway reservation System using Rational Rose Software.

(I) PROBLEM ANALYSIS AND PROJECT PLANNING

In the Airline/Railway reservation System the main process is a applicant have to login the database then the database verifies that particular username and password then the user must fill the details about their personal details then selecting the flight and the database books the ticket then send it to the applicant then searching the flight or else cancelling the process.

(II) OVERALL DESCRIPTION

Functionality

The database should be act as an main role of the e-ticketing system it can be booking the ticket in easy way.

Usability

The User interface makes the Credit Card Processing System to be efficient.

Performance

It is of the capacities about which it can perform function for many users at the same times efficiently that are without any error occurrence.

Reliability

The system should be able to process the user for their corresponding request.

(III) USE CASE DIAGRAM

The passenger can view the status of the reserved tickets. So the passenger can confirm his/her travel.

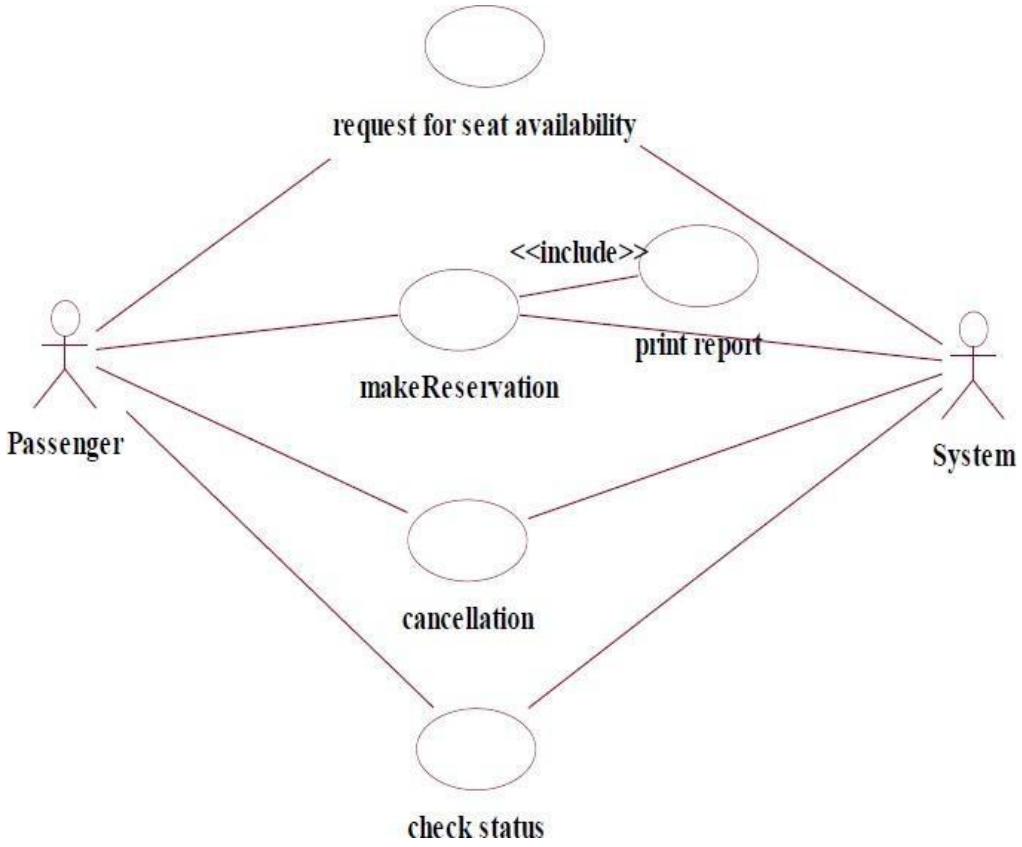
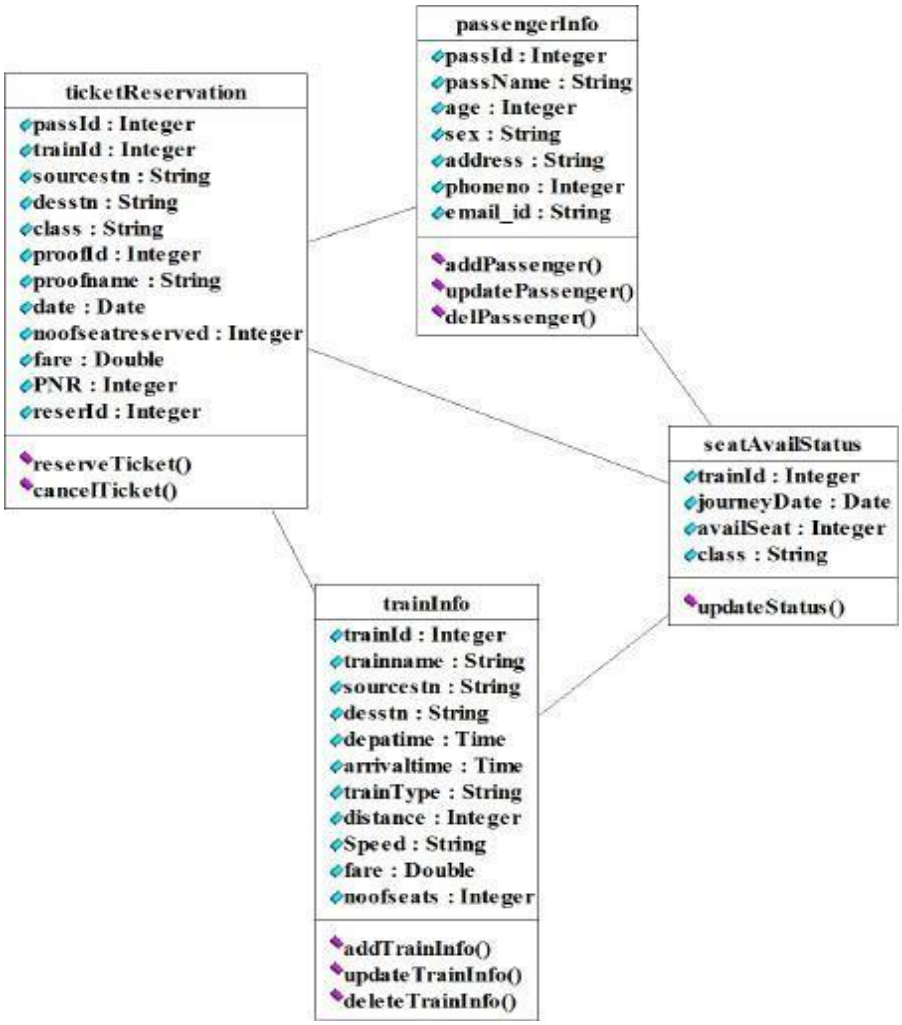


Fig. USE-CASE DIAGRAM FOR AIRLINE RESERVATION

(IV) CLASS DIAGRAM

The online ticket reservation system makes use of the following classes:

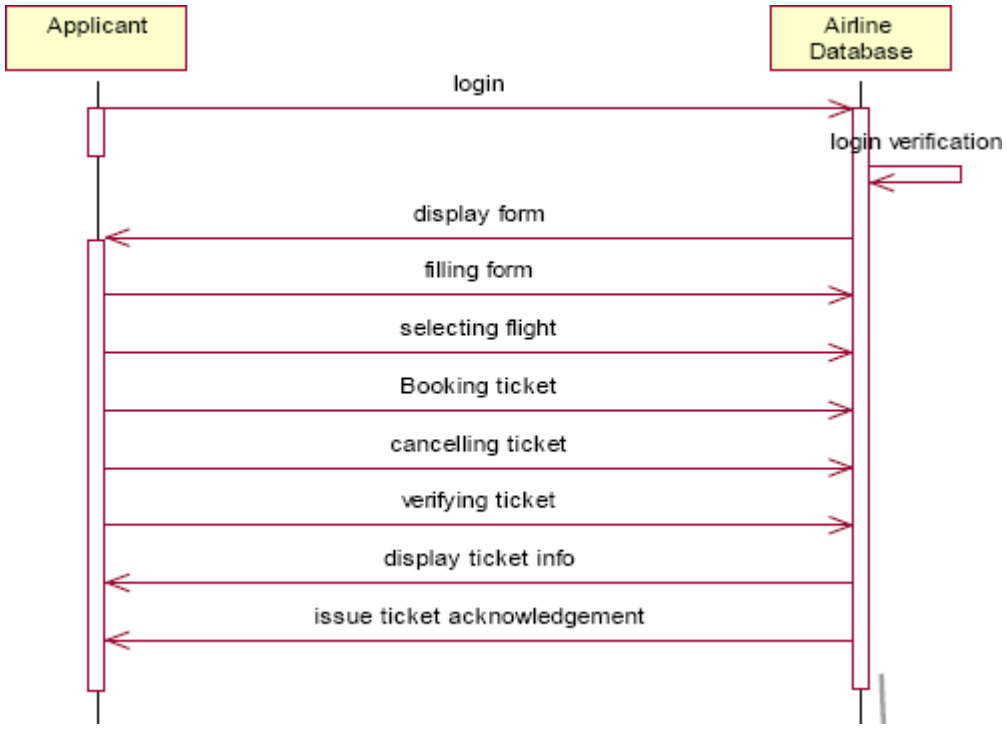
1. ticketReservation
2. trainInfo
3. passengerInfo
4. seatAvailStatus



SEQUENCE DIAGRAM

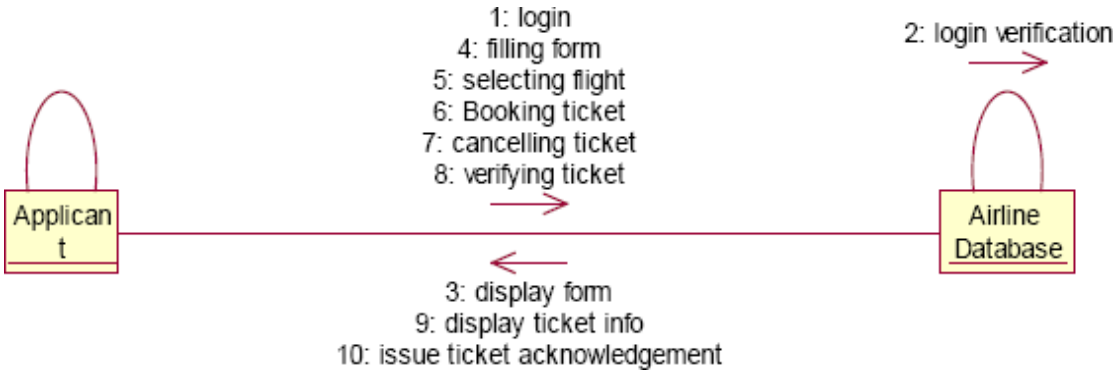
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

1. Vertical dimension-represent time.
2. Horizontal dimension-represent different objects.



COLLABRATION DIAGRAM

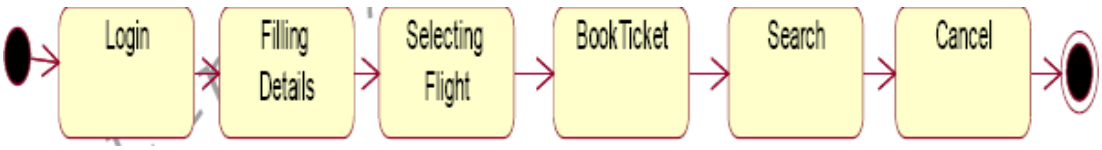
A collaboration diagram, also called a communication diagram or interaction diagram,. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.



STATE CHART DIAGRAM

The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is

represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle surrounding a small dot.

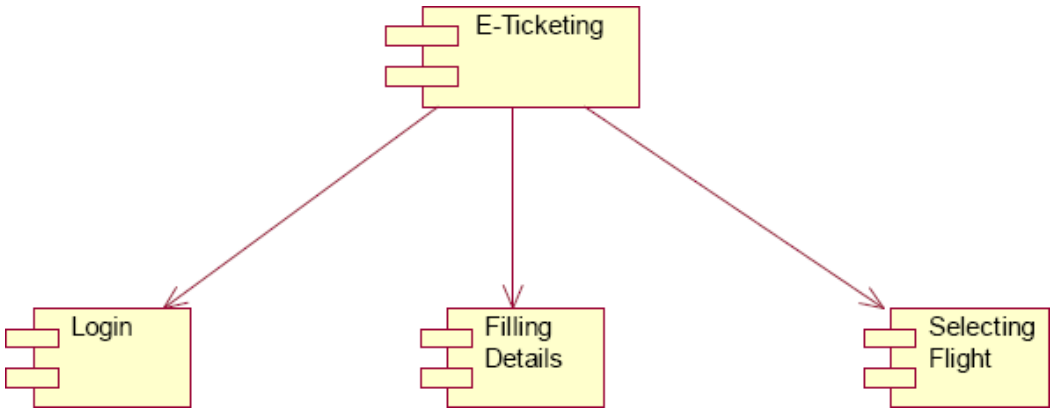


ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing the name of the operation.

COMPONENT DIAGRAM

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association.



RESULT

Thus the mini project for Airline/Railway reservation System has been successfully executed and codes are generated.

Ex no:7 SOFTWARE PERSONNEL MANAGEMENT SYSTEM

Date:

AIM:

To implement a software for software personnel management system.

(I) PROBLEM STATEMENT:

Human Resource management system project involves new and/or system upgrades of software of send to capture information relating to the hiring termination payment and management of employee. He uses system to plan and analyze all components and performance of metrics driven human resource functions, including recruitment, attendance, compensation, benefits and education. Human resources management systems should align for maximum operating efficiency with financial accounting operations customer relationship management,security and business lines as organization.

(II)USECASE DIAGRAM:

The HR of an organization involves recruitment training, monitoring and motivation of an employee. The HR also involves gives salary as observed in the payroll sheet. The employee undergoes training, receives the salary , gives the expected performance and manages time in order to complete a given task within the required period.

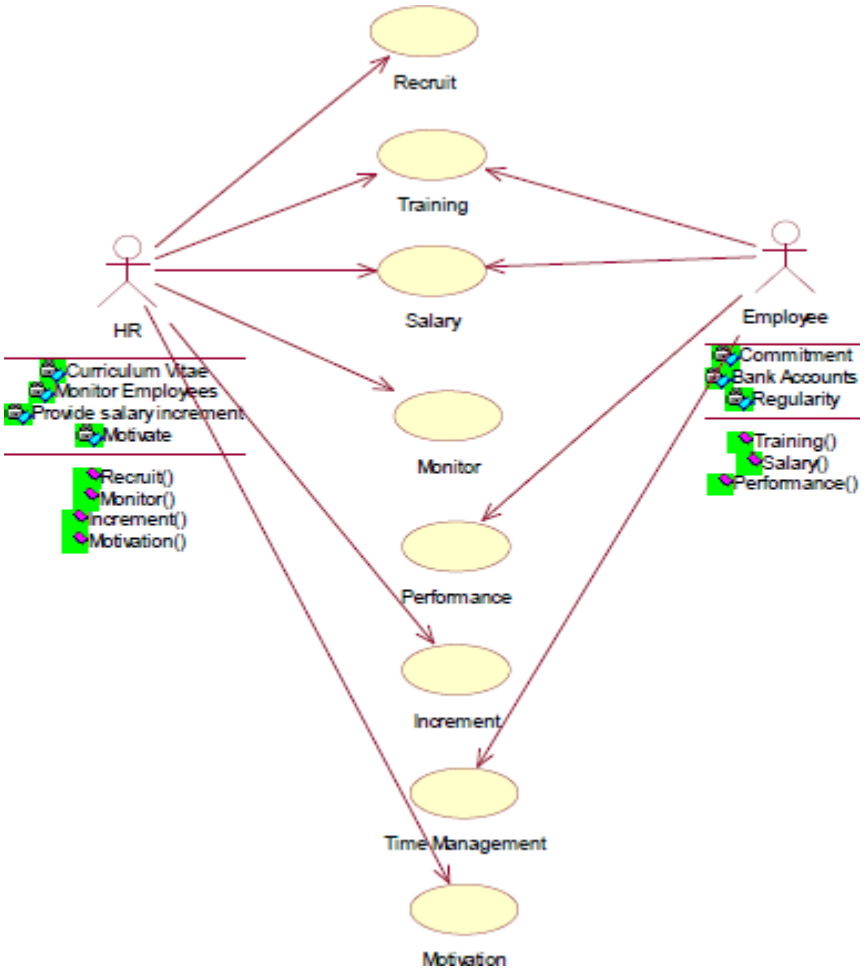


Fig.3. USE CASE DIAGRAM

(III)ACTIVITY DIAGRAM:

The activity diagram notation is an action, partition, fork join and object node. Most of the notation is self explanatory, two subtle points. Once an action finished, there is an automatic outgoing transaction. The diagram can show both control flow and data flow.

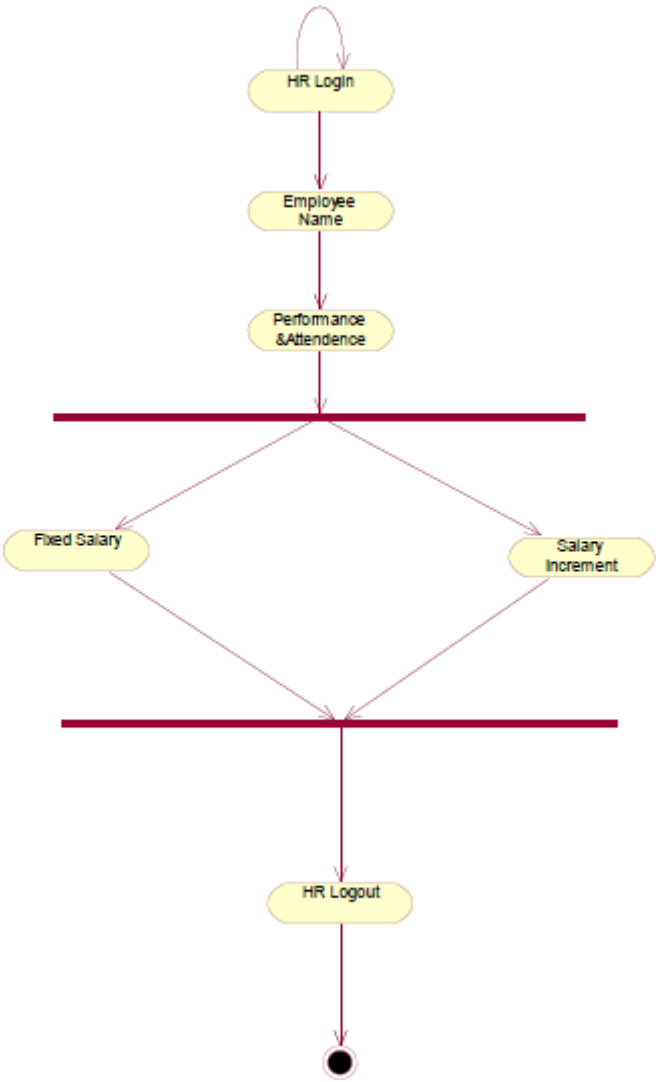


Fig.4. ACTIVITY DIAGRAM

(IV) CLASS DIAGRAM:

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

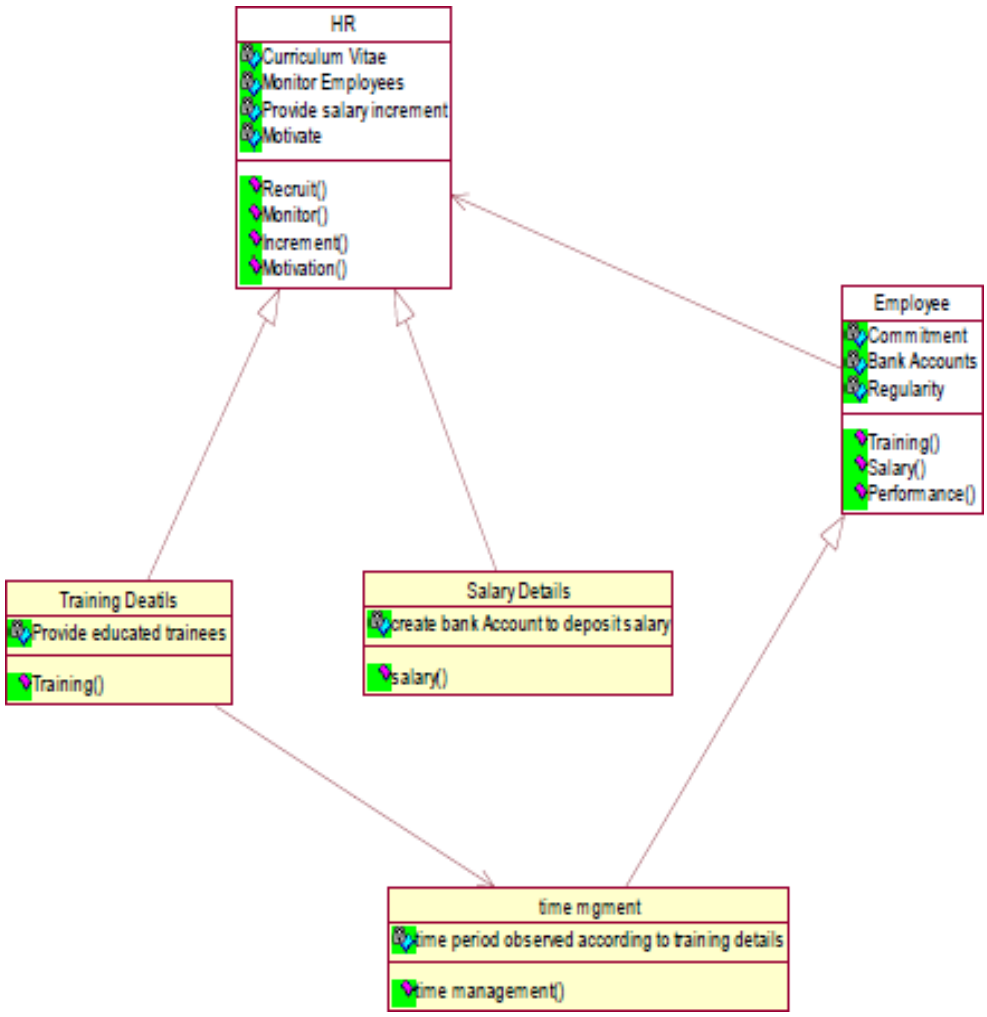


Fig.5.CLASS DIAGRAM

(V) INTERACTION DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

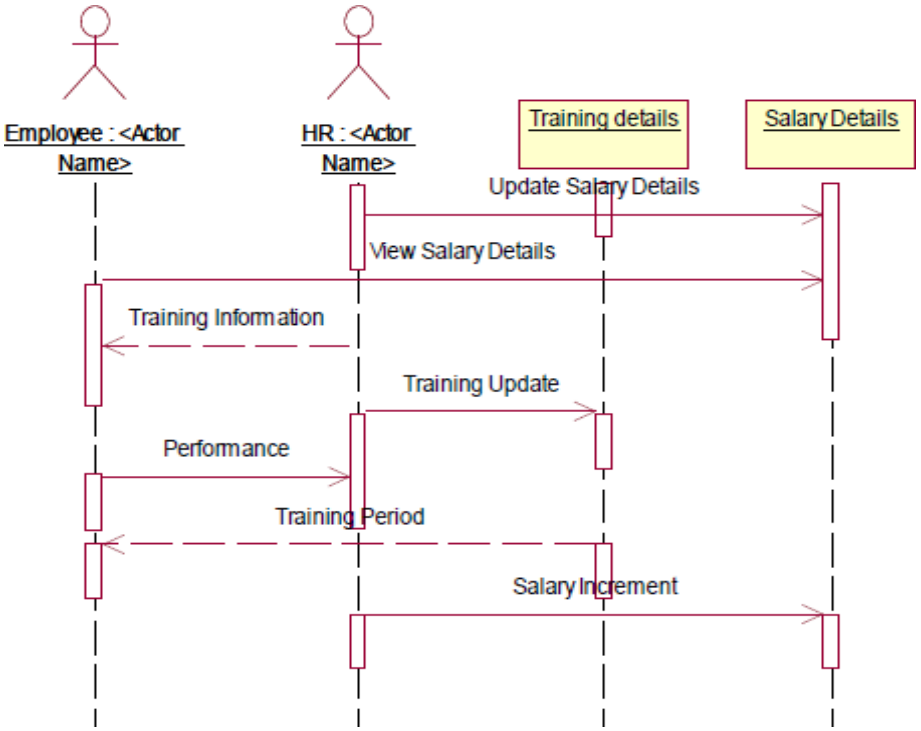


Fig.6.1.SEQUENCE DIAGRAM

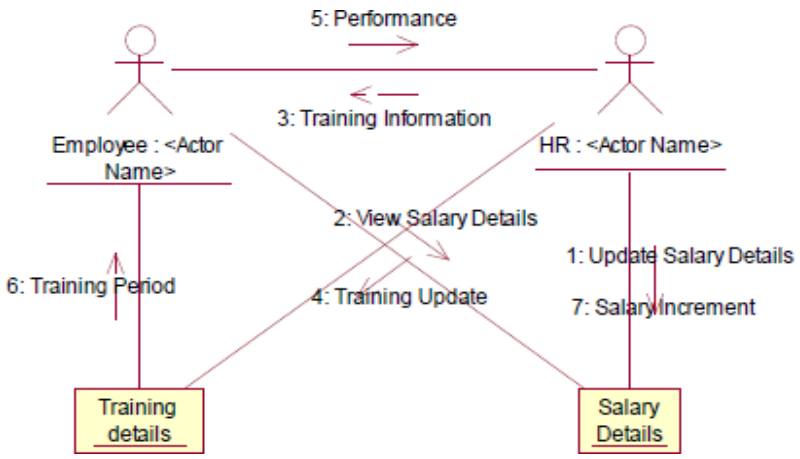


Fig.6.2.COLLABORATION DIAGRAM

(VI) State Transition Diagram

States of object are represented as rectangle with round corner, the transaction between the different states. A transition is a relationship between two state that indicates that when an event occur the object moves from the prior state to the subsequent.

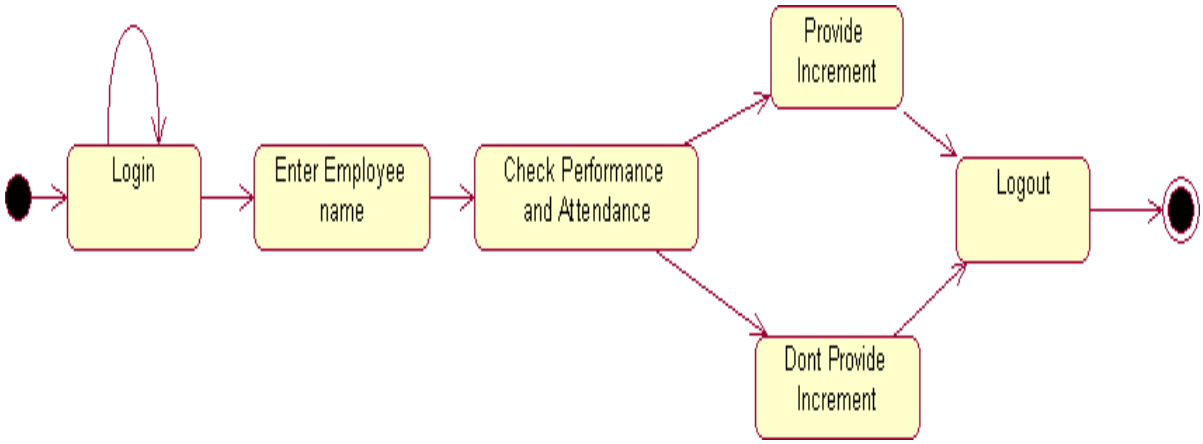


Fig.7.STATE TRANSITION DIAGRAM

(VII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

HR recruits employee for a company employee recruited by HR goes under training before actually working. Training period is given to the employee with the training details. The salary details for the employee are provided.

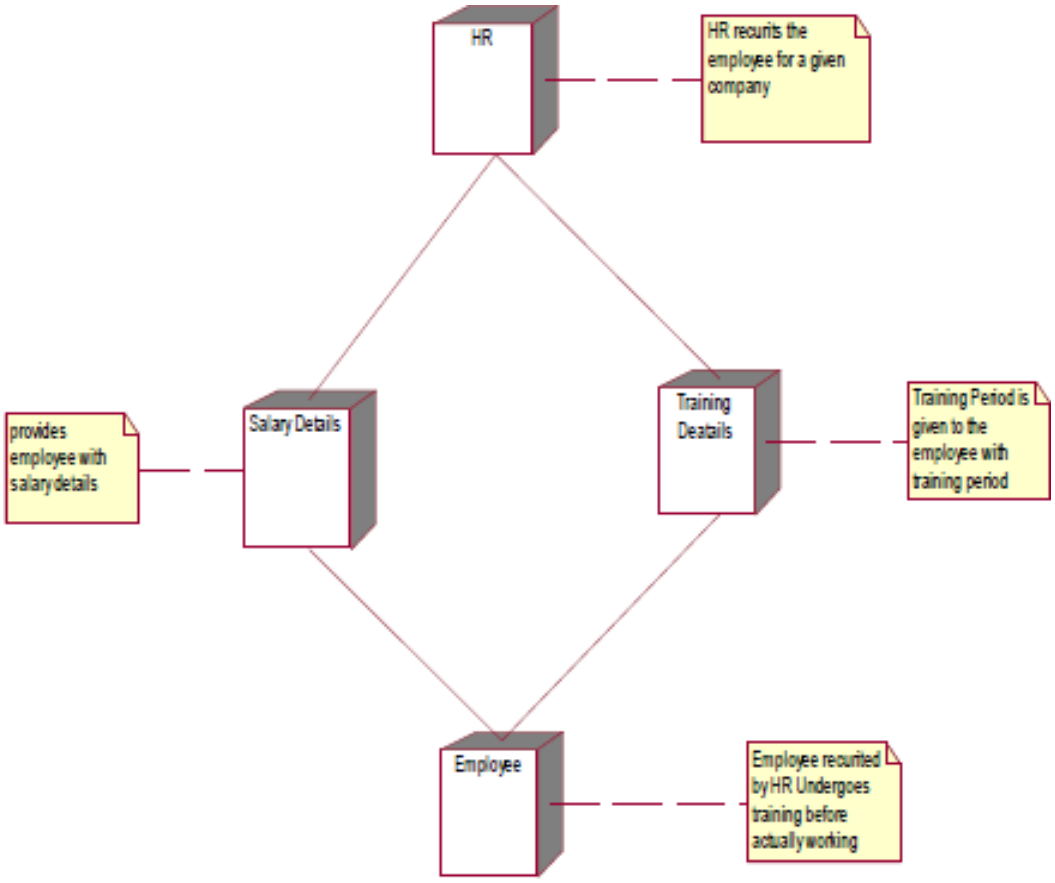


Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

The HR recruits, motivate and monitor the employee, HR also update the salary details and training details for reference. The employee are those who are recruited by HR and work for the company. The training details provide employees with training details which is updated by HR

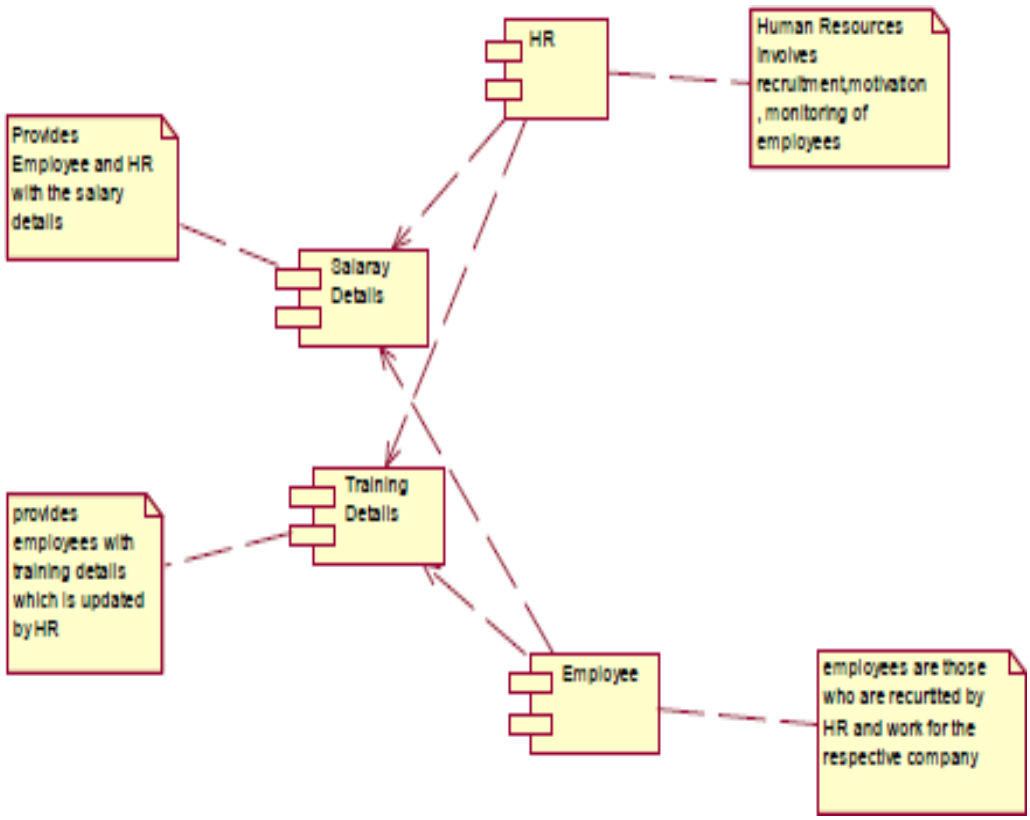


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for software personnel management system has been successfully executed and codes are generated.

Ex. No:8

CREDIT CARD PROCESSING

Date:

AIM:

To create a system to perform the credit card processing

(I) PROBLEM STATEMENT:

Credit card processing through offline involves the merchant collecting order information (including credit card numbers), storing this in a database on your site, and entering it using their on-site merchant credit card processing system. Takes time to manually enter credit card information for each order. This solution creates following cons:

(II)USECASE DIAGRAM:

USE-CASE NAME: PAYMENT APPROVAL

The transaction details are recorded by the credit card processor and results are securely relayed to the merchant. Merchant's site receives transaction result and does appropriate actions (e.g. saves the order & shows message).

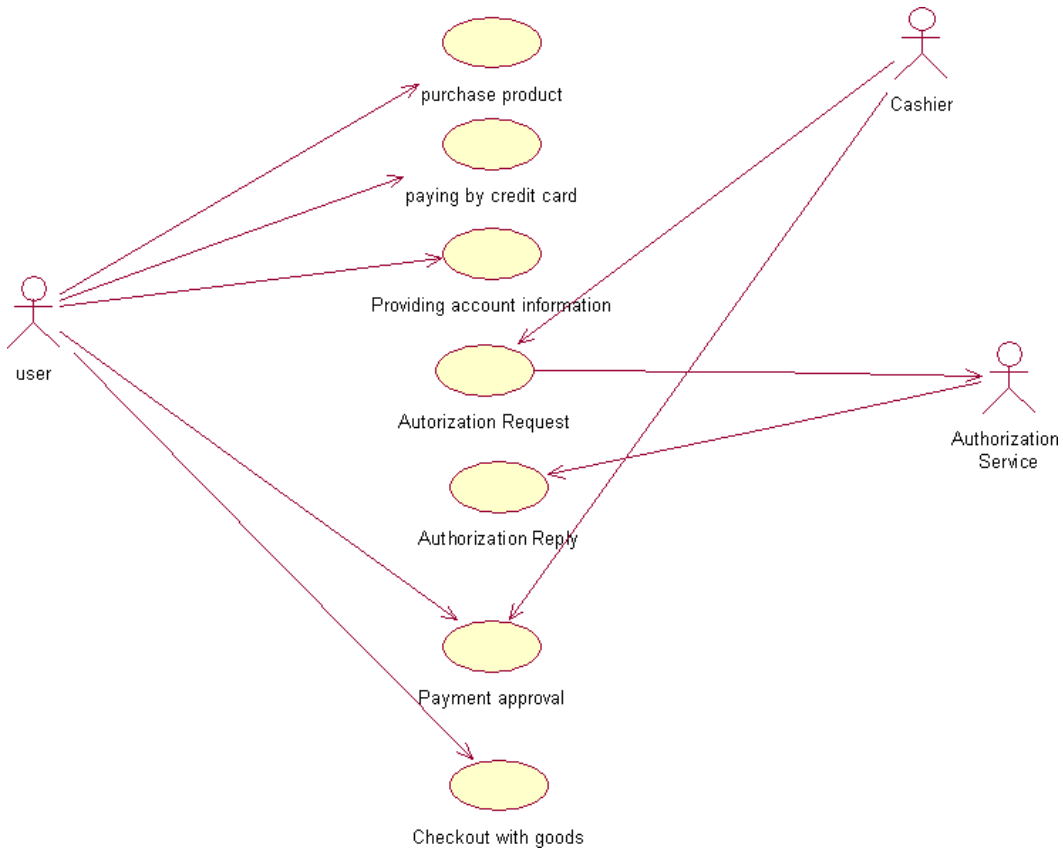


Fig.3. USECASE DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(III)CLASS DIAGRAM:

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

The Credit Card Processing system class diagram consists of three classes.

They are

Cashier

User

Authorization Service

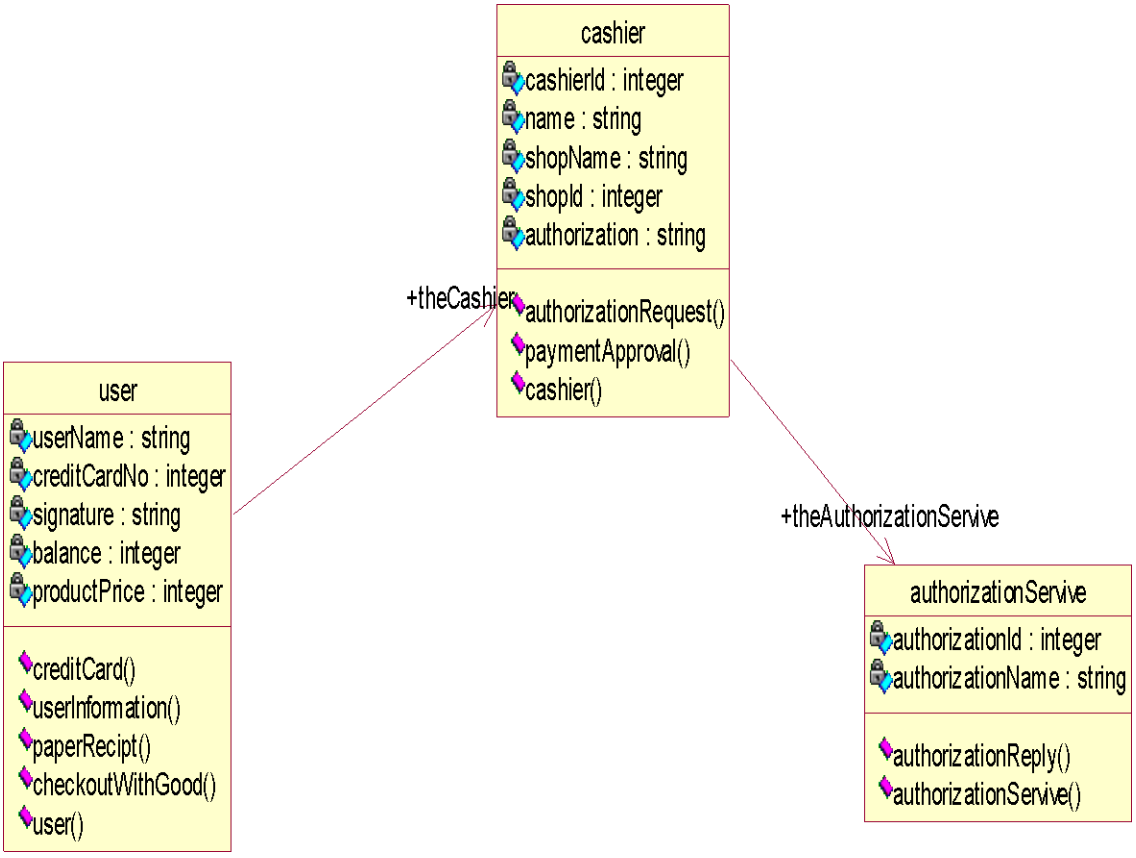


Fig.4.CLASS DIAGRAM

(IV) INTERACTION DIAGRAM:

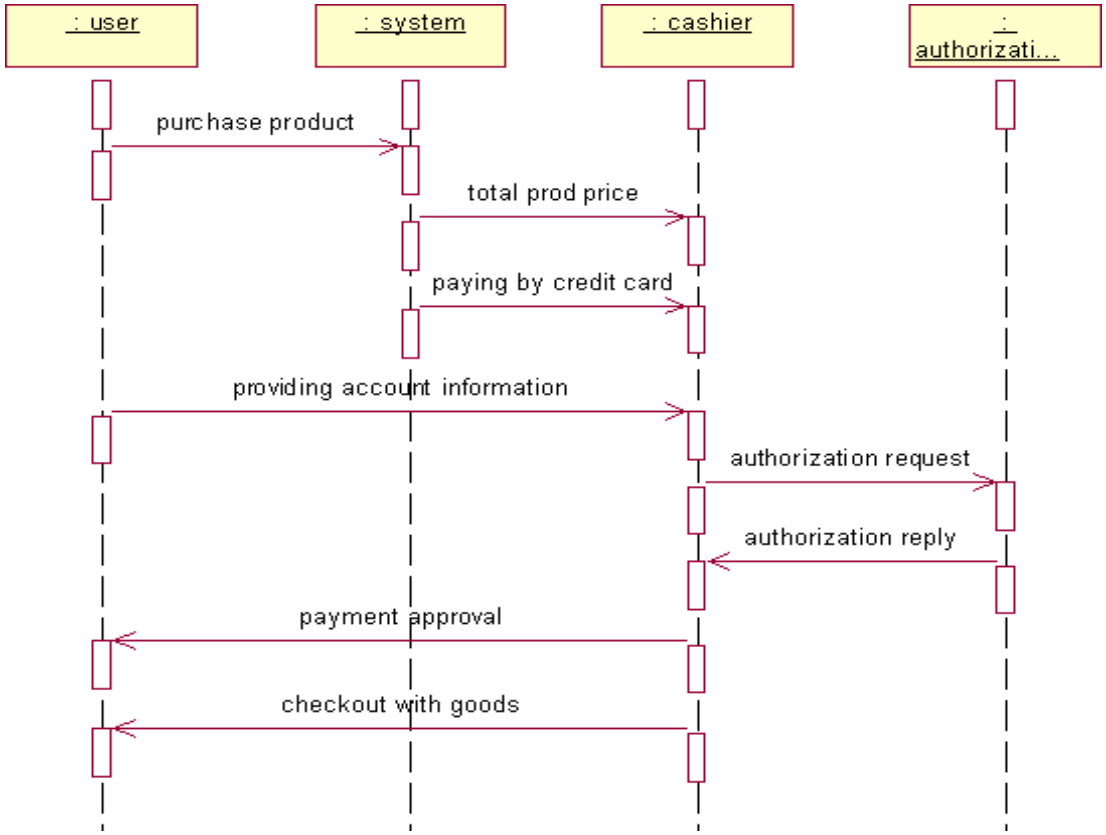


Fig.5.1.SEQUENCE DIAGRAM

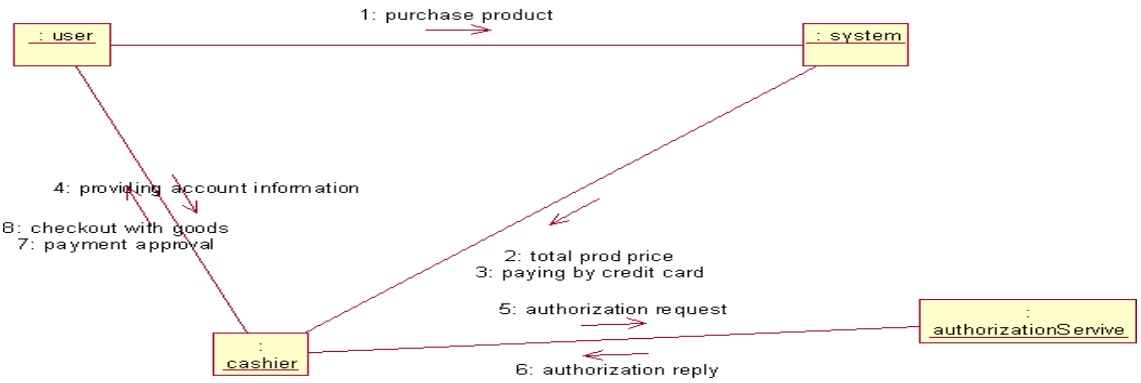
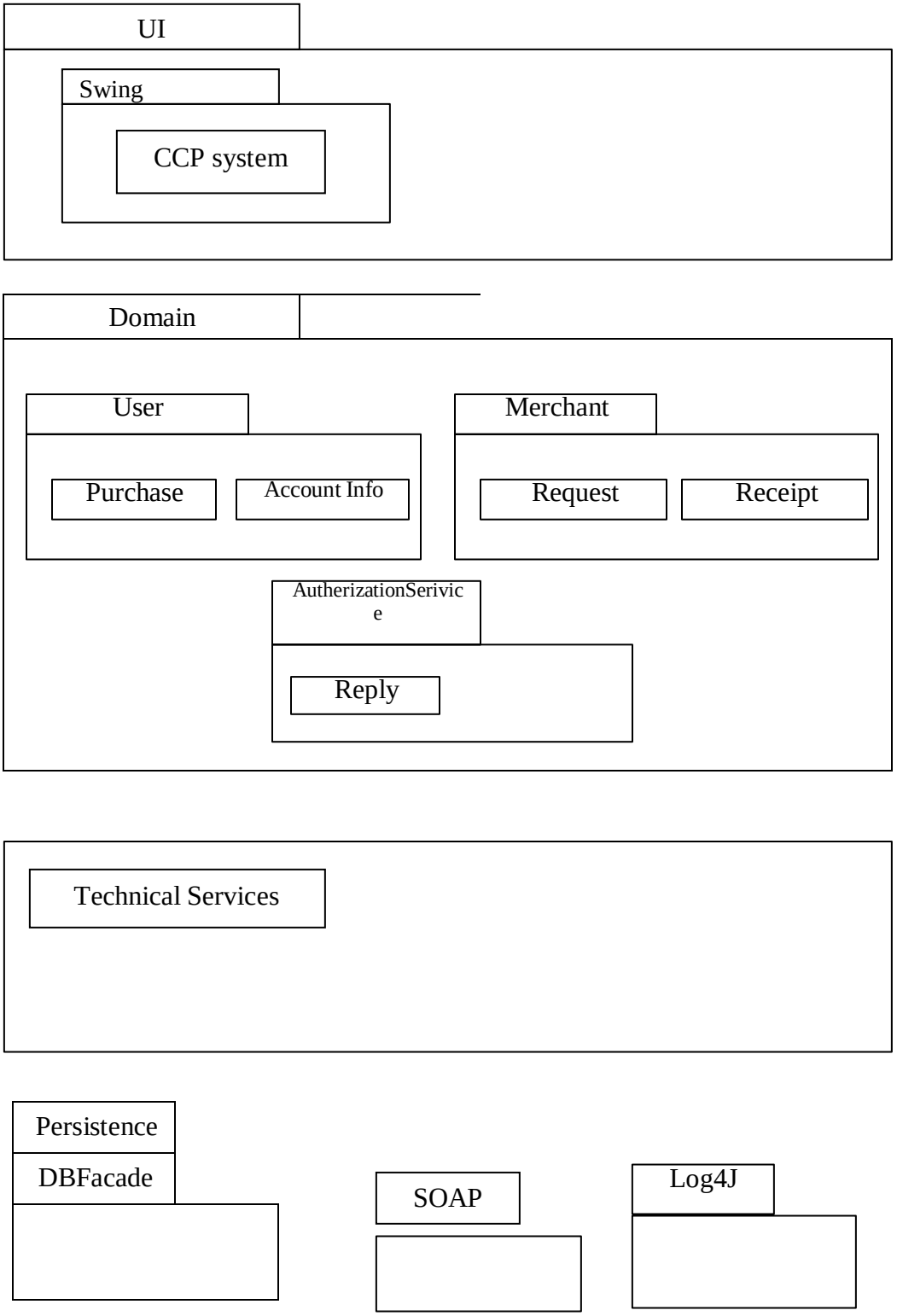


Fig.5.2.COLLABORATION DIAGRAM

(V) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(VI) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

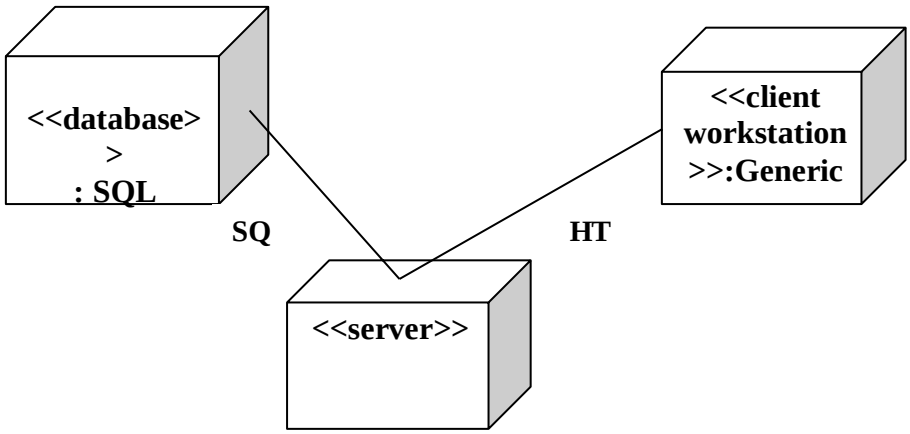


Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a

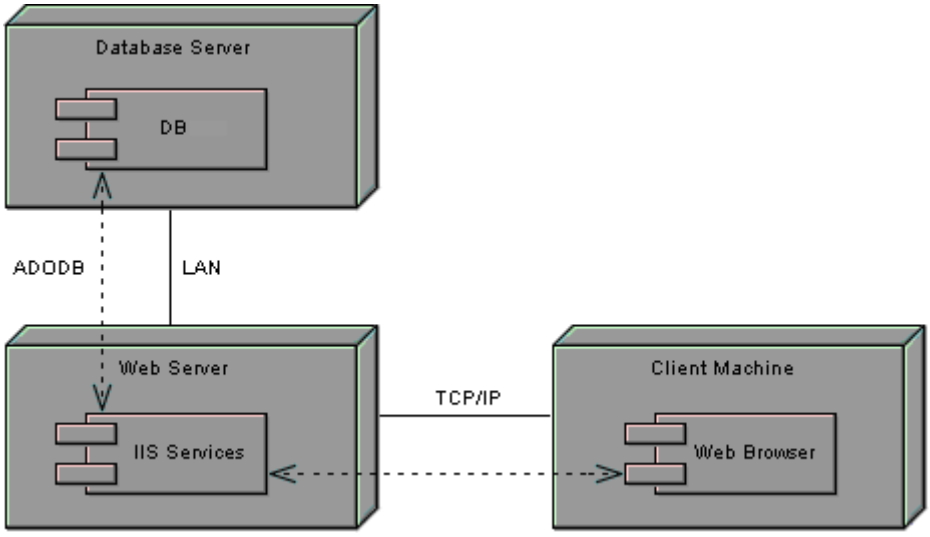


Fig.7.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for credit card processing system has been successfully executed and codes are generated.

Ex. No:9

OOK MANAGEMENT SYSTEM

Date:

AIM:

To create a system to perform E- book Management System.

(I) PROBLEM STATEMENT:

An E- Book lends books and magazines to member, who is registered in the system. Also it handles the purchase of new titles for the Book Bank. Popular titles are brought into multiple copies. Old books and magazines are removed when they are out or date or poor in condition. A member can reserve a book or magazine that is not currently available in the book bank, so that when it is returned or purchased by the book bank, that person is notified. The book bank can easily create, replace and delete information about the tiles, members, loans and reservations from the system.

(II) USE-CASE DIAGRAM:

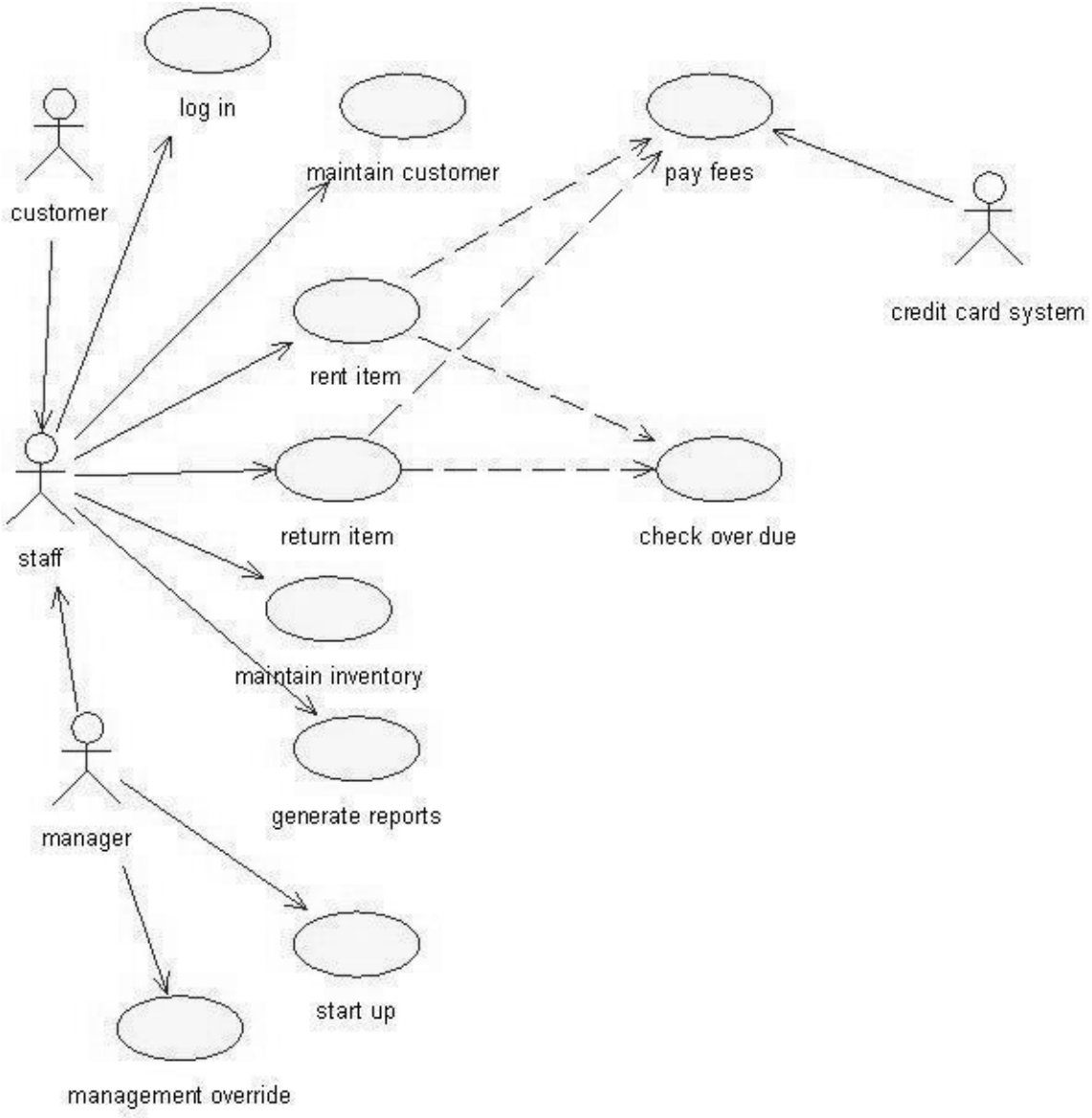
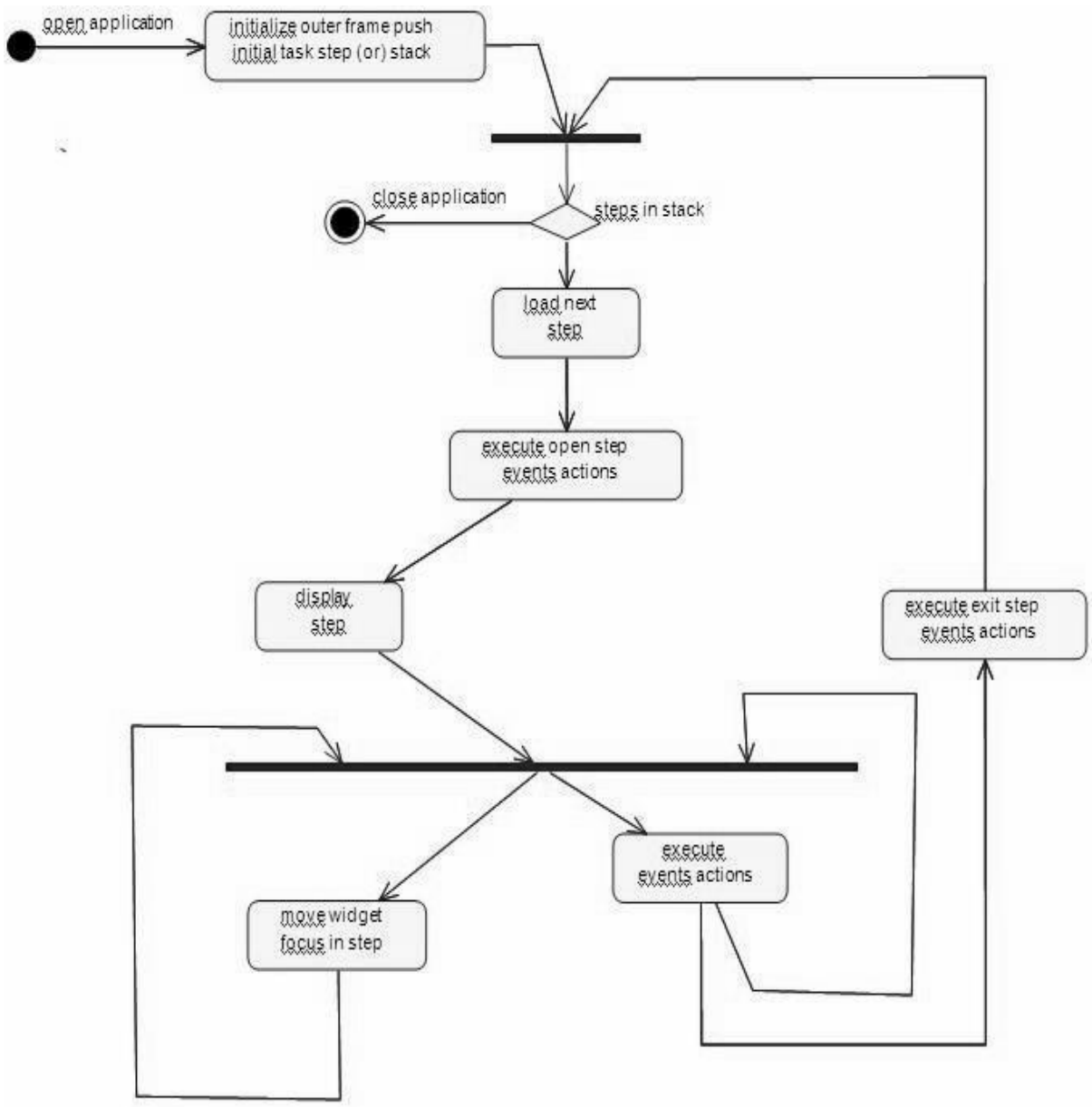


Fig.3.USE-CASE DIAGRAM FOR E-BOOK SYSTEM

(III) ACTIVITY DIAGRAM:



(IV) CLASS DIAGRAM

The class diagram, also referred to as object modeling is the main static analysis diagram. The main task of object modeling is to graphically show what each object will do in the problem domain. The problem domain describes the structure and the relationships among objects.

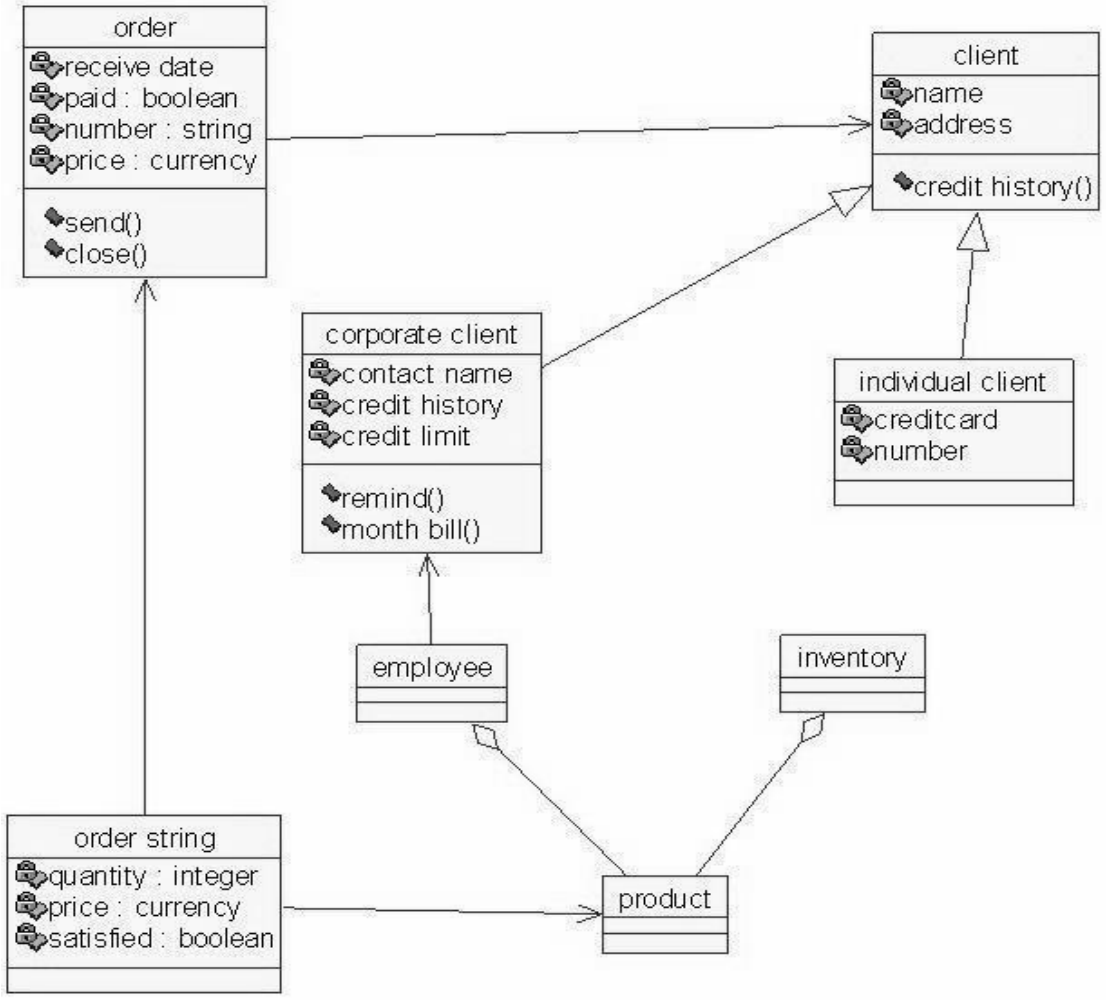


Fig.5.CLASS DIAGRAM FOR E-BOOK SYSTEM

(V) INTERACTION DIAGRAM:

A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions and actions to or from users or external devices.

An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the "form" object is requesting an operation be performed by the "to" object. The "to" object performs the operation using a method that the class contains.

It is also represented by the order in which things occur and how the objects in the system send message to one another.

The sequence diagram and collaboration diagram are given below.

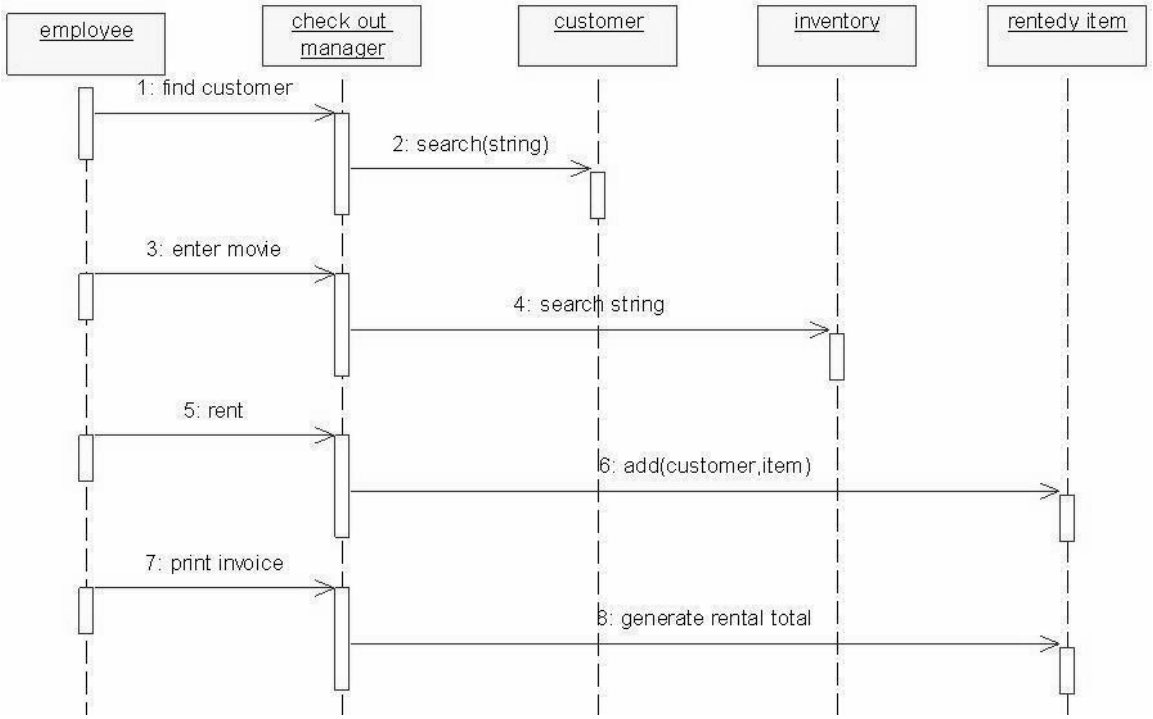


Fig.6.1.SEQUENCE DIAGRAM

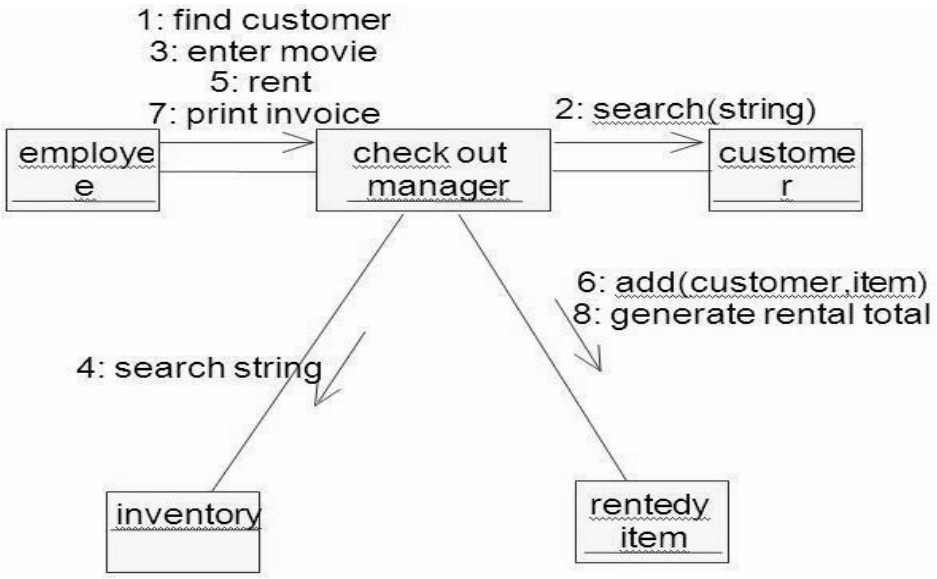
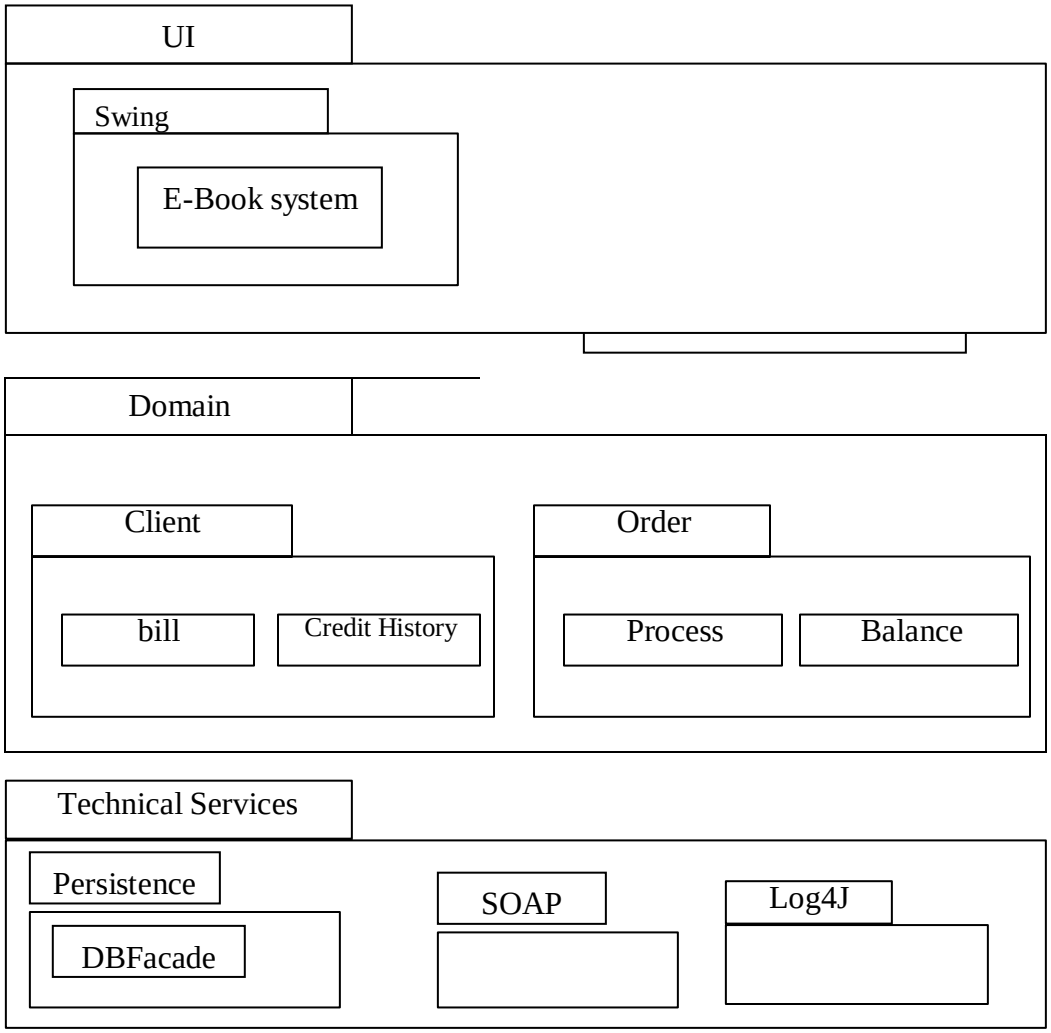


Fig.6.2.COLLABORATION DIAGRAM

(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(VII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

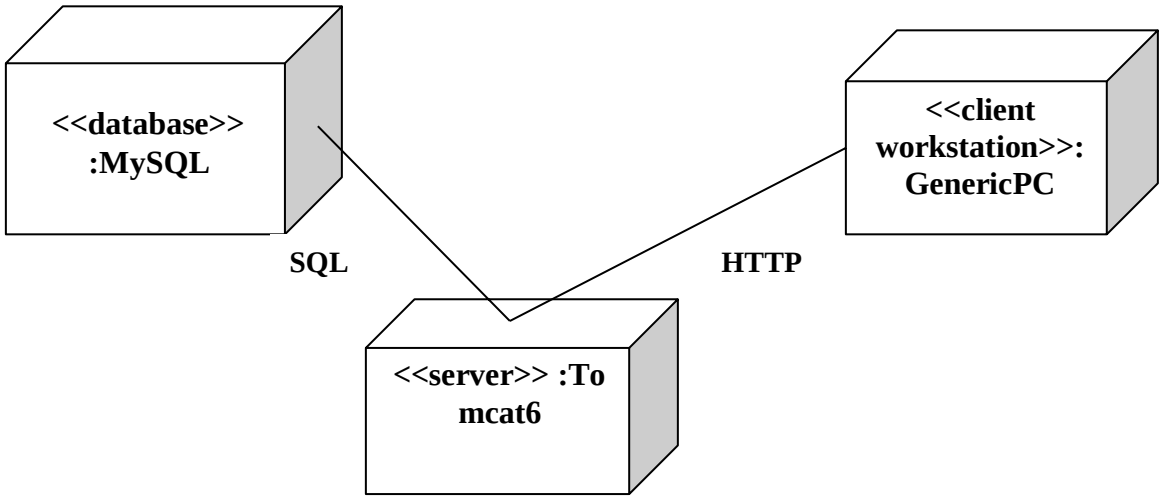


Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.

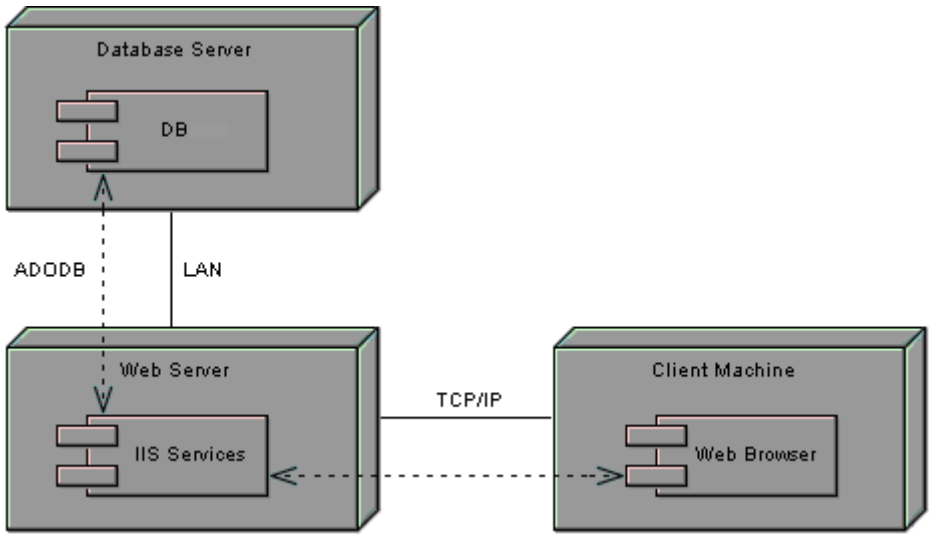


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for E-Book System has been successfully executed and codes are generated.

Ex.No:10

RECRUITMENT SYSTEM

Date:

AIM:

To create an automated system to perform the Recruitment System Process.

(I) 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.

(II)USECASE DIAGRAM:

The Recruitment Automation system use cases are:

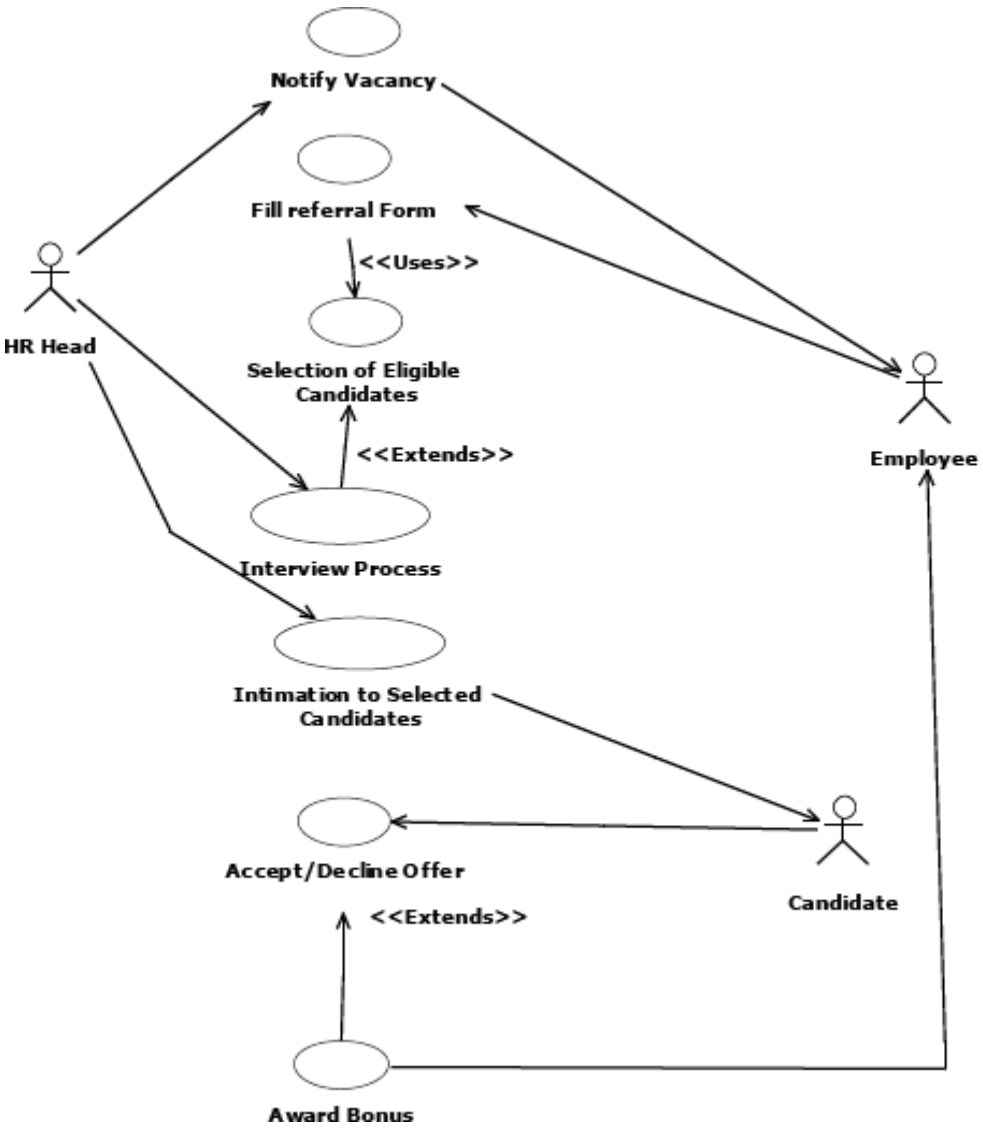


Fig.3. UML USE CASE DIAGRAM

(III)ACTIVITY DIAGRAM:

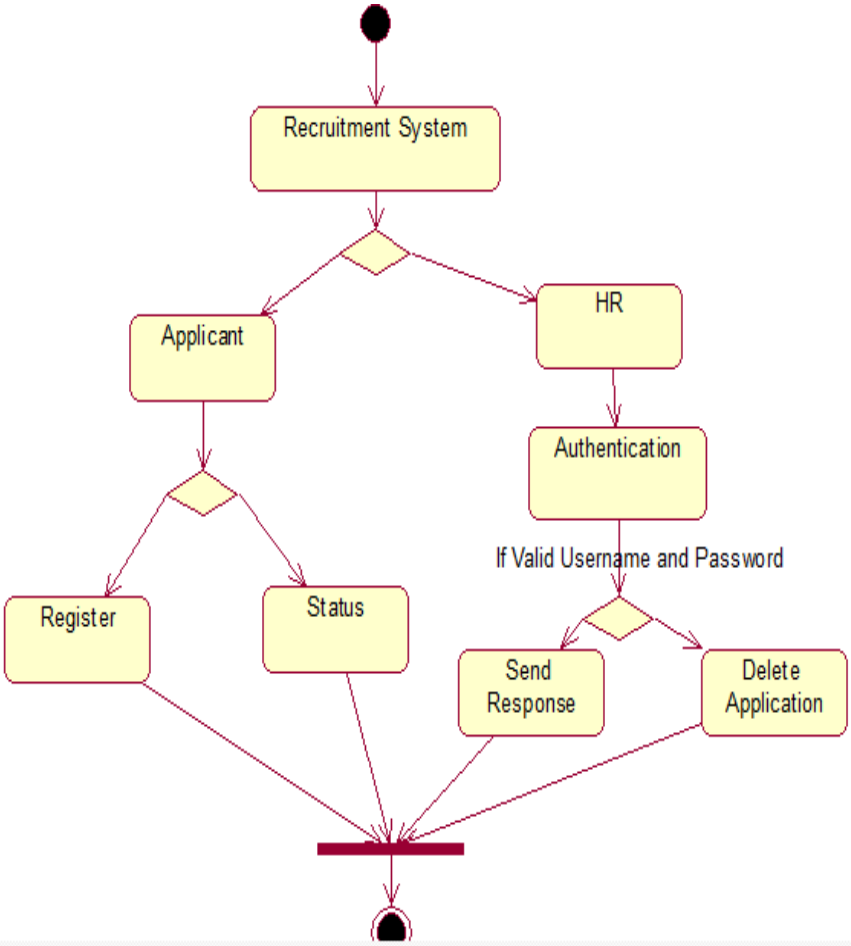


Fig.4. ACTIVITY DIAGRAM

(IV) UML CLASS DIAGRAM:

The UML class diagram is to illustrate class interfaces and their actions. They are used for static object modeling, we have already introduced and used their UML diagram while domain modeling.

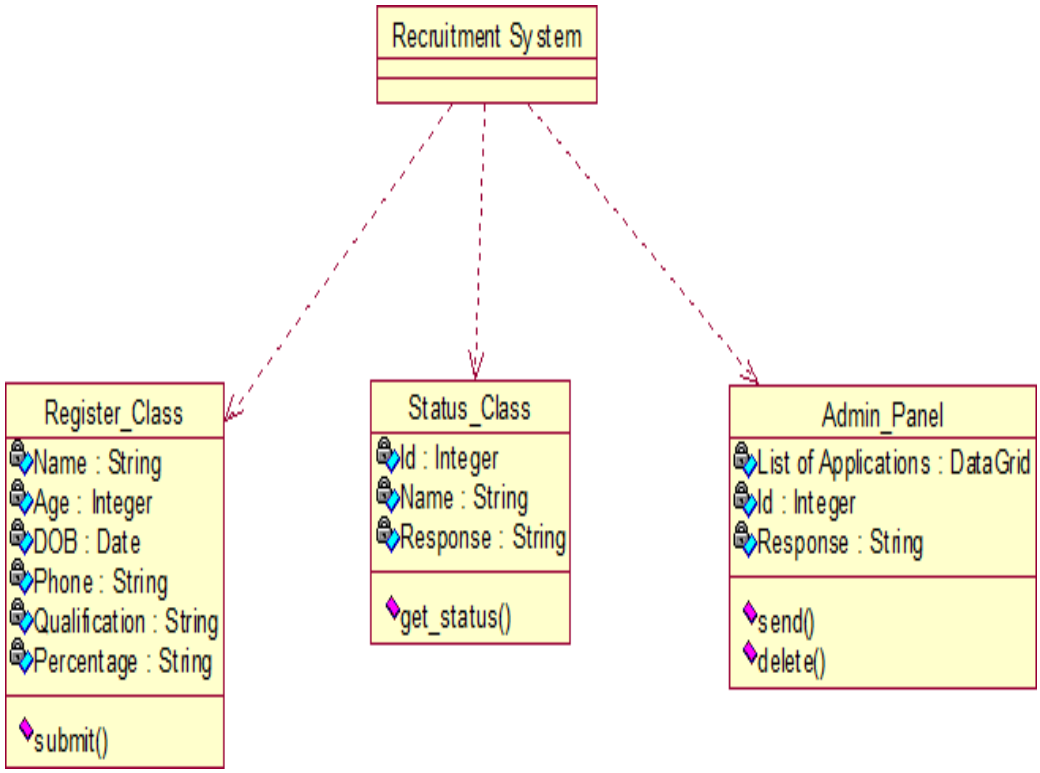


Fig.5. UML CLASS DIAGRAM

(V) UML SEQUENCE DIAGRAM:

A sequence diagram illustrates a kind of format in which each object interacts via message. It is generalize between two or more specialized diagram.

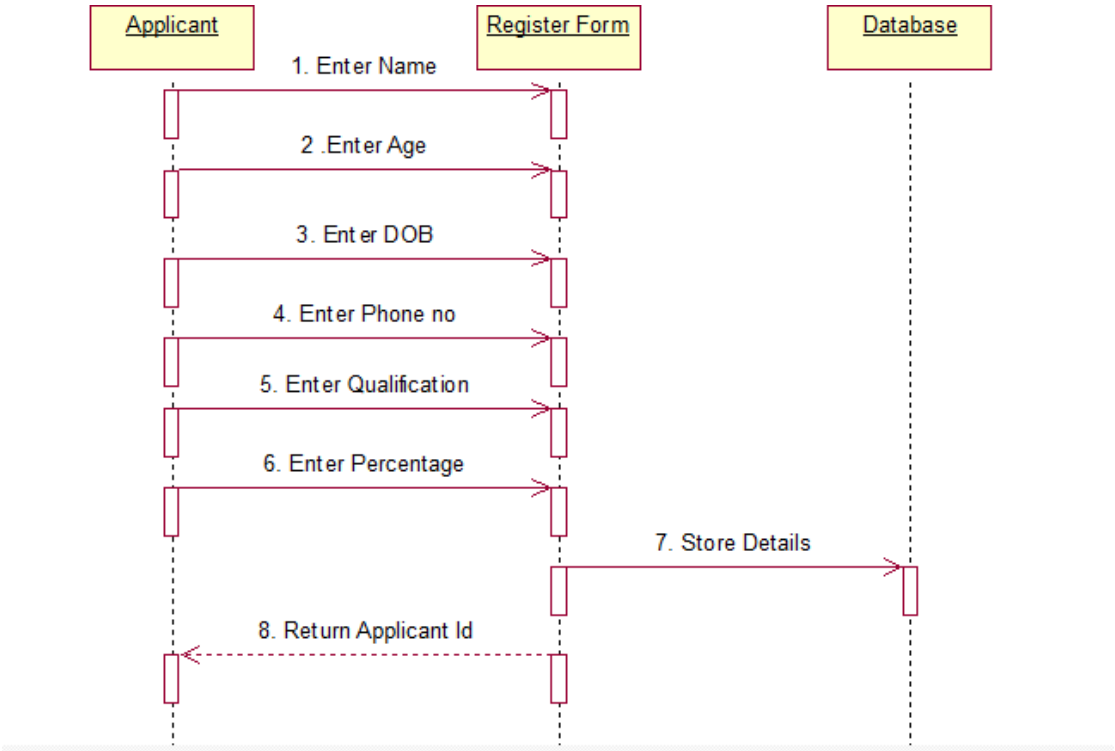


Fig. 6.1SEQEUENCE DIAGRAM FOR Register:

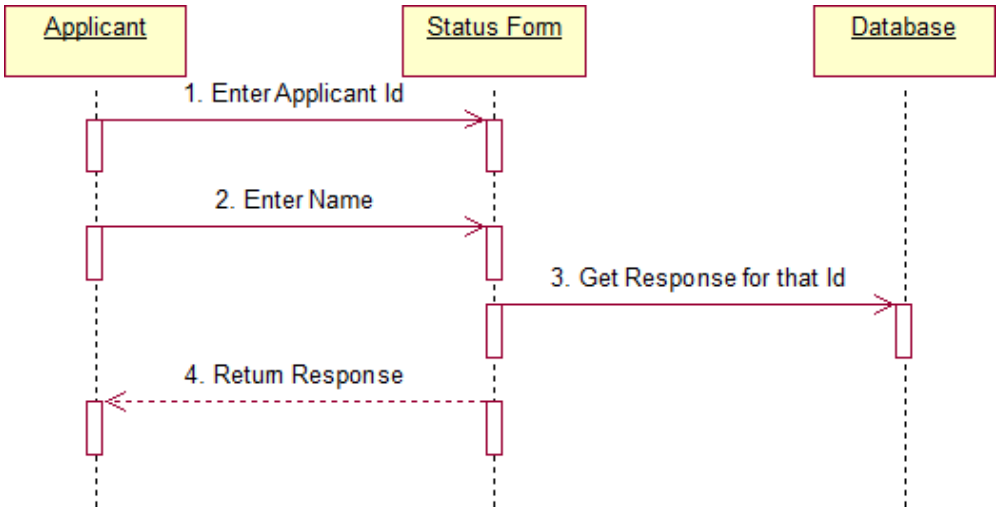


Fig.6.2. SEQUENCE DIAGRAM FOR STATUS

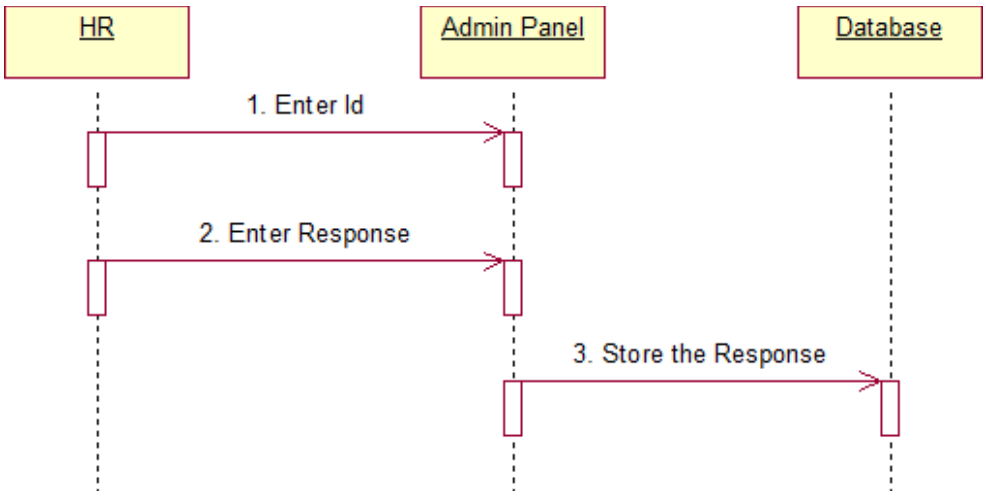
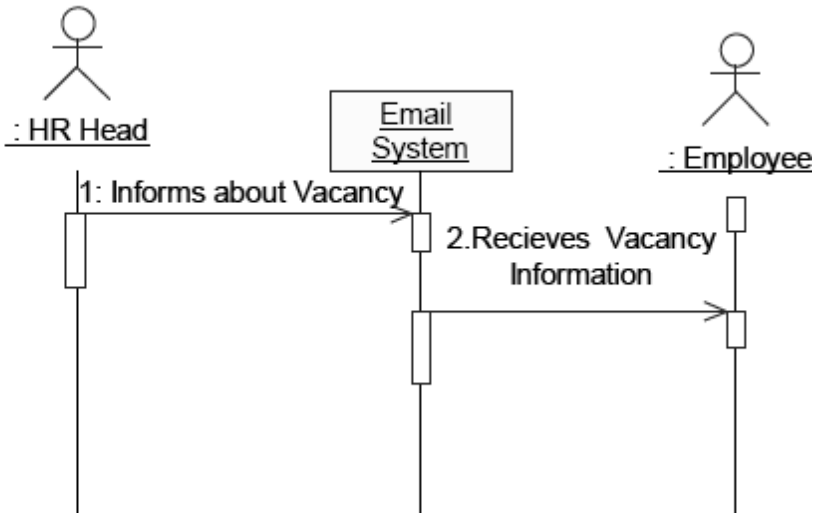


Fig.6.3. SEQUENCE DIAGRAM FOR Admin

Notify Vacancy:



(VI) UML COLLABRATION DIAGRAM:

Communication diagram illustrate that object interact on a graph or network format in which object can be placed where on the diagram. In collaboration diagram the object can be placed in anywhere on the diagram. The collaboration comes from sequence diagram.

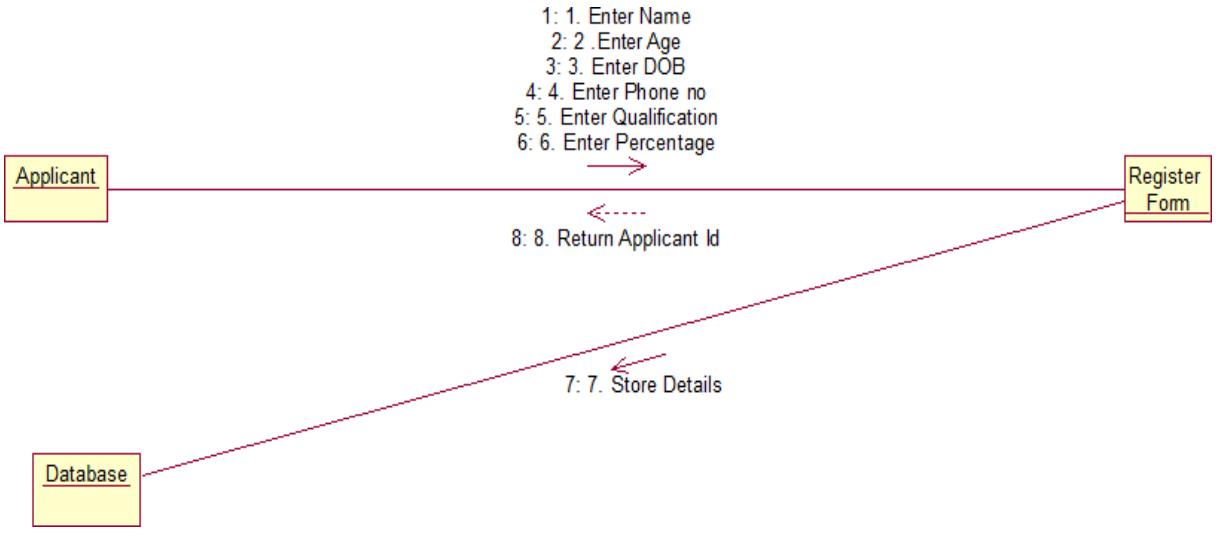


Fig.7.1COLLABRATION DIAGRAM For Register

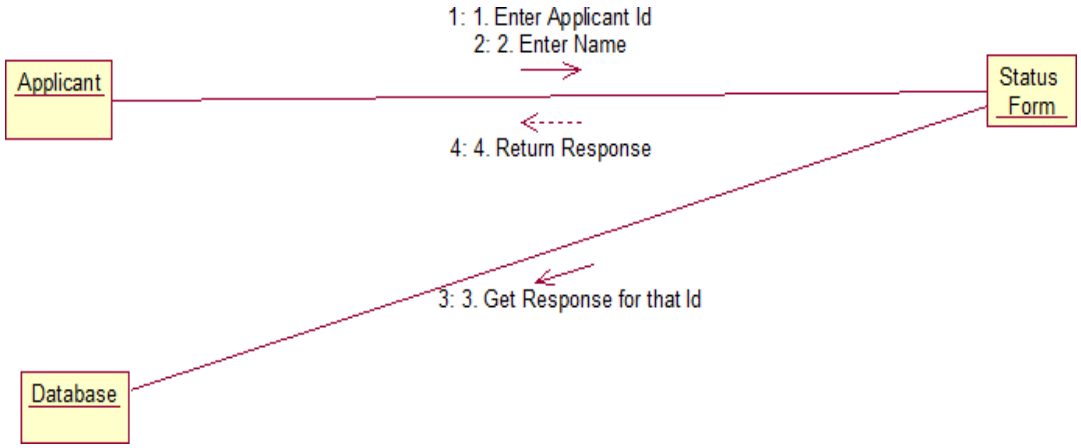


Fig.7.2. COLLABORATION DIAGRAM FOR Status

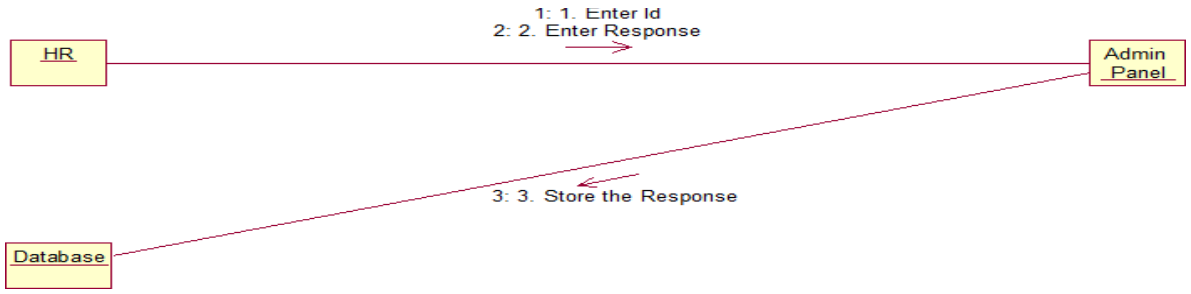
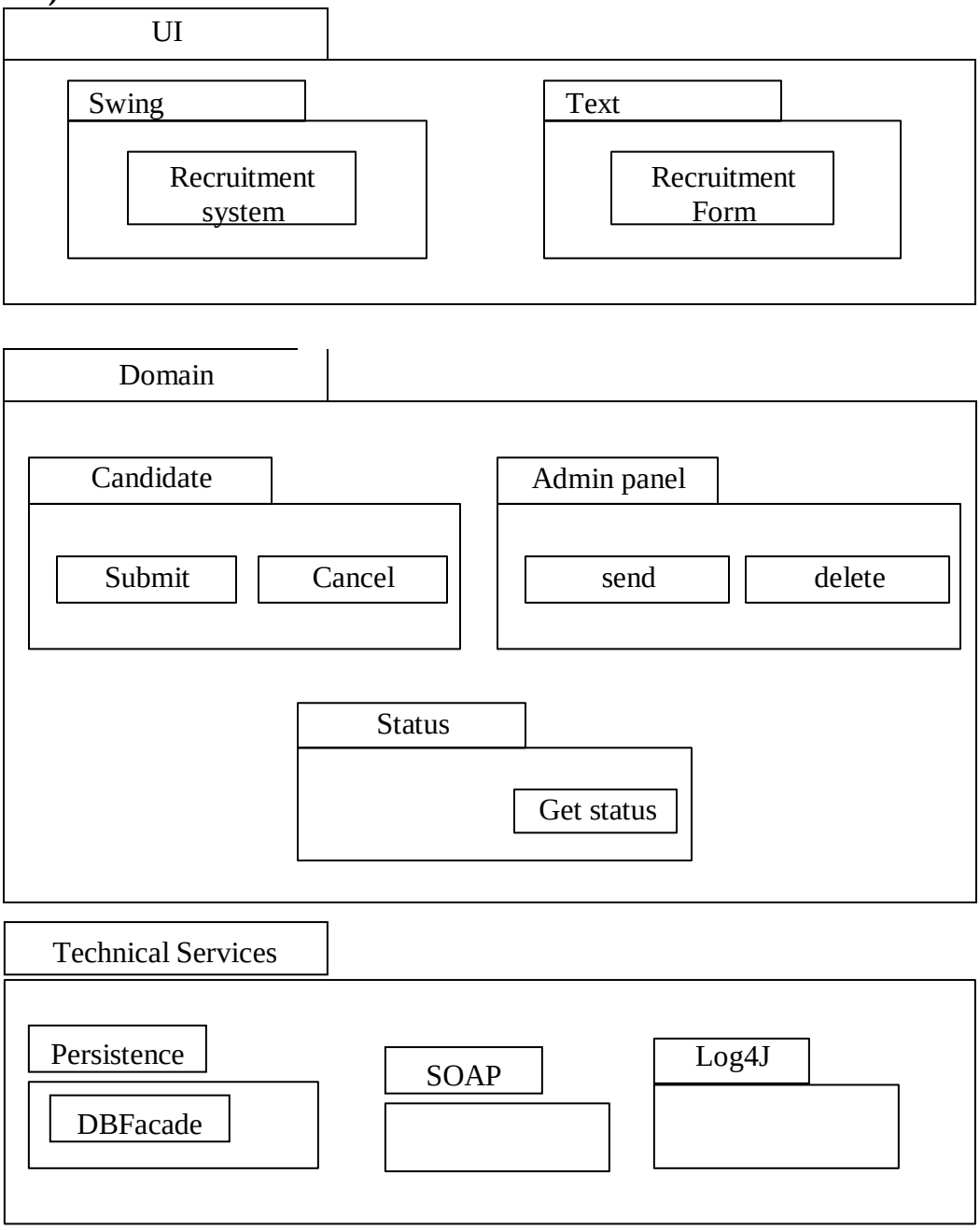


Fig.7.3.COLLABORATION DIAGRAM FOR Admin

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

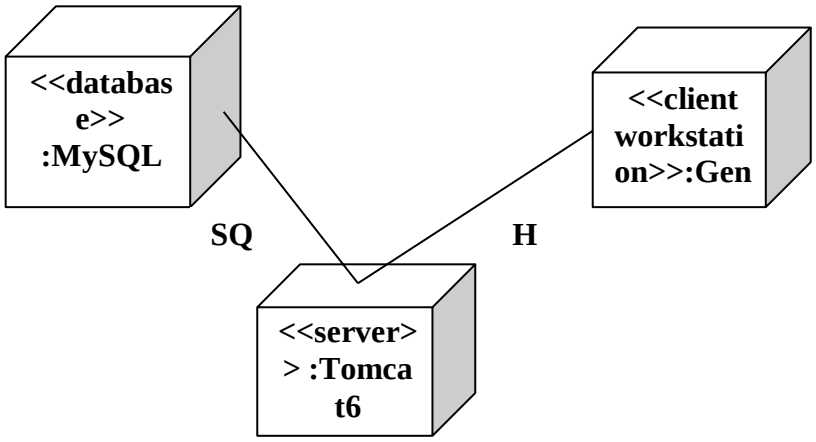


Fig.9.1.DEPLOYMENT DIAGRAM

Component Diagram

Component diagrams are used to visualize the organization and relationships among components in a system.

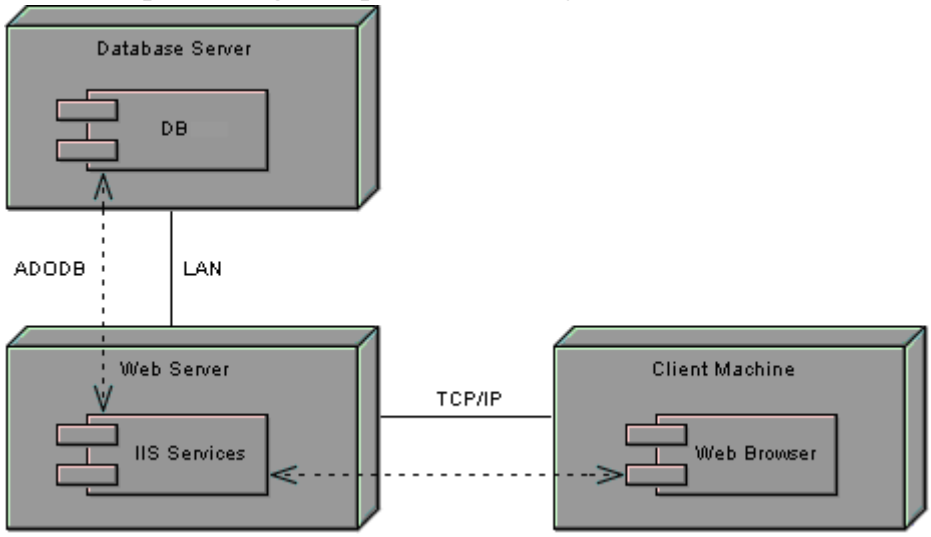


Fig.9.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for recruitment system has been successfully executed and codes are generated.

Ex.No : 11 **FOREIGN TRADING SYSTEM**

Date :

AIM

To design a project Foreign Trading System using Rational Rose Software and to implement the software in Visual Basic

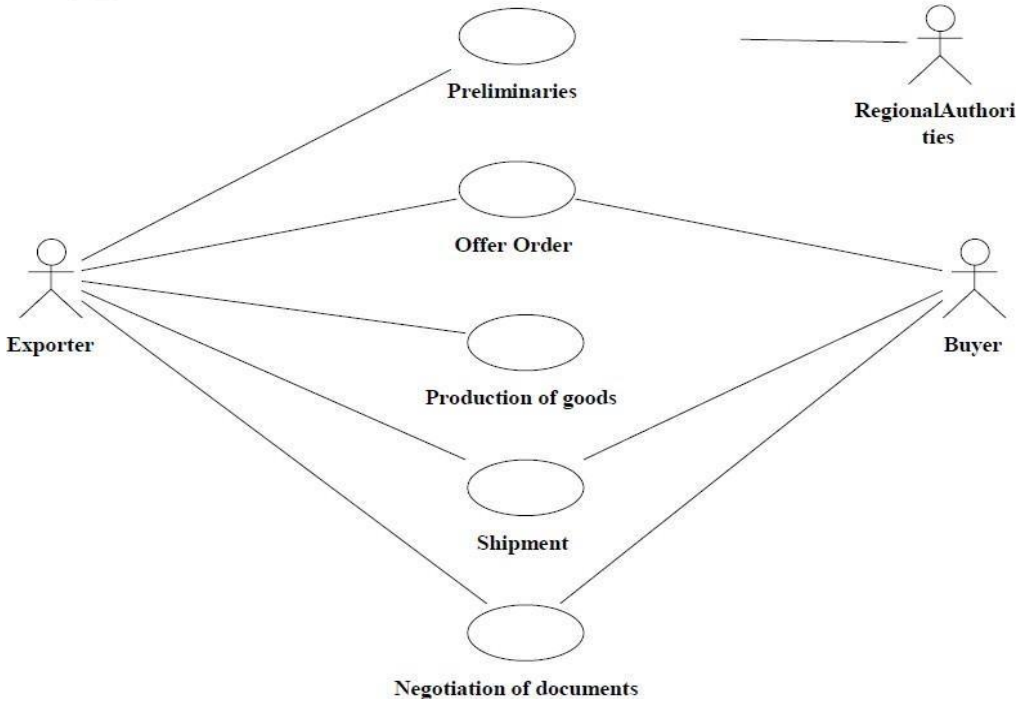
() PROBLEM STATEMENT

The steps involved in Foreign Trading System are:

The forex system begins its process by getting the username and password from the trader. After the authorization permitted by the administrator, the trader is allowed to perform the sourcing to know about the commodity details.After the required commodities are chosen, the trader places the order.The administrator checks for the availability for the required commodities and updates it in the database. After the commodities are ready for the trade, the trader pays the amount to the administrator.The administrator in turn provides the bill by receiving the amount and updates it in the database.The trader logouts after the confirmation message has been received.

UML DIAGRAMS

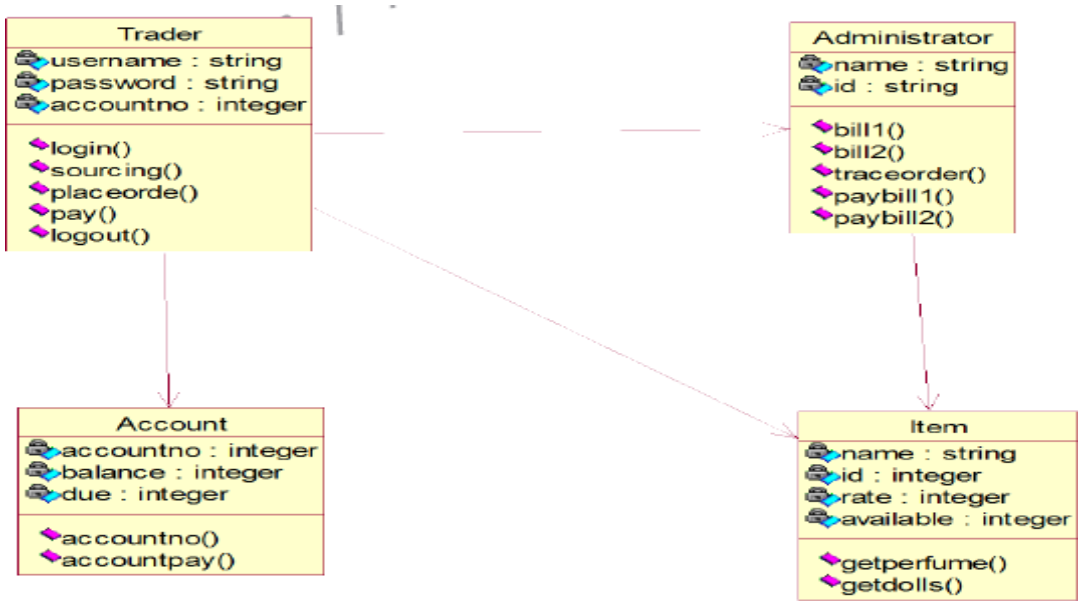
The exporter submits the relevant documents to his buyer (banker) for getting the payment for the goods exported.



(II) USE CASE DIAGRAM

CLASS DIAGRAM

A class diagram is a type of static structure diagram that describes the structure of a system. The classes in the class diagram represent both the main objects and or interactions in the application.The class diagram is represented using rectangular boxes each of which contains three parts:

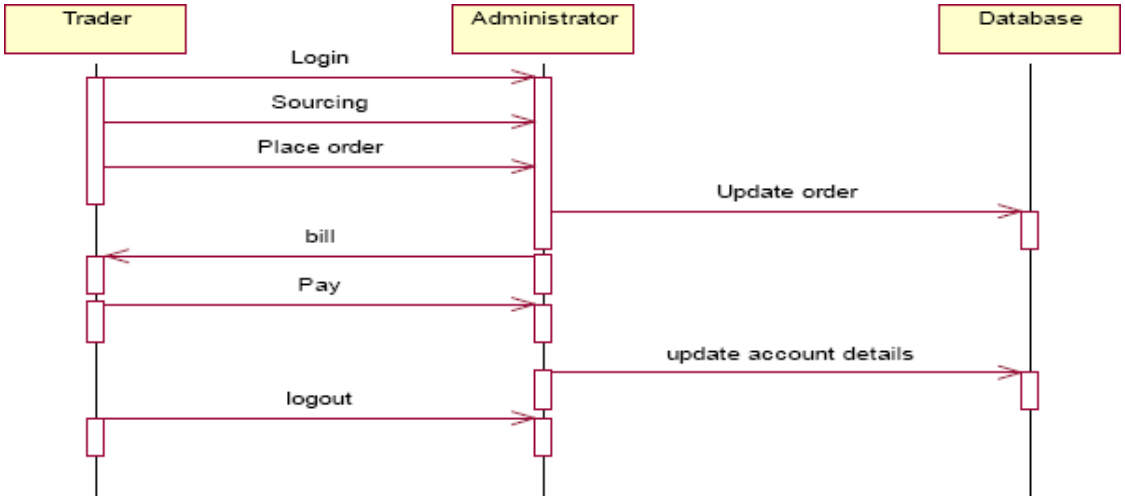


SEQUENCE DIAGRAM

A sequence diagram in unified modeling language is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. This diagram shows a parallel vertical lines called lifelines. There are two dimensions in this diagram

1. Vertical dimension-represents time.

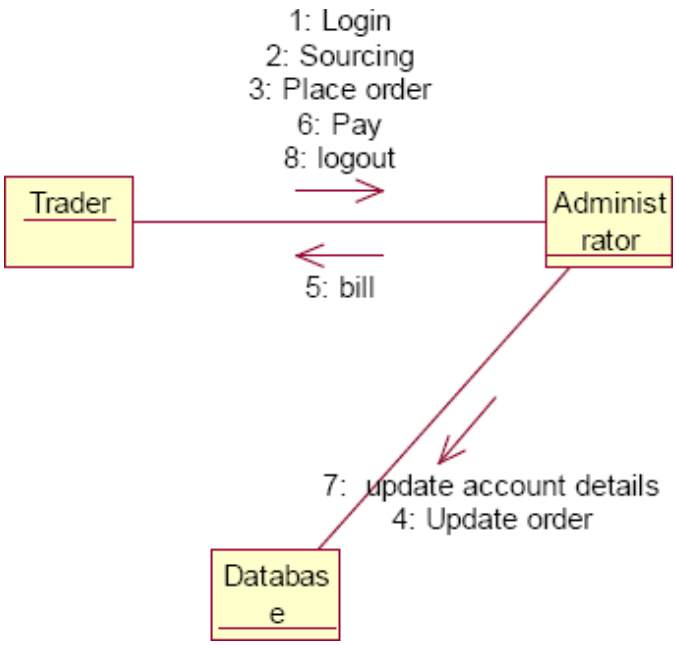
2. Horizontal dimension-represent different object



TraderTraderAdministratorAdministratorDatabaseDatabaseLoginSourcingPl ace orderUpdateorderPay update account details bill logout

COLLABORATION DIAGRAM

A collaboration diagram belongs to a group of UML diagrams called Interaction Diagrams. collaboration diagrams, like sequence diagrams, show how the objects interact over the course of time. collaboration diagrams show the sequence by numbering the messages on the diagram.



DOCUMENTATION OF COLLABORATION DIAGRAM

The collaboration diagram shows how the trader performs the sourcing and places order for which the administrator provides the bill and updates it in the database.

STATE CHART DIAGRAM

The state chart is used to model dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. So these diagrams are useful for reactive systems i.e.., a system that responds to external or internal events. It describes the flow of control from one state to other state. The initial state is represented using the small dot. The final state is represented using a circle surrounded by a small dot

DOCUMENTATION OF STATE CHART DIAGRAM

The state diagram represents the following states.

• The trader logins the register in the first state and performs sourcing in the second state.

• The trader places the order in the third state.

• The trader receives the bill in the fourth state and pay the required amount in fifth state.

• The trader logouts from the system in the sixth state

ACTIVITY DIAGRAM

This diagram represents the graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. It shows the overall flow of control.

DOCUMENTATION OF ACTIVITY DIAGRAM

This activity diagram represents the flow of stepwise activities performed in foreign trading system.

• The first action represents the trader logins to the system.

• The second action is the place where the trader places the order.

• The decision state is the state where the trader decides to place the order.

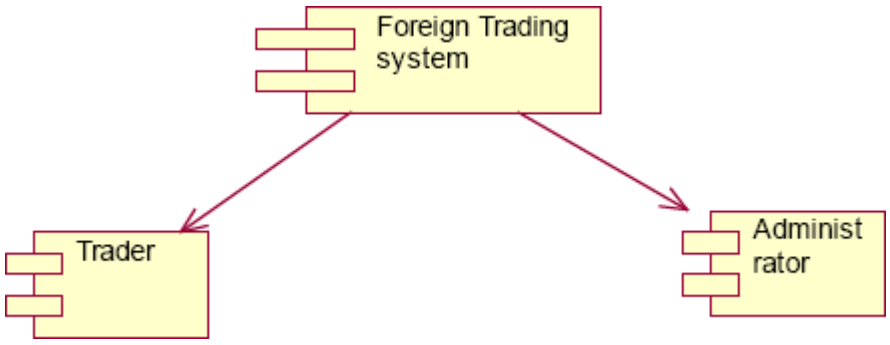
• If the trader places the order, fill the form for the required commodities.

• The next activity is that the administrator provides the bill for those commodities.

• The trader pays for the bill and logout from the system.

COMPONENT DIAGRAM

A component diagram depicts how the components are wired together to form larger components and or software systems. Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component.

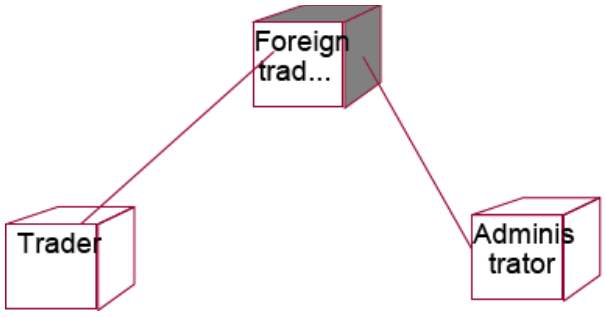


DOCUMENTATION OF COMPONENT DIAGRAM

The main component in the component diagram is foreign trading system. The trader who come to do the trading process and administrator who manages all the other processes is the sub components.

DEPLOYMENT DIAGRAM

A deployment diagram models the physical deployment of artifacts on nodes. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes.



DOCUMENTATION OF DEPLOYMENT DIAGRAM

The processor in this diagram is the foreign trading system. The devices are the trader and administrator who perform the main activities in the system.

PACKAGE DIAGRAM

A package diagram in the unified modeling language depicts the dependencies between the packages that make up a model. It provides a way to group the elements. There are three types of layers in package diagram. They are

- User interface layer
- Domain layer
- Technical services layer

User interface layer

The user interface layer may call upon its directly subordinate application logic layer, and also upon elements of a lower technical service layer, for logging and so forth.

Domain layer

Software objects representing domain concepts (for example, a software class administrator) that fulfill application requirements, such as tracing order and providing the bill.

Technical services layer

General purpose objects and subsystems that provide supporting technical services, such as interfacing with a database or error logging.These services are usually application-independent.

DOCUMENTATION OF PACKAGE DIAGRAM

The three layers in the foreign trading system are

- **User interface layer –** consists of web and login. This layer describes how the trader logins to the website and trades for the commodities.
- **Domain layer –** shows the activities that are performed inside the trading system. The activities are place order, pay for the bill and logouts.
- **Technical service layer –** The sourcing and updating the details are performed in this layer.

RESULT

Thus the mini project for foreign trading system has been successfully executed and codes are generated.

Ex. No:12 CONFERENCE MANAGEMENT SYSTEM

Date:

AIM

To develop a project on Conference management system using Rational Rose Software.

(I)PROBLEM STATEMENT

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

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

PURPOSE

The purpose of the conference management system is that the system can easily review the process. The main process in this document is the submission of paper by the candidate, reviewing process by the reviewer and sending of acknowledgement to the candidates whose paper is selected.

SCOPE

The scope of this conference management process is to select the best candidate from the list of candidates based on their performance in the process.

FUNCTIONALITY

The main functionality of conference system is to select the candidate for the presentation in conference.

USABILITY

The user interface to make the process should be effective that is the system will help the candidates to register easily. The system should be user friendly.

PERFORMANCE

It describes the capability of the system to perform the conference process of the candidate without any error and performing it efficiently.

RELIABILITY

The conference system should be able to serve the applicant with correct information and day-to-day update of information.

FUNCTIONAL REQUIREMENTS

Functional requirements are those that refer to the functionality of the system that is the services that are provided to the candidate who register for the conference.

UML DIAGRAMS

The following UML diagrams describe the process involved in the conference management system.

USE CASE DIAGRAM

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using ellipse. Actor is any external entity that makes use of the system being modeled. It is represented using stick figure.

DOCUMENTATION OF USE CASE DIAGRAM

The actors in this use case diagram are candidate, reviewer and database. The use cases are the activities performed by actors.
The actors in this use case diagram are

• **Candidate** - Logins the conference system and submits the paper then do the registration process.

• **Reviewer** – Review the paper , select best candidate and send acknowledgement to them.

• **Databases** - verify the login and register details and selected candidate details are stored in it.

The use cases in this use case diagram are

• **Login** - Candidate enter their username and password to login to the conference system.

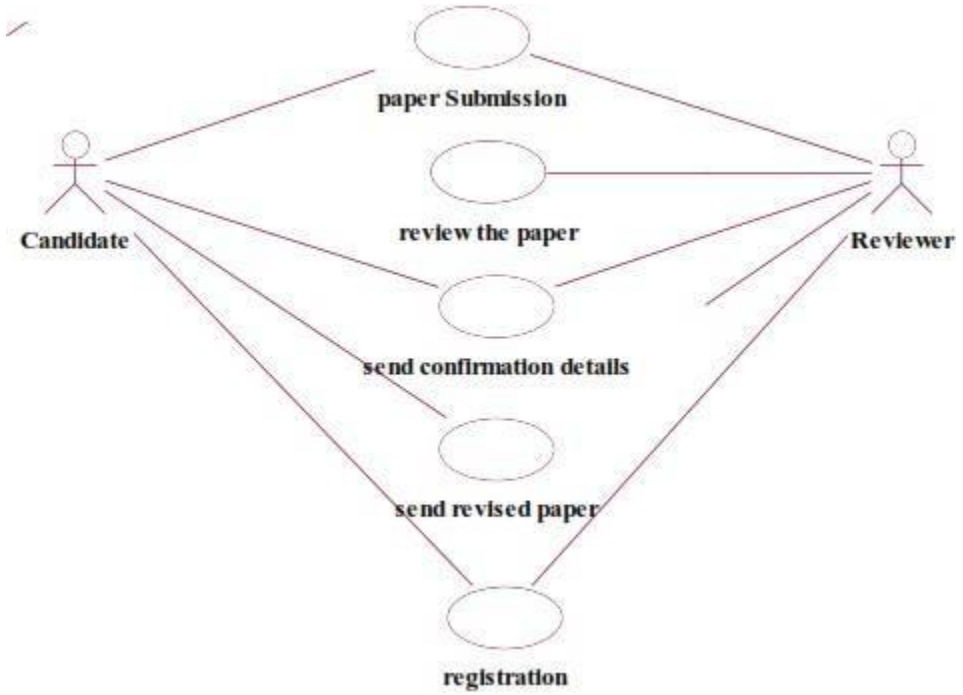
Paper submission– Candidate submits the paper.

• **Review the paper**– The paper is been reviewed by the reviewer and the paper is selected.

• **Paper confirmation details –** The reviewer can send the confirmation details to the candidate.

• **Revised and camera ready paper** – After the paper is selected and the camera ready paper should be submitted to the reviewer by candidate.

• **Registration** – After submitting the revised paper the candidate wants to register.



CLASS DIAGRAM

A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment have the class name, middle compartment the attributes and the bottom compartment with operations.

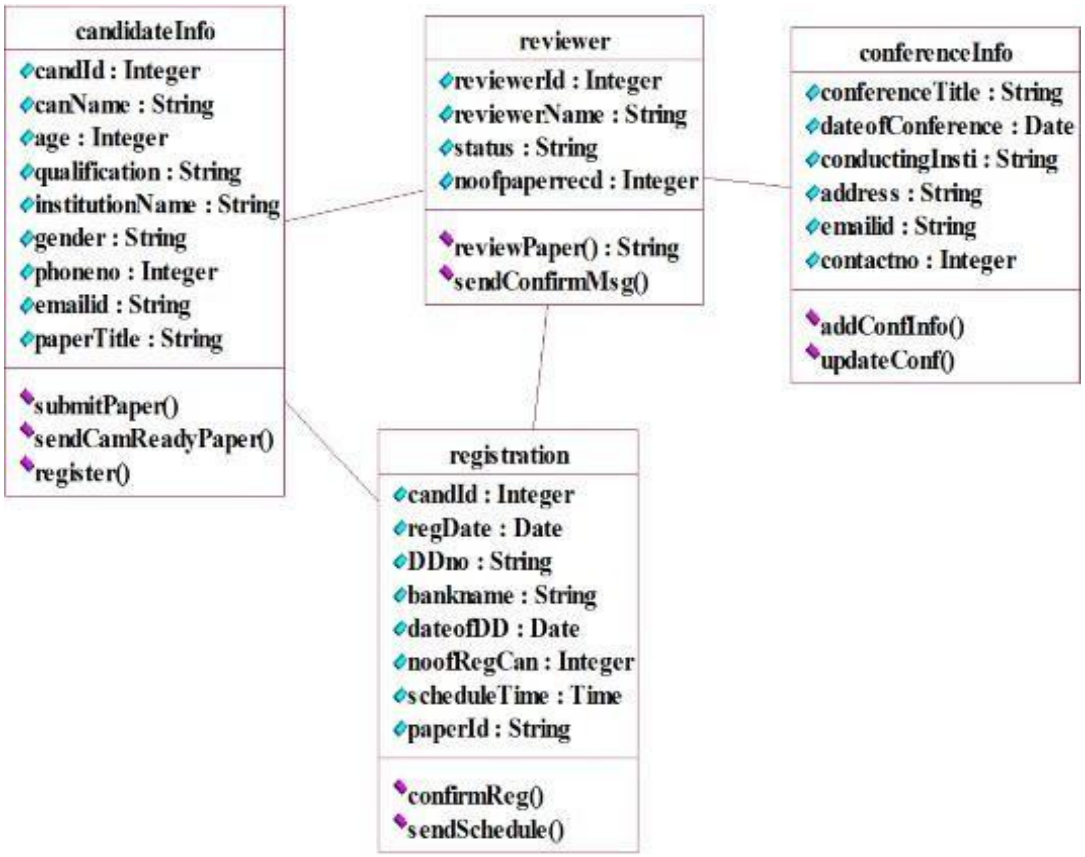
DOCUMENTATION OF CLASS DIAGRAM

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

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

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

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

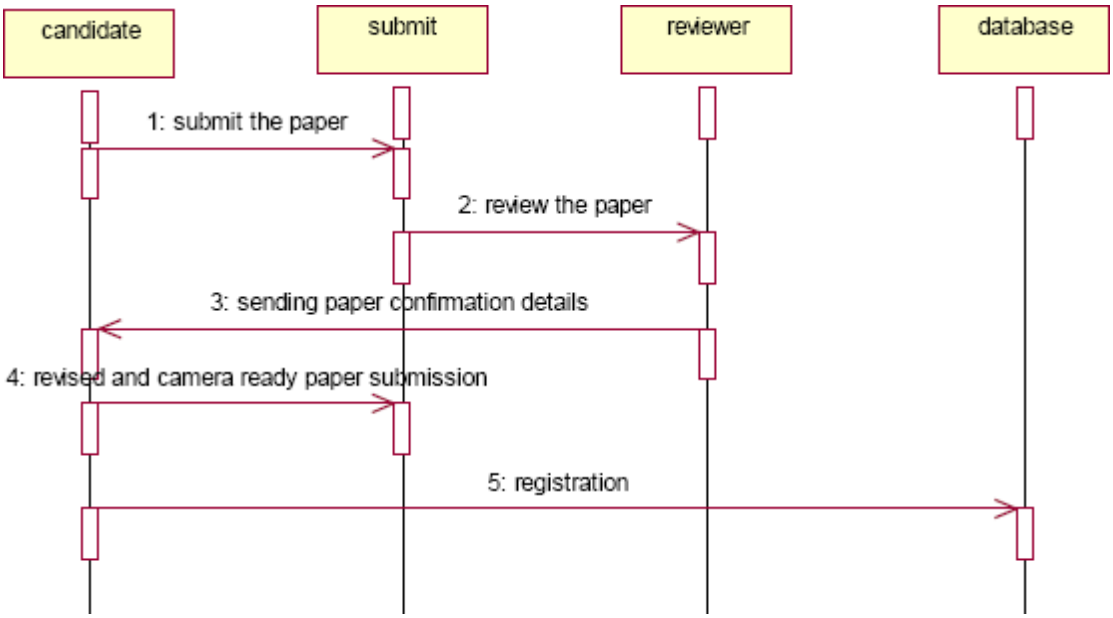


SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

1. Vertical dimension-represent time.
2. Horizontal dimension-represent different objects.

SEQUENCE DIAGRAM FOR SUBMITTING PAPER



DOCUMENTATION OF SEQUENCE DIAGRAM

LOGIN

This sequence diagram describes the sequence of steps to show

• The candidate login in to the conference system and register for job.

• The verification done in the database .

PAPER SUBMISSION

This sequence diagram shows steps to show

• The candidate submit the paper.

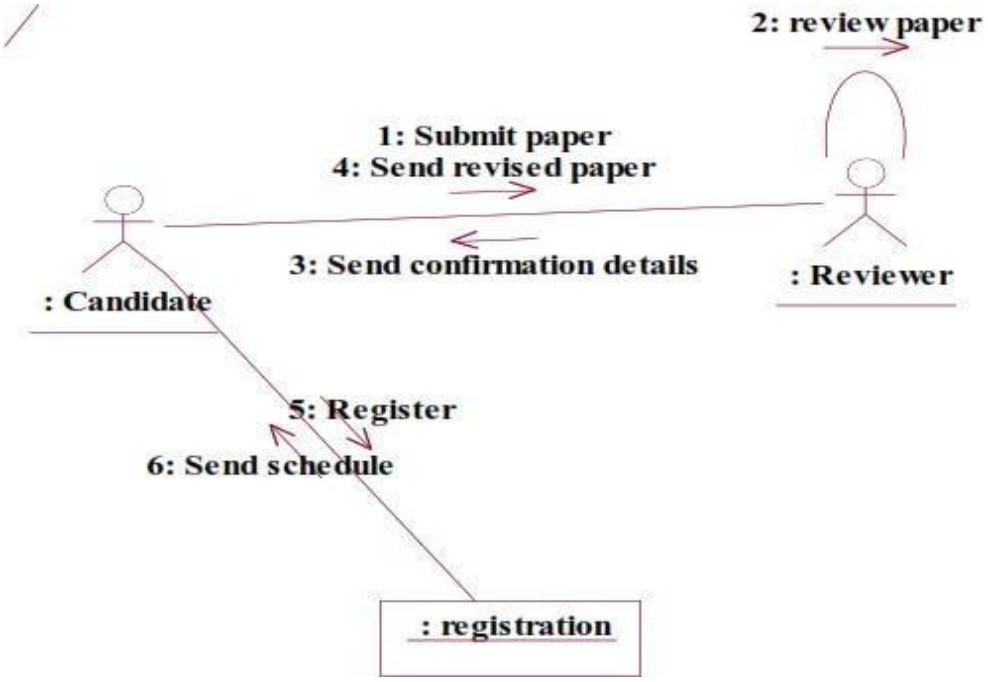
• The reviewer reviews the paper and sends acknowlegement to the candidate.

• The candidate submits revised and camera ready paper.

• This candidate will registers their details.

COLLABRATION DIAGRAM

A collaboration diagram, also called a communication diagram or interaction diagram.A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time.



DOCUMENTATION OF COLLABRATION DIAGRAM

LOGIN

This collaboration diagram is to show how the applicant login in the conference system. Here the sequence is numbered according to the flow of execution.

PAPER SUBMISSION

This collaboration diagram is to show the submitting paper process of the candidate for the conference. The flow of execution of this selection process is represented using the numbers.

STATE CHART DIAGRAM

The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle surrounding a small dot.

DOCUMENTATION OF STATE CHART DIAGRAM

This state diagram describes the behaviour of the system.

• First state is login where the candidate login to the conference system.

• The next state is submitting the paper .

• Then review the paper if it is selected the process will continue..

• The candidate should submit revised and camera ready paper.

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control. An activity is shown as an rounded box containing the name of the operation.

DOCUMENTATION OF ACTIVITY DIAGRAM

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

• First the candidate login to the database.

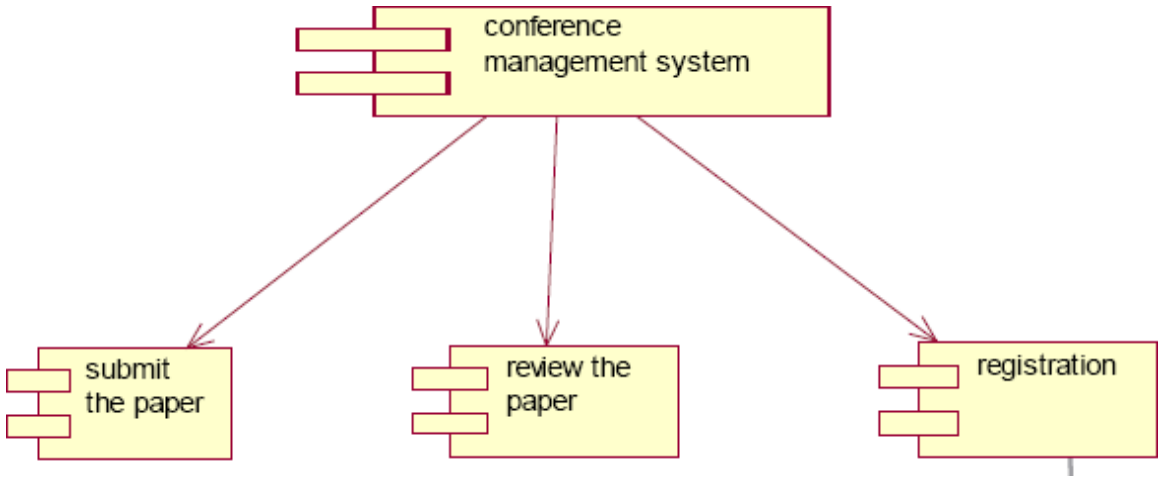
• Then the candidate should submit the paper.

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

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

COMPONENT DIAGRAM

The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association.

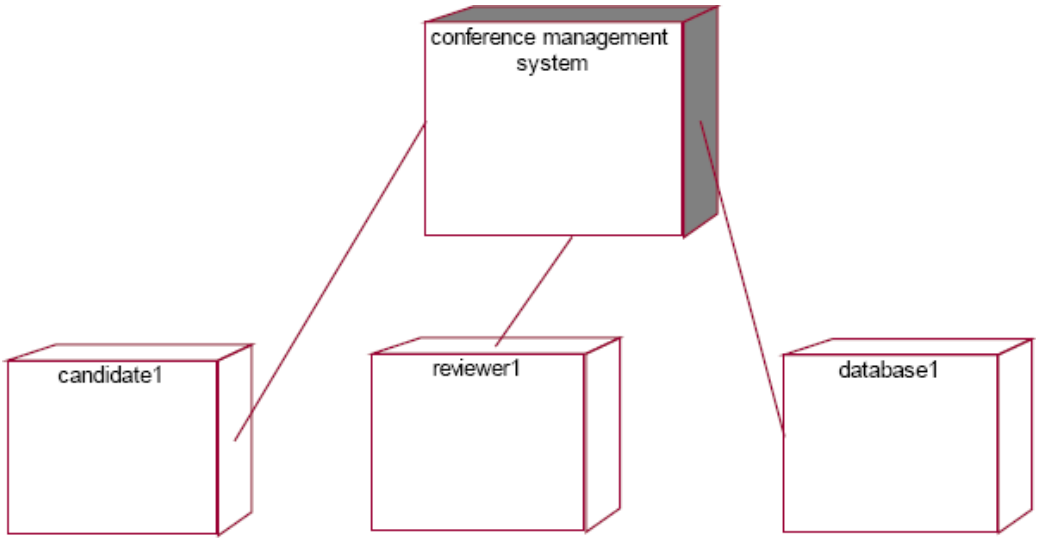


DOCUMENTATION OF COMPONENT DIAGRAM

The main component in this component diagram is conference management system. And submit the paper, review the paper and registration.

DEPLOYMENT DIAGRAM

A deployment diagram in the unified modeling language serves to model the physical deployment of artifacts on deployment targets. Deployment diagrams show "the allocation of artifacts to nodes according to the Deployments defined between them. It is represented by 3-dimensional box. Dependencies are represented by communication association.



DOCUMENTATION OF DEPLOYMENT DIAGRAM

The processor in this deployment diagram is the conference management system which is the main part and the devices are the candidate, appear for do conference , reviewer will reviews paper , database will store all details which are the some of the main activities performed in the system.

PACKAGE DIAGRAM

A package diagram in unified modeling language that depicts the dependencies between the packages that make up a model. A Package Diagram (PD) shows a grouping of elements in the OO model, and is a Cradle extension to UML. PDs can be used to show groups of classes in Class Diagrams (CDs), groups of components or processes in Component Diagrams (CPDs), or groups of processors in Deployment Diagrams (DPDs).

DOCUMENTATION OF PACKAGE DIAGRAM

The three layers in the online recruitment system are

• **The User interface layer** - consists of the web and login. This layer describes how the candidate login.

• **The Domain layer** – shows the activities that are performed in the conference management system. The activities are paper submission , review paper , registration.

• **The Technical service layer -** the verification details and the selected candidate details will stored into the database.

RESULT

Thus the mini project for Conference management system has been successfully executed and codes are generated.

Ex.No: 13 **BPO MANAGEMENT SYSTEM**
Date:

AIM:

To implement a software for BPO management system

(I) PROBLEM STATEMENT:

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

(II)USECASE DIAGRAM:

Fig.3. UML USE CASE DIAGRAM

(IV) UML CLASS DIAGRAM:

The UML class diagram is to illustrate class interfaces and their actions. They are used for static object modeling, we have already introduced and used their UML diagram while domain modeling.

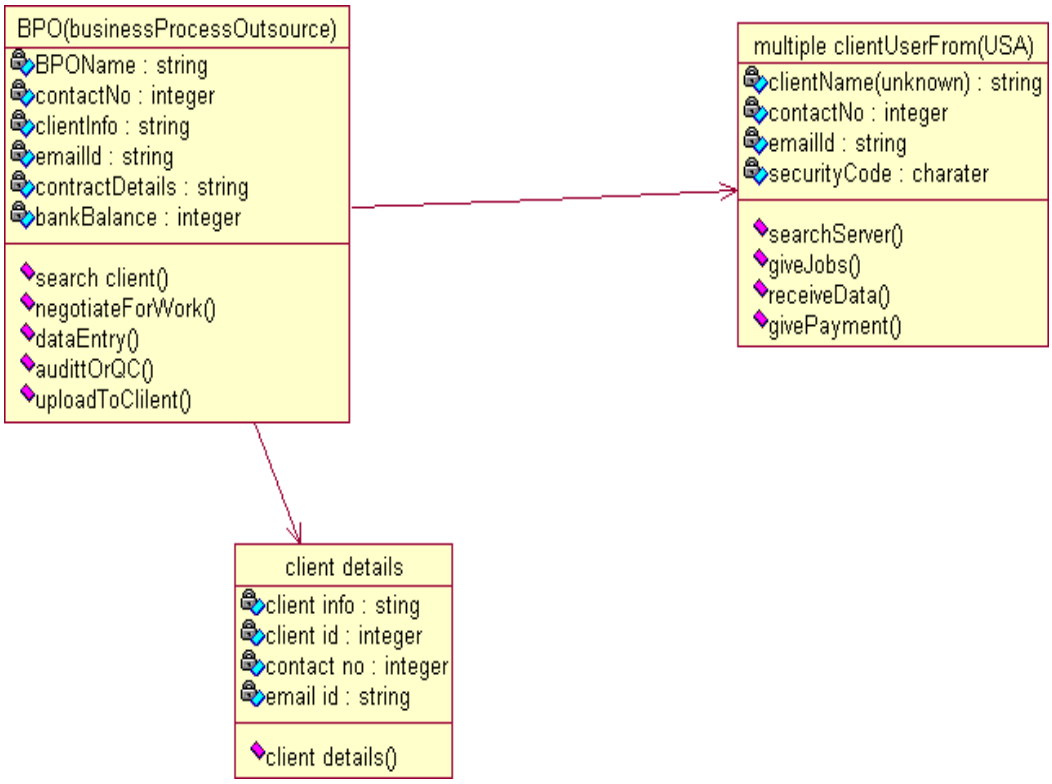


Fig.5. UML CLASS DIAGRAM

(V) UML SEQUENCE DIAGRAM:

A sequence diagram illustrates a kind of format in which each object interacts via message. It is generalize between two or more specialized diagram.

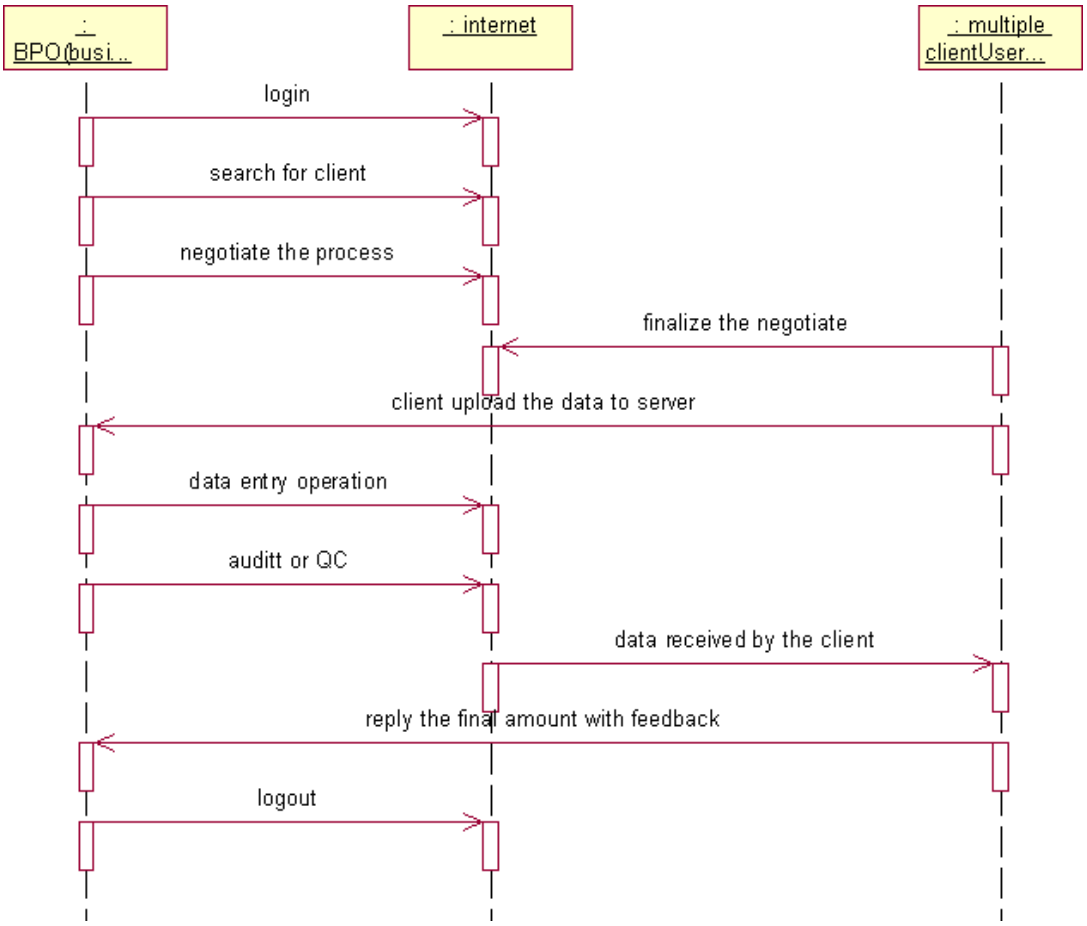


Fig. 5.1SEQEUENCE DIAGRAM

Communication diagram illustrate that object interact on a graph or network format in which object can be placed where on the diagram. In collaboration diagram the object can be placed in anywhere on the diagram. The collaboration comes from sequence diagram.

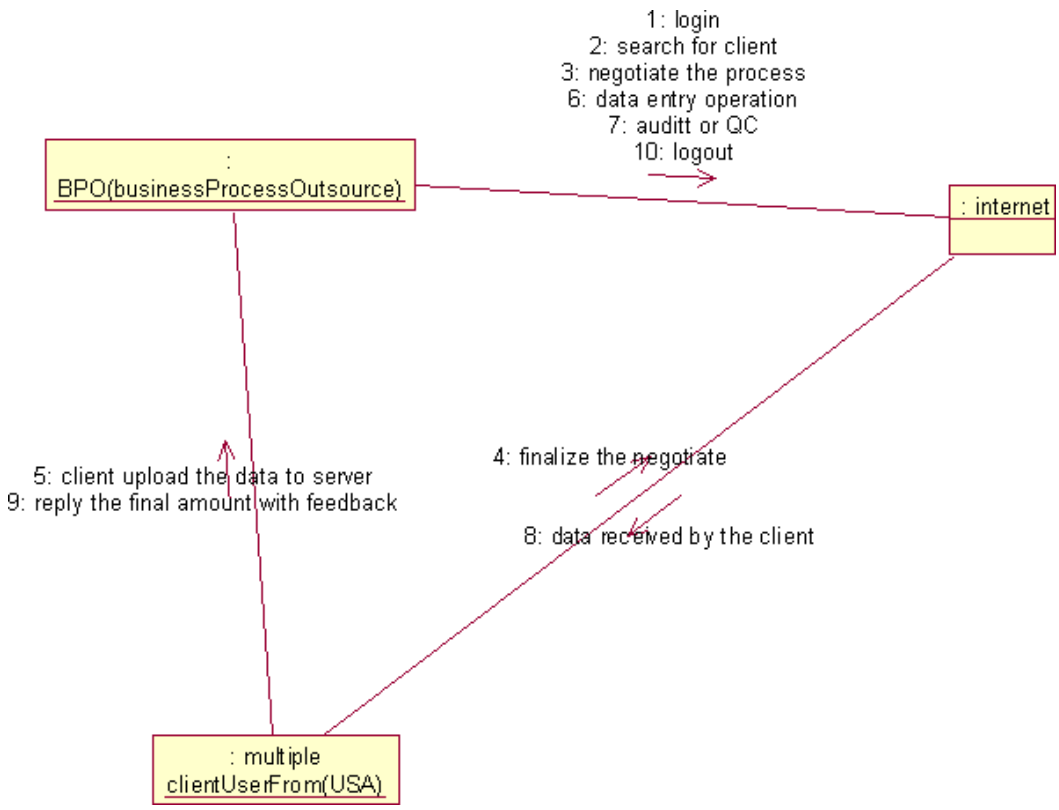
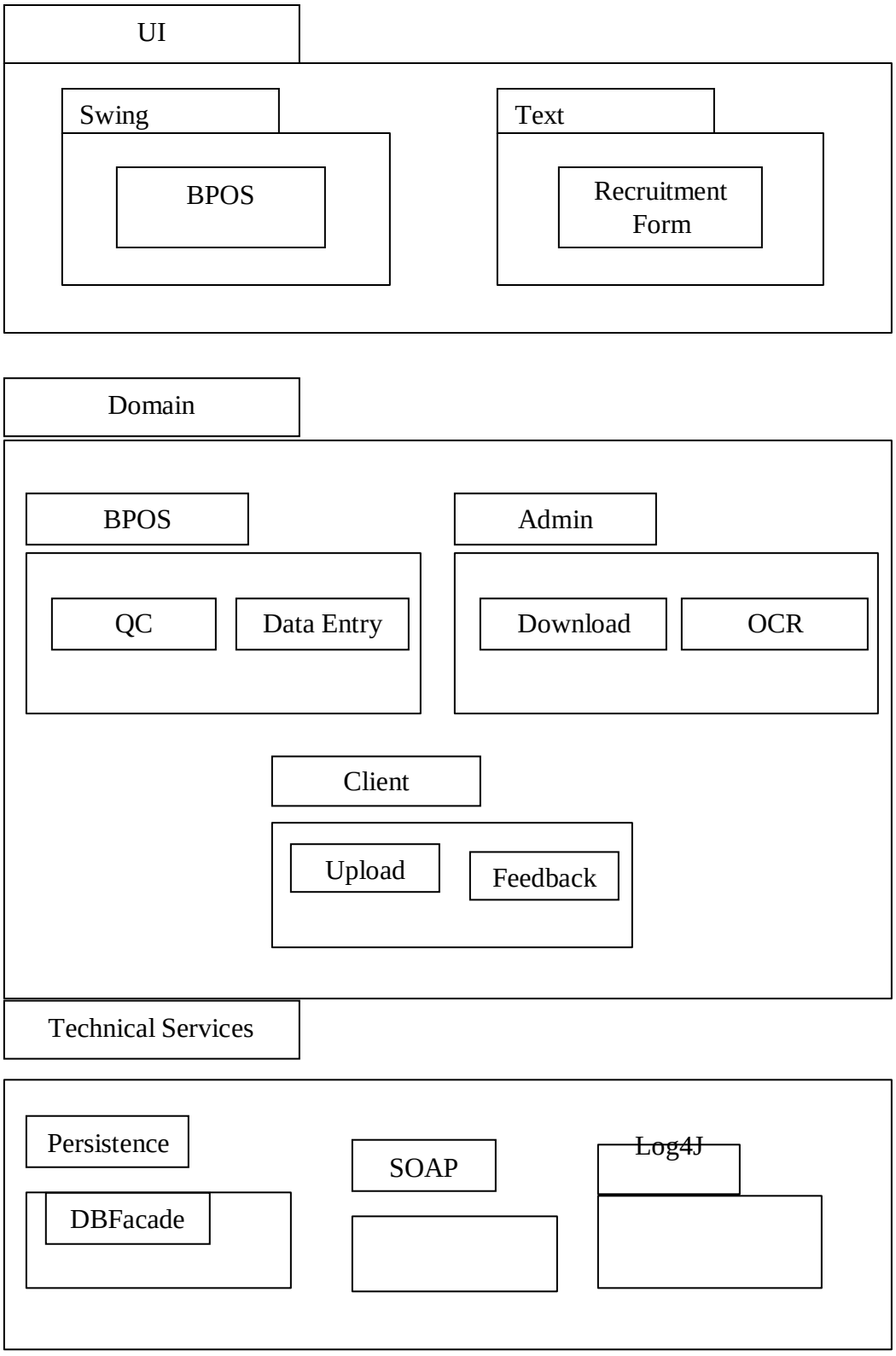


Fig.5.2COLLABRATION DIAGRAM

(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

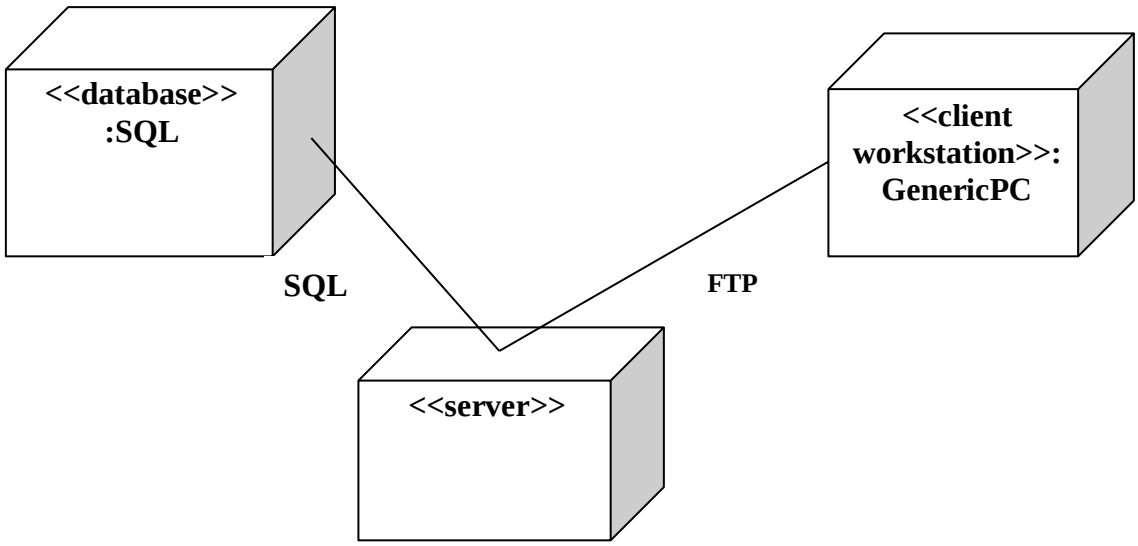
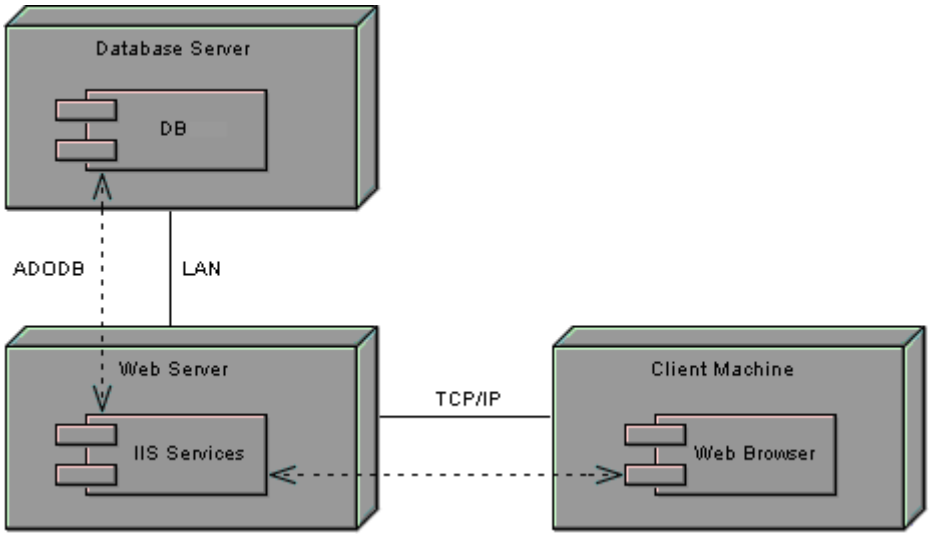


Fig.9.1.DEPLOYMENT DIAGRAM

Component Diagram

Component diagrams are used to visualize the organization and relationships among components in a system.



RESULT :

Thus the mini project for BPO management system has been successfully executed and codes are generated.

Ex.No: 14 **LIBRARY MANAGEMENT SYSTEM**
Date:

AIM

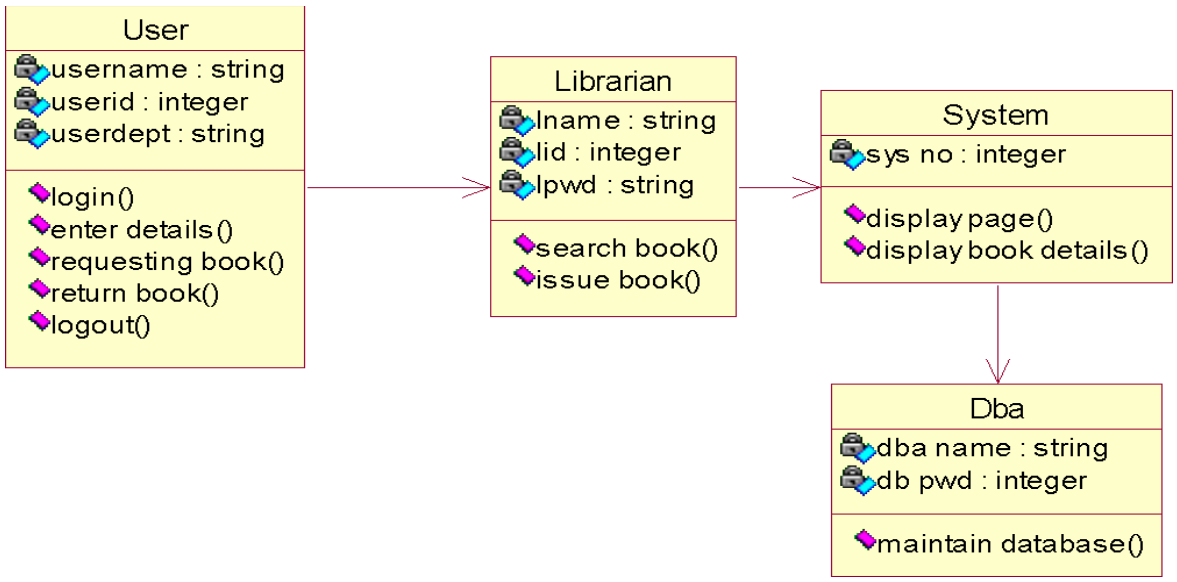
To design an object oriented model for Library Management System using Rational Rose software and to implement it using Java.

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.

CLASS DIAGRAM

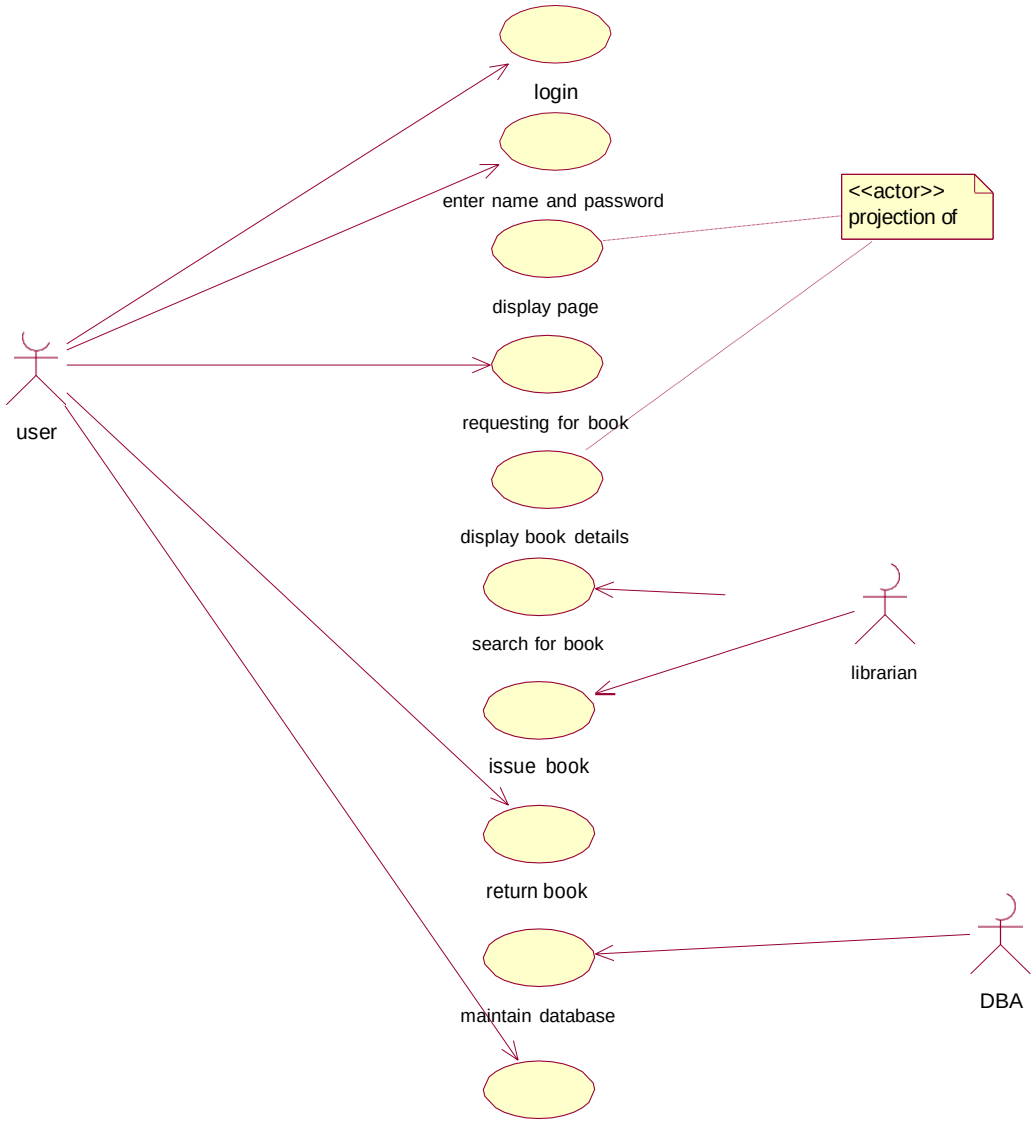
A class diagram in the unified modeling language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects. The library management system makes use of the following classes user,



librarian, system and DBA.

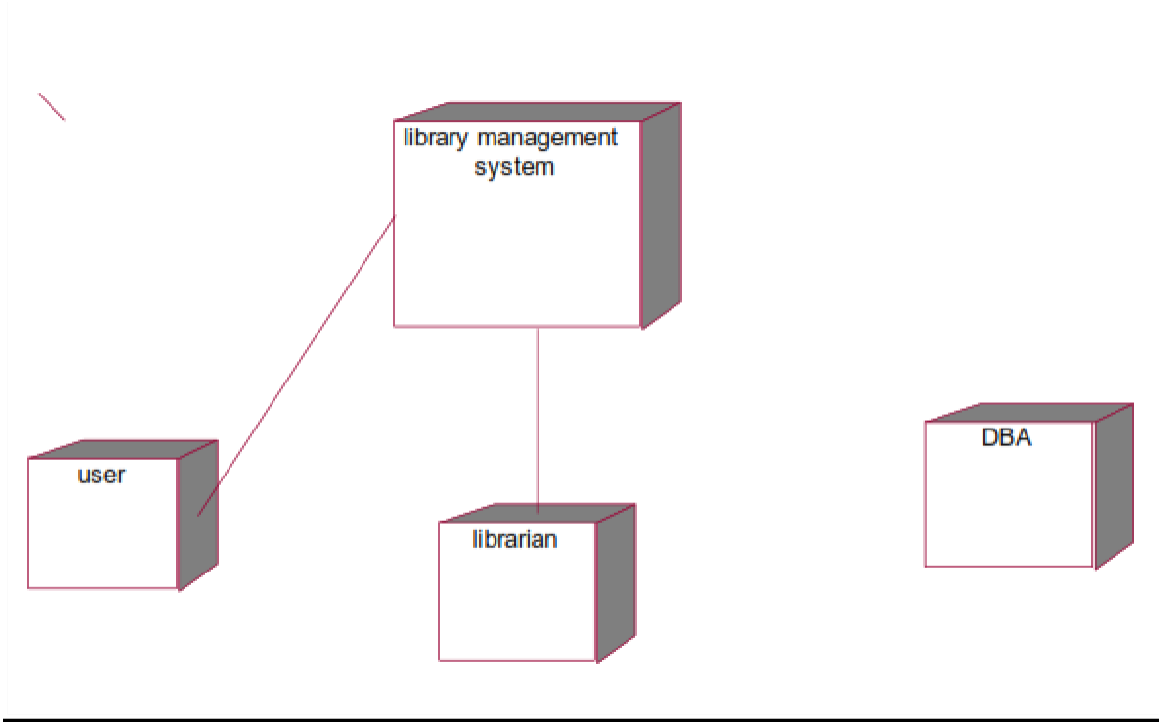
USE CASE DIAGRAM

Use case is a list of actions or events. Steps typically defining the interactions between a role and a system to achieve a goal. The use case diagram consists of various functionality performed by actors like user, librarian, system and DBA.



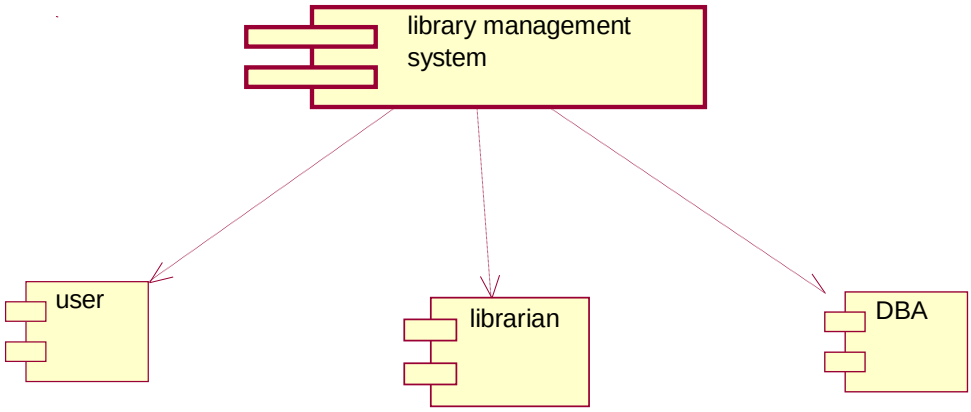
DEPLOYMENT DIAGRAM

Deployment diagram is a structure diagram which shows architecture of the system as deployment of software artifacts to deployment target. It is the graph of nodes connected by communication association. It is represented by three dimensional box. The device node is library management system and execution environment nodes are user, librarian, system and DBA.



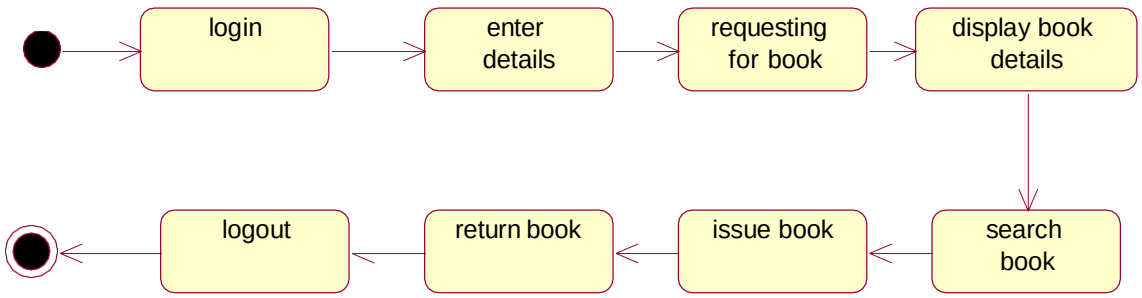
COMPONENT DIAGRAM

Component diagram shows the dependencies and interactions between software components. Component diagram carries the most important living actors of the system i.e, user, librarian and DBA.



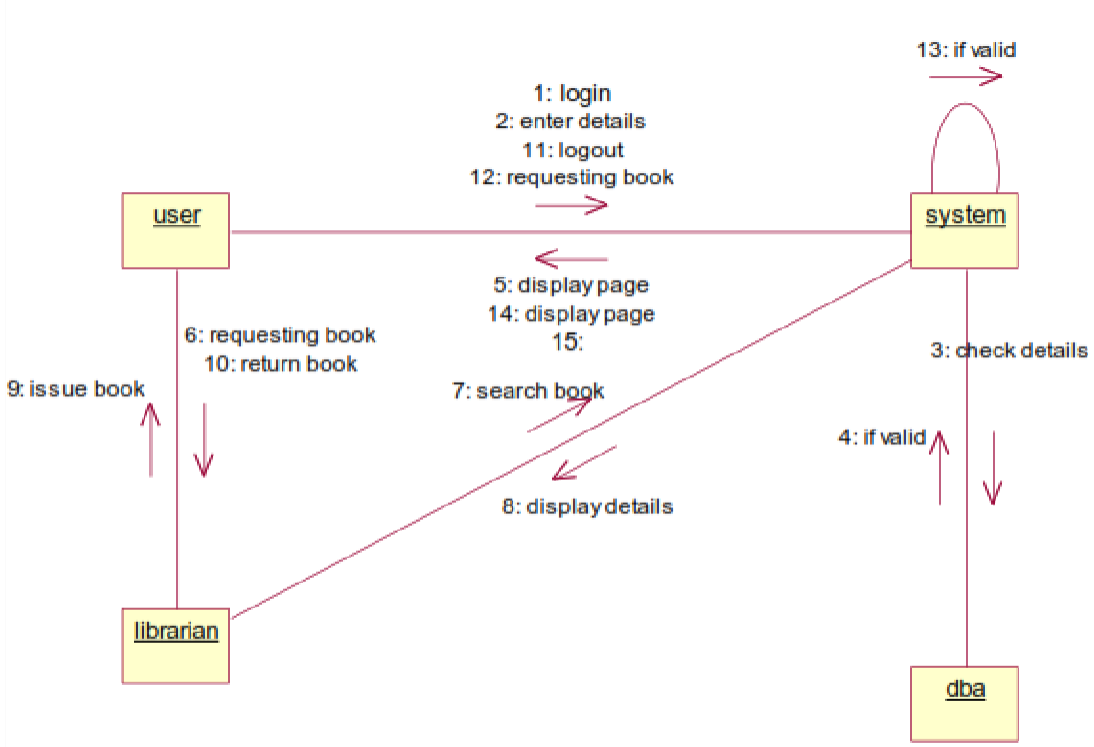
STATECHART DIAGRAM

State chart diagram is also called as state machine diagram. The state chart diagram contains the states in the rectangular boxes and the states are indicated by the dot enclosed. The state chart diagram describes the behavior of the system. The state chart diagram involves eight stages such as login, enter details, requesting for book, display book details, search book, issue book, return book and logout.



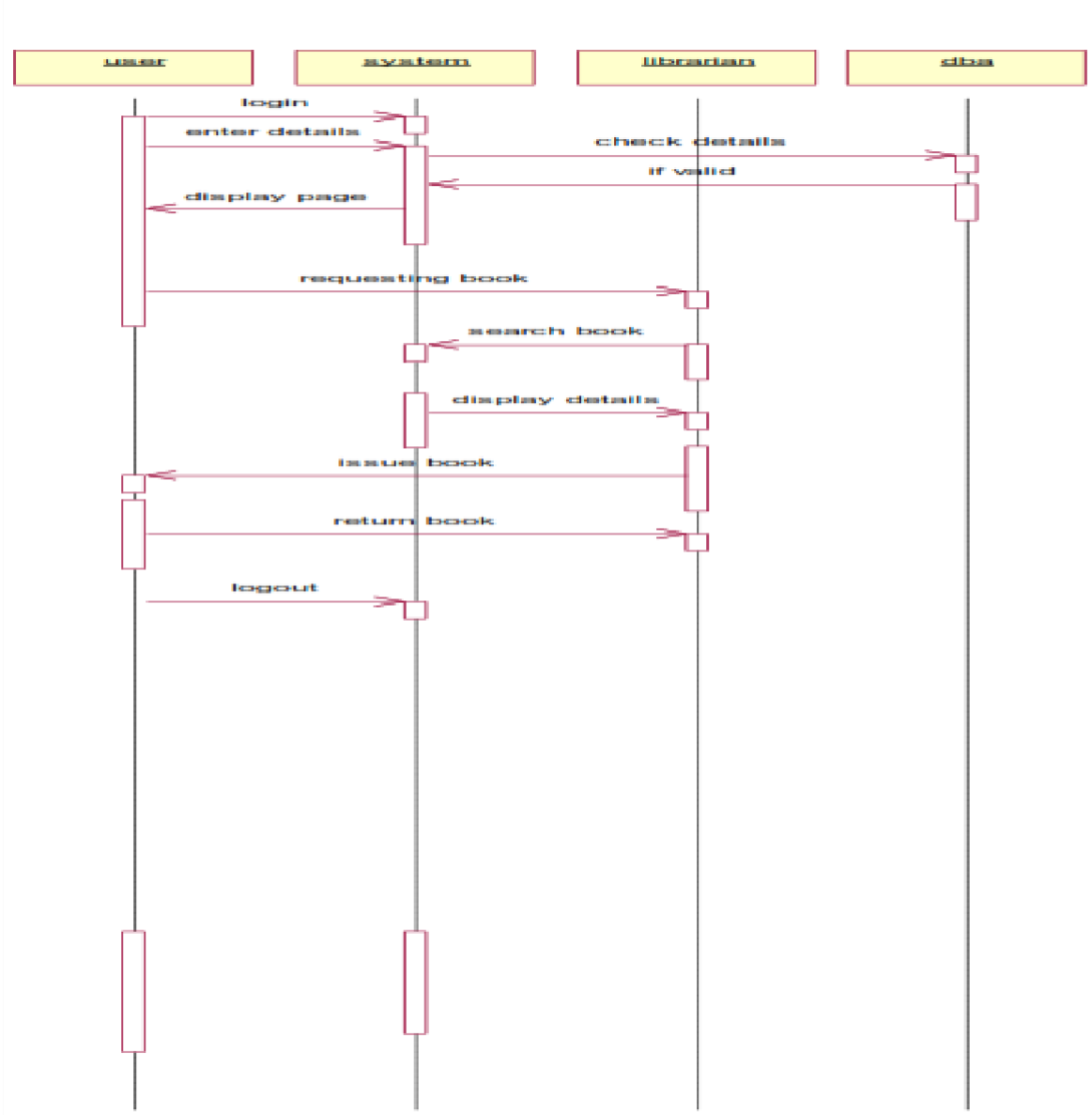
COLLABORATION DIAGRAM

Like sequence diagram collaboration diagrams are also called as interaction diagram. Collaboration diagram convey the same informations as sequence diagram but focus on the object roles instead of the times that messages are sent. Here the actions between various classes are represented by number format for the case of identification.



SEQUENCE DIAGRAM

A sequence diagram represent the sequence and interactions of a given use case or scenario. Sequence diagram capture most of the information about the system. It is also represent in order by which they occur and have the object in the system send message to one another. Here the sequence starts with interaction between user and the system followed by database. Once the book have been selected the next half of sequence starts between librarian and user followed by database.



ACTIVITY DIAGRAM

Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Here in the activity diagram the user login to the system and perform some main activity which is the main key element to the system.

RESULT

Thus the various UML diagrams for library management system was drawn and the code was generated successfully.

Ex.No:15 **STUDENT INFORMATION SYSTEM**
Date:

AIM

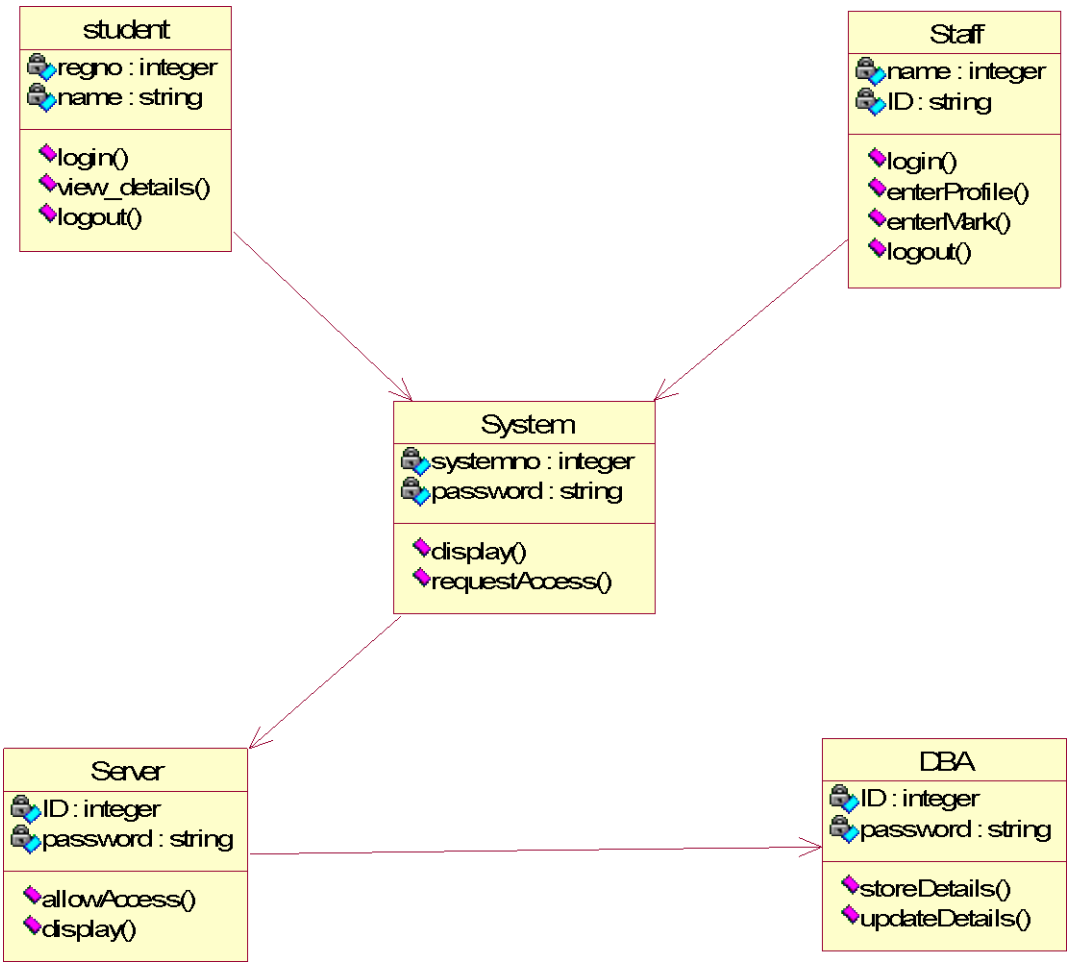
To design an object oriented model for Student information system using Rational Rose software.

PROBLEM STATEMENT

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

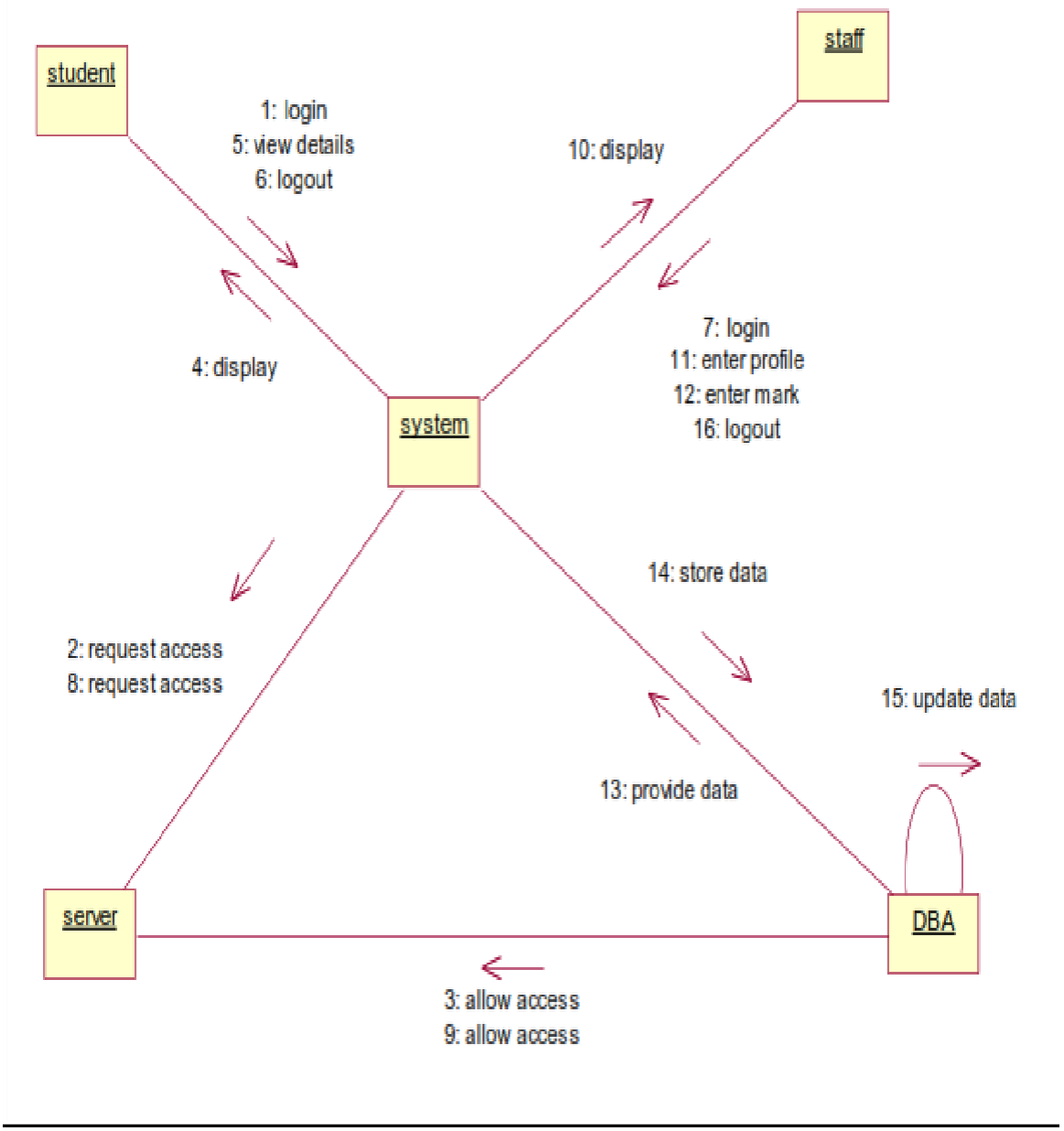
CLASS DIAGRAM

The class diagram is the graphical representation of all classes used in the system. The class diagram is drawn as rectangular box with three components or compartments like class name, attributes and operations. The student information system makes use of the following classes like student, staff, system, DBA and server.



COLLABORATION DIAGRAM

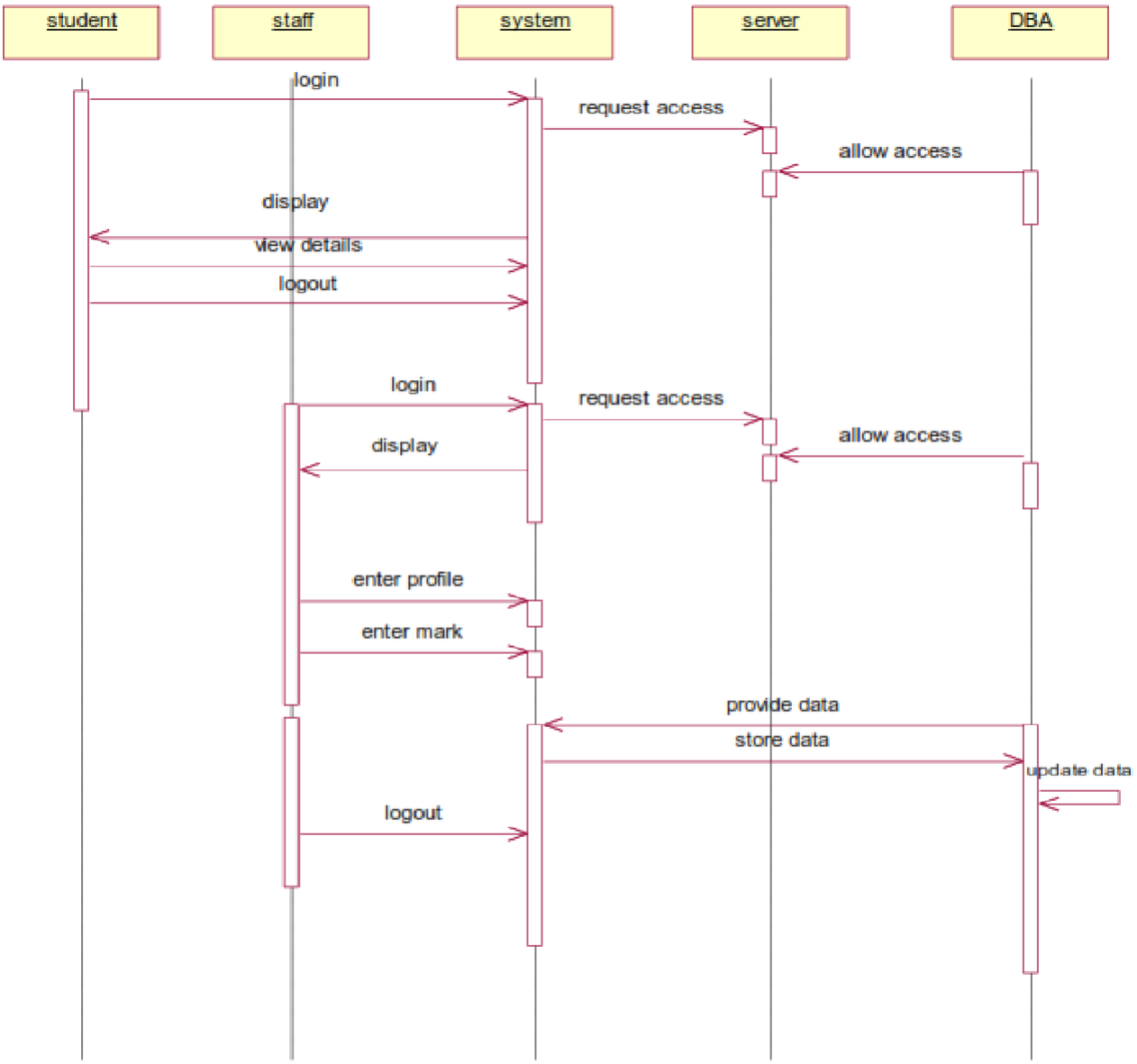
A Collaboration diagram represents the collaboration in which is a set of objects related to achieve a desired outcome. In collaboration, the sequence is indicated by numbering the message several numbering schemes are available. Login, request access, allow access, display, view details, logout, login, request access, allow access, display, enter profile, enter mark, provide data, logout, store data, update data.



SEQUENCE DIAGRAM

A Sequence diagram represent the sequence and interaction of a given usecase or scenario. Sequence diagram capture most of the information about the system. Here the sequence starts between the student and the system. The second half of interaction takes place between staff and system then by police and followed

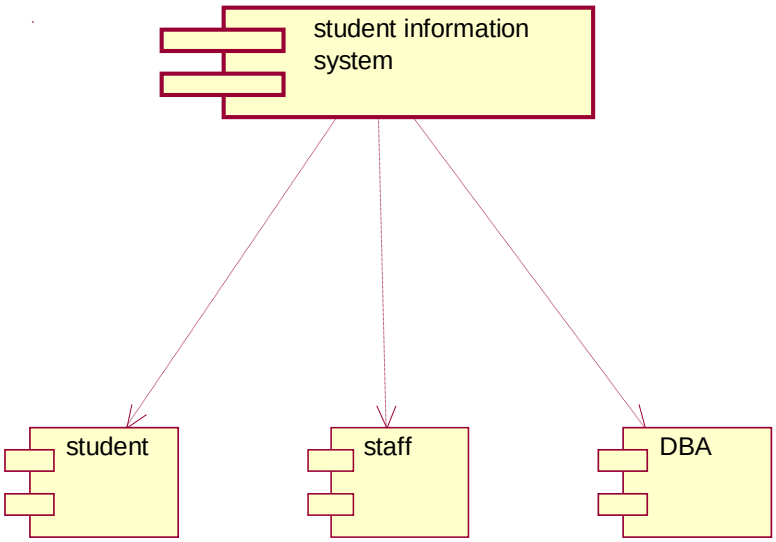
by database. The student first login to the system and then view the details of the details. Staff login to the system enter mark and enter the details of the student. DBA store and update the details of the student.



COMPONENT DIAGRAM

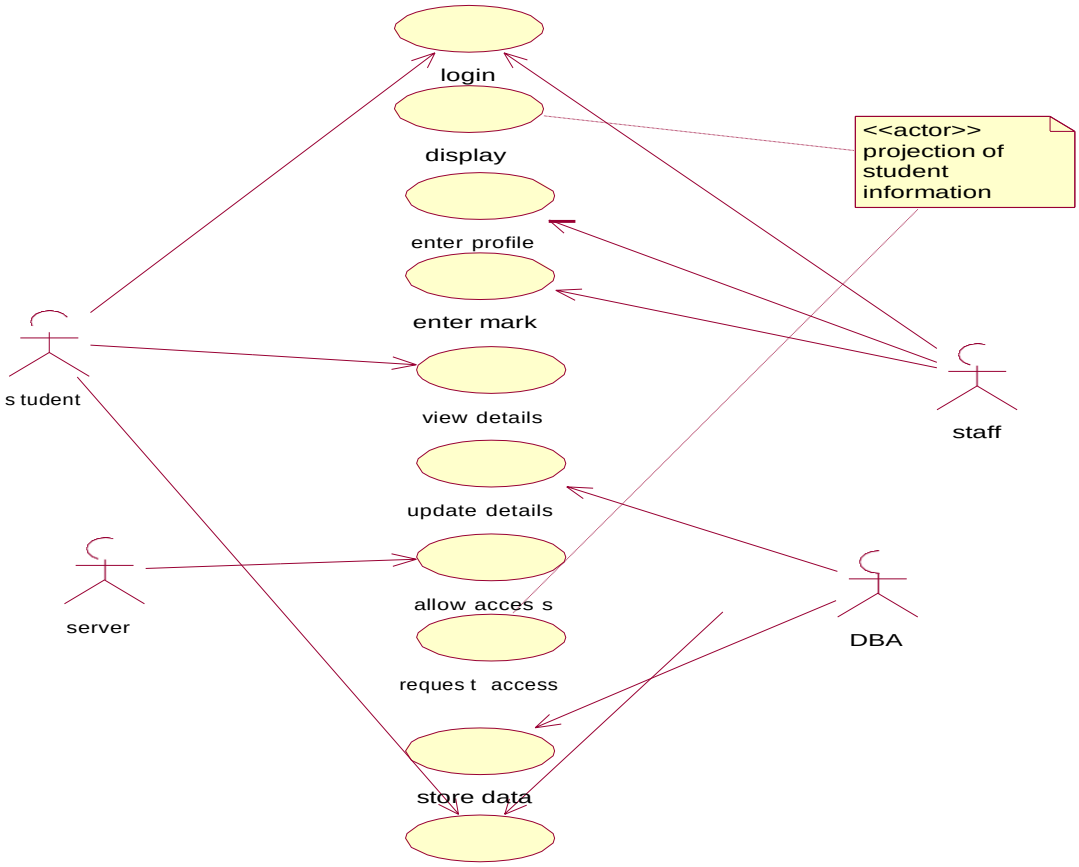
Component diagram carries the major living actors of the system. The component diagram main purpose is to show the

structural relationship between components of the system.The main component of the system is student information system and the other components of the system are student, staff and DBA.



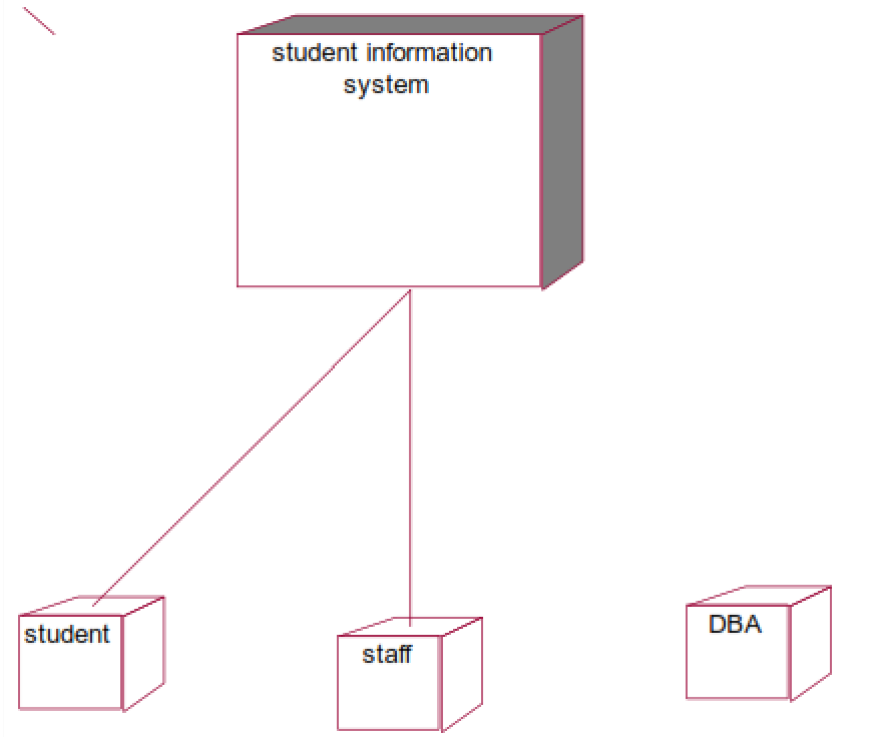
USE CASE DIAGRAM

Use case diagram is a graph of actors, a set of use cases, association between the actors and the use cases and generalization among the cases. Use case diagram is a list of actions or events. Use case diagram was drawn to represent the static design view of the system. Steps typically defined the interactions between a role and a system to achieve a goal. The use case diagram consists of various functionality performed by the actors like student, staff, system, DBA and server. The use case diagram consists of various functionality like login, display, enter profile, enter mark, view details, update details, allow access, request access, store details, logout.



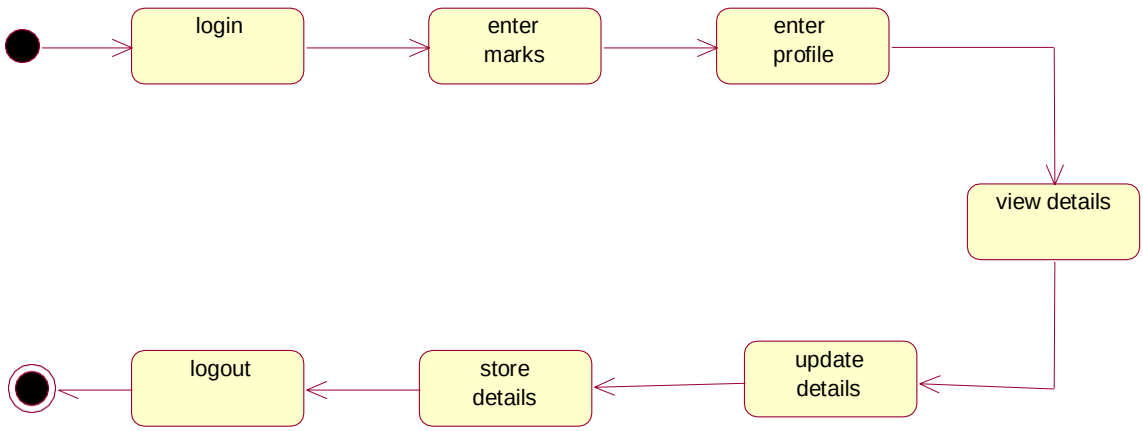
DEPLOYMENT DIAGRAM

Deployment diagram shows the configuration of runtime processing elements and the software components processes and objects that live in them. Component diagram are used in conjunction with deployment diagram to show how physical modules code are distributed on various hardware platform. The processor node in the system is student information system and the execution environment nodes or device nodes are student, staff and DBA.



STATE CHART DIAGRAM

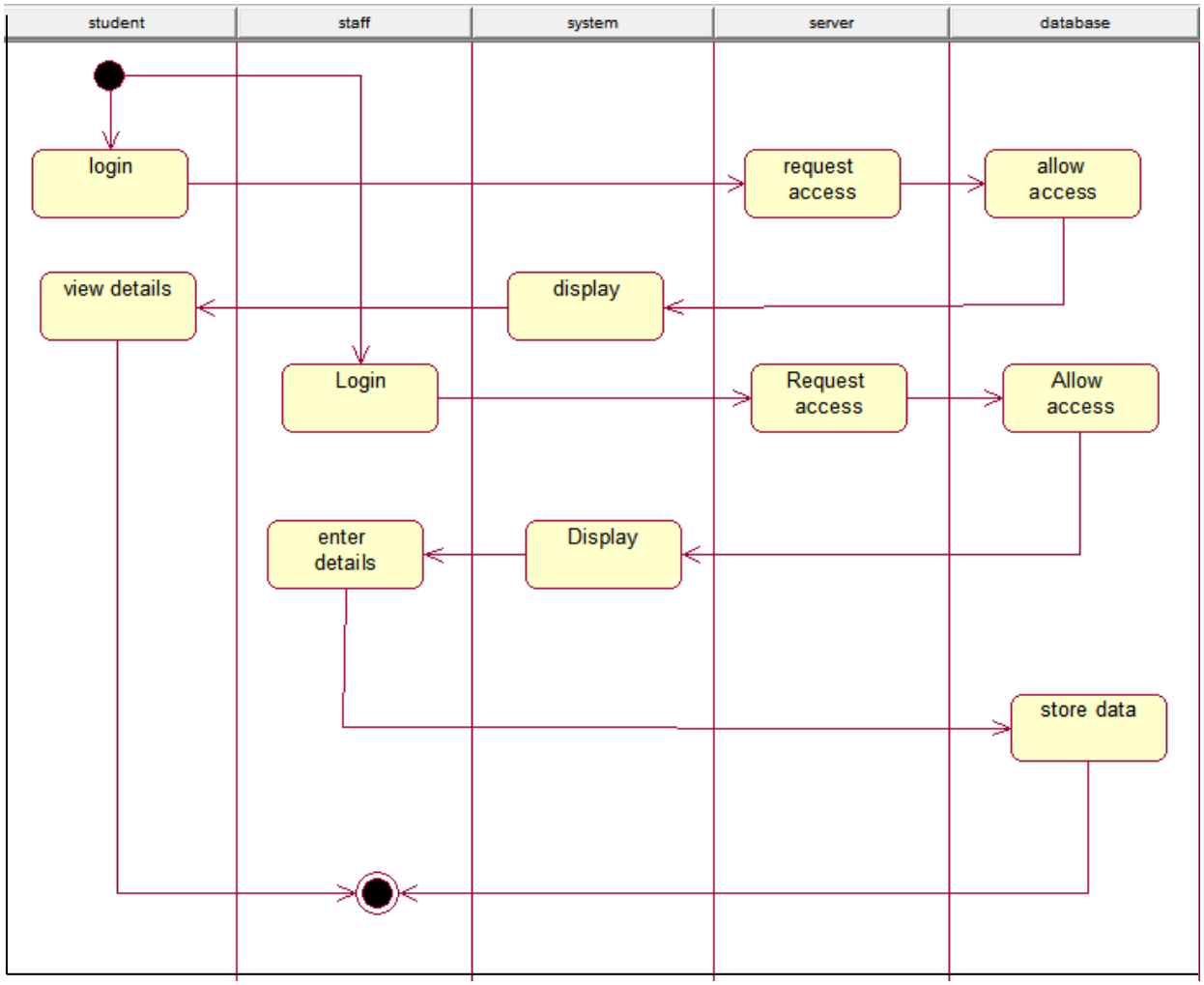
A State chart diagram is also called as state machine diagram. The state chart contains the states in the rectangular boxes and the states are indicated by the dot enclosed. The state chart diagram describes the behavior of the system. The state chart involves six stages such as login, enter mark, enter profile, view details, provide details, update details, store details and logout.



ACTIVITY DIAGRAM

Activity diagram are graphical representation of stepwise

activities and actions with support for choice, interaction and concurrency. Here in the activity diagram the student login to the system and view the details of the student. The staff login to the system for entering the student details and update the details in the database. The final interaction is the DBA store the details of the student.



RESULT

Thus the various UML diagrams for student information system were drawn and code was generated successfully.