ST. ANNE'S
COLLEGE OF ENGINEERING AND TECHNOLOGY
ANGUCHETTYPALAYAM, PANRUTI – 607 110

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

CS6511 – CASE TOOLS LABORATORY

Regulation 2013

Year / Semester: III / V

Jun 2018 – Dec 2018

PREPARED BY

Ms. S. VANATHI, M.Tech.,
Assistant Professor / CSE

CS6511 - CASE TOOLS LABORATORY

OBJECTIVE: To develop a mini-project following the 12 exercises listed below.
1. To develop a problem statement.
2. Develop an IEEE standard SRS document. Also develop risk management and project plan (Gantt chart).
3. Identify Use Cases and develop the Use Case model.
4. Identify the business activities and develop an UML Activity diagram.
5. Identity the conceptual classes and develop a domain model with UML Class diagram.
6. using the identified scenarios find the interaction between objects and represent them using UML Interaction diagrams.
7. Draw the State Chart diagram.
8. Identify the User Interface, Domain objects, and Technical services. Draw the partial layered, logical architecture diagram with UML package diagram notation.
9. Implement the Technical services layer.
10. Implement the Domain objects layer.
11. Implement the User Interface layer.
12. Draw Component and Deployment diagrams.

Suggested domains for Mini-project.
1. Passport automation system.
2. Book bank
3. Exam Registration
4. Stock maintenance system.
5. Online course reservation system
6. E-ticketing
7. Software personnel management system
8. Credit card processing
9. E-book management system
10. Recruitment system
11. Foreign trading system
12. Conference Management System
13. BPO Management System

TABLE OF CONTENTS

S.NO.	DATE	EXPERIMENT TITLE	MARKS/10	SIGN.
1		Case Tools		
2		Passport automation system		
3		Book bank		
4		Exam Registration		
5		Stock maintenance system		
6		Online course reservation system		
7		E-ticketing		
8		Software personnel management system		
9		Credit card processing		
10		E-book management system		
11		Recruitment system		
12		Foreign trading system		
13		Conference Management System		
14		BPO Management System		

Ex. No.: 1	
	CASE TOOLS
Date:	

INTRODUCTION:

CASE tools known as Computer-aided software engineering tools is a kind of component-based development which allows its users to rapidly develop information systems. The main goal of case technology is the automation of the entire information systems development life cycle process using a set of integrated software tools, such as modeling, methodology and automatic code generation. Component based manufacturing has several advantages over custom development. The main advantages are the availability of high quality, defect free products at low cost and at a faster time. The prefabricated components are customized as per the requirements of the customers. The components used are pre-built, ready-tested and add value and differentiation by rapid customization to the targeted customers. However the products we get from case tools are only a skeleton of the final product required and a lot of programming must be done by hand to get a fully finished, good product.

CHARACTERISTICS OF CASE:

Some of the characteristics of case tools that make it better than customized development are;
- ❖ It is a graphic oriented tool.
- ❖ It supports decomposition of process.

Some typical CASE tools are:
- ❖ Unified Modeling Language
- ❖ Data modeling tools, and
- ❖ Source code generation tools

INTRODUCTION TO UML (UNIFIED MODELING LANGUAGE):

The UML is a language for specifying, constructing, visualizing, and documenting the software system and its components. The UML is a graphical language with sets of rules and semantics. The rules and semantics of a model are expressed in English in a form known as OCL (Object Constraint Language). OCL uses simple logic for specifying the properties of a system. The UML is not intended to be a visual programming language.

However it has a much closer mapping to object-oriented programming languages, so that the best of both can be obtained. The UML is much simpler than other methods preceding it. UML is appropriate for modeling systems, ranging from enterprise information system to distributed web based application and even to real time embedded system. It is a very expensive language addressing all views needed to develop and then to display system even though understand to use. Learning to apply UML effectively starts forming a conceptual mode of languages which requires learning.

Three major language elements:
- ❖ UML basic building blocks
- ❖ Rules that dictate how this building blocks put together
- ❖ Some common mechanism that apply throughout the language

The primary goals in the design of UML are:
1. Provides users ready to use, expressive visual modeling language as well so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts.
7. Integrate best practices and methodologies.

Every complex system is best approached through a small set of nearly independent views of a model. Every model can be expressed at different levels of fidelity. The best models are connected to reality. The UML defines nine graphical diagrams:

1. Class diagram
2. Use-case diagram
3. Behavior diagram
   3.1. Interaction diagram
         3.1.1. sequence diagram
         3.1.2. collaboration diagram
   3.2. state chart diagram
   3.3. activity diagram
4. Implementation diagram

1. UML class diagram:

The UML class diagram is also known as object modeling. It is a static analysis diagram. These diagrams show the static structure of the model. A class diagram is a connection of static model elements, such as classes and their relationships, connected as a graph to each other and to their contents.

2. 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 generalization among the use-cases.

3. Behavior diagram:

It is a dynamic model unlike all the others mentioned before. The objects of an object oriented system are not static and are not easily understood by static diagrams. The behavior of the class's instance (an object) is represented in this diagram. Every use-case of the system has an associated behavior diagram that indicates the behavior of the object. In conjunction with the use-case diagram we may provide a script or interaction diagram to show a time line of events. It consists of sequence and collaboration diagrams.

4. Interaction diagram

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.

5. State chart diagram:

It consists of state, events and activities. State diagrams are a familiar technique to describe the behavior of a system. They describe all of the possible states that a particular object can get into and how the object's state changes as a result of events that reach the object. In most OO techniques, state diagrams are drawn for a single class to show the lifetime behavior of a single object.

6. 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.

7. Implementation diagram:

It shows the implementation phase of the systems development, such as the source code structure and the run-time implementation structure. These are relatively simple high level diagrams compared to the others seen so far. They are of two sub-diagrams, the component diagram and the deployment diagram.

8. Component diagram:

These are organizational parts of a UML model. These are boxes to which a model can be decomposed. They show the structure of the code itself. They model the physical components such as source code, user interface in a design. It is similar to the concept of packages.

9. Deployment diagram:

The deployment diagram shows the structure of the runtime system. It shows the configuration of runtime processing elements and the software components that live in them. They are usually used in conjunction with deployment diagrams to show how physical modules of code are distributed on the system.

NOTATION ELEMENTS:

These are explanatory parts of UML model. They are boxes which may apply to describe and remark about any element in the model. They provide the information for understanding the necessary details of the diagrams.

Relations in the UML:

These are four kinds of relationships used in an UML diagram, they are:
- ❖ Dependency
- ❖ Association
- ❖ Generalization
- ❖ Realization

Dependency:

It is a semantic relationship between two things in which a change one thing affects the semantics of other things. Graphically a dependency is represented by a non-continuous line.

Association:

It is a structural relationship that describes asset of links. A link is being connected among objects. Graphically association is represented as a solid line possibly including label.

Generalization:

It is a specialized relationship in which the specialized elements are substitutable for object of the generalized element. Graphically it is a solid line with hollow arrow head parent.

Realization:

It is a semantic relation between classifiers. Graphically it is represented as a cross between generalization and dependency relationship.

Where UML can be used:

UML is not limited to modeling software. In fact it is expressive to model non-software such as to show in structure and behavior of health case system and to design the hardware of the system.

Conceptual model be UML:

UML you need to form the conceptual model of UML. This requires three major elements:
- ❖ UML basic building blocks.
- ❖ Rules that dictate how this building blocks are put together.
- ❖ Some common mechanism that apply throughout the language.

Once you have grasped these ideas, you may be able to read. UML create some basic ones. As you gain more experience in applying conceptual model using more advanced features of this language.

Building blocks of the UML:

The vocabulary of UML encompasses these kinds of building blocks.

Use CASE definition:

Description:

A use case is a set of scenarios tied together by a common user goal. A use case is a behavioral diagram that shows a set of use case actions and their relationships.

Purpose:

The purpose of use case is login and exchange messages between sender and receiver (Email client).

Main flow:

First, the sender gives his id and enters his login. Now, he enters the message to the receiver id.

Alternate flow:

If the username and id by the sender or receiver is not valid, the administrator will not allow entering and "Invalid password" message is displayed.

Pre-condition:

A person has to register himself to obtain a login ID.

Post-condition:

The user is not allowed to enter if the password or user name is not valid.

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.

During analysis, we use class diagram to show roles and responsibilities of entities that provide email client system behaviors design. We use to capture the structure of classes that form the email client system architecture.

A class diagram is represented as:

<<Class name>>
<<Attribute 1>>
<<Attribute n>>
<<Operation ()>>

Relationship used:

A change in one element affects the other

Generalization:

It is a kind of relationship

State chart:

Description:

- ❖ The state chart diagram made the dynamic behavior of individual classes.
- ❖ State chart shows the sequences of states that an object goes through events and state transitions.
- ❖ A state chart contains one state 'start' and multiple 'end' states.

The important objectives are:

Decision:

It represents a specific location state chart diagram where the work flow may branch based upon guard conditions.

Synchronization:

It gives a simultaneous workflow in a state chart diagram. They visually define forks and joints representing parallel workflow.

Forks and joins:

- ❖ A fork construct is used to model a single flow of control.
- ❖ Every work must be followed by a corresponding join.
- ❖ Joints have two or more flow that unit into a single flow.

State:

A state is a condition or situation during a life of an object in which it satisfies condition or waits for some events.
Transition:

It is a relationship between two activities and between states and activities.

Start state:

A start state shows the beginning of a workflow or beginning of a state machine on a state chart diagram.

End state:

It is a final or terminal state.

Activity diagram

Description:

Activity diagram provides a way to model the workflow of a development process. We can also model this code specific information such as class operation using activity diagram. Activity diagrams can model different types of diagrams. There are various tools involved in the activity diagram.

Activity:

An activity represents the performance of a task on duty. It may also represent the execution of a statement in a procedure.

Decision:

A decision represents a condition on situation during the life of an object, which it satisfies some condition or waits for an event.

Start state:

It represents the condition explicitly the beginning of a workflow on an activity.

Object flow:

An object on an activity diagram represents the relationship between activity and object that creates or uses it.

Synchronization:

It enables us to see a simultaneous workflow in an activity.

End state:

An end state represents a final or terminal state on an activity diagram or state chart diagram.

Sequence diagram:

Description:

A sequence diagram is a graphical view of scenario that shows object interaction in a time based sequence what happens first what happens next. Sequence diagrams are closely related to collaboration diagram.

The main difference between sequence and collaboration diagram is that sequence diagram show time based interaction while collaboration diagram shows objects associated with each other.

The sequence diagram for the e-mail client system consists of the following objectives:

Object:

An object has state, behavior and identity. An object is not based is referred to as an instance.

The various objects in e-mail client system are:
- ❖ User
- ❖ Website
- ❖ Login
- ❖ Groups

Message icon:

A message icon represents the communication between objects indicating that an action will follow. The message icon is the horizontal solid arrow connecting lifelines together.

Collaboration diagram:

Description:

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. Collaboration diagram is an interaction diagram that shows the order of messages that implement an operation or a transaction. Collaboration diagram shows object s, their links and their messages. They can also contain simple class instances and class utility instances.

During, analysis indicates the semantics of the primary and secondary interactions. Design, shows the semantics of mechanisms in the logical design of system.
Toggling between the sequence and collaboration diagrams

When we work in either a sequence or collaboration diagram, it is possible to view the corresponding diagram by pressing F5 key.

CONCLUSION:
Thus the study for case tools was done.

Ex no:2

Date:

PASSPORT AUTOMATION SYSTEM

AIM:

To create an automated system to perform the Passport Process.

(I) 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.

(II)SOFTWARE REQUIREMENT SPECIFICATION:

2.1SOFTWARE INTERFACE
• **Front End Client** - The applicant and Administrator online

interface is built using

JSP and HTML. The Administrators's local interface is built using

Java.

•**Web Server** - Glassfish application server(Oracle Corporation).

•**Back End** - Oracle database.

2.2HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

(III) USECASE DIAGRAM :

registration

enter applicant id

check status

administrator

applicant

process applicant

dispatch passport

Fig.3. USECASE DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(IV) ACTIVITY DIAGRAM:

Passport Automation System

New Registration

Enter Details

Register

Fig.4.1. ACTIVITY DIAGRAM FOR REGISTER
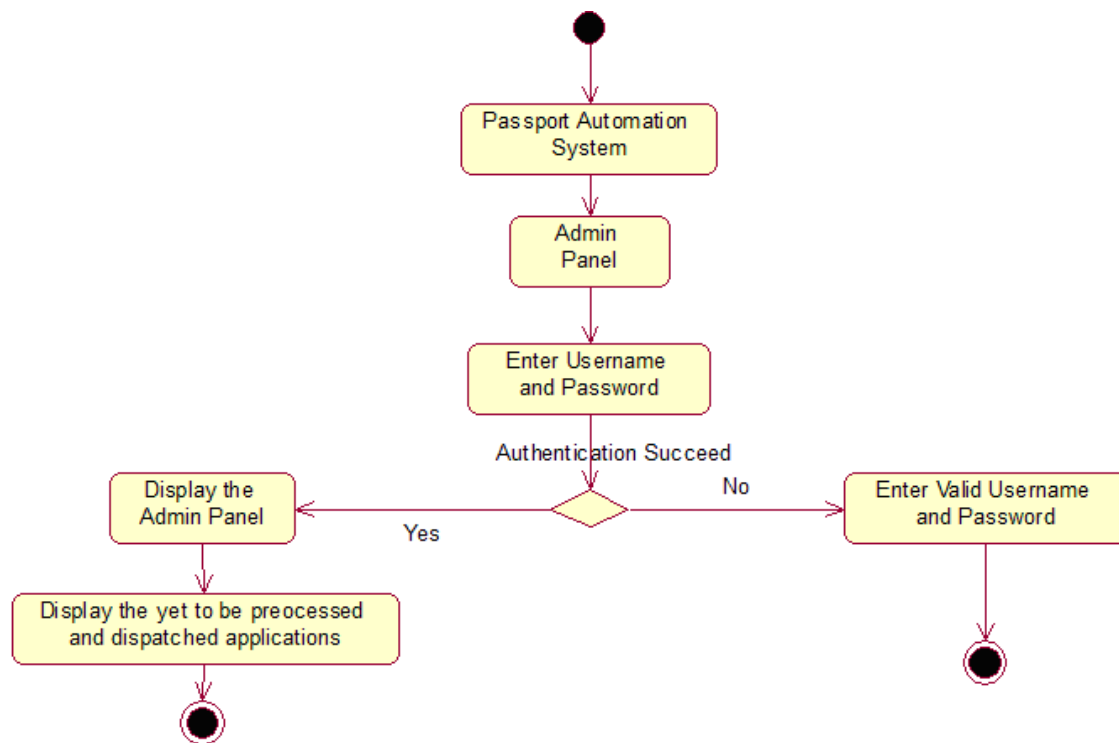
Fig.4.2. ACTIVITY DIAGRAM FOR ADMINISTRATION
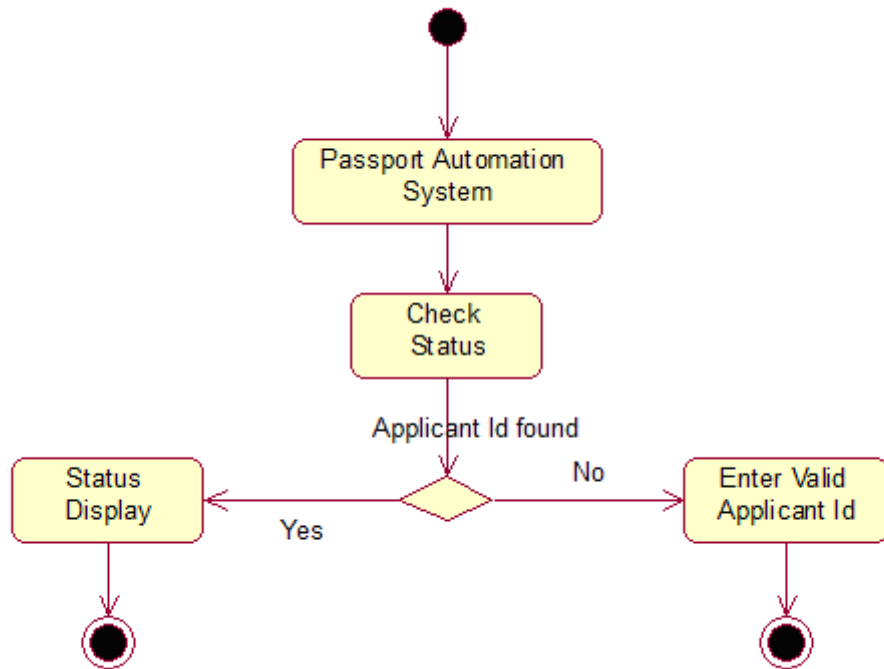
Fig.4.3. ACTIVITY DIAGRAM FOR CHECKING STATUS

(V) 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 Passport Automation system class diagram consists of four classesPassport Automation System

1. New registration
2. Gender
3. Application Status
4. Admin authentication
5. Admin Panel
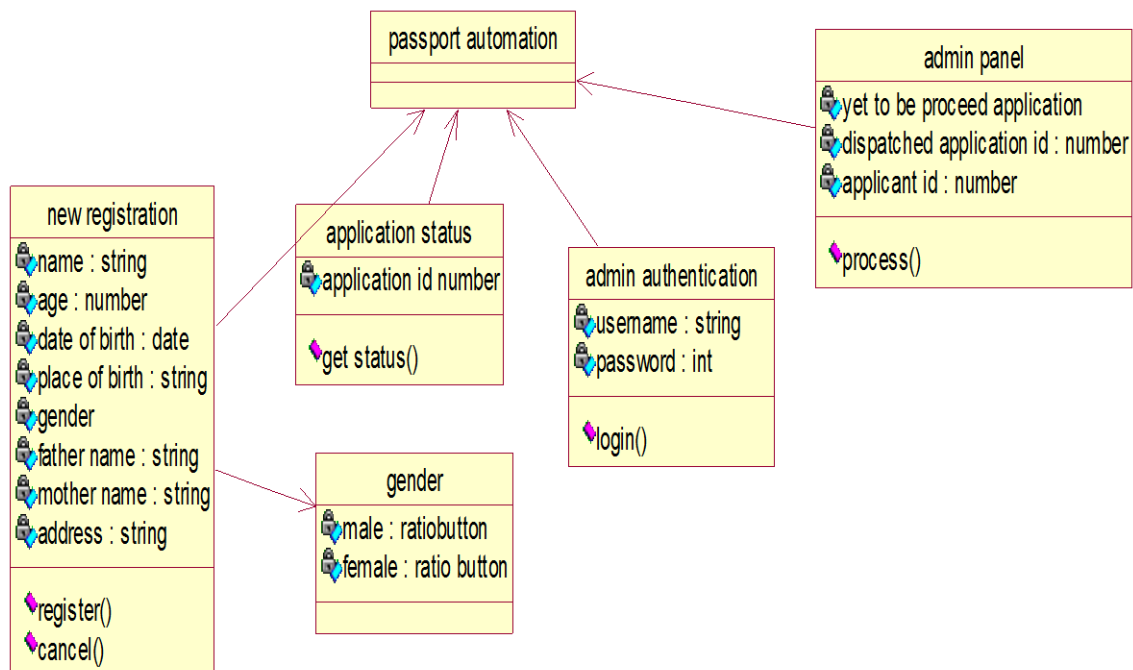
Fig.5. CLASS DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(VI) 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 for each USE-CASE that exists when a user administrator, check status and new registration about passport automation system are given.
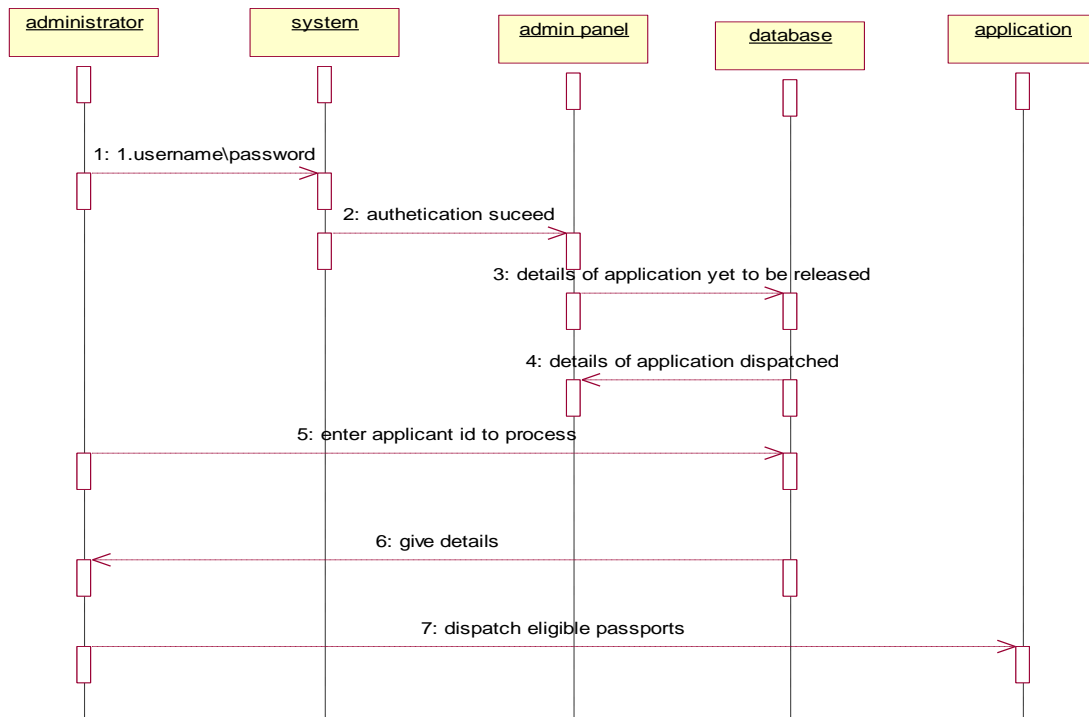
Fig.6.1.SEQUENCE DIAGRAM FOR ADMINISTRATOR
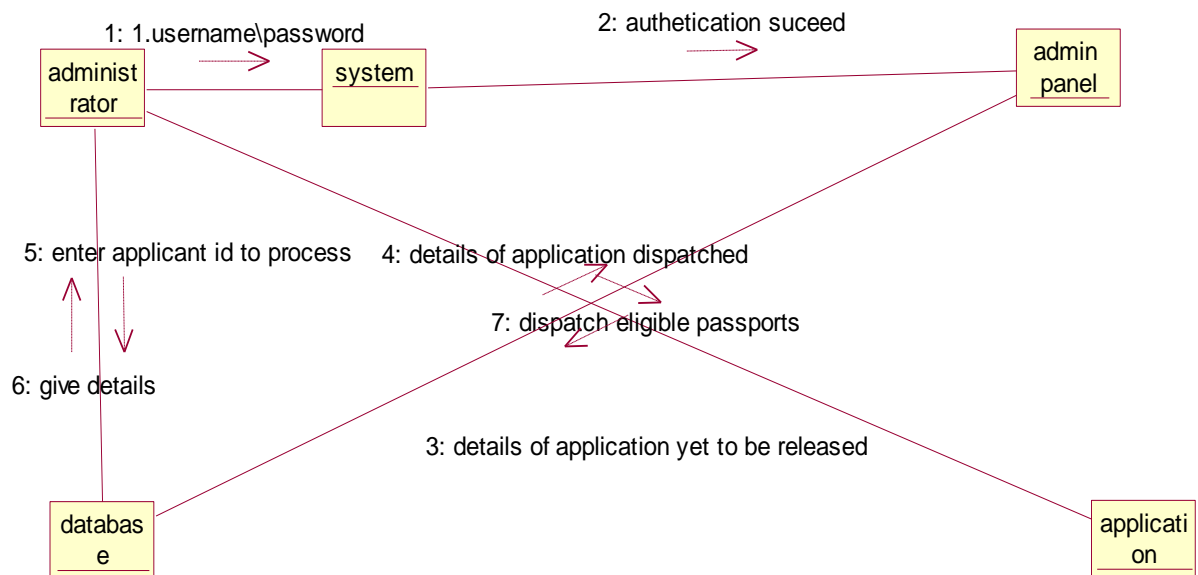
Fig.6.2.COLLABORATION DIAGRAM FOR ADMINISTRATOR

The diagrams show the process done by the administrator to the Passport Automation system. The applicant has to enter his details. The

details entered are verified by the administrator and the applicant is approved if the details match then the passport is dispatch, otherwise an appropriate error message is displayed.
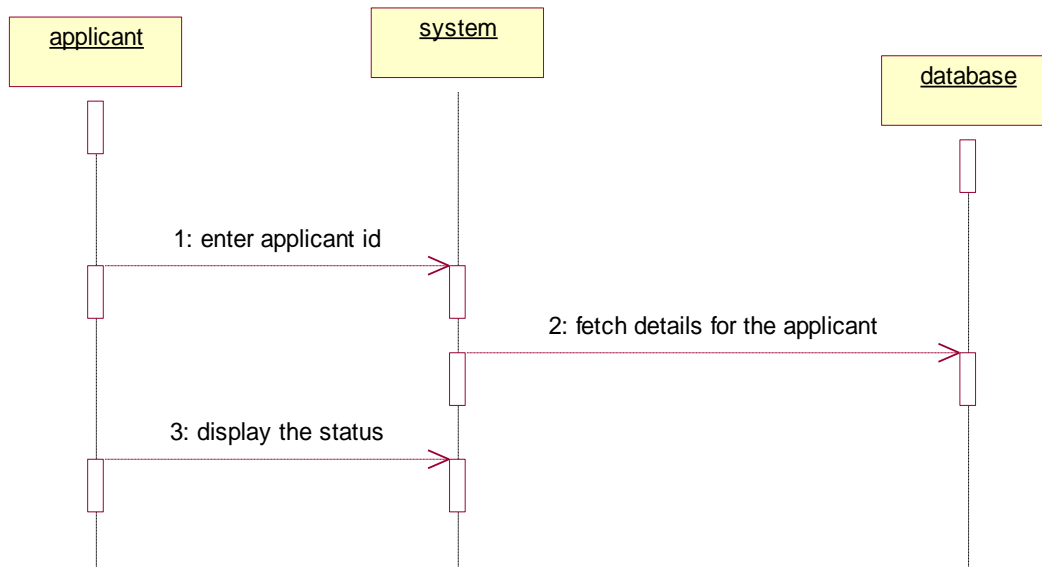
Fig.6.3.SEQUENCE DIAGRAM FOR CHECKING STATUS
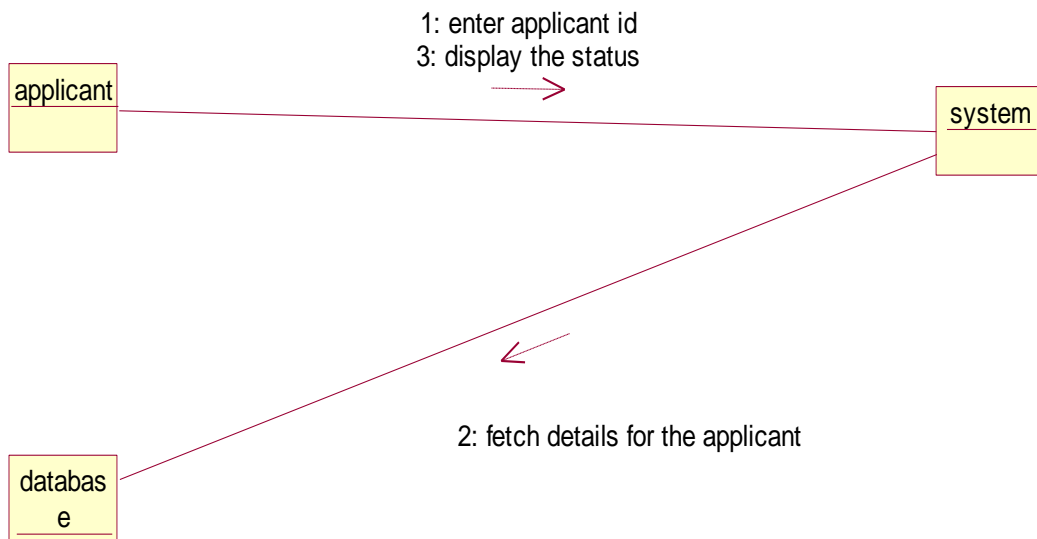
Fig.6.4.COLLABORATION DIAGRAM FOR CHECKING STATUS

The diagrams show the applicant enters his id and the system fetch the details from the database and display the status.
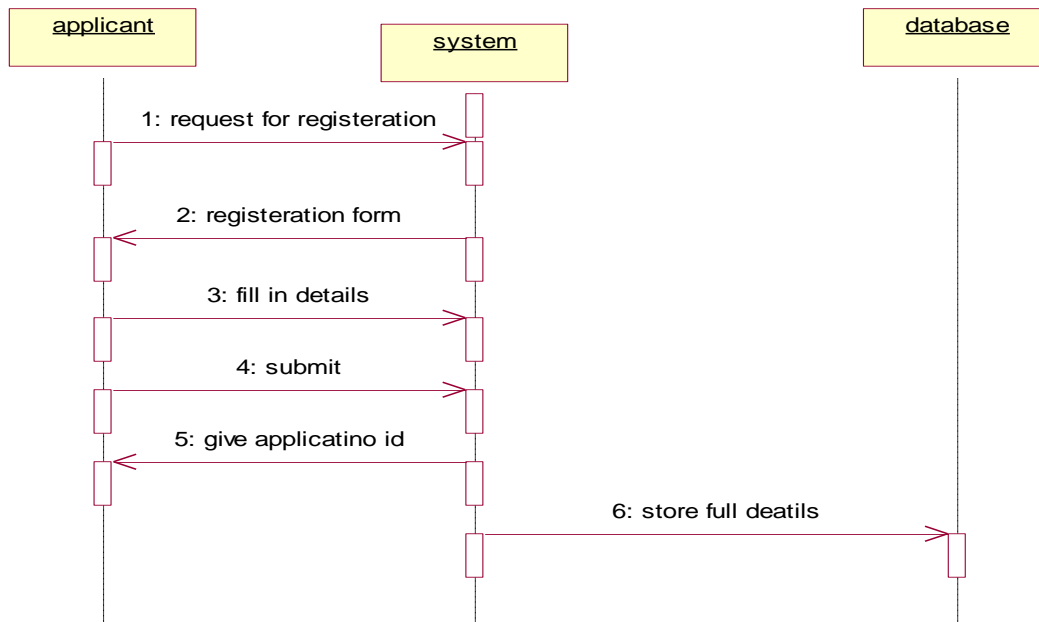
Fig.6.5.SEQUENCE DIAGRAM FOR NEW REGISTRATION
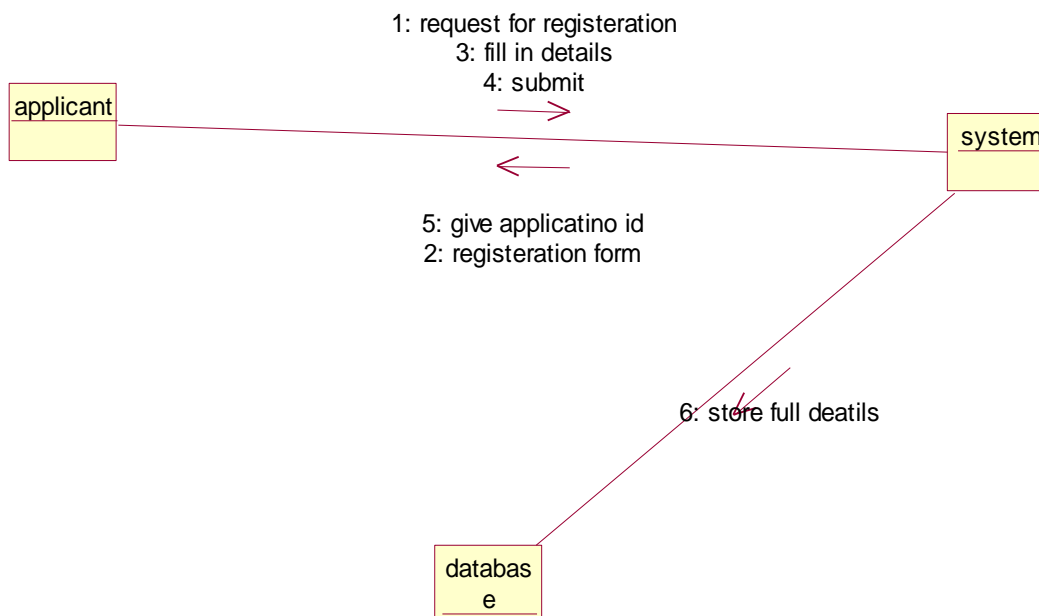
Fig.6.6.COLLABORATION DIAGRAM FOR NEW REGISTRATION

The diagrams show the applicant request the system for registration and the system provide the register form and applicant fill the form and submit and the system give the applicant id. The database stores the full details.

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM
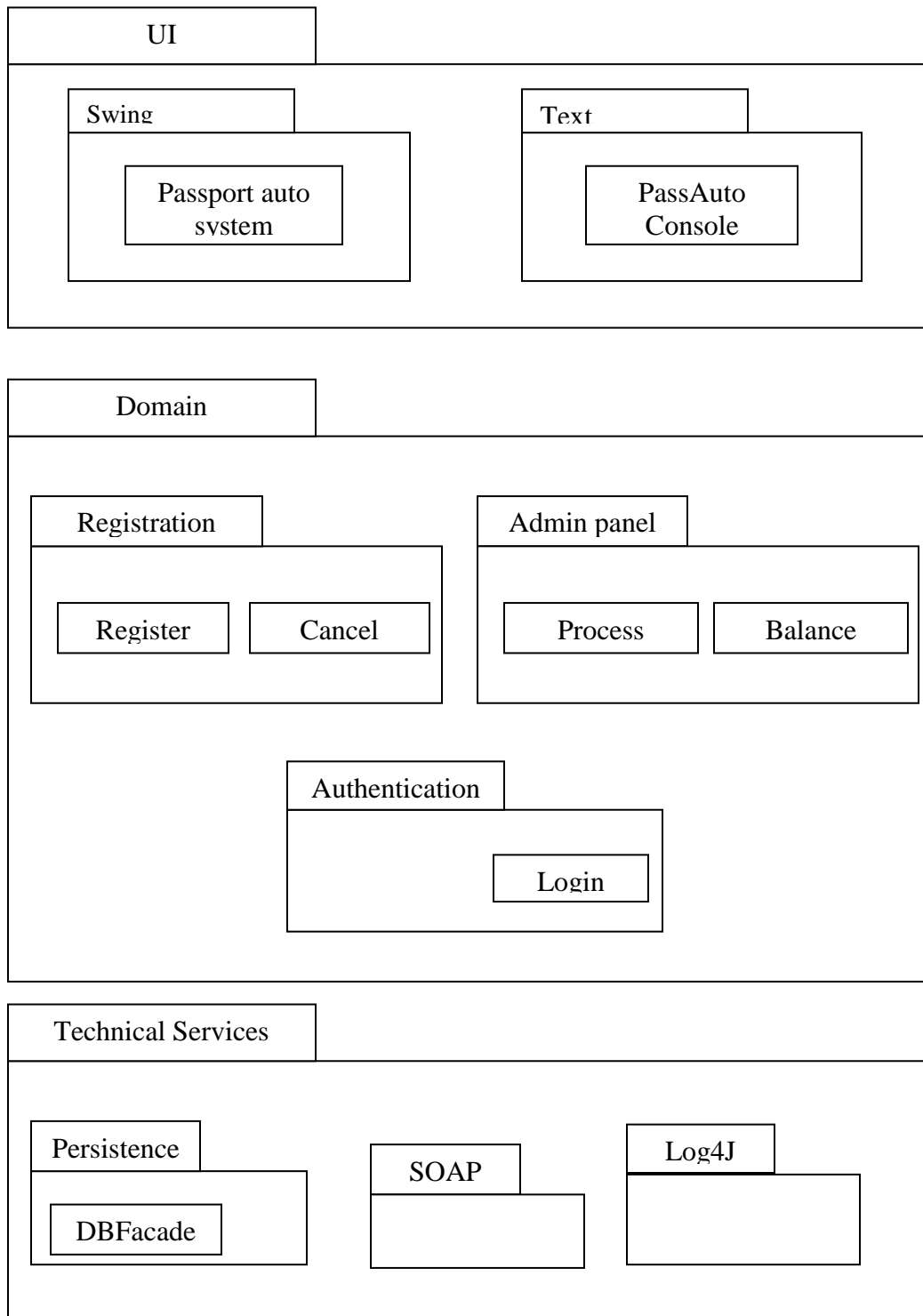
UI

Swing

Passport auto system

Text

PassAuto Console

Domain

Registration

Register | Cancel

Admin panel

Process | Balance

Authentication

Login

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
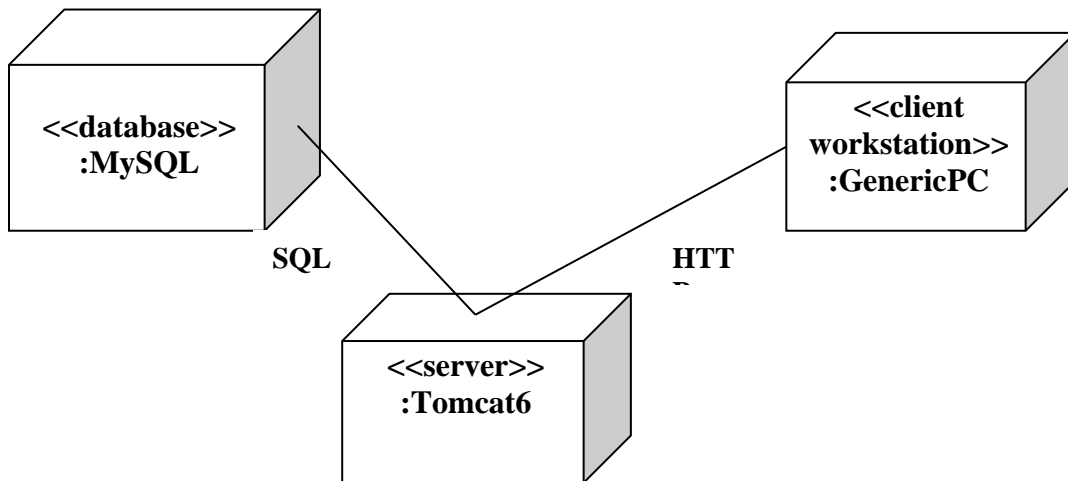
DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationshipamong components in a
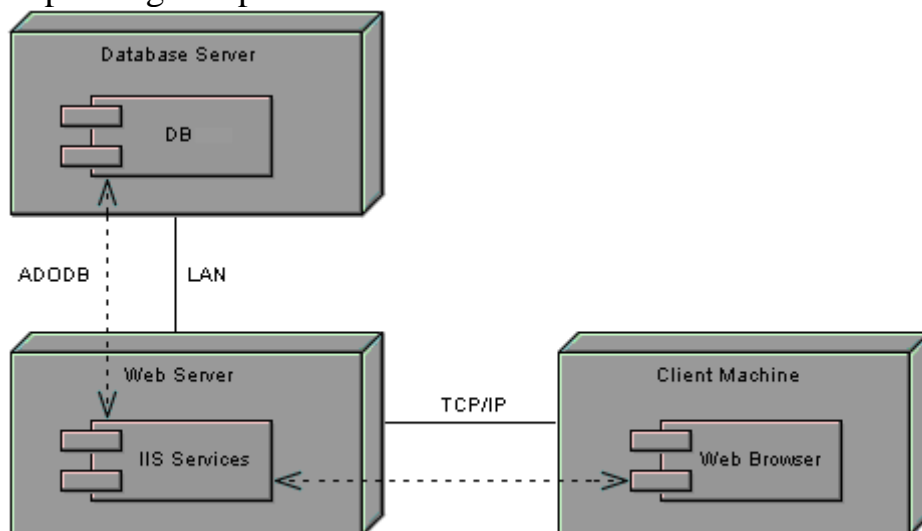
system

RESULT:

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

Ex no:3

Date:

BOOK BANK SYSTEM

AIM:

To create a system to perform book bank operation

(I) PROBLEM STATEMENT:

A Book Bank 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) SOFTWARE REQUIREMENTS SPECIFICATION:

2.1 SOFTWARE INTERFACE

- **Front End Client** - The Student and Librarian online interface is built using JSP and HTML. The Librarians local interface is built using Java.
- **Web Server** - Glassfish application server (Oracle Corporation).
- **Back End** - Oracle database

2.2 HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.
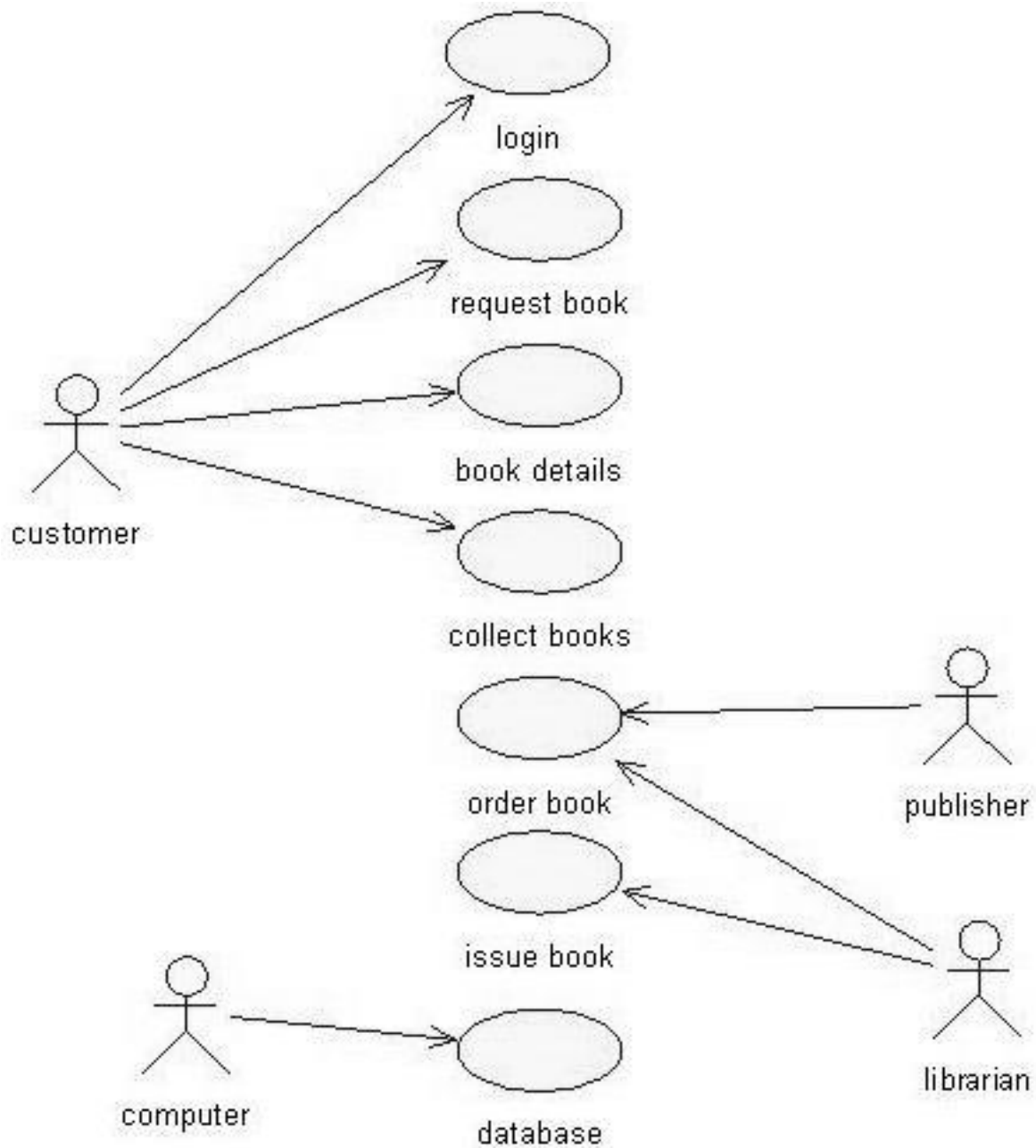
(III)USE-CASE DIAGRAM:

Fig 3. USE-CASE DIAGRAM FOR BOOK BANK SYSTEM
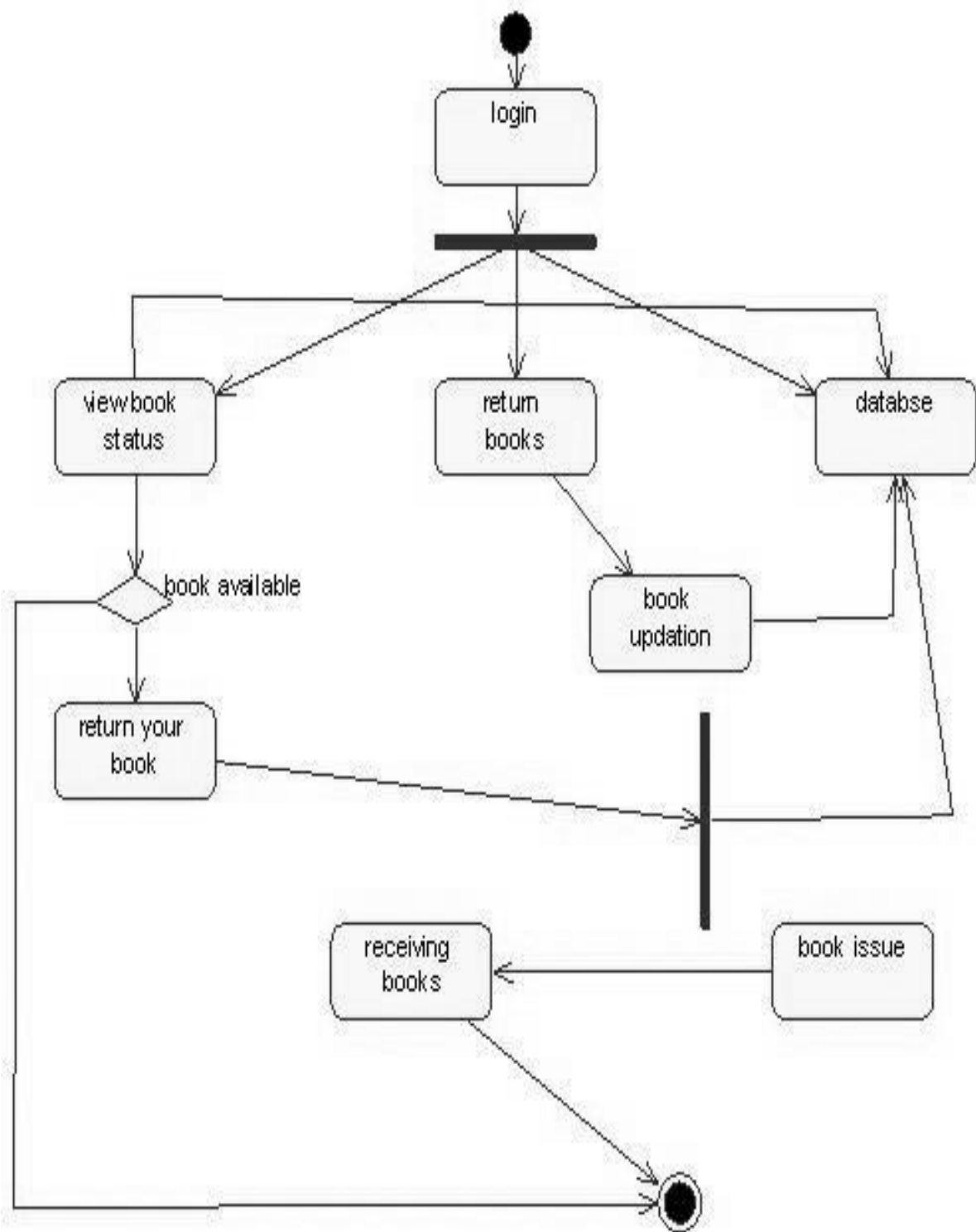
(IV) ACTIVITY DIAGRAM:

Fig.4. ACTIVITY DIAGRAM

(V) 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.
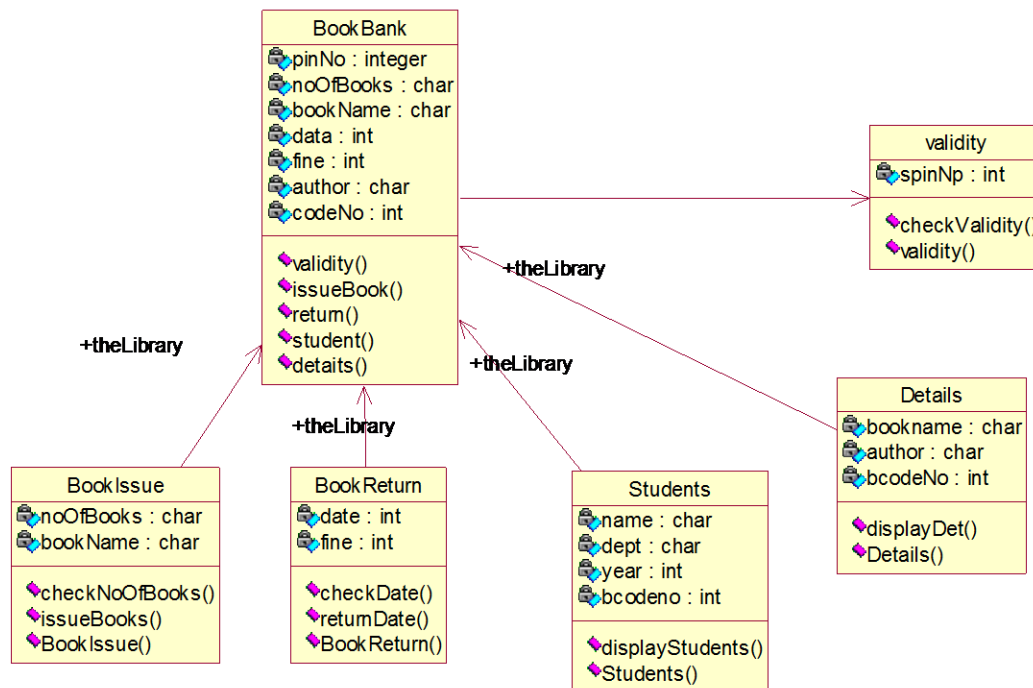

BookBank
- pinNo : integer
- noOfBooks : char
- bookName : char
- data : int
- fine : int
- author : char
- codeNo : int

- validity()
- issueBook()
- return()
- student()
- details()

validity
- spinNp : int

- checkValidity()
- validity()

+theLibrary

Details
- bookname : char
- author : char
- bcodeNo : int

- displayDet()
- Details()

+theLibrary

+theLibrary

+theLibrary

+theLibrary

BookIssue
- noOfBooks : char
- bookName : char

- checkNoOfBooks()
- issueBooks()
- BookIssue()

BookReturn
- date : int
- fine : int

- checkDate()
- returnDate()
- BookReturn()

Students
- name : char
- dept : char
- year : int
- bcodeno : int

- displayStudents()
- Students()

Fig.5. CLASS DIAGRAM FOR BOOK BANK SYSTEM

(VI) SEQUENCE 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.
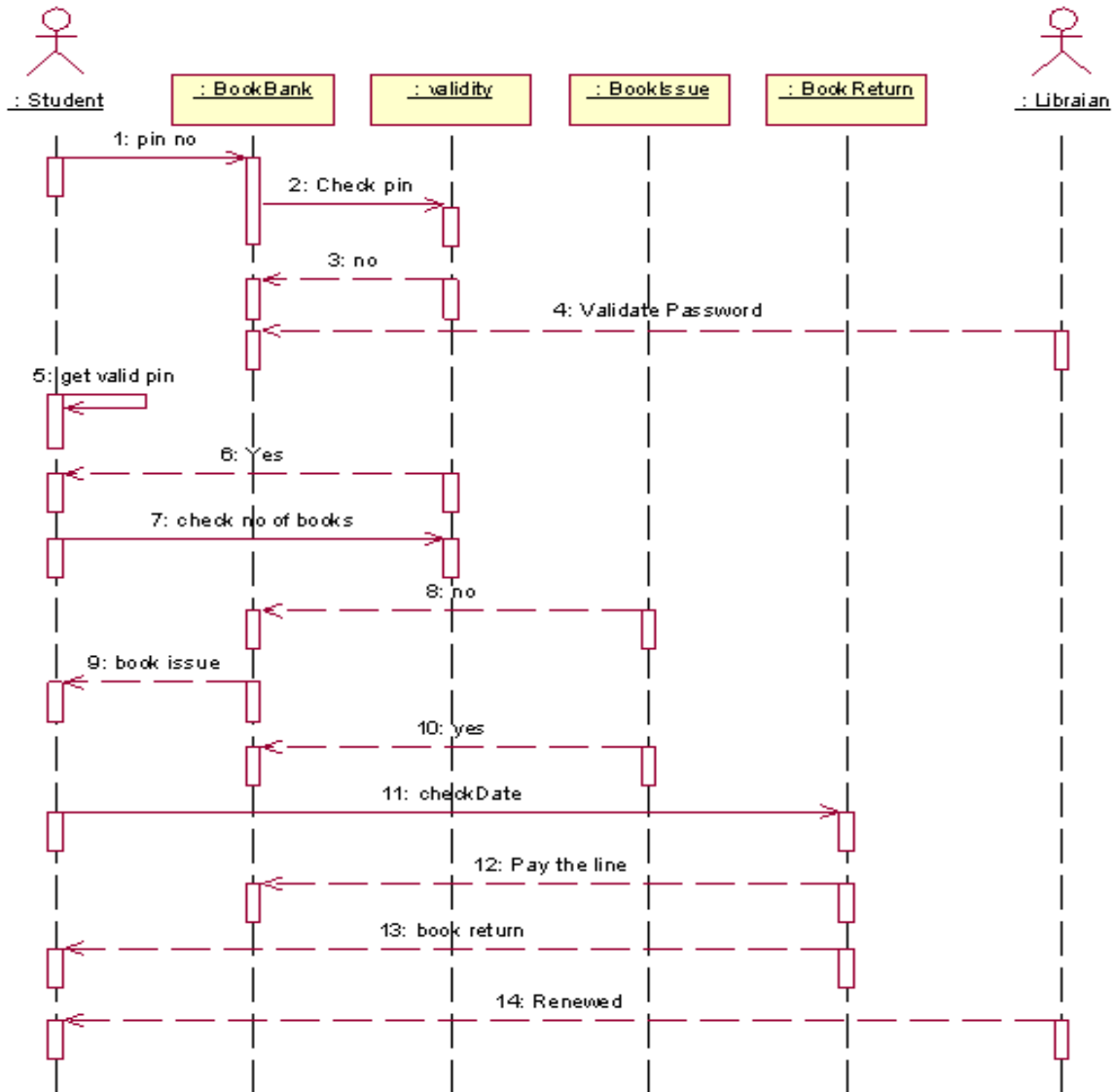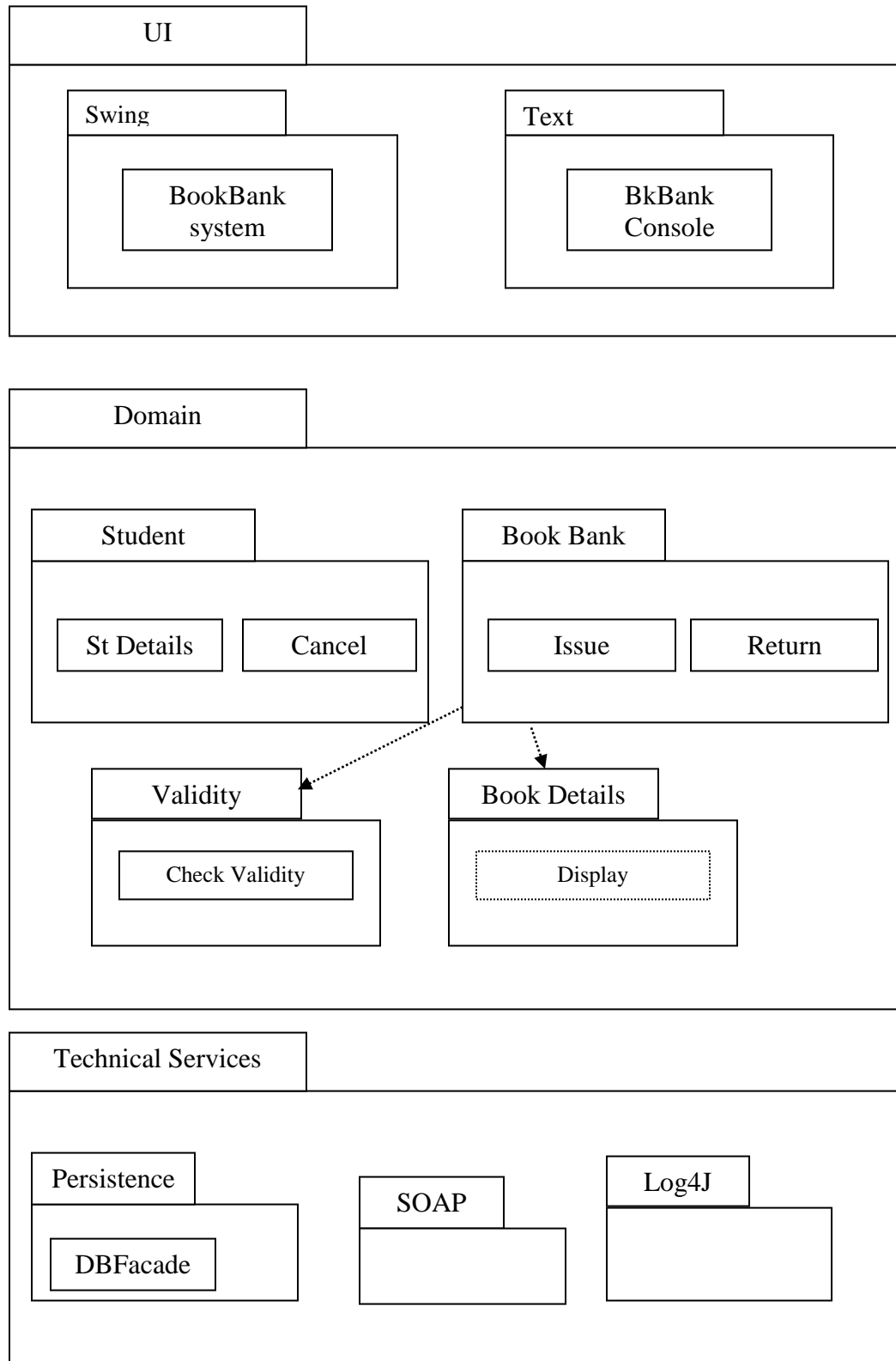

Fig. 6.1. SEQUENCE DIAGRAM FOR DEPOSIT PROCESS

The diagrams show the pin no is entered and check the pin .Get no and validate password check the condition based on condition book issue and return are done. Pay the online and renewed.

5: get valid pin

14: Renewed

: Libraian

4: Validate Password

1: pin no

:
BookBank

9: book issue

: Student

7: check no of books

12: Pay the line

3: no

13: book return

2: Check pin

8: no

10: yes

11: checkDate

6: Yes

: BookReturn

: BookIssue

: validity

Fig. 6.2. COLLABORATION DIAGRAM FOR DEPOSIT PROCESS

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:

UI

Swing

BookBank
system

Text

BkBank
Console

Domain

Student

St Details | Cancel

Book Bank

Issue | Return

Validity

Check Validity

Book Details

Display

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
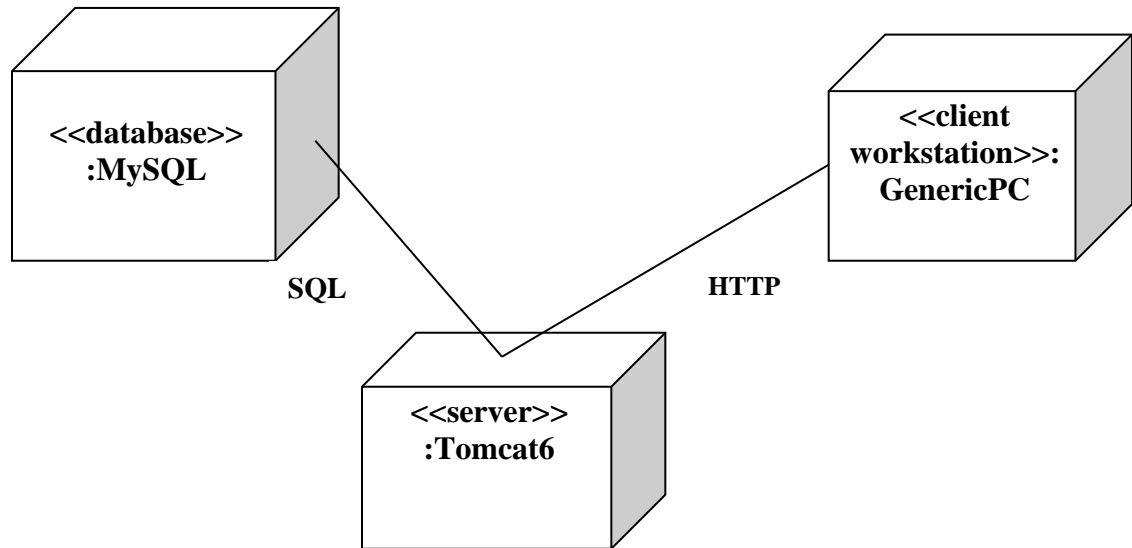
Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships
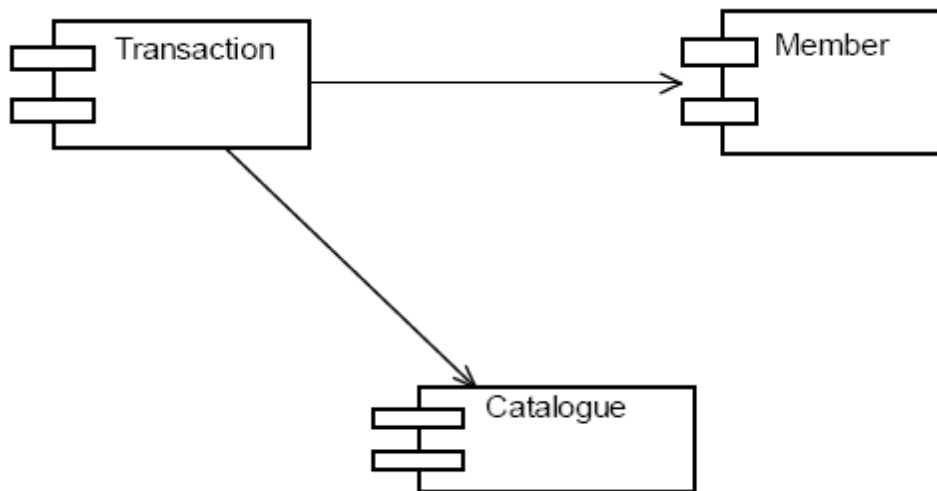
Fig.8.2.COMPONENT DIAGRAM

RESULT:

Thus the mini project for Book Bank System has beensuccessfully executed and codes are generated.

Ex no:4

Date:

EXAM REGISTRATION SYSTEM

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)SOFTWARE REQUIREMENT SPECIFICATION:

2.1SOFTWARE INTERFACE

• **Front End Client** - The student and Controller online interface is built using
JSP and HTML. The Exam Controller's local interface is built using Java.
• **Web Server** - Glassfish application server(SQlCorporation).
• **Back End** - SQL database.

2.2HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have
access to the database in the server.

(III)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 :
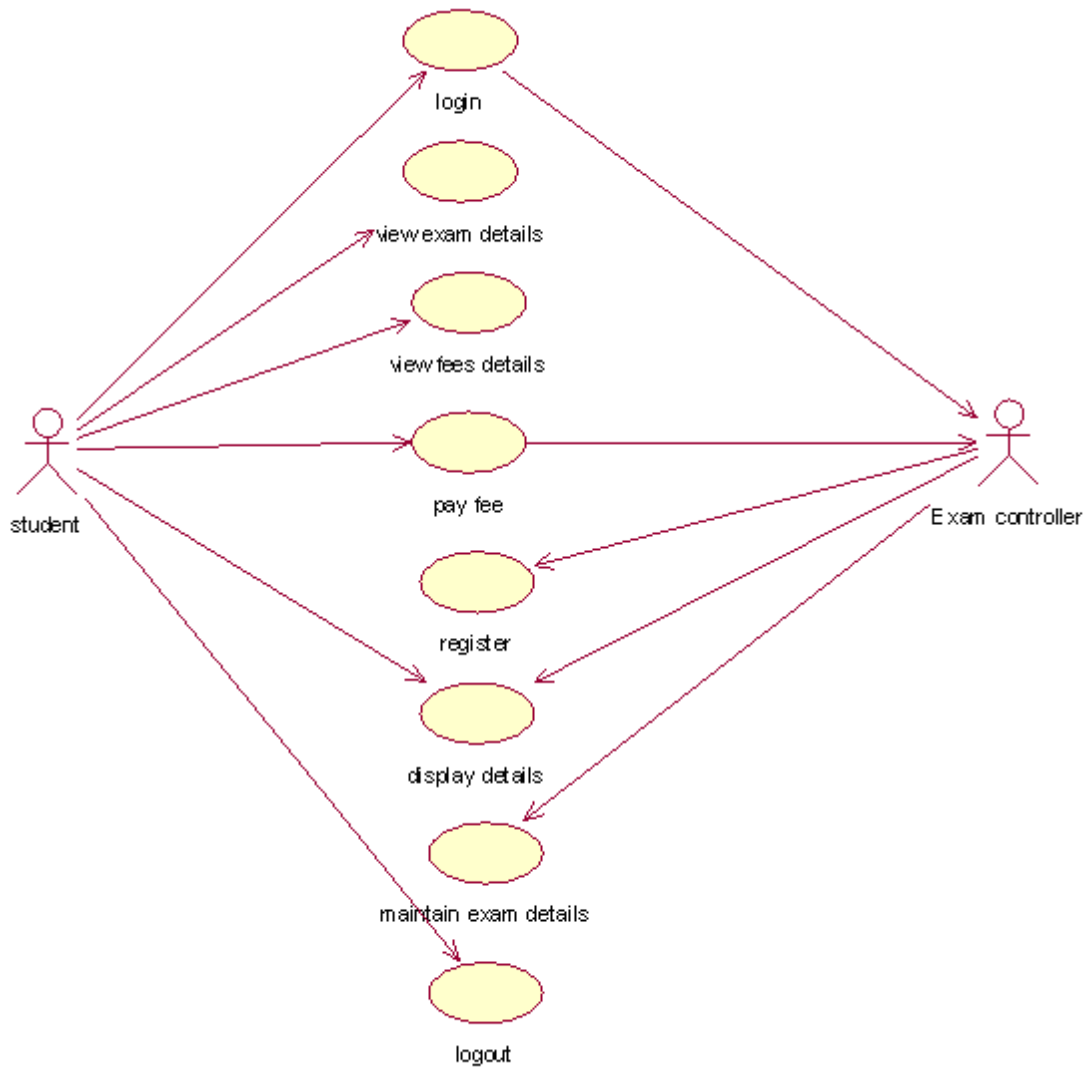
Fig. 3.USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(IV) ACTIVITY DIAGRAM:
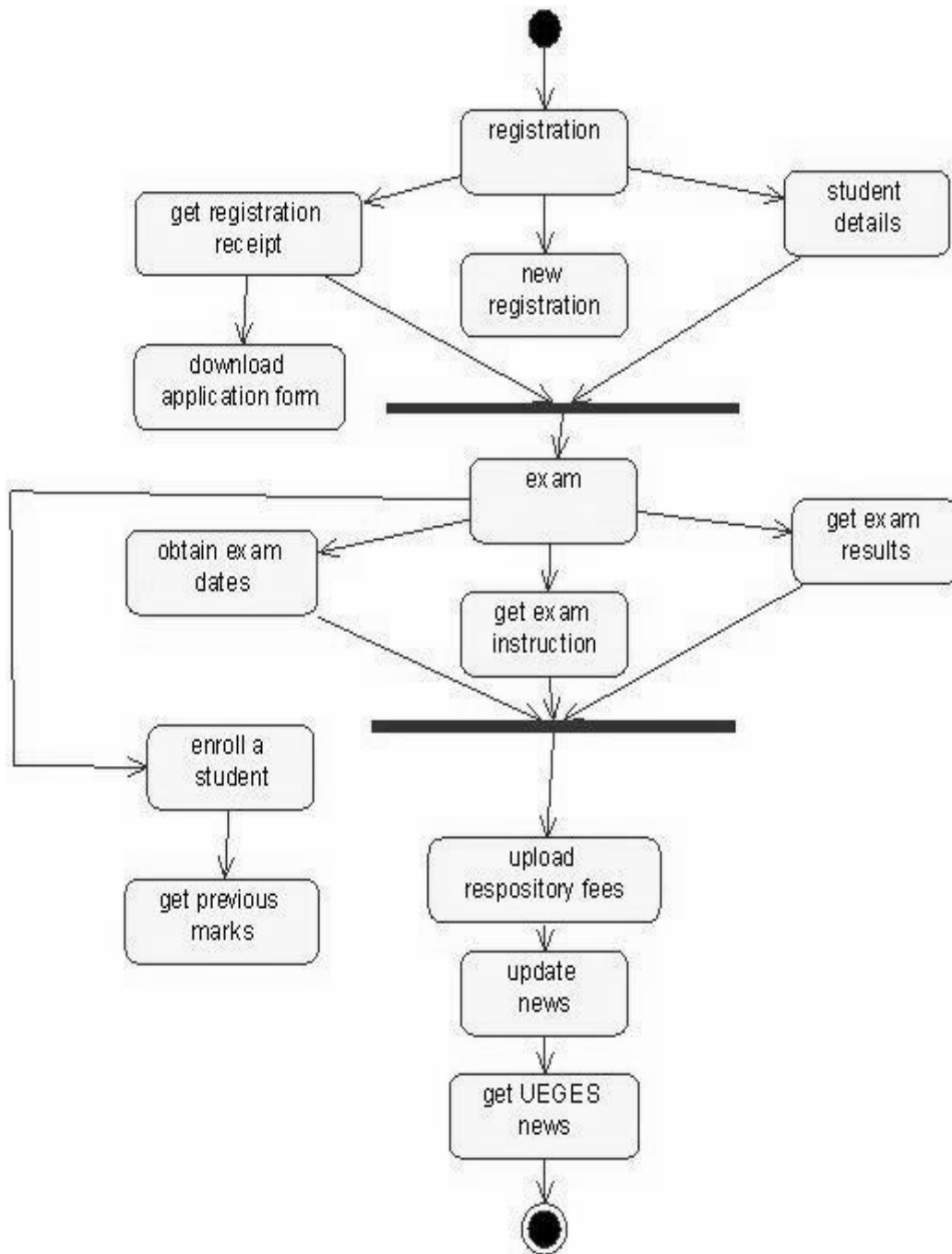
Fig. 4.USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(V)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.
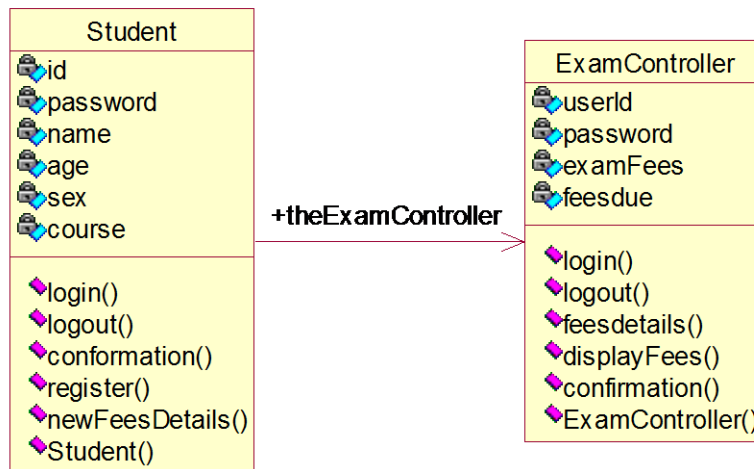
Fig.5. CLASS DIAGRAM FOR EXAM REGISTRATION SYSTEM

(VI)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
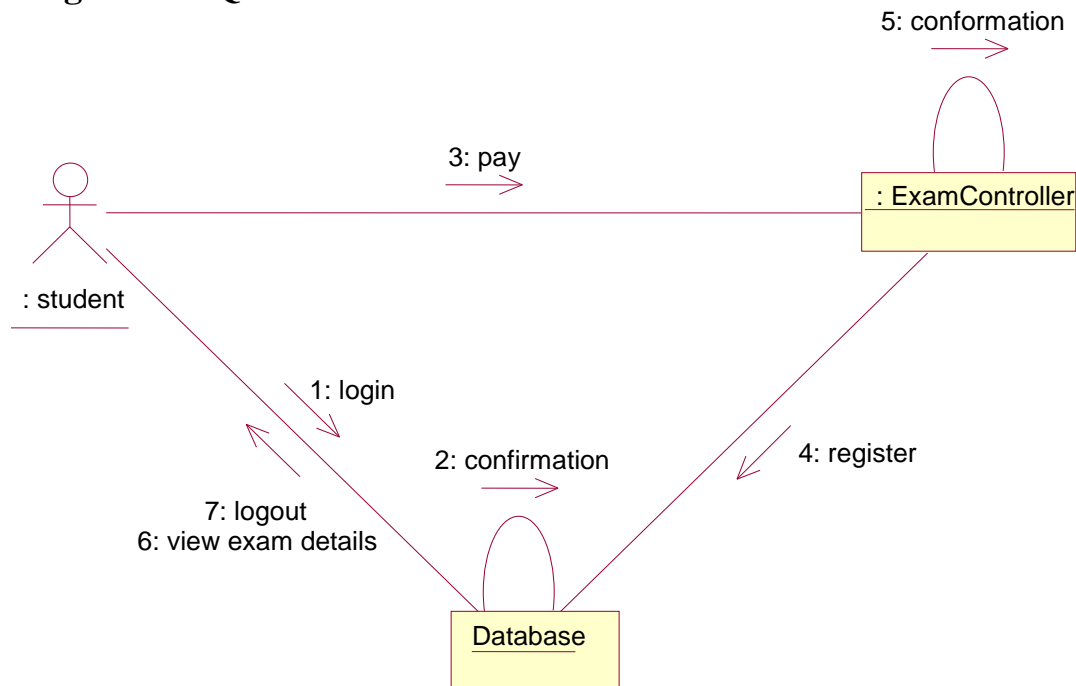
Fig. 6.2. COLLABORATION DIAGRAM FOR REGISTRATION SYSTEM

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:
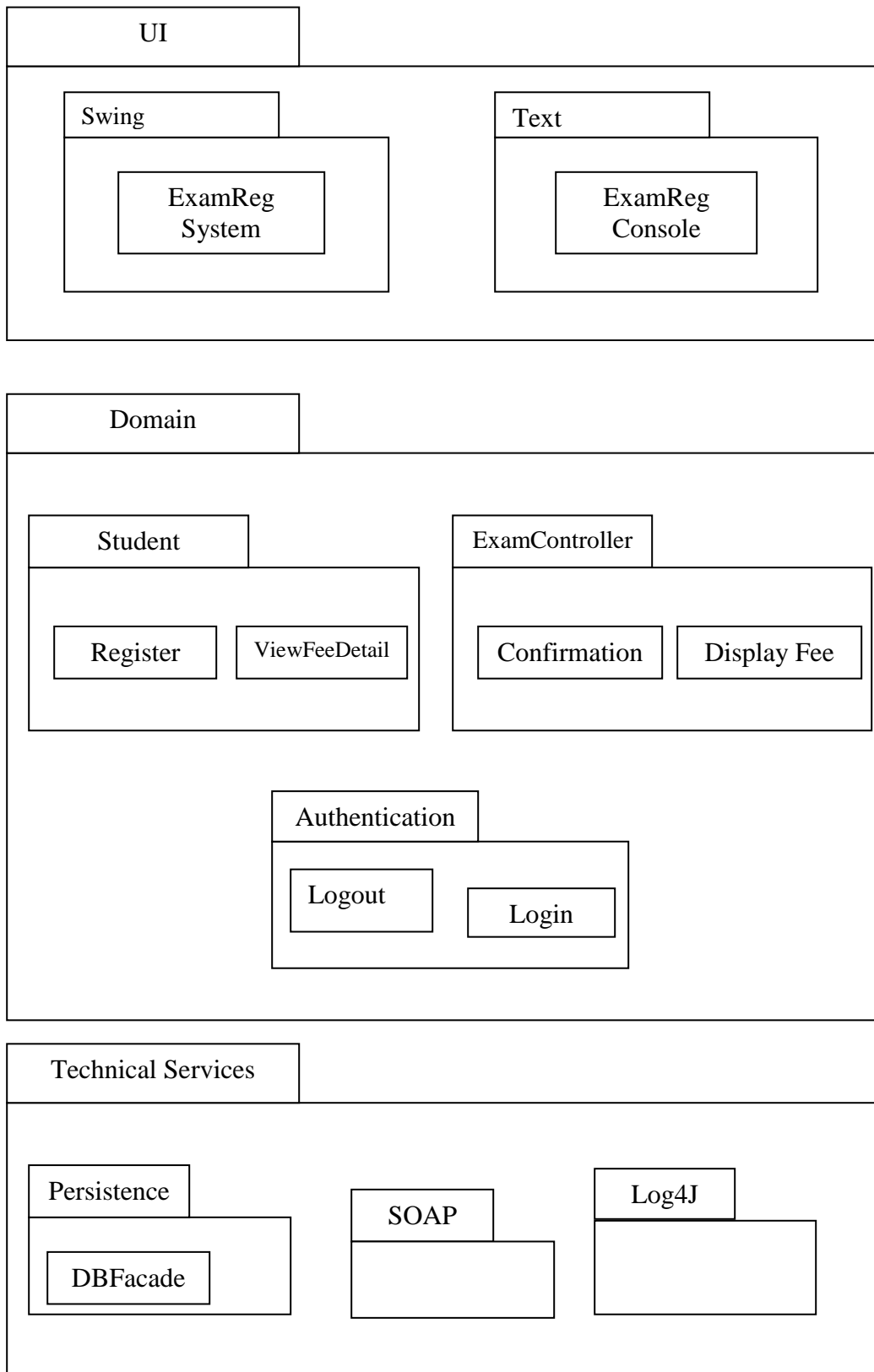
UI

Swing

ExamReg
System

Text

ExamReg
Console

Domain

Student

Register | ViewFeeDetail

ExamController

Confirmation | Display Fee

Authentication

Logout | Login

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
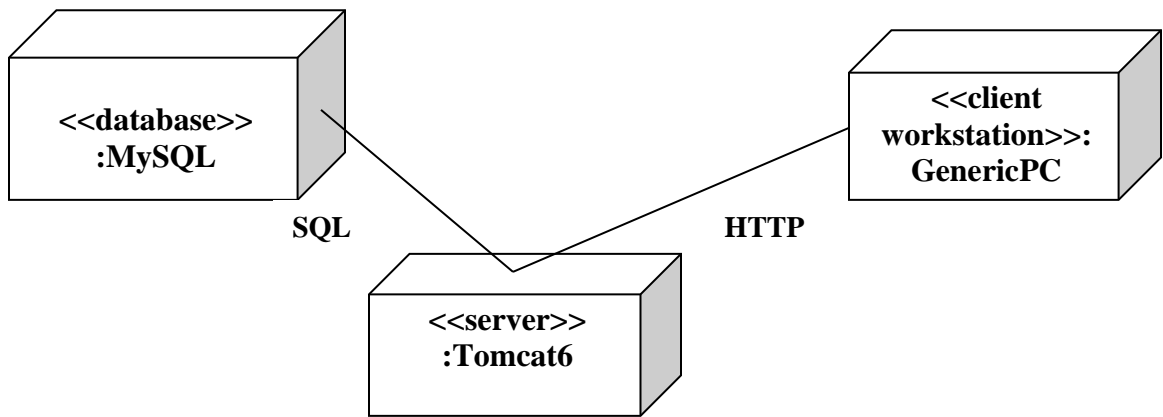
Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships among components in a system.
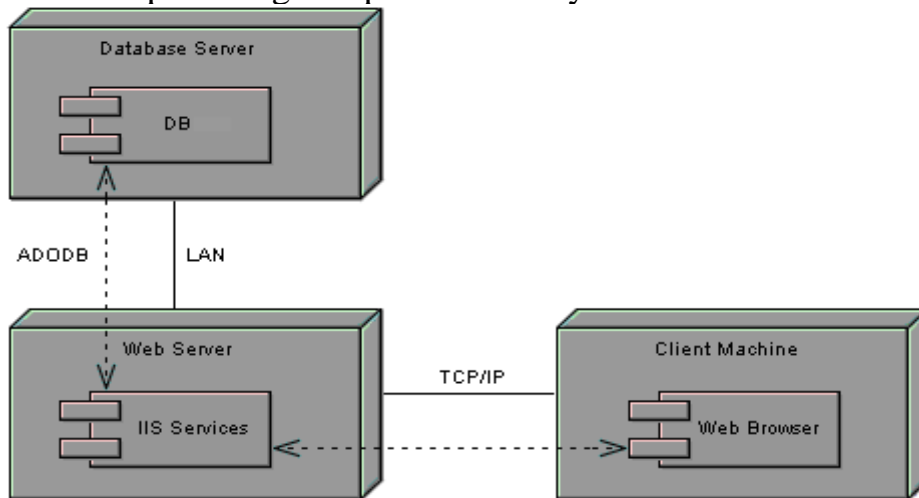
Fig.7.2.COMPONENT DIAGRAM

RESULT:

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

STOCK MAINTENANCE

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

1.1PURPOSE

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.

1.2 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.

1.3 TOOLS TO BE USED

• Eclipse IDE (Integrated Development Environment)

• Rational Rose tool (for developing UML Patterns)

(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
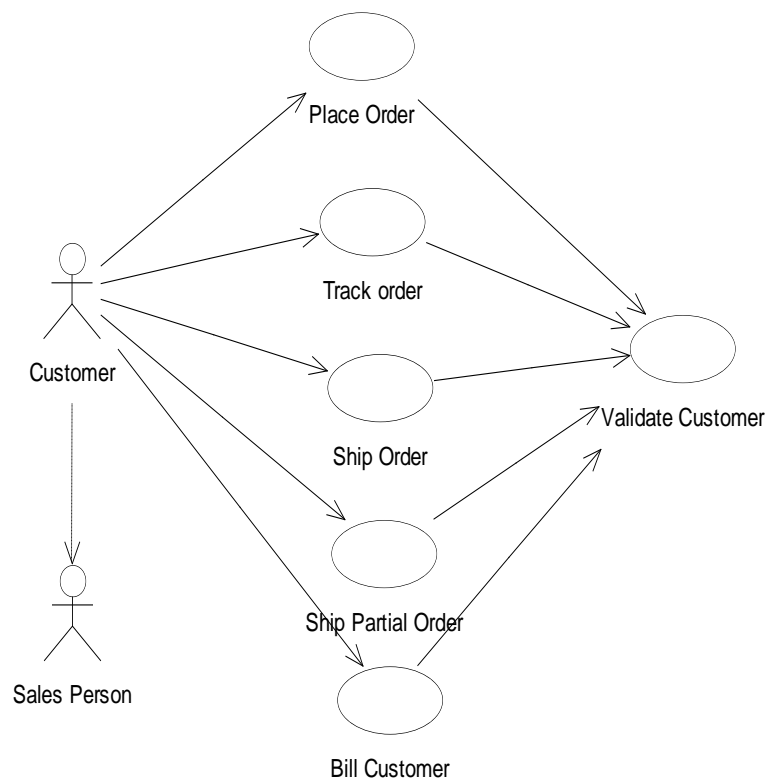
Fig.3. USE CASE DIAGRAM

(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.
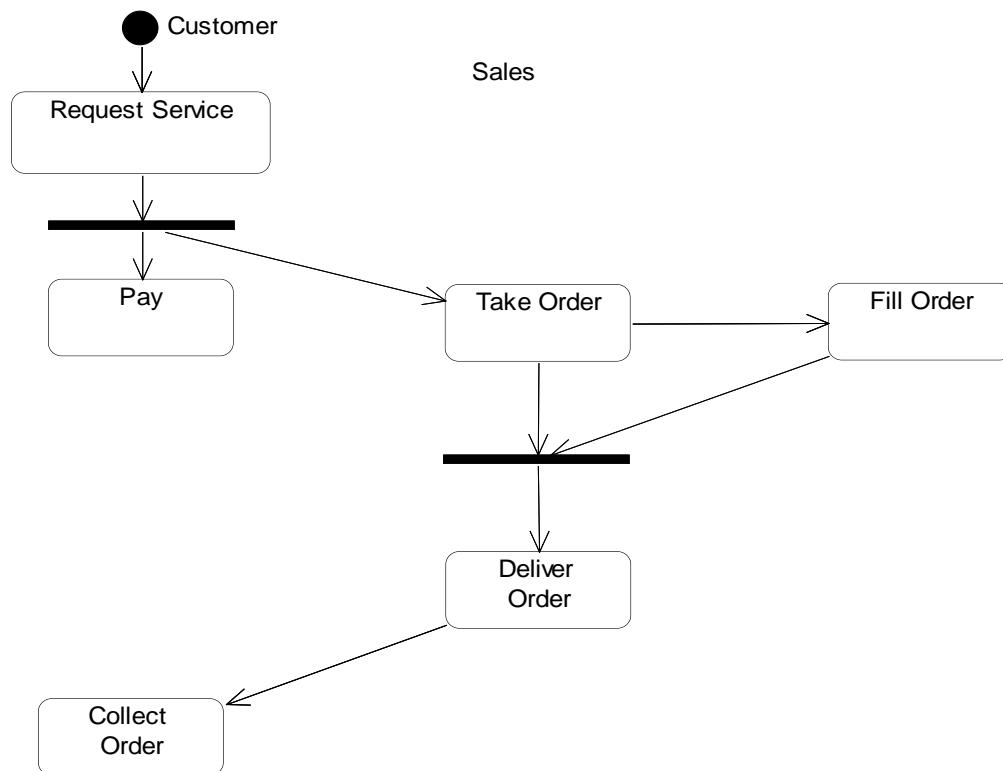
Fig.4. ACTIVITY DIAGRAM

(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.
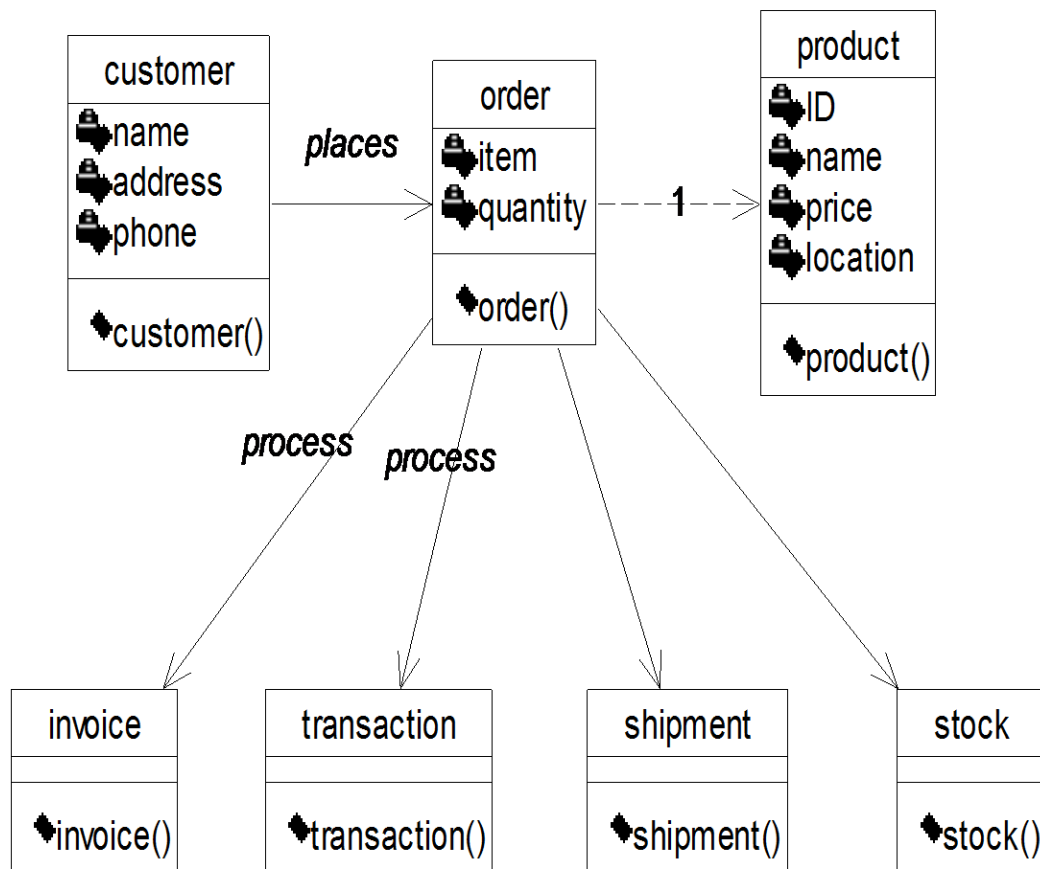
Fig.5. CLASS DIAGRAM

(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.
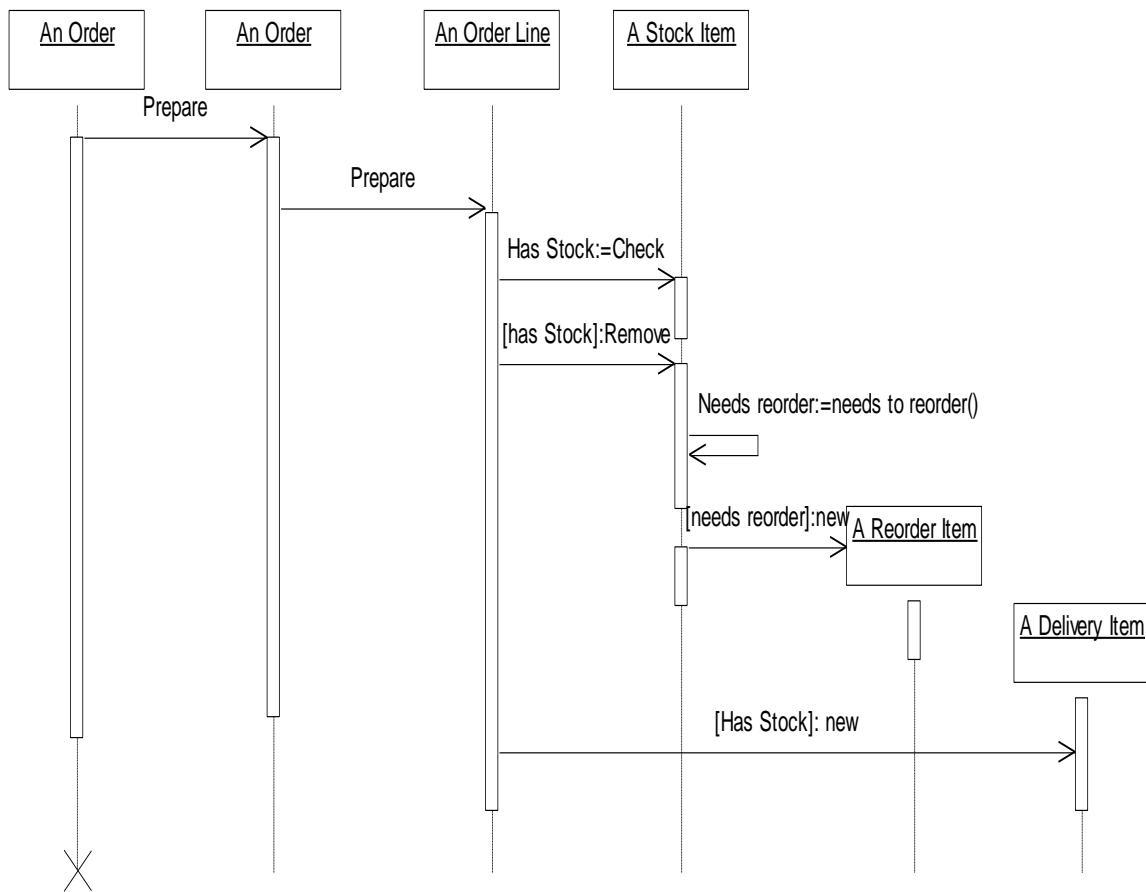
Fig.6.1 SEQUENCE DIAGRAM

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
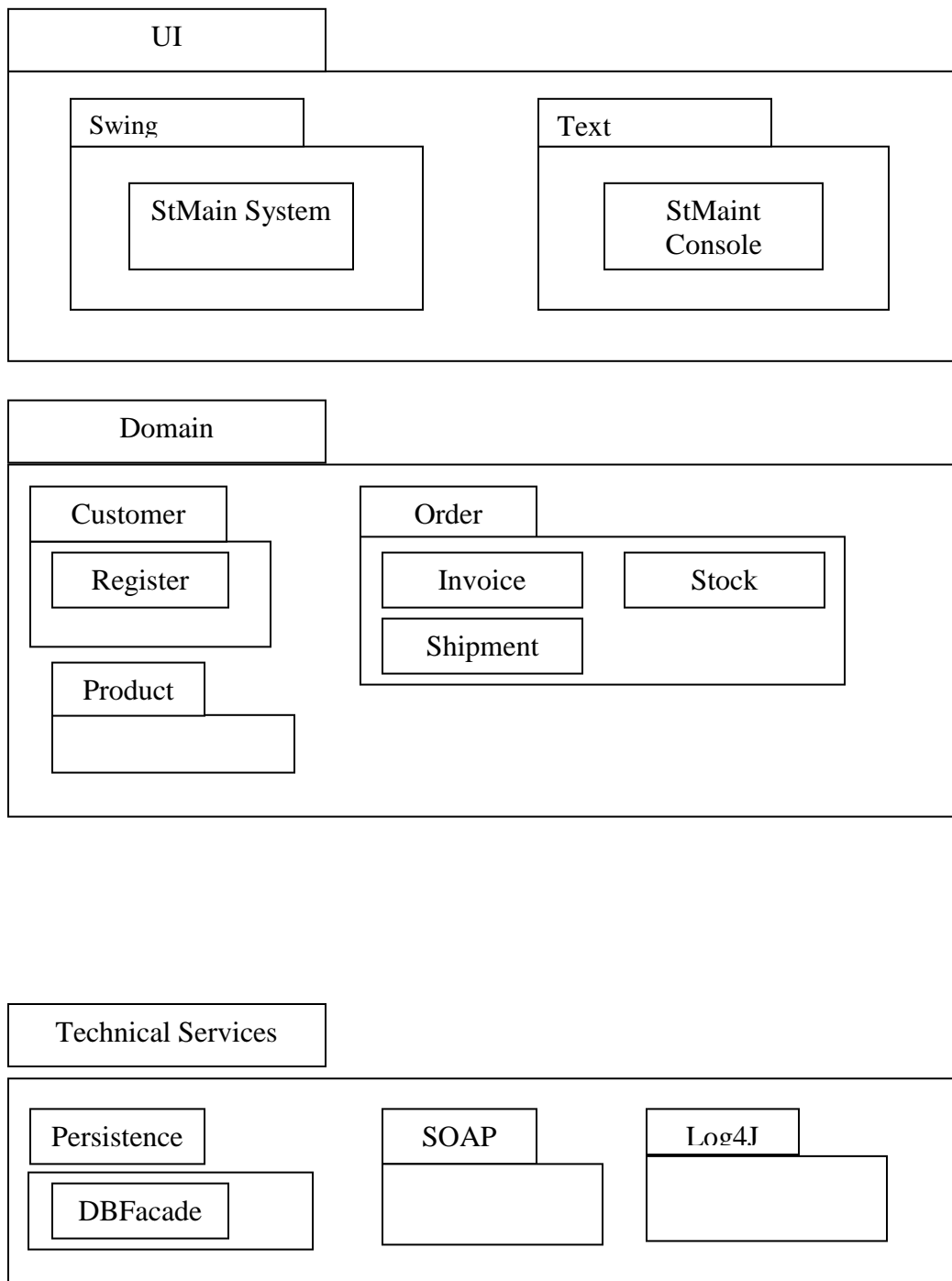
(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM

UI

Swing

StMain System

Text

StMaint Console

Domain

Customer

Register

Product

Order

Invoice   Stock

Shipment

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
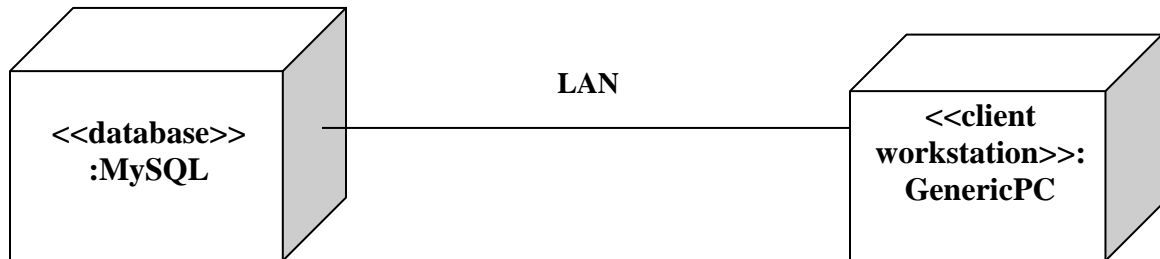
Fig.8.1.DEPLOYMENT DIAGRAM

Component Diagram

Component diagrams are used to visualize the organization and relationships among components in a system.
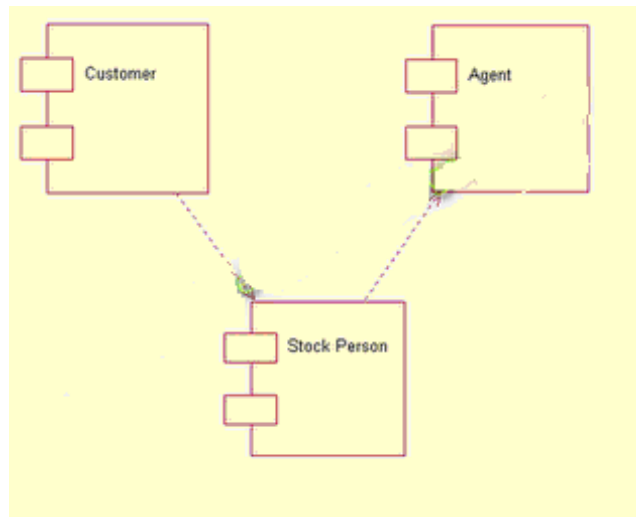
Fig.8.2.COMPONENT DIAGRAM

RESULT:

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

EXNO :6   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
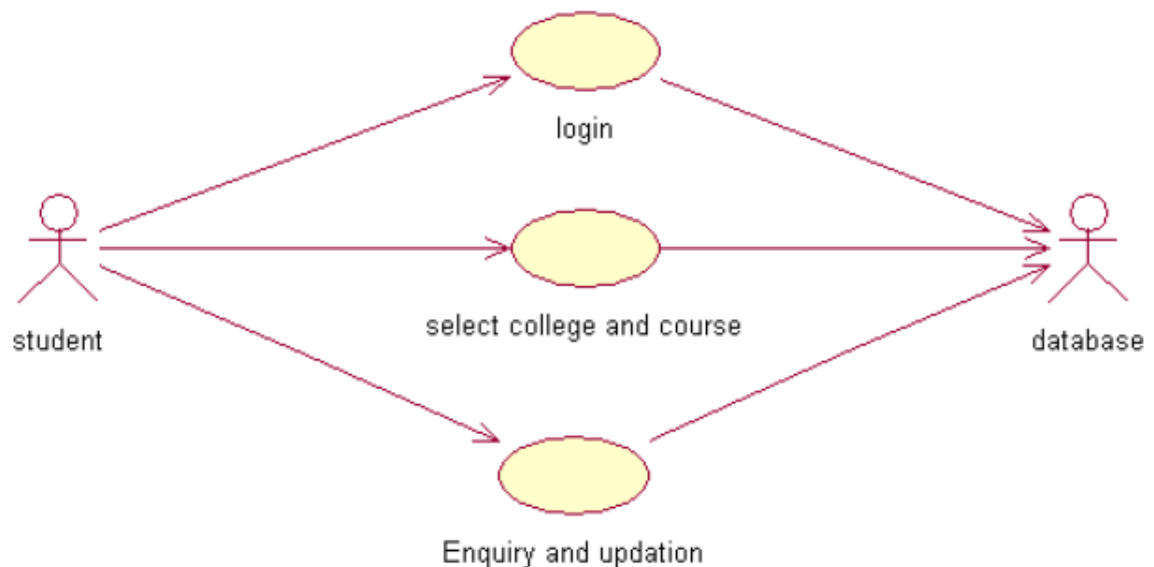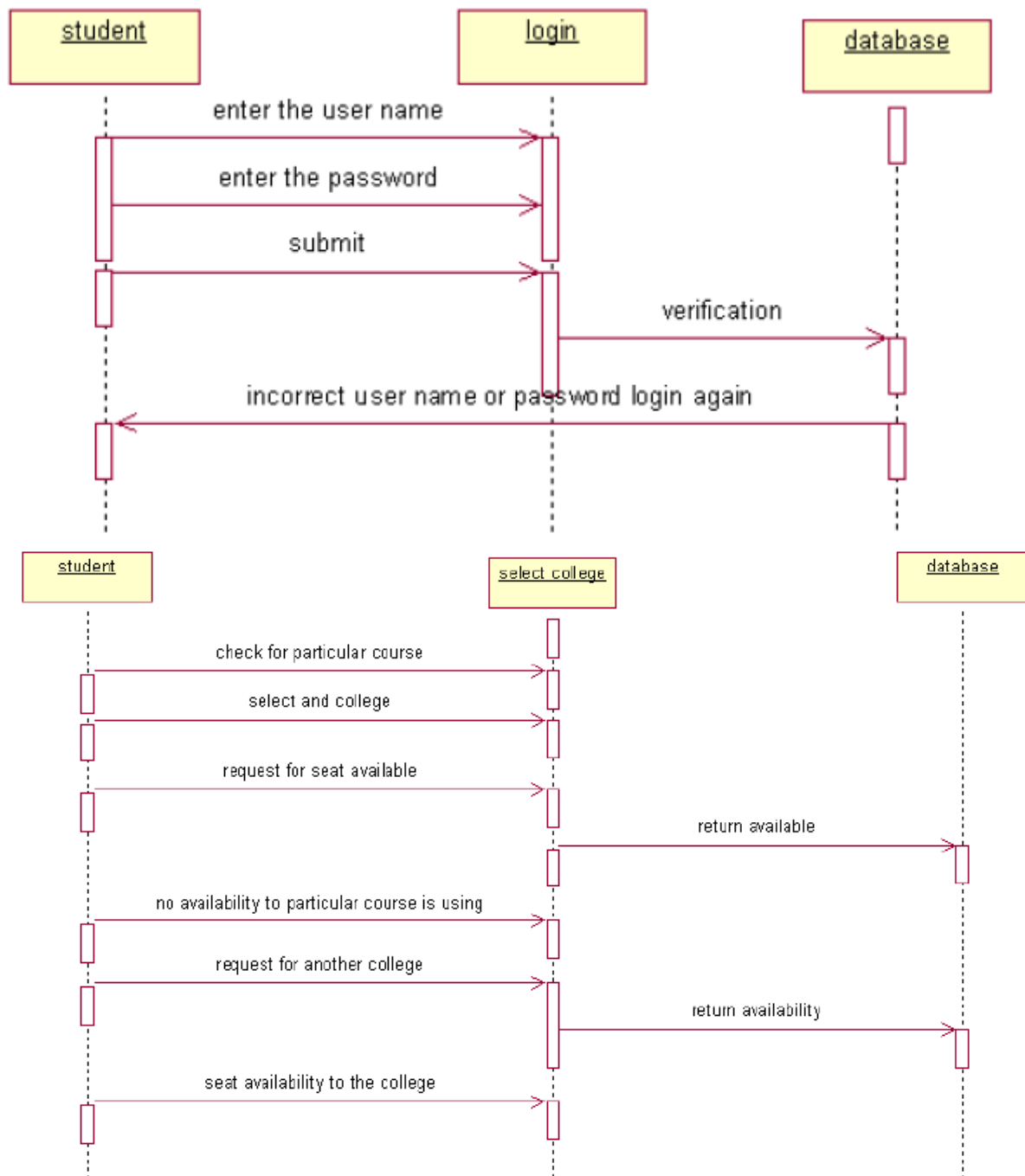

CLASS DIAGRAM:

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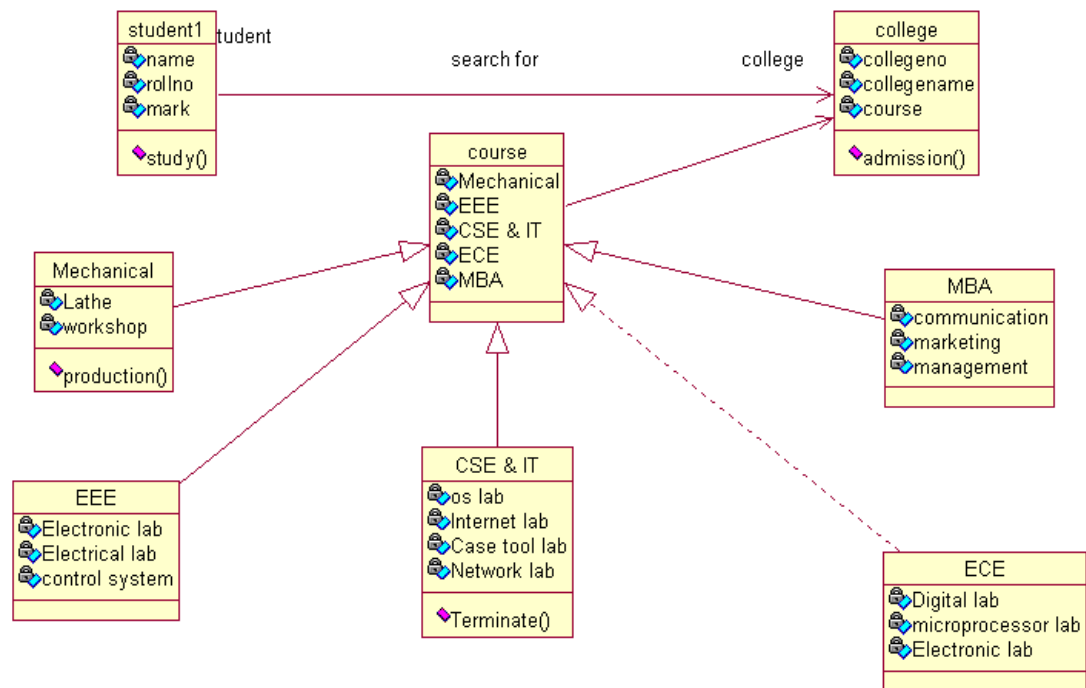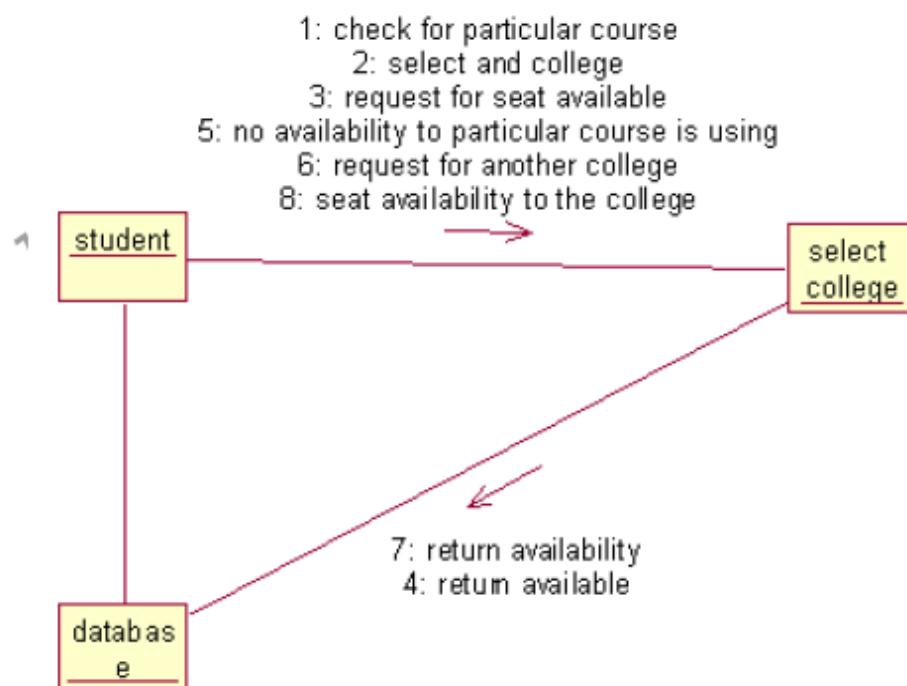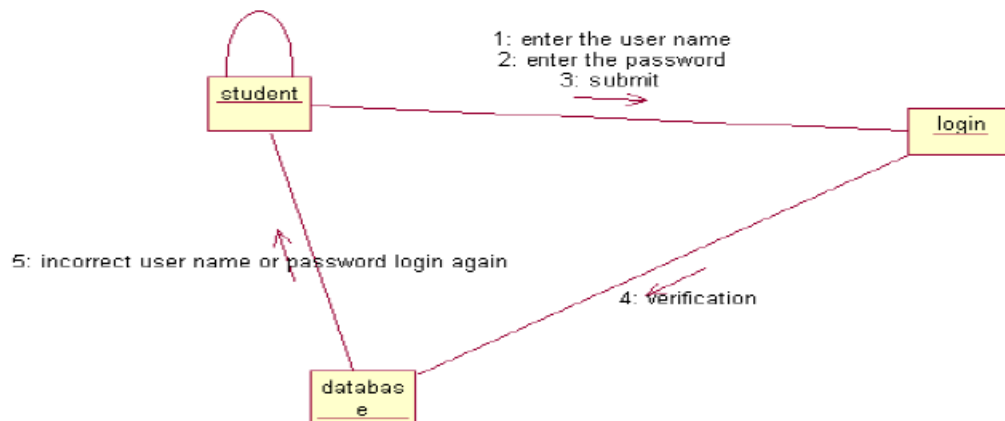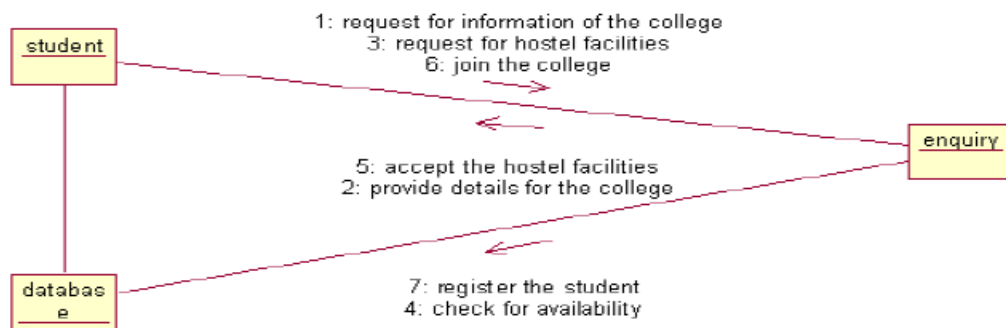
SEQUENCE DIAGRAM

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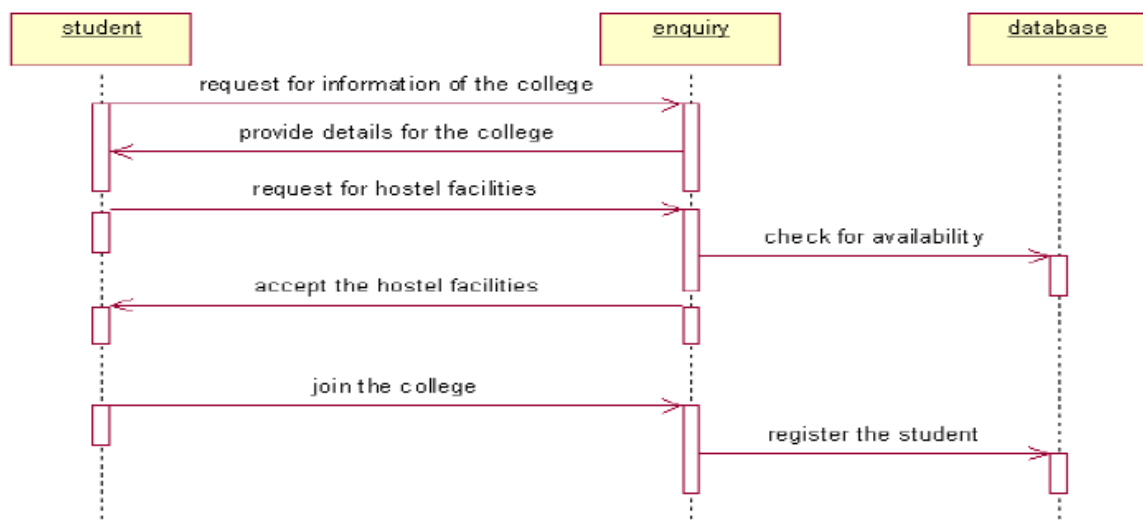
COLLOBORATIION DIAGRAM

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.
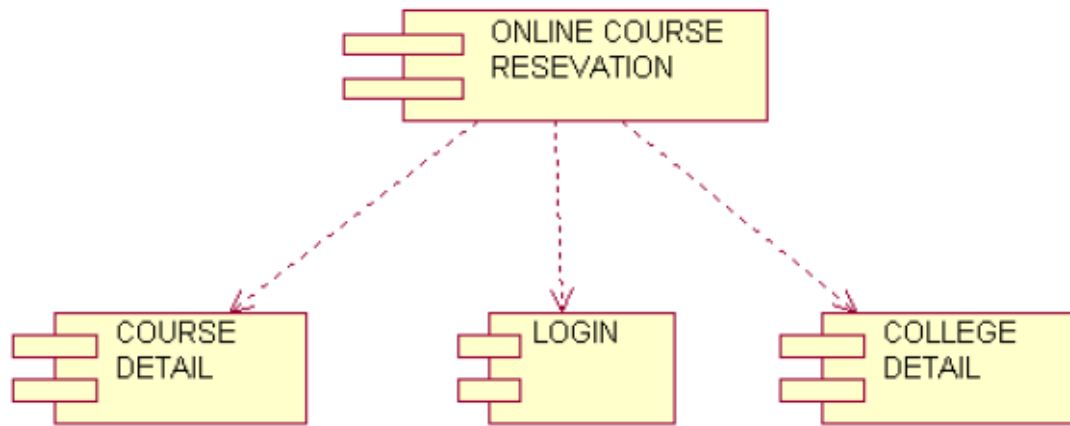
Diagram 1:

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

database

7: register the student
4: check for availability

Diagram 2:

student

1: enter the user name
2: enter the password
3: submit

login

5: incorrect user name or password login again

4: verification

database

Diagram 3:

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

database

ACTIVIY DIAGRAM

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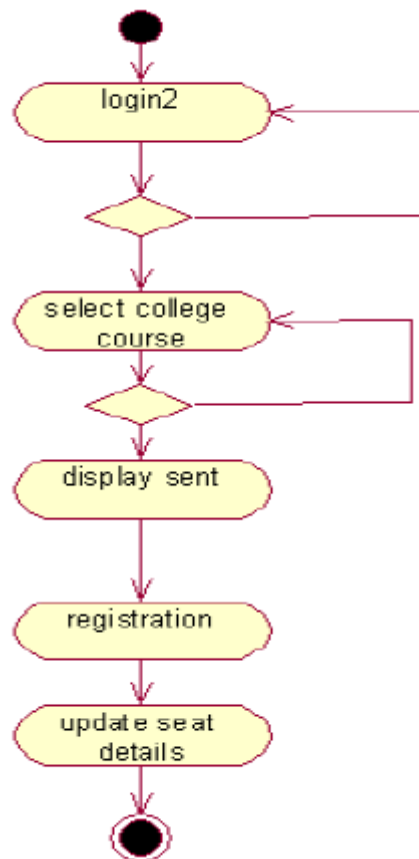

COMPONENT DIAGRAM

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
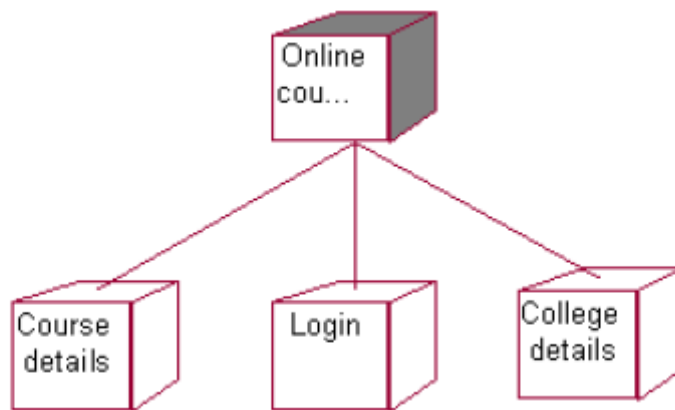

PACKAGE DIAGRAM

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
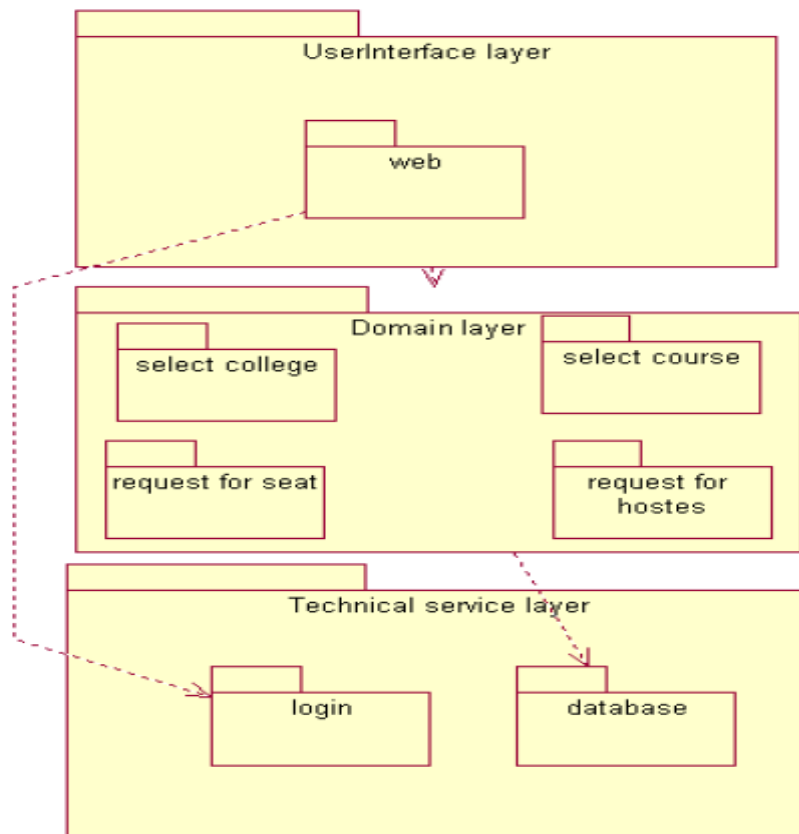
RESULT

Thus the project to develop online course reservation system was developed using Rational Rose Software and to implement the software in Visual Basic is done successfully.

EXNO:7 **E-TICKETING**

Date :

AIM

To develop the E-Ticketing System using Rational Rose Software and to implement the software in visual basic.

(I) PROBLEM ANALYSIS AND PROJECT PLANNING

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

(II) OVERALL DESCRIPTION
2.1 Functionality
The database should be act as an main role of the e-ticketing system it can be booking the ticket in easy way.

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

2.3 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.
2.4 Reliability
The system should be able to process the user for their corresponding request.

(III) 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 modelled. It is represented using stick figure
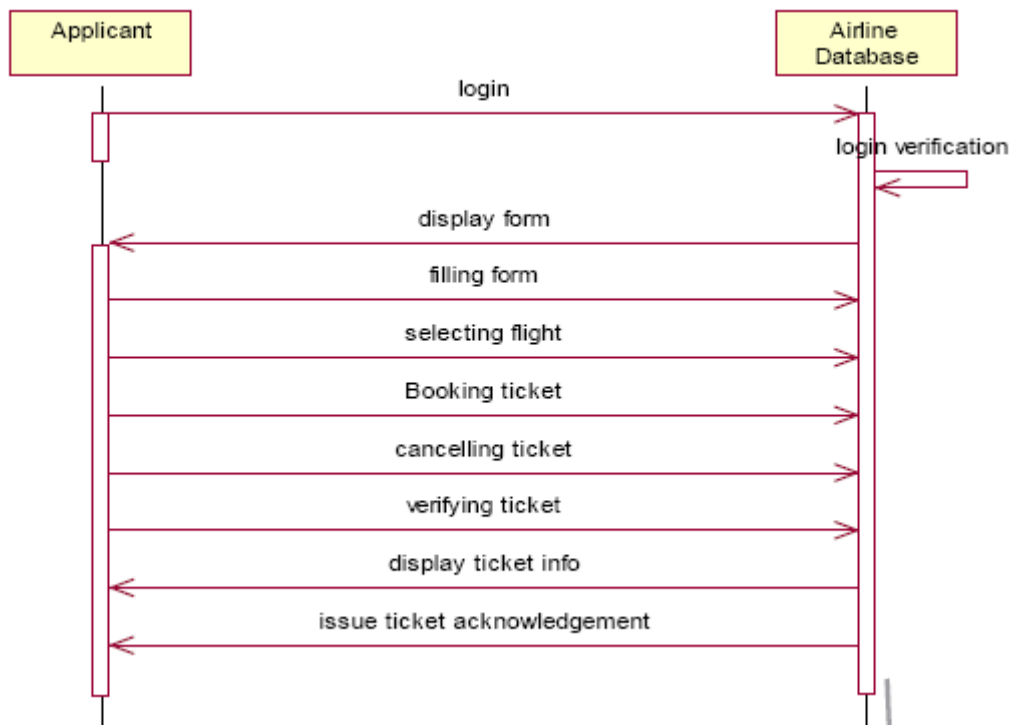
(IV)CLASS DIAGRAM

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 classname, middle compartment the attributes and the bottom compartment with operations.

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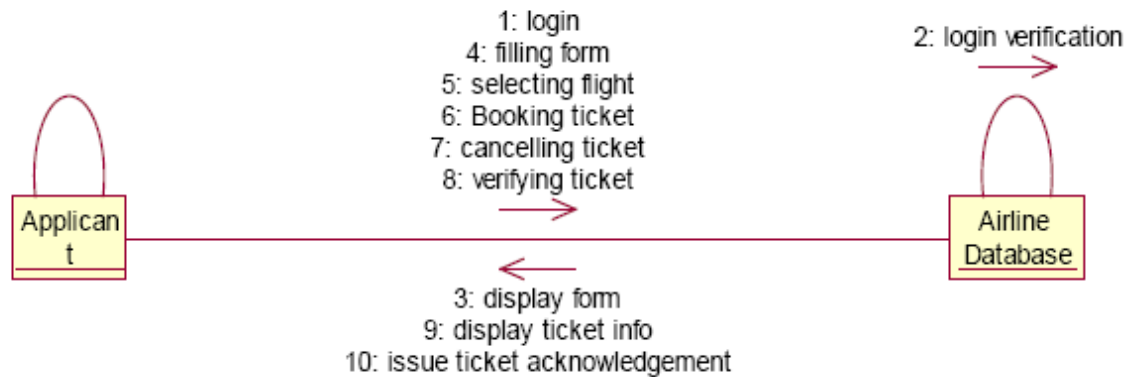1. Vertical dimension-represent time.
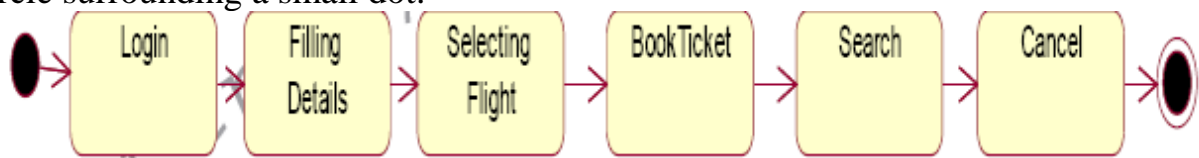2. Horizontal dimension-represent different objects.

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.
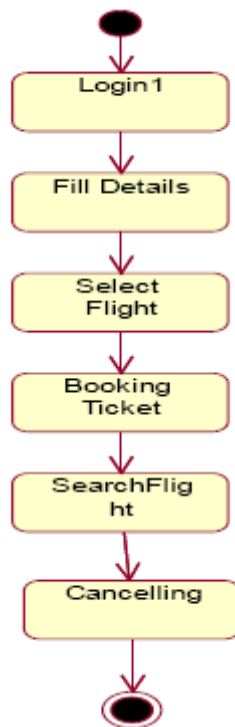
1: login
4: filling form
5: selecting flight
6: Booking ticket
7: cancelling ticket
8: verifying ticket

2: login verification

Applicant

Airline Database

3: display form
9: display ticket info
10: issue ticket acknowledgement

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.

Login | Filling Details | Selecting Flight | Book Ticket | Search | Cancel

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.
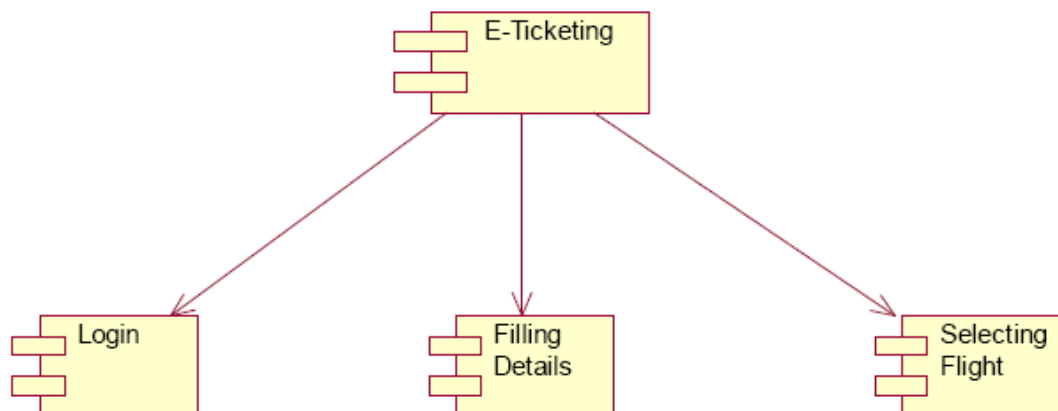
RESULT

Thus the project to develop E-Ticketing system using Rational Rose Software and to implement the project in Visual Basic is done successfully.

Ex no:8   SOFTWARE PERSONNEL MANAGEMENT SYSTEM

Ex. No:8

SOFTWARE PERSONNAL MANAGEMENT

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)SOFTWARE REQUIREMENT SPECIFICATION:

2.1SOFTWARE INTERFACE

• **Front End Client** - The applicant and Administrator online interface is built using

JSP and HTML. The HR's local interface is built using Java.

• **Server** - Glassfish application server(SQL Corporation).

• **Back End** - SQL database.

2.2HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

(III)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

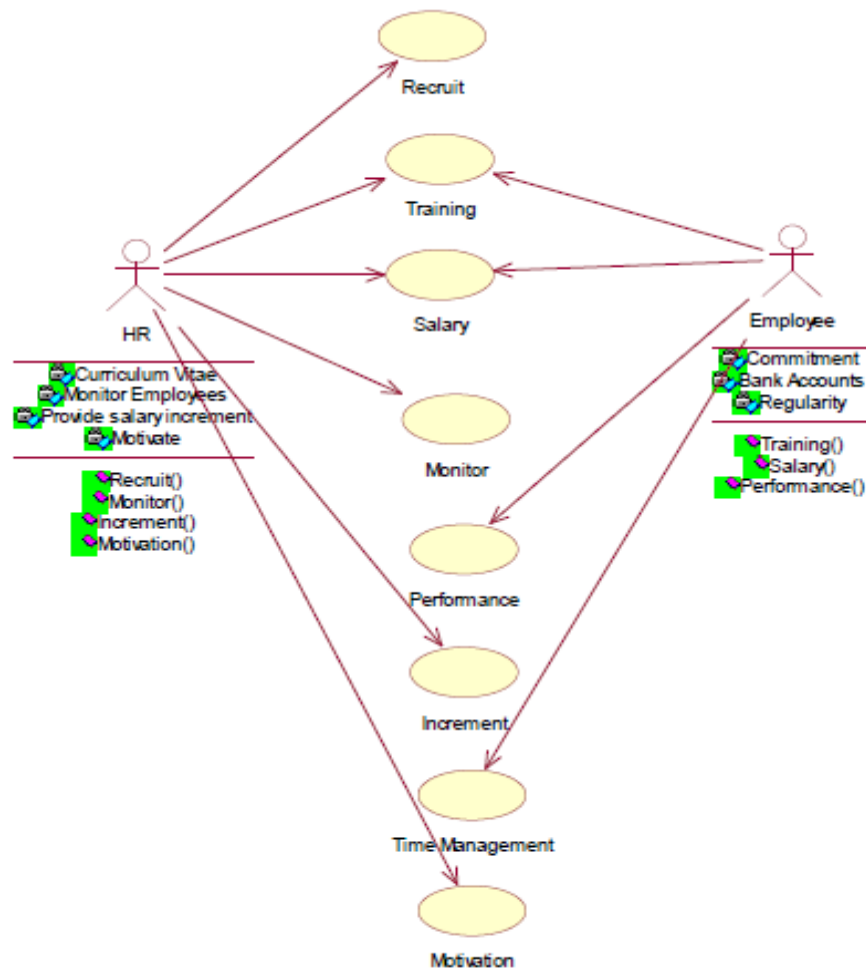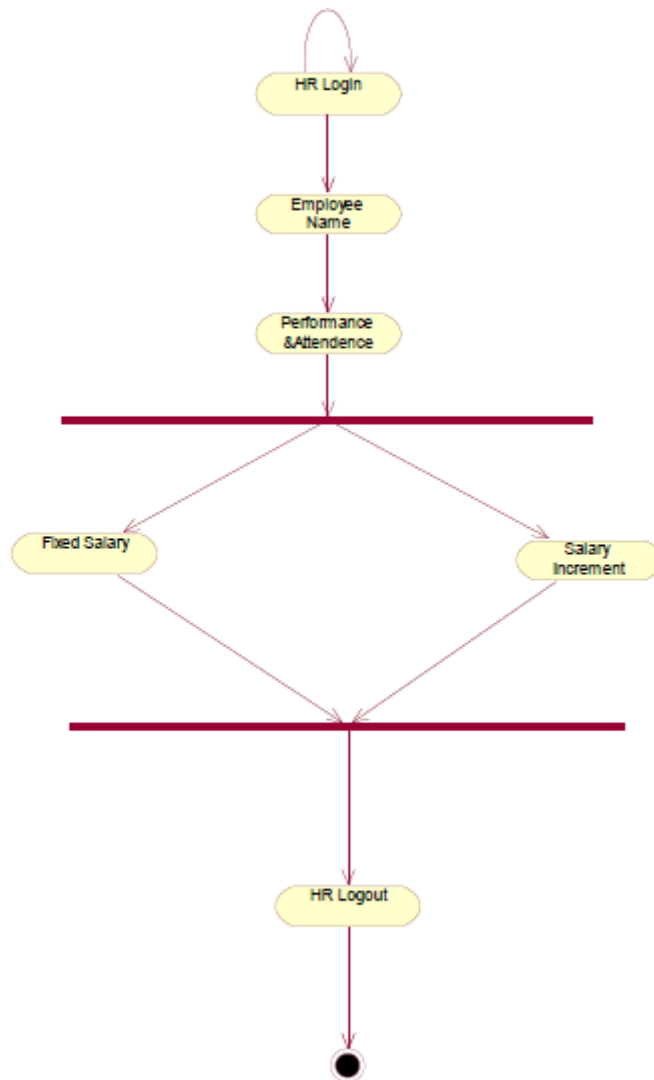order to complete a given task within the required period.

Fig.3. USE CASE DIAGRAM

(IV) 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.

q

Fig.4. ACTIVITY DIAGRAM

(V) 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.
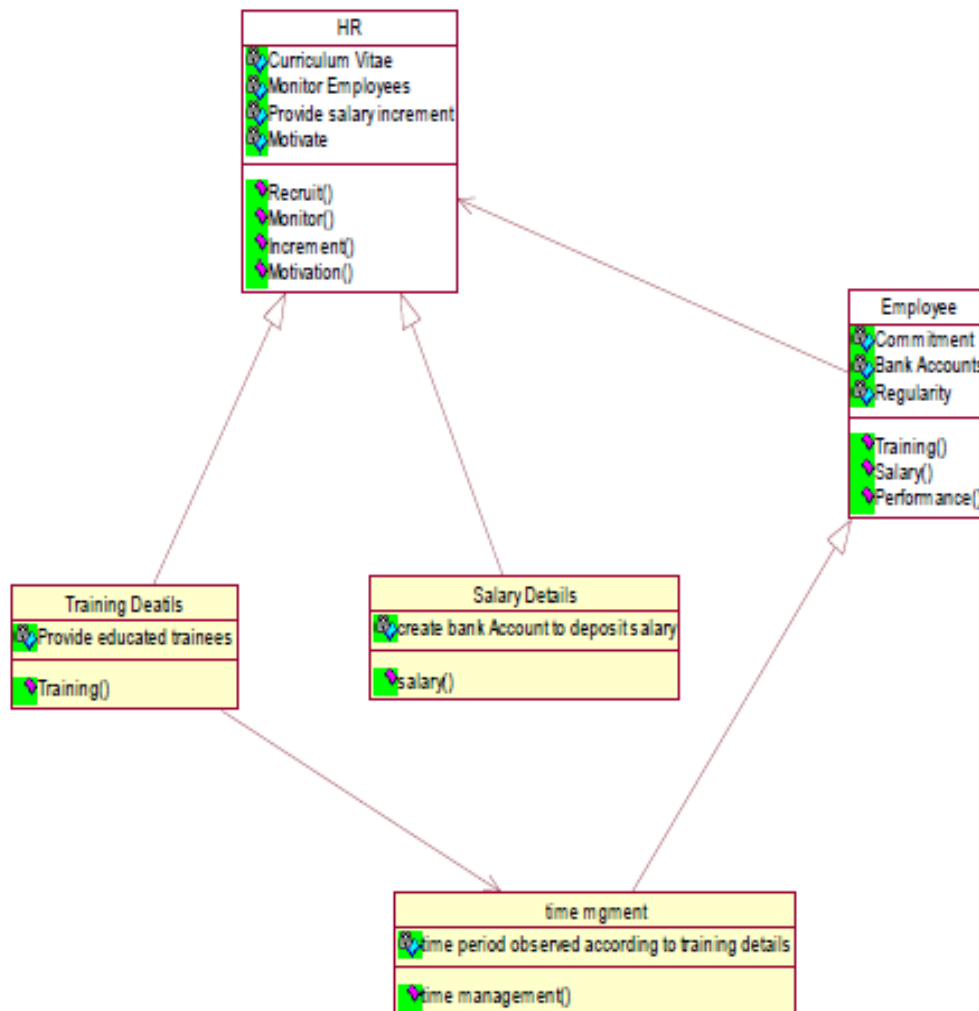

Fig.5.CLASS DIAGRAM

(VI) 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.
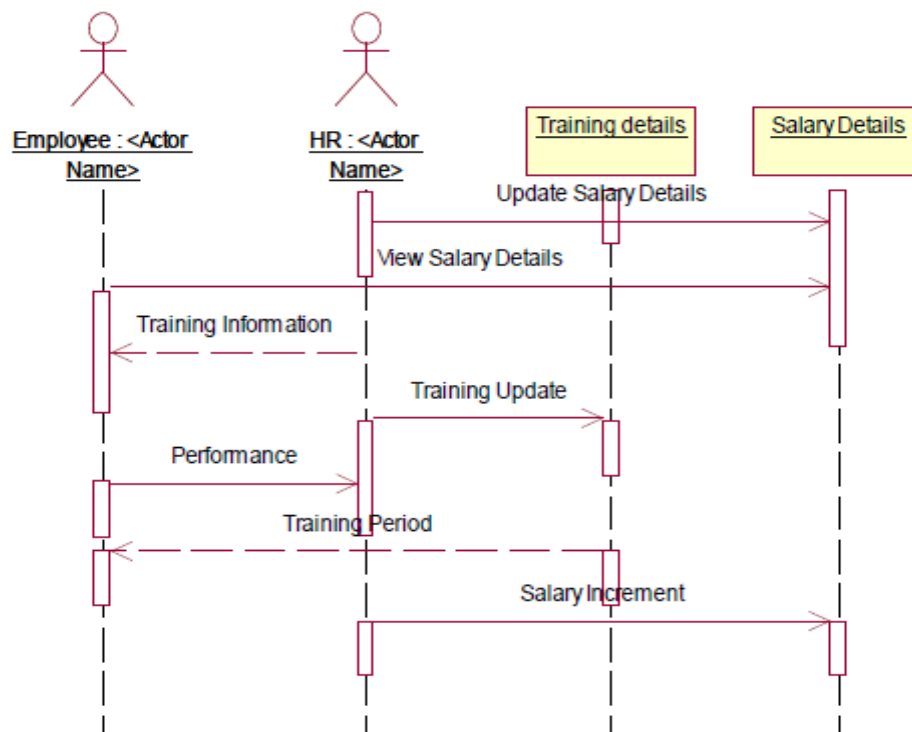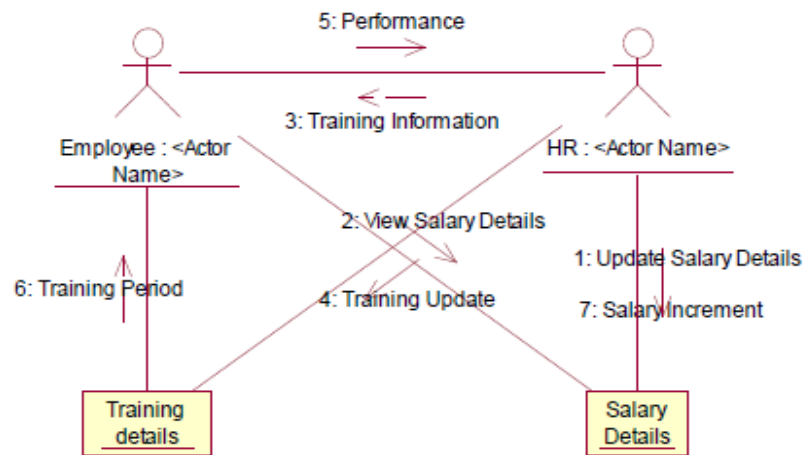

Fig.6.1.SEQUENCE DIAGRAM

Fig.6.2.COLLABORATION DIAGRAM

(VII) 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.
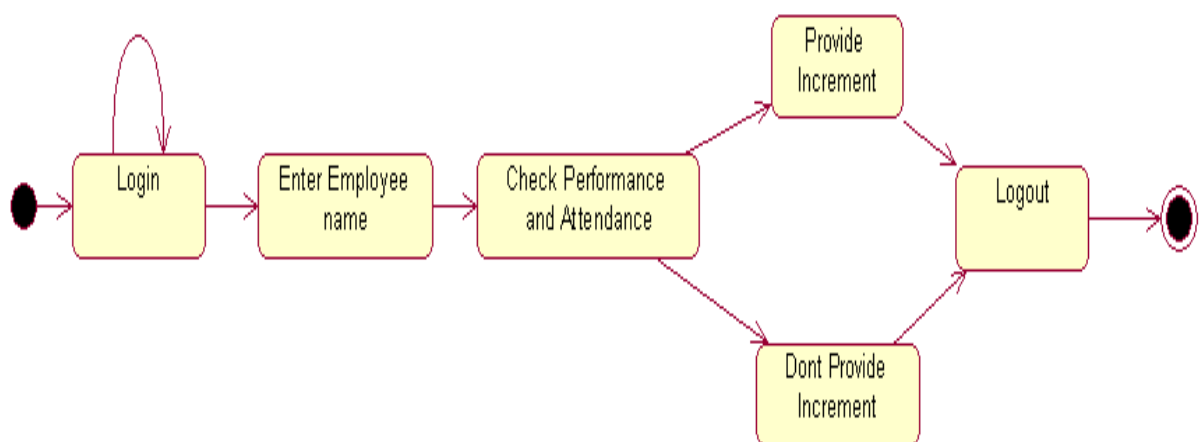

Fig.7.STATE TRANSITION DIAGRAM

(VIII) 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.
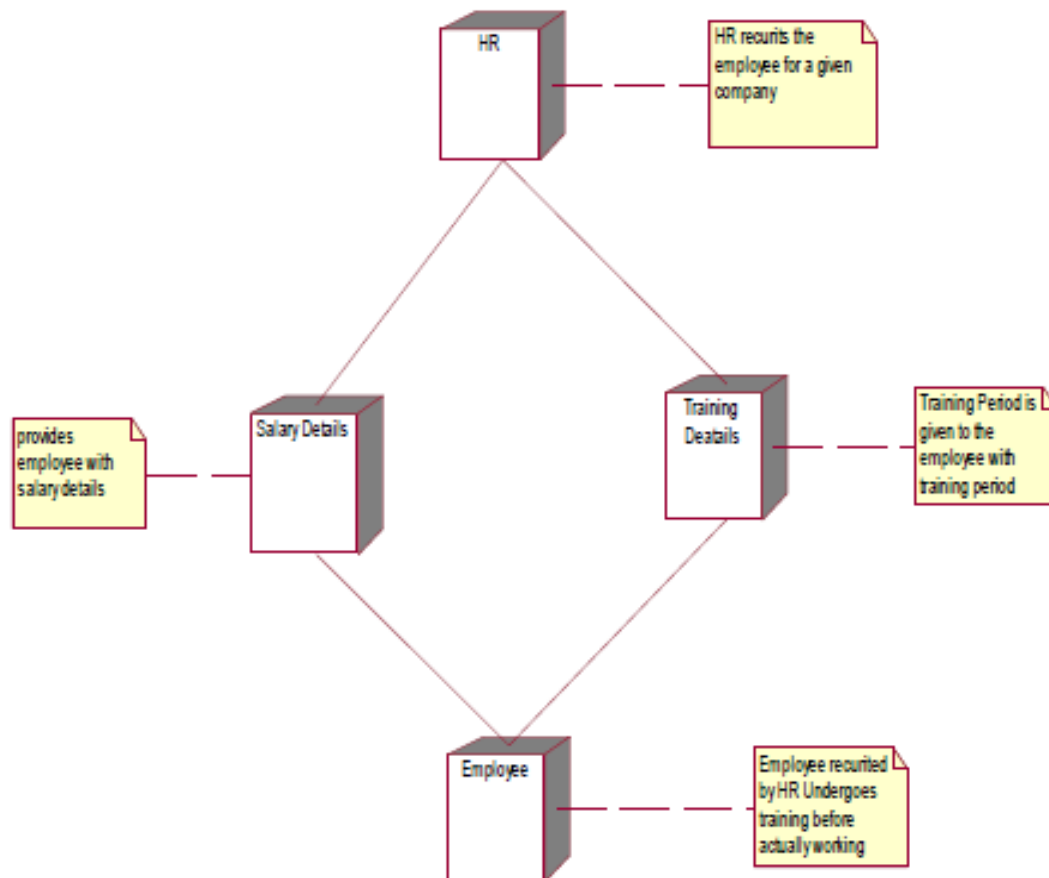

Fig.8.1.DEPLOYMENT DIAGRAM

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
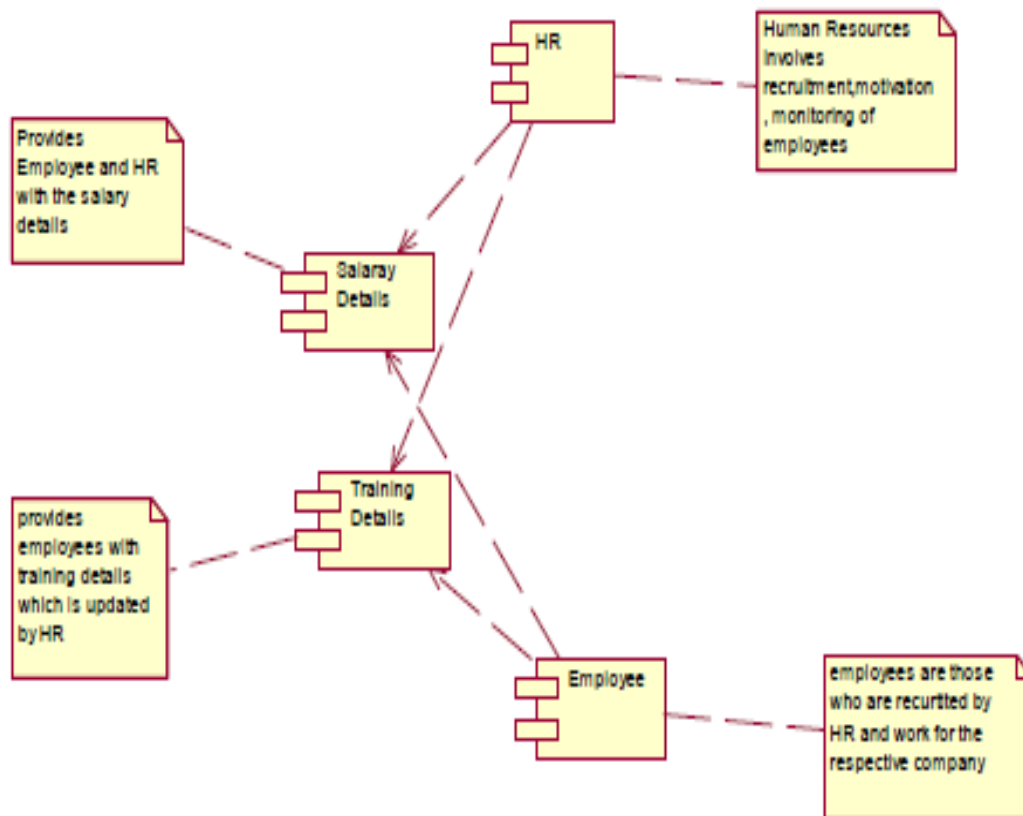

Fig.8.2.COMPONENT DIAGRAM

RESULT:

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

CREDIT CARD PROCESSING

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)SOFTWARE REQUIREMENT SPECIFICATION:

2.1 PRODUCT PERSPECTIVE

This solution involves signing up for a free Business Account. Once this is done and the e-commerce site is properly configured, you can accept payments from Visa, MasterCard, Amex, and Discover cards payments.

2.2SOFTWARE INTERFACE

• **Front End Client** - The applicant and Administrator online interface is built using

JSP and HTML. The Administrators's local interface is built using Java.

• **Web Server** - Glassfish application server(SQL Corporation).

• **Back End** - SQL database.

2.3HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

(III)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).
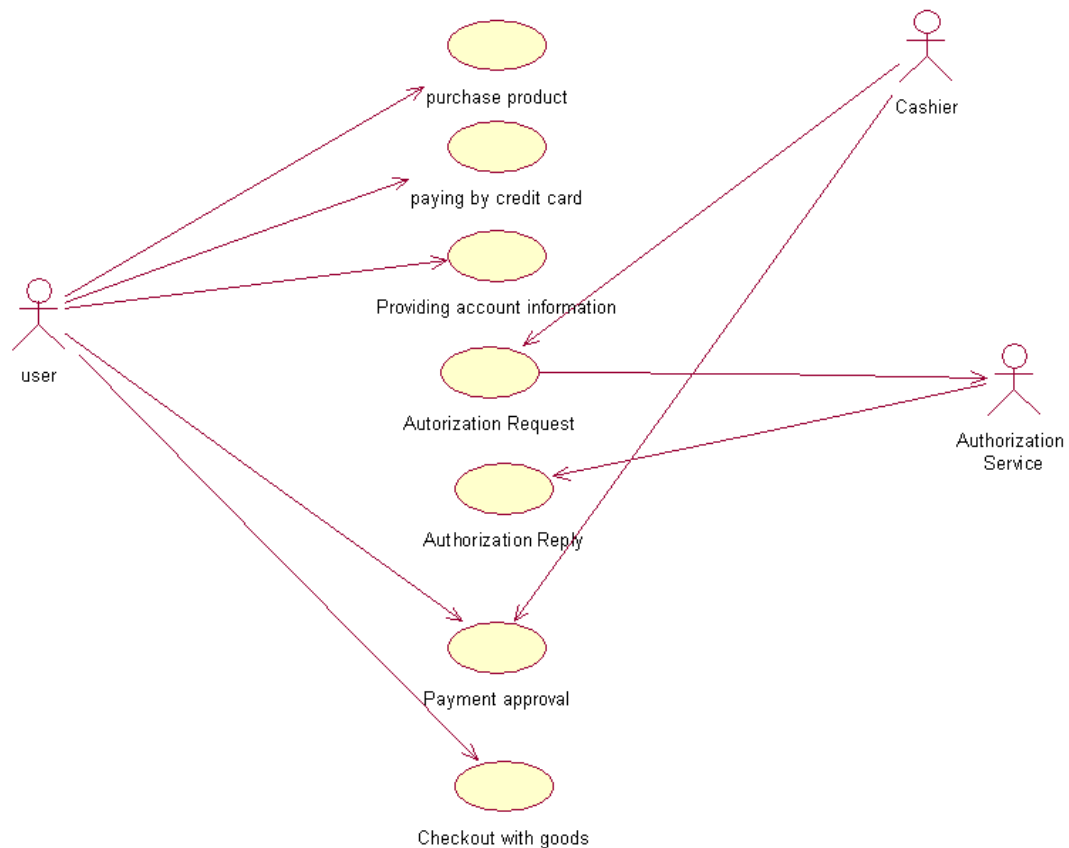

Fig.3. USECASE DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(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.

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

They are

Cashier
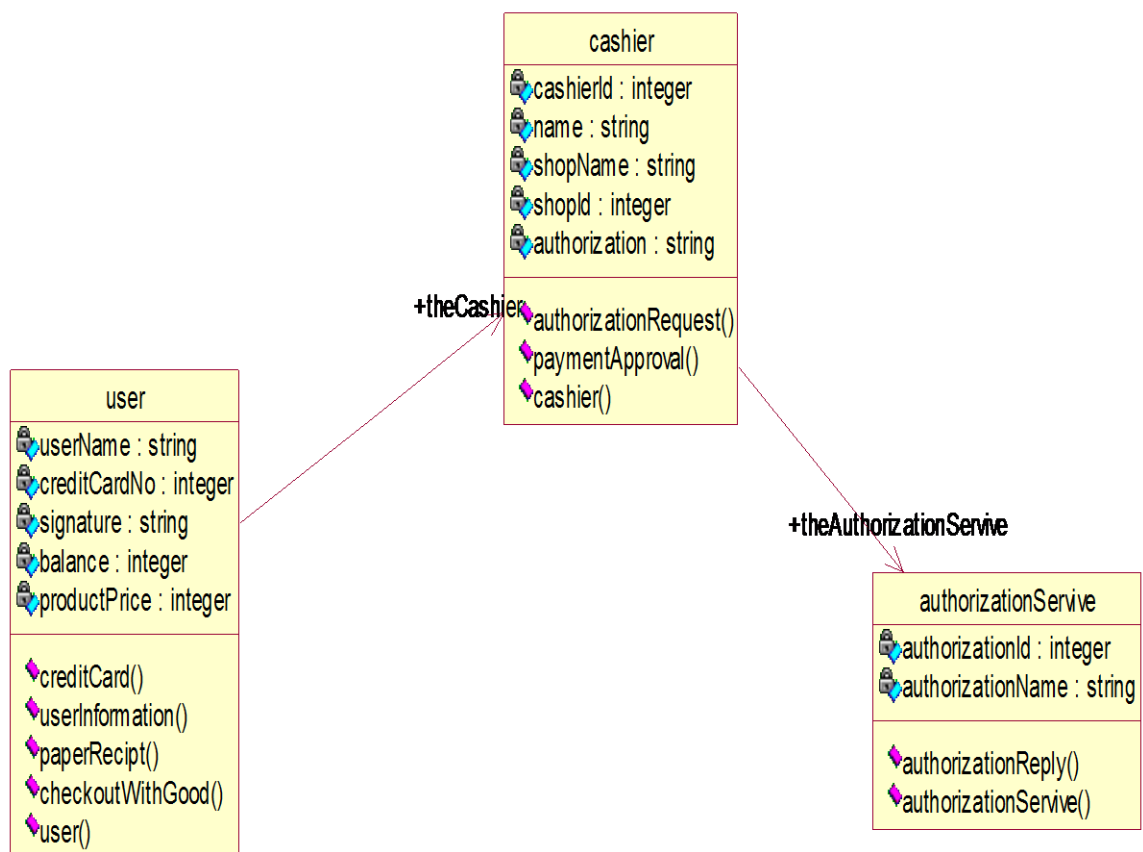
User

Authorization Service

Fig.4.CLASS DIAGRAM

(V) INTERACTION DIAGRAM:
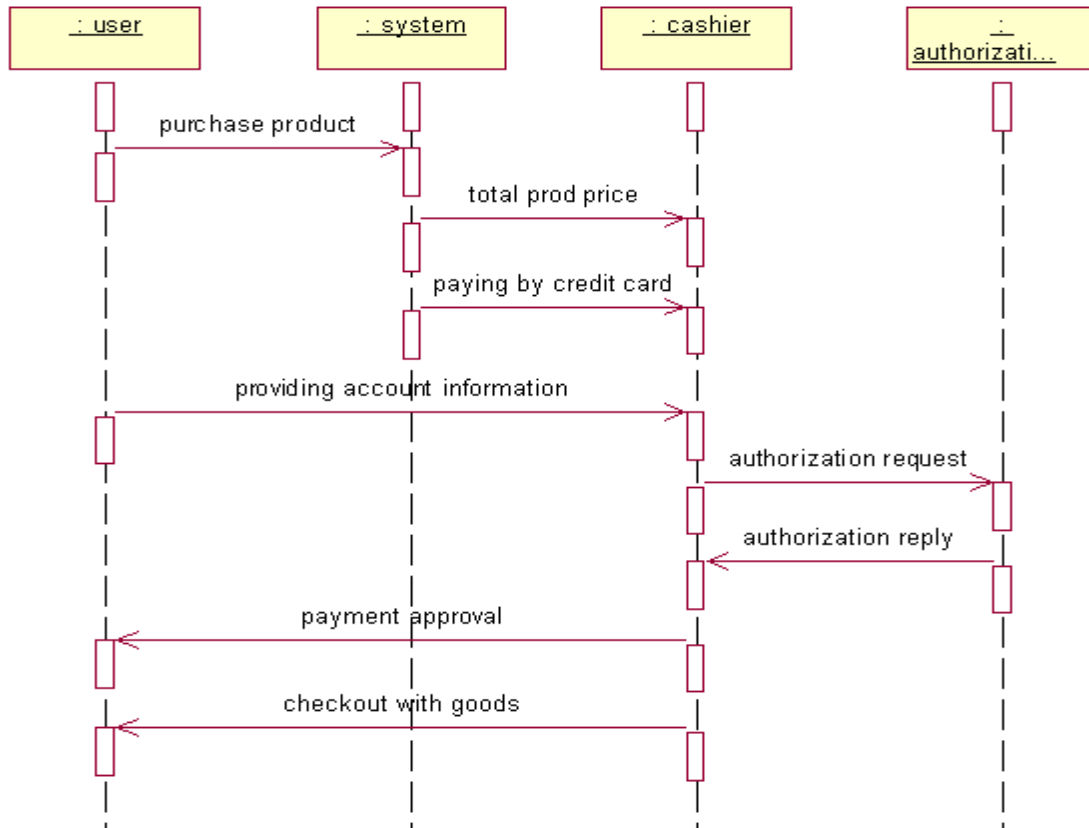
Fig.5.1.SEQUENCE DIAGRAM
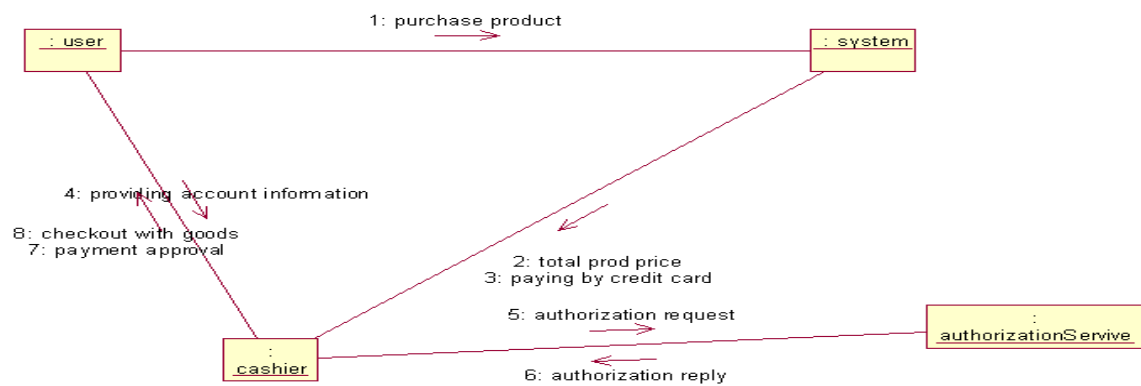
Fig.5.2.COLLABORATION DIAGRAM
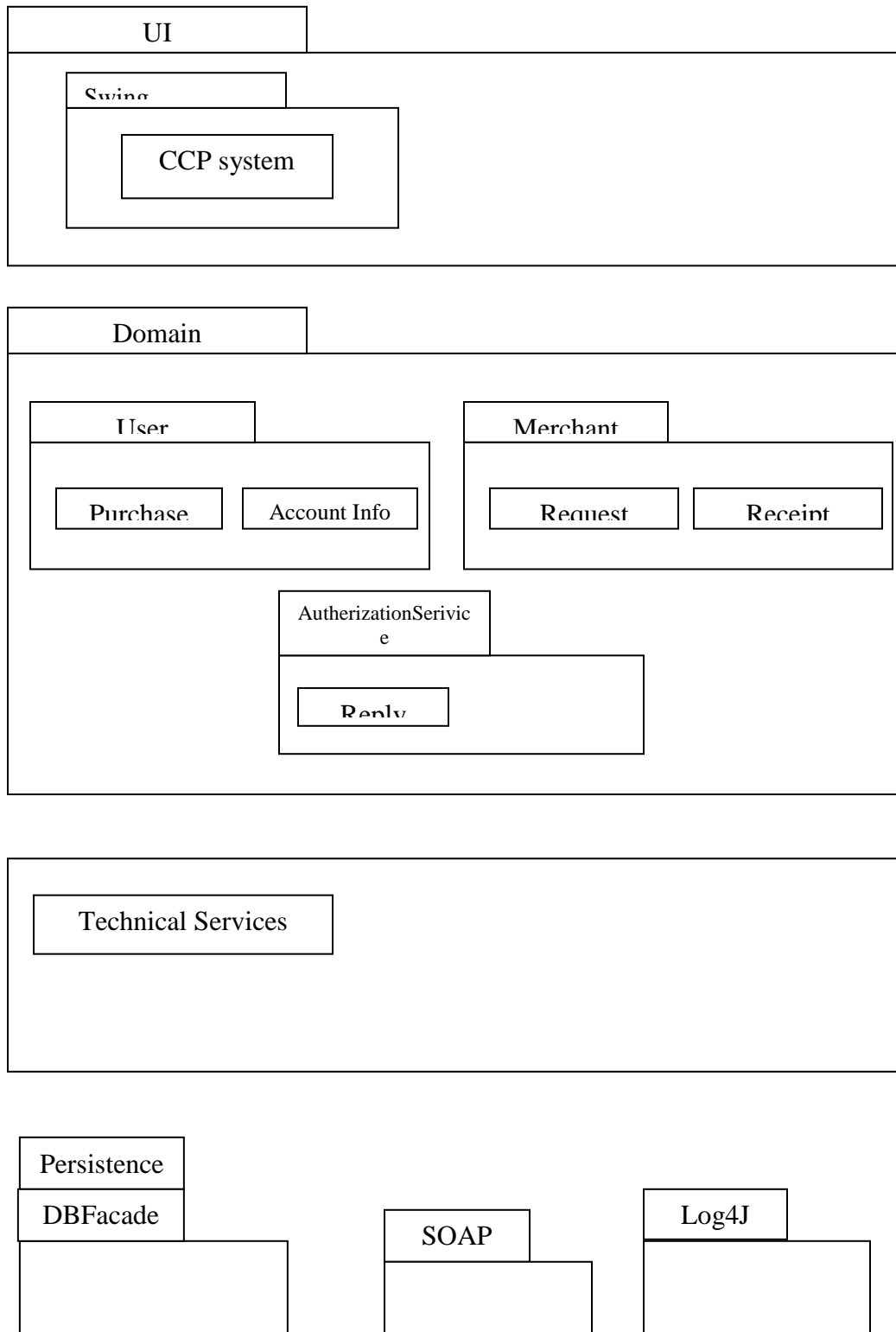
(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM

UI

Swing

CCP system

Domain

User

Purchase | Account Info

Merchant

Request | Receipt

AutherizationSerivice

Reply

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
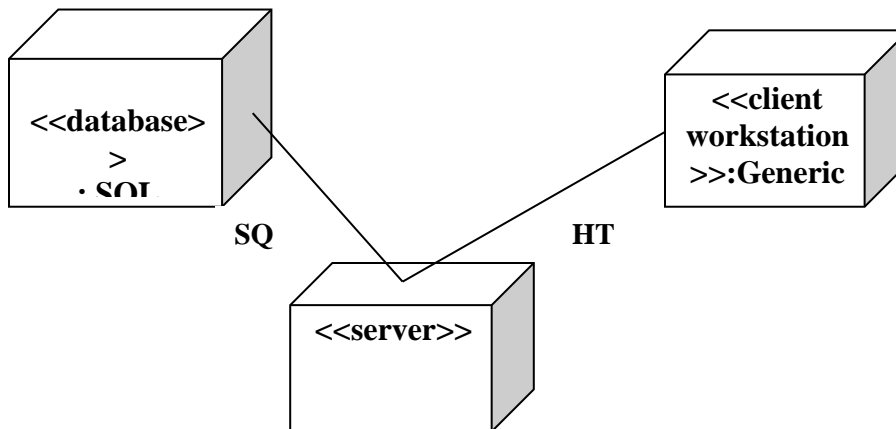

Fig.7.1.DEPLOYMENT DIAGRAM

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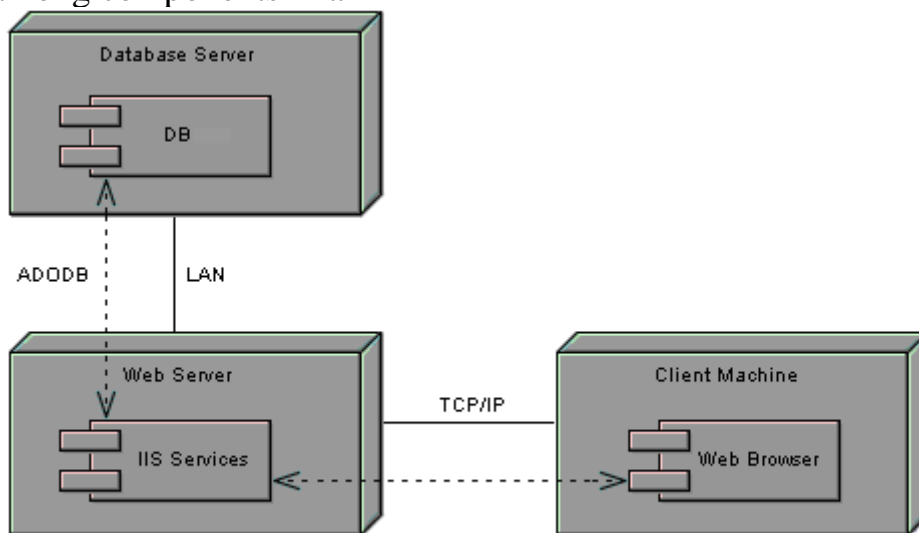

Fig.7.2.COMPONENT DIAGRAM

RESULT:

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

Ex. No:10

Date:

E-BOOK MANAGEMENT SYSTEM

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) SOFTWARE RESOURCE SPECIFICATION:

2.0 OVERALL DESCRIPTION

Itwill describe major role of the system components and inter-connections.

2.1PRODUCT PERSPECTIVE

The ORS acts as an interface between the user and the 'e-book manager'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the books or magazines.

2.2SOFTWARE INTERFACE

Front End Client - The Student and Librarian online interface is built using JSP and HTML. The Librarians local interface is built using Java.

Web Server - Glassfish application server (Oracle Corporation).

Back End - Oracle database

2.3HARDWARE INTERFACE

The server is directly connected to the client systems. The client systems have access to the database in the server.

(III)USE-CASE DIAGRAM:

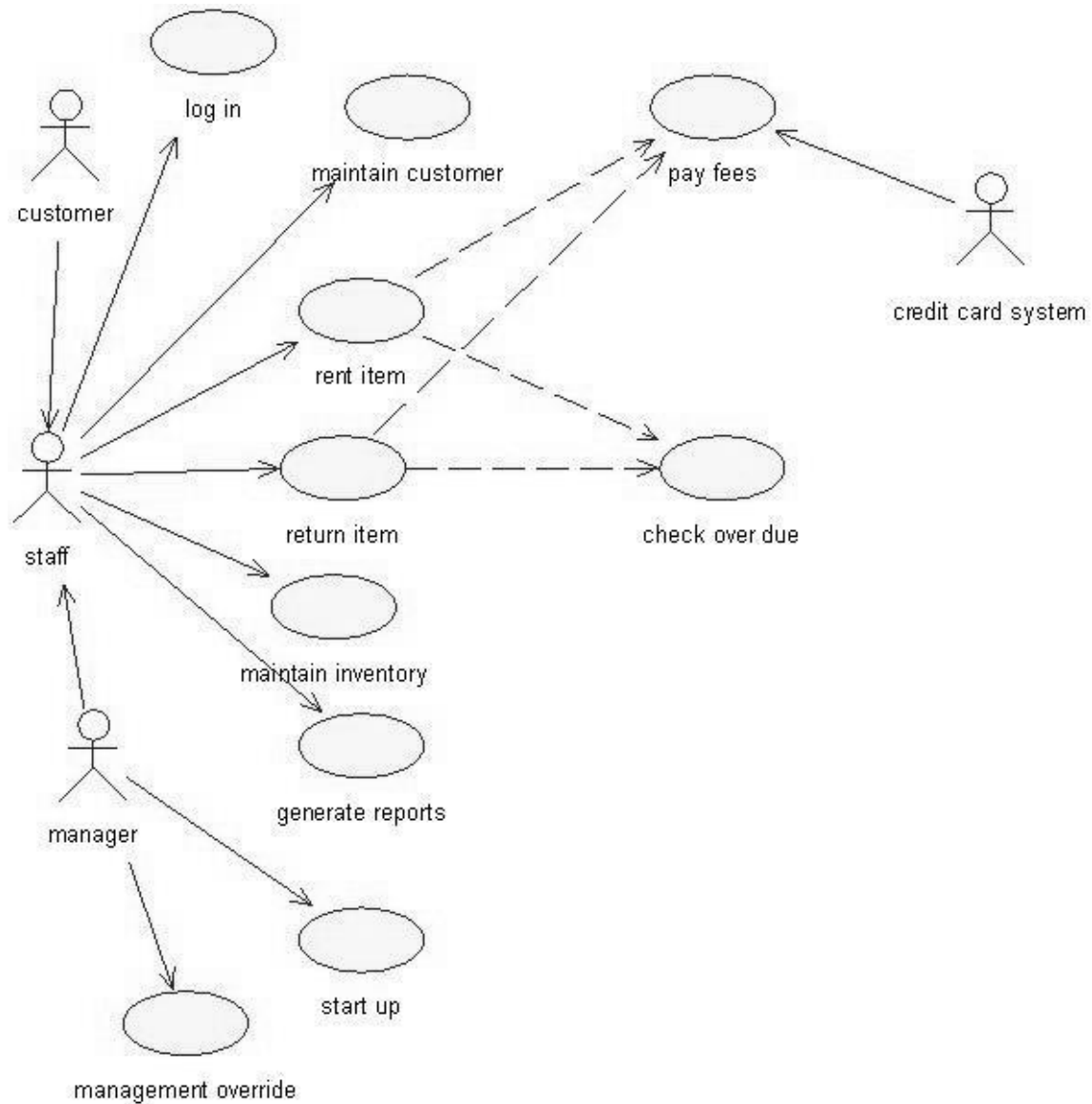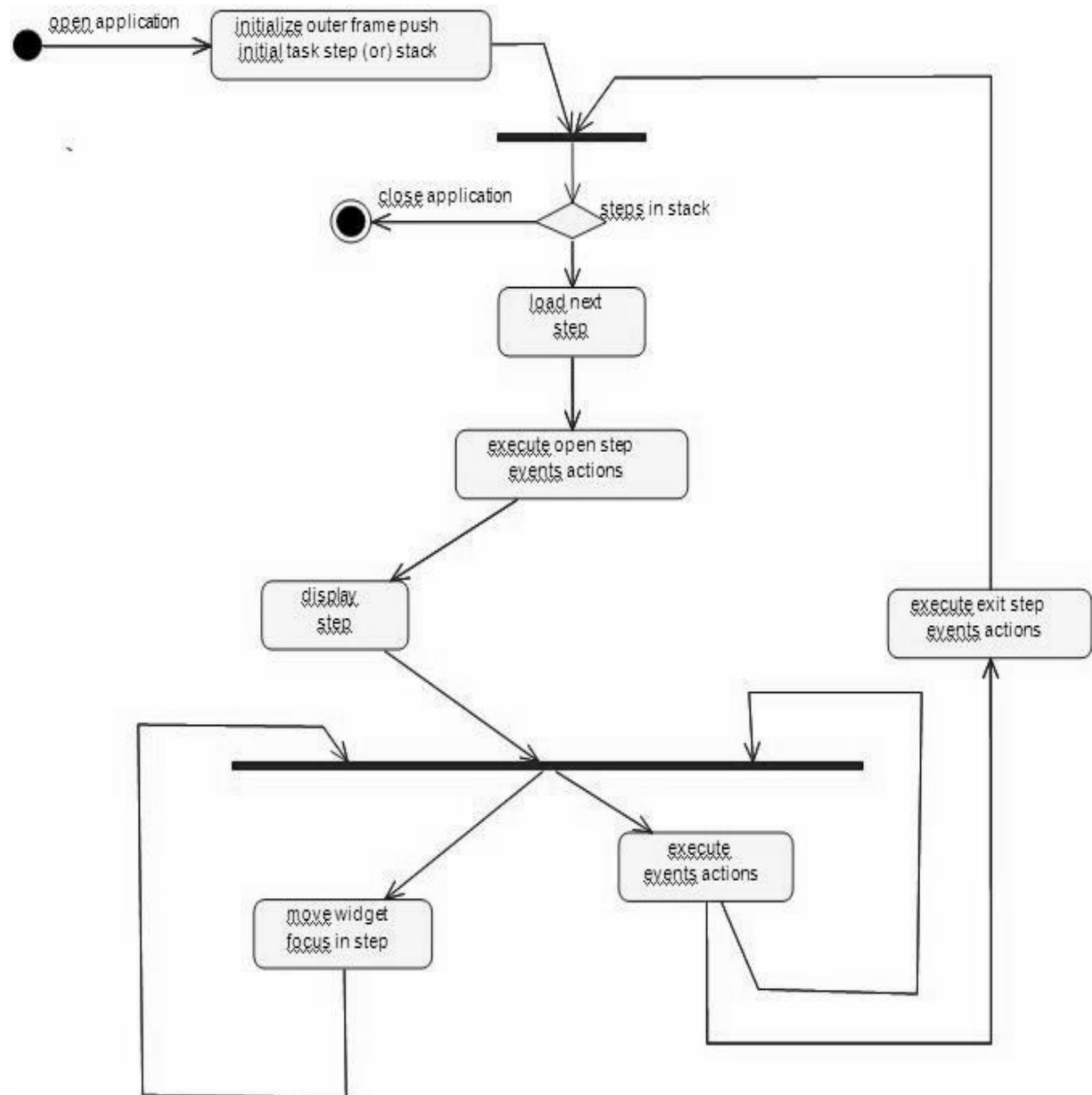**Fig.3.USE-CASE DIAGRAM FOR E-BOOK SYSTEM**

(IV) ACTIVITY DIAGRAM:

(V)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.
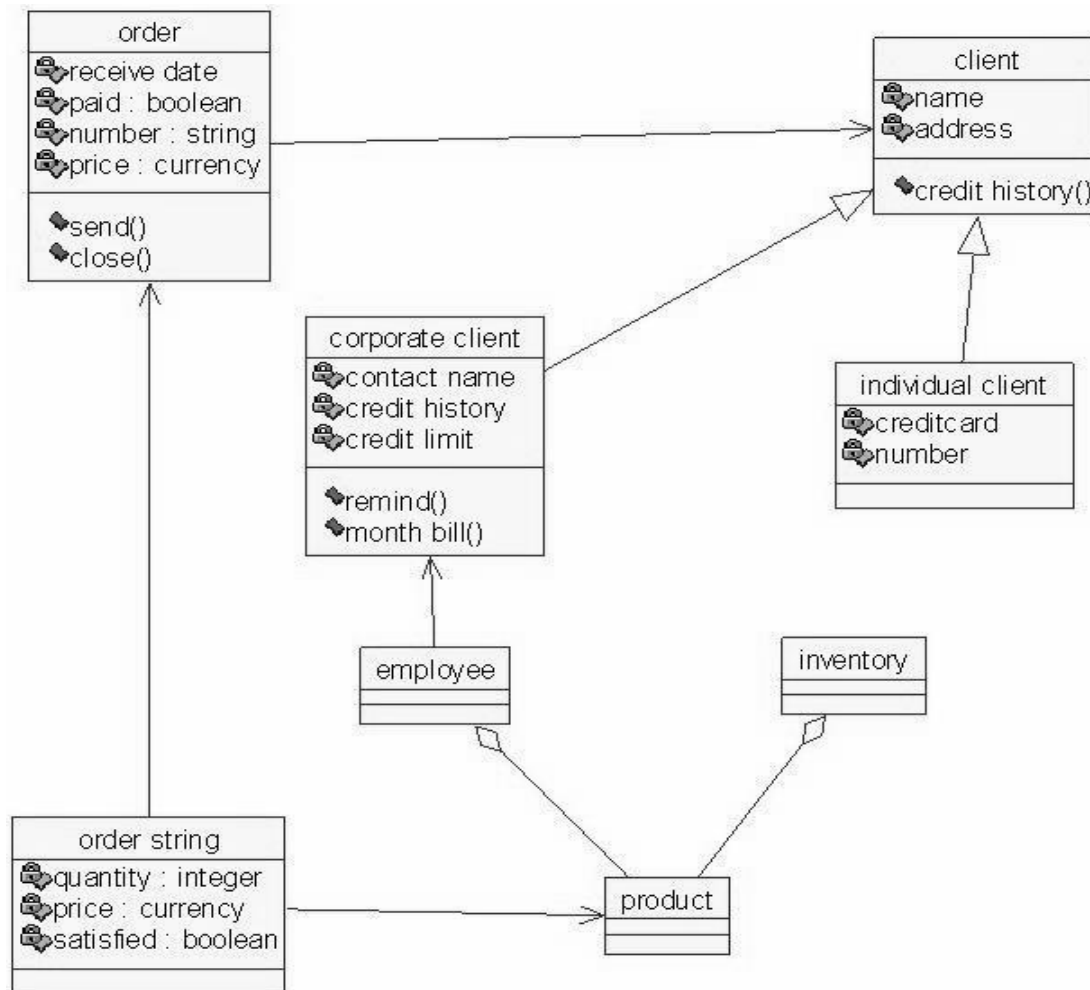

Fig.5.CLASS DIAGRAM FOR E-BOOK SYSTEM

(VI) 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.
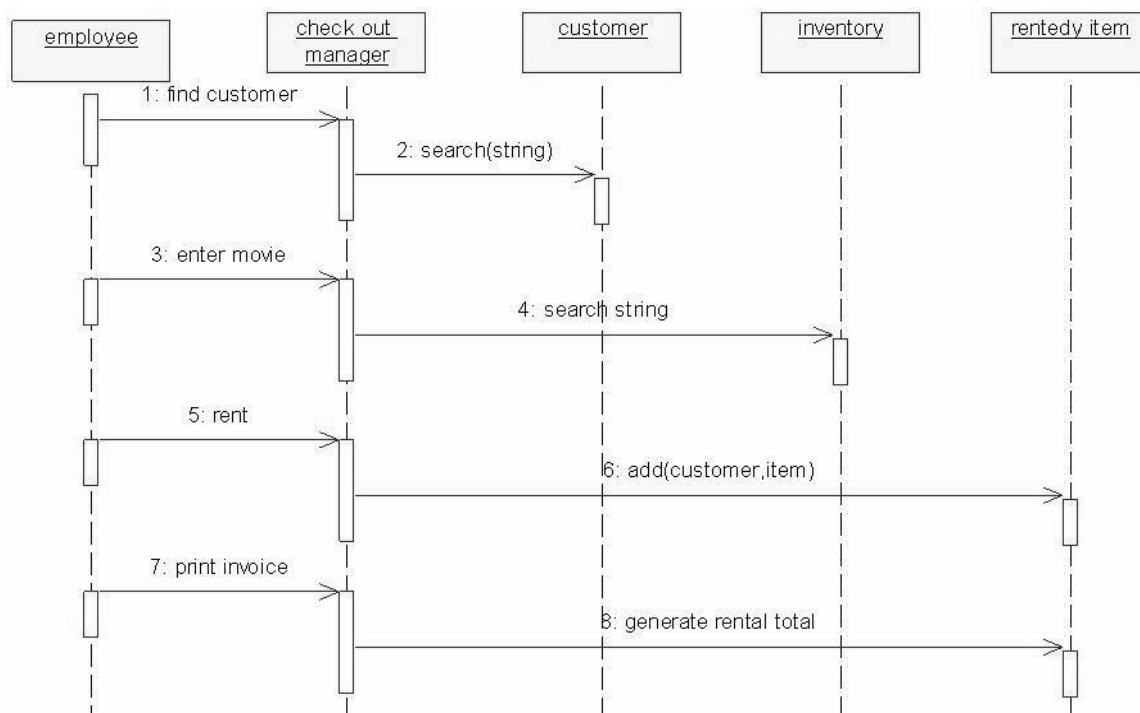
Fig.6.1.SEQUENCE DIAGRAM
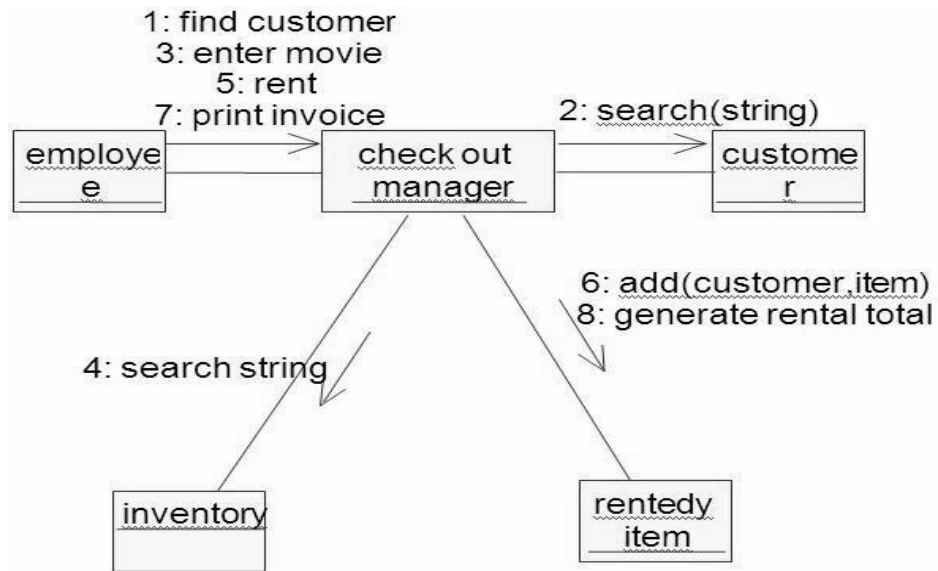
Fig.6.2.COLLABORATION DIAGRAM

(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.
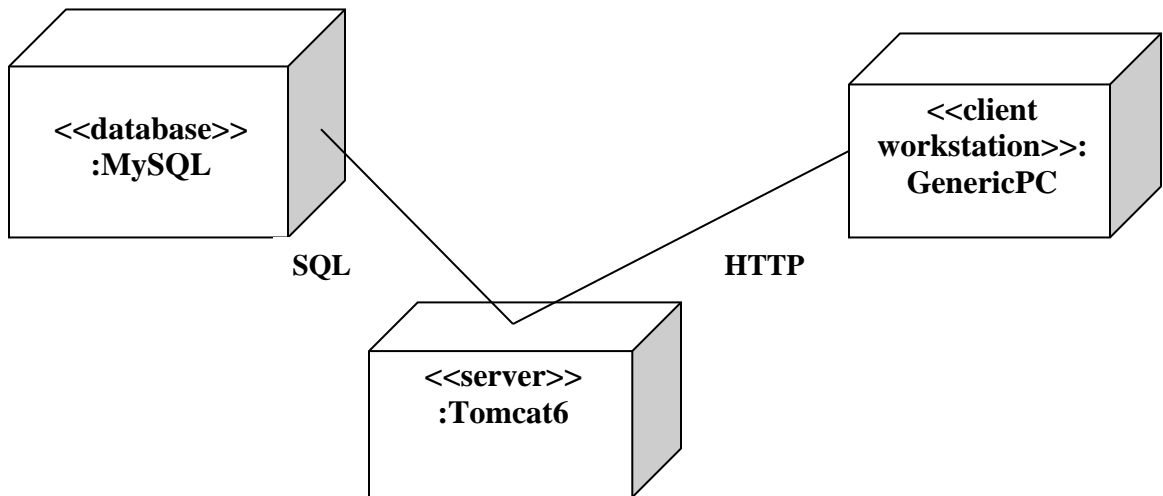
Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM
Component diagrams are used to visualize the organization and relationships among components in a system.
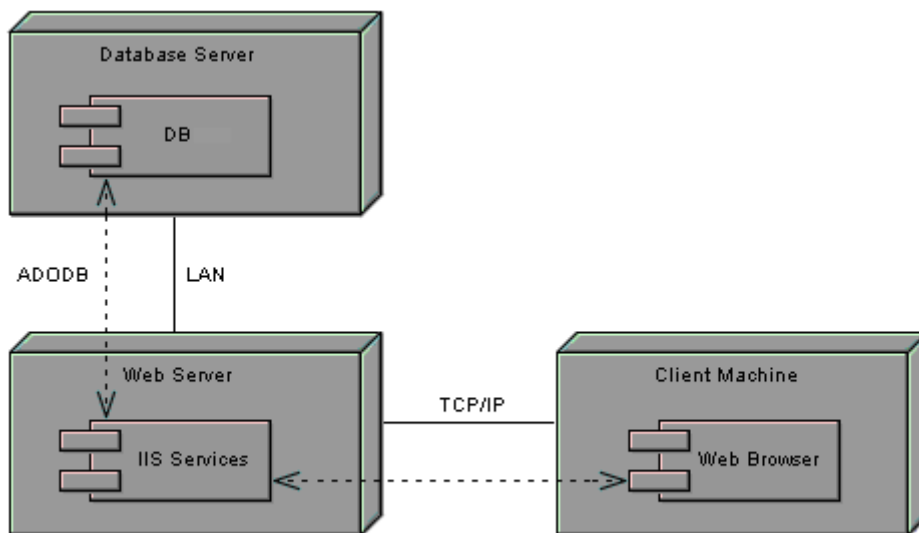
Fig.8.2.COMPONENT DIAGRAM

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

Ex.No: 11

Date:

<div align="center">

RECRUITMENT SYSTEM

</div>

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)SOFTWARE REQUIREMENT SPECIFICATION:

2.1 PRODUCT PERSPECTIVE

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

2.2SOFTWARE INTERFACE

      • **Front End Client** - The applicant and Administrator online interface is built using JSP and HTML. The Administrators's local interface is built using Java.

      • **Web Server** - Glassfish application server (SQL Corporation).

• **Back End** - SQL database.

2.3HARDWARE INTERFACE

The server is directly connected to the client systems. The client

systems have access to the database in the server.

(III)USECASE DIAGRAM:

The Recruitment Automation system use cases are:

Notify Vacancy

Fill referral Form

<<Uses>>

HR Head

Selection of Eligible
Candidates

<<Extends>>

Employee

Interview Process

Intimation to Selected
Candidates

Candidate

Accept/Decline Offer

<<Extends>>

Award Bonus
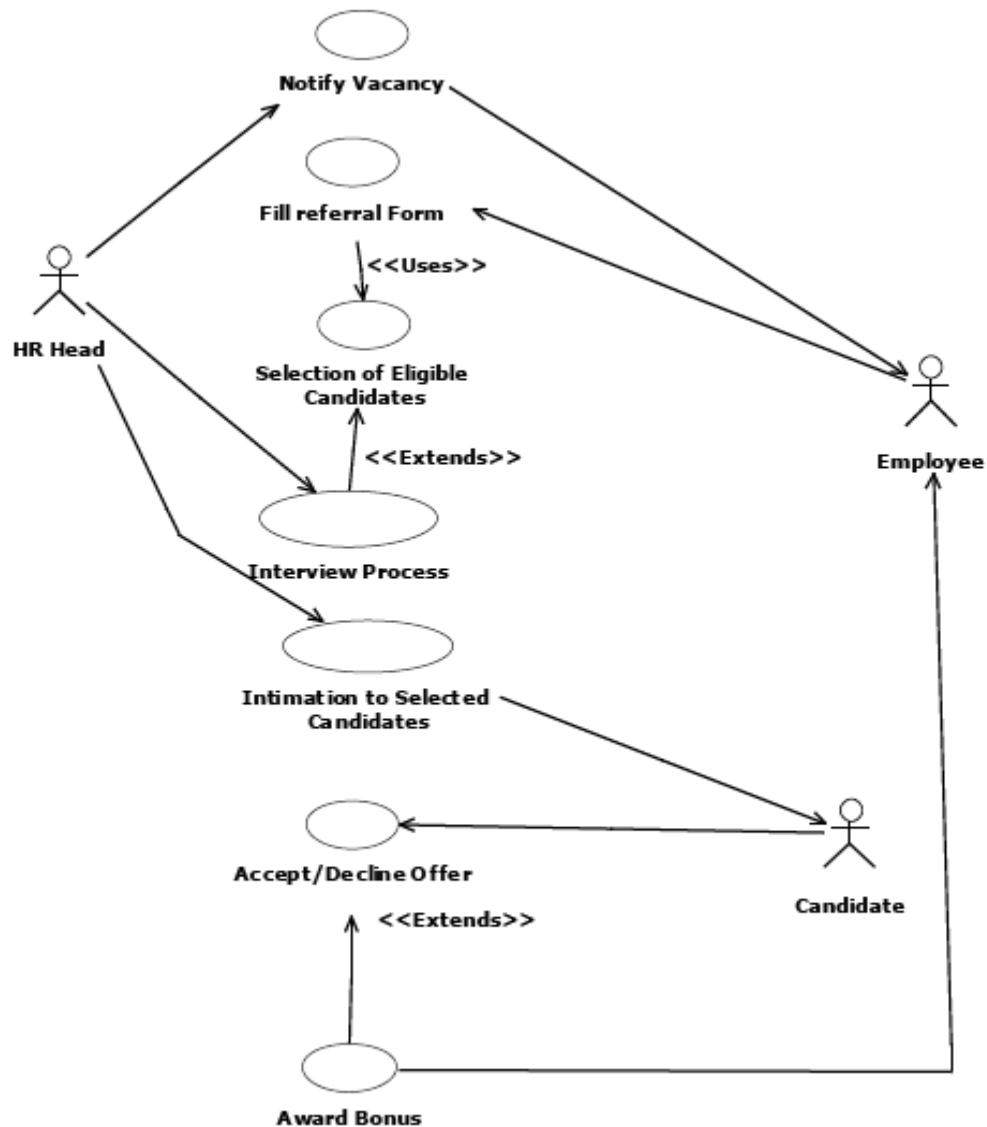
Fig.3. UML USE CASE DIAGRAM

(IV)ACTIVITY DIAGRAM:
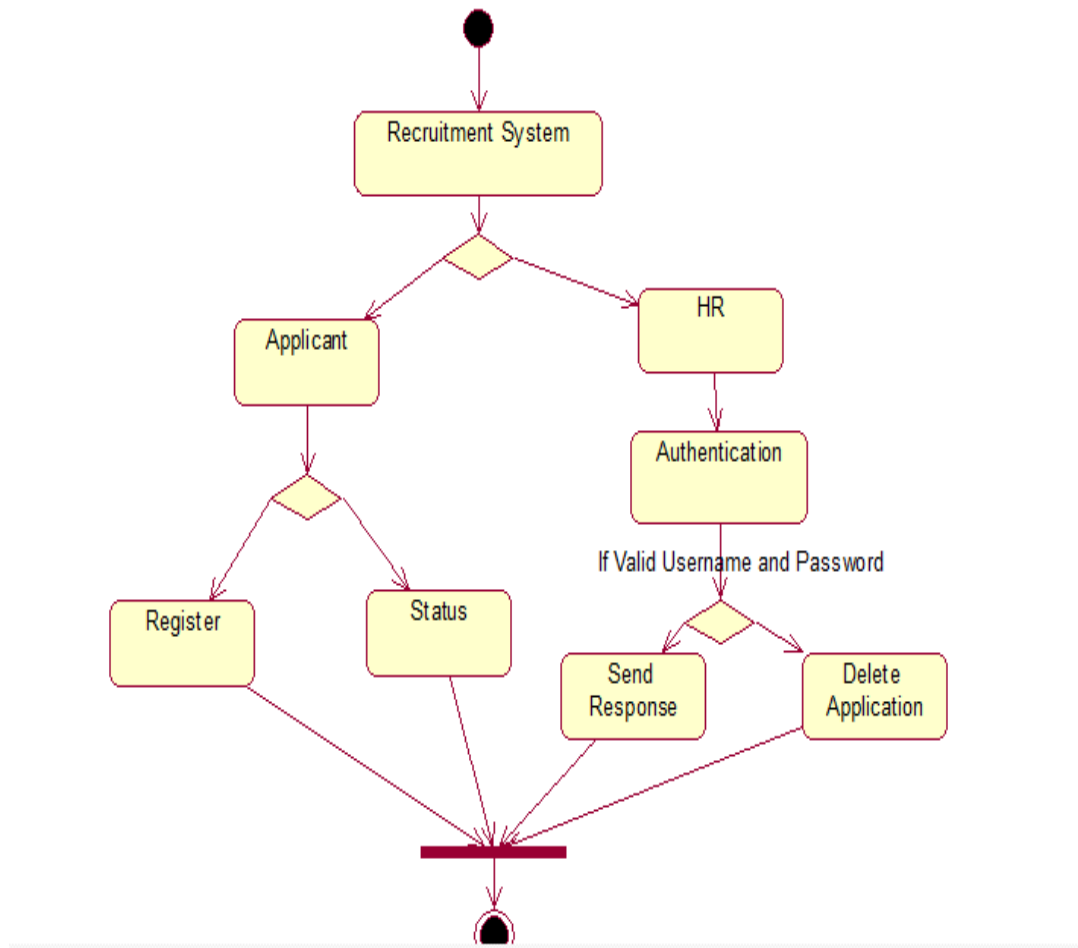
Fig.4. ACTIVITY DIAGRAM

(V)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.
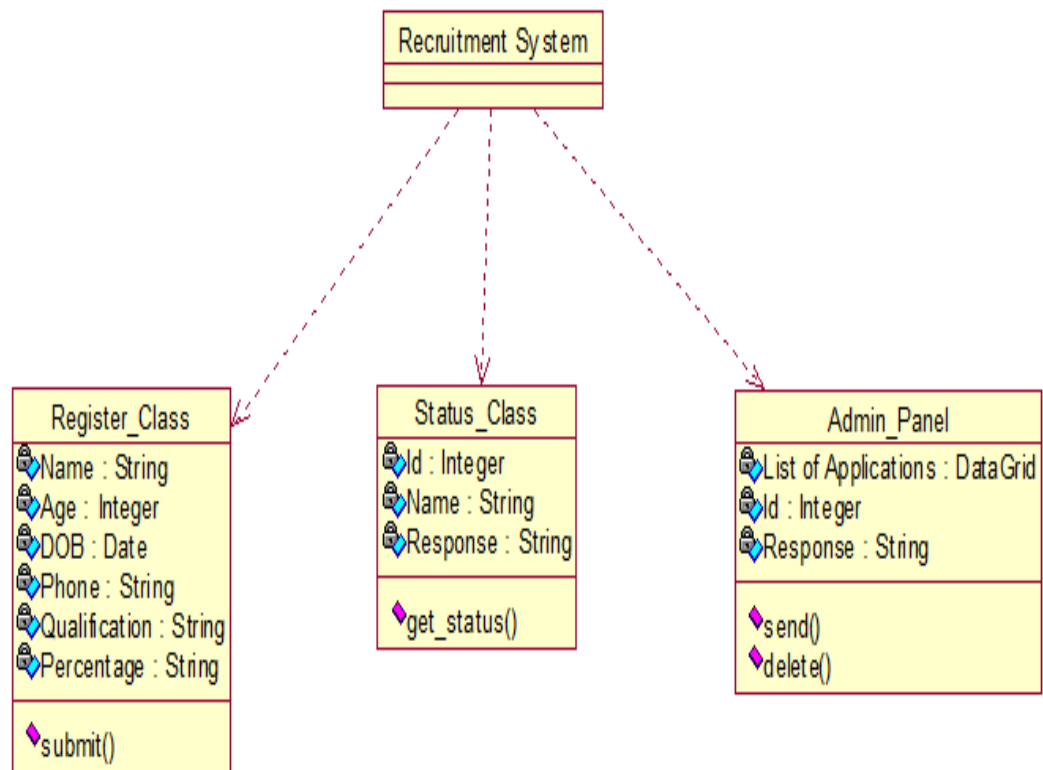
Fig.5. UML CLASS DIAGRAM

(VI)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.
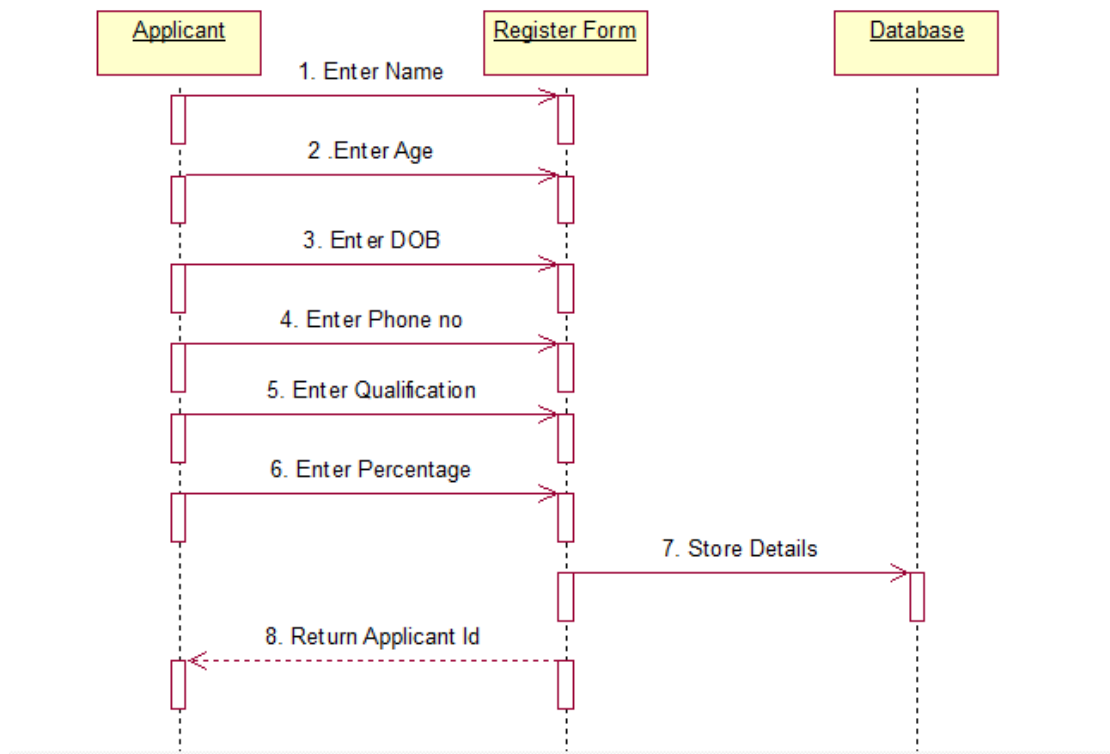
Fig. 6.1SEQEUENCE DIAGRAM FOR Register:
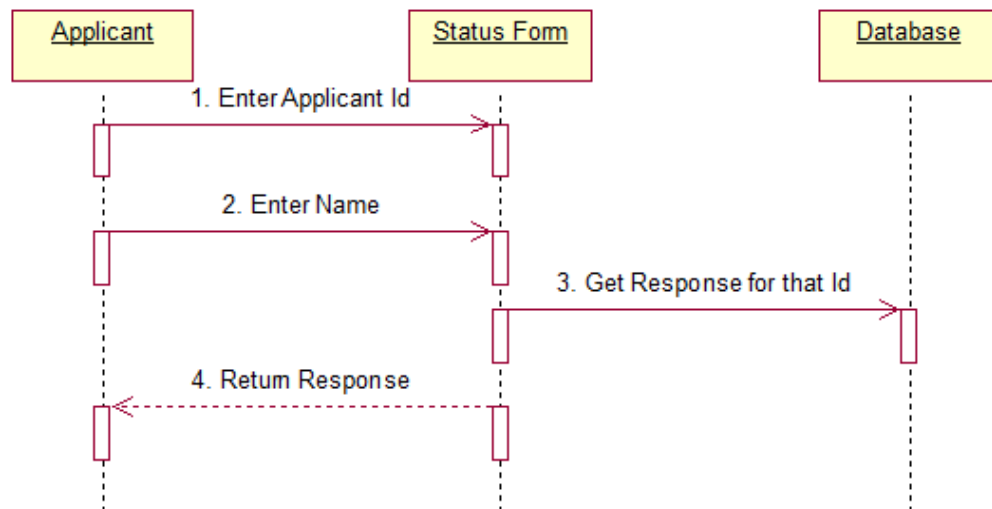
Fig.6.2. SEQUENCE DIAGRAM FOR Status
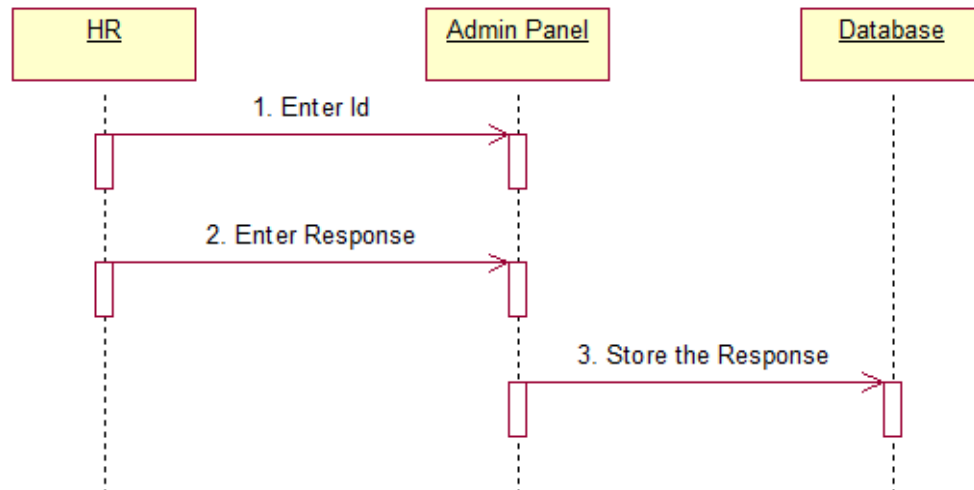
Fig.6.3. SEQUENCE DIAGRAM FOR Admin

Notify Vacancy:
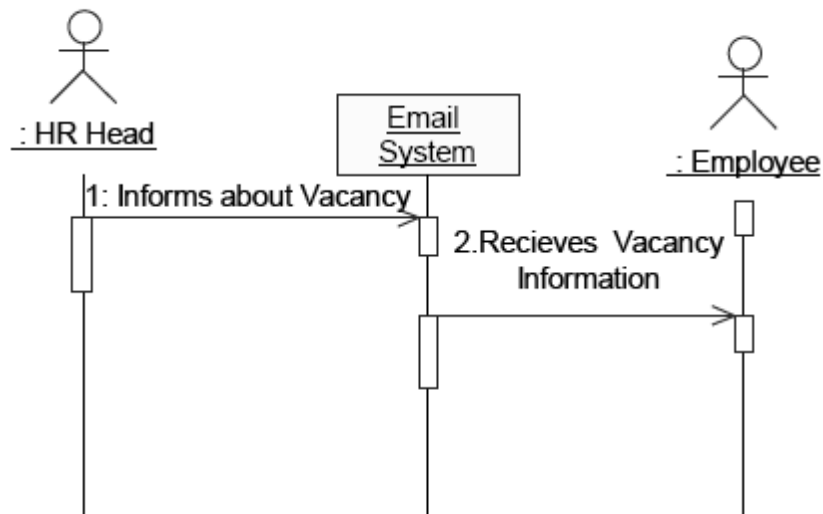
(VII) UML COLLABRATION DIAGRAM:

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.
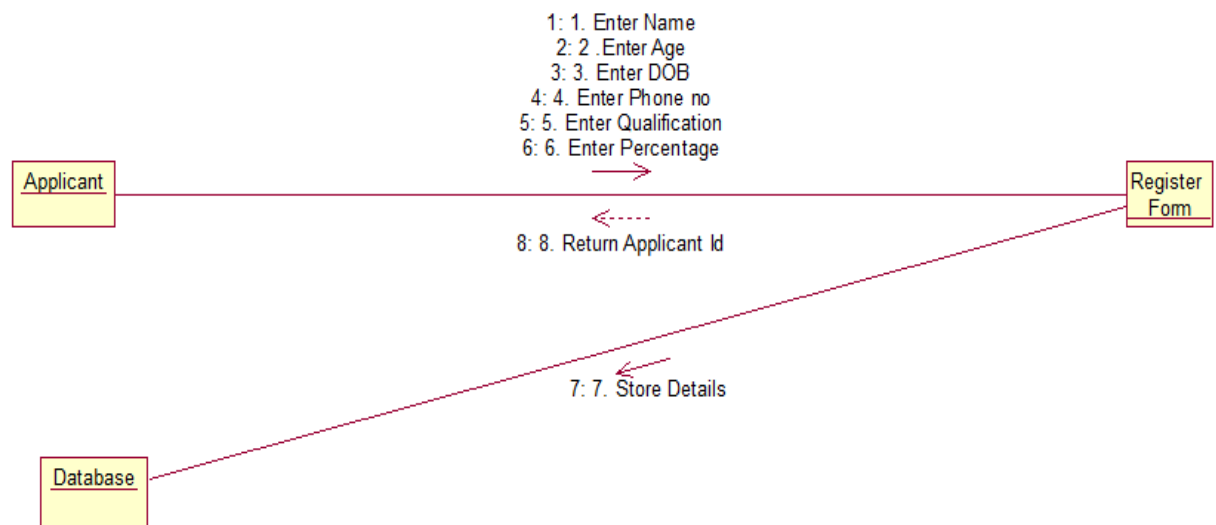
Fig.7.1COLLABRATION DIAGRAM For Register
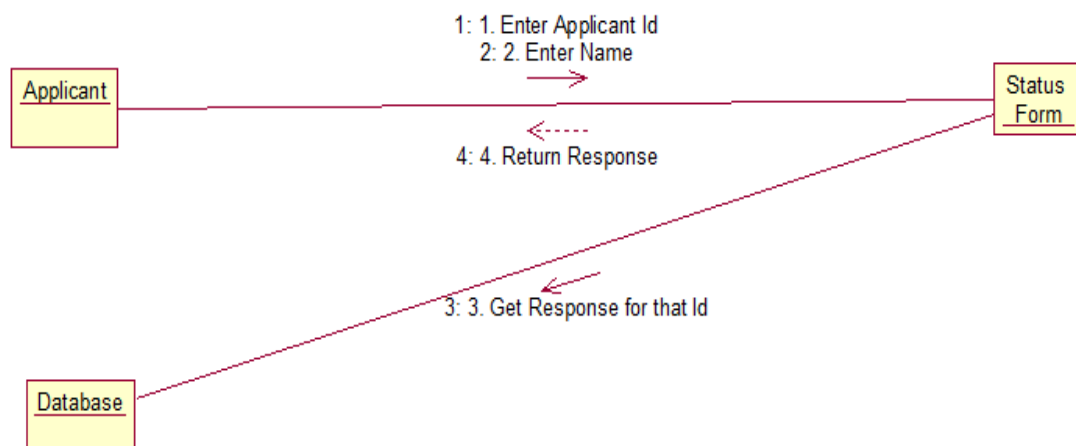
Fig.7.2. COLLABORATION DIAGRAM FOR Status
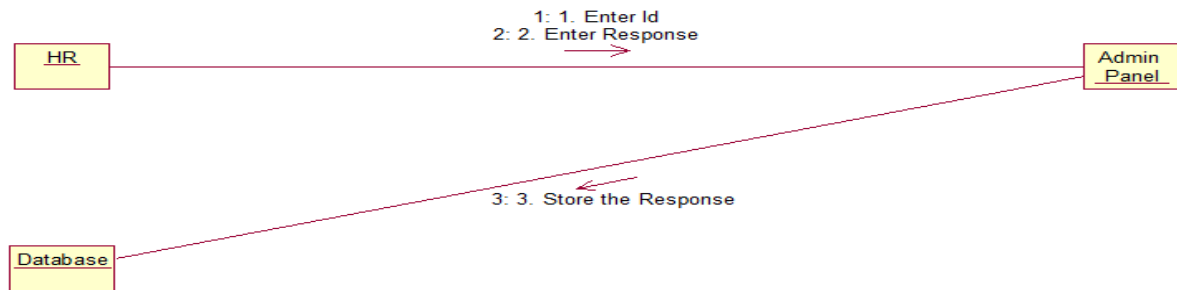
Fig.7.3.COLLABORATION DIAGRAM FOR Admin

(VIII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:
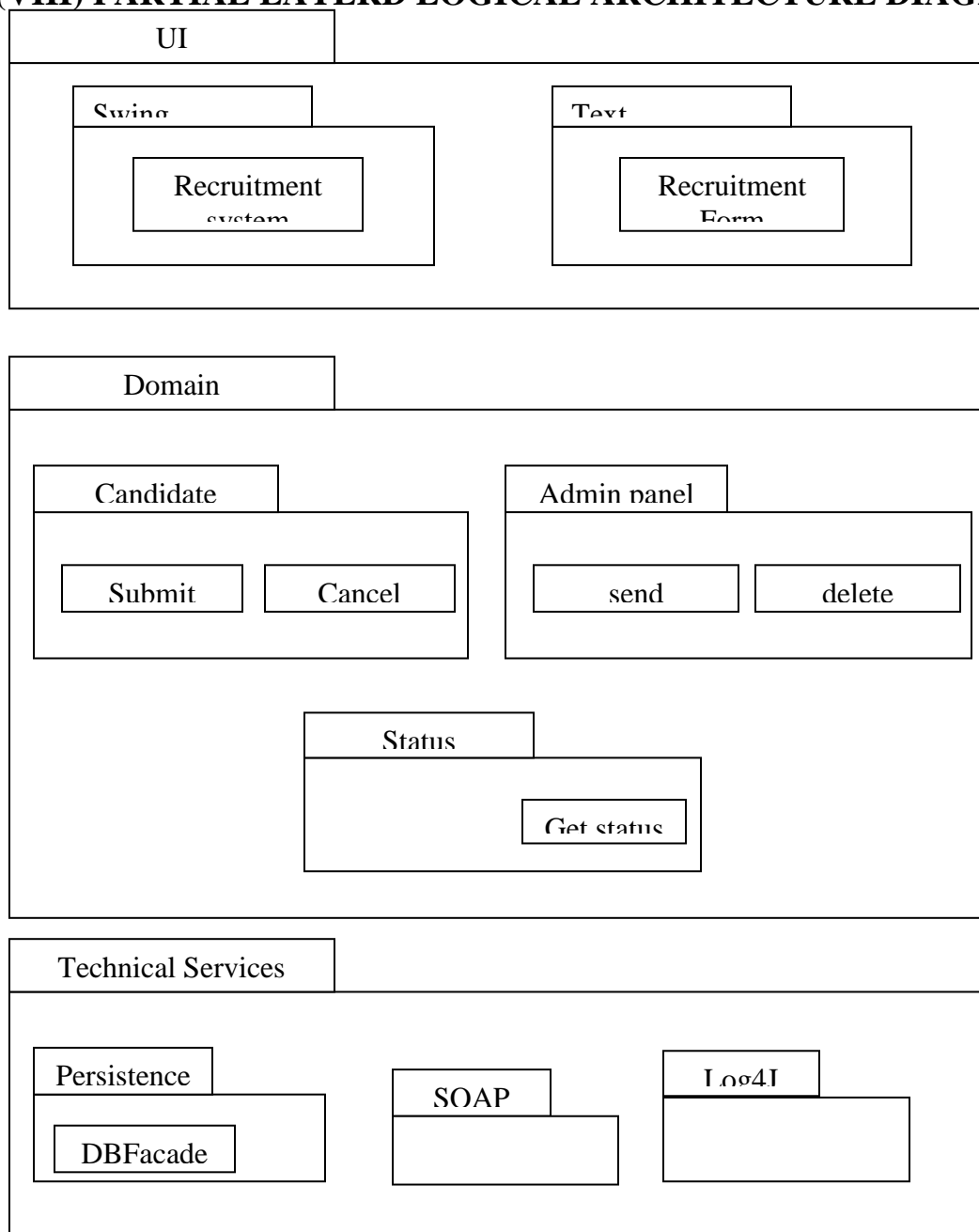
(IX) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

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

<<databas e>> :MySQL

SQ

<<server> > :Tomcat6

H

<<client workstati on>>:Gen

Fig.9.1.DEPLOYMENT DIAGRAM

Component Diagram

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

Database Server

DB

ADODB LAN

Web Server

IIS Services

TCP/IP

Client Machine

Web Browser
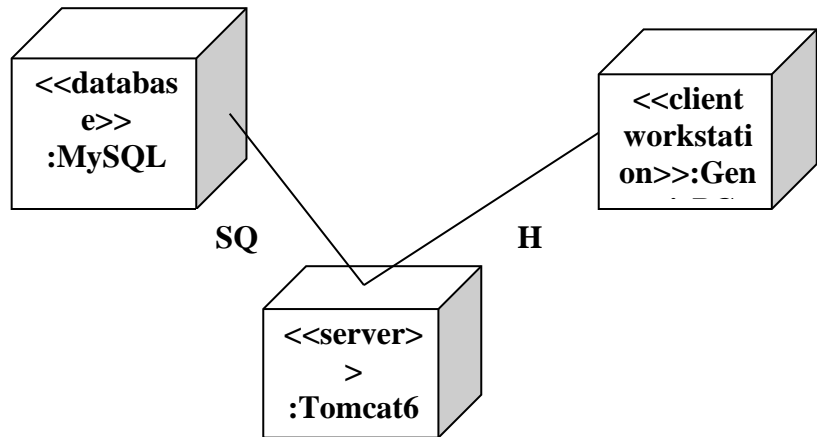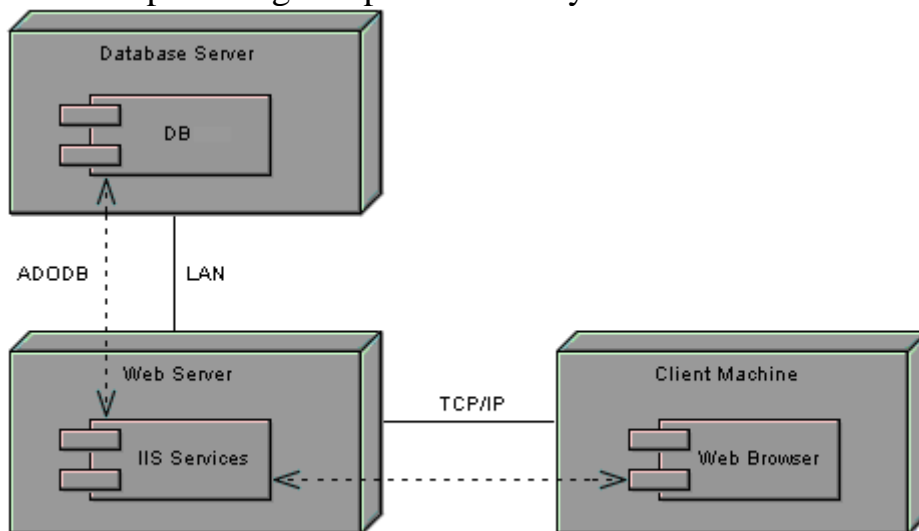
Fig.9.2.COMPONENT DIAGRAM

RESULT:

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

Ex.No :12 **FOREIGN TRADING SYSTEM**

Date :

AIM

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

(I) 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.

(II)SOFTWARE REQUIREMENT SPECIFICATION

ADMINISTRATOR	One who coordinates the entire trading process.
DATABASE	All the transaction details are stored here.
READER	Person who is viewing the website.
USER	The traders and the viewers are the users.
SOFTWARE REQUIREMENT SPECIFICATION	This software specification documents full set of features and function for foreign trading system.

FUNCTIONALITY
Transfer purchasing power between countries. Obtain credit for international trade transactions. Minimize exposure to the risks of exchange rate changes.

FUNCTIONALITY REQUIREMENTS

Functional requirements refers to the functionality of the system. The services that are provided to the trader who trades.

UML DIAGRAMS

The following UML diagrams describe the process involved in the foreign trading system.

(III)USE CASE DIAGRAM

A use case diagram purpose is to present a graphical overview of the functionality provided by the system in terms of actors, their goals, and any dependencies between those use cases.A use case is an interaction between users and a system in a particular environment. It captures the goal of the users and the responsibility of the system to the user. It is represented using ellipse. Actor is a user playing a role with respect to the system. A single actor may perform many usecases. It is represented using a stick figure along with a label.

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:
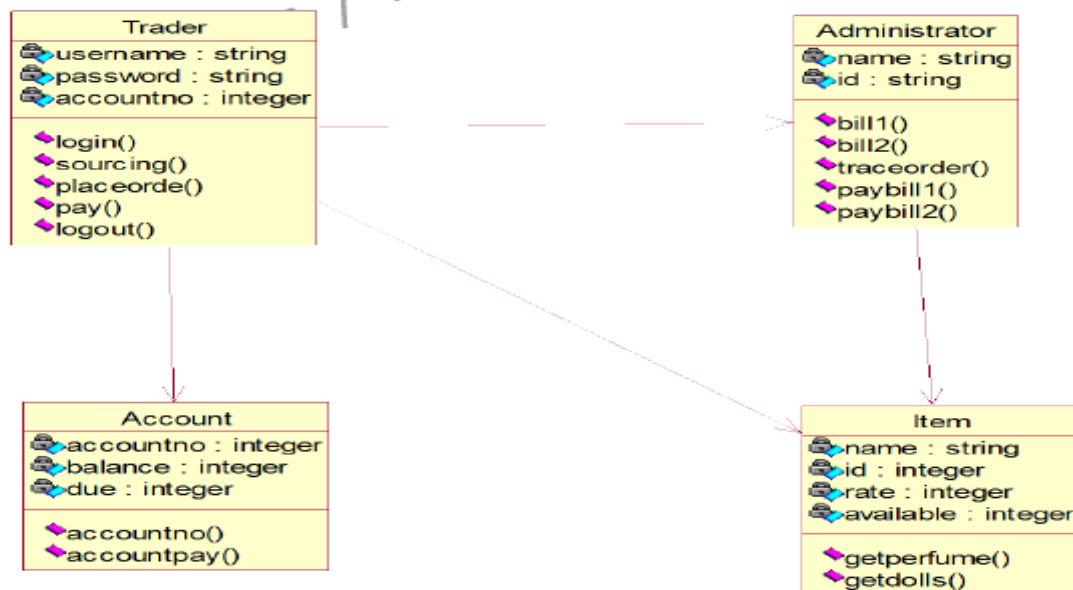
SEQUENCE DIAGRAM

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
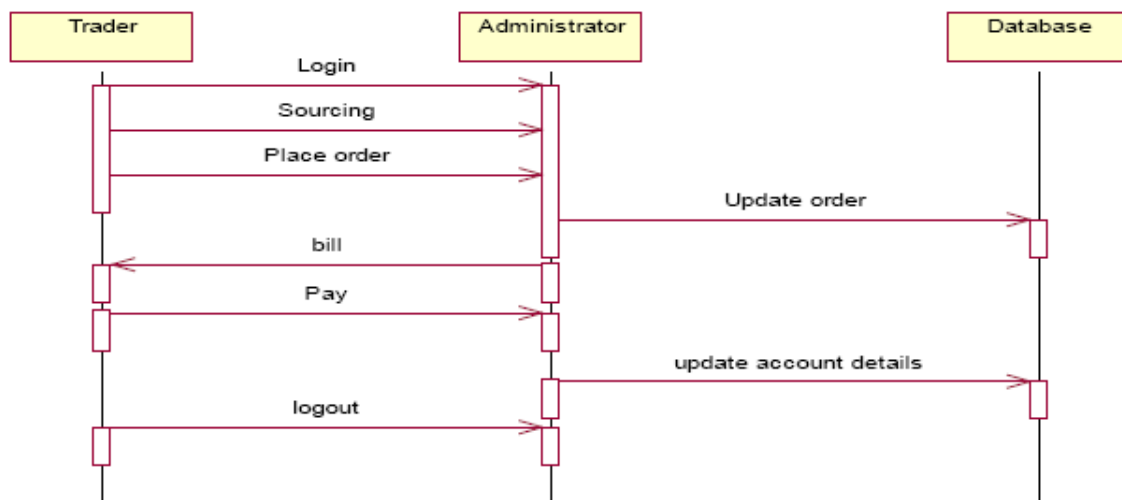
TraderTraderAdministratorAdministratorDatabaseDatabaseLoginSourcingPl ace orderUpdateorderPay update account details bill logout

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.

```
          1: Login
          2: Sourcing
          3: Place order
             6: Pay
             8: logout
```

Trader → Administrator

5: bill

7: update account details
4: Update order
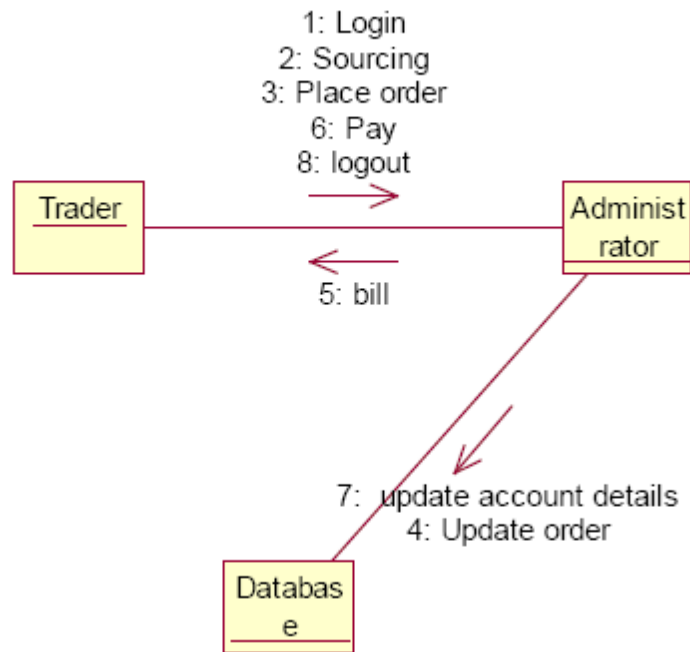
Database

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.
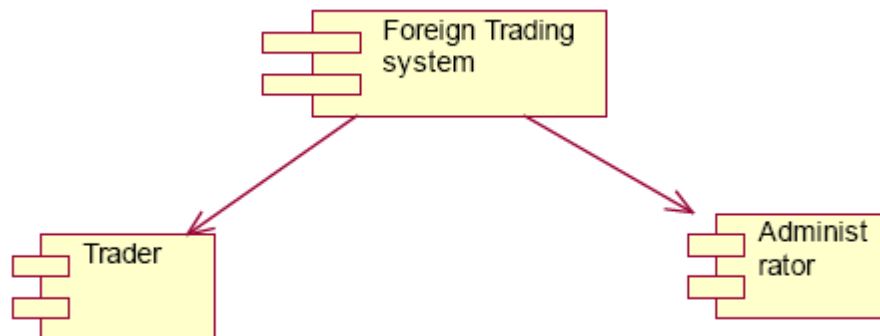
DOCUMENTATION OF COMPONENT DIAGRAM

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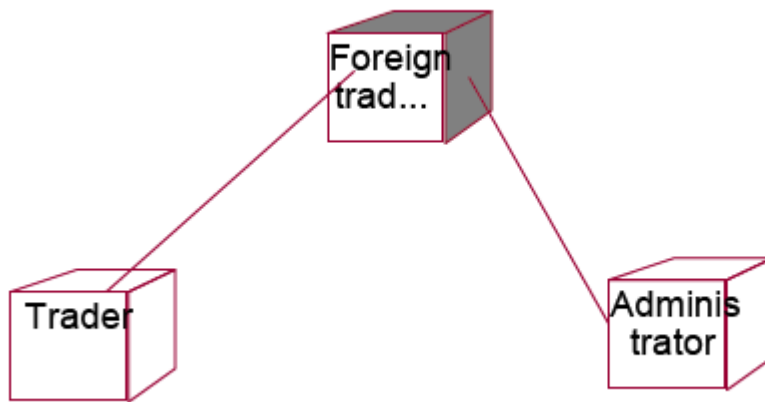
DOCUMENTATION OF DEPLOYMENT DIAGRAM
The processor in this diagram is the foreign trading system. The devices are the trader and administrator who perform the main activities in the system.

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 project to develop foreign trading system using Rational Rose software and to implement the software in Visual Basic is done successfully.

Ex.No: 13   CONFERENCE MANAGEMENT SYSTEM

Date :

AIM

      To develop a project on Conference management system using Rational Rose Software and to implement the project in Visual Basic.

(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.

(II)SOFTWARE REQUIREMENT SPECIFICATION

CANDIDATE	The candidate can login and submit the paper to the reviewer. After getting acknowledgement the candidate will submit the revised and camera ready paper then registration process will be carried out.
REVIEWER	Reviewer will reviews the paper and sending acknowledgement to the candidate
DATABASE	Database is used to verify login and store the details of selected candidates.

SOFTWARE REQUIREMENT SPECIFICATION

This software specification documents full set of features and function for conference management system.

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 sumbission– 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
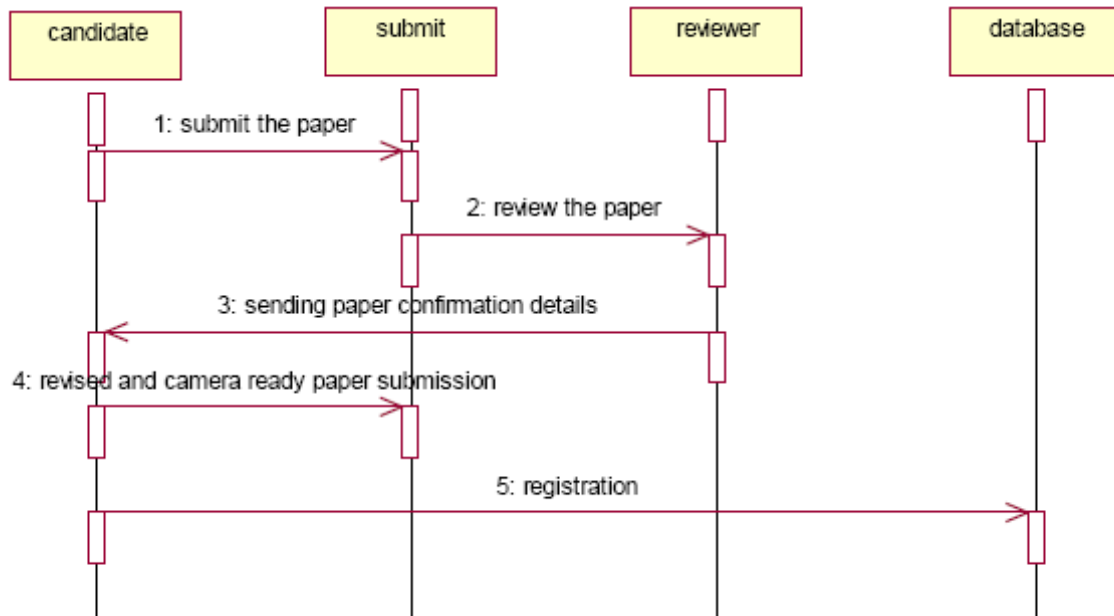
DOCUMENTATION OF SEQUENCE DIAGRAM
LOGIN
This sequence diagram describes the sequence of steps to show

• 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 sumbit 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 detials.

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.
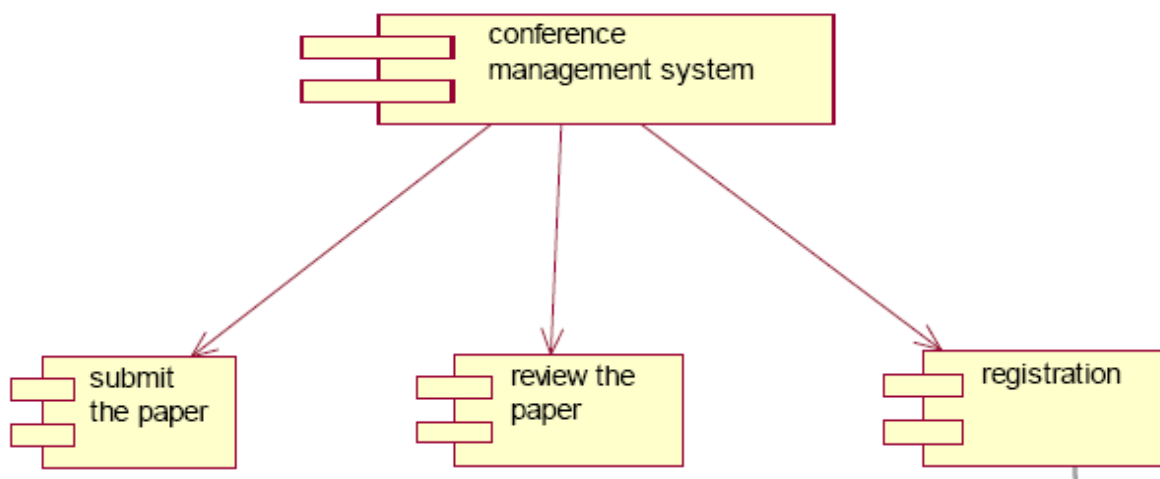
DOCUMENTATION OF COMPONENT DIAGRAM

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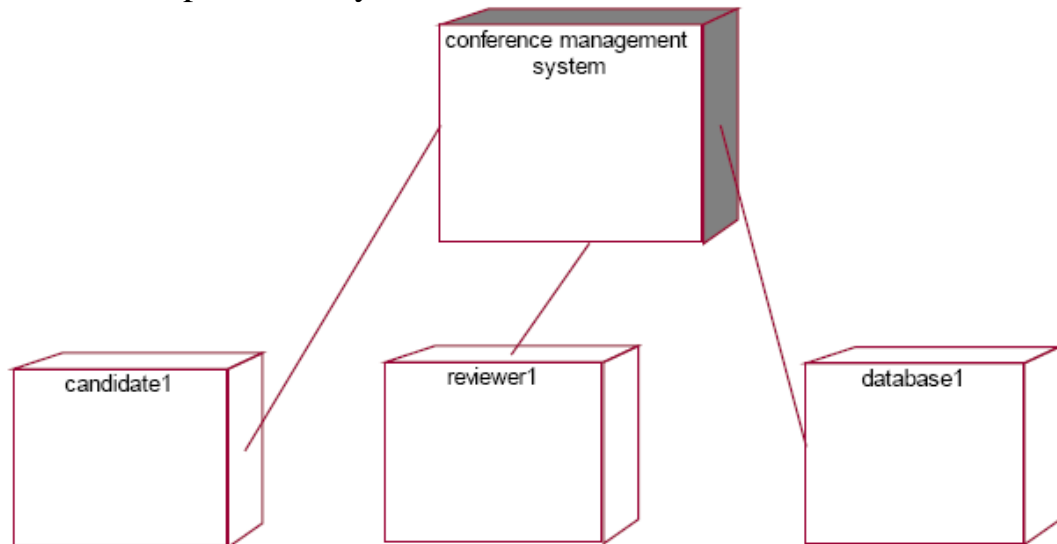The processor in this deployment diagram is the conference management system which is the main part and the devices are the candidate, appear for do conference , reviewer will reviews paper , database will store all details which are the some of the main activities performed in the system.

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 project to develop conference management system using Rational Rose Software and to implement the project in Visual Basic is done successfully.

Ex.No:14
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)SOFTWARE REQUIREMENT SPECIFICATION:

2.1 PRODUCT PERSPECTIVE
        The BPOS acts as an interface between the 'client' and the 'administrator'. This system tries to make the interface as simple as possible and at the same time not risking the security of data stored in. This minimizes the time duration in which the user receives the documents.

2.2SOFTWARE INTERFACE
        • **Front End Client** - The applicant and Administrator online interface is built using JSP and HTML. The Administrators's local interface is built using Java.

        • **Web Server** - Glassfish application server (SQL Corporation).

        • **Back End** - SQL database.

2.3HARDWARE INTERFACE
        The BPO system's server is directly connected to the client systems via ftp. The client systems have access to the database in the server.

(III)USECASE DIAGRAM:

The BPO management system use cases are:

login

search for jobs

negotiate the process

finalize the negotiate

client upload the data

BPO (business proc...

multiple client user from (USA)

data entry operation

auditt or QC

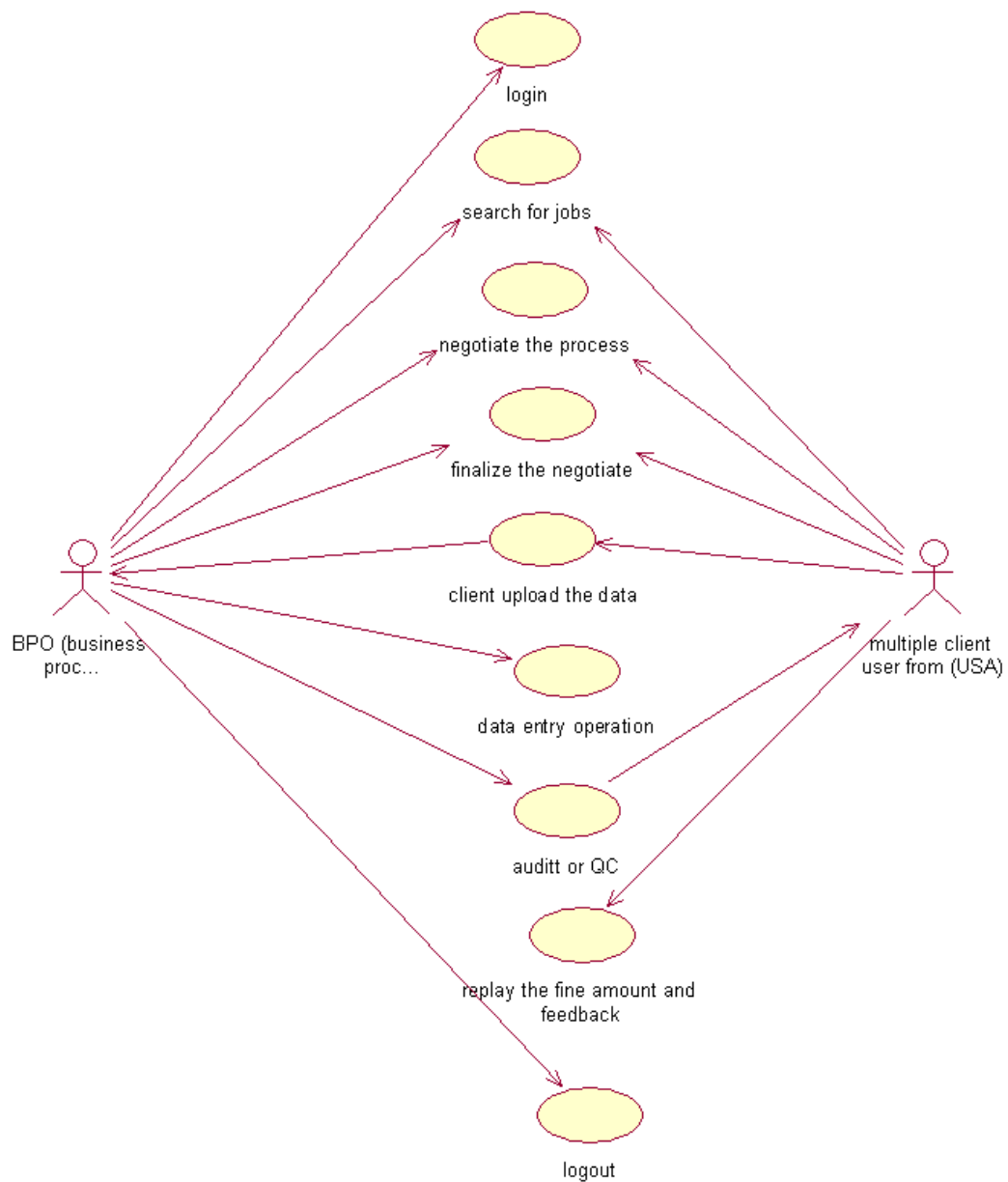replay the fine amount and feedback

logout

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.

BPO(businessProcessOutsource)
- BPOName : string
- contactNo : integer
- clientInfo : string
- emailId : string
- contractDetails : string
- bankBalance : integer

- search client()
- negotiateForWork()
- dataEntry()
- audittOrQC()
- uploadToClilent()

multiple clientUserFrom(USA)
- clientName(unknown) : string
- contactNo : integer
- emailId : string
- securityCode : charater

- searchServer()
- giveJobs()
- receiveData()
- givePayment()

client details
- client info : sting
- client id : integer
- contact no : integer
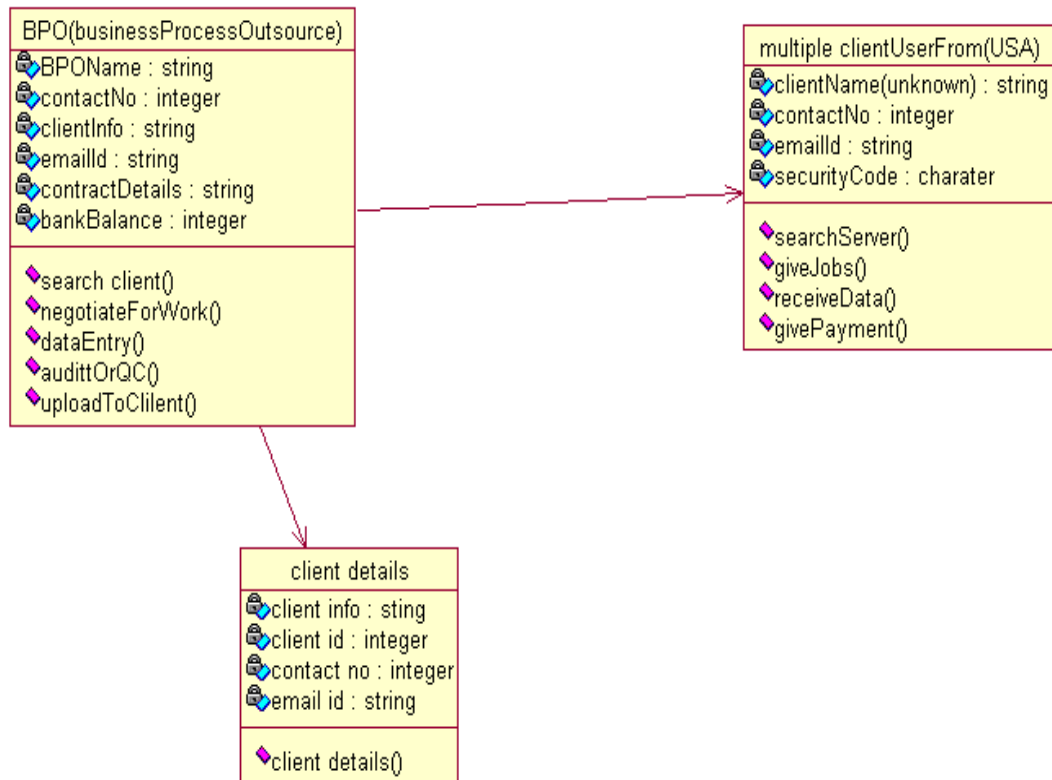- email id : string

- client details()

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.
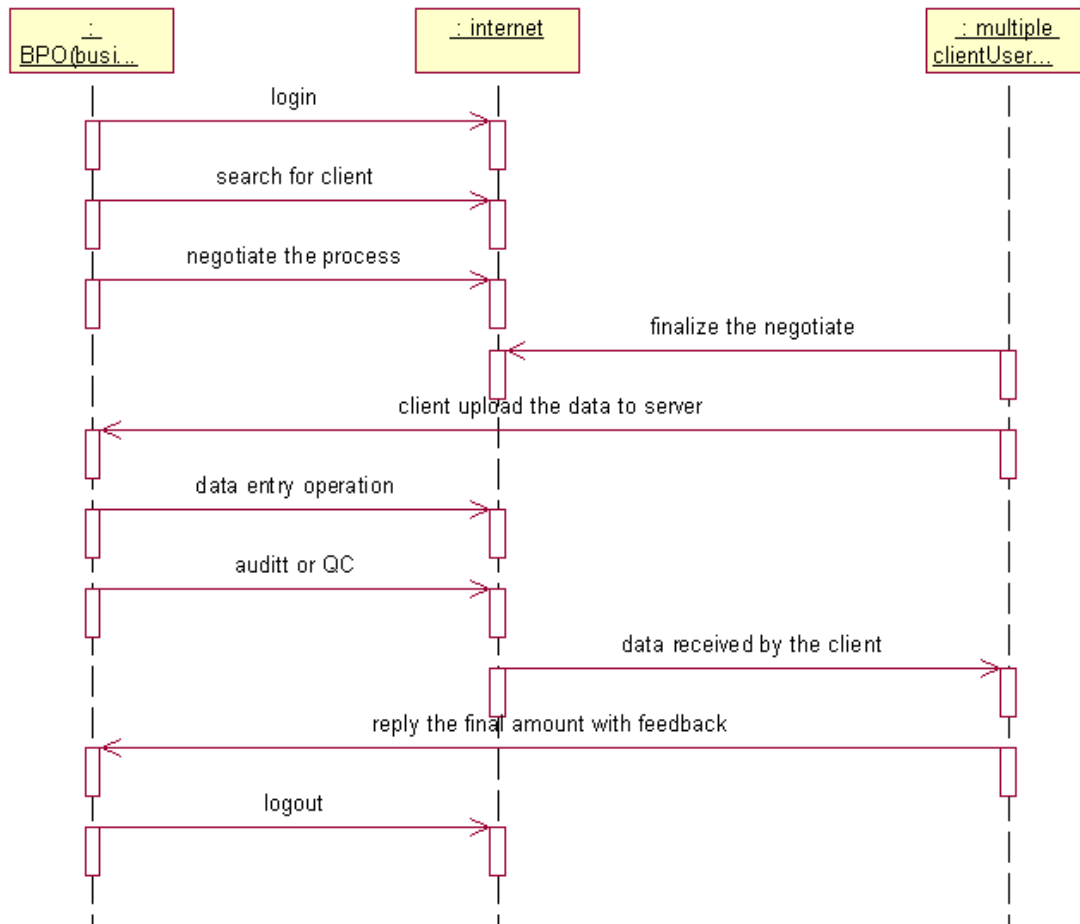

Fig. 5.1SEQEUENCE DIAGRAM

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.

1: login
2: search for client
3: negotiate the process
6: data entry operation
7: auditt or QC
10: logout

:
BPO(businessProcessOutsource)

: internet

5: client upload the data to server
9: reply the final amount with feedback

4: finalize the negotrate

8: data received by the client

: multiple
clientUserFrom(USA)
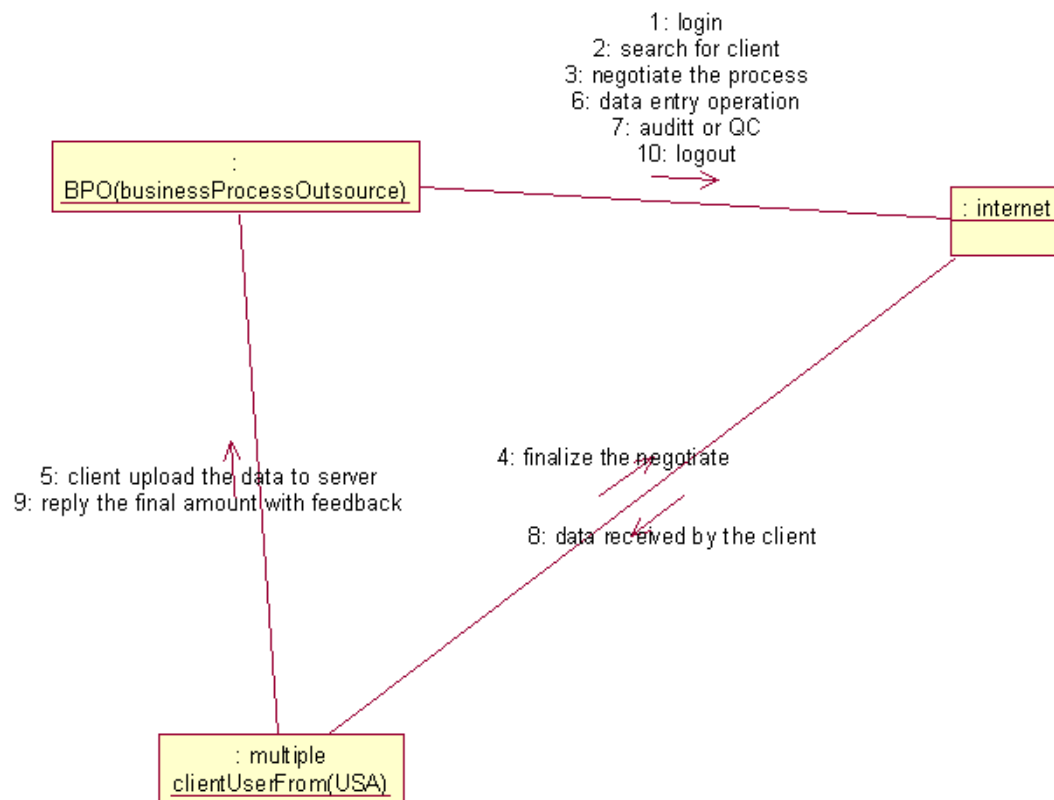
Fig.5.2COLLABRATION DIAGRAM

(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:
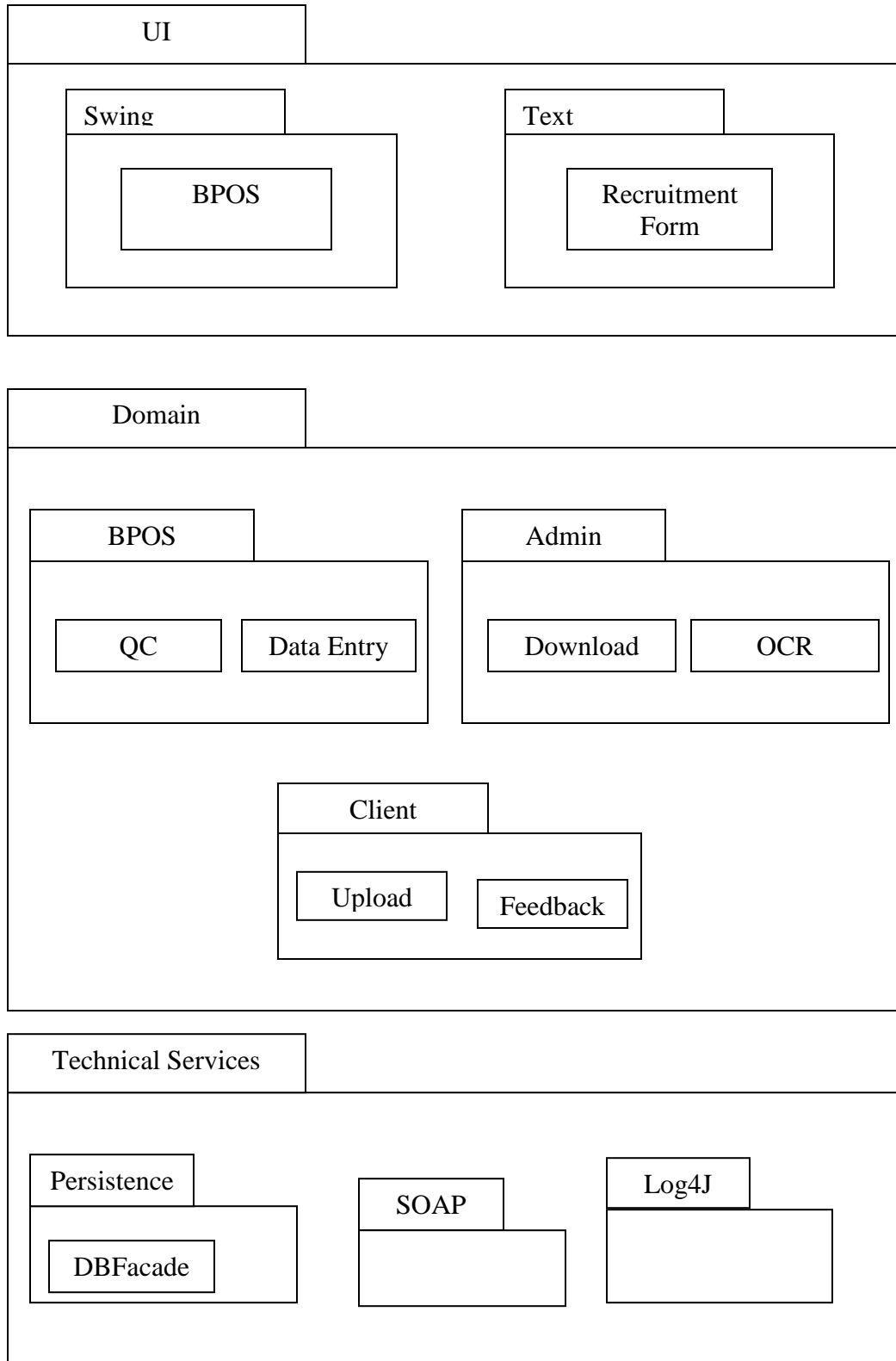
UI

Swing

BPOS

Text

Recruitment Form

Domain

BPOS

QC | Data Entry

Admin

Download | OCR

Client

Upload | Feedback

Technical Services

Persistence

DBFacade

SOAP

Log4J

(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.
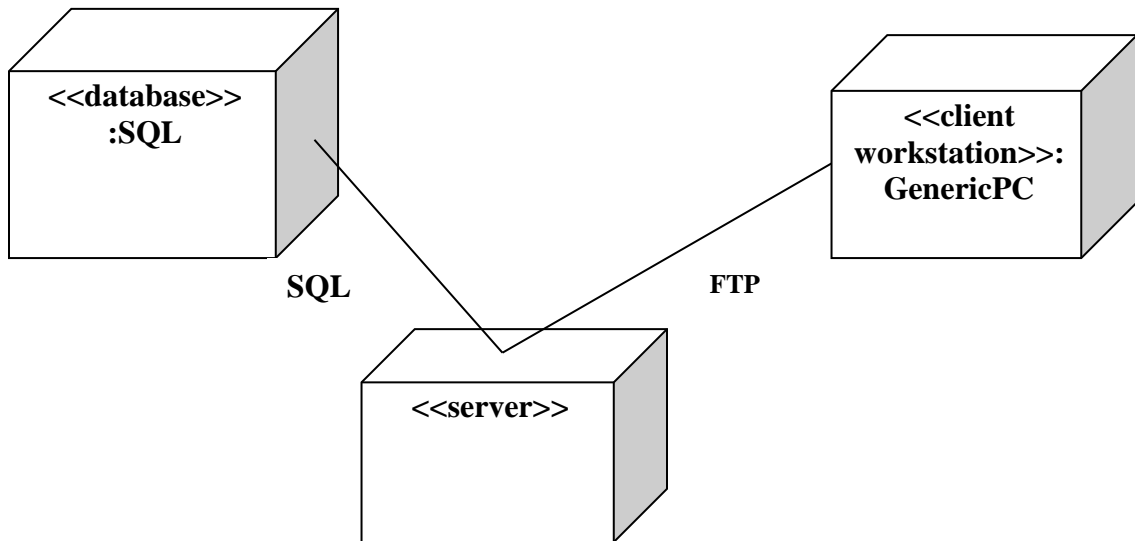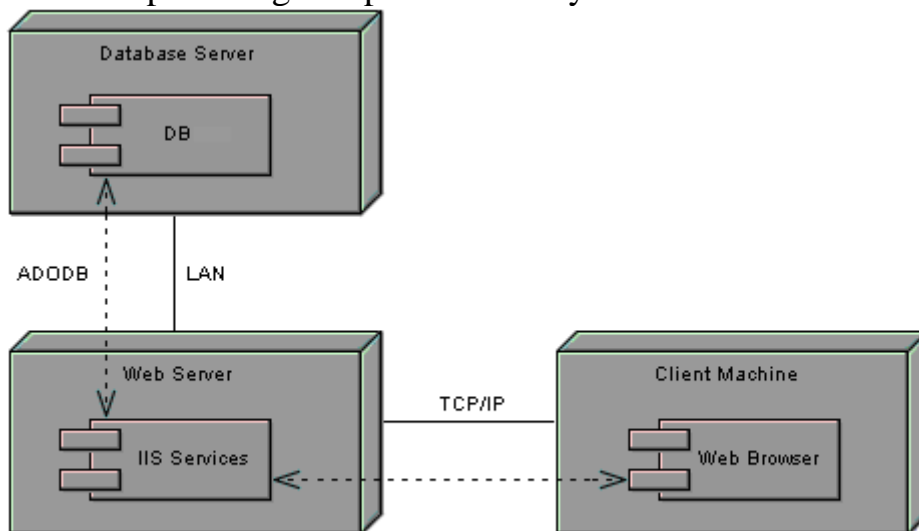

Fig.9.1.DEPLOYMENT DIAGRAM

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.

VIVA QUESTION

1. What is UML ?

   Unified Modeling Language is a visual language for specifying ,constructing , and documenting the artifacts of a system .

2. List the three ways to apply UML ?

   1. UML as a sketch
   2. UML as a Blueprint
   3. UML as a programming Language

3. What is Use case modeling ?

   It is a form of requirement engineering and a complementary way of eliciting ,and documenting requirements .

4. Define event ,state , and transitions .

   Event – It is a significant and noteworthy occurrence .

   State - It is a condition of a object at a moment .

   Transitions –It is a relationship between two states .

5. Define actor andscenario .

   Actor – It is something with behavior such as person , computer etc.

   Scenario – It is a sequence of action and interactions between actor and a system .

6. What is Class diagram ?

   The class diagram to illustrate classes , interfaces ,and their associations .

7. What is Interaction diagram ?

   It describes how to group the objects to get the job done .and also captures behavior of single scenario .

8. What is Sequence diagram ?

   It describes the behavior of a system by viewing the interaction between system and its environment .

9. What is Collaboration diagram ?

   It is a set of objects related in particular context and interactions .

10. What is State Chart diagram ?

    A state chart diagram is normally used to model how the state of an object changes in its lifetime. State chart diagrams are good at describing how the behavior of an object changes across several use case executions.

11. What is Activity diagram ?

    The activity diagram focuses on representing activities or chunks of processing which may or may not correspond to the methods

of classes. An activity is a state with an internal action and one or more outgoing transitions which automatically follow the termination of the internal activity.

12. What is Component diagram ?

Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries, etcIt does not describe the functionality of the system but it describes the components used to make those functionalities.

13. What is Deployment diagram ?

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

14. What is Package diagram ?

Package diagram is UML structure diagram which shows packages and dependencies between the packages.

15. Define GRASP .

General responsibility assignment software patterns (or principles), abbreviated GRASP, consist of guidelines for assigning responsibility to classes and objects in object-oriented design.

16. What is coupling ?

Coupling is the degree to which one class knows about another class

17. Define cohesion .

Cohesion is used to indicate the degree to which a class has a single, well-focused purpose. Coupling is all about how classes interact with each other

18. What is pattern ?

A pattern is a name problem or solution pair that can be applied in new context.

19. Define bridge .

It divides the component in to abstraction and implementation .

20. What is inception ?

It is the initial short step to establish a common vision and basic scope for the project .

21. Define artifacts .

An artifact is a classifier that represents some physical entity, a piece of information that is used or is produced by a software development process .

22. What is association ?

An association is a relationship between classes that indicates some meaningful and interesting connections .

23. What is interaction diagram ?

This interactive behavior is represented in UML by two diagrams known as Sequence diagram and Collaboration diagram. The purpose of interaction diagrams is to visualize the interactive behavior of the system.

24. What is GUI Testing ?

GUI Testing is the process of ensuring proper functionality of the graphical user interface for a given application .

25. List down the testing activities .

Establish , Design , Write ,test , Execute and Evaluate the test cases .