

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)
Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMMON TO: DEPARTMENT OF INFORMATION TECHNOLOGY



CS8581– NETWORKS LABORATORY

R 2017

LAB MANUAL

INDEX

S.No.	Name of the Experiment	Page No	Date of Experiment	Remarks
1.	Networking Commands			
2.	Write a HTTP web client program to download a web page using TCP sockets			
3a.	Socket program For Echo			
3b.	Client- Server application for Chat			
3c.	File Transfer in Client & Server			
4.	Simulation of DNS using UDP sockets.			
5a.	Write a code simulating ARP protocols.			
5b.	Write a code simulating RARP protocols.			
6.	Study of Network simulator and Simulation of Congestion Control Algorithms using NS.			
7.	Study of TCP/UDP performance using Simulation tool.			
8a.	Simulation of Distance Vector Routing algorithm.			
8b.	Simulation of Link State Routing algorithm.			
9.	Simulation of error correction code (like CRC).			
10.	Performance evaluation of Routing protocols using Simulation tool.			

DATE:

1.NETWORKING COMMANDS

AIM:

To study the basic networking commands.

NETWORKING COMMANDS:

C:\>arp -a: ARP is short form of address resolution protocol, It will show the IP address of your computer along with the IP address and MAC address of your router.

C:\>hostname: This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

C:\>ipconfig: The ipconfig command displays information about the host (the computer your sitting at)computer TCP/IP configuration.

C:\>ipconfig /all: This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system.

C:\>Ipconfig /renew: Using this command will renew all your IP addresses that you are currently (leasing) borrowing from the DHCP server. This command is a quick problem solver if you are having connection issues, but does not work if you have been configured with a static IP address.

C:\>Ipconifg /release: This command allows you to drop the IP lease from the DHCP server.

C:\>ipconfig /flushdns: This command is only needed if you're having trouble with your networks DNS configuration. The best time to use this command is after network configuration frustration sets in, and you really need the computer to reply with flushed.

C:\>nbtstat -a: This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

C:\>netdiag: Netdiag is a network testing utility that performs a variety of network diagnostic tests, allowing you to pinpoint problems in your network. Netdiag isn't installed by default, but can be installed from the Windows XP CD after saying no to the install. Navigate to the CD ROM drive letter and open the support\tools folder on the XP CD and click the setup.exe icon in the support\tools folder.

C:\>netstat: Netstat displays a variety of statistics about a computers active TCP/IP connections. This tool is most useful when you're having trouble with TCP/IP applications such as HTTP, and FTP.

C:\>nslookup: Nslookup is used for diagnosing DNS problems. If you can access a resource by specifying an IP address but not it's DNS you have a DNS problem.

C:\>pathping: Pathping is unique to Windows, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along each hop.

C:\>ping: Ping is the most basic TCP/IP command, and it's the same as placing a phone call to your best friend. You pick up your telephone and dial a number, expecting your best friend to reply with "Hello" on the other end. Computers make phone calls to each other over a network by using a Ping command. The Ping command's main purpose is to place a phone call to another computer on the network, and request an answer. Ping has 2 options it can use to place a phone call to another computer on the network. It can use the computer's name or IP address.

C:\>route: The route command displays the computer's routing table. A typical computer, with a single network interface, connected to a LAN, with a router is fairly simple and generally doesn't pose any network problems. But if you're having trouble accessing other computers on your network, you can use the route command to make sure the entries in the routing table are correct.

C:\>tracert: The tracert command displays a list of all the routers that a packet has to go through to get from the computer where tracert is run to any other computer on the internet.

RESULT:

DATE: **2. Write a HTTP web client program to download a web page using TCP sockets**

AIM:

To Write a HTTP web client program to download a web page using TCP sockets.

ALGORITHM:

CLIENT SIDE:

- 1) Start the program.
- 2) Create a socket which binds the Ip address of server and the port address to acquire service.
- 3) After establishing connection send the url to server.
- 4) Open a file and store the received data into the file.
- 5) Close the socket.
- 6) End the program.

SERVER SIDE

- 1) Start the program.
- 2) Create a server socket to activate the port address.
- 3) Create a socket for the server socket which accepts the connection.
- 4) After establishing connection receive url from client.
- 5) Download the content of the url received and send the data to client.
- 6) Close the socket.
- 7) End the program.

CLIENT PROGRAM

SERVER PROGRAM

OUTPUT

RESULT:

DATE:

3. a. SOCKET PROGRAM FOR ECHO

AIM:

To write a socket program for implementation of echo.

ALGORITHM:

CLIENT SIDE

1. Start the program.
2. Create a socket which binds the Ip address of server and the port address to acquire service.
3. After establishing connection send a data to server.
4. Receive and print the same data from server.
5. Close the socket.
6. End the program.

SERVER SIDE

1. Start the program.
2. Create a server socket to activate the port address.
3. Create a socket for the server socket which accepts the connection.
4. After establishing connection receive the data from client.
5. Print and send the same data to client.
6. Close the socket.
7. End the program.

PROGRAM:

OUTPUT:

RESULT:

DATE:

3.B. CLIENT- SERVER APPLICATION FOR CHAT

AIM:

To write a client-server application for chat using TCP

ALGORITHM:

CLIENT

1. Start the program
2. Include necessary package in java
3. To create a socket in client to server.
4. The client establishes a connection to the server.
5. The client accept the connection and to send the data from client to server.
6. The client communicates the server to send the end of the message
7. Stop the program.

SERVER

1. Start the program
2. Include necessary package in java
3. To create a socket in server to client
4. The server establishes a connection to the client.
5. The server accept the connection and to send the data from server to client and
6. vice versa
7. The server communicate the client to send the end of the message.
8. Stop the program.

PROGRAM:

OUTPUT:

RESULT:

DATE:

3.C. FILE TRANSFER IN CLIENT & SERVER

AIM:

To Perform File Transfer in Client & Server Using TCP/IP.

ALGORITHM:

CLIENT SIDE

1. Start.
2. Establish a connection between the Client and Server.
3. Socket ss=new Socket(InetAddress.getLocalHost(),1100);
4. Implement a client that can send two requests.
 - i) To get a file from the server.
 - ii) To put or send a file to the server.
5. After getting approval from the server ,the client either get file from the server or send
6. file to the server.

SERVER SIDE

1. Start.
2. Implement a server socket that listens to a particular port number.
3. Server reads the filename and sends the data stored in the file for the 'get' request.
4. It reads the data from the input stream and writes it to a file in the server for the 'put' instruction.
5. Exit upon client's request.
6. Stop.

PROGRAM:

OUTPUT:

RESULT:

DATE:

4. Simulation of DNS using UDP sockets.

AIM:

To write a program to Simulation of DNS using UDP sockets..

ALGORITHM:

- 1.Start the program.
- 2.Get the frame size from the user
- 3.To create the frame based on the user request.
- 4.To send frames to server from the client side.
- 5.If your frames reach the server it will send ACK signal to client otherwise it will send NACK signal to client.
- 6.Stop the program

PROGRAM:

OUTPUT:

RESULT:

DATE:

5.a. Write a code simulating ARP protocols.

AIM

To implement Address Resolution Protocol .

ALGORITHM

CLIENT SIDE

1. Establish a connection between the Client and Server. `Socket ss=new Socket(InetAddress.getLocalHost(),1100);`
2. Create instance output stream writer
`PrintWriter ps=new PrintWriter(s.getOutputStream(),true);`
3. Get the IP Address to resolve its physical address.
4. Send the IPAddress to its output Stream.`ps.println(ip);`
5. Print the Physical Address received from the server.

SERVER SIDE

1. Accept the connection request by the client.
`ServerSocket ss=new ServerSocket(2000);Socket s=ss.accept();`
2. Get the IPAddress from its inputstream.
`BufferedReader br1=new BufferedReader(newInputStreamReader(s.getInputStream()));`
`ip=br1.readLine();`
3. During runtime execute the processRuntime `r=Runtime.getRuntime();`
`Process p=r.exec("arp -a "+ip);`
4. Send the Physical Address to the client.

PROGRAM:

OUTPUT:

RESULT:

DATE: **5.b. Write a code simulating RARP protocols.**

AIM:

To write a java program for simulating RARP protocols.

ALGORITHM:

CLIENT

1. Start the program
2. using datagram sockets UDP function is established.
2. Get the MAC address to be converted into IP address.
3. Send this MAC address to server.
4. Server returns the IP address to client.

SERVER

1. Start the program.
2. Server maintains the table in which IP and corresponding MAC addresses are stored.
3. Read the MAC address which is send by the client.
4. Map the IP address with its MAC address and return the IP address to client.

PROGRAM:

OUTPUT:

RESULT:

DATE: **6. Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS.**

OBJECTIVE:

To study of network simulator (ns) and simulation of congestion control algorithms using ns.

SYSTEM REQUIREMENTS:

PC: Pentium or higher

One LAN card onboard or on PCI slot with 10/100Mbps speed.

128MB RAM

500MB free space on Hard drive
CD ROM drive

Serial port, LPT port & USB port installed on system
Operating System: Windows 2000 or higher

THEORY:

LTS-01 Local area network / wireless local area network trainer system:

It is designed to help students understand the basic concepts, modes of operation and protocols involved in networking. The trainer has integrated hardware flow control on panel board for better understanding of different types of LAN topologies involved in networking. The trainer system is provided with windows-based user friendly software with analysis of protocols, different layers, network and measurement of error rate and throughput.

Students can easily do connections in different topologies and can learn actual data transfer either through hardware or through simulated network concept. Facility is provided into system software to introduce errors into packets being sent and analyze the effect of error on different protocols and hence find the effect on through put graph as well.

Trainer and its various types of experimentation using this system. This system works into server-client base. For any topology user has to select one server and select the network type whether it is LAN or WLAN. To understand the topology concept user can connect two or more clients to hardware. Depending on the topology selected user will have option to select protocols for the selected topology. Upon selection of protocol user can then create network of connected computers.

In any network which is created by user server can send or can communicate with any of the clients however clients can communicate only with server, no client to client communication is possible. Transmitter port protocol & network analysis can be done after communication is over between server and clients. Throughput v/s Packet size graph can be plotted for which at least two file transfers should be carried out. This plot can be printed to attach in the lab exercise sheet.

For the LAN network LAN cards must be installed prior to start work on this trainer. For wire less LAN USB ports should be available on the computers which are to be used for experimentation.In WLAN wireless access cards gets connected to computer USB ports and access point gets connected to hardware device.

L-SIM LAN Protocol Simulator & Analyzer Software:

It is designed to teach the basic concepts, topologies & various protocols involved in networking. The software is provided with analysis of protocols, different layers, network and measurement of error rate and throughput. Facility is provided to introduce errors into packets beingsent and analyze the effect of error on different protocols and hence find the effect on throughput graph as well. Software is supported with neat operating instruction manual and online help.

MODEL WINDOW DIAGRAM FOR L-SIM



N-S IM Network simulation software:

It is developed to provide basic understanding and implementation of various advanced concepts in networking. The software provides an opportunity to understand network fundamentals through animations & simulations. The simulation provides for network experimentation with various LAN and WAN protocols, network devices, routers, encryption, decryption, file transfer, error insertion and analysis of error rate and throughput etc. This software covers Ethernet LAN, wireless LAN and router. All networking theory is explained using simulation and animation.

MODEL WINDOW DIAGRAM FOR N-SIM



Rapid advances in computer & communication technologies have resulted in the increasing merger of these two fields. The lines have blurred among computing, switching & digital transmission equipment; and the same digital techniques are being used for data, audio & video transmission. Merging & evolving technologies, coupled with increasing demands for efficient & timely collection, processing & dissemination of information, have led to the development of integrated systems that transmit & process all types of data.

These integrated systems are broadly divided as follows

- **DATA COMMUNICATION** dealing with transmission, transmission media, signal decoding, interfacing, data link control & multiplexing
- **NETWORKING** deals with the technology & architecture of communication network
- **COMMUNICATION PROTOCOLS** which covers the architecture as well as analysis of individual protocols at various layers depending on the hardware & software Network laboratory is designed & developed considering the curriculum offered by Anna University. Trainers offered under network laboratory are designed for students at all level to study and understand all the concepts of data communication, data transfer using serial and parallel ports, Ethernet and wireless LAN with complete protocol understanding and actual hands on with hardware & software with ease.

Network laboratory consists of DCT-03 Data communication trainer kit, LTS- 01 LAN / Wireless LAN training system, L-SIM LAN / WLAN protocol simulator and analyzer software & N-SIM Network simulation software.

The DCT-03: Data communication trainer is a unique trainer kit for the development of exercises and theoretical-experimental courses to understand the basic concept and working of modes and protocols in serial and parallel communication.

The trainer kit consists of functional blocks for serial and parallel communication system.

The trainer kit is highly innovative from a technological as well as an educational point of view. The trainer kit is used as “basic unit” to examine all the peculiar operating standards of serial and parallel communication system. The only external equipments required are two Computers with

serial and parallel communication ports and an Oscilloscope. Utmost care has been laid in the design and quality control of all circuits, to ensure the repeatability of the results of the experiments.

Data communication is a term referred when the sender and receiver are digital devices, which communicate with each other by means of binary information. The objective of this trainer kit is to clear the various aspects of the data communications which comprise of

- The information source or sender.
- The medium for carrying information.
- The information receiver.
- The communication protocols, which ensure proper transfer of data.

With an increasing demand in information exchange the field of data communication technique is emerging as the only solution, to satisfy the various needs of today's communication sector and to achieve very high bandwidth along with highest accuracy. The communication media is shifting from analog signal transfer towards digital communication.

With PC becoming the biggest storage devices in digital form, it becomes the main source and destination for information exchange. With rapid growth in both the communication technologies as well as computer hardware and software technologies, these two fields are merged to form a data communication network. Now the digital data is used for data, voice and image transmission.

Depending upon the application the communication link can be of point to point communication between two devices or a multipoint communication between at least 3 devices and data transfer can be serial or in parallel form.

RESULT:

DATE: **7. Study of TCP/UDP performance using Simulation tool.**

AIM:

To Study of TCP/UDP performance using Simulation tool.

TOOLS USED:

Opnet Simulator

INTRODUCTION:

The transport layer protocols provide connection- oriented sessions and reliable data delivery services. This paper seeks to reflect a comparative analysis between the two transport layer protocols, which are TCP/IP and UDP/IP, as well to observe the effect of using these two protocols in a client server network. The similarities and differences between TCP and UDP over the Internet are also presented in our work. We implement a network structure using Opnet Modeler and finally, based on the practical results obtained we present the conclusions-showing the difference between these two protocols and how they work.

The transport layer is not just another layer. It is the heart of the whole protocol hierarchy. Its task is to provide reliable, cost-effective data transport from the source machine to the destination machine, independently of the physical network or networks currently in use.

TCP and UDP are transport layer components that provide the connection point through which applications access network services. TCP and UDP use IP, which is a lower-layer best effort delivery service. IP encapsulates TCP packets and UDP datagrams and delivers this information across router-connected internet works.

The ultimate goal of the transport layer is to provide efficient, reliable, and cost-effective service to its users, normally processes in the application layer. To achieve this goal, the transport layer makes use of the services provided by the network layer. Without the transport layer, the whole concept of layered protocols would make little sense e.g. The Transport Layer prepares applications data for transport over the network and processes network data to be used by applications. It is responsible for the end-to-end transfer of data over the network and is the four of the OSI model. The Transport layer meets a number of functions:

- enabling the applications to communicate over the network at the same time when using a single device;
- ensure that all amount of data is receive by the correct application;
- responsible for fragmentation and reassembly;
- develop mechanism for handling errors.

Comparison Between TCP And UDP

Service	TCP	UDP
Flow controls	The receiver can signal the sender to slow down.	ACKs, which are used in TCP to control packet flow, are not returned.
Connection setup	It takes time, but with TCP reliability is ensured.	No connection is required.
Guaranteed message delivery	Returns acknowledgments.	UDP does not return ACKs, the receiver can't signal that packets have been successfully delivered.
Congestion controls	Network devices can take advantage of TCP ACK to control the behavior of sender.	If ACK, are missing, the network cannot signal congestion to the sender.

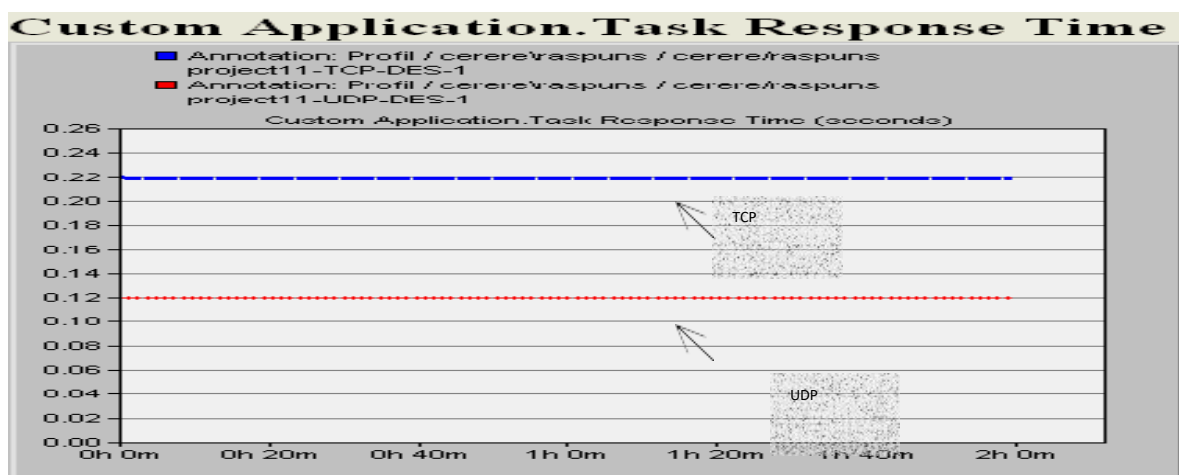
A big difference between TCP and UDP is the congestion control algorithm. For the TCP, congestion algorithm prevents the sender from overrunning the network capacity, while TCP can adapt the sender's rate with the network capacity and attempt to avoid potential congestions problems.

User Datagram Protocol (UDP), another transport protocol in IP networks, is described e.g. The User Datagram Protocol (UDP) provides an unreliable connectionless delivery service using IP to transport messages between machines e.g. [5]. It uses IP to carry messages, but adds the ability to distinguish among multiple destinations within a given host computer. Is a connectionless protocol which doesn't provide flow control, reliability or error recovery and the retransmissions of data in case of errors must be ordered by other protocols. UDP is designed for applications that do not have to recombine the data segment that arrives from the sender. In another way, application-level protocols are directly responsible for the security of data transmitted.

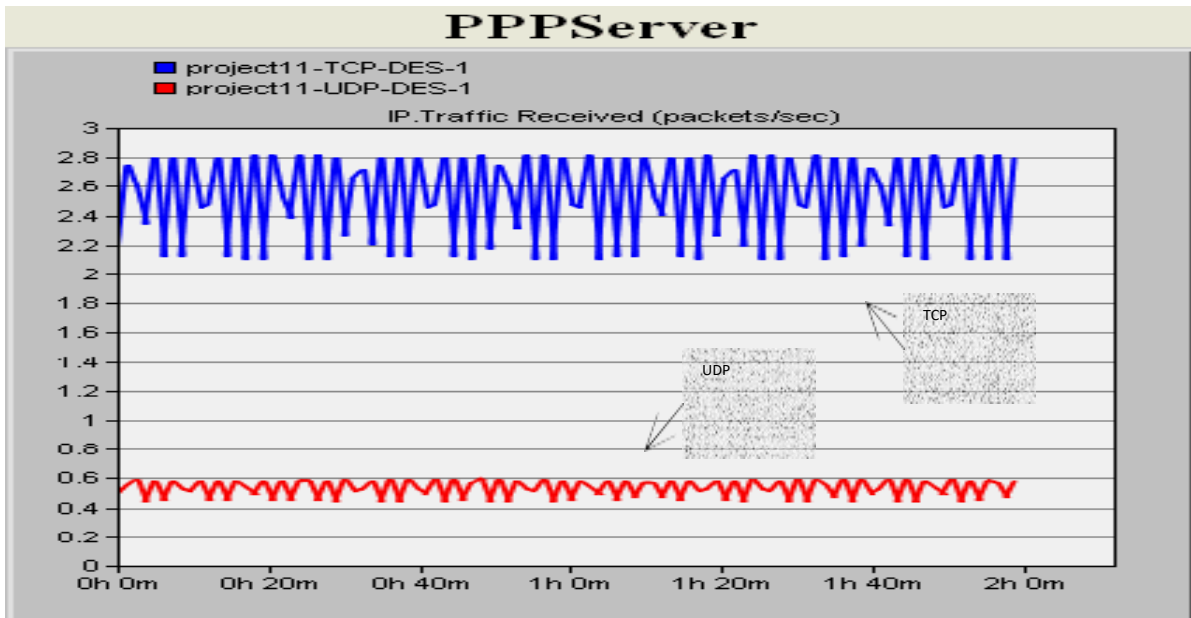
Difference from the TCP is that there is no mechanism for error detections. If applications that use UDP doesn't have their own mechanism for information retrieval can lose those data and be forced to retransmitted again. On the other side this applications are not slow down by the confirmation process and the memory will be available for work much faster.

SIMULATION RESULTS:

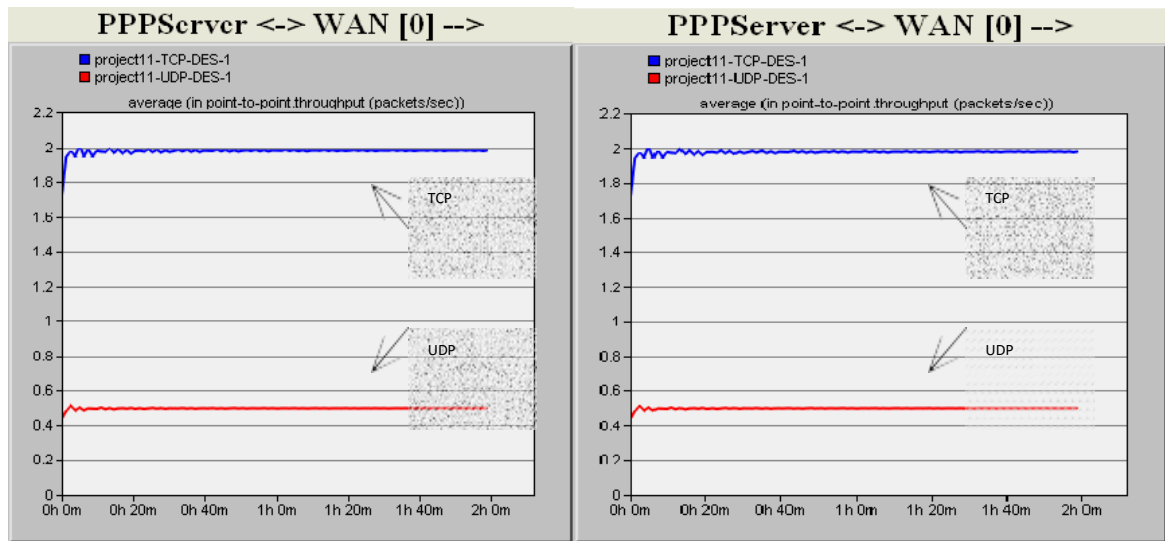
The simulation time is set for two hours data transfer between LAN network and the server with no packet latency and packet discard ratio of 0% while packets traverse thru the WAN. The task response time, in seconds, Fig. 1, shows how long the application need to be completed. The time when using TCP to complete the task is greater that the one using UDP. When using TCP, source and destination need to perform a three-way handshake before starting sending data and all amount of data need to be acknowledge by the destination when it is receive, so is taking more time than UDP, which doesn't perform this tasks.



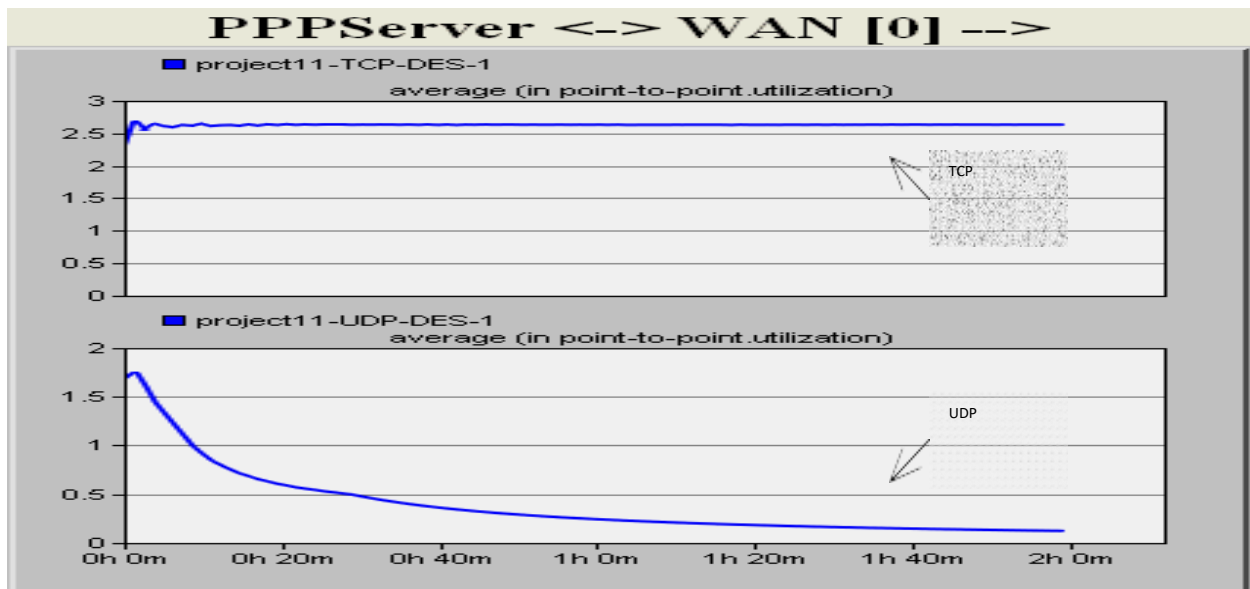
1. Response time for TCP and UDP



2. Traffic received (packets/sec) for the server



3. Traffic/Link utilization from the WAN to the server



Link utilization with a 0.5% packets discard ratio

The main difference between these two protocols is that TCP provides reliability and congestion control services, while UDP is orientated to improve performance.

The most important and common thing that TCP and UDP are using is the ability to set a host-to-host communication channel, so the packets will be delivered between processes running on two different computers. UDP is the right choice for application where reliability is not a must but the speed and performance is. Instead, TCP, even if it takes more time for the processes, has additional functions like same order delivery, reliability and flow control. As future work, we plan to conduct several studies regarding packets routing in computer networks to improve the fairness of data transmissions using different network protocols.

RESULT:

DATE: **8.a Simulation of Distance Vector Routing algorithm.**

AIM:

To implement the Distance – Vector Routing Algorithm

APPARATUS REQUIRED:

1. VI-RTSIM software.
2. Personal computer.

THEORY:

Distance Vector Algorithm:

- ❖ A Distance vector routing, each router periodically share its knowledge about the entire network with it's neighbors.
- ❖ The three keys to under this algorithm are
 1. Knowledge about the whole network.
 2. Routing only to neighbor.
 3. Information sharing at regular intervals.

Knowledge about the whole work:

- ❖ Each router shares its knowledge about entire network. It sends all of its collected knowledge about the network to its neighbors.

Routing only to neighbor:

- ❖ Each router periodically sends its knowledge about the network only to those routers to which it has direct links. It sends whatever knowledge it has.

Information sharing at regular intervals:

- ❖ The every 30 seconds, each router sends its information about the whole network to its neighbors.

Sharing Information:

- ❖ LAN's are connected by router, represented by the assuming A, B, C, D, E and F.
- ❖ Distance vector routing simplifies the routing process by assuming a lost of one unit for every link.
- ❖ The efficiency of transmission is a function only of the number of links required to reach a destination. In this, the cost on hop count.

Routing Table:

- ❖ Each router gets its initial knowledge about the internet work and how it uses shared information to update that knowledge.
- ❖ The routing table has e columns network lost router ID.
- ❖ The first block is final destination of packet.
- ❖ The second block is no of hop count.
- ❖ The third block is that to which a packet delivers must.

Updating algorithm:

- ❖ Updating algorithm requires that the router first has one hop to the hop count field for each advertised router.
- ❖ The router should apply the below rules to each router, if the advertised destination is not in routing table
- ❖ If next hop field is same, router should replace the entry in the table with advertised one.
- ❖ If next hop field is same, router should replace the entry in the table with advertised one.
- ❖ . If next hop field is not the same, advertised hop count is smaller than the one in the table, the router should replace the entry in the table with new one.
- ❖ IF advertised hop count is not smaller, the router should do no routing.

PROCEDURE

1. Open VI-RTSIM software from desktop
2. Click the Simulation menu bar
3. Select the “Distance – Vector Routing Algorithm” option from Routing algorithm menu bar.
4. Network with routers connected through link is drawn by using option in editor(add router, join link, delete router, delete link, Add caption to link, add caption to router)
5. Select any two nodes to find the shortest distance between them.
6. Click the Find path Button to run the program.
7. Now the shortest paths between the two nodes are calculated.

Distance Vector Routing Algorithm

Find! Editor

From: 2 To: 5

```

graph TD
    2((2)) --- 800 --- 3((3))
    3 --- 1200 --- 4((4))
    4 --- 1500 --- 5((5))
    4 --- 1000 --- 6((6))
    5 --- 250 --- 6
    1((1)) --- 300 --- 2
    1 --- 1000 --- 3
    1 --- 1700 --- 8((8))
    6 --- 900 --- 7((7))
    6 --- 1400 --- 8
    7 --- 1000 --- 8
    style 2 fill:#ffff00
    style 5 fill:#ffff00
    style 1 fill:#ff0000
    style 3 fill:#ff0000
    style 4 fill:#ff0000
    style 6 fill:#ff0000
    style 7 fill:#ff0000
    style 8 fill:#ff0000
  
```

The path is: 5 <- 6 <- 4 <- 3 <- 2

The distance is: 3250

You should click on any two objects to find the path

Available Paths :

Paths from 2 to 5

1. 2->1->3->4->5
2. 2->1->3->4->6->5
3. 2->1->8->6->5
4. 2->1->8->6->4->5
5. 2->1->8->7->6->5
6. 2->1->8->7->6->4->5
7. 2->3->1->8->6->5
8. 2->3->1->8->6->4->5
9. 2->3->1->8->7->6->5
10. 2->3->1->8->7->6->4->5
11. 2->3->4->5
12. 2->3->4->6->5

Find Path Quit

Find Shortest Path



Distance vector table:

To	1	2	3	4	5	6	7	8
1	0	300	1000	0	0	0	0	1700
2	300	0	800	0	0	0	0	0
3	1000	800	0	1200	0	0	0	0
4	0	0	1200	0	1500	1000	0	0
5	0	0	0	1500	0	250	0	0
6	0	0	0	1000	250	0	900	1400
7	0	0	0	0	0	0	0	1000

Calculate

Distance: From: 2 To:

Node	1	2	3	4	5	6	7	8
Distance	300	0	800	2000	3250	3000	3000	2000

Path:

Node	1	2	3	4	5	6	7	8
Path	2	0	2	3	6	4	8	1

The path is: 5 < 6 < 4 < 3 < 2

The distance is: 3250

DATE:

8.b Simulation of Link State Routing Algorithm.

AIM:

To implement the Link State Routing Algorithm

APPARATUS REQUIRED:

1. VI-RTSIM software.
2. Personal computer.

THEORY:

Link State Vector Algorithm:

- ❖ In Link state routing, each router share its information of its neighbors with every other router in the inter-network.

Knowledge about the neighborhood:

- ❖ Instead of sending its entire routing table, a router sends information about its neighborhood only.

To all router:

- ❖ Each router send this information to every other router on the internetworking, not just to its neighbors.
- ❖ If s does so by a process called “flooding” it means that a router sends its information.

Information sharing when there is a Change:

- ❖ Each router sends out information about the neighbors when there is a change.

Information sharing:

- ❖ Link state routing process use the same internet work as distance vector algorithm.
- ❖ Here each other sends its knowledge about is neighbors to every other router in the internet work.
- ❖ Cost is applied only by routers and not by any other station on a network, if cost was added by every station, instead of by routers alone, it would accumulate unpredictably.
- ❖ Cost is applied as a packet leaves the router rather than as if enters. Most networks are broadcast networks. When a packet is in network every station, including the router, can pick it up, we cannot assign any cost to a packet.

Link state packet:

- ❖ When a router floods the network with information about its neighborhood, it is said to be advertising. The basis of this advertising is a short packet called a link state packet (LSP).

Advertiser	Network	Cost	Neighbor
------------	---------	------	----------

Getting information about neighbors:

- ❖ A router gets its information about its neighbors by periodically sending them a short greeting packet.
- ❖ If the neighbor responds to the greeting as expected, it is assumed to be alive and functioning.

Initialization:

- ❖ Imagine that all routers in our sample internet work come up at the same time.
- ❖ Each router sends a greeting packet to its neighbors to find out the state of each link.

Link – State Database:

- ❖ Every router every LSP and puts the information into a link-state database.
- ❖ Because every router receives the same LSPs every router builds the same database.
- ❖ It stores this database on its disk and uses it to calculate its routing table. If a router is added to be deleted from the system, the whole database must be shared for fast updating.

PROCEDURE

1. Open VI-RTSIM software from desktop
2. Click the Simulation menu bar
3. Select the “Link State Routing Algorithm” option from Routing algorithm menu bar.
4. Network with routers connected through link is drawn by using option in editor(add router, join link, delete router, delete link, Add caption to link, add caption to router)
5. Select any two nodes to find the shortest distance between them.
6. Click the Find path Button to run the program.
7. Now the shortest paths between the two nodes using link state routing algorithm was calculated.

Link State Routing

Find! Editor

From: A To: F

```
graph TD; A --- B; B --- C; A --- E; E --- B; E --- C; D --- E; D --- G; E --- F;
```

Available Paths :

Paths from A to F

1. A → B → C → F
2. A → B → C → E → F
3. A → B → E → F
4. A → B → E → C → F
5. A → D → E → F
6. A → D → E → B → C → F
7. A → D → E → C → F
8. A → E → F
9. A → E → B → C → F
10. A → E → C → F

The path is: F ← E ← A

The distance is: 2

You should click on any two objects to find the path

Find Path Quit

Find Shortest Path



Link State table:

To	A	B	C	D	E	F	G
A	0	1	0	1	1	0	0
B	1	0	1	0	1	0	0
C	0	1	0	0	1	1	0
D	1	0	0	0	1	0	1
E	1	1	1	1	0	1	0
F	0	0	1	0	1	0	0
G	0	0	0	1	0	0	0

Calculate

Distance: From: A To:

Node	A	B	C	D	E	F	G
Distance	0	1	2	1	1	2	2

Path:

Node	A	B	C	D	E	F	G
Path	0	1	2	1	1	5	4

The path is: F <- E <- A

The distance is: 2

DATE: **9. Simulation of error correction code (like CRC)**

AIM:

To implement and check the error detection/error correction techniques in networks using a c program.

APPARATUS REQUIRED:

1. Pc-ino
2. C/c++compiler

THEORY:

Error Detection

- Bit errors occur in frames due to electrical interference or thermal noise.
- Detecting errors is one part of the problem; correcting errors is the other.
- What happens when an error is detected?
- Two basic approaches:
 - Notify the sender that message is corrupt so the sender can retransmit it; (most often used in every day applications)
 - Use an error-correcting code to reconstruct the correct message

Transmission Errors

- External electromagnetic signals can cause incorrect delivery of data
 - · Data can be received incorrectly
 - · Data can be lost
 - · Unwanted data can be generated
- Any of these problems are called *transmission errors*

Error Detection

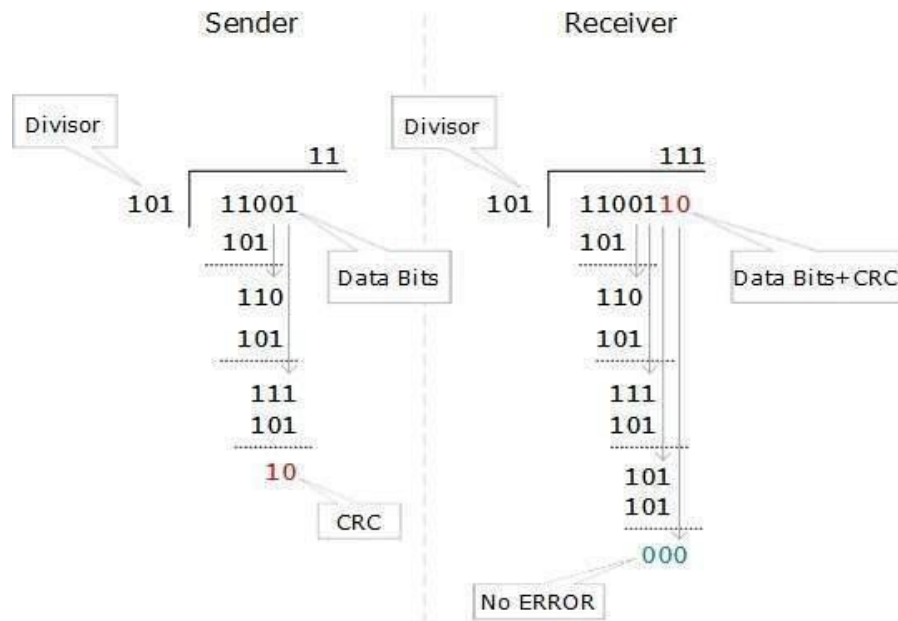
- Detecting Transmission Errors: basic idea is to add redundant information to a frame that can determine if errors have been introduced.

Error Correction or Error Detection?

- When error is detected, frame is discarded and resent, using bandwidth and causing latency, waiting for its arrival.
- Error correction requires additional bit to be sent with every frame.
- Correction is useful when
 - 1) errors are probable or
 - 2) the cost of retransmission is too high

Cyclic Redundancy Check (CRC)

CRC is a different approach to detect if the received frame contains valid data. This technique involves binary division of the data bits being sent. The divisor is generated using polynomials. The sender performs a division operation on the bits being sent and calculates the remainder. Before sending the actual bits, the sender adds the remainder at the end of the actual bits. Actual data bits plus the remainder is called a codeword. The sender transmits data bits as code words.



At the other end, the receiver performs division operation on codewords using the same CRC divisor. If the remainder contains all zeros the data bits are accepted, otherwise it is considered as there some data corruption occurred in transit.

PROCEDURE:

- Start the process.
- Give the data which is the message.
- Compile and run the program.
- Enter the received hamming code.
- The error is corrected codeword.

PROGRAM

OUTPUT

RESULT:

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)
Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMMON TO: DEPARTMENT OF INFORMATION TECHNOLOGY



**CS8582– OBJECT ORIENTED ANALYSIS AND
DESIGN LABORATORY**

R 2017

LAB MANUAL

INDEX

S.No.	Name of the Experiment	Page No	Date of Experiment	Remarks
1	Case Tools			
2	Passport automation system			
3	Book bank			
4	Exam Registration			
5	Stock maintenance system			
6	Online course reservation system			
7	Airline/Railway reservation system			
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			
15	Library management system			
16	Student information 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
 - 4.1 component diagram
 - 4.2 deployment 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 objects, their links and their messages. They can also contain simple class instances and class utility instances.

During analysis, the semantics of the primary and secondary interactions are indicated. Design shows the semantics of mechanisms in the logical design of the system.

Toggle 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 the F5 key.

CONCLUSION:

Ex no:2

PASSPORT AUTOMATION SYSTEM

Date:

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 :

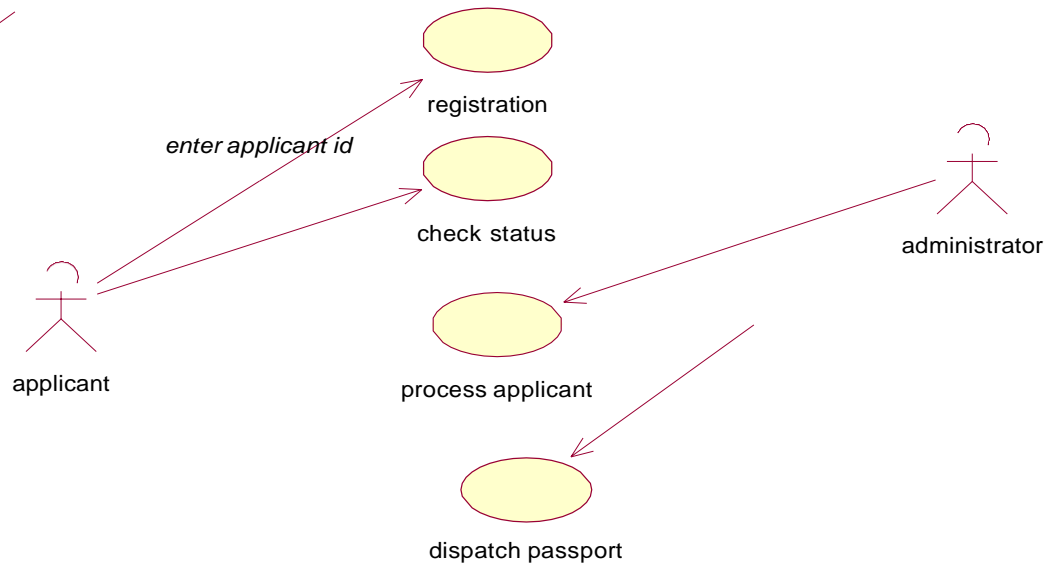


Fig.3. USECASE DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(IV) ACTIVITY DIAGRAM:

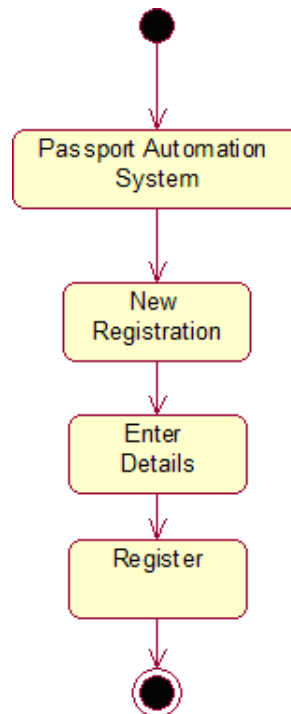


Fig.4.1. ACTIVITY DIAGRAM FOR REGISTER

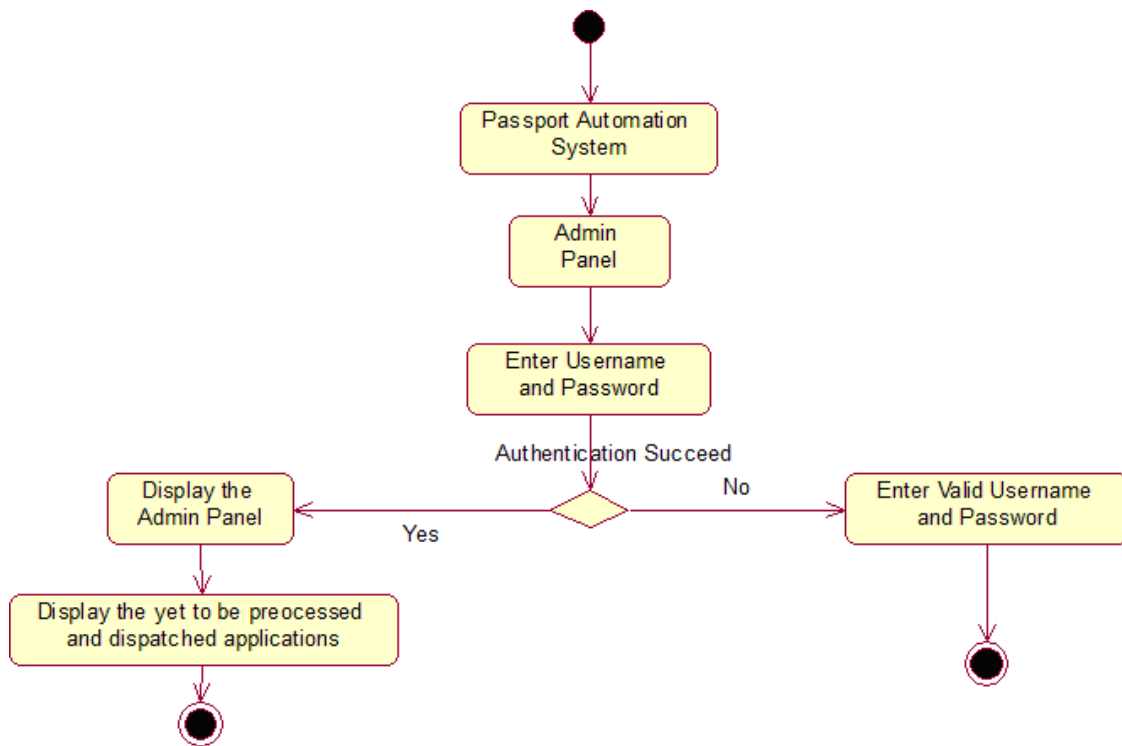


Fig.4.2. ACTIVITY DIAGRAM FOR ADMINISTRATION

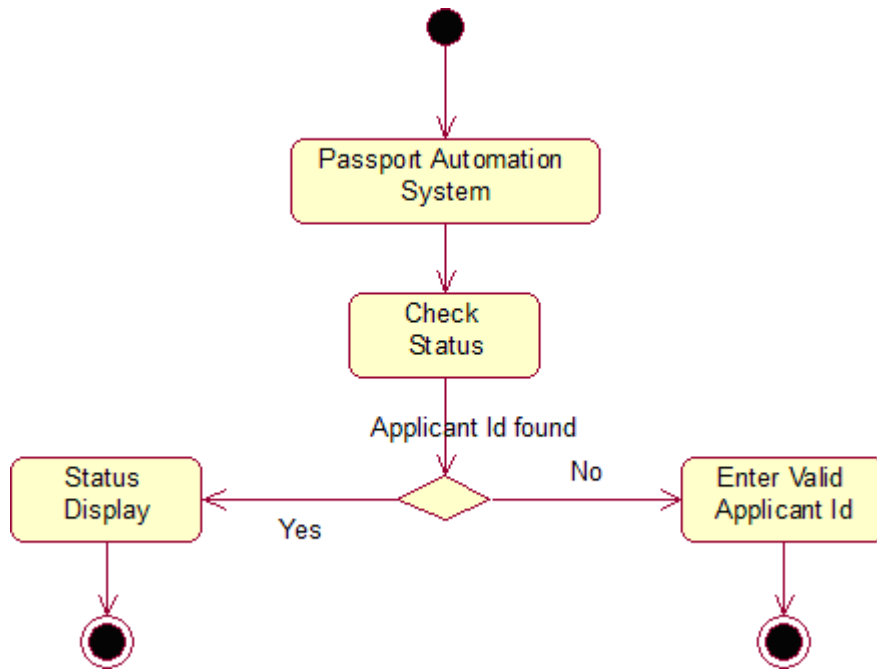


Fig.4.3. ACTIVITY DIAGRAM FOR CHECKING STATUS

(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 classes

1. New registration
2. Gender
3. Application Status
4. Admin authentication
5. Admin Panel

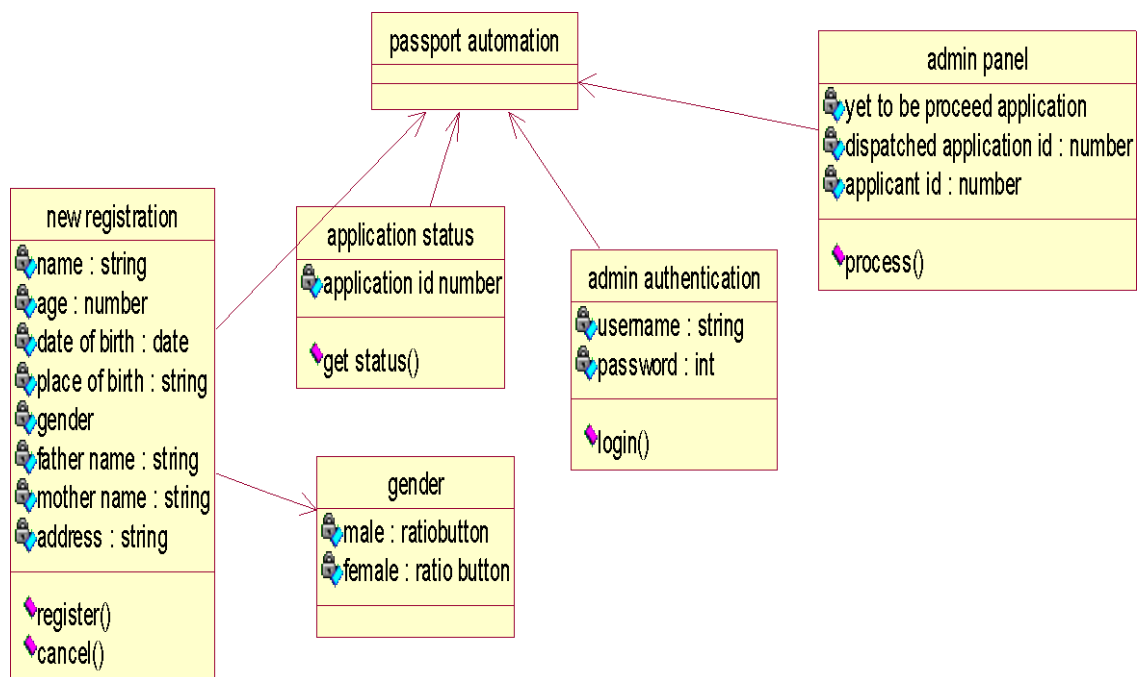


Fig.5. CLASS DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(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 “from” 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.

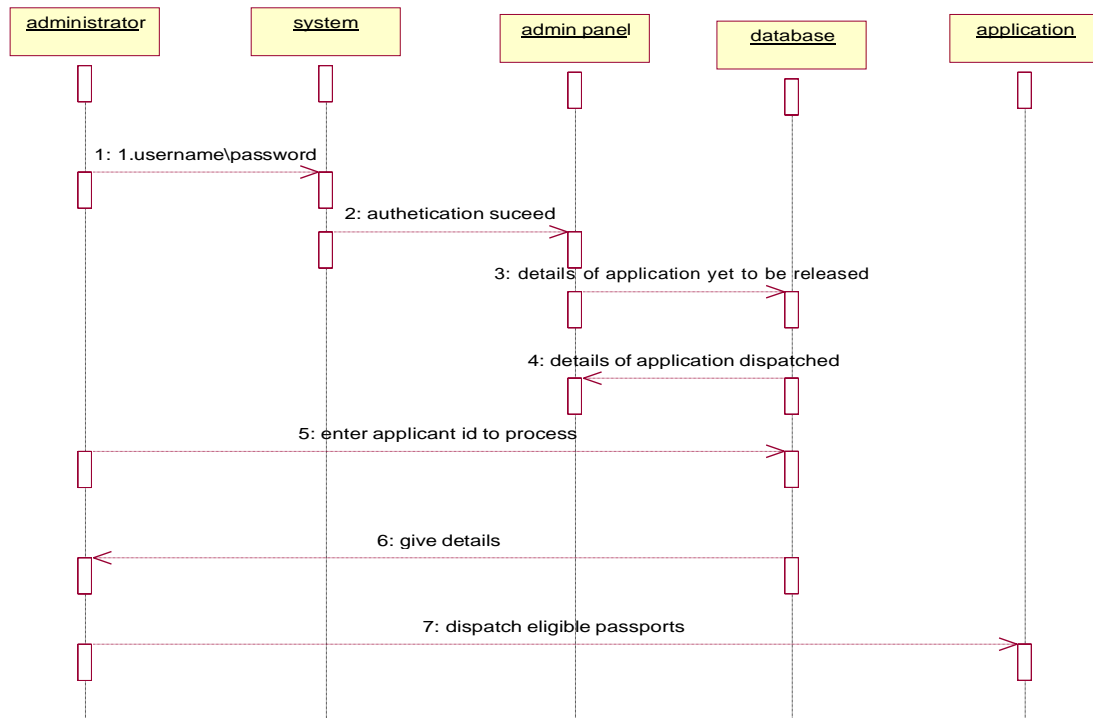


Fig.6.1.SEQUENCE DIAGRAM FOR ADMINISTRATOR

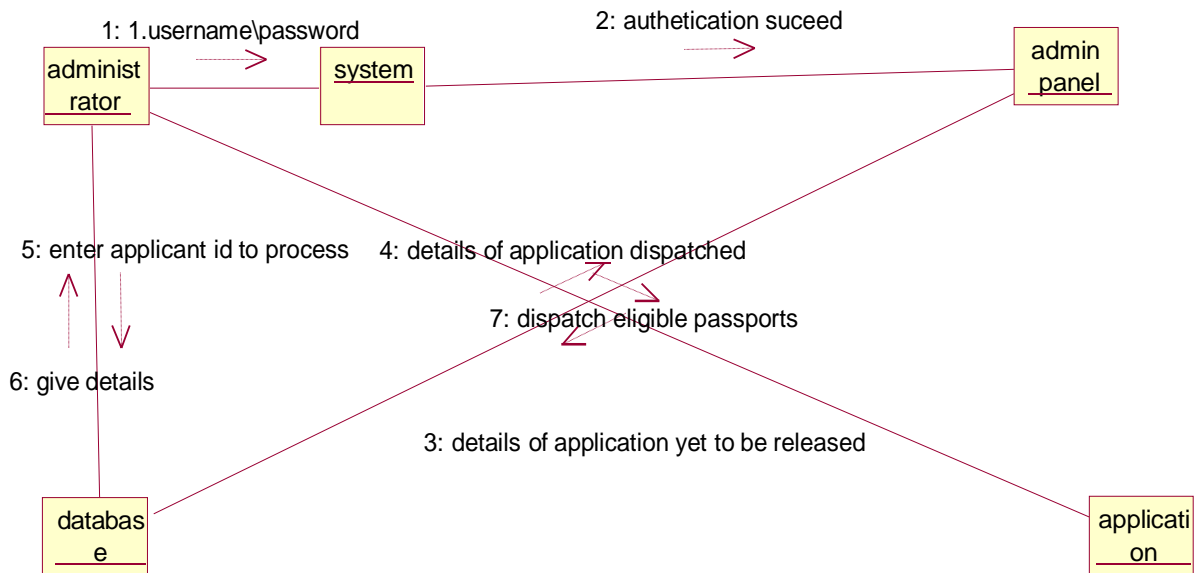


Fig.6.2.COLLABORATION DIAGRAM FOR ADMINISTRATOR

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.

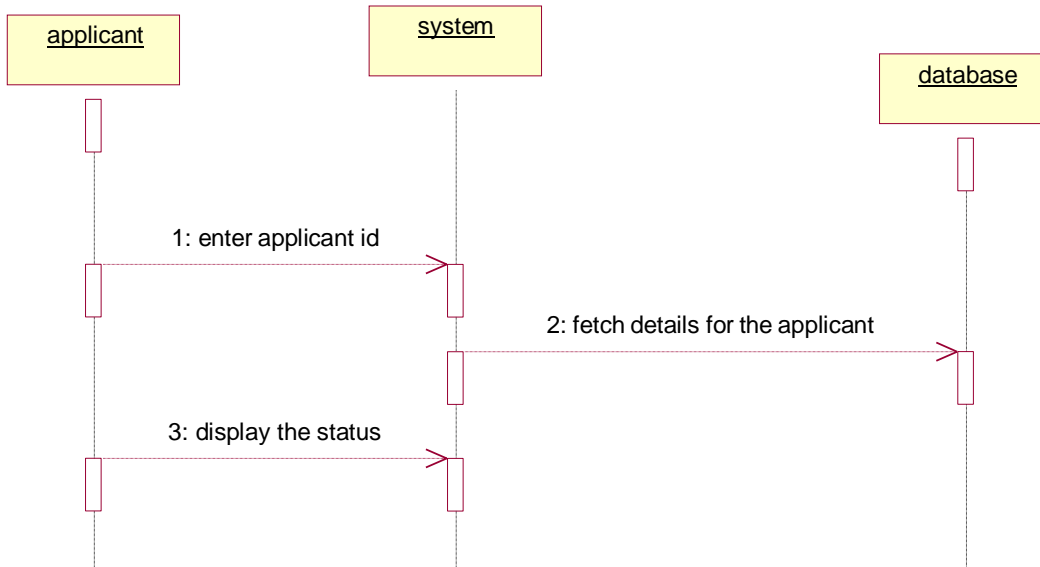


Fig.6.3.SEQUENCE DIAGRAM FOR CHECKING STATUS

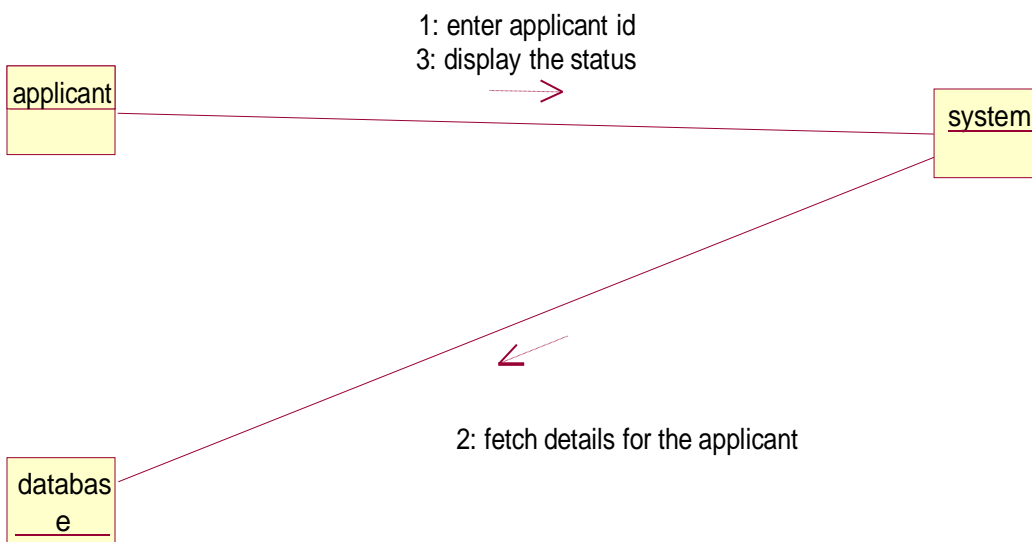


Fig.6.4.COLLABORATION DIAGRAM FOR CHECKING STATUS

The diagrams show the applicant enters his id and the system fetch the details from the database and display the status.

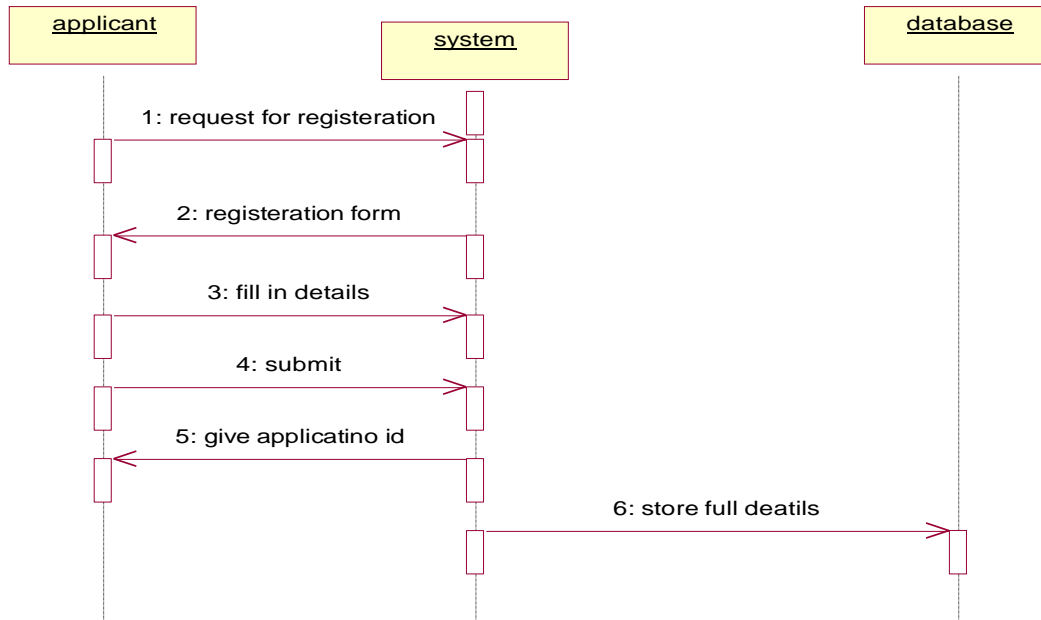


Fig.6.5.SEQUENCE DIAGRAM FOR NEW REGISTRATION

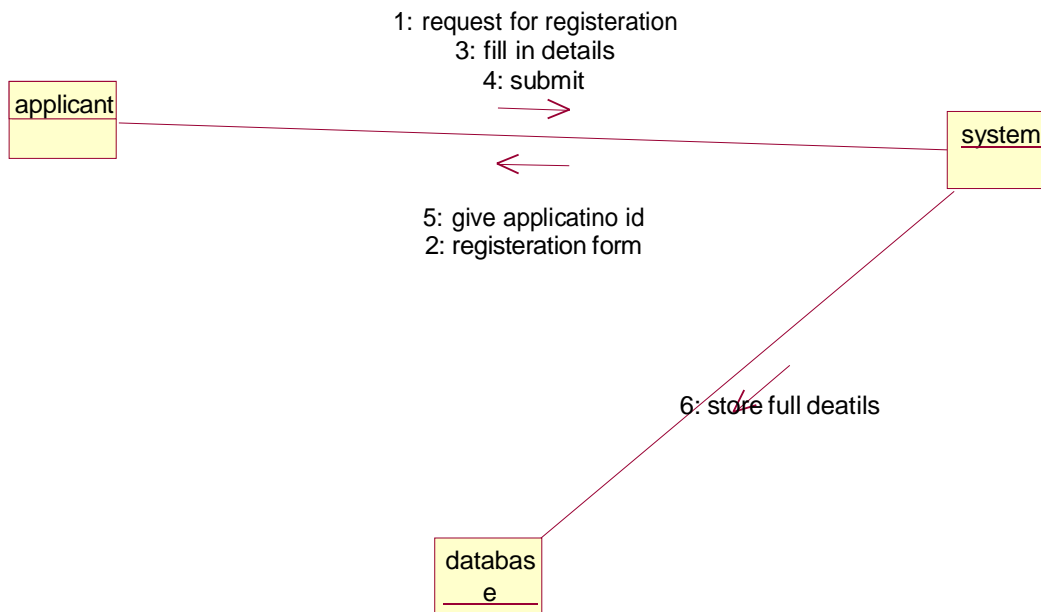
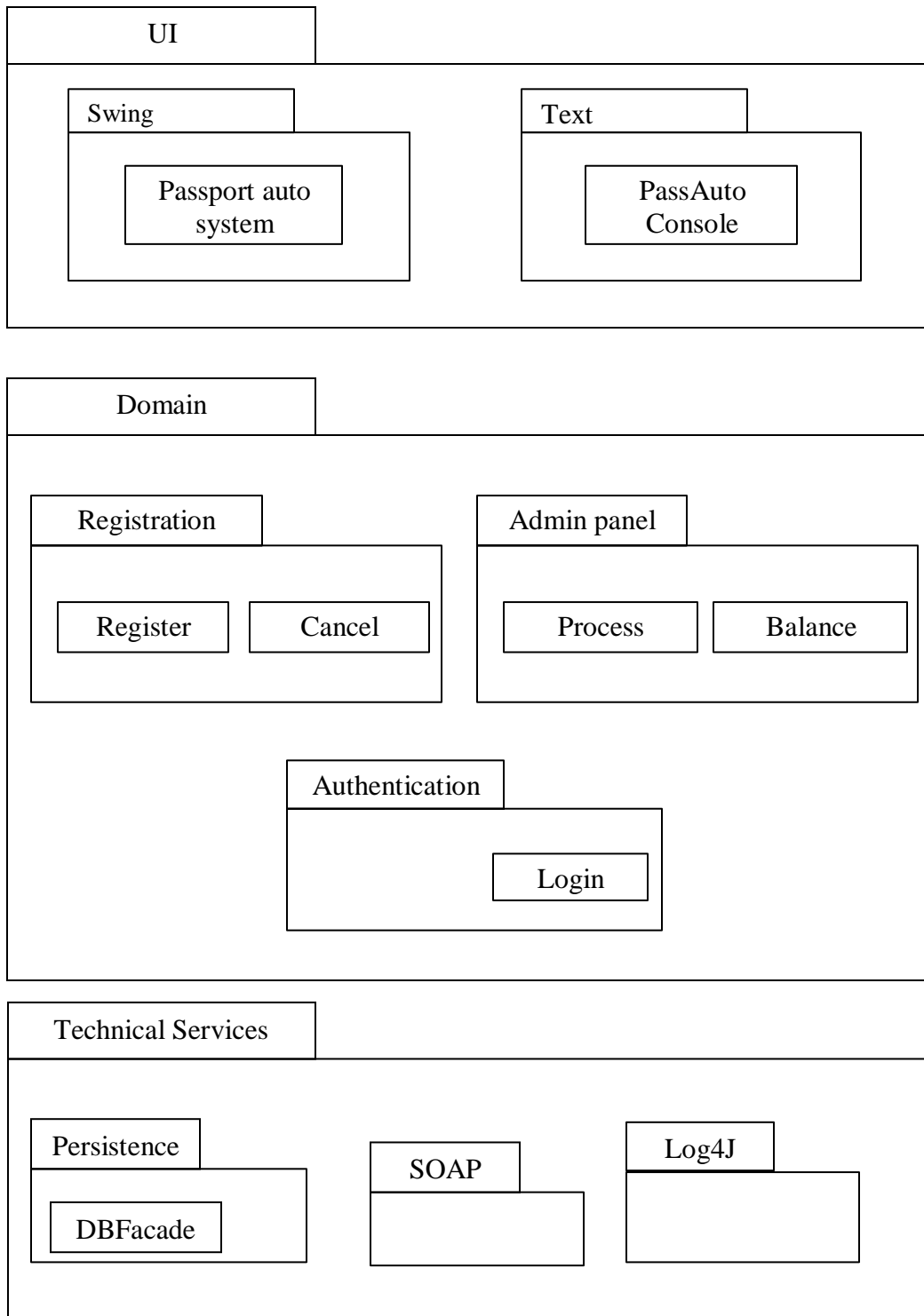


Fig.6.6.COLLABORATION DIAGRAM FOR NEW REGISTRATION

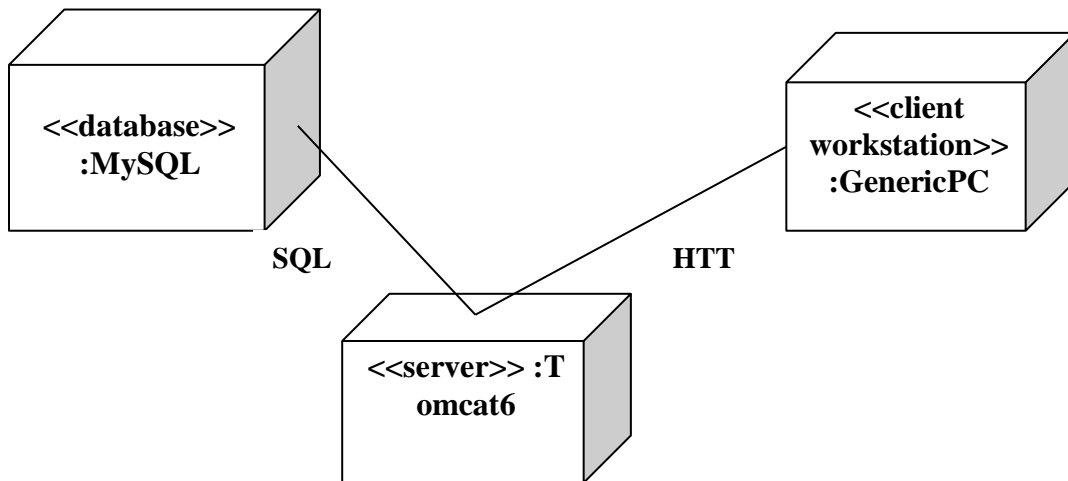
The diagrams show the applicant request the system for registration and the system provide the register form and applicant fill the form and submit and the system give the applicant id. The database stores the full details.

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

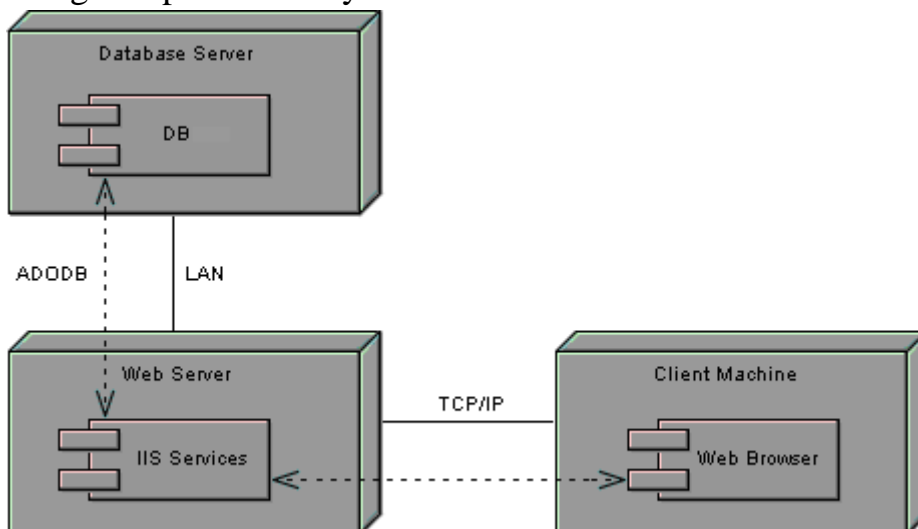
Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.



DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationship among components in system



RESULT:

Ex no:3

BOOK BANK SYSTEM

Date:

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:

2.1SOFTWARE 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

2.2HARDWARE 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:

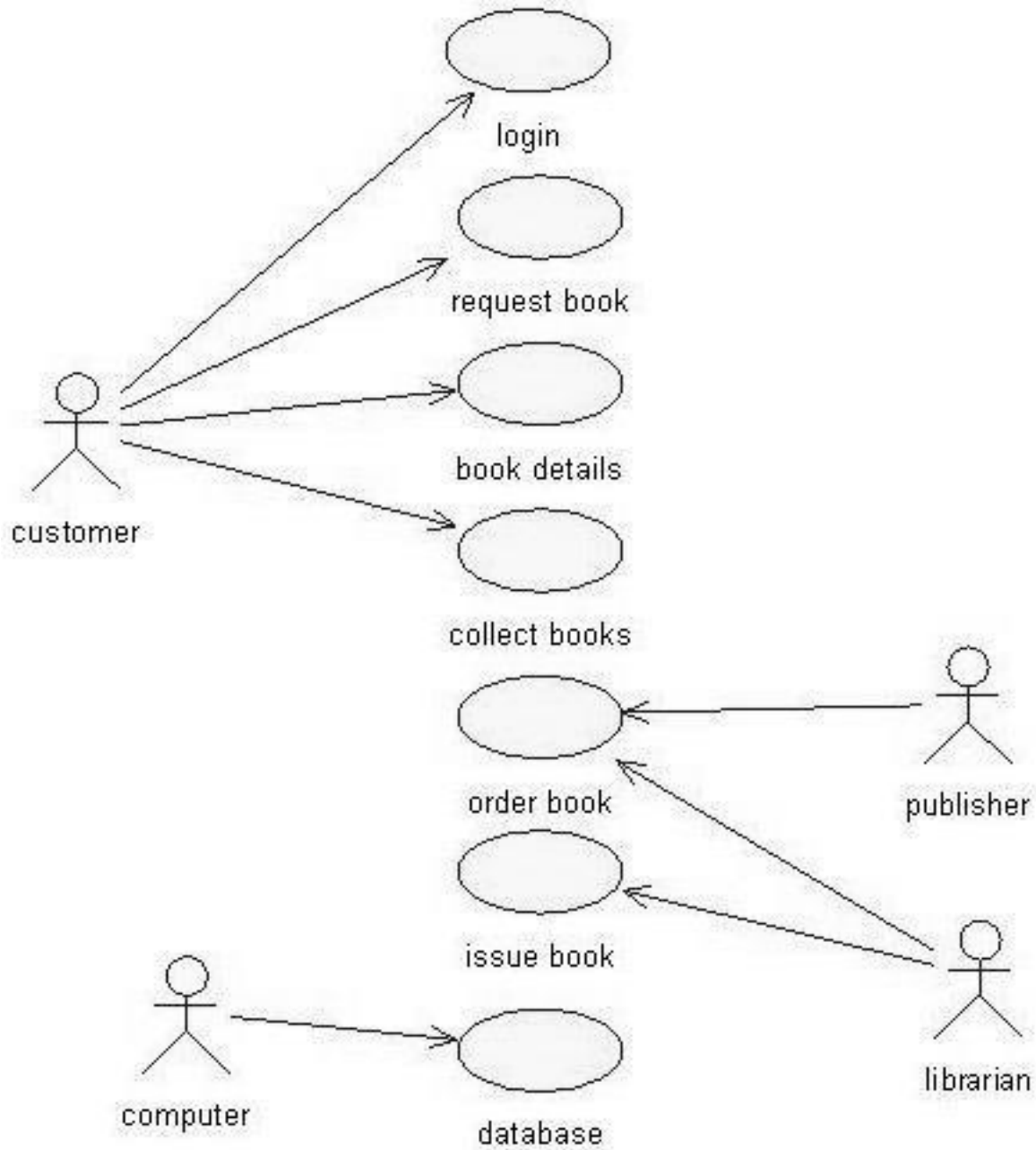


Fig 3. USE-CASE DIAGRAM FOR BOOK BANK SYSTEM

(IV) ACTIVITY DIAGRAM:

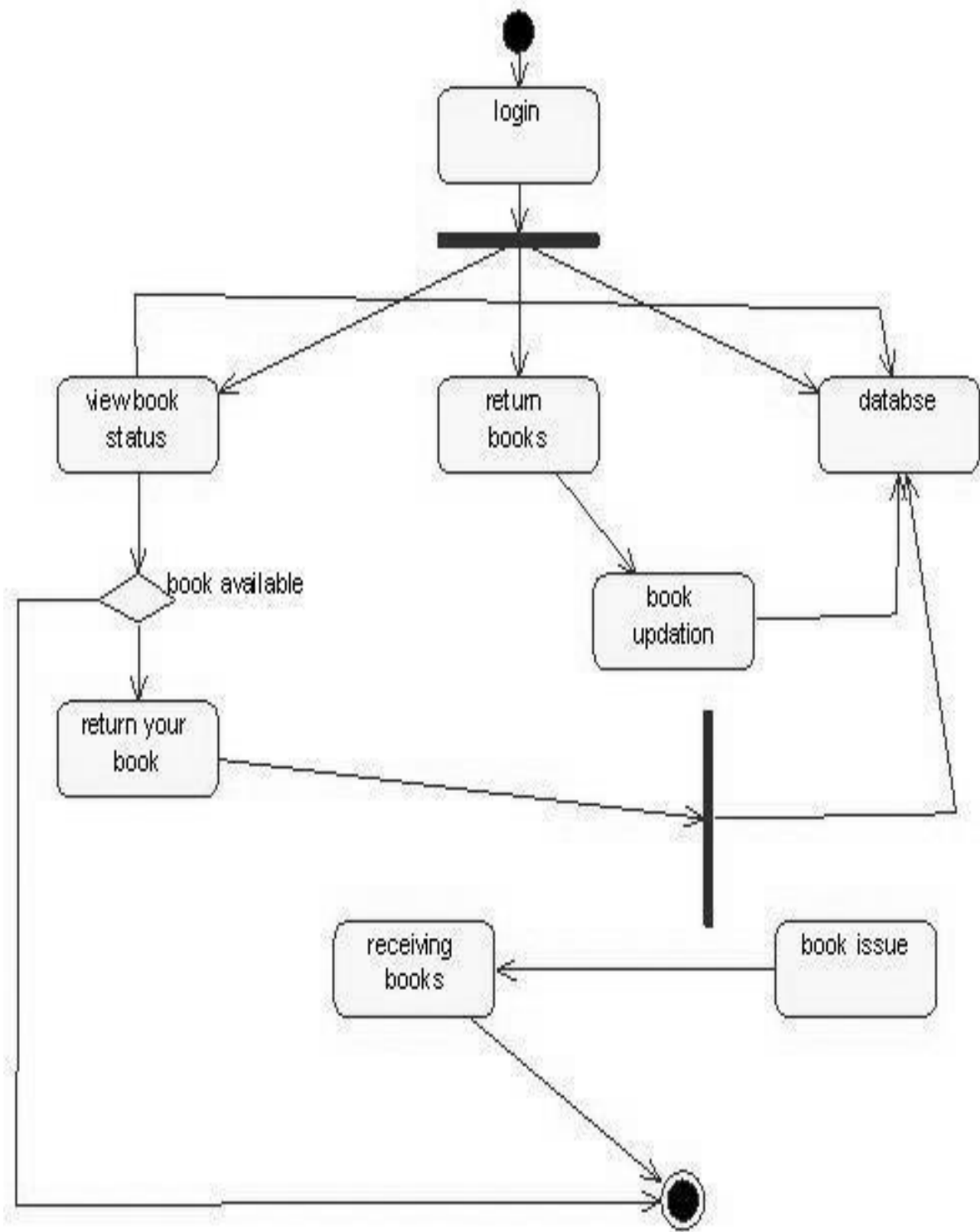


Fig.4. ACTIVITY DIAGRAM

(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.

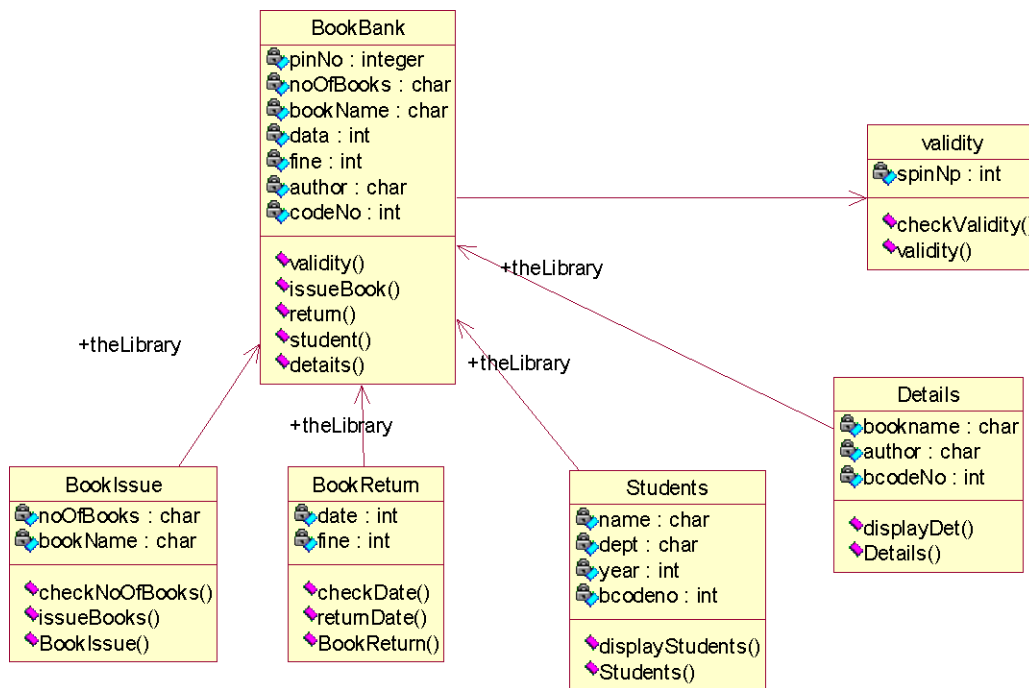


Fig.5. CLASS DIAGRAM FOR BOOK BANK SYSTEM

(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.

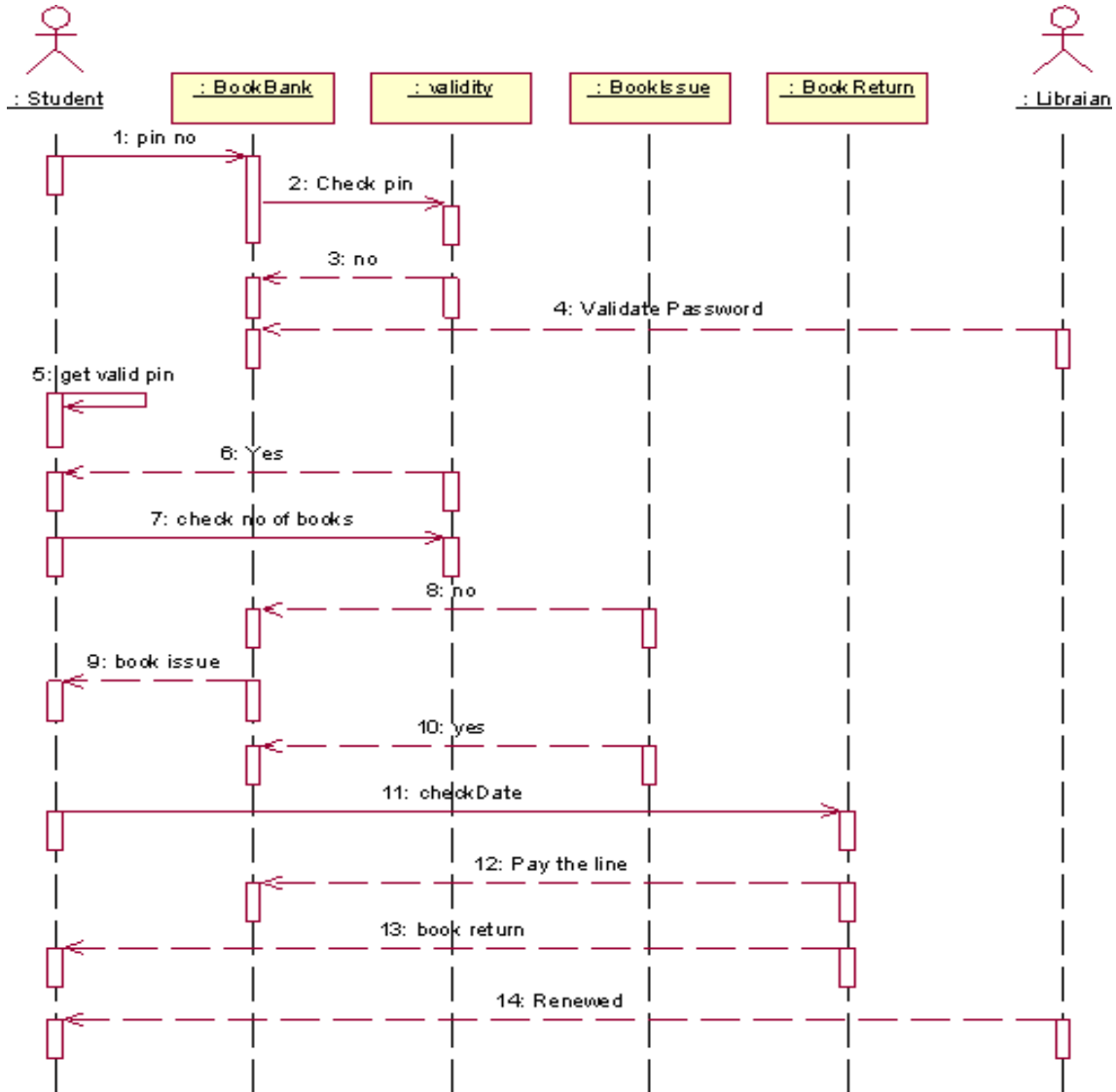


Fig. 6.1. SEQUENCE DIAGRAM FOR DEPOSIT PROCESS

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.

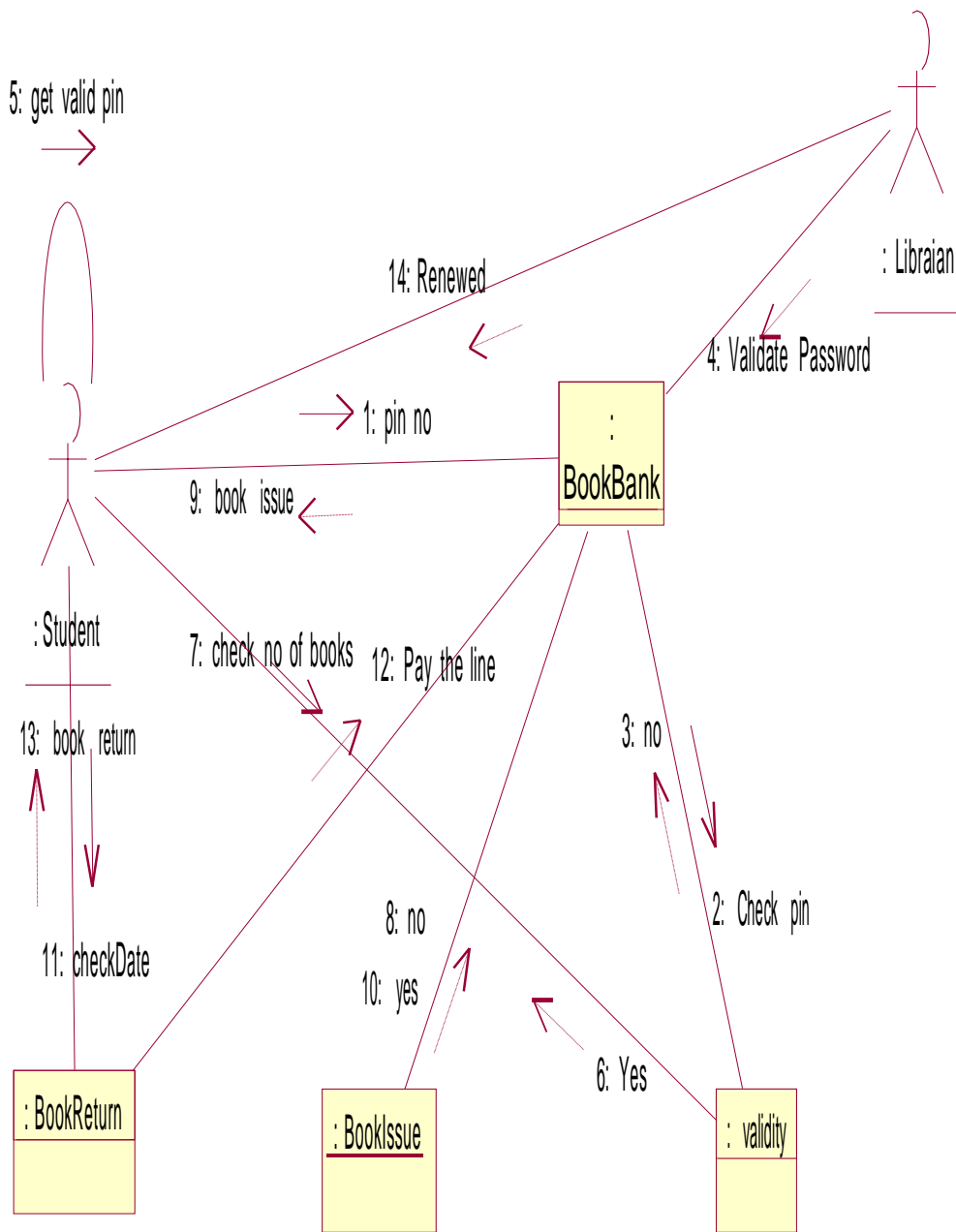
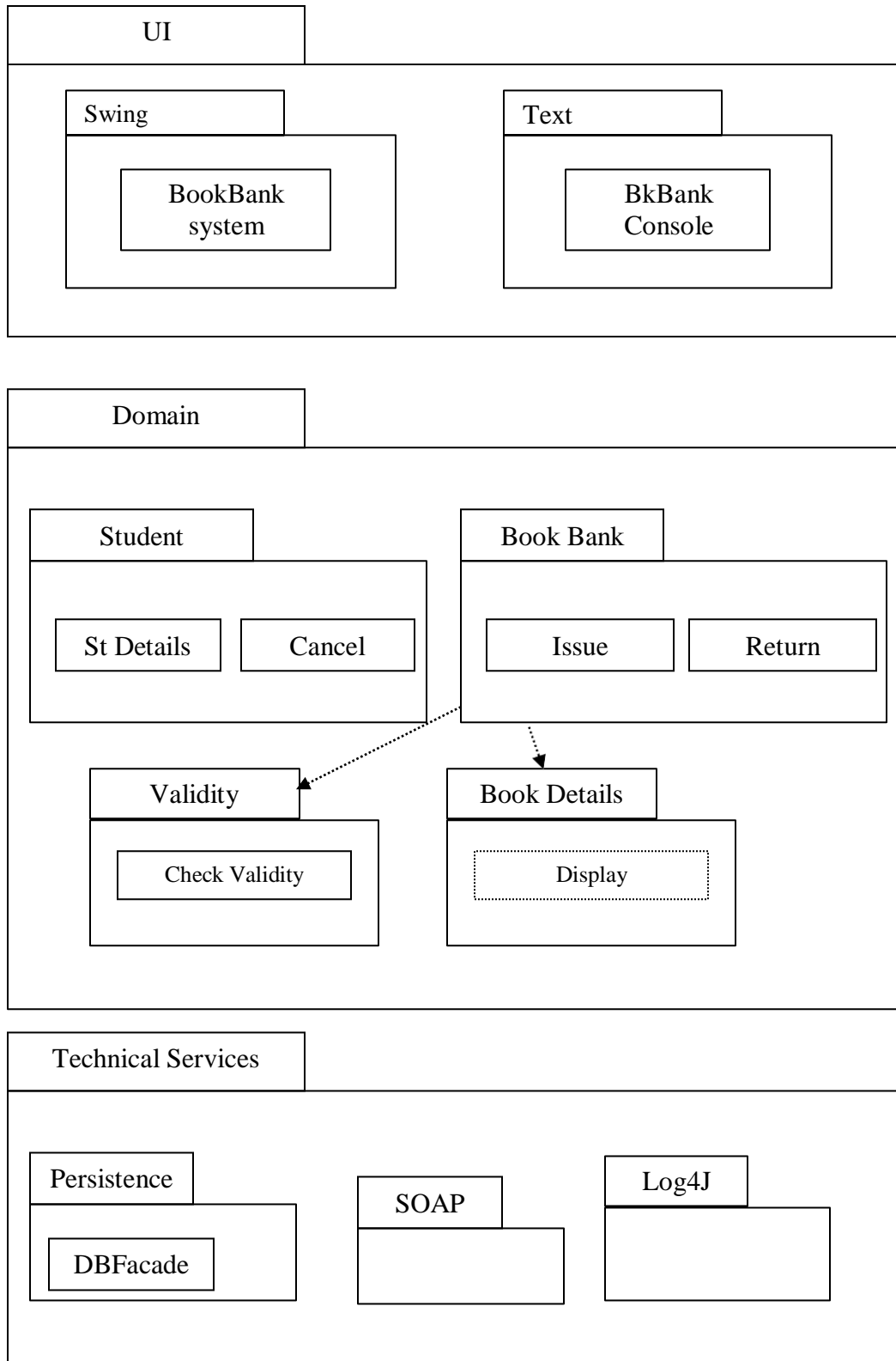


Fig. 6.2. COLLABORATION DIAGRAM FOR DEPOSIT PROCESS

(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.

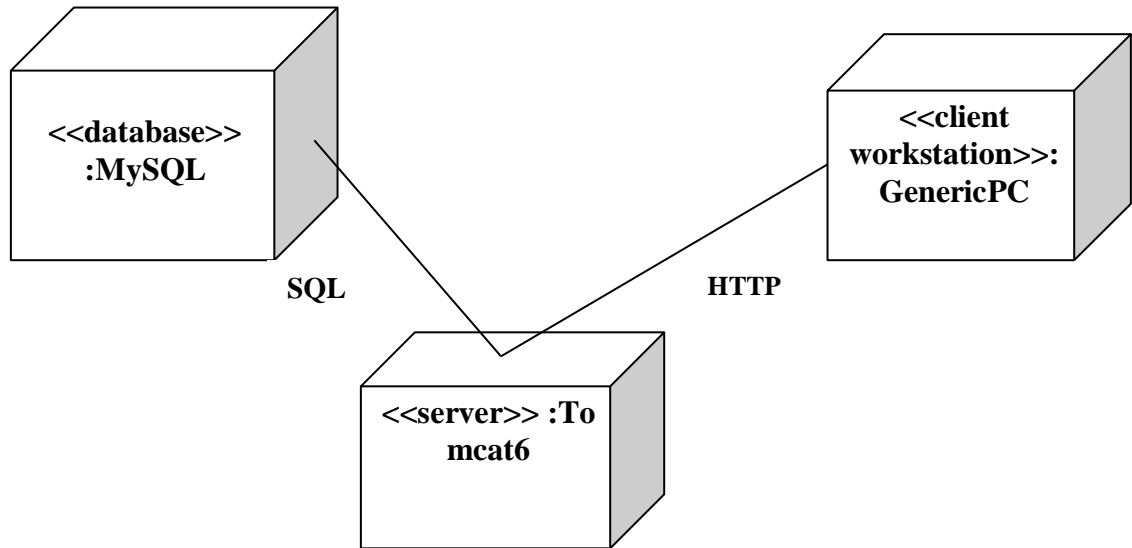


Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships

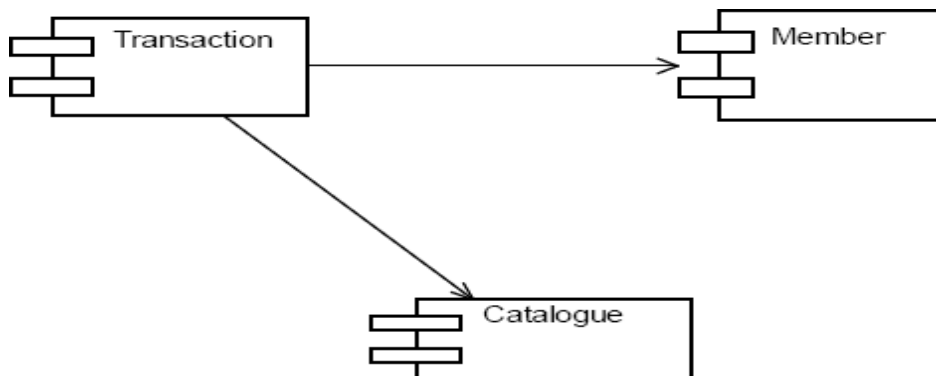


Fig.8.2.COMPONENT DIAGRAM

RESULT:

4. EXAM REGISTRATION SYSTEM

Date:

AIM:

To create a system to perform the Exam Registration system

(I) PROBLEM STATEMENT:

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

(II) SOFTWARE REQUIREMENT SPECIFICATION:

2.1 SOFTWARE 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(SQI Corporation).
- **Back End** - SQL 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) 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 :

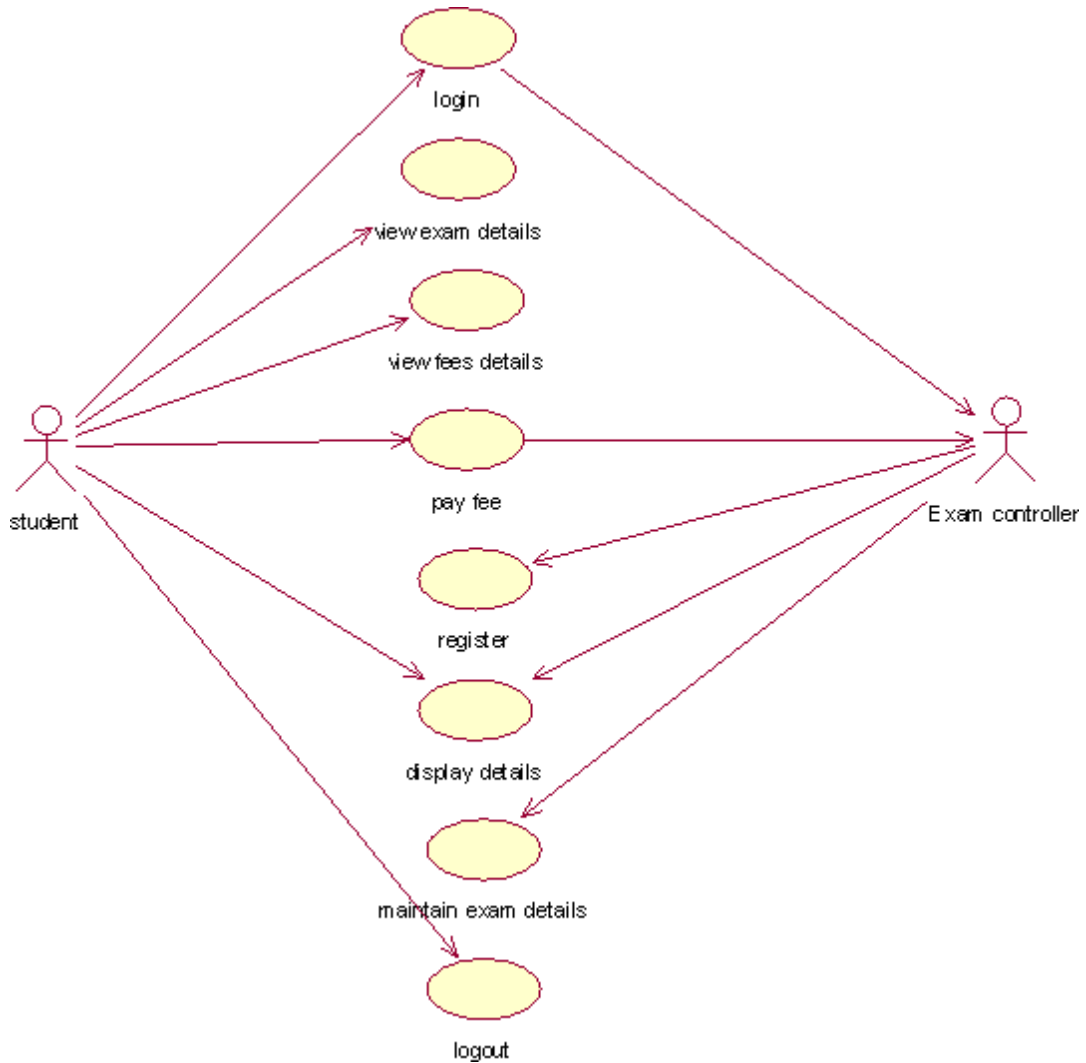


Fig. 3. USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(IV) ACTIVITY DIAGRAM:

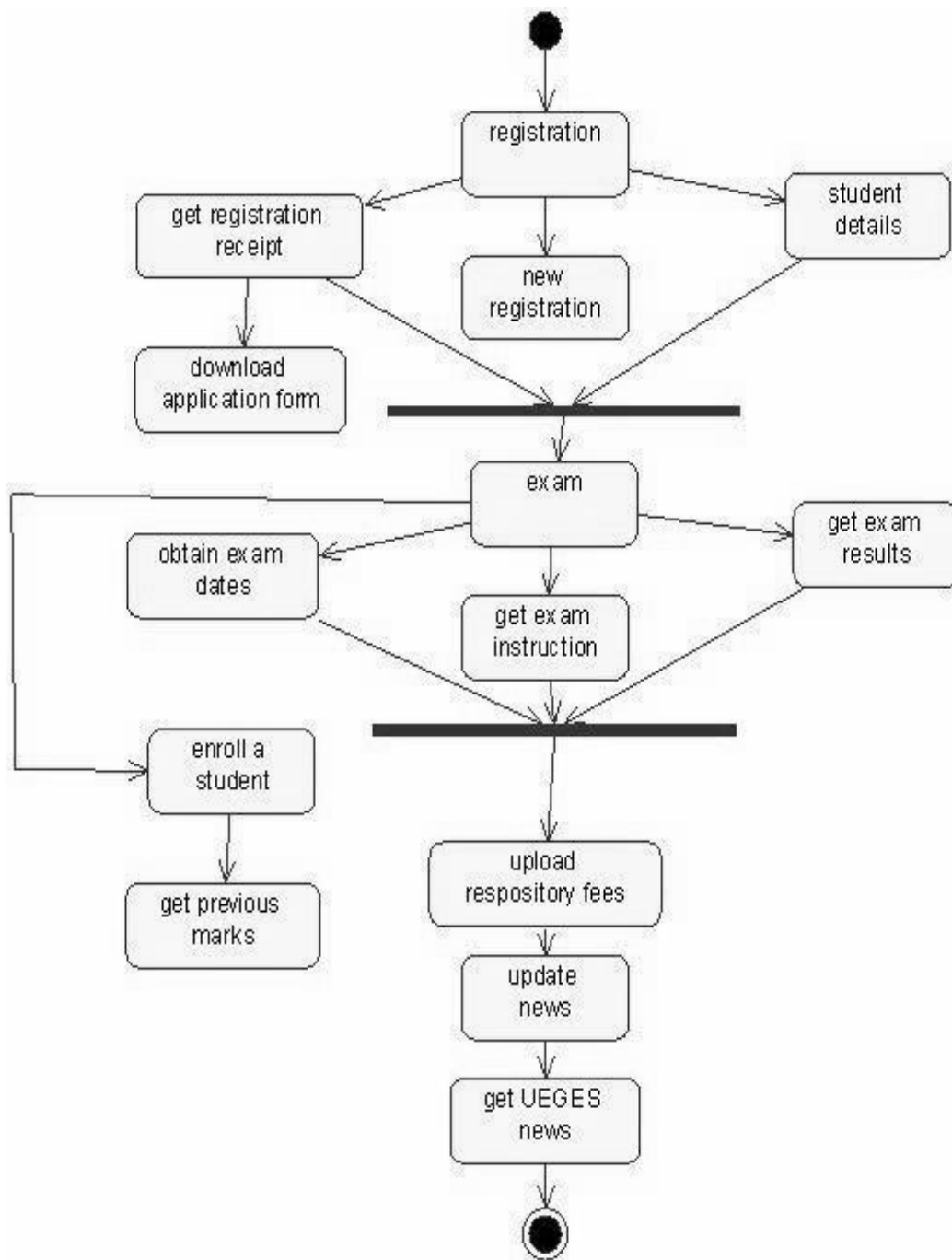


Fig. 4. USECASE DIAGRAM FOR EXAM REGISTRATION SYSTEM

(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.

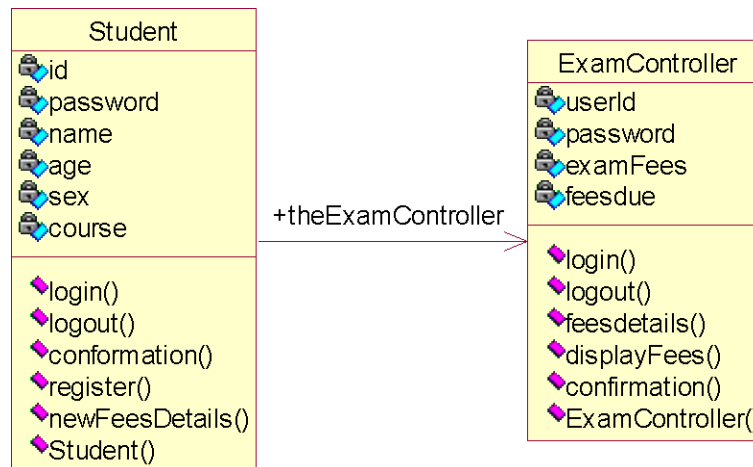


Fig.5. CLASS DIAGRAM FOR EXAM REGISTRATION SYSTEM

(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

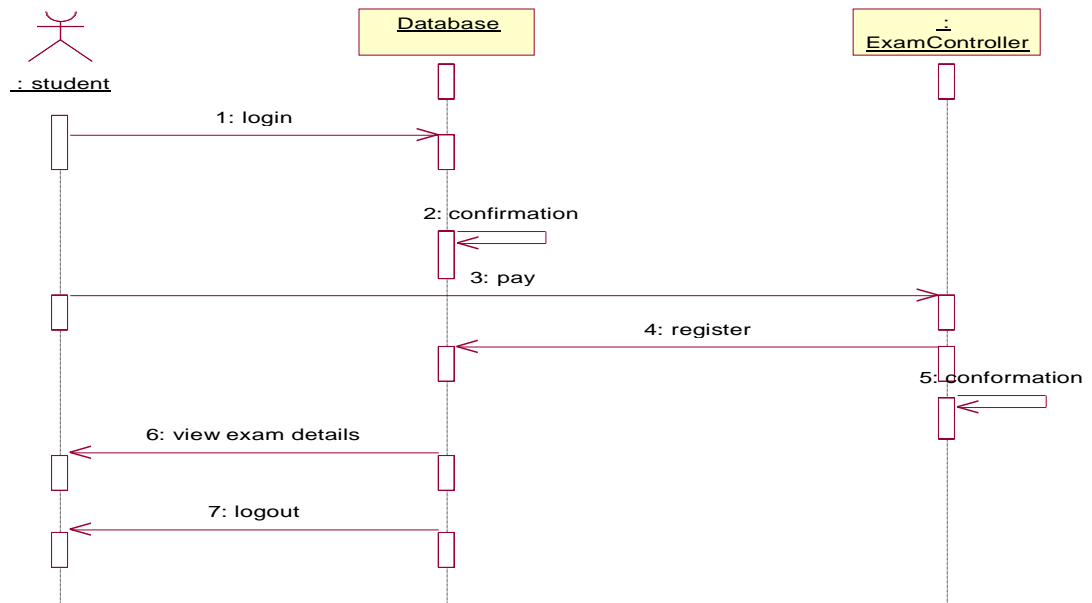


Fig. 6.1. SEQUENCE DIAGRAM FOR REGISTRATION SYSTEM

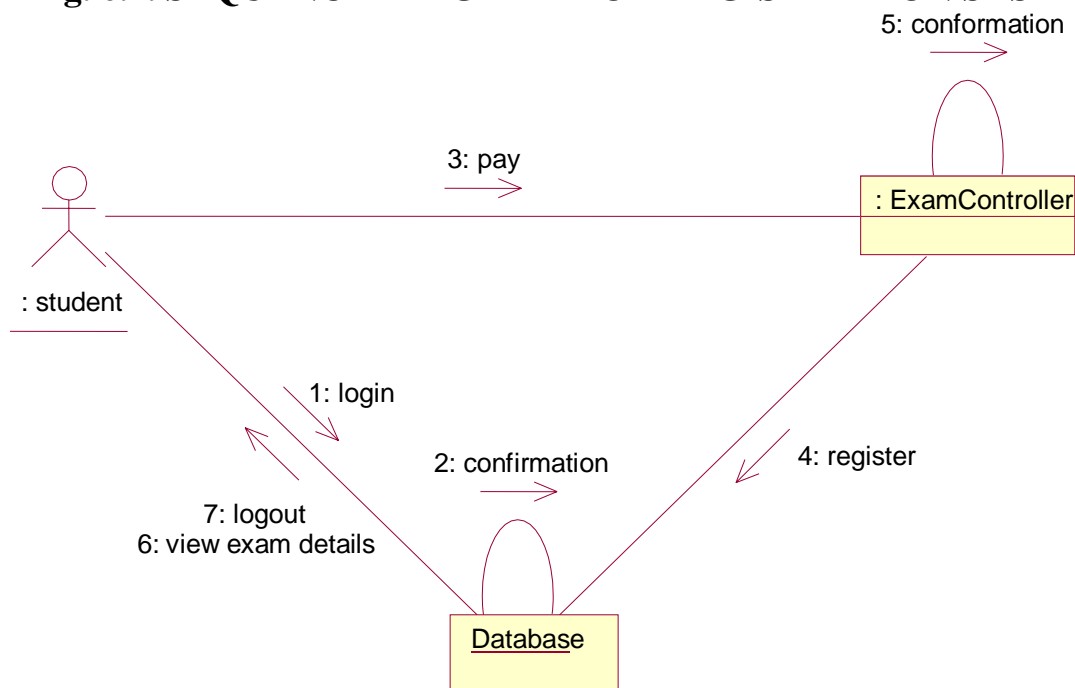
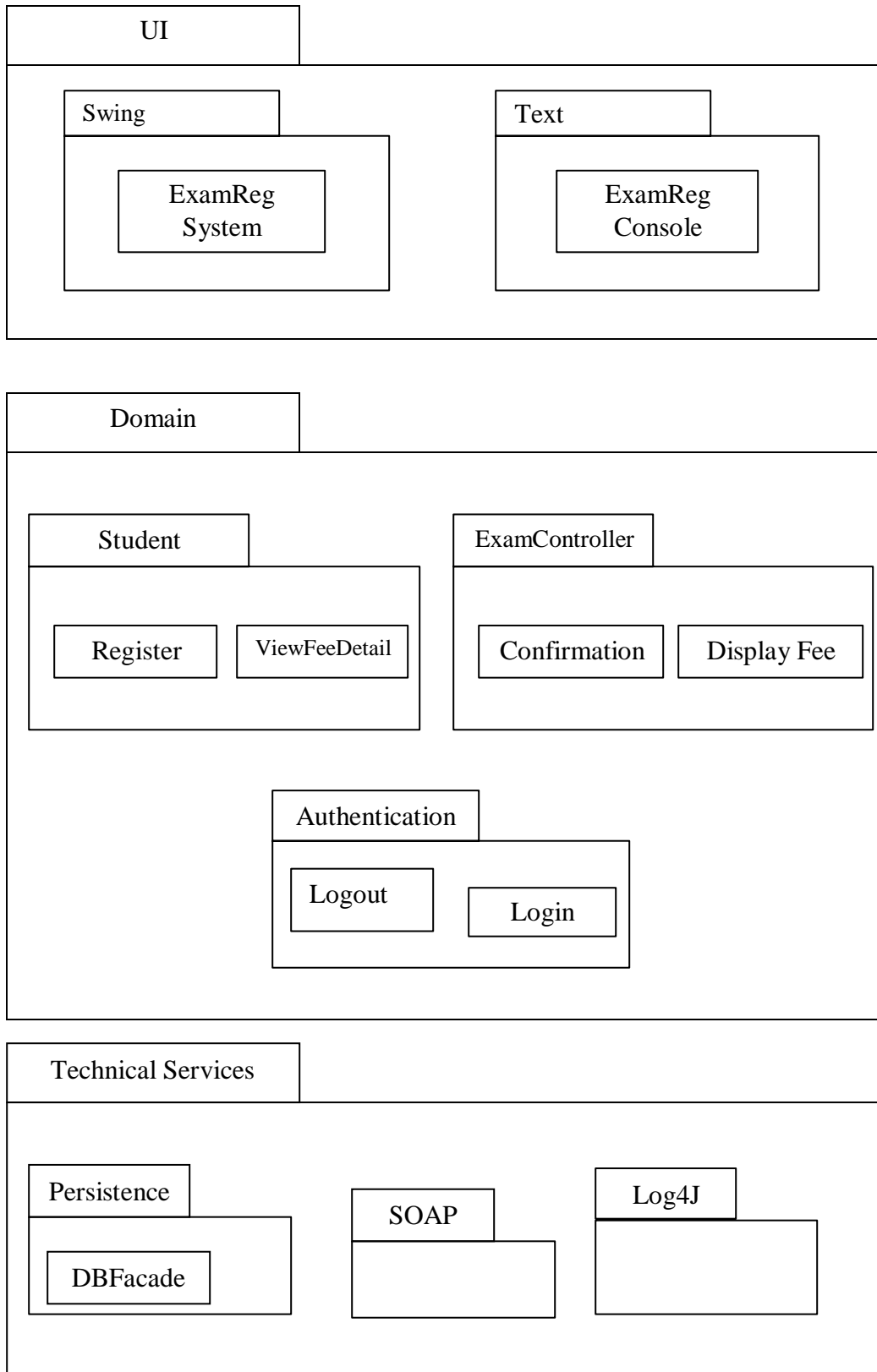


Fig. 6.2. COLLABORATION DIAGRAM FOR REGISTRATION SYSTEM

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:



(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.

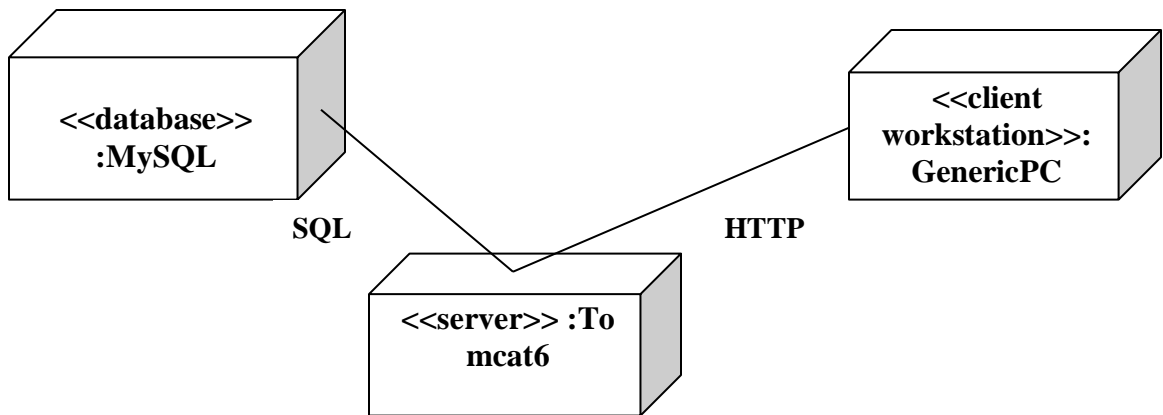


Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

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

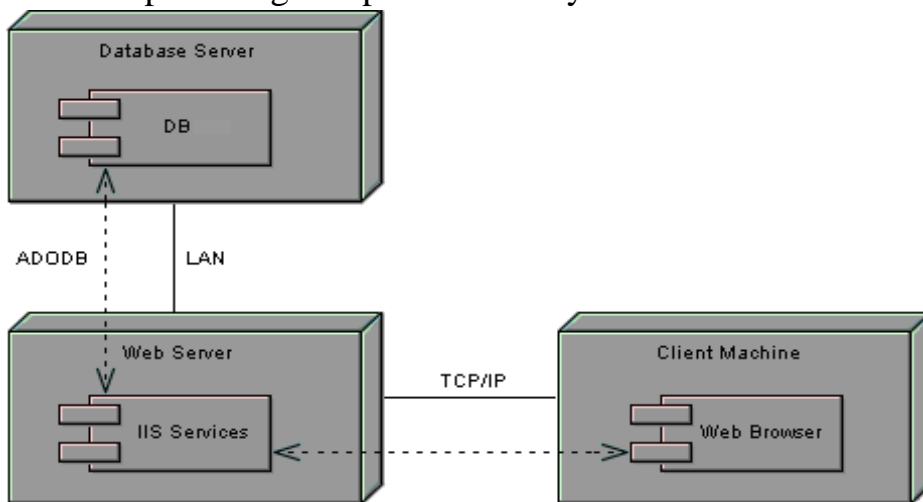


Fig.7.2.COMPONENT DIAGRAM

RESULT:

Ex no: 5

STOCK MAINTENANCE

Date:

AIM:

To create a system to perform the Stock maintenance

(I) PROBLEM STATEMENT

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

(II) SOFTWARE REQUIREMENT SPECIFICATION

1.1 PURPOSE

The entire process of Stock maintenance is 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 generalization among the use-cases

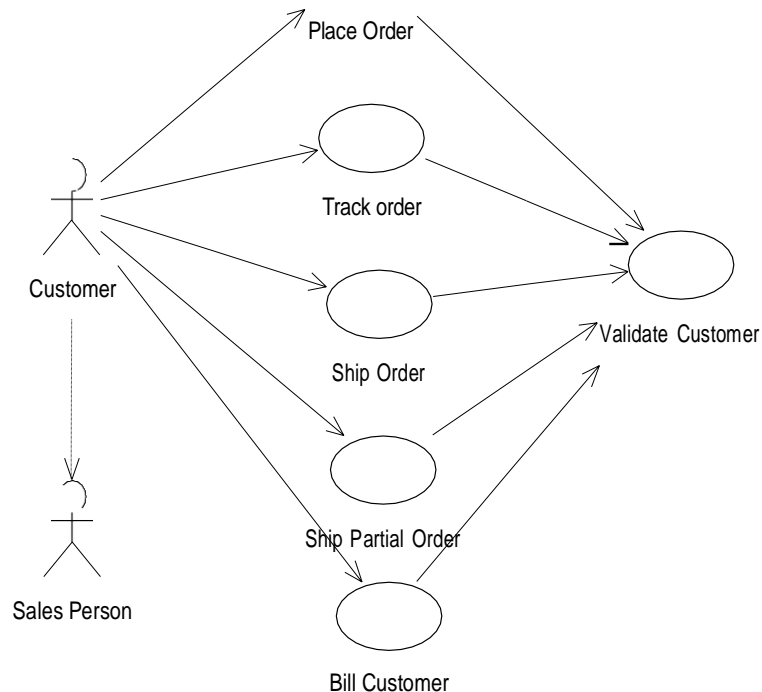


Fig.3. USE CASE DIAGRAM

(IV) ACTIVITY DIAGRAM

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

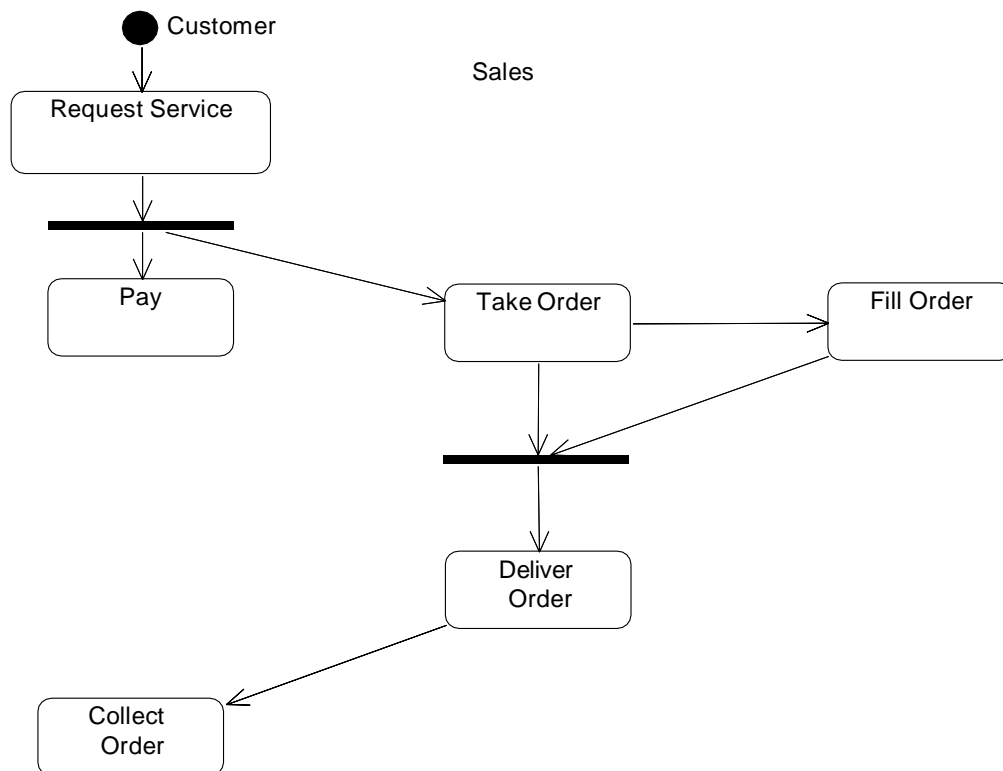


Fig.4. ACTIVITY DIAGRAM

(V) CLASS DIAGRAM

Description:

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

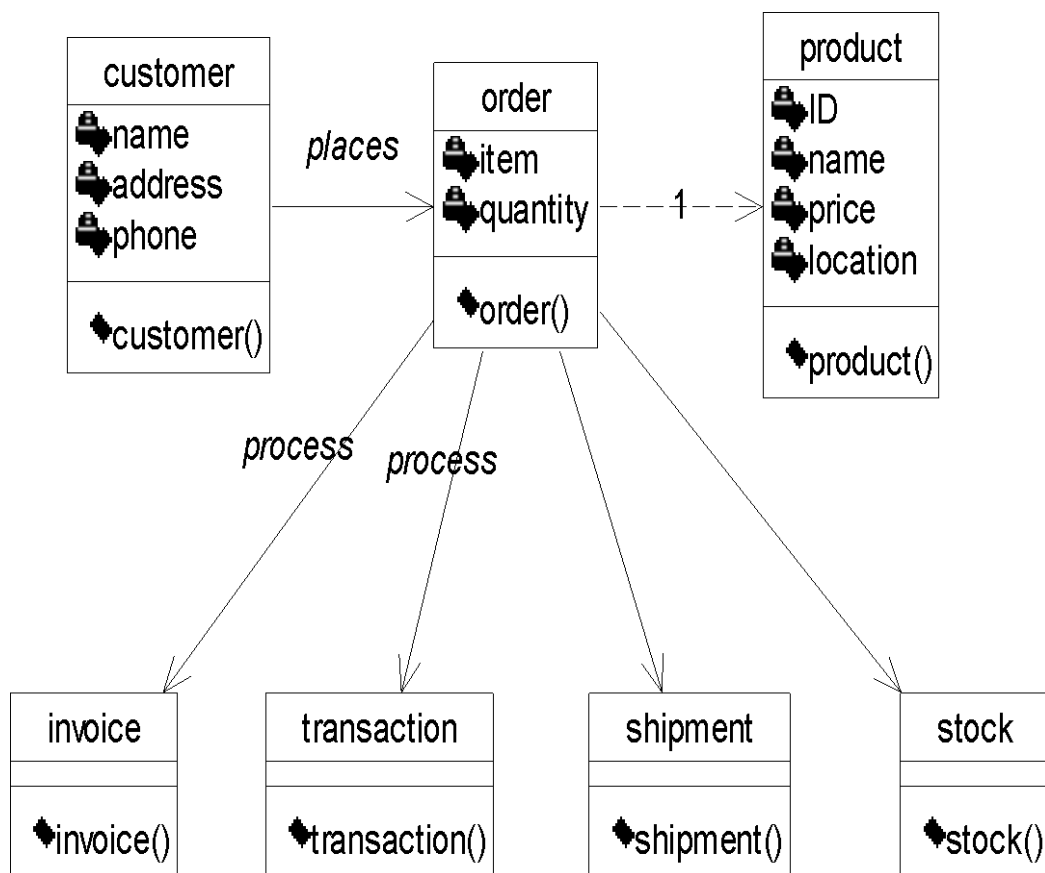


Fig.5. CLASS DIAGRAM

(VI) UML INTERACTION DIAGRAMS

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

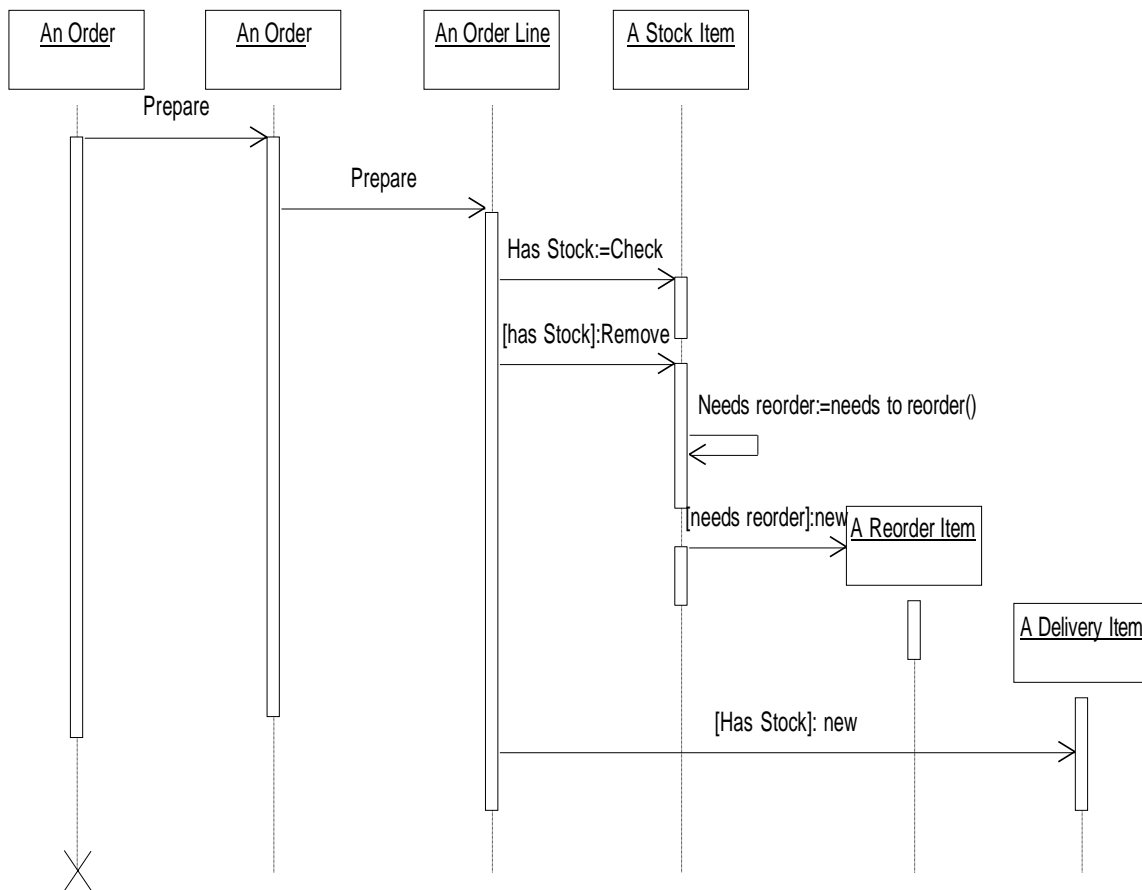


Fig.6.1 SEQUENCE DIAGRAM

COLLABORATION DIAGRAM

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

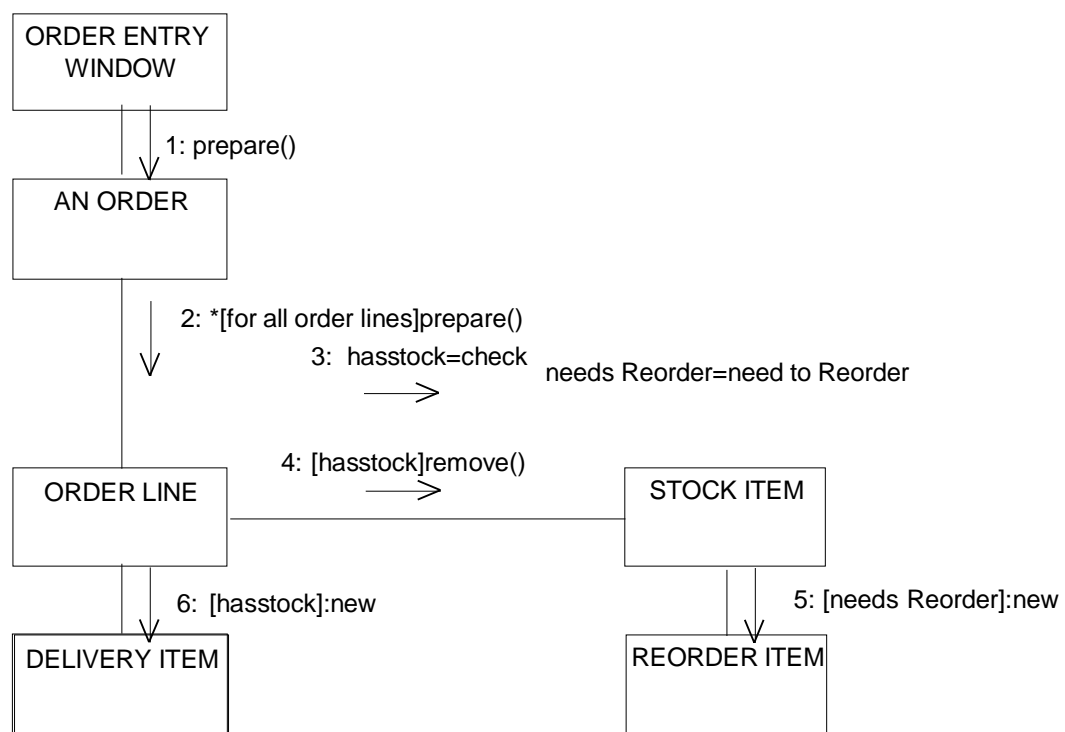
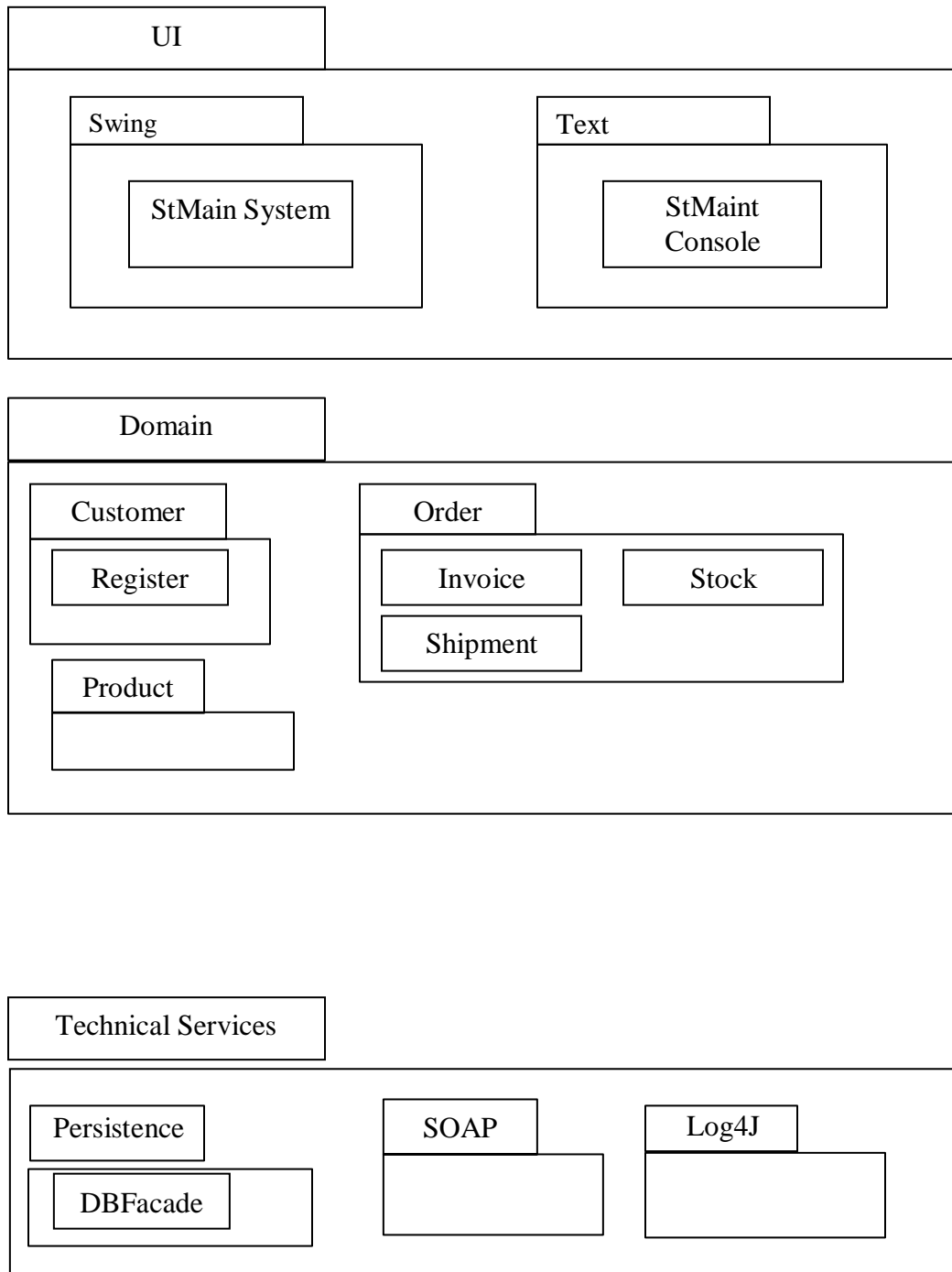


Fig.6.2 COLLABORATION DIAGRAM

(VII) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

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

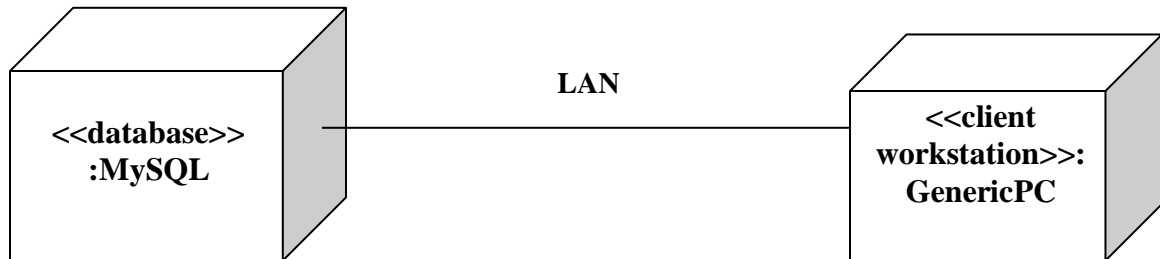


Fig.8.1.DEPLOYMENT DIAGRAM

Component Diagram

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

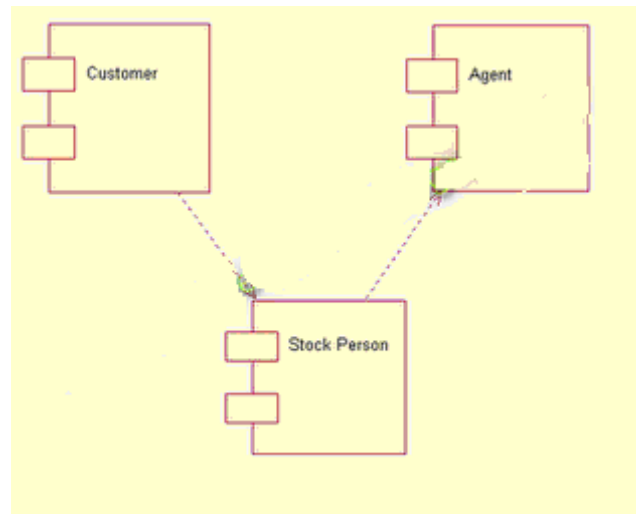


Fig.8.2.COMPONENT DIAGRAM

RESULT:

EX NO: 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 REQUIREMENT 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

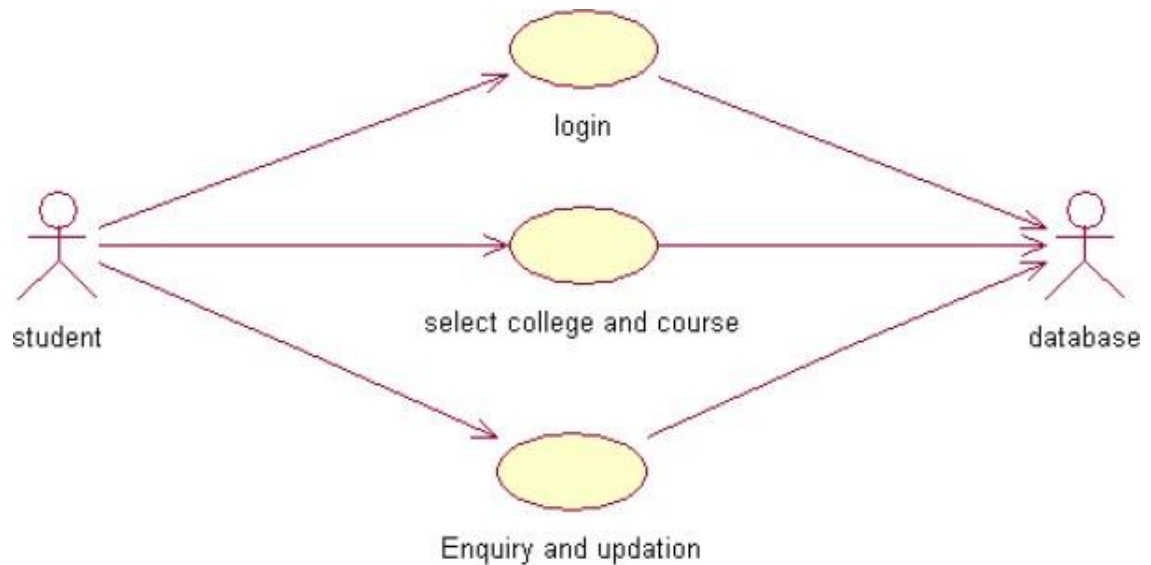
It is the capability about which it can performed function for many user at sometimes efficiently (ie) without any ever occurrences

RELIABILITY

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

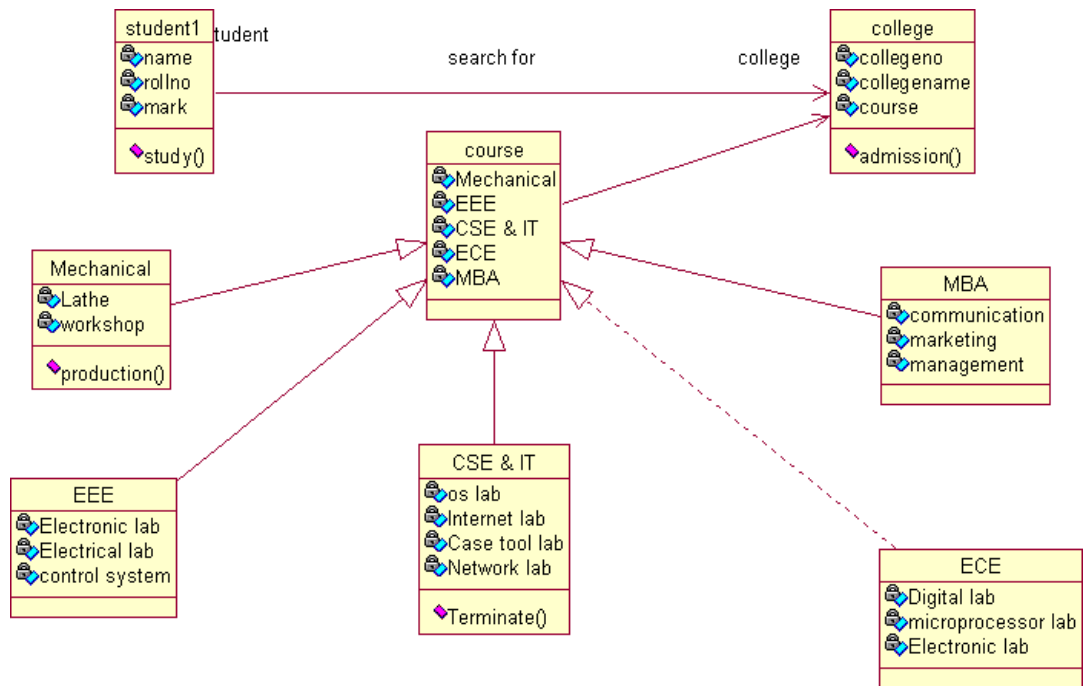
(III) USERCASE DIAGRAM

- 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
- Use case is a set of scenarios together by a common user goal
- A scenario is a sequence of step describing as interaction between a user and a system



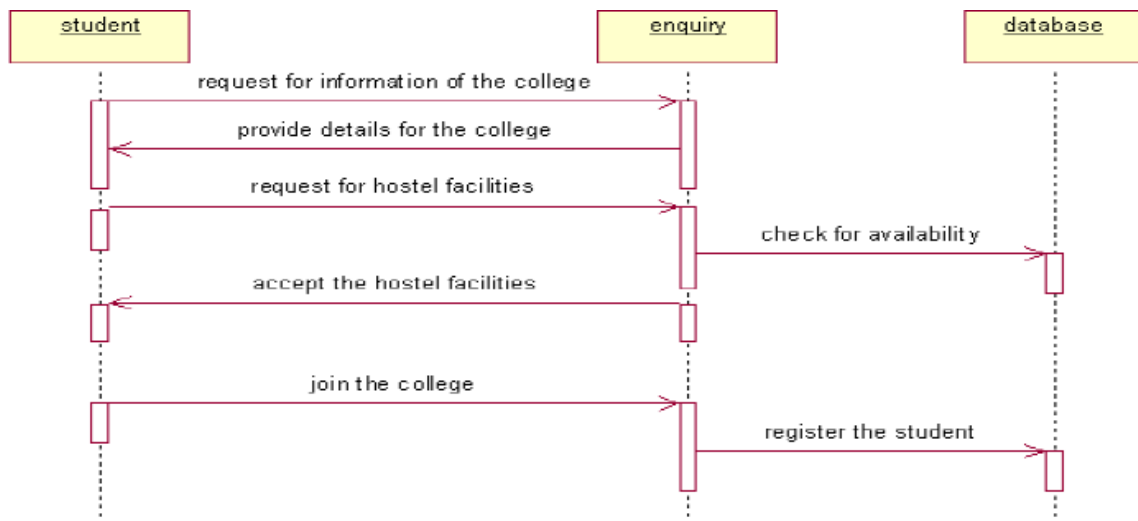
CLASS DIAGRAM:

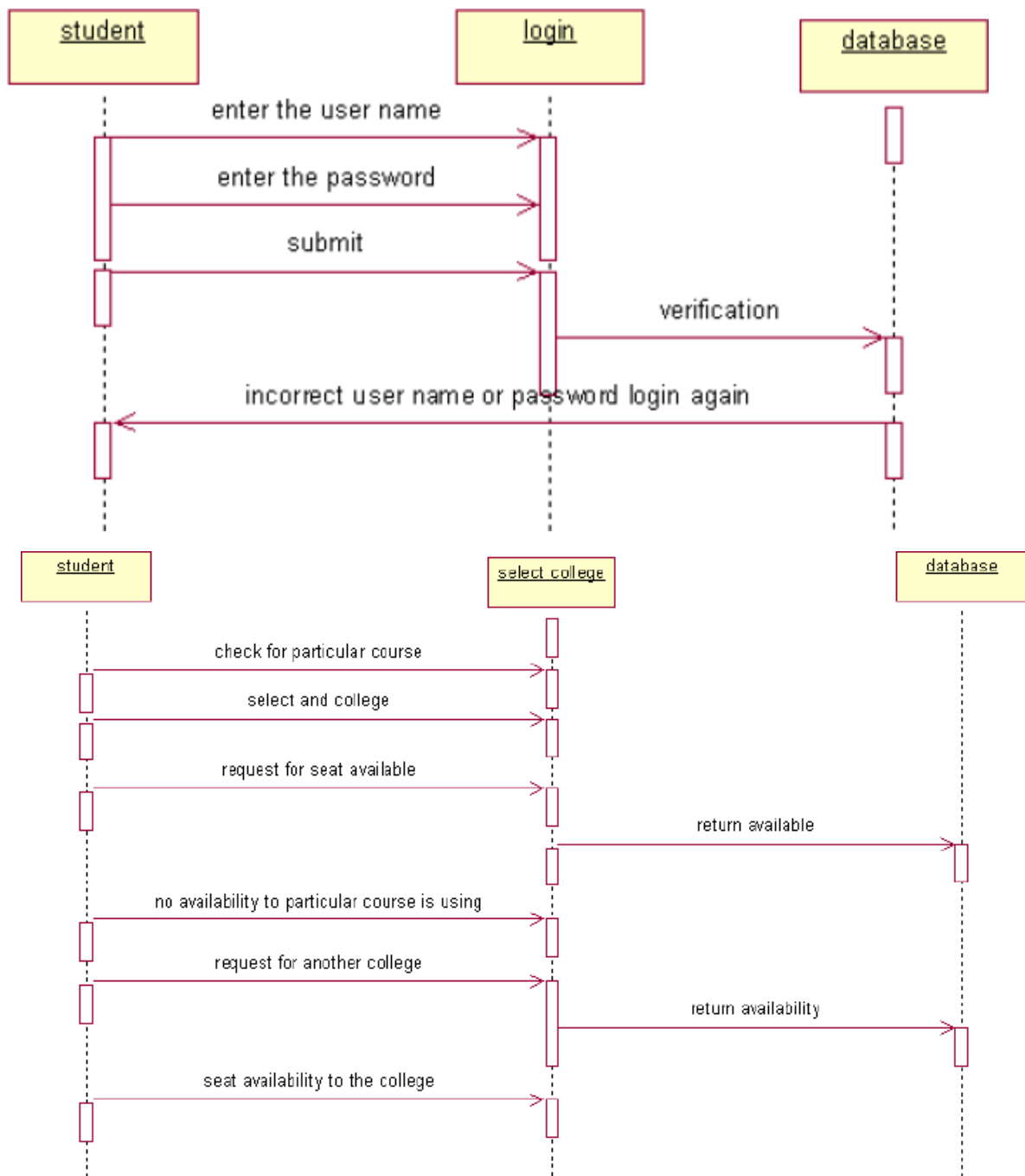
A class diagram describes the type of objects in the system the various kinds of static relationship that exist among them.



SEQUENCE DIAGRAM

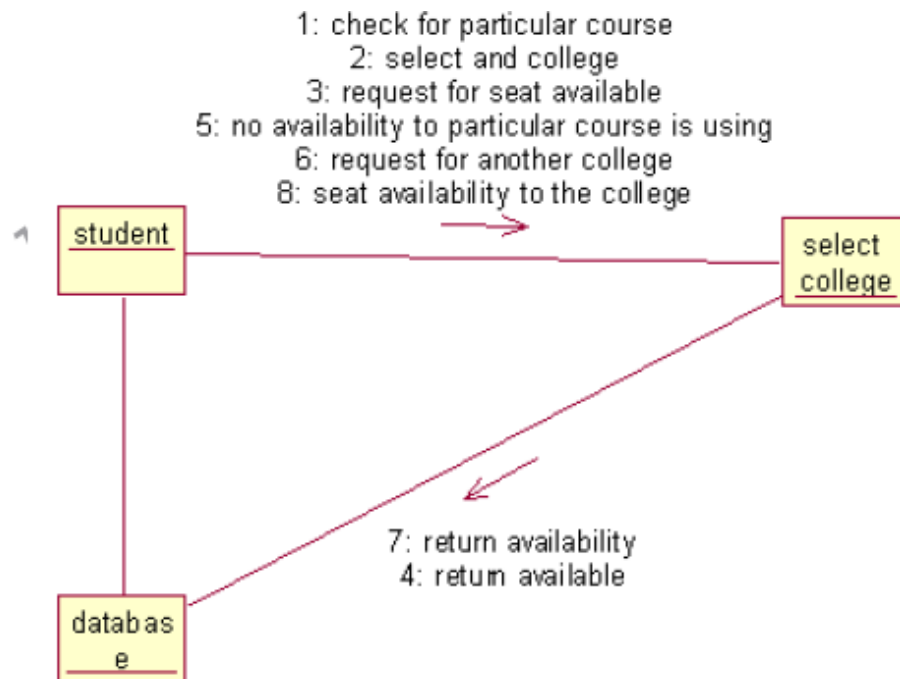
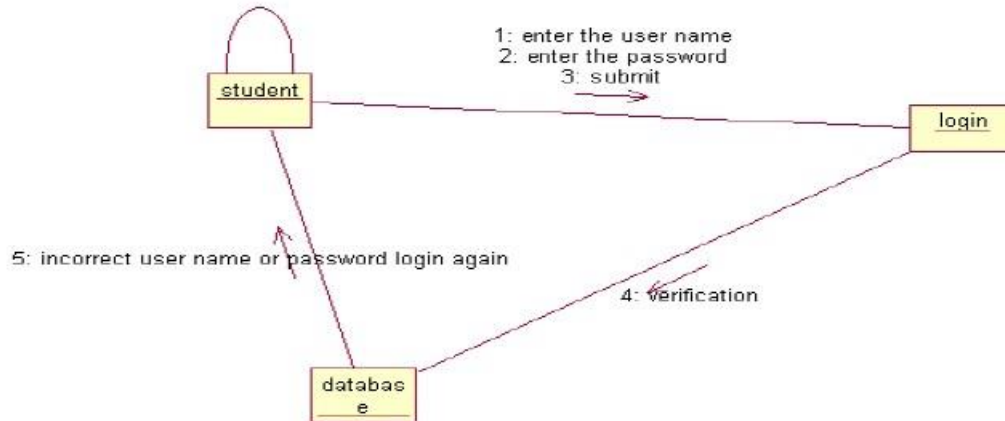
A sequence diagram is one that includes the object of the projects and tells the lifetimes and also various action performed between objects.





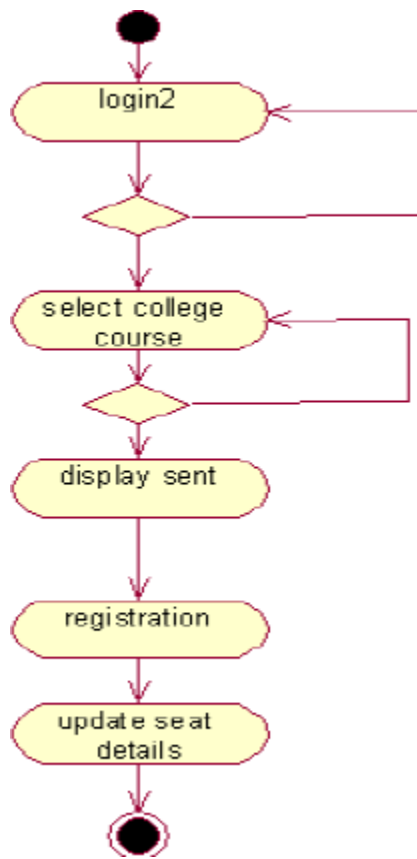
COLLABORATION DIAGRAM

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.



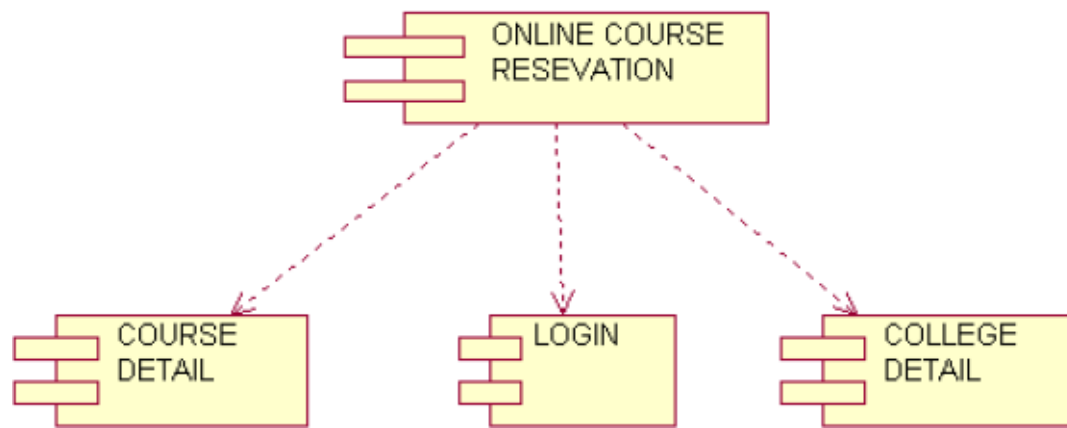
ACTIVIY DIAGRAM

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



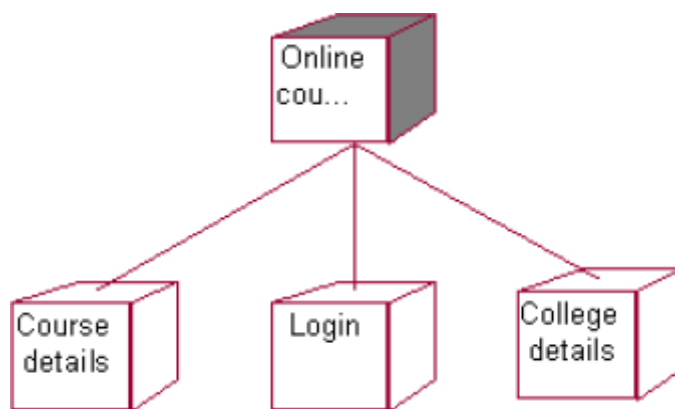
COMPONENT DIAGRAM

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-dimensional box. Dependencies are represented by communication association. The basic element of a deployment diagram is a node of two types



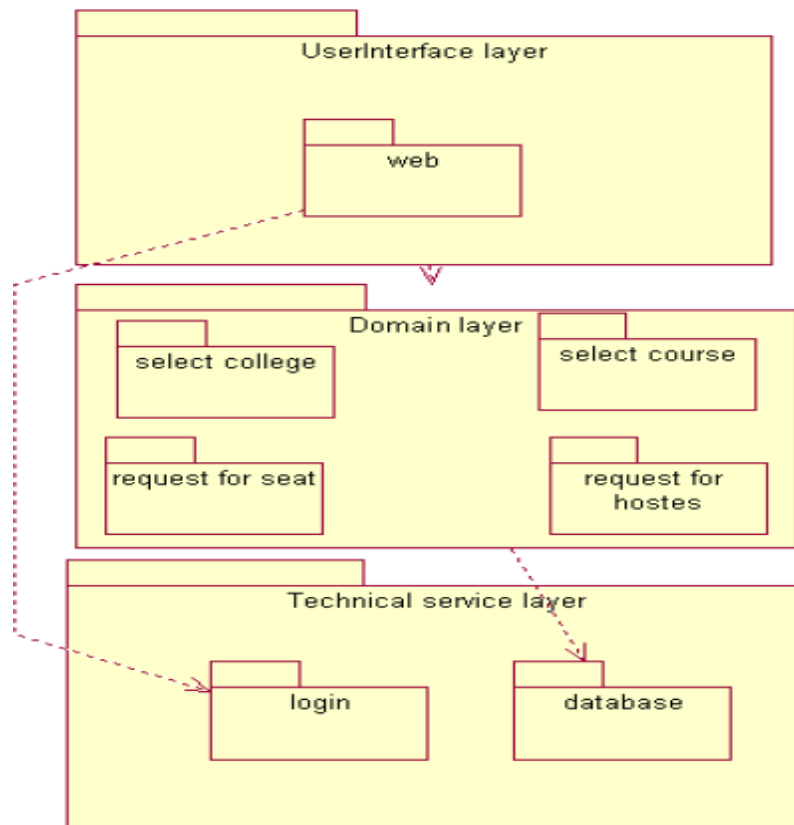
PACKAGE DIAGRAM

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 subordinate 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

EX NO: 7

AIRLINE/RAILWAY RESERVATION SYSTEM

Date:

AIM

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

(I) PROBLEM ANALYSIS AND PROJECT PLANNING

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

(II) OVERALL DESCRIPTION

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

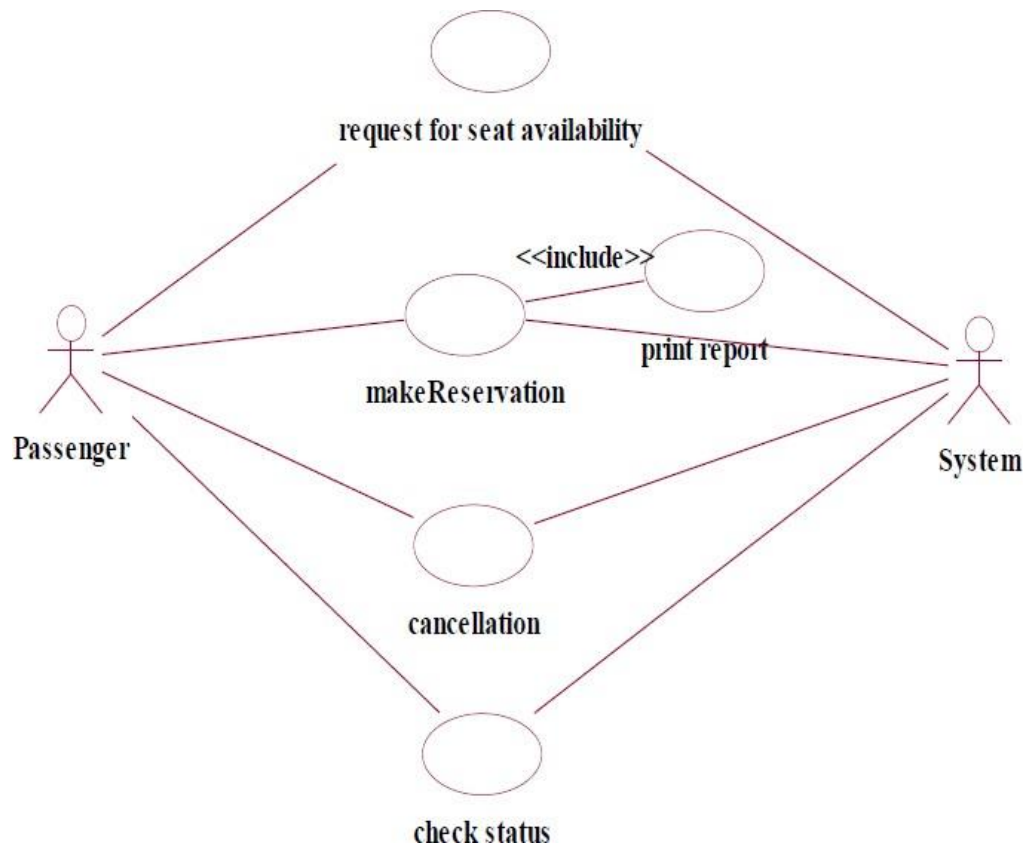
It is of the capacities about which it can perform function for many users at the same times efficiently that are without any error occurrence.

2.4 Reliability

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

(III) USE CASE DIAGRAM

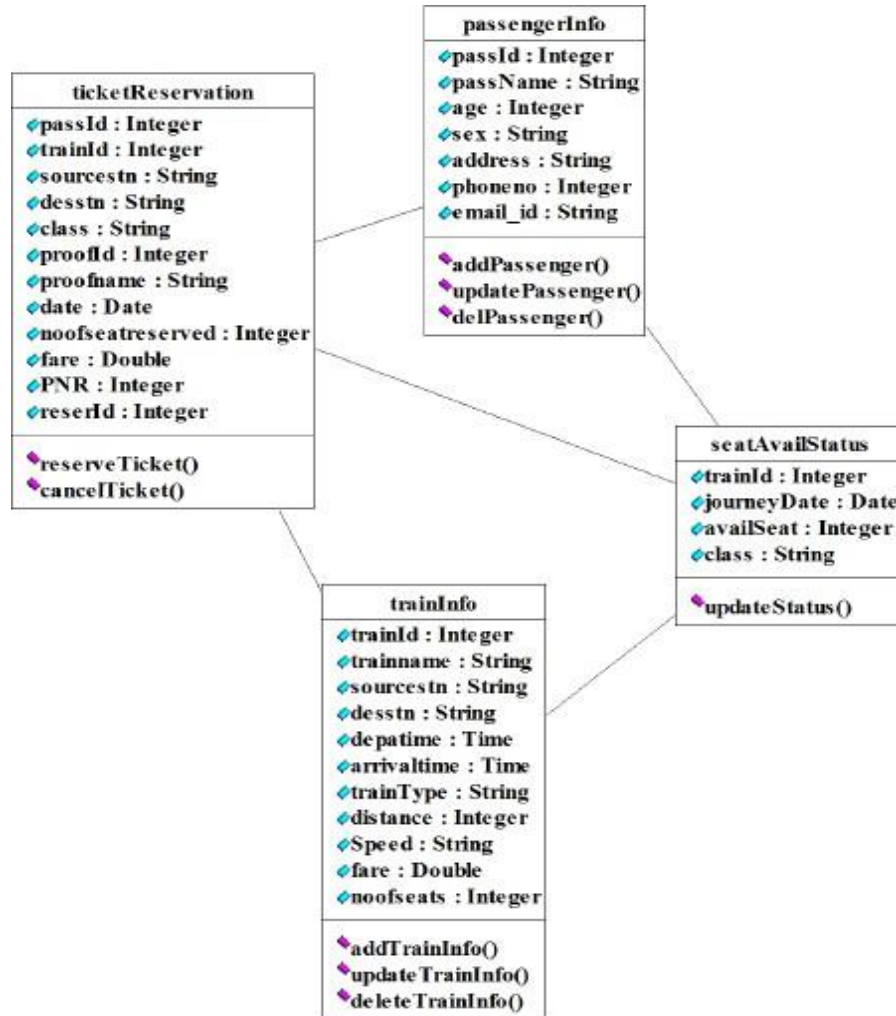
The passenger can view the status of the reserved tickets. So the passenger can confirm his/her travel.



**Fig. USE-CASE DIAGRAM FOR AIRLINE RESERVATION
(IV) CLASS DIAGRAM**

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

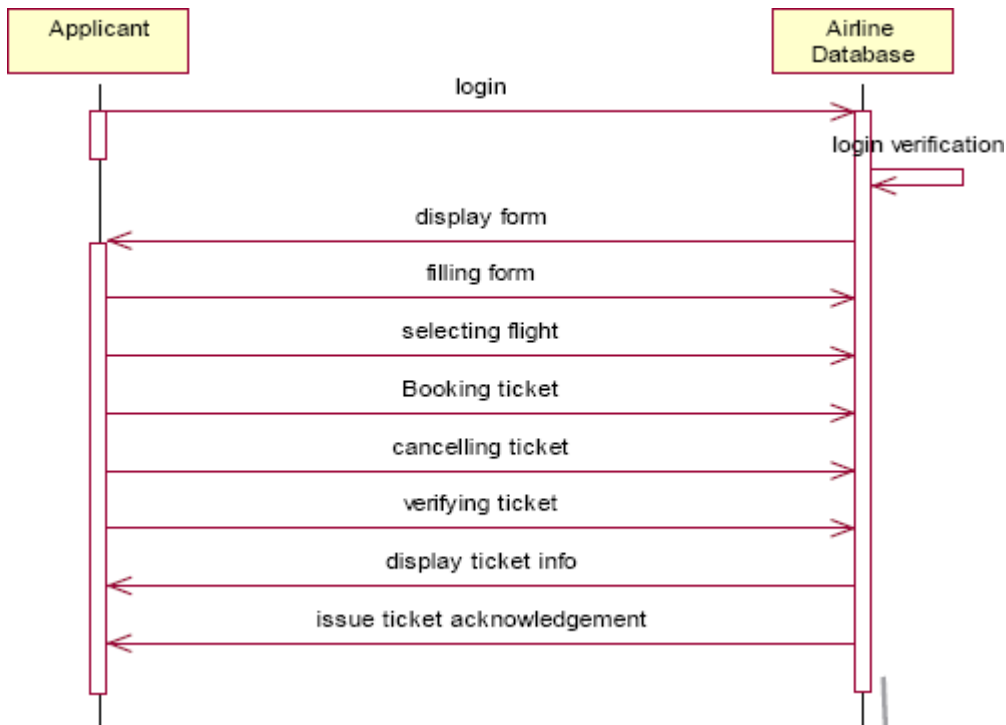
1. ticketReservation
2. trainInfo
3. passengerInfo
4. seatAvailStatus



SEQUENCE DIAGRAM

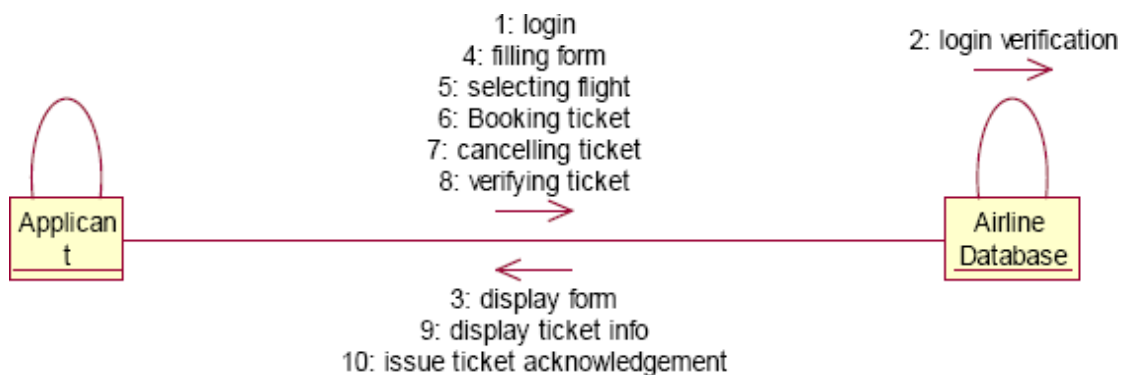
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

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



COLLABRATION DIAGRAM

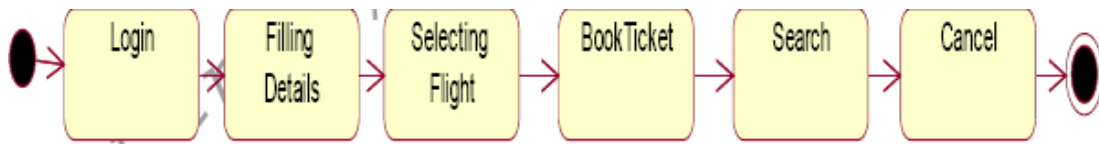
A collaboration diagram, also called a communication diagram or interaction diagram,. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice versa. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior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



STATE CHART DIAGRAM

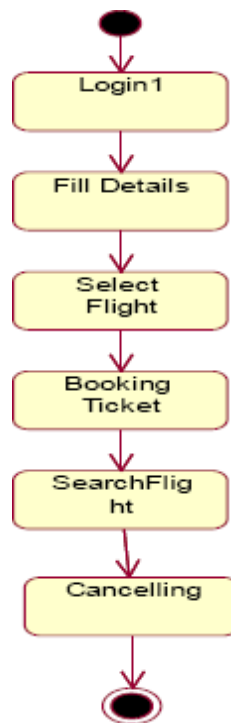
The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is

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



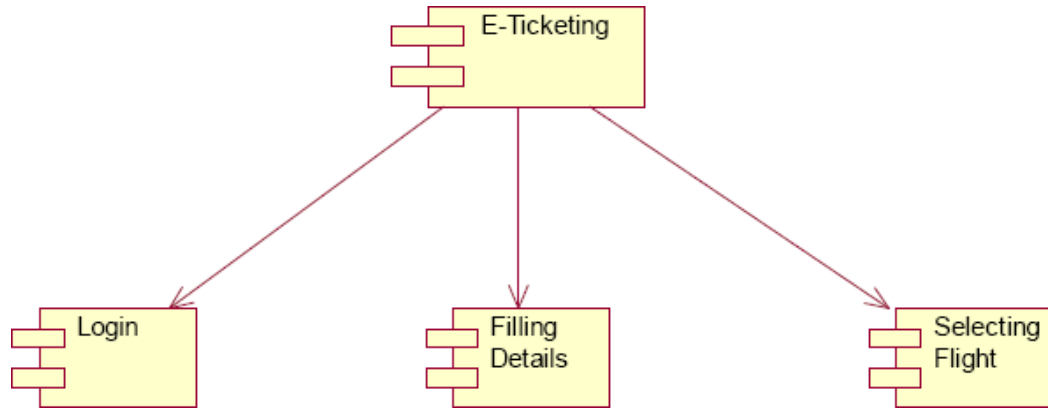
ACTIVITY DIAGRAM

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



COMPONENT DIAGRAM

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



RESULT

Ex no: 8 SOFTWARE PERSONNEL MANAGEMENT SYSTEM

Date:

AIM:

To implement a software for software personnel management system.

(I) PROBLEM STATEMENT:

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

(II) SOFTWARE REQUIREMENT SPECIFICATION:

2.1 SOFTWARE 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.

2.2 HARDWARE 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.

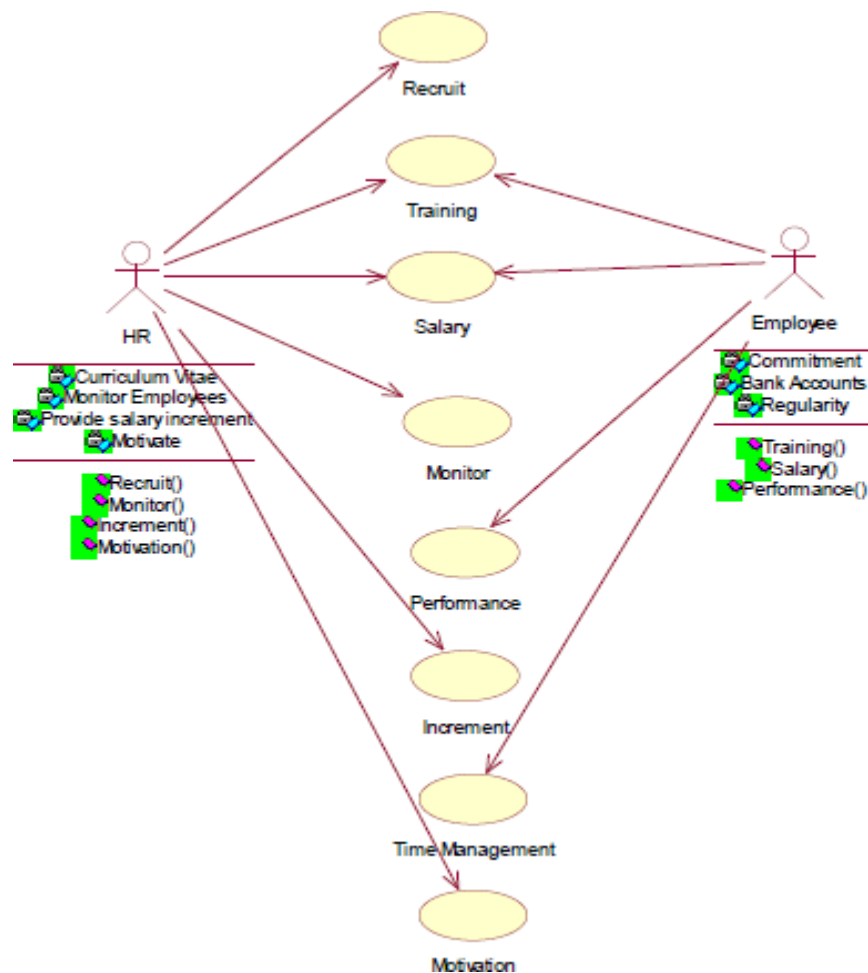


Fig.3. USE CASE DIAGRAM

(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.

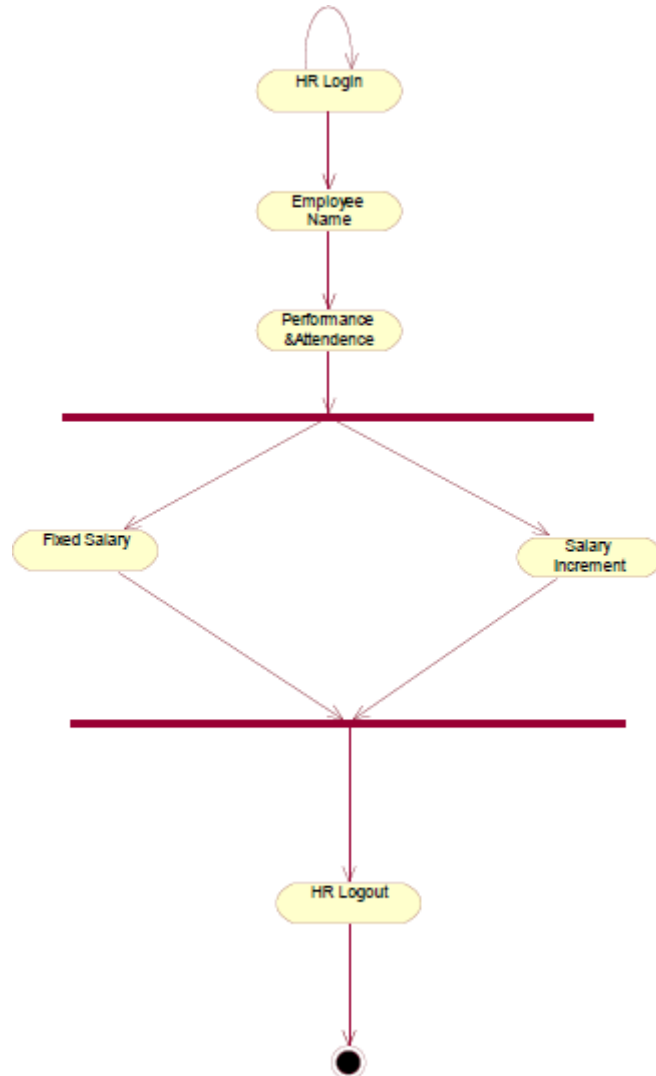


Fig.4. ACTIVITY DIAGRAM

(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.

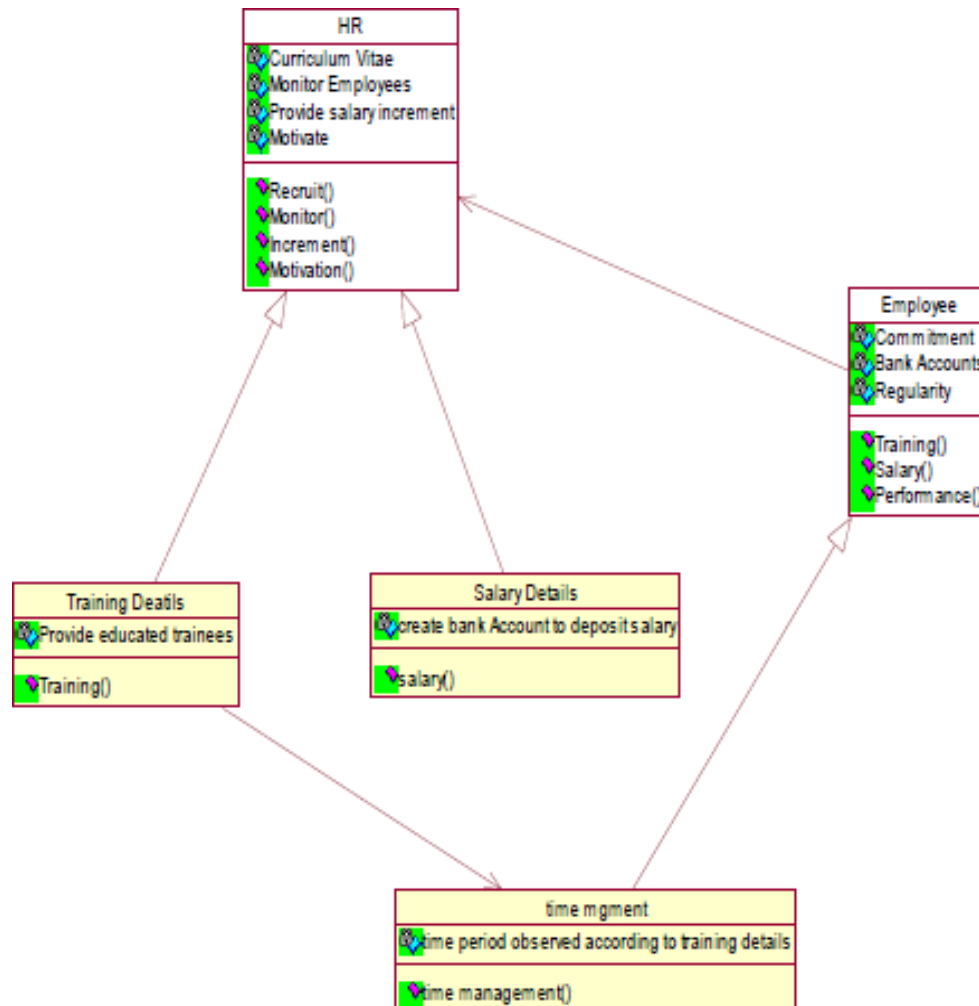


Fig.5.CLASS DIAGRAM

(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.

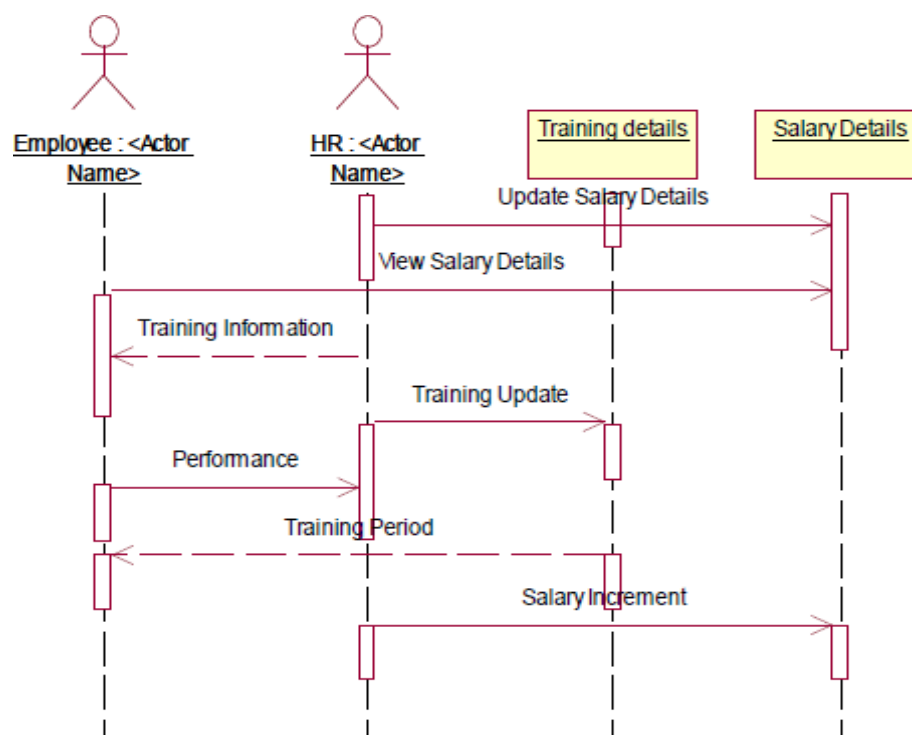


Fig.6.1.SEQUENCE DIAGRAM

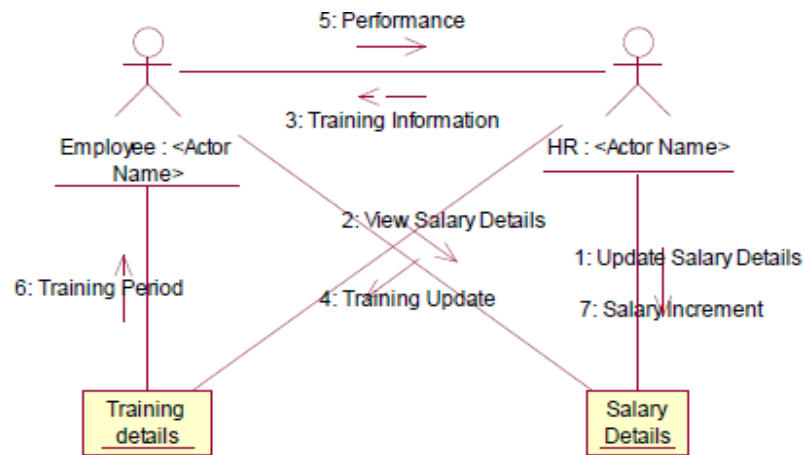


Fig.6.2.COLLABORATION DIAGRAM

(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.

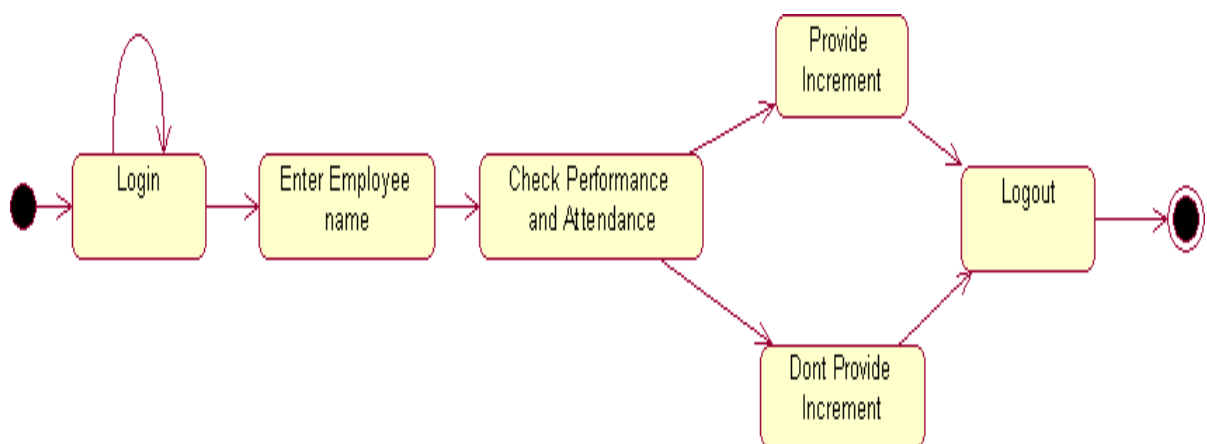


Fig.7.STATE TRANSITION DIAGRAM

(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.

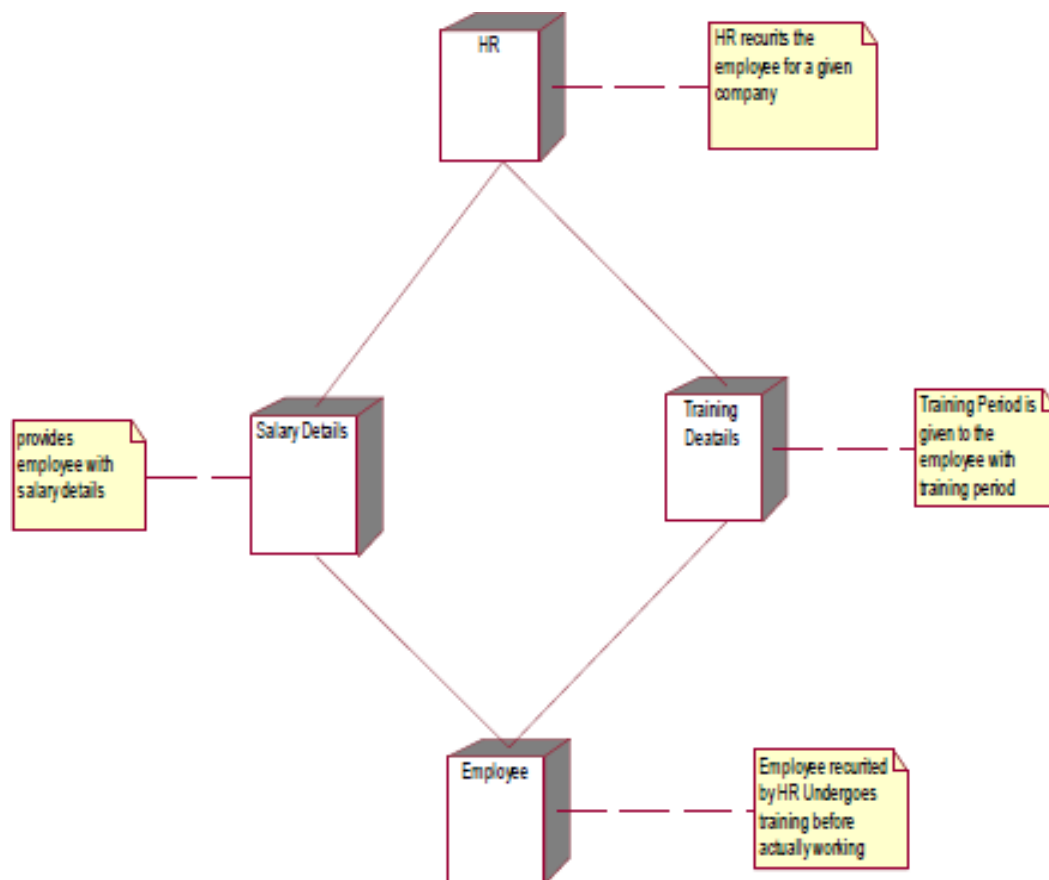


Fig.8.1.DEPLOYMENT DIAGRAM

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

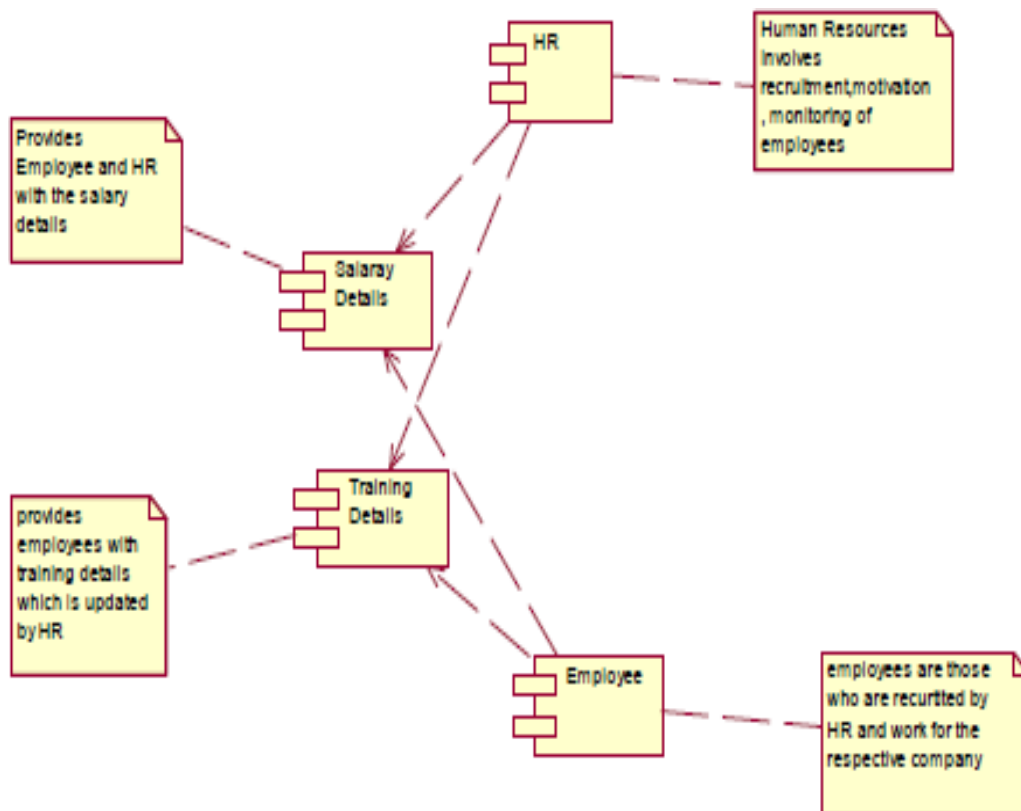


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Ex. No:9

CREDIT CARD PROCESSING

Date:

AIM:

To create a system to perform the credit card processing

(I) PROBLEM STATEMENT:

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

(II)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).

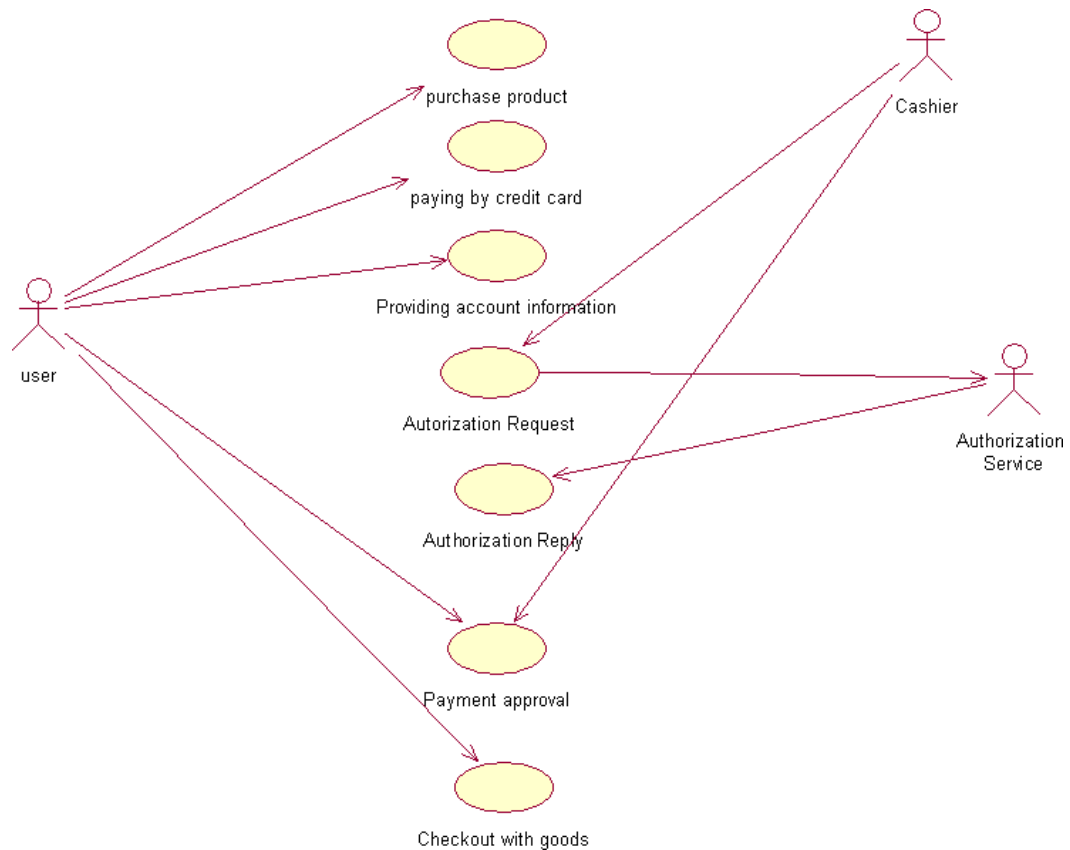


Fig.3. USECASE DIAGRAM FOR PASSPORT AUTOMATION SYSTEM

(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

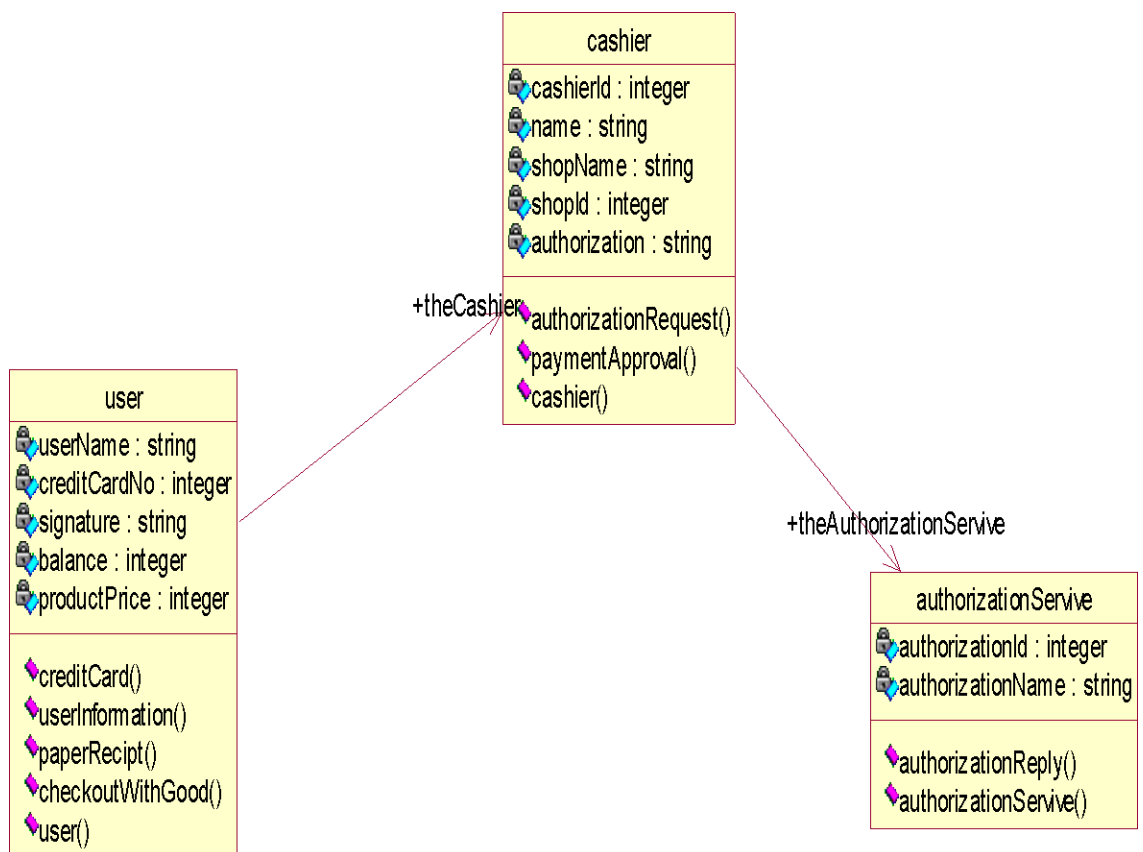


Fig.4.CLASS DIAGRAM

(V) INTERACTION DIAGRAM:

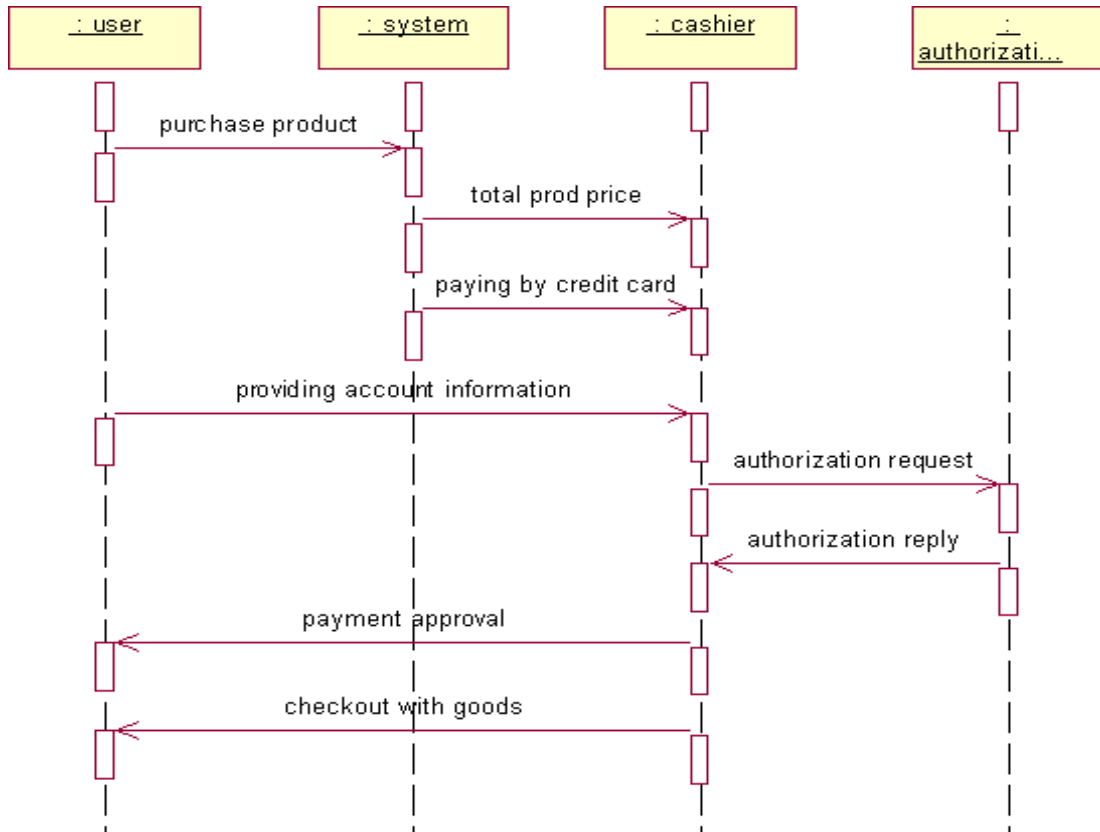


Fig.5.1.SEQUENCE DIAGRAM

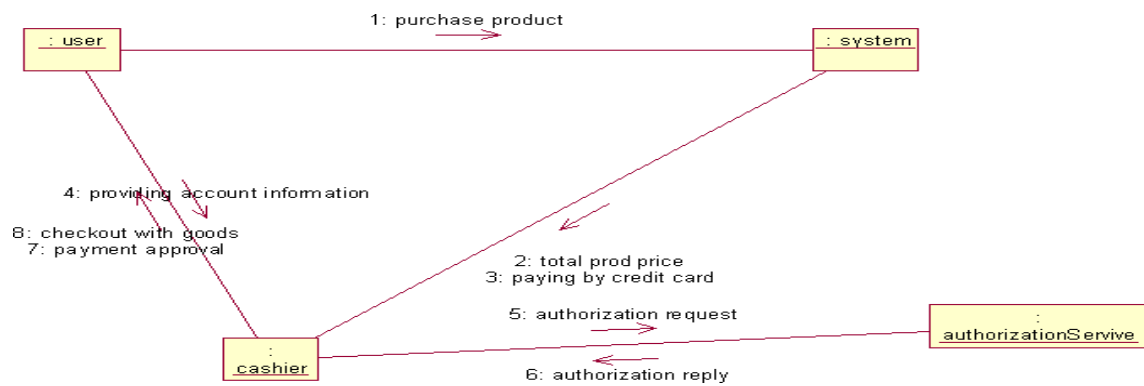
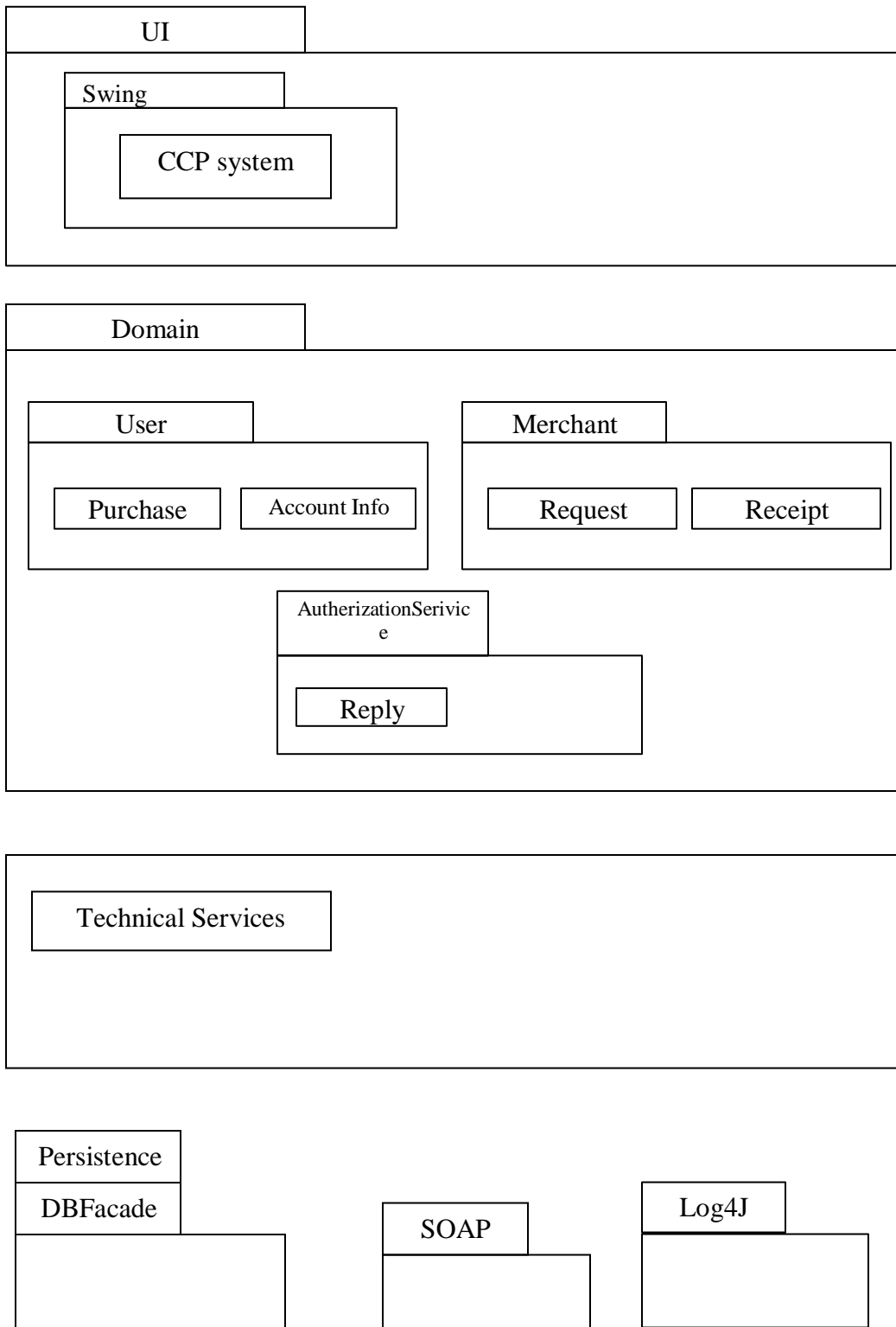


Fig.5.2.COLLABORATION DIAGRAM

(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM



(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.

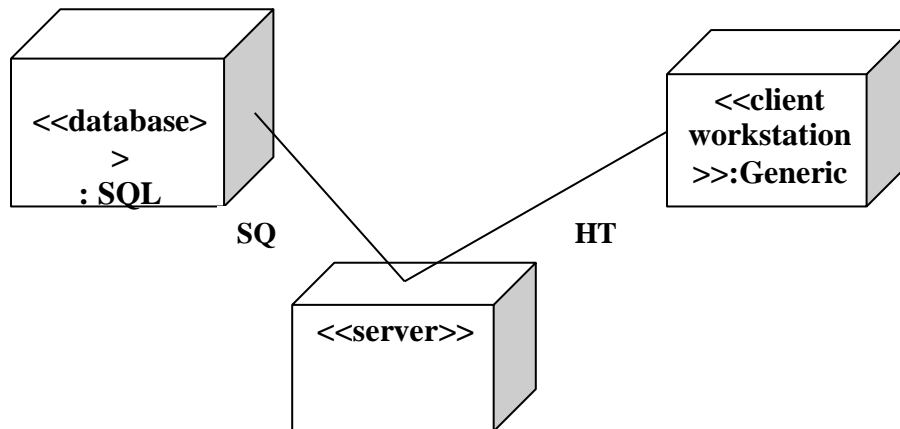


Fig.7.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

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

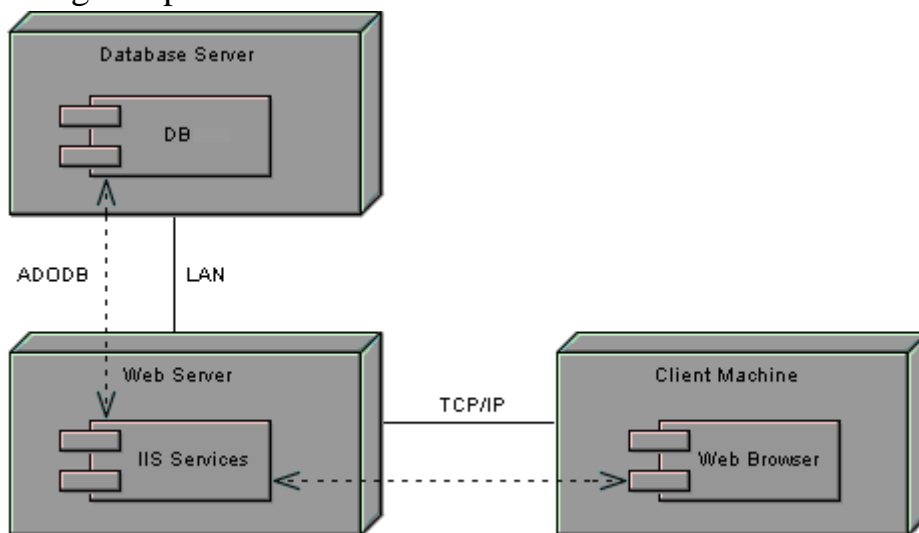


Fig.7.2.COMPONENT DIAGRAM

RESULT:

Ex. No:10

E-BOOK MANAGEMENT SYSTEM

Date:

AIM:

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

(I) PROBLEM STATEMENT:

An E- Book lends books and magazines to member, who is registered in the system. Also it handles the purchase of new titles for the Book Bank. Popular titles are brought into multiple copies. Old books and magazines are removed when they are out of 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 titles, members, loans and reservations from the system.

(II) SOFTWARE RESOURCE SPECIFICATION:

2.0 OVERALL DESCRIPTION

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

2.1 PRODUCT 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.

2.2 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

2.3 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:

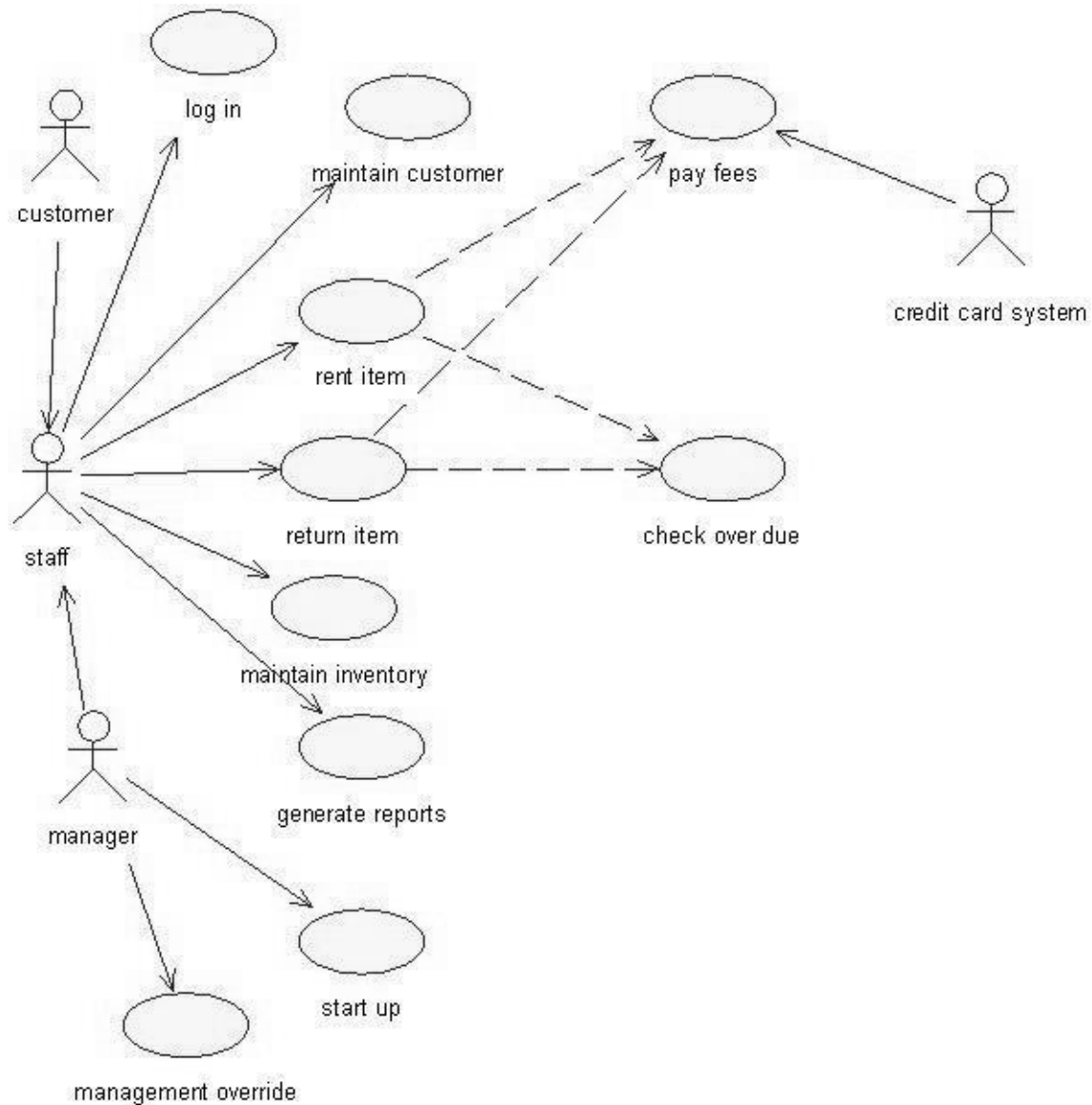


Fig.3.USE-CASE DIAGRAM FOR E-BOOK SYSTEM

(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.

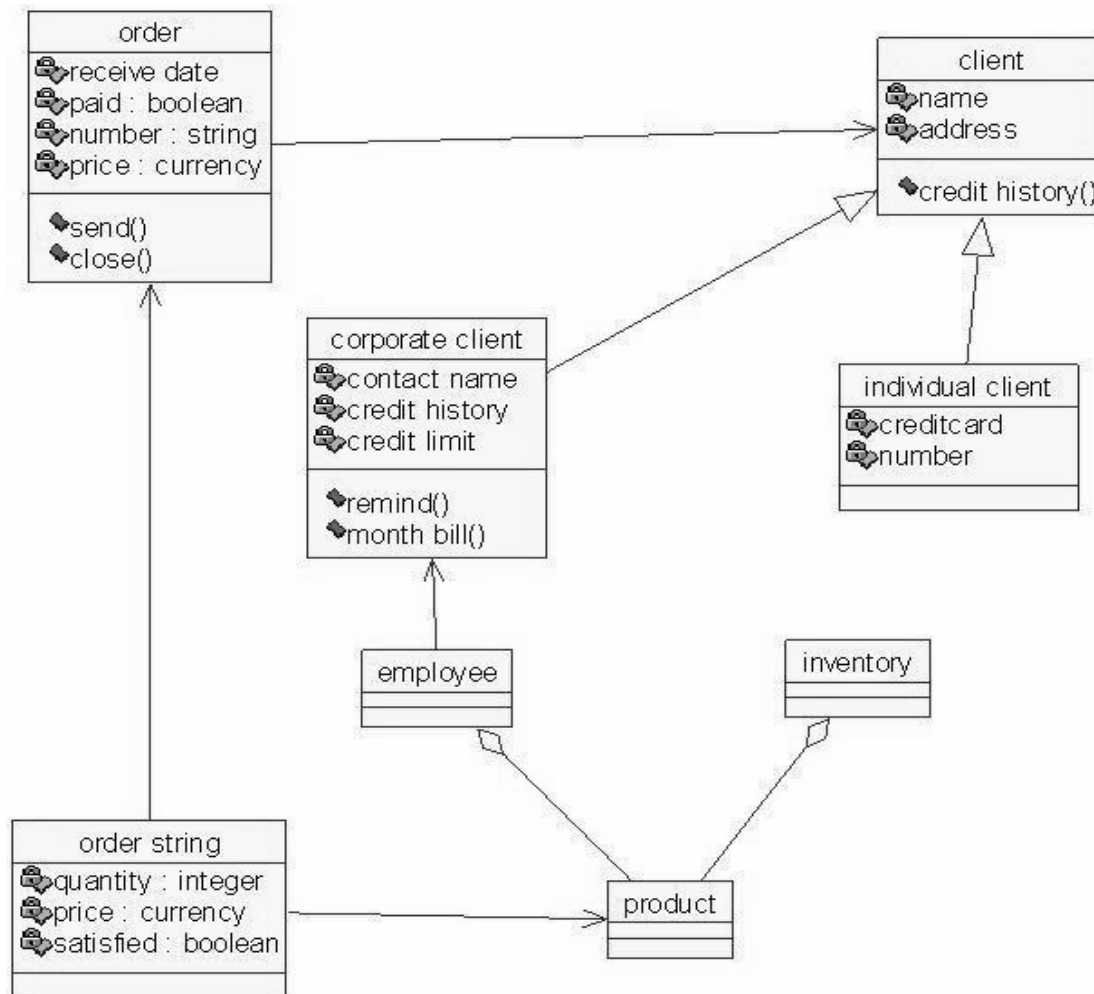


Fig.5.CLASS DIAGRAM FOR E-BOOK SYSTEM

(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 “from” 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.

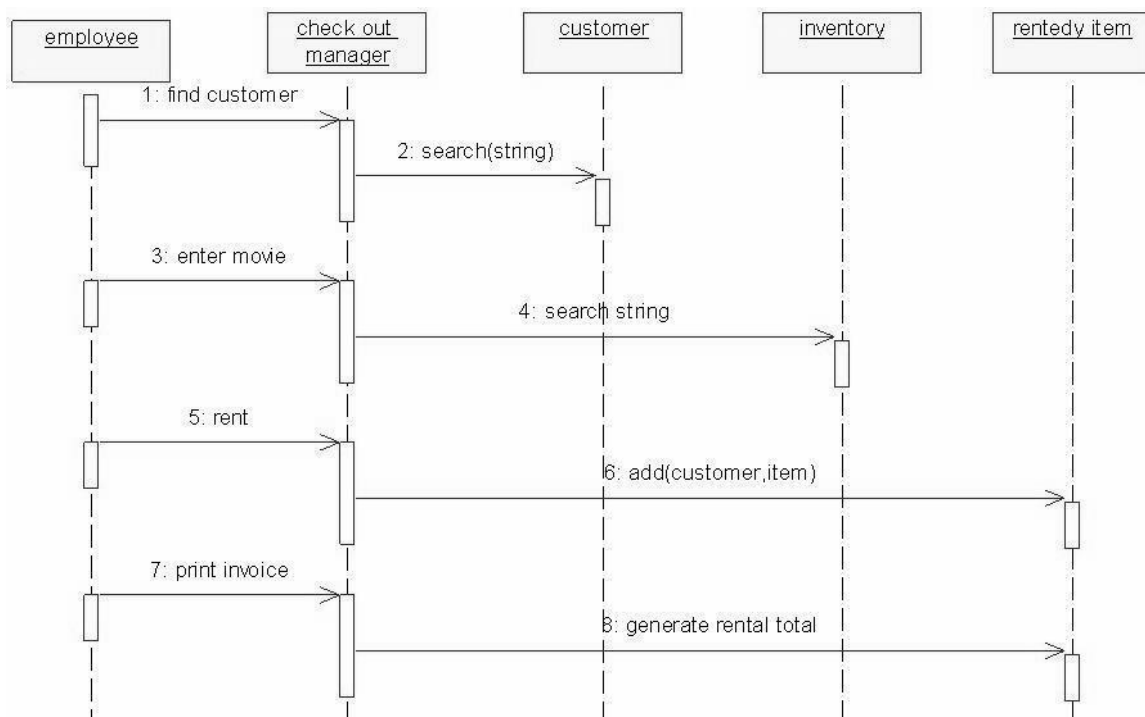


Fig.6.1.SEQUENCE DIAGRAM

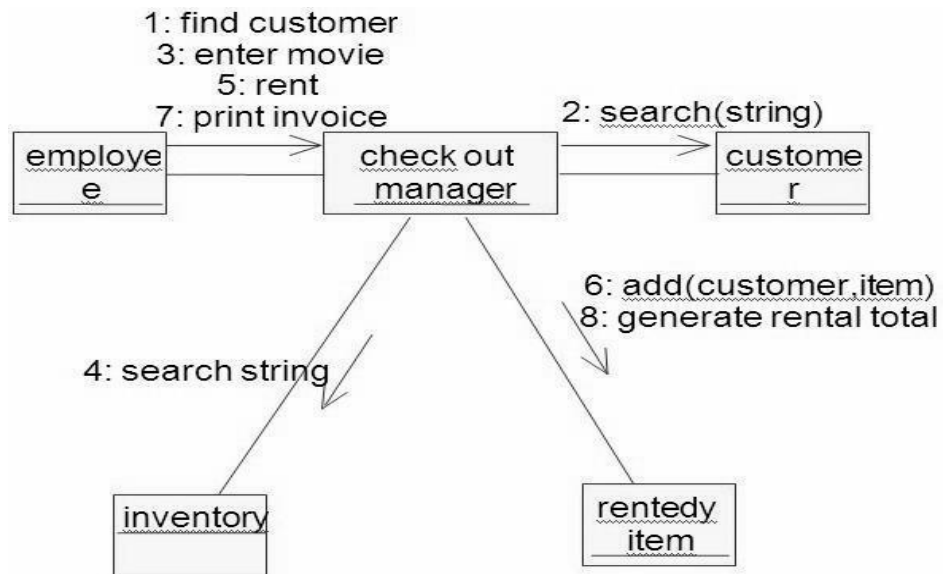
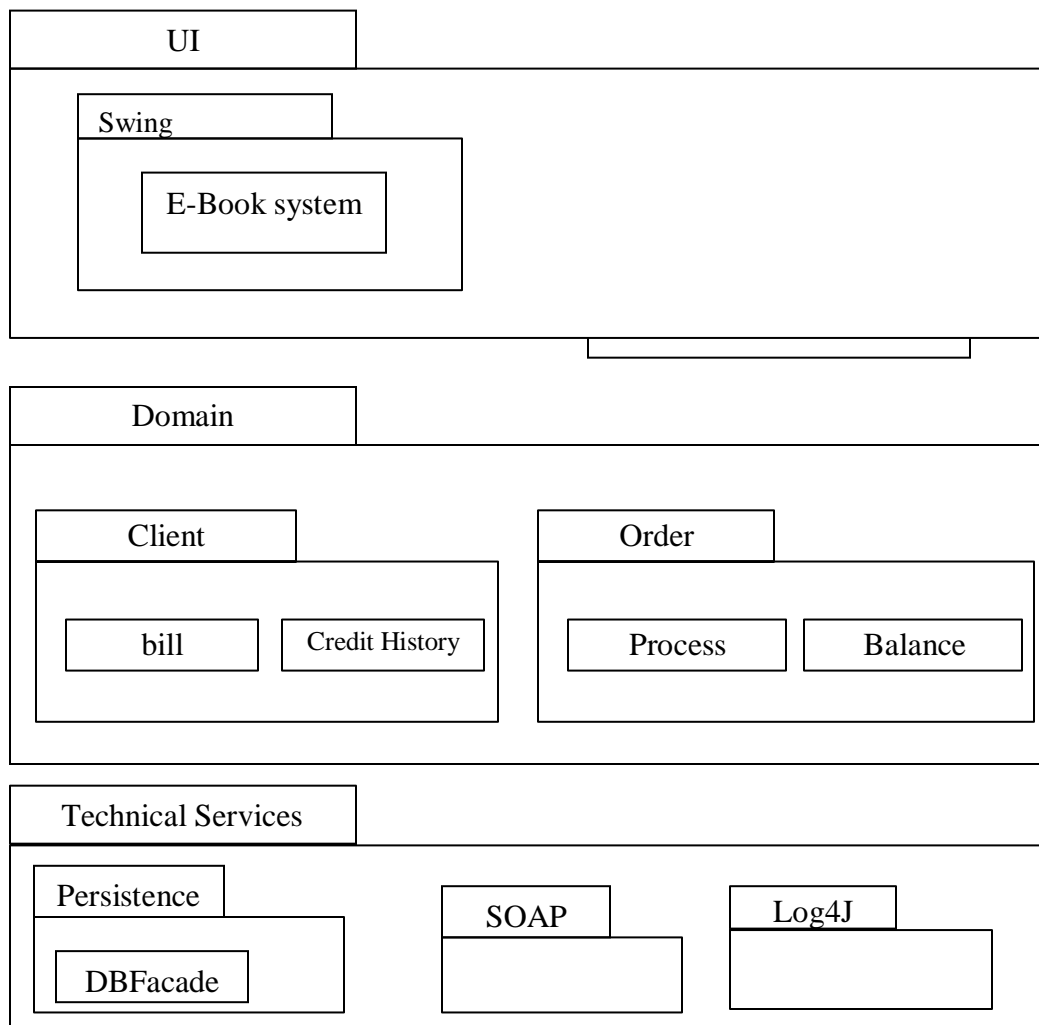


Fig.6.2.COLLABORATION DIAGRAM

(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.

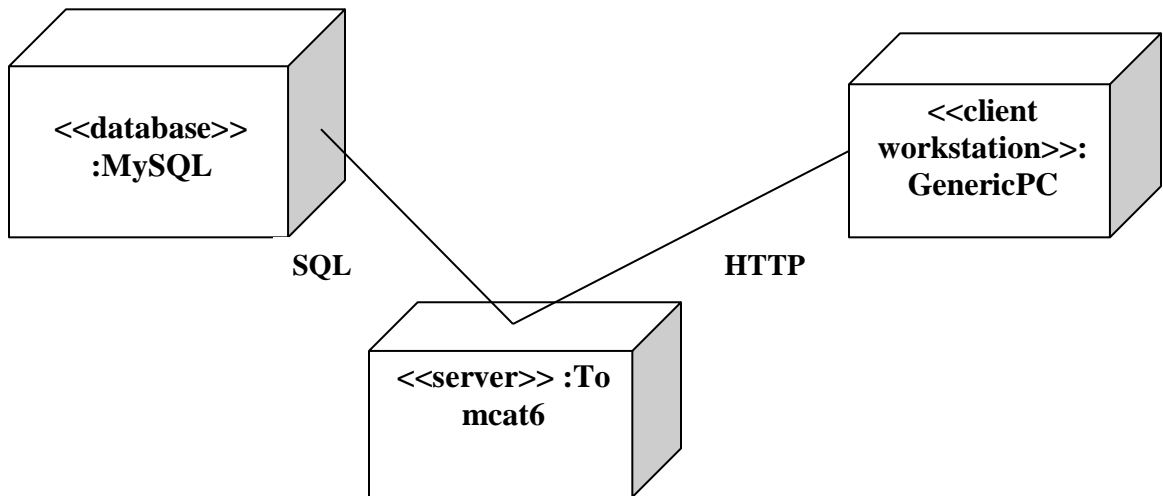


Fig.8.1.DEPLOYMENT DIAGRAM

COMPONENT DIAGRAM

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

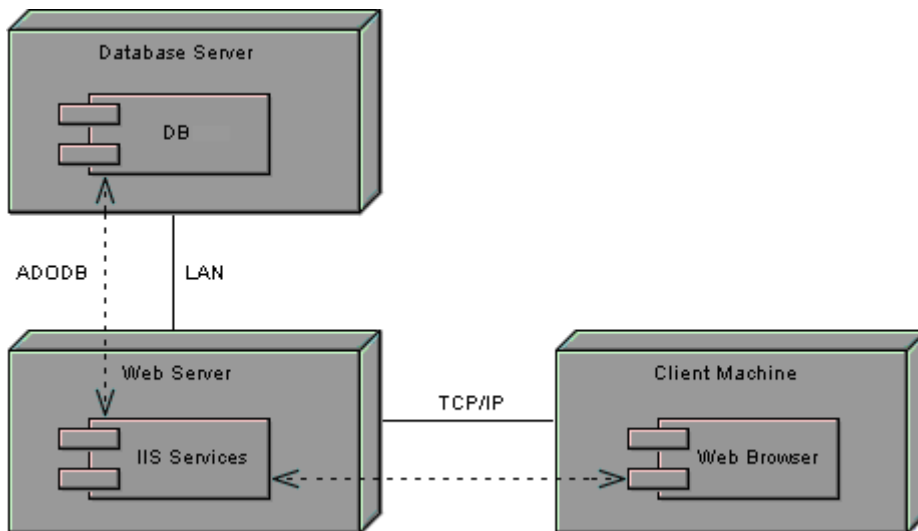


Fig.8.2.COMPONENT DIAGRAM

RESULT:

Ex.No: 11

RECRUITMENT SYSTEM

Date:

AIM:

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

(I) PROBLEM STATEMENT:

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

(II) 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.

2.2 SOFTWARE 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.

2.3 HARDWARE 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:

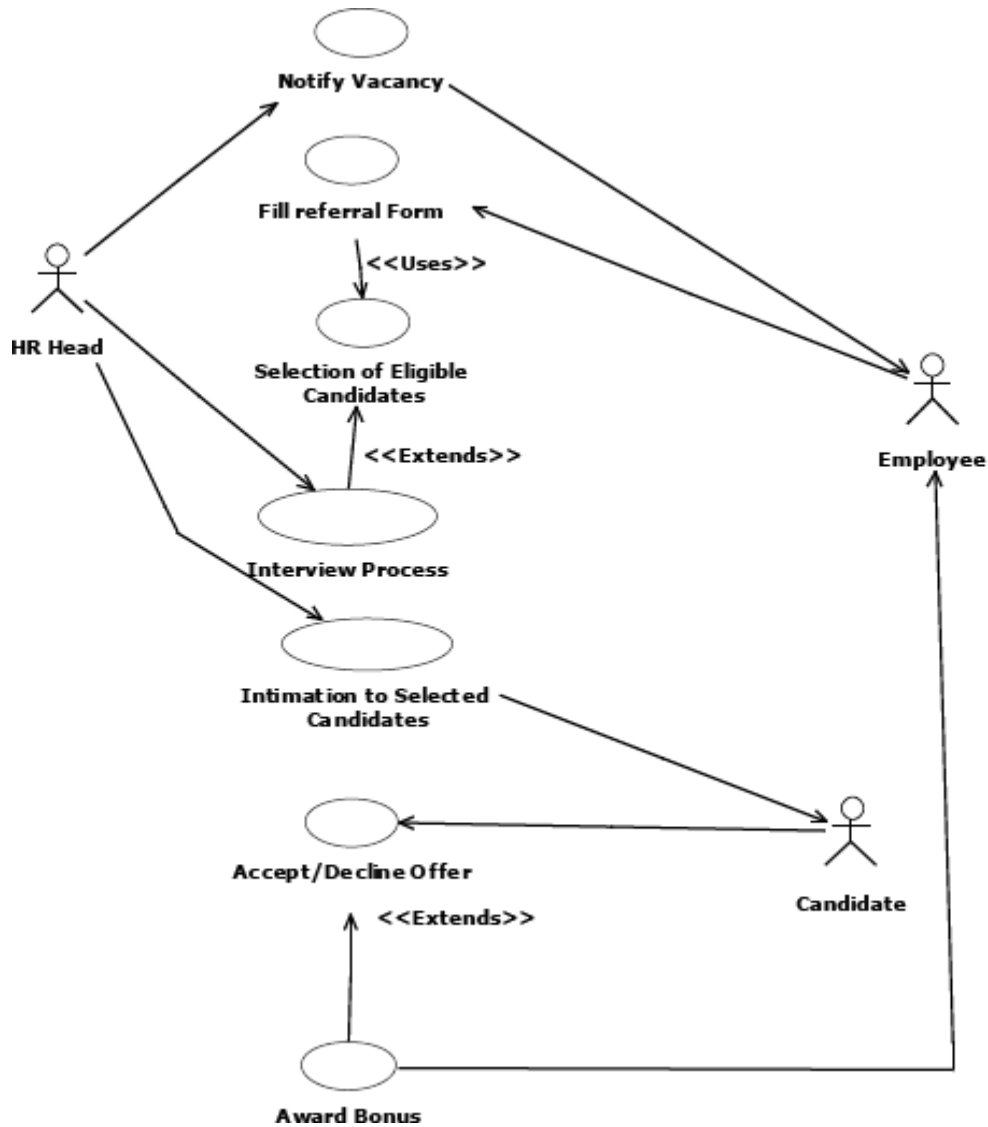


Fig.3. UML USE CASE DIAGRAM

(IV) ACTIVITY DIAGRAM:

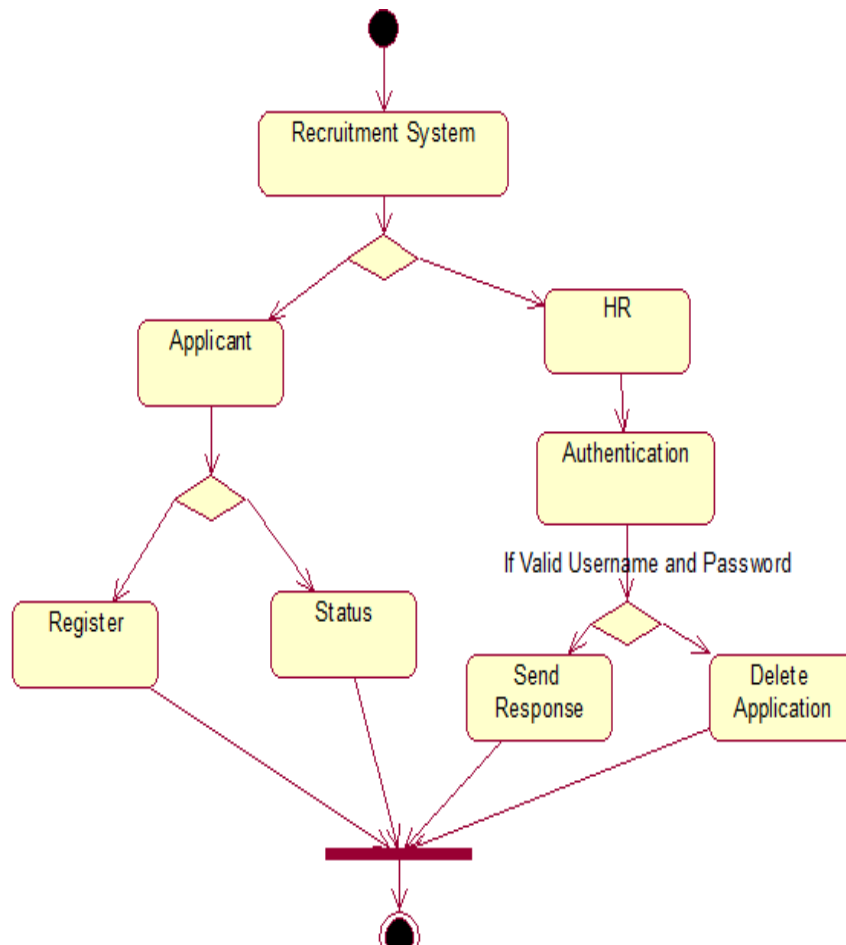


Fig.4. ACTIVITY DIAGRAM

(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.

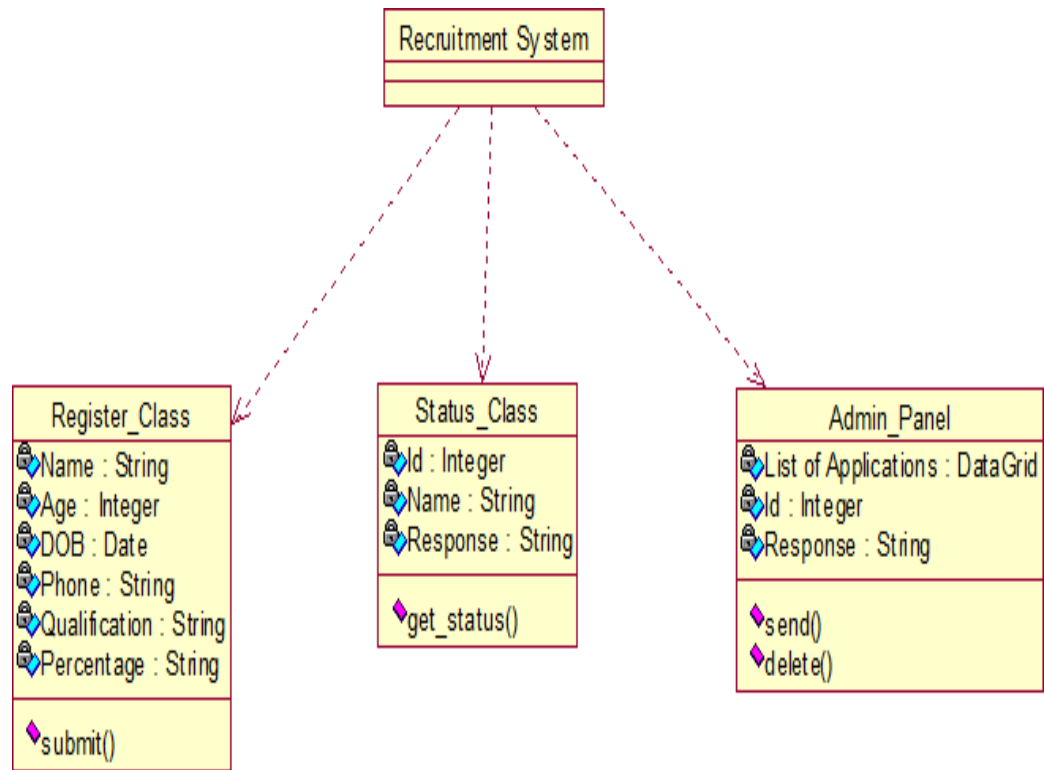


Fig.5. UML CLASS DIAGRAM

(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.

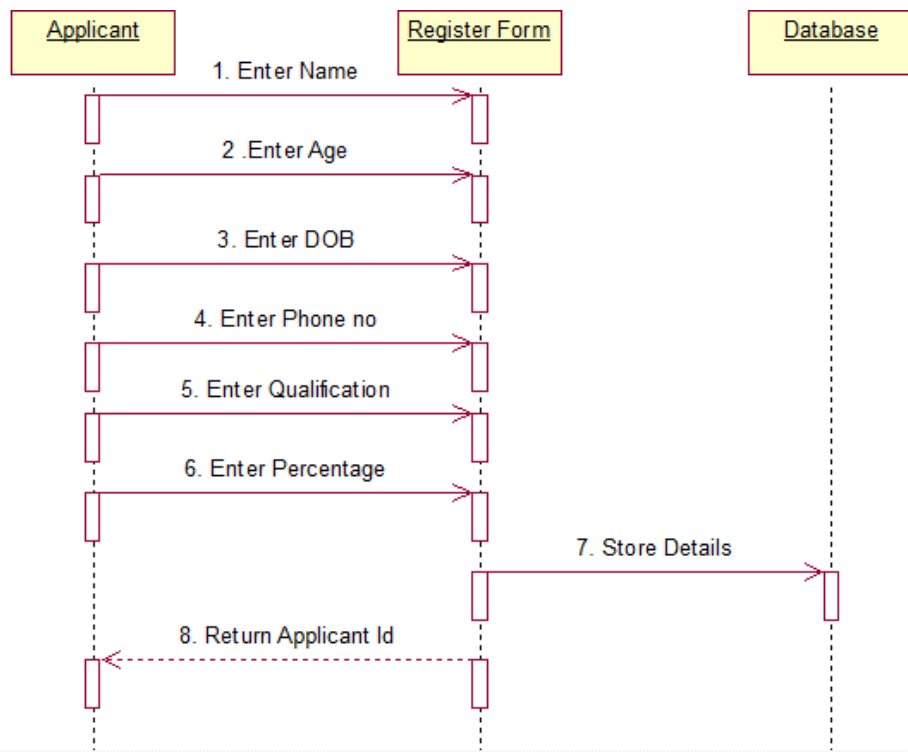


Fig. 6.1 SEQUENCE DIAGRAM FOR Register:

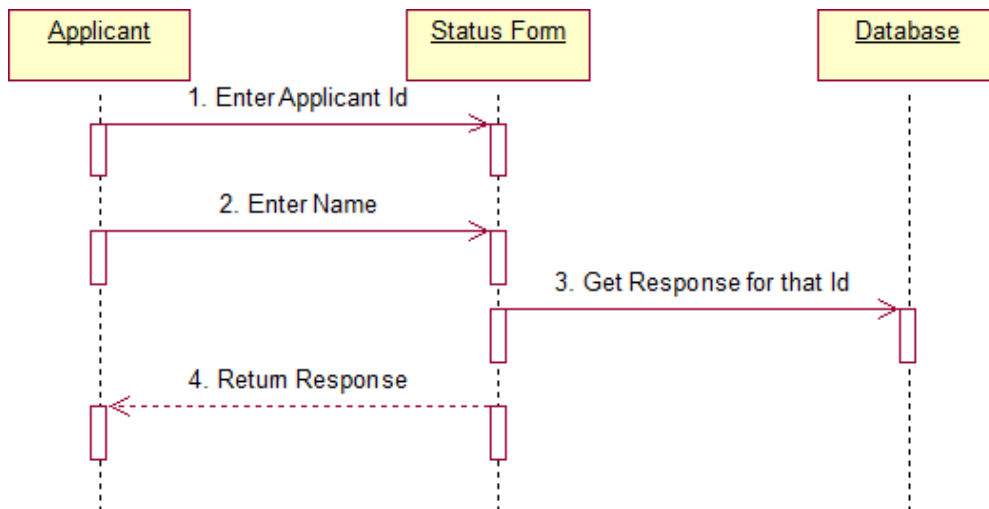


Fig.6.2. SEQUENCE DIAGRAM FOR STATUS

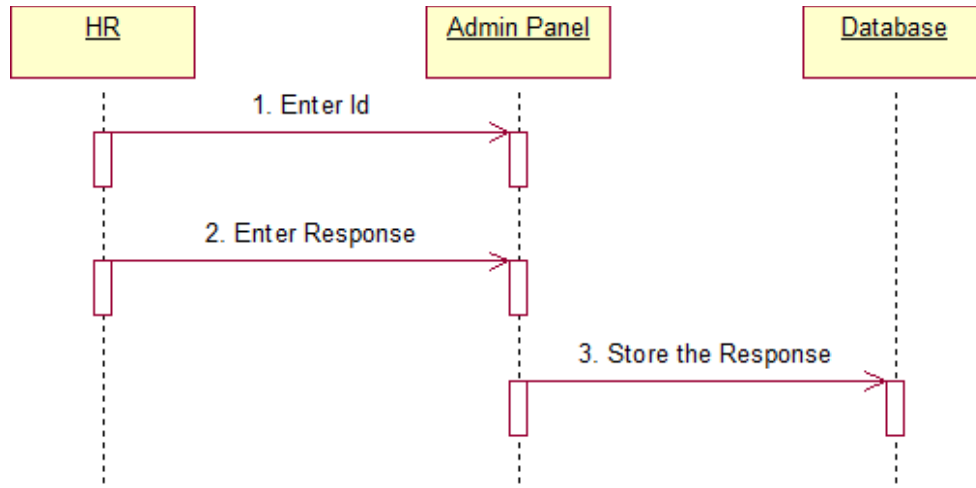
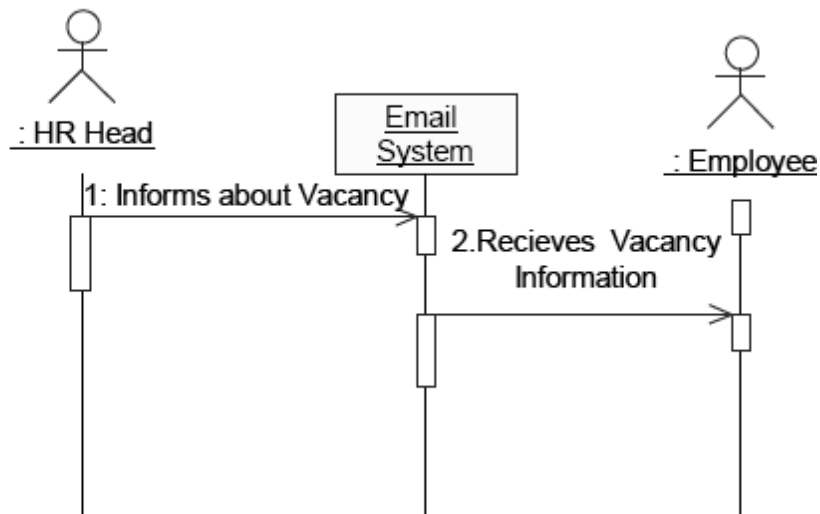


Fig.6.3. SEQUENCE DIAGRAM FOR Admin

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.

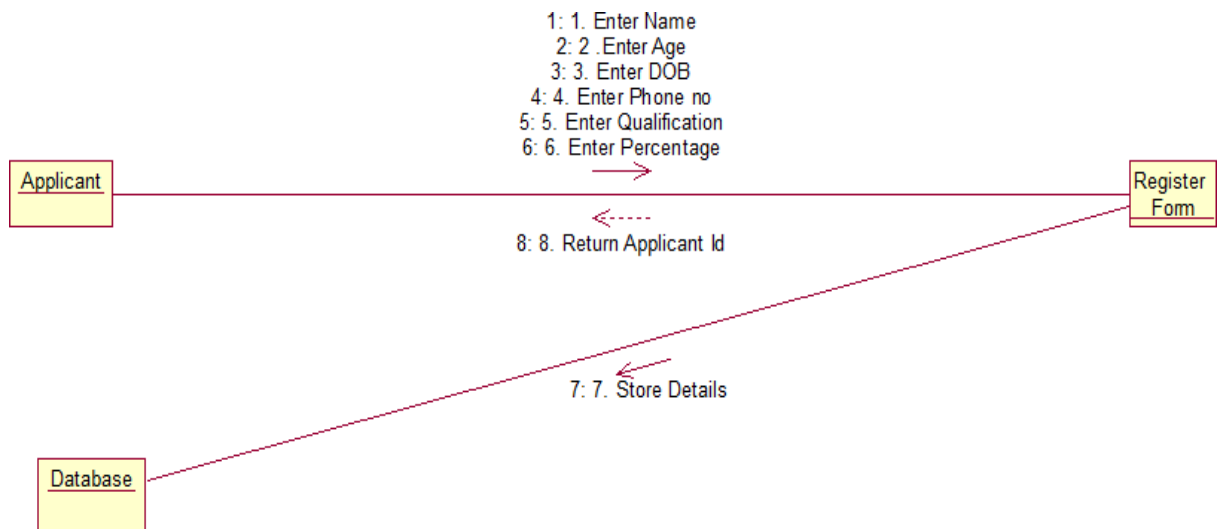


Fig.7.1 COLLABRATION DIAGRAM For Register

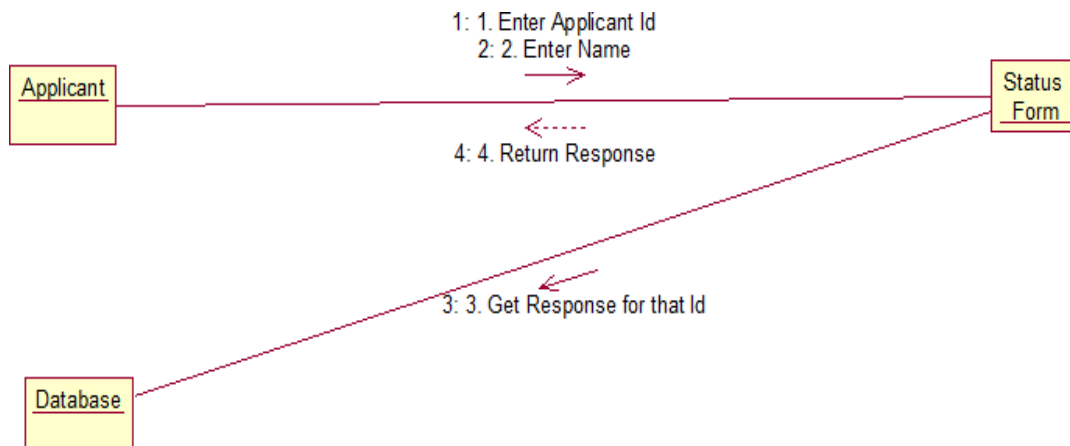


Fig.7.2. COLLABORATION DIAGRAM FOR Status

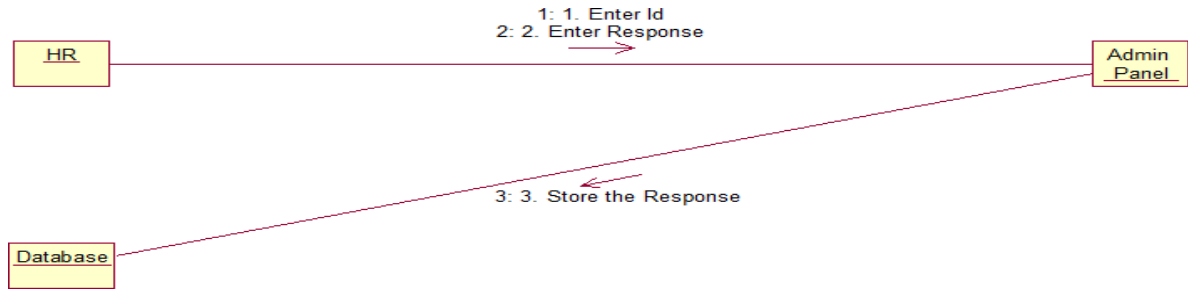
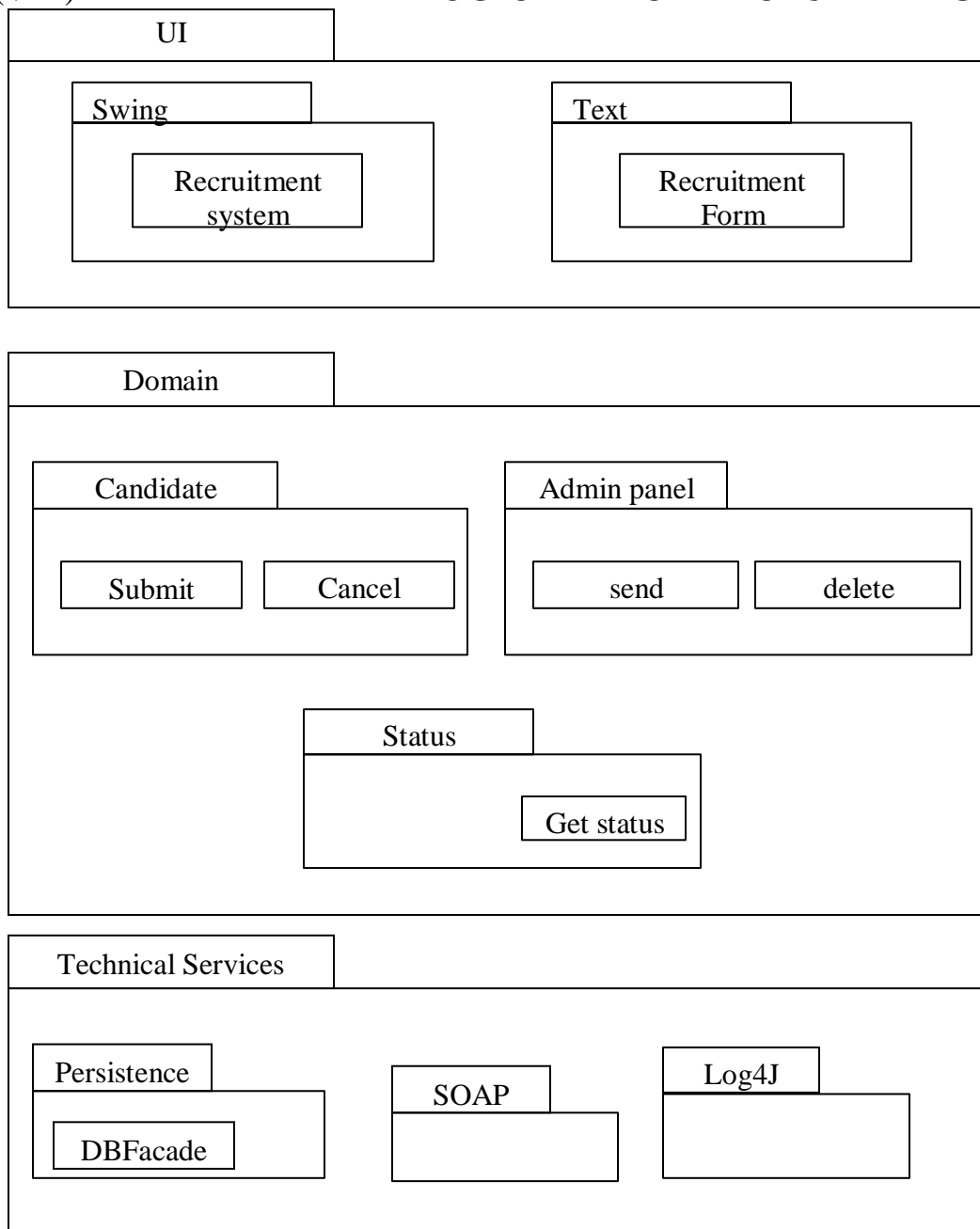


Fig.7.3.COLLABORATION DIAGRAM FOR Admin

(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.

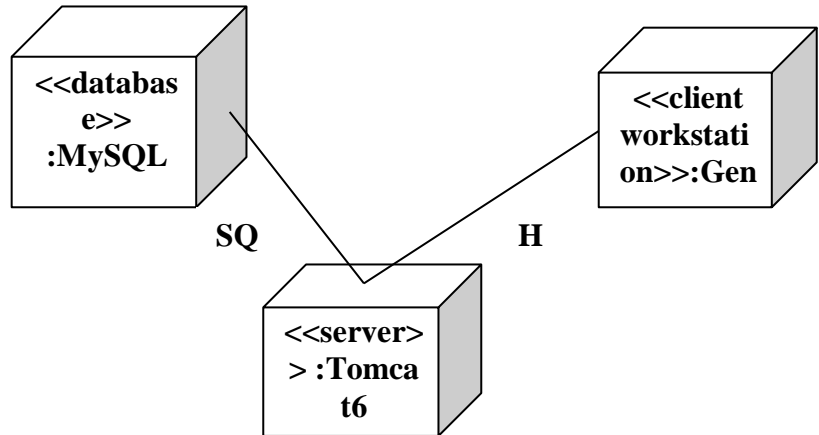


Fig.9.1.DEPLOYMENT DIAGRAM

Component Diagram

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

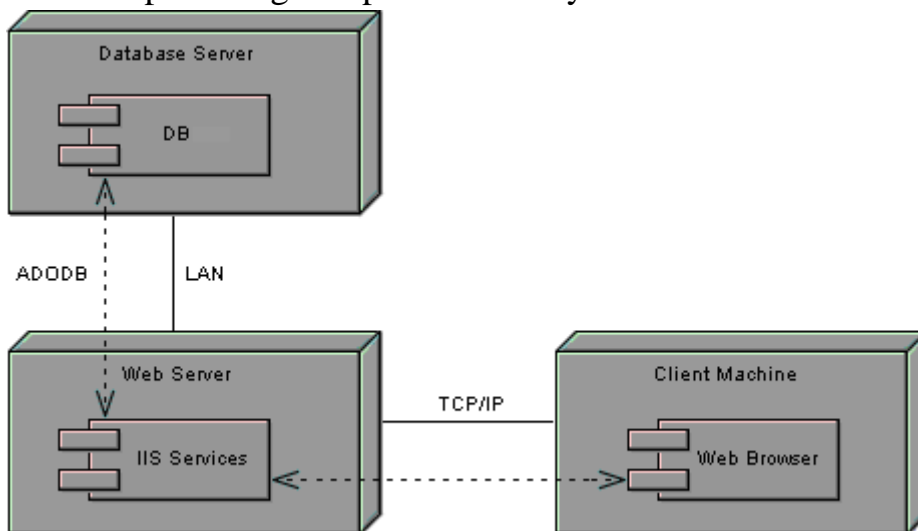


Fig.9.2.COMPONENT DIAGRAM

RESULT:

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 logs out 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

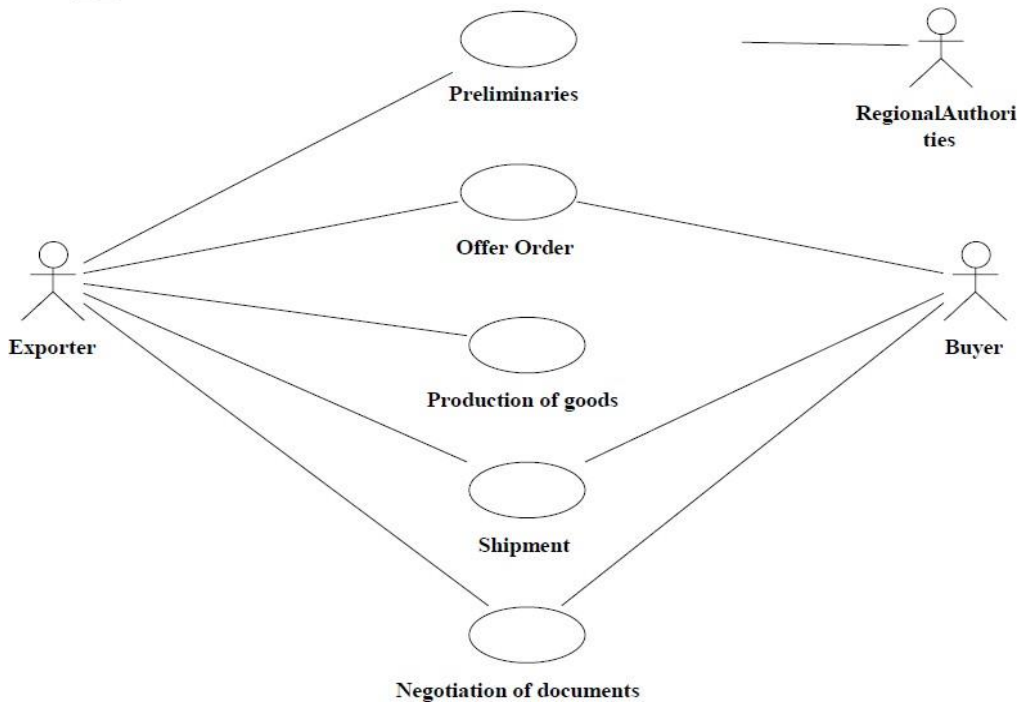
Transfer purchasing power between countries. Obtain credit for international trade transactions. Minimize exposure to the risks of exchange rate changes.

FUNCTIONALITY REQUIREMENTS

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

UML DIAGRAMS

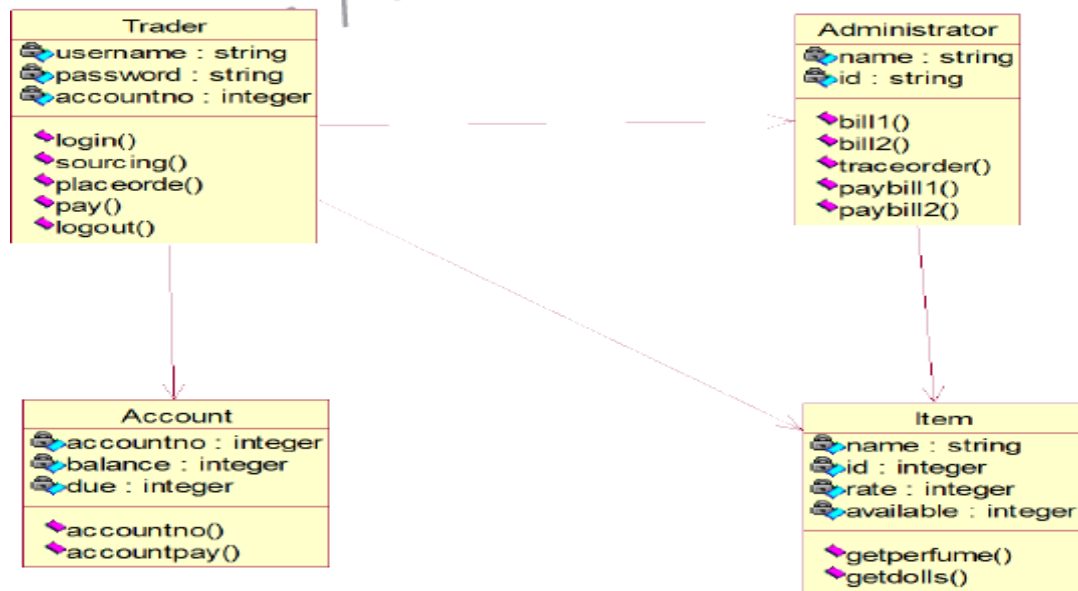
The exporter submits the relevant documents to his buyer (banker) for getting the payment for the goods exported.



(III) USE CASE DIAGRAM

CLASS DIAGRAM

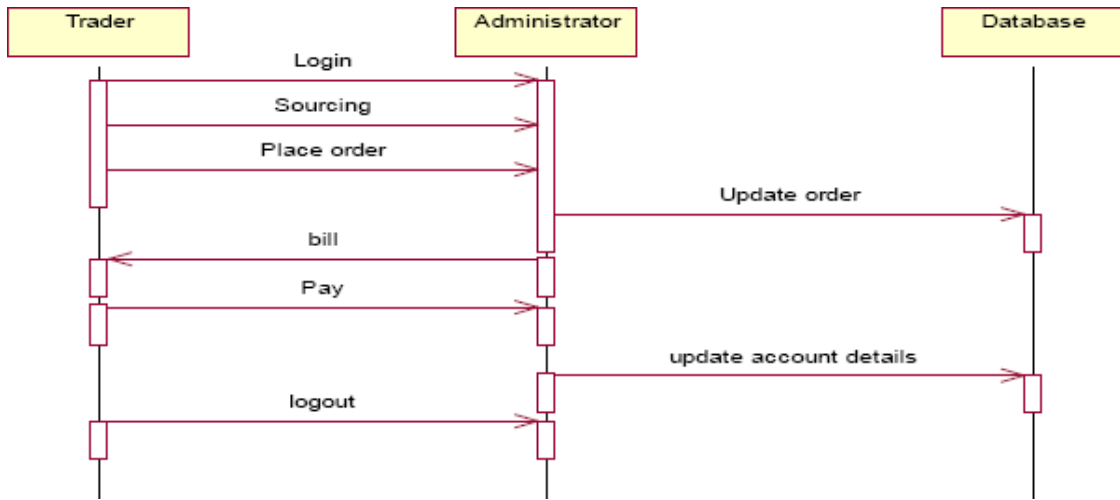
A class diagram is a type of static structure diagram that describes the structure of a system. The classes in the class diagram represent both the main objects and or interactions in the application. The class diagram is represented using rectangular boxes each of which contains three parts:



SEQUENCE DIAGRAM

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

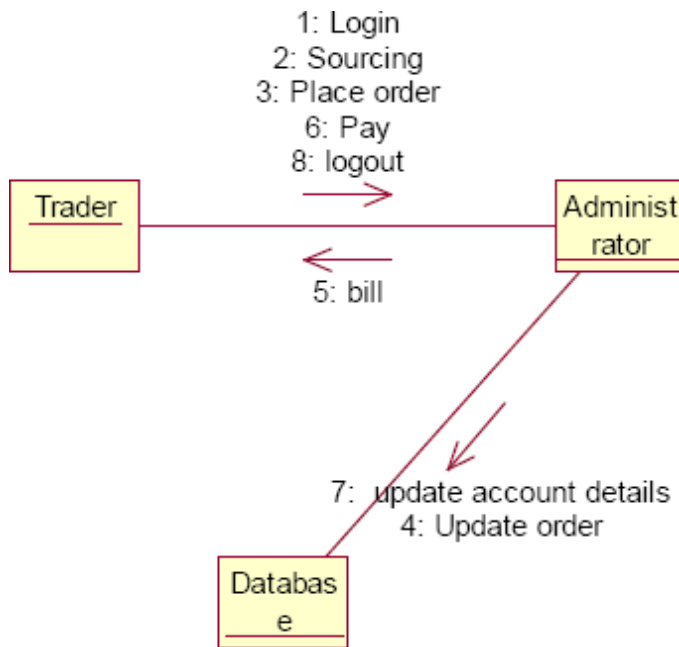
1. Vertical dimension-represents time.
2. Horizontal dimension-represent different object



TraderTraderAdministratorAdministratorDatabaseDatabaseLoginSourcingPlace orderUpdateorderPay update account details bill logout

COLLABORATION DIAGRAM

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



DOCUMENTATION OF COLLABORATION DIAGRAM

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

STATE CHART DIAGRAM

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

DOCUMENTATION OF STATE CHART DIAGRAM

The state diagram represents the following states.

- The trader logs in 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

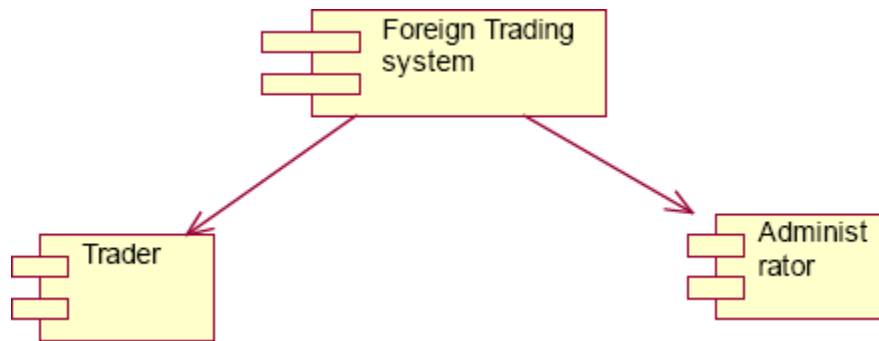
This activity diagram represents the flow of stepwise activities performed in foreign trading system.

- The first action represents the trader logs in 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.

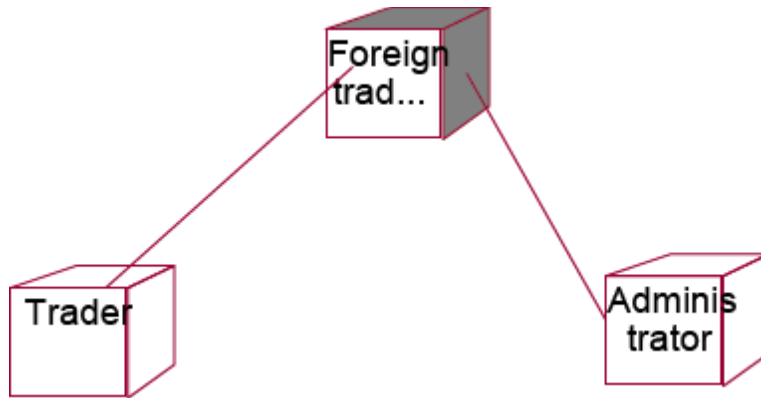


DOCUMENTATION OF COMPONENT DIAGRAM

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 logs in 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

Date:

AIM

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

(I)PROBLEM STATEMENT

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

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

(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.

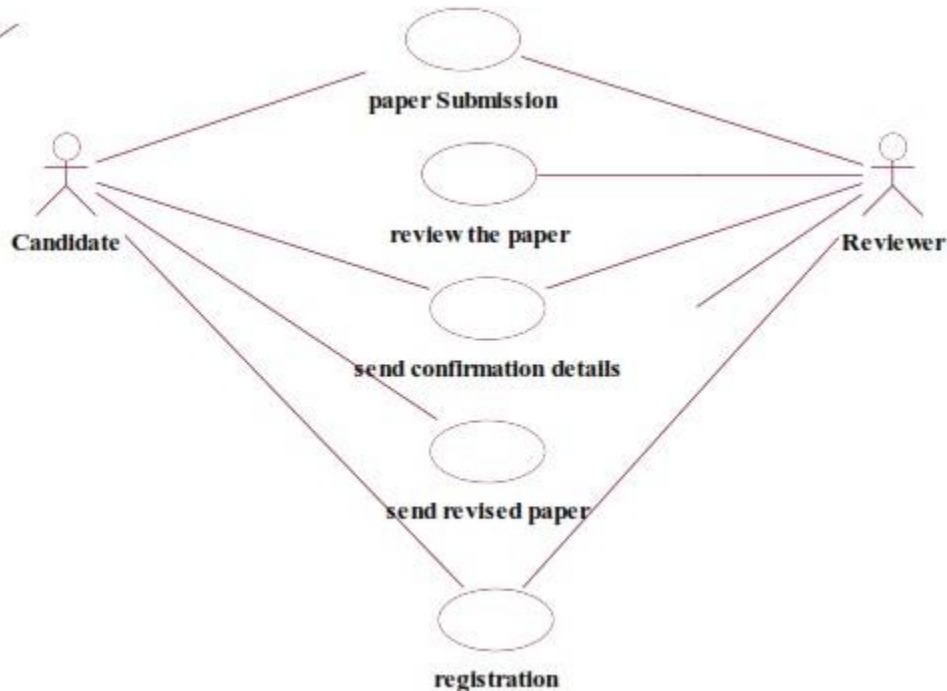
The use cases in this use case diagram are

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

Paper submission– Candidate submits the paper.

- **Review the paper**– The paper is been reviewed by the reviewer and the paper is selected.
- **Paper confirmation details** – The reviewer can send the confirmation details to the candidate.

- **Revised and camera ready paper** – After the paper is selected and the camera ready paper should be submitted to the reviewer by candidate.
- **Registration** – After submitting the revised paper the candidate wants to register.



CLASS DIAGRAM

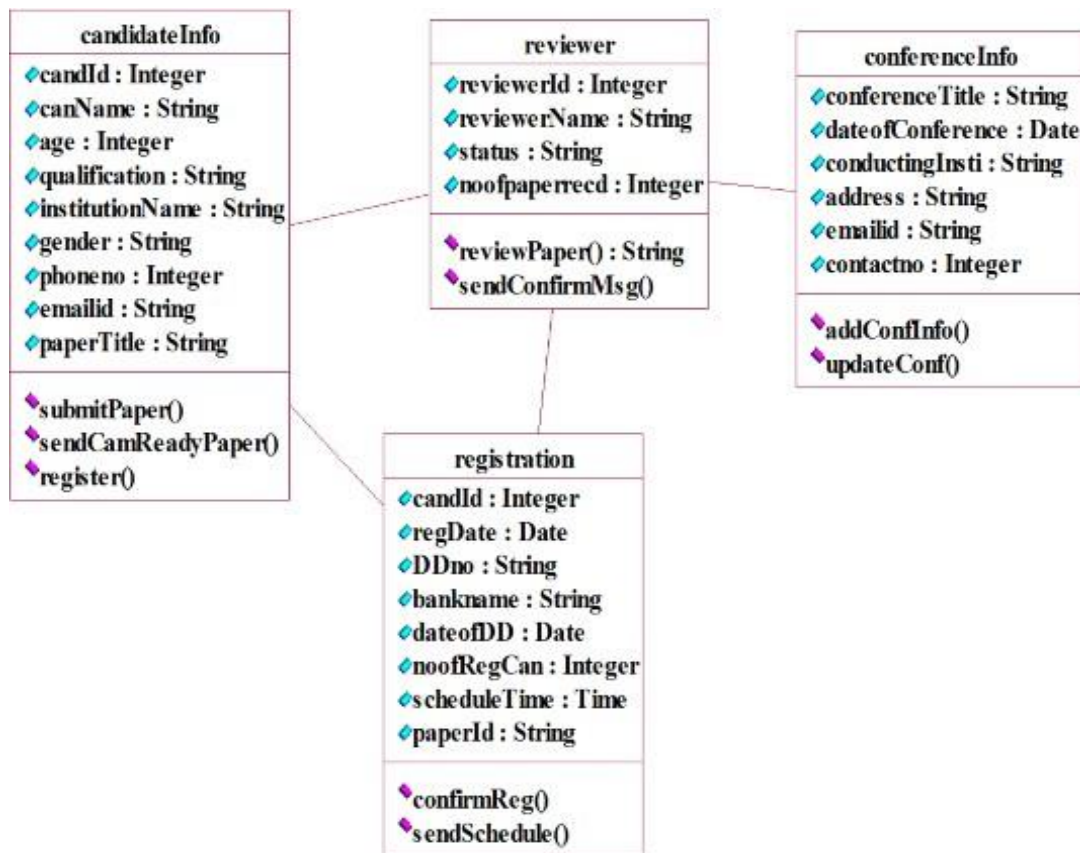
A class diagram in the unified modeling language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment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

DOCUMENTATION OF CLASS DIAGRAM

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

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

- **Reviewer** – Its attributes are name, department, reviewer ID The operations performed are review the paper and send the paper confirmation details.
- **Database** –The operations performed are storing candidate details and verifying login .

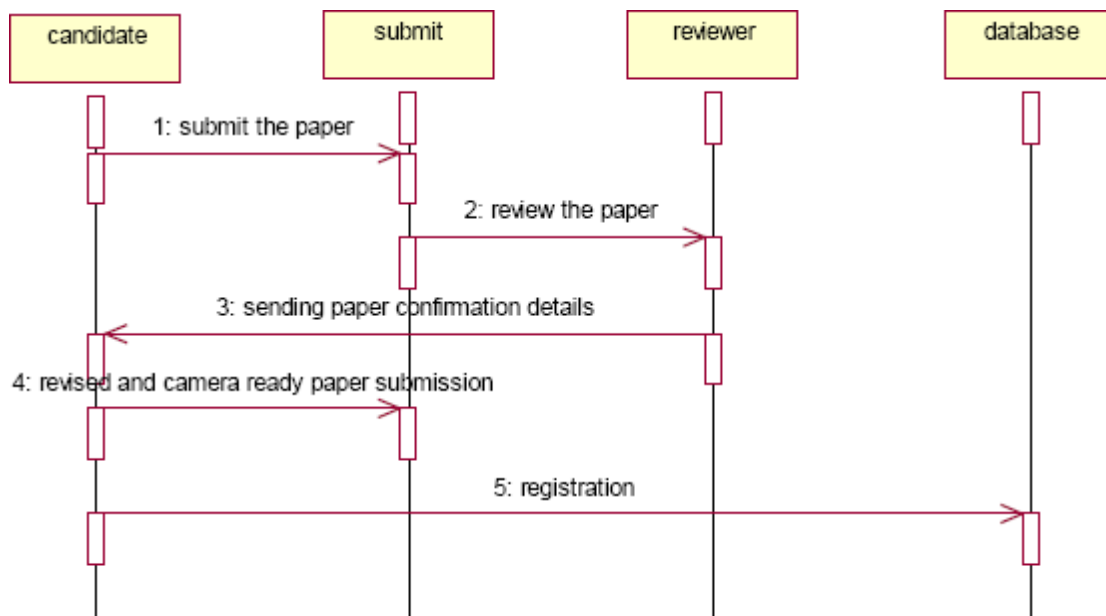


SEQUENCE DIAGRAM

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

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

SEQUENCE DIAGRAM FOR SUBMITTING PAPER



DOCUMENTATION OF SEQUENCE DIAGRAM LOGIN

This sequence diagram describes the sequence of steps to show

- The candidate login in to the conference system and register for job.
- The verification done in the database .

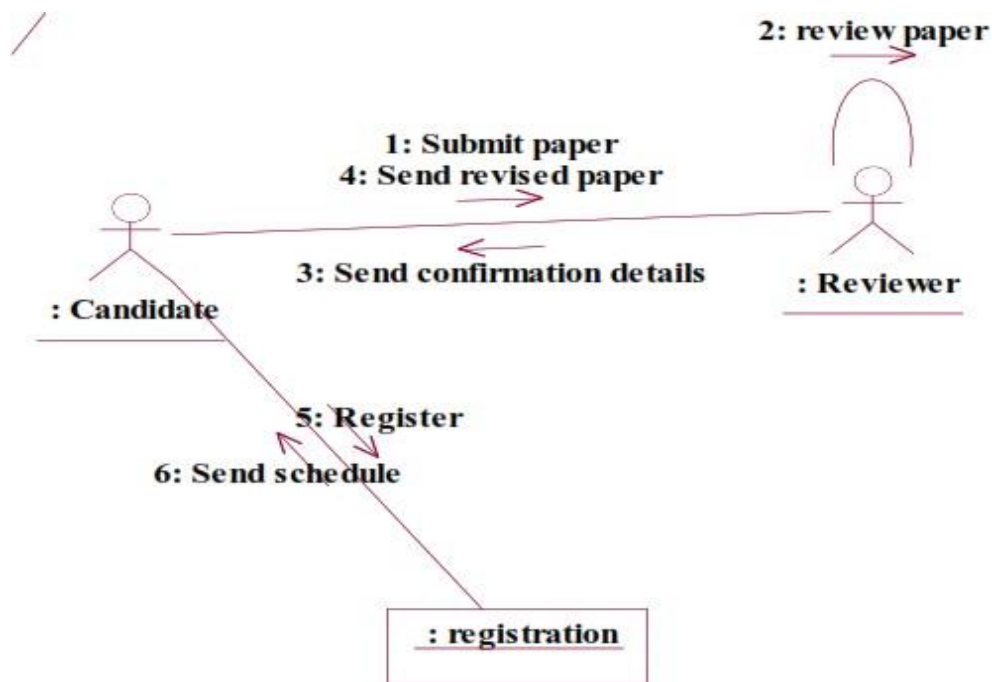
PAPER SUBMISSION

This sequence diagram shows steps to show

- The candidate submit the paper.
- The reviewer reviews the paper and sends acknowledgement 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, is a sophisticated modeling tool that can easily convert a collaboration diagram into a sequence diagram and 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.

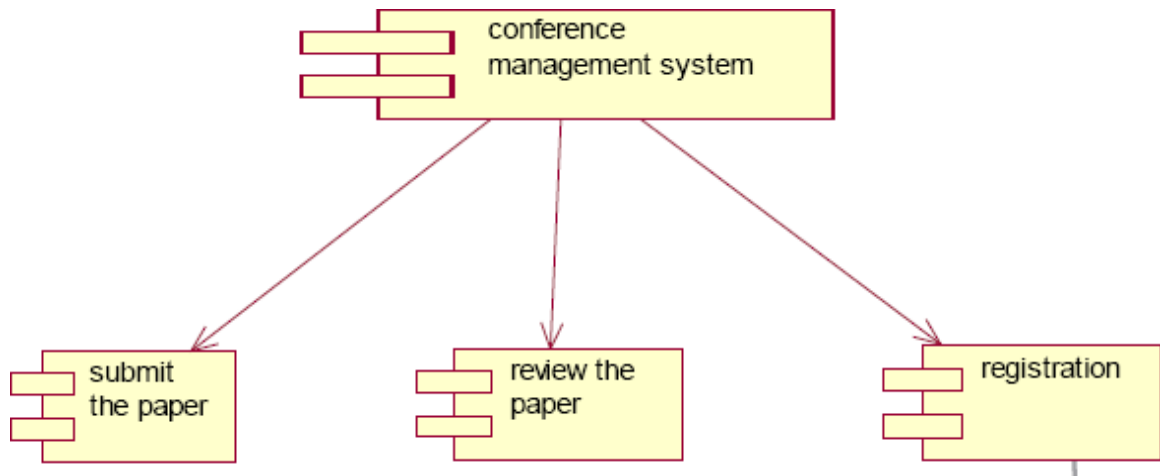
DOCUMENTATION OF ACTIVITY DIAGRAM

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.

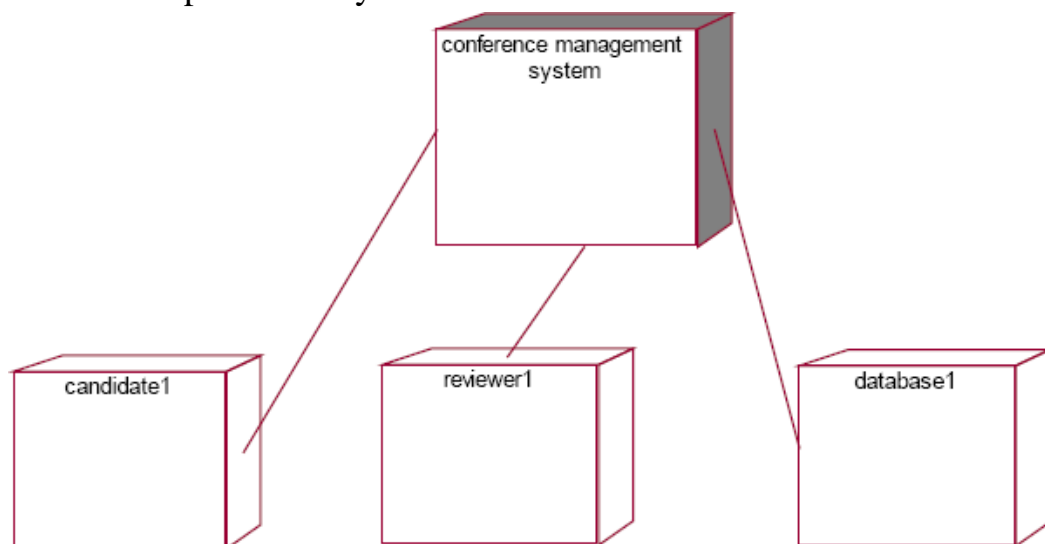


DOCUMENTATION OF COMPONENT DIAGRAM

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.

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.

2.2 SOFTWARE 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.

2.3 HARDWARE 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:

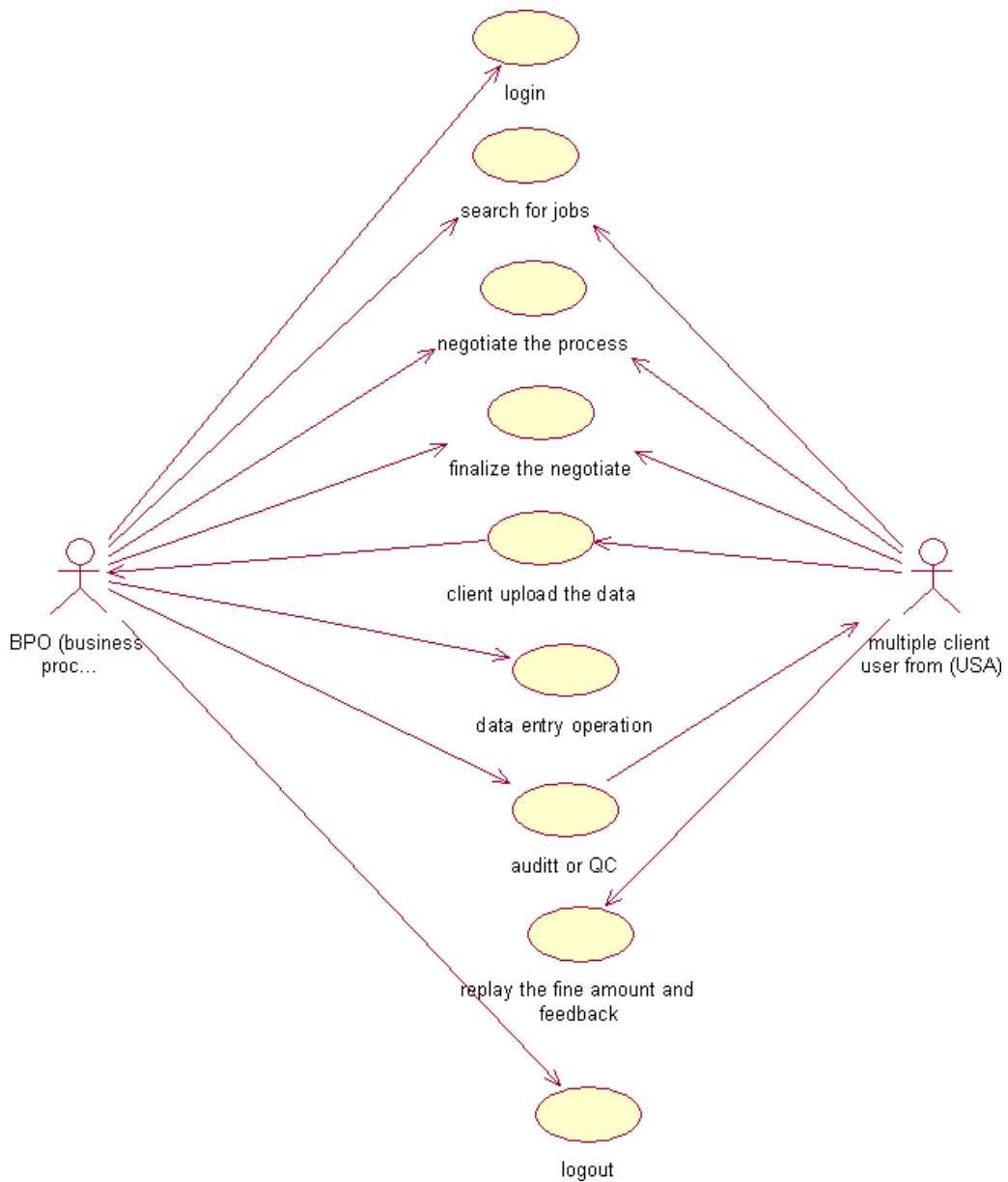


Fig.3. UML USE CASE DIAGRAM

(IV) UML CLASS DIAGRAM:

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

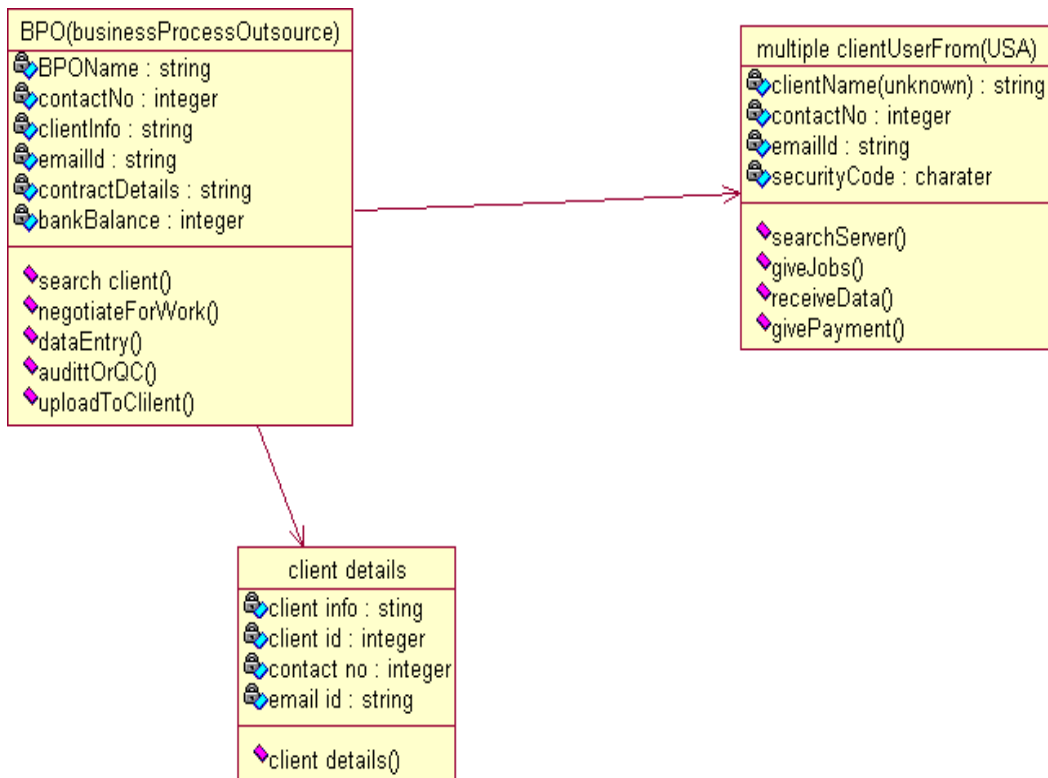


Fig.5. UML CLASS DIAGRAM

(V)UML SEQUENCE DIAGRAM:

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

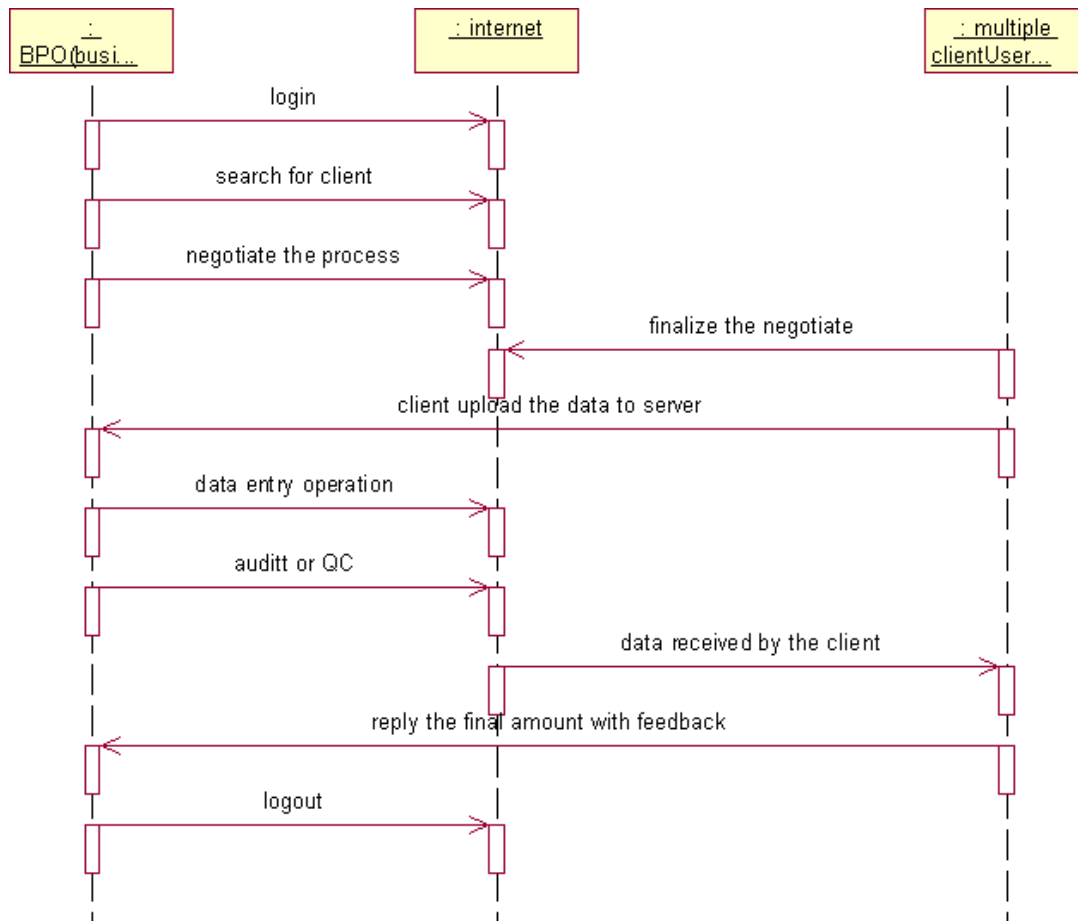


Fig. 5.1 SEQUENCE DIAGRAM

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

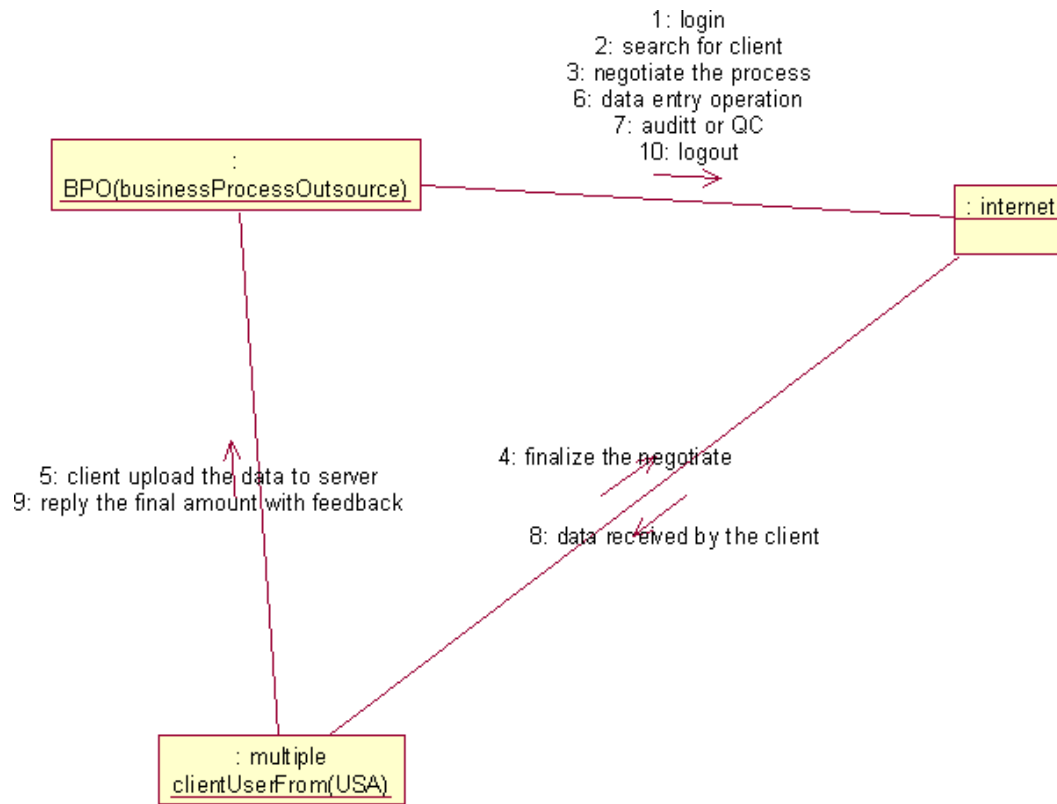
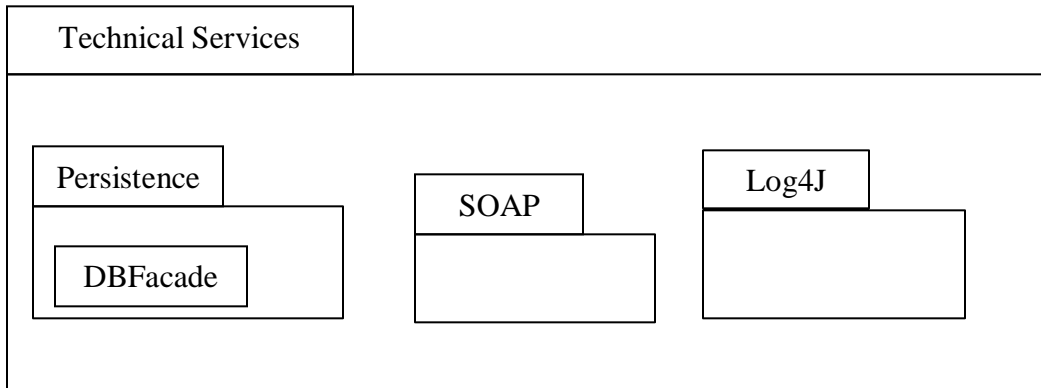
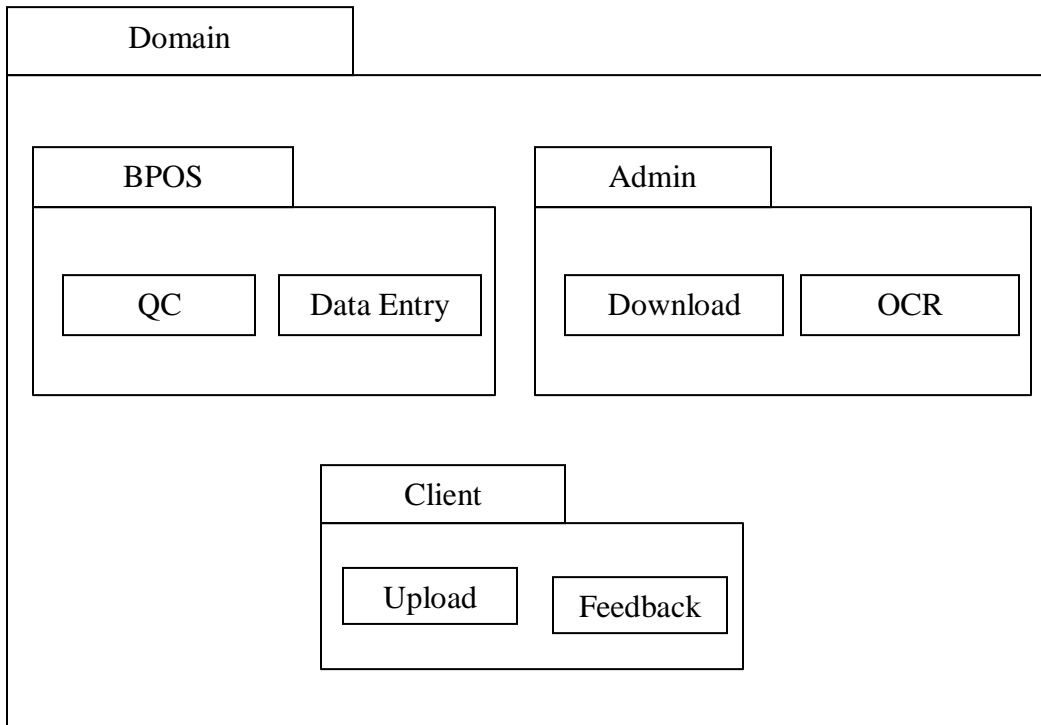
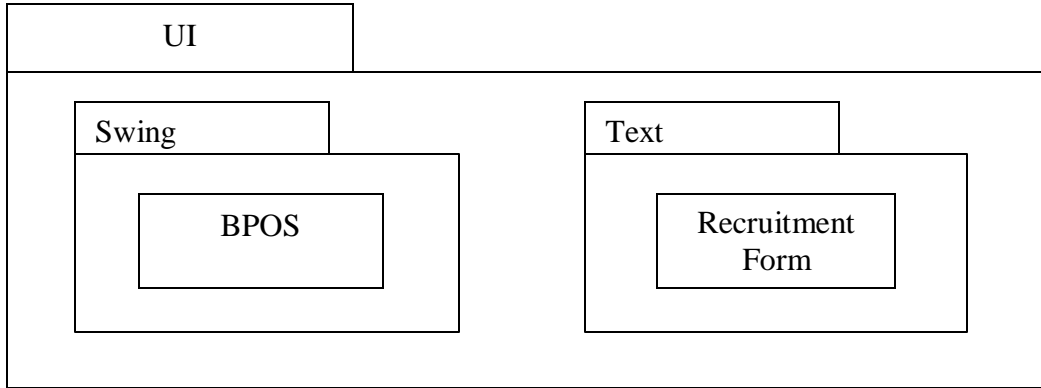


Fig.5.2COLLABRATION DIAGRAM

(VI) PARTIAL LAYERD LOGICAL ARCHITECTURE DIAGRAM:



(VIII) DEPLOYMENT DIAGRAM AND COMPONENT DIAGRAM

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

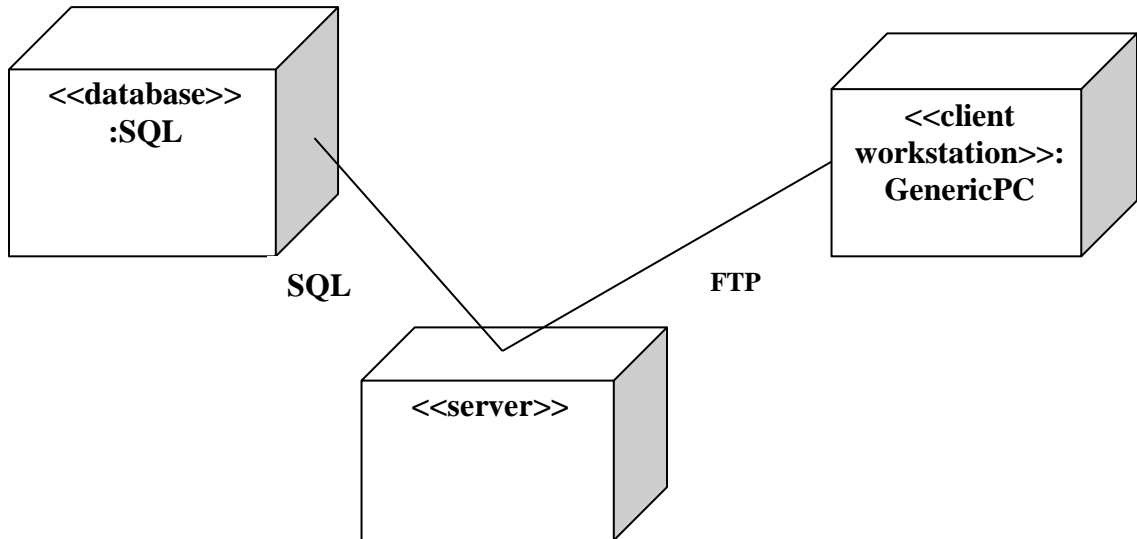
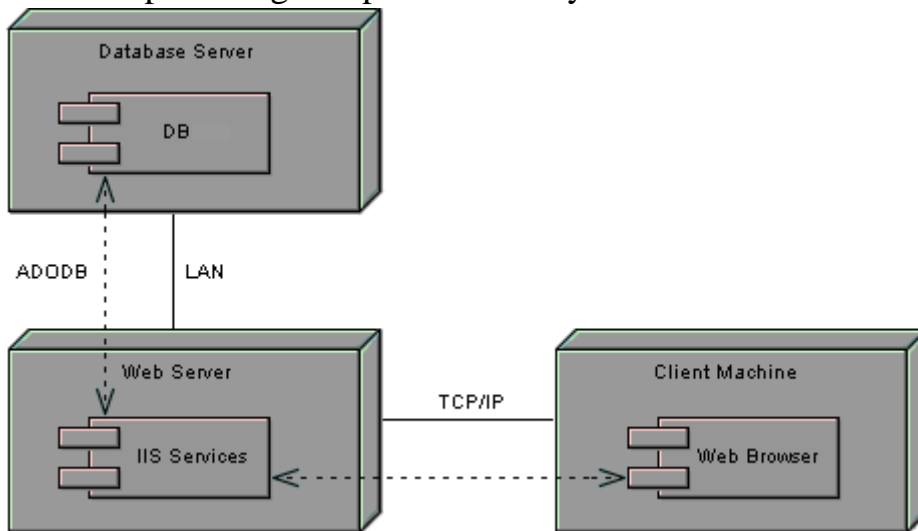


Fig.9.1.DEPLOYMENT DIAGRAM

Component Diagram

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



RESULT :

Ex.No:15

LIBRARY MANAGEMENT SYSTEM

Date:

AIM

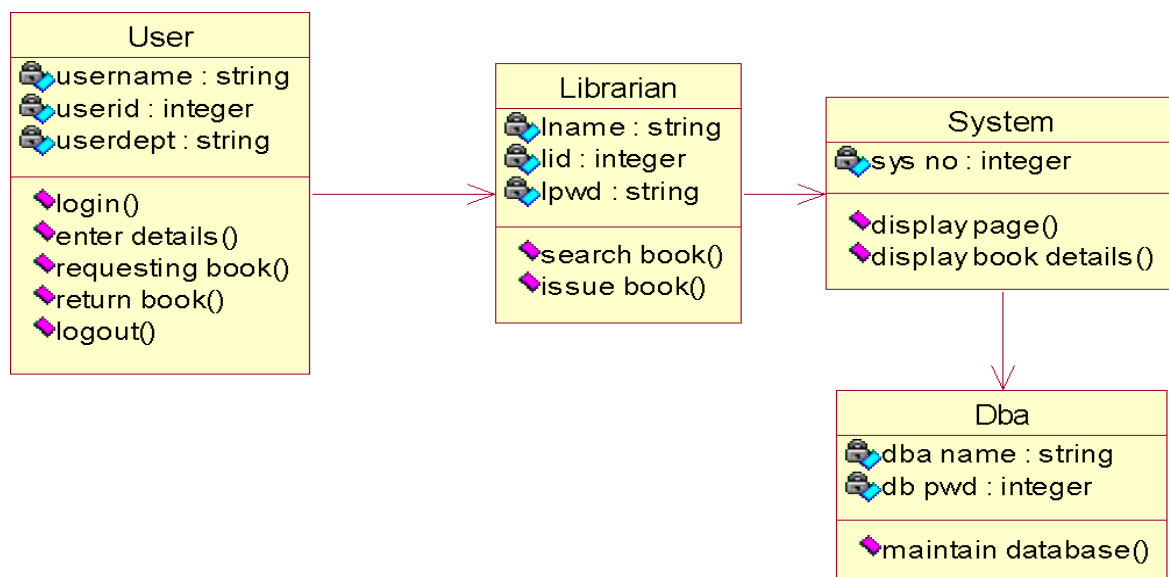
To design an object oriented model for Library Management System using Rational Rose software and to implement it using Java.

PROBLEM STATEMENT

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

CLASS DIAGRAM

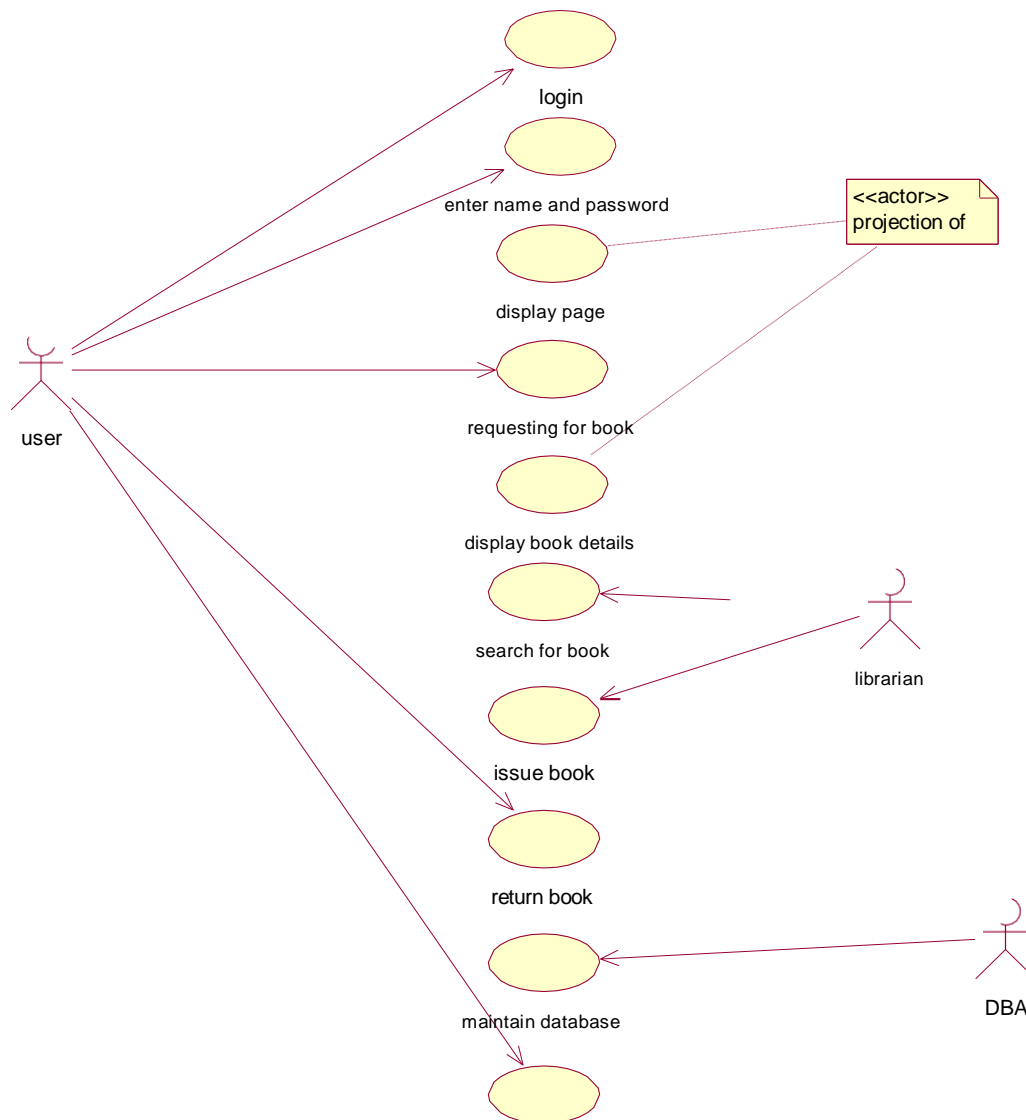
A class diagram in the unified modeling language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and the relationships among objects. The library management system makes use of the following classes user,



librarian, system and DBA.

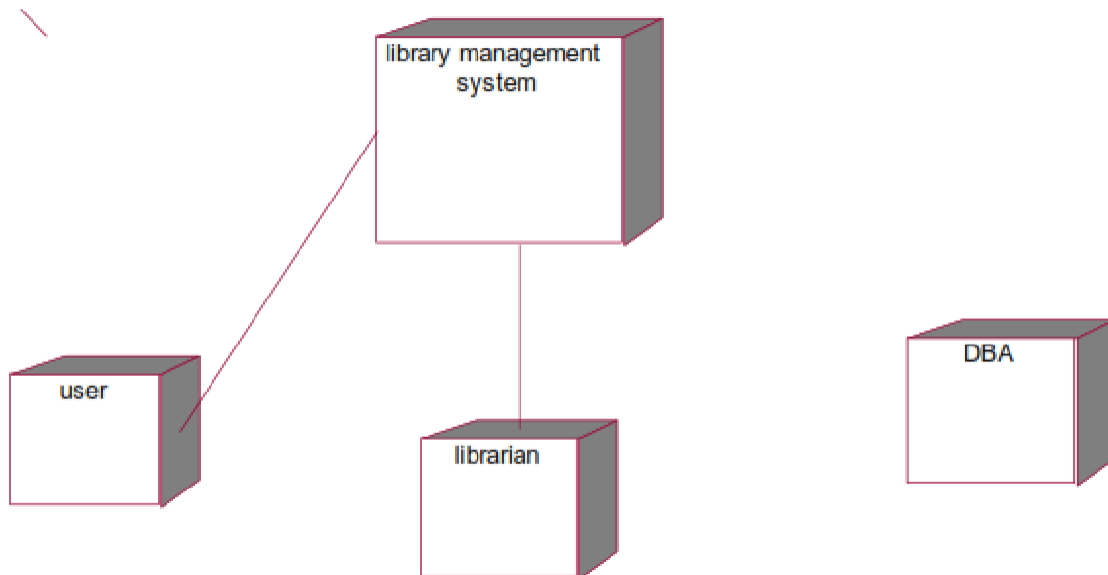
USE CASE DIAGRAM

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



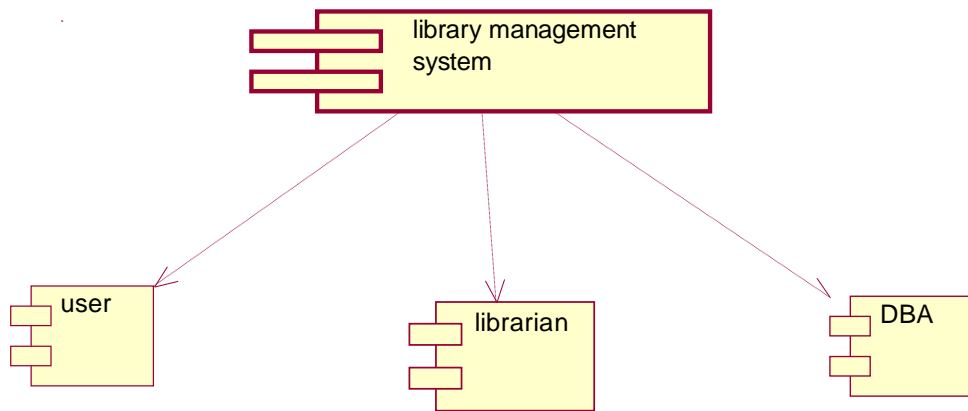
DEPLOYMENT DIAGRAM

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



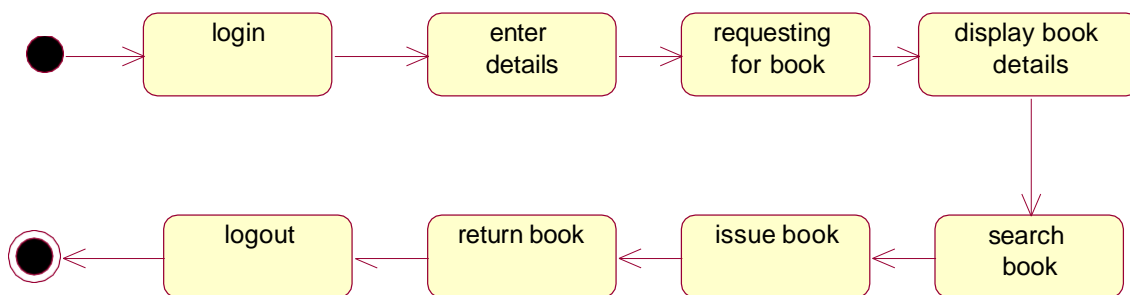
COMPONENT DIAGRAM

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



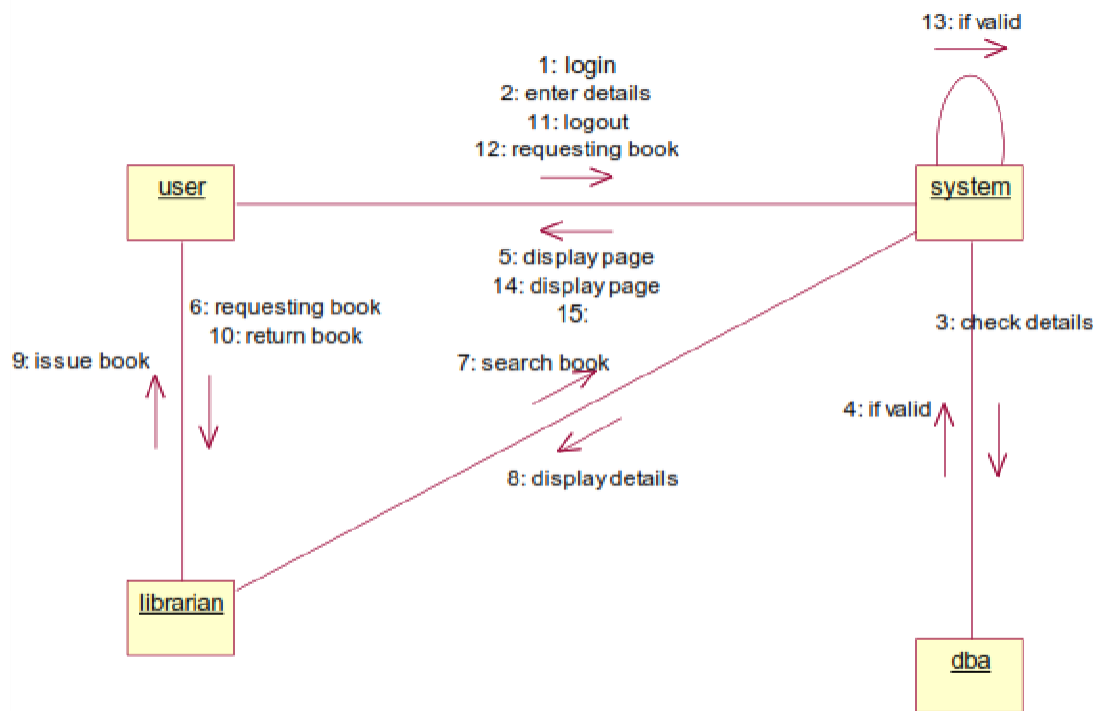
STATECHART DIAGRAM

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



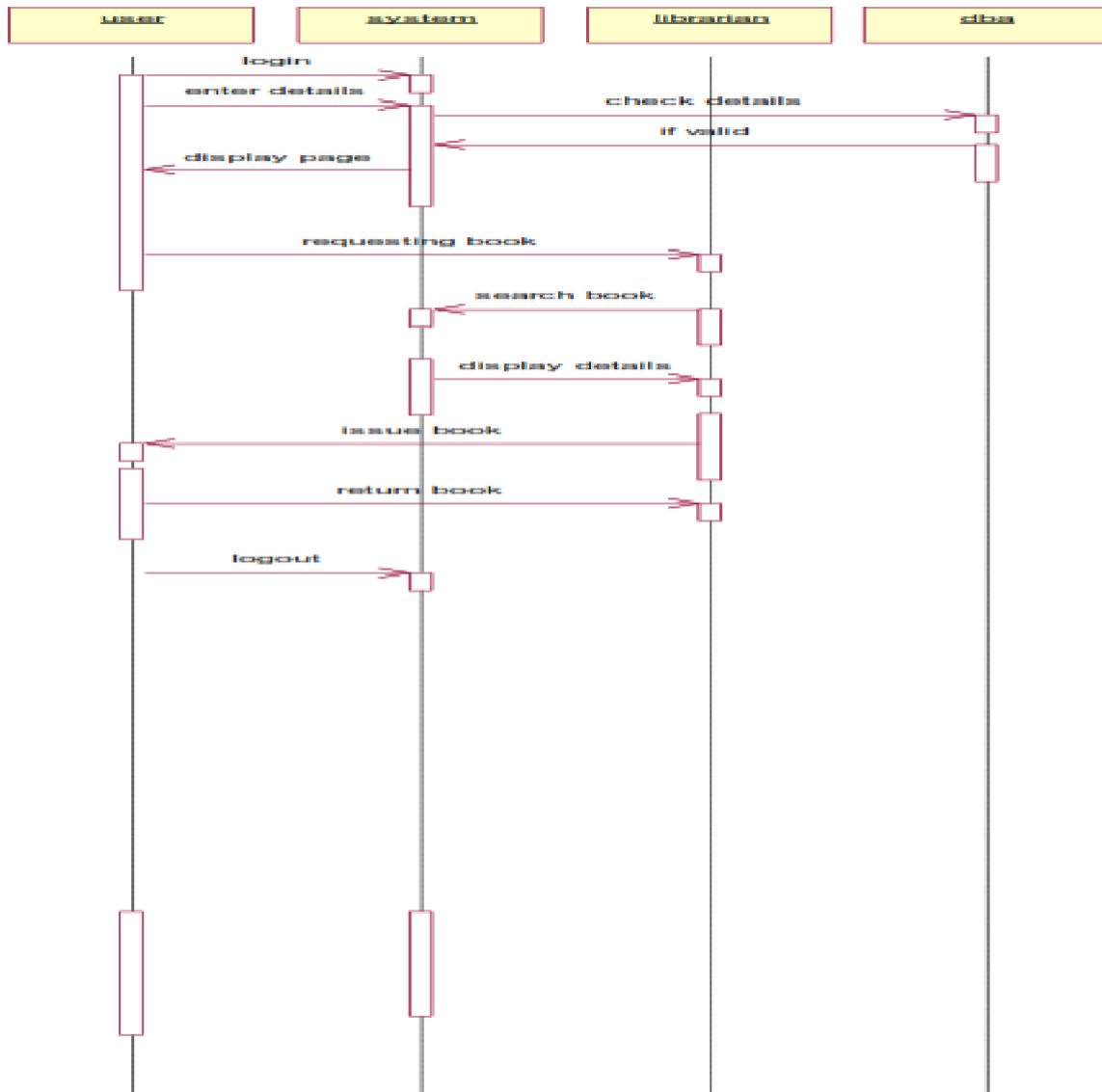
COLLABORATION DIAGRAM

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



SEQUENCE DIAGRAM

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



ACTIVITY DIAGRAM

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

RESULT

Ex.No:16

STUDENT INFORMATION SYSTEM

Date:

AIM

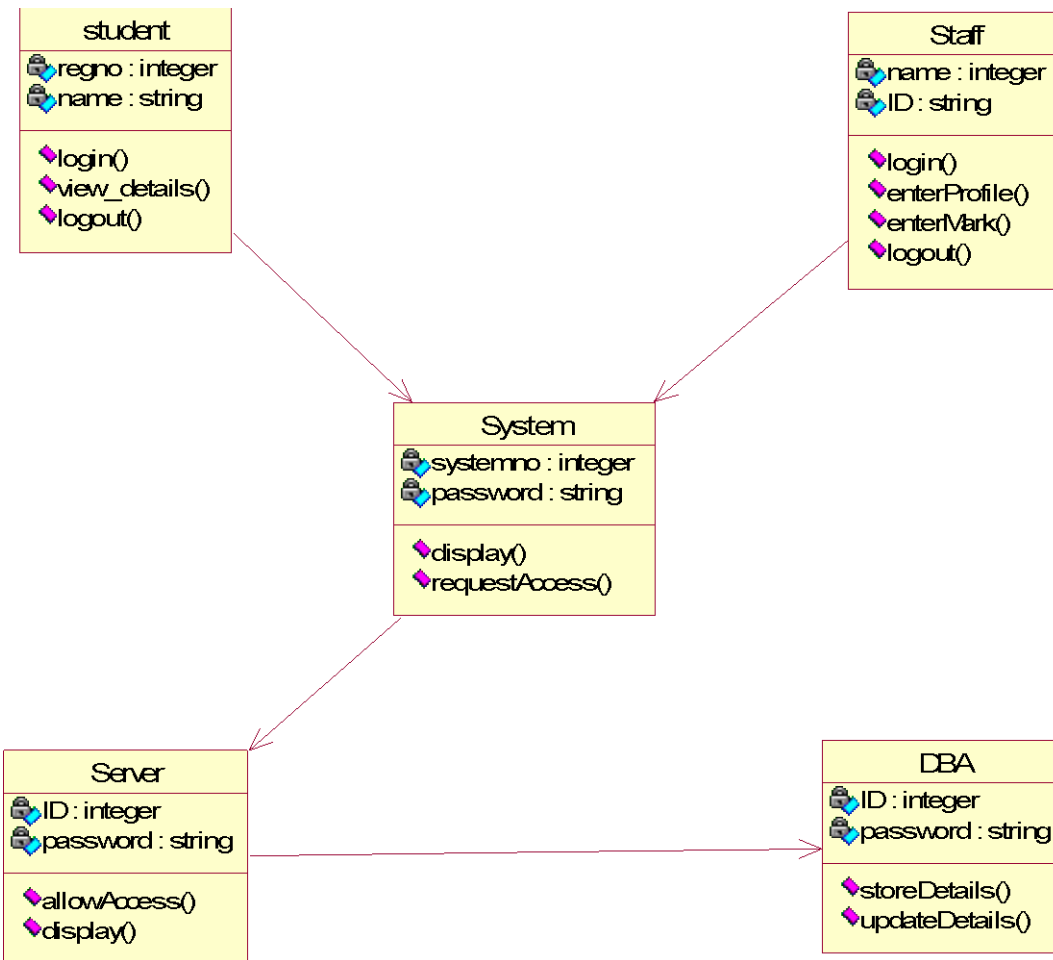
To design an object oriented model for Student information system using Rational Rose software.

PROBLEM STATEMENT

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

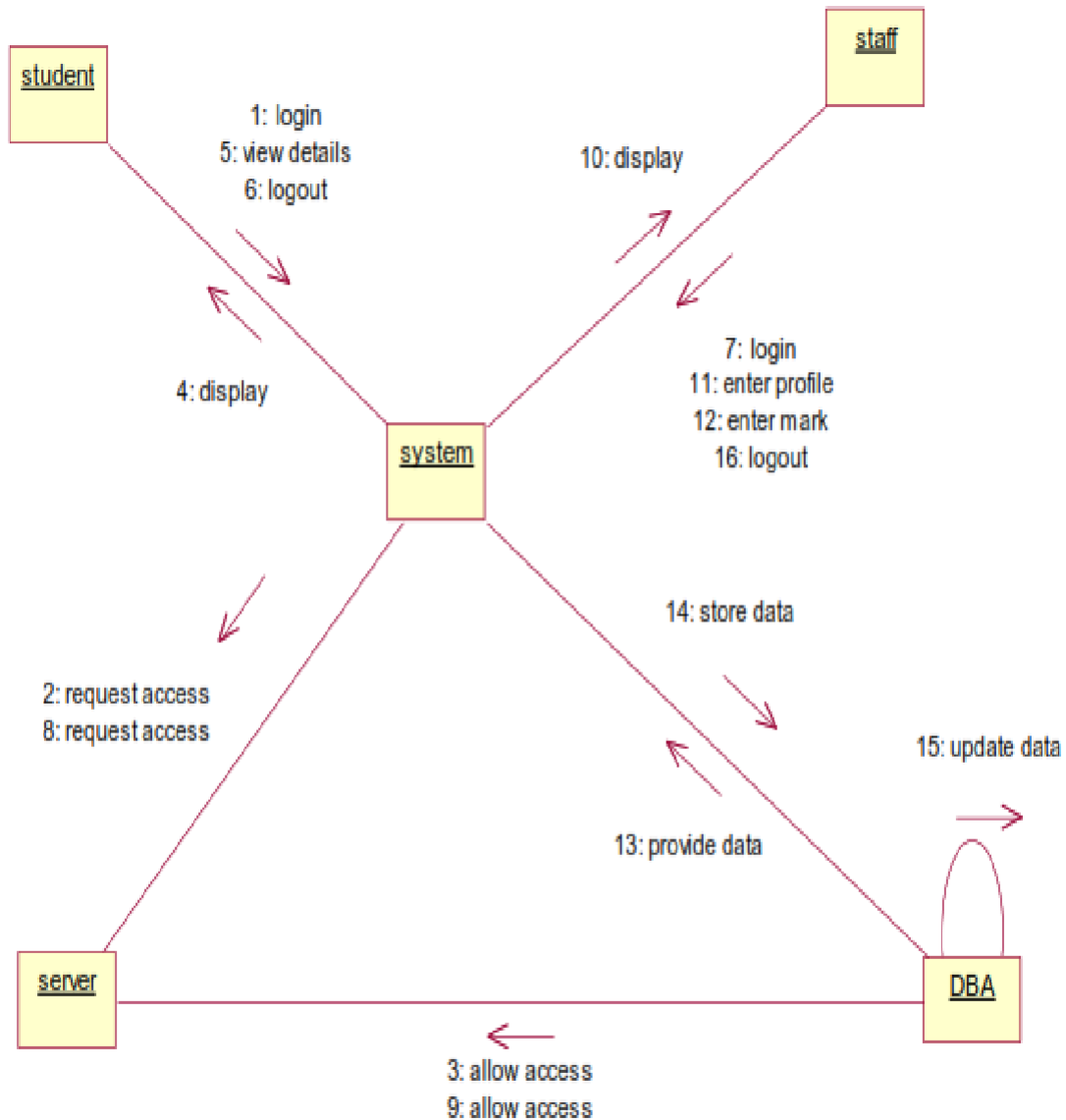
CLASS DIAGRAM

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



COLLABORATION DIAGRAM

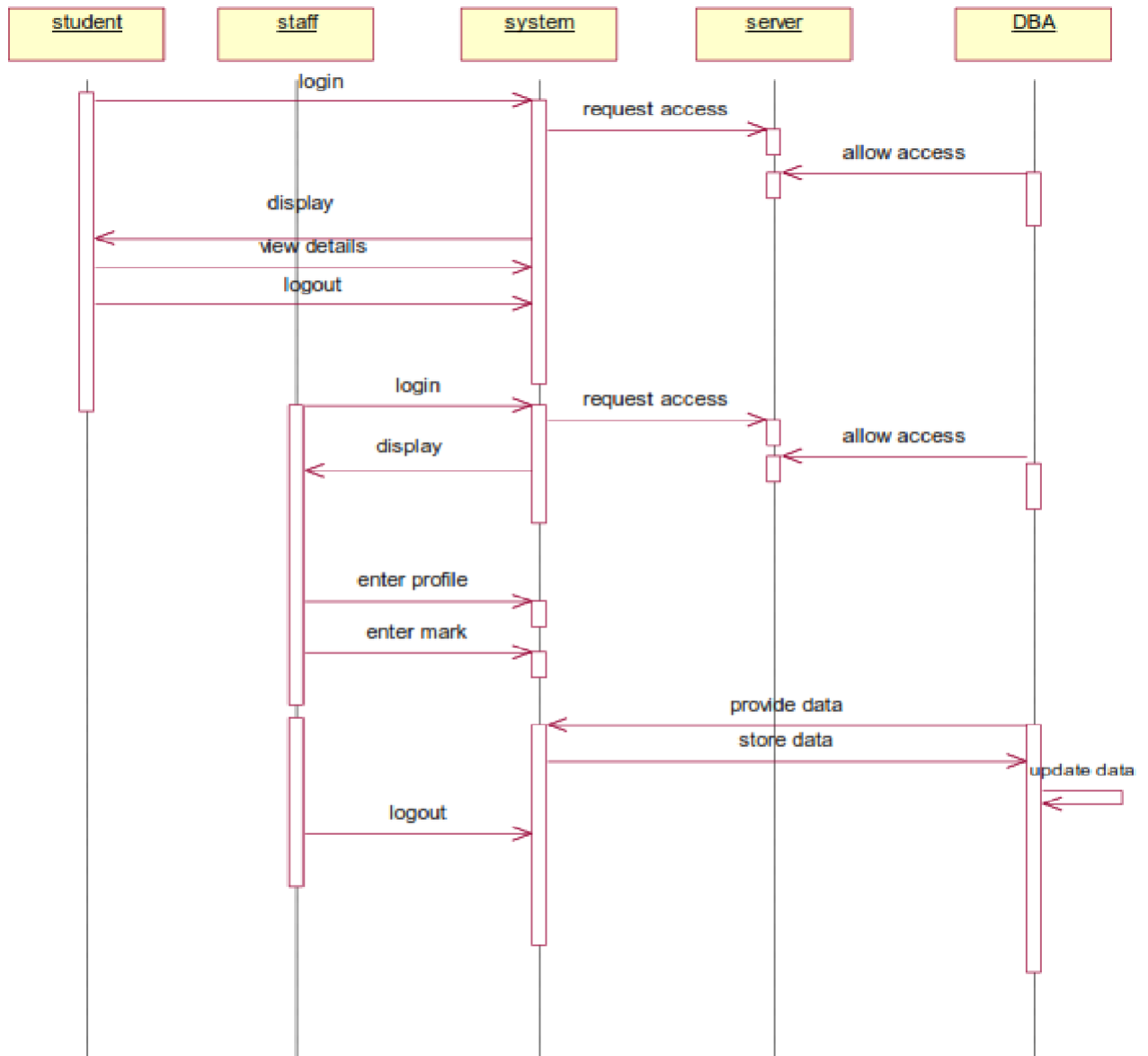
A Collaboration diagram represents the collaboration in which is a set of objects related to achieve a desired outcome. In collaboration, the sequence is indicated by numbering the message several numbering schemes are available. Login, request access, allow access, display, view details, logout, login,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



SEQUENCE DIAGRAM

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

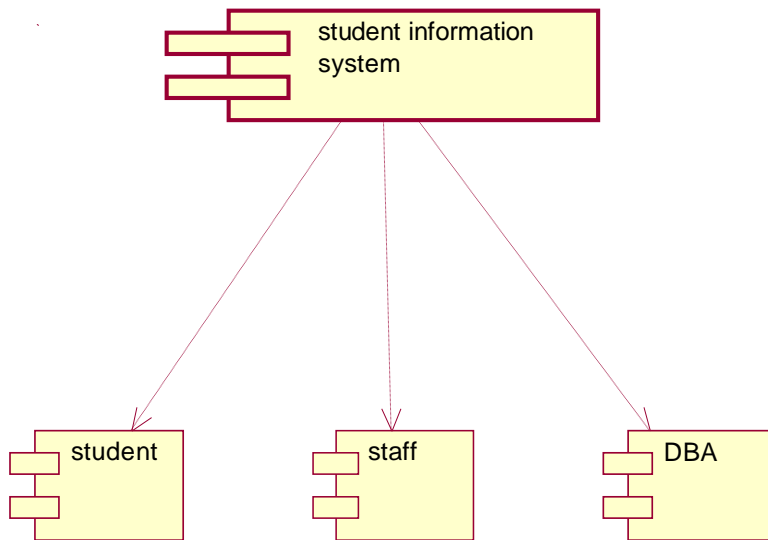
by database. The student first login to the system and then view the details of the details. Staff login to the system enter mark and enter the details of the student. DBA store and update the details of the student.



COMPONENT DIAGRAM

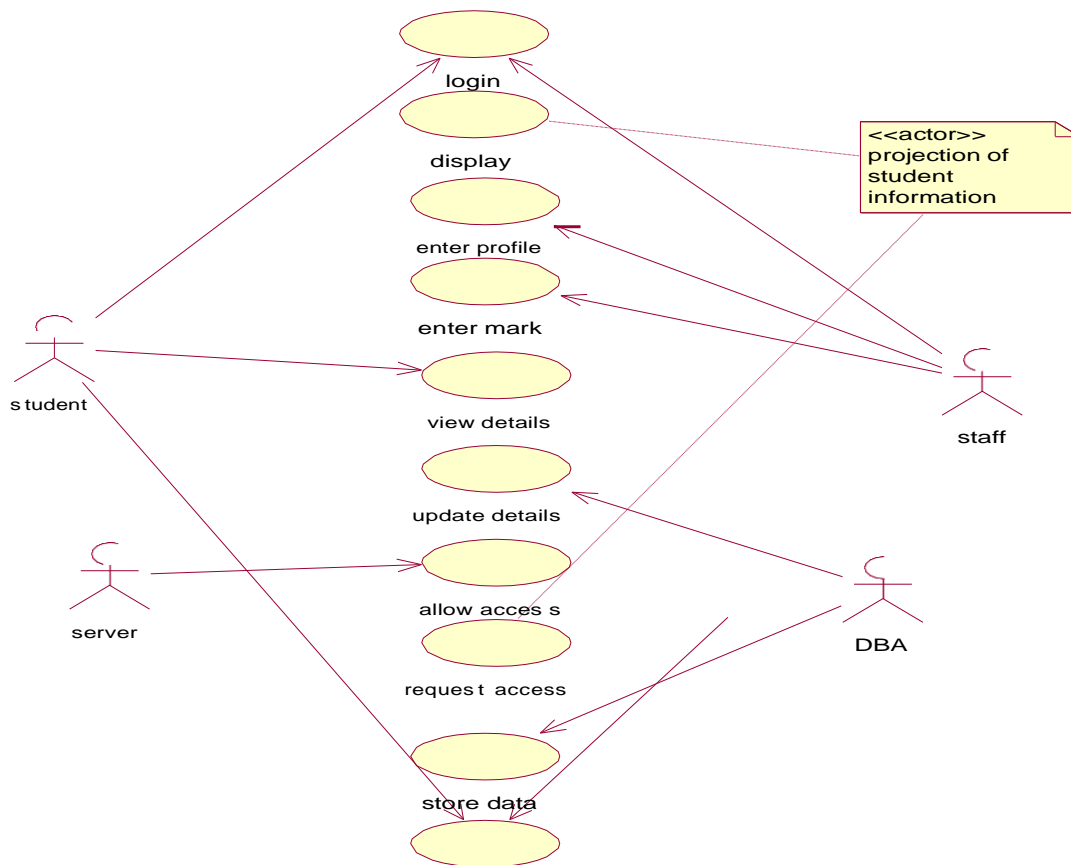
Component diagram carries the major living actors of the system. The component diagram main purpose is to show the

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



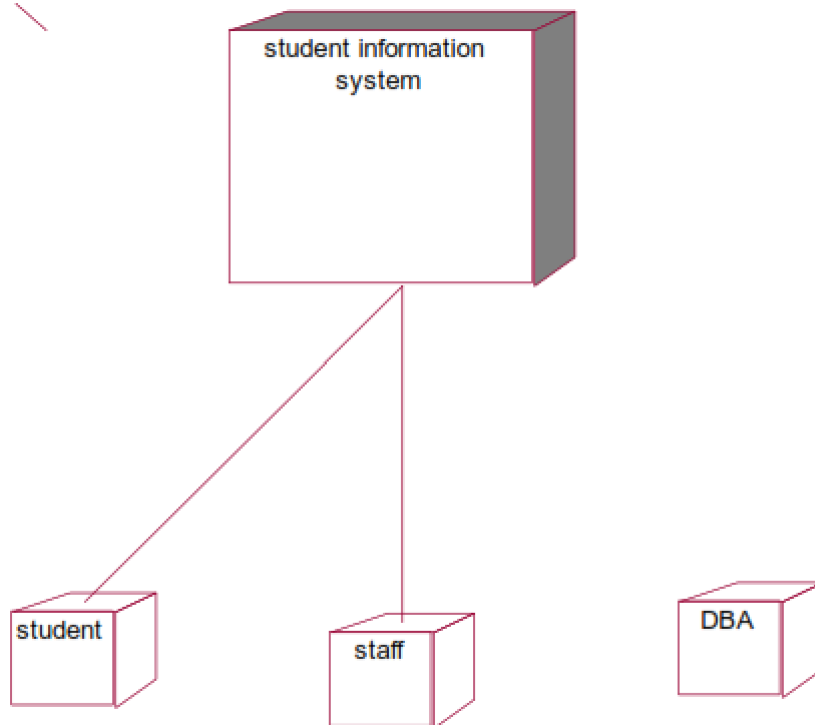
USE CASE DIAGRAM

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



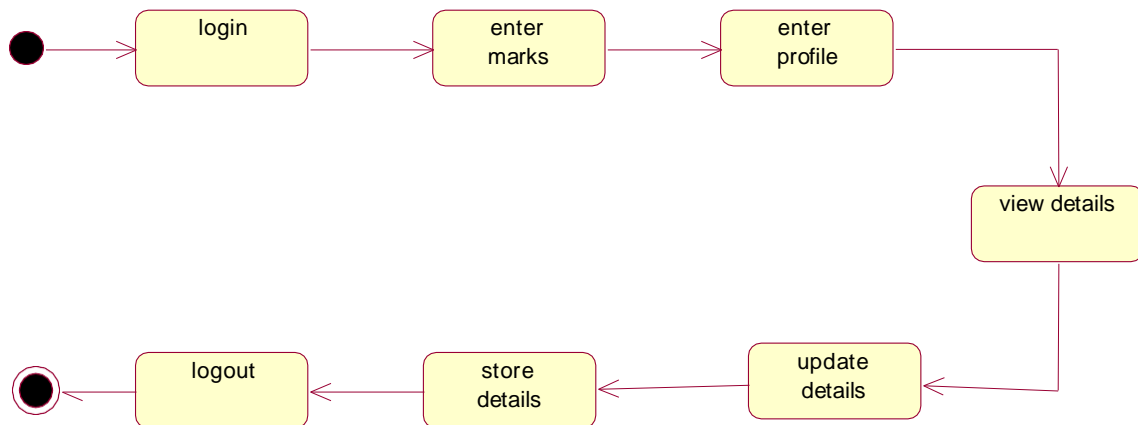
DEPLOYMENT DIAGRAM

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



STATE CHART DIAGRAM

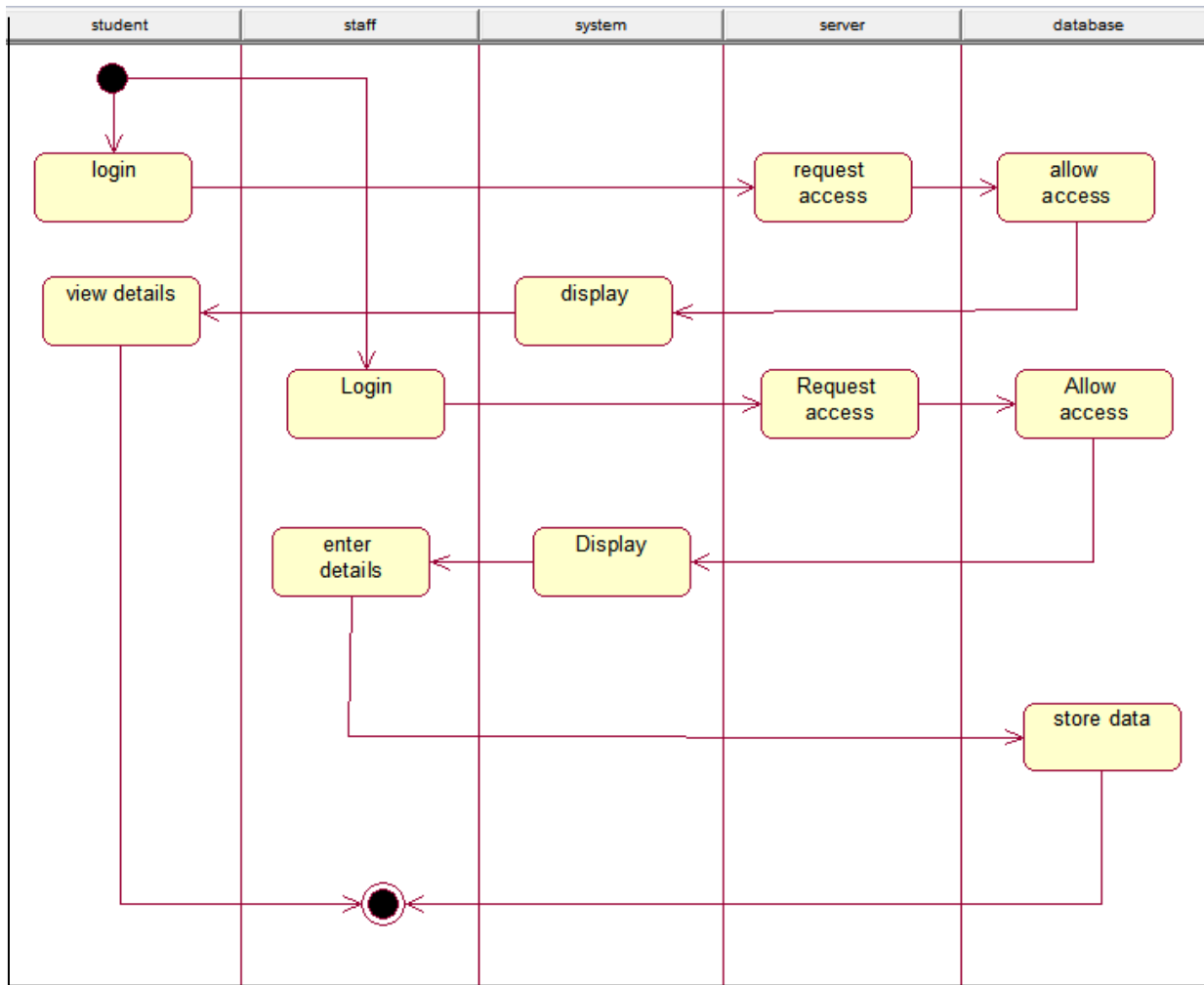
A State chart diagram is also called as state machine diagram. The state chart contains the states in the rectangular boxes and the states are indicated by the dot enclosed. The state chart diagram describes the behavior of the system. The state chart involves six stages such as login, enter mark, enter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



ACTIVITY DIAGRAM

Activity diagram are graphical representation of stepwise

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



RESULT:

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)

Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMMON TO: DEPARTMENT OF INFORMATION TECHNOLOGY



CCS334– BIG DATA ANALYTICS

R2021

LAB MANUAL

INDEX

S.NO.	NAME OF THE EXPERIMENT	PAGE NO	DATE OF EXPERIMENT	REMARKS
1	DOWNLOADING AND INSTALLING HADOOP; UNDERSTANDING DIFFERENT HADOOP MODES, STARTUP SCRIPTS, CONFIGURATION FILES			
2	HADOOP IMPLEMENTATION OF FILE MANAGEMENT TASKS			
3	IMPLEMENTATION OF MATRIX MULTIPLICATION WITH HADOOPMAP REDUCE			
4	WORD COUNT MAP REDUCE PROGRAM USING MAP REDUCE PARADIGM			
5	INSTALLATION OF HIVE ALONG WITH PRACTICE EXAMPLES			
6	INSTALLATION OF HBASE, INSTALLING THRIFT ALONG WITH PRACTICE EXAMPLES			
7	HIVE EXECUTION WITH PRACTICE EXAMPLES			
8	IMPORT DATA BETWEEN HDFS AND RDBMS USING APACHE SQOOP			
9	IMPORT DATA BETWEEN HDFS AND RDBMS USING INCREMENTAL IMPORT			

EX.NO:1

**DOWNLOADING AND INSTALLING HADOOP;
UNDERSTANDING DIFFERENT HADOOP MODES, STARTUP
SCRIPTS, CONFIGURATION FILES**

AIM:

To downloading and installing Hadoop; Understanding different Hadoop modes. Startup scripts, Configuration files.

PROCEDURE:



1. Hardware Requirement

- * RAM — Min. 8GB, if you have SSD in your system then 4GB RAM would also work.
- * CPU — Min. Quad core, with at least 1.80GHz

2. JRE 1.8 — Offline installer for JRE

3. Java Development Kit — 1.8

4. A Software for Un-Zipping like 7Zip or Win Rar

5. Download Hadoop zip



2. Unzip and Install Hadoop

After Downloading the Hadoop, we need to Unzip the hadoop-3.5.5.tar.gz file.

Now we can organize our Hadoop installation, we can create a folder and move the final extracted file in it.

Please note while creating folders, DO NOT ADD SPACES IN BETWEEN THE FOLDER NAME

3. Setting Up Environment Variables

Another important step in setting up a work environment is to set your Systems environment variable.

To edit environment variables, go to Control Panel > System > click on the “Advanced system settings” link

Alternatively, We can Right click on This PC icon and click on Properties and click on the “Advanced system settings” link Or, easiest way is to search for Environment Variable in search bar and there you go

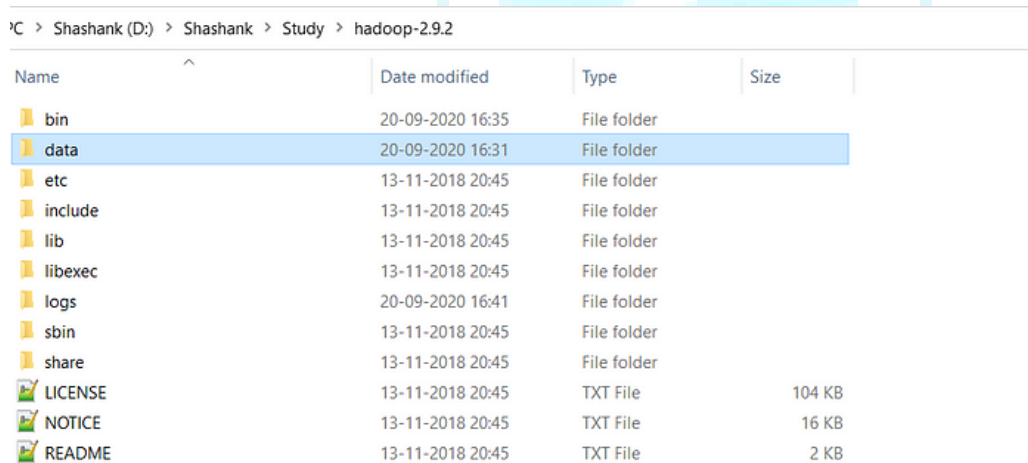
INSTALLATION PROCEDURE:

Editing Hadoop files

Once we have configured the environment variables next step is to configure Hadoop. It has 3 parts:-

Creating Folders

We need to create a folder data in the hadoop directory, and 2 sub folders namenode and datanode

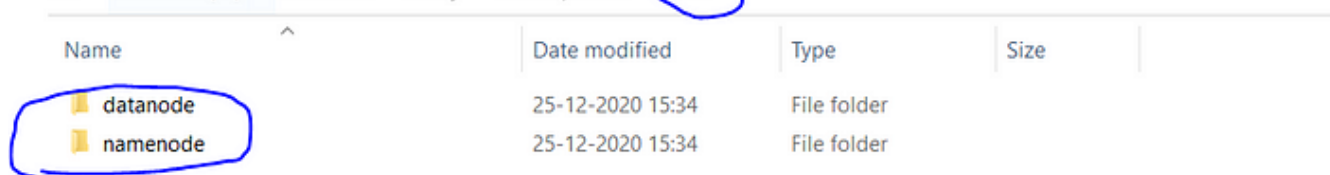


Name	Date modified	Type	Size
bin	20-09-2020 16:35	File folder	
data	20-09-2020 16:31	File folder	
etc	13-11-2018 20:45	File folder	
include	13-11-2018 20:45	File folder	
lib	13-11-2018 20:45	File folder	
libexec	13-11-2018 20:45	File folder	
logs	20-09-2020 16:41	File folder	
sbin	13-11-2018 20:45	File folder	
share	13-11-2018 20:45	File folder	
LICENSE	13-11-2018 20:45	TXT File	104 KB
NOTICE	13-11-2018 20:45	TXT File	16 KB
README	13-11-2018 20:45	TXT File	2 KB

Creating Data Folder

- Create **DATA folder** in the Hadoop directory

PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2 > data



Name	Date modified	Type	Size
datanode	25-12-2020 15:34	File folder	
namenode	25-12-2020 15:34	File folder	

Creating Sub-folders

- Once DATA folder is created, we need to create 2 new folders namely, **namenode and datanode** inside the data folder
- These folders are important because files on HDFS resides inside the datanode.

Editing Configuration Files

Now we need to edit the following config files in hadoop for configuring it :-

(We can find these files in Hadoop -> etc -> hadoop)

- * core-site.xml
- * hdfs-site.xml
- * mapred-site.xml
- * yarn-site.xml
- * hadoop-env.cmd

Editing core-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Editing hdfs-site.xml

Right click on the file, select edit and paste the following content within <configuration></configuration>tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

Also replace PATH~1 and PATH~2 with the path of namenode and datanode folder that we created recently(step 4.1).

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>PATH~1\namenode</value>
  <final>true</final>
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>PATH~2\datanode</value>
  <final>true</final>
</property>
</configuration>
```

Editing mapred-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Editing yarn-site.xml


```

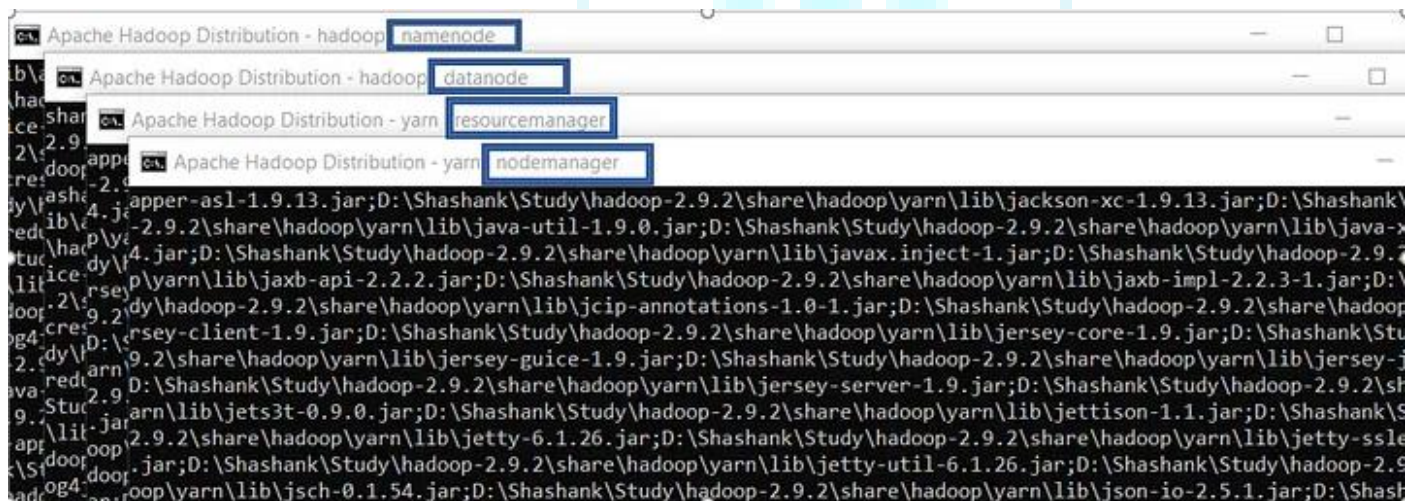
C:\Users\shash>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
starting yarn daemons
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\shash>

```

This will open 4 new cmd windows running 4 different Daemons of hadoop:-

- * Namenode
- * Datanode
- * Resourcemanager
- * Nodemanager



OUTPUT:

RESULT

EX.NO:2

HADOOP IMPLEMENTATION OF FILE MANAGEMENT TASKS

AIM: To implement file management tasks in Hadoop

- a) Adding Files and Directories
- b) Retrieving files
- c) Deleting Files

DESCRIPTION :

Hadoop provides a set of command line utilities that work similarly to the Linux file commands, and serve as your primary interface with HDFS.

HDFS is a scalable distributed filesystem designed to scale to petabytes of data while running on top of the underlying filesystem of the operating system. HDFS keeps track of where the data resides in a network by associating the name of its rack (or network switch) with the dataset. This allows Hadoop to efficiently schedule tasks to those nodes that contain data, or which are nearest to it, optimizing bandwidth utilization. Interacting with HDFS can be done from the command line.

Most common file management tasks in Hadoop are

- Adding files and directories to HDFS
- Retrieving files from HDFS to local filesystem
- Deleting files from HDFS

ALGORITHM:-

SYNTAX AND COMMANDS TO ADD, RETRIEVE AND DELETE DATA FROM HDFS

Step-1: Adding Files and Directories to HDFS

Before you can run Hadoop programs on data stored in HDFS, you'll need to put the data into HDFS first. Let's create a directory and put a file in it. HDFS has a default working directory of /user/\$USER, where \$USER is your login user name. This directory isn't automatically created for you, though, so let's create it with the mkdir command. For the purpose of illustration, we use chuck. You should substitute your user name in the example commands.

```
hadoop fs -mkdir /user/chuck
```

```
hadoop fs -put example.txt
```

```
hadoop fs -put example.txt /user/chuck
```


EX.NO:3

IMPLEMENTATION OF MATRIX MULTIPLICATION WITH HADOOP MAP REDUCE

AIM:

To implement Matrix Multiplication using Hadoop Map Reduce.

$$\begin{matrix} \mathbf{A} & \mathbf{B} & & \mathbf{A} * \mathbf{B} \\ \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} & \begin{pmatrix} 6 & 3 \\ 5 & 2 \\ 4 & 1 \end{pmatrix} & = & \begin{pmatrix} 1*6 + 2*5 + 3*4 & 1*3 + 2*2 + 3*1 \\ 4*6 + 5*5 + 6*4 & 4*3 + 5*2 + 6*1 \end{pmatrix} \end{matrix}$$

input output

reduce()

ALGORITHM:

Map Reduce paradigm is the soul of distributed parallel processing in Big Data.

Requirements

You need Hadoop's HDFS and map-reduce framework to test the program. I am using Cloudera's distribution Hadoop as it's really great for learning Hadoop without facing any complexity of installation. You can use any distribution of Hadoop as HDFS and map reduce works essentially the same.

Prepare HDFS

Before writing the code let's first create matrices and put them in HDFS.

- Create two files M1, M2 and put the matrix values. (separate columns with spaces and rows with a line break)

For this example I am taking matrices as:

```
1 2 3    7 8
4 5 6    9 10
        11 12
```

- Put the above files to HDFS at location `/user/cloudera/matrices/`

```
hdfs dfs -mkdir /user/cloudera/matrice
```

```
hdfs dfs -put /path/to/M1 /user/cloudera/matrices/
```

```
hdfs dfs -put /path/to/M2 /user/cloudera/matrices/
```

Procedure

We need to create two programs Mapper and Reduces

Mapper.py

- First, define the dimensions of the matrices (m,n)

```
#!/usr/lib/python
```

```
import sys
```

```
m_r=2
```

```
m_c=3
```

```
n_r=3
```

```
n_c=2
```

```
i=0
```

Read each line i.e a row from stdin and split then to separate elements. Map in to each element as we read elements as string from stdin.

```
for line in sys.stdin:el=map(int,line.split())
```

The mapper will first read the first matrix and then the second. To differentiate them we can keep a **count i** of the **line number** we are reading and the first **m_r** lines will belong to the **first matrix**.

```
if(i<m_r):
```

```
    for j in range(len(el)):
```

```
        for k in range(n_c): print "{0}\t{1}\t{2}\t{3}".format(i, k, j, el[j])
```

```
else:
```

```
    for j in range(len(el)):
```

```
        for k in range(m_r): print "{0}\t{1}\t{2}\t{3}".format(k, j, i-m_r, el[j])
```

```
i=i+1
```

Now comes the crucial part, **printing the key value**. We need to think of a key which will group elements that need to be **multiplied**, elements that need to be **summed** and elements that belong to the **same row**.

{0} {1} {2} are the part of key and {3} is the value.

To understand how a key is assigned, let's refer to the below image.

A **B** **A * B**

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \begin{pmatrix} 6 & 3 \\ 5 & 2 \\ 4 & 1 \end{pmatrix} = \begin{pmatrix} 1*6+2*5+3*4 & 1*3+2*2+3*1 \\ 4*6+5*5+6*4 & 4*3+5*2+6*1 \end{pmatrix}$$

{0} {1} {2} actually represents the position of element from A or B to A*B

- {0} is the row position of the element
- {1} is the column position of the element

- $\{2\}$ is the position of the element in addition. (like 1, 6 are at position 0 in addition and 2,5 are at position 1)

We can see that A's element is repeated B's number of column times i.e 2 and B's element is repeated A's number of row times i.e 2.

In the program,

- i is used to iterate through each row.
- j is used to iterate through each column.
- k is used to iterate through each duplicate produced.

For each element in matrix A:

Element remains in same row, therefore $\{0\}=i$

Element is duplicated and distributed to each column, therefore, column pos in $A*B =$ Duplication order of element i.e $\{1\}=k$

As you can see in the picture, the position of the element, in addition, is the same as it's column's number therefore $\{2\}=j$

For each element in matrix B:

Elements remain in the same column, therefore $\{1\}=j$

Element is duplicated and distributed to each row, therefore, row pos in $A*B =$ Duplication order of element i.e $\{0\}=k$

As you can see in the picture, the position of the element, in addition, is the same as it's row's position therefore $\{2\}=i-m_r$



OUTPUT:

Output of Mapper.py

```
[cloudera@quickstart matrix-mu]
0 0 0 1
0 1 0 1
0 0 1 2
0 1 1 2
0 0 2 3
0 1 2 3
1 0 0 4
1 1 0 4
1 0 1 5
1 1 1 5
1 0 2 6
1 1 2 6
0 0 0 7
1 0 0 7
0 1 0 8
1 1 0 8
0 0 1 9
1 0 1 9
0 1 1 10
1 1 1 10
0 0 2 11
1 0 2 11
0 1 2 12
1 1 2 12
```

output of mapper.py

If you will look closely you will realize that elements with the same key (first 3 numbers are key), will get multiplied. Elements with the same first two numbers of the key are part of the same sum and elements with same first num of key belong to the same row.

After mapper produces output, Hadoop will sort by key and provide it to reducer.py

Reducer.py

Our reducer program will get sorted mapper result which will look like this.

0	0	0	1
0	0	0	7
0	0	1	2
0	0	1	9
0	0	2	11
0	0	2	3
0	1	0	1
0	1	0	8
0	1	1	10
0	1	1	2
0	1	2	12
0	1	2	3
1	0	0	4
1	0	0	7
1	0	1	5
1	0	1	9
1	0	2	11
1	0	2	6
1	1	0	4
1	1	0	8
1	1	1	10
1	1	1	5
1	1	2	12
1	1	2	6

Input to reducer.py

If you look closely at the output and image of matrix multiplication, you will realize:

- Every 2 numbers need to be multiplied
- Every m_c multiplied results need to get summed
- Every n_c summed result belong to the same row
- There will be m_r number of rows

RESULT:

EX. No: 4

WORD COUNT MAP REDUCE PROGRAM USING MAP REDUCE PARADIGM

AIM:

To implement word count map reduce program using map reduce paradigm

ALGORITHM:

First Open Eclipse -> then select File -> New -> Java Project ->Name it WordCount -> then Finish.

Create Three Java Classes into the project. Name them WCDriver(having the main function), WCMapper, WCReducer.

You have to include two Reference Libraries for that:

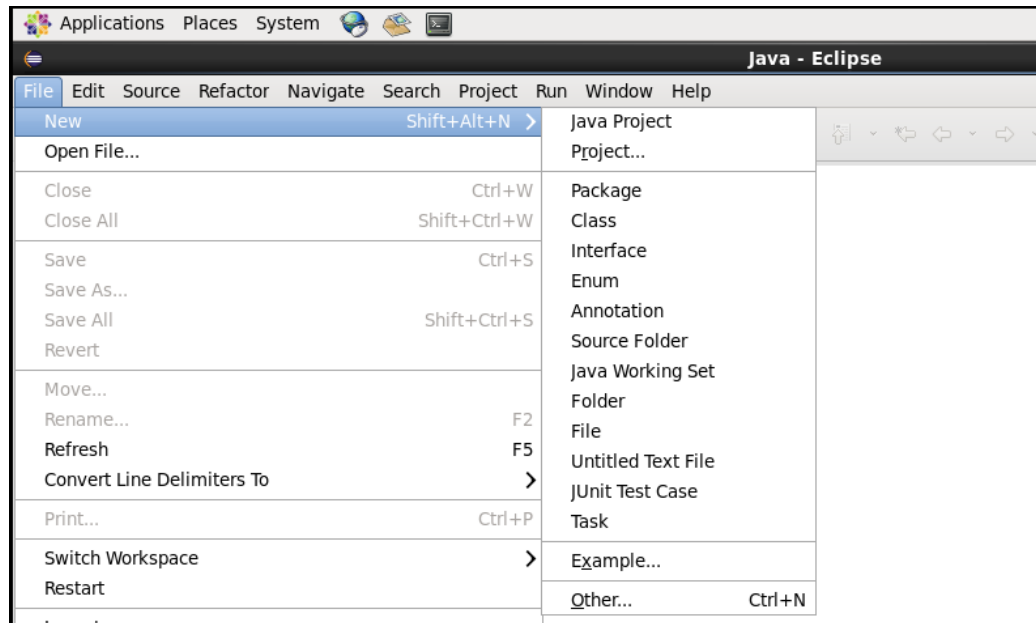
Right Click on Project -> then select Build Path-> Click on Configure Build Path

You can see the Add External JARs option on the Right Hand Side. Click on it and add the below mention files. You can find these files in */usr/lib/*

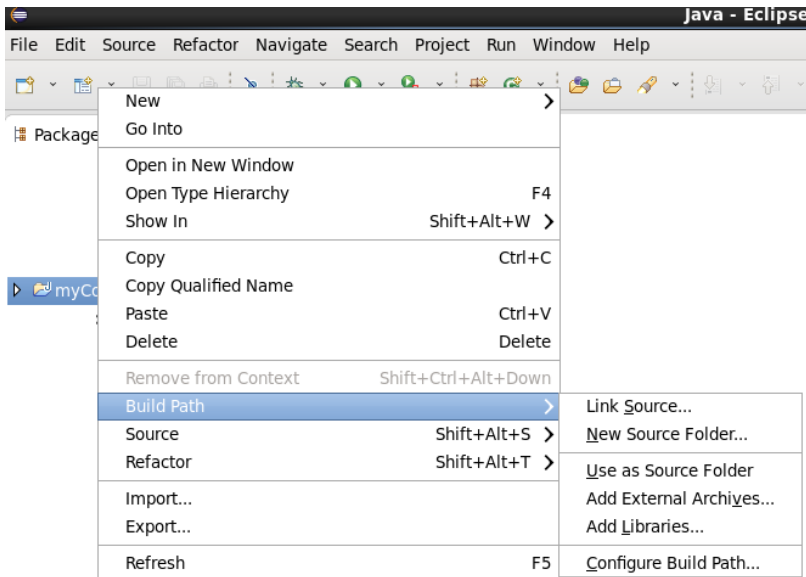
- 1. /usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.6.0-mr1-cdh5.13.0.jar**
- 2. /usr/lib/hadoop/hadoop-common-2.6.0-cdh5.13.0.jar**

STEPS:

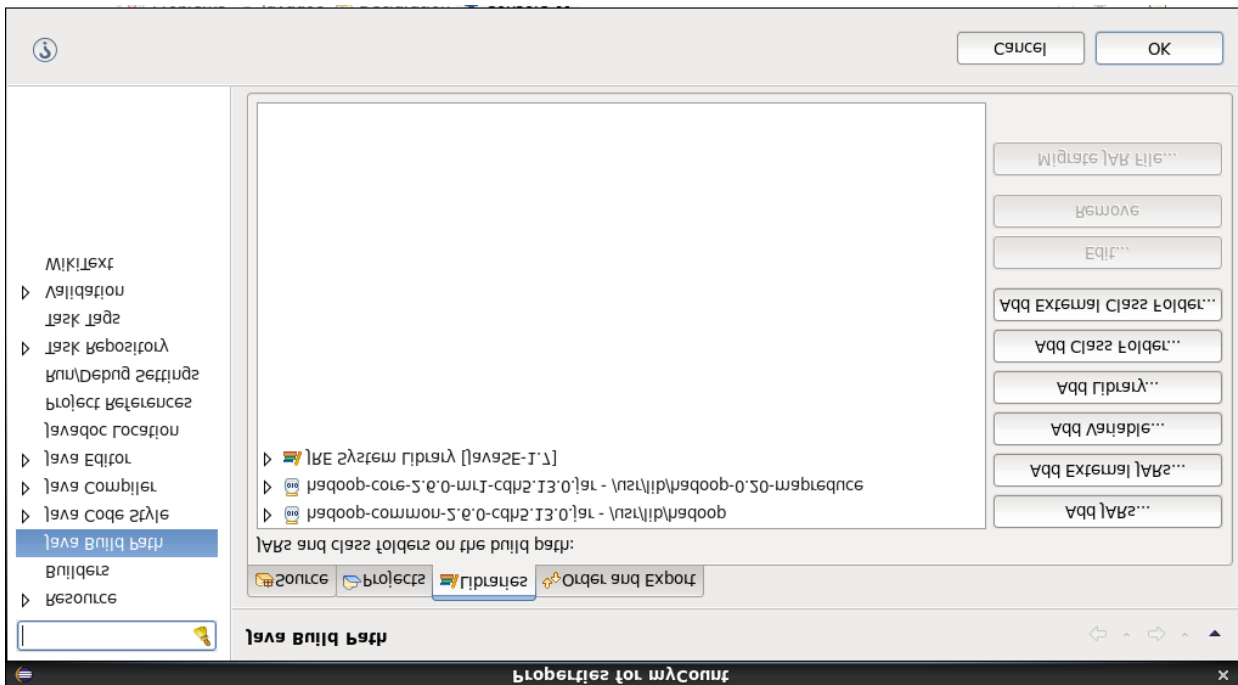
- **First Open Eclipse -> then select File -> New -> Java Project ->Name it WordCount -> then Finish.**



- Create Three Java Classes into the project. Name them **WCDriver**(having the main function), **WCMapper**, **WCReducer**.
- You have to include two Reference Libraries for that:
Right Click on **Project** -> then select **Build Path**-> Click on **Configure Build Path**



- In the above figure, you can see the Add External JARs option on the Right Hand Side. Click on it and add the below mention files. You can find these files in */usr/lib/*
 1. */usr/lib/hadoop-0.20-mapreduce/hadoop-core-2.6.0-mr1-cdh5.13.0.jar*
 2. */usr/lib/hadoop/hadoop-common-2.6.0-cdh5.13.0.jar*



Mapper Code: You have to copy paste this program into the WCMapper Java Class file.

```
// Importing libraries
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

public class WCMapper extends MapReduceBase implements Mapper<LongWritable,Text, Text, IntWritable> {

    // Map function
    public void map(LongWritable key, Text value, OutputCollector<Text,
                    IntWritable> output, Reporter rep) throws IOException
    {

        String line = value.toString();

        // Splitting the line on spaces
        for (String word : line.split(" "))
        {
            if (word.length() > 0)
            {
                output.collect(new Text(word), new IntWritable(1));
            }
        }
    }
}
```

Reducer Code: You have to copy paste this program into the WCReducer Java Class file.

```
// Importing libraries
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

public class WCReducer extends MapReduceBase implements Reducer<Text,IntWritable, Text, IntWritable> {

    // Reduce function
    public void reduce(Text key, Iterator<IntWritable> value,
                      OutputCollector<Text, IntWritable> output,
                      Reporter rep) throws IOException
    {
```

```

        int count = 0;

        // Counting the frequency of each words
        while (value.hasNext())
        {
            IntWritable i = value.next();
            count += i.get();
        }

        output.collect(key, new IntWritable(count));
    }
}

```

Driver Code: You have to copy paste this program into the WCDriver Java Class file.

```

// Importing libraries
import java.io.IOException;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;

public class WCDriver extends Configured implements Tool {

    public int run(String args[]) throws IOException
    {
        if (args.length < 2)
        {
            System.out.println("Please give valid inputs");
            return -1;
        }

        JobConf conf = new JobConf(WCDriver.class);
        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(WCMapper.class);
        conf.setReducerClass(WCReducer.class);
        conf.setMapOutputKeyClass(Text.class);
        conf.setMapOutputValueClass(IntWritable.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);
        JobClient.runJob(conf);
        return 0;
    }
}

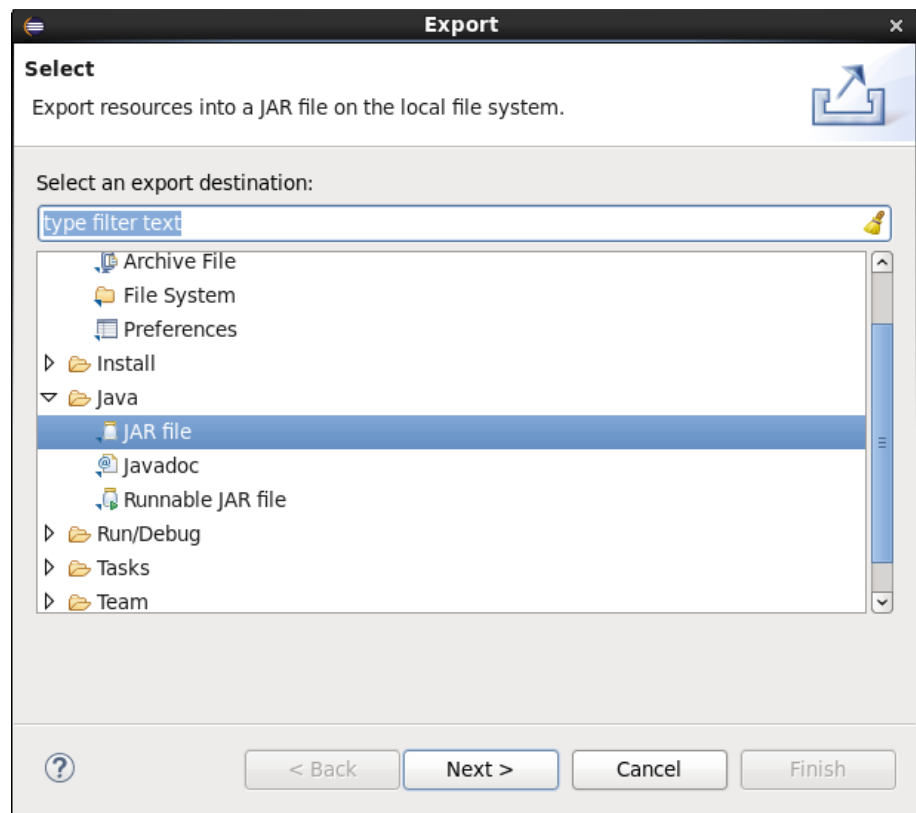
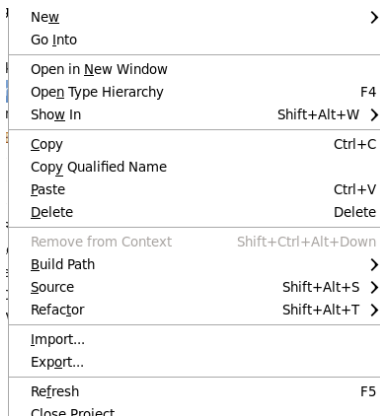
```

```

// Main Method
public static void main(String args[]) throws Exception
{
    int exitCode = ToolRunner.run(new WCDriver(), args);
    System.out.println(exitCode);
}
}

```

- Now you have to make a jar file. Right Click on **Project**-> **Click on Export**-> **Select export destination as Jar File**-> **Name the jar File**(WordCount.jar) -> **Click on next** -> at last **Click on Finish**. Now copy this file into the Workspace directory of Cloudera



Export generated class files and resources
 Export all output folders for checked projects
 Export Java source files and resources
 Export refactorings for checked projects. [Select refactorings...](#)

Select the export destination:

JAR file:

Options:

Compress the contents of the JAR file
 Add directory entries
 Overwrite existing files without warning

- Open the terminal on CDH and change the directory to the workspace. You can do this by using “cd workspace/” command. Now, Create a text file(**WCFile.txt**) and move it to HDFS. For that open terminal and write this code(remember you should be in the same directory as jar file you have created just now).

```
cloudera@quickstart:~/workspace
File Edit View Search Terminal Help
[cloudera@quickstart workspace]$ cat WCFile.txt
Hello I am GeeksforGeeks
Hello I am an Intern
[cloudera@quickstart workspace]$
```

- Now, run this command to copy the file input file into the HDFS.

OUTPUT:

RESULT:



EX. No: 5 INSTALLATION OF HIVE ALONG WITH PRACTICE EXAMPLES

AIM:

To Installation of Hive along with practice examples.

Prerequisites

There are some prerequisites to install hive on any machine:

1. Java Installation
2. Hadoop Installation

Step #1

- Verify Java is installed.
- Open the Terminal and Type the Command.

Java-Version

- If java is installed on the system, it will give you the version or else an error. In my case, Java is already installed and below is the output of the command.
- In case, Java is not installed in your system. You can visit the below link and download java and install it.
- [≡ 1880260.html](#).

Java Installation

1. Extract the downloaded.
2. Move it to “/usr/local”.

3. Set up PATH and JAVA_HOME variables.

Step #2

- Verify Hadoop is installed.
- Open the Terminal and type the command.

Hadoop-Version

- If Hadoop is already installed, this command will give you the version or else an error.
- In my case, Hadoop has already installed hence the below output.
- You can now observe I am working with a CDH5 machine.
- If Hadoop is not installed, Download the Hadoop from Apache software foundation.

Hadoop Installation

1. Setup Hadoop
2. Configure Hadoop

Files required to be edited to configure Hadoop are:

- core-site.xml
- hdfs-site.xml
- yarn-site.xml
- mapred-site.xml

3. Setup Namenode using the command:

```
Hdfs namenode -format
```

4. Start dfs using the following command:

```
start -dfs.sh
```

5. Start yarn using the command:

```
Start -yarn.sh
```

Install Hive:

Below the points in respect to Hive Installation

- The first thing we need to do is download the hive release which can be performed by clicking the link below: <https://apachemirror.wuchna.com/hive/>
- Above link will give the link from which you have to choose stable-2 highlighted below in yellow:
- After opening stable-2, choose the bin file (highlighted yellow in the screenshot) and right click and “copy link address”.

Steps to Install Hive

Below are the steps in Hive Installation:

Step 1: Download the tar file.

<http://apachemirror.wuchna.com/hive/stable-2/apache-hive-2.3.6-bin.tar.gz0>

Step 2: Extract the file.

```
sudo tar zxvf /Downloads/apache-hive-* -C /usr/local
```

Step 3: Move apache files to /usr/local/hive directory.

```
sudo mv /usr/local/apache-hive-* /usr/local/hive
```

Step 4: Set up the Hive environment by appending the following lines to ~/.bashrc file

```
export HIVE_HOME=/usr/local/hive
```

```
export PATH=$PATH:$HIVE_HOME/bin
```

```
export CLASSPATH=$CLASSPATH:/usr/local/Hadoop/lib/*:.
```

```
export CLASSPATH=$CLASSPATH:/usr/local/hive/lib/*:.
```

Step 5: Execute the bashrc file.

```
$ source ~/.bashrc
```

Step 6: Hive Configuration- Edit hive-env.sh file to append this:

```
export HADOOP_HOME=/usr/local/Hadoop
```

Step 7: Edit using the below commands:

```
$ cd $HIVE_HOME/conf
```

```
$ cp hive-env.sh.template hive-env.sh
```

- Now to verify the hive is installed or not, use command hive-version.
- Here, hive-version enters the hive shell, which means the hive is installed. However, in my case, it is the older version hence giving the warning.

Practice Example:

Let's create a simple Hive table and query it as a practice example. In the Hive shell, you can run the following commands:

```
CREATE TABLE employees (  
  id INT,  
  name STRING,  
  salary FLOAT  
);
```

```
INSERT INTO employees VALUES (1, 'John', 50000);
```

```
INSERT INTO employees VALUES (2, 'Jane', 60000);
```

```
INSERT INTO employees VALUES (3, 'Bob', 55000);
```

```
SELECT * FROM employees;
```

This will create a table called "employees," insert some sample data, and then retrieve all records from the table.

OUTPUT:

RESULT:

**EX:NO:6 INSTALLATION OF HBASE, INSTALLING THRIFT ALONG WITH
PRACTICE EXAMPLES**

AIM:

Installation of HBase, Installing thrift along with Practice examples

ALGORITHM:

Standalone mode installation (No dependency on Hadoop system)

- This is default mode of HBase
- It runs against local file system
- It doesn't use Hadoop HDFS
- Only HMaster daemon can run
- Not recommended for production environment
- Runs in single JVM

PROGRAM:

1. Connect to hbase
hbase shell
2. To list the tables in hbase
list
3. To check the hbase master and region server stopped or working
sudo service --status-all

sudo service hbase-master restart

sudo service hbase-regionserver restart
4. Exit from hbase and connect back
exit
5. create 'students', 'personal_details', 'contact_details', 'marks'
6. list
7. put 'students', 'student1', 'personal_details:name', 'kiruba'
8. put 'students', 'student1', 'personal_details:email', 'kirubamay20@gmail.com'
9. To see all records using scan

```
scan 'students'  
get 'students','student1'  
get 'students','student1', {column=> 'personal_details'}
```

10. Delete email id column for student1

```
delete 'students','students1','personal_details:email'  
scan 'students'  
describe 'students'  
exists 'students'
```

11. Drop a table

Drop is used to delete a hbase table. But this operator can't be applied directly to the table. Instead, the table is first disabled and then dropped.

```
disable 'students'  
drop 'students'
```

OUTPUT:



RESULT:

EX.NO: 7

HIVE EXECUTION WITH PRACTICE EXAMPLES

AIM:

To execute Hive Commands in Cloudera

PROCEDURE:

1.To enter hive shell

hive

2. To display databases/ list the databases

Show databases;

3. Create a new database

Create database employeedb;

4. use a database

Use employeedb;

5. create table employee (id bigint, name string, address string);

6. to list tables in a database

Show tables;

7. Display schema of a hive table

Describe customers;

8. Describe table with extra information

Describe formatted customers;

9. insert record into hive table

Insert into employee values (11,"Abi","Banglore");

10. insert multiple record into hive table:

Insert into employee values(

22,"Ane","Chennai"),(33,"vino","salem"),(44,"Diana","theni");

[Insert will trigger a mapreduce job in background]

23. To check the hive warehouse directory in terminal

```
Hadoop fs -ls /user/hive/warehouse
```

```
Hadoop fs -ls /user/hive/warehouse/employee.db
```

24. To check the folder structure of customers table in

employee database from terminal:

```
Hadoop fs -ls /user/hive/warehouse/employee.db/employee
```

25. read the contents of customers table from terminal

```
Hadoop fs -cat /user/hive/warehouse/employee.db/employee/*
```

26. Display table data with where condition:

```
Select * from employee where address = "Chennai";
```

```
Select name, address from employee where address = "chennai";
```

```
Select name, address from employee where address="Chennai" and id > 22;
```

27. to display distinct values

```
Select DISTINCT address from employee;
```

28. to display records with order by clause

```
Select name, address from employee order by address;
```

29. to display no. of records in a table

```
Select count(*) from employee;
```

30. to display records with group by clause

```
Select address, count(*) from employee group by address;
```

```
Select address, count(*) as employee_count from employee group by address;
```

31. display records using limit clause

Select * from employee limit 1;

32. to exit from hive shell

Exit

33. create a new hive table with <if not exist> statement

Create table if not exists menu_orders{

Id bigint, product_id string, customer_id bigint, quantity int, amount double);

34. insert a record into orders table

Insert into menu_orders values(11,"pho",11,3,120);

Insert into menu_orders values(12,"phon",12,4,130), (13,"phone",13,5,140);

OUTPUT:

RESULT:

EX.NO: 8 IMPORT DATA BETWEEN HDFS AND RDBMS USING APACHE SQOOP

AIM:

To Import data between HDFS and RDBMS using APACHE

PROCEDURE:

Sqoop Commands

Sqoop — IMPORT Command

Import command is used to importing a table from relational databases to HDFS. In our case, we are going to import tables from MySQL databases to HDFS.

Sqoop — IMPORT Command with target directory

You can also import the table in a specific directory in HDFS using the below command:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --m 1 --target-dir /employees
```

OUTPUT:

RESULT:

EX.NO: 9 IMPORT DATA BETWEEN HDFS AND RDBMS USING INCREMENTAL IMPORT

AIM:

To Import data between HDFS and RDBMS using Incremental Import in APACHE

PROCEDURE:

Sqoop Commands

Sqoop — IMPORT Command

Import command is used to importing a table from relational databases to HDFS. In our case, we are going to import tables from MySQL databases to HDFS.

Sqoop — IMPORT Command with target directory

You can also import the table in a specific directory in HDFS using the below command:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --m 1 --target-dir /employees
```

The command for incremental import is:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --target-dir /Latest_Employees --incremental append --check-column emp_no --last-value 499999
```

Sqoop — Import All Tables

You can import all the tables from the RDBMS database server to the HDFS. Each table data is stored in a separate directory and the directory name is same as the table name. It is mandatory

that every table in that database must have a primary key field. The command for importing all the table from a database is:

```
sqoop import-all-tables --connect jdbc:mysql://localhost/employees --username edureka
```

OUTPUT:



RESULT:

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)
Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMMON TO: DEPARTMENT OF INFORMATION TECHNOLOGY



CCS335- CLOUD COMPUTING

R2021

LAB MANUAL

INDEX

S.NO.	NAME OF THE EXPERIMENT	PAGE NO	DATE OF EXPERIMENT	REMARKS
1.	INSTALL VIRTUALBOX/VMWARE WORKSTATION WITH DIFFERENT FLAVOURS OF LINUX OR WINDOWS OS ON TOP OF WINDOWS7 OR 8			
2.	INSTALL A C COMPILER IN THE VIRTUAL MACHINE AND EXECUTE A SAMPLE PROGRAM			
3.	INSTALL GOOGLE APP ENGINE CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA.			
4.	USE GAE LAUNCHER TO LAUNCH THE WEB APPLICATIONS			
5.	SIMULATE A CLOUD SCENARIO USING CLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENT IN CLOUDSIM			
6.	FILES TRANSFER FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE.			
7.	TO LAUNCH VIRTUAL MACHINE USING TRYSTACK			
8.	INSTALL HADOOP SINGLE NODE CLUSTER AND RUNSIMPLE APPLICATIONS LIKE WORDCOUNT			

EX.NO 1:

DATE :

INSTALL VIRTUALBOX/VMWARE WORKSTATION WITH DIFFERENT FLAVOURS OF LINUX OR WINDOWS OS ON TOP OF WINDOWS7 OR 8

Aim:

Find procedure to Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.

PROCEDURE TO INSTALL

Step 1- Download Link

Link for downloading the software is <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>. Download the software for windows. Good thing is that there is no signup process. Click and download begins. Software is around 541 MB.

Step 2- Download the installer file

It should probably be in the download folder by default, if you have not changed the settings in your browser. File name should be something like VMware-workstation-full-15.5.1-15018445.exe. This file name can change depending on the version of the software currently available for download. But for now, till the next version is available, they will all be VMware Workstation 15 Pro.

Step 3- Locate the downloaded installer file

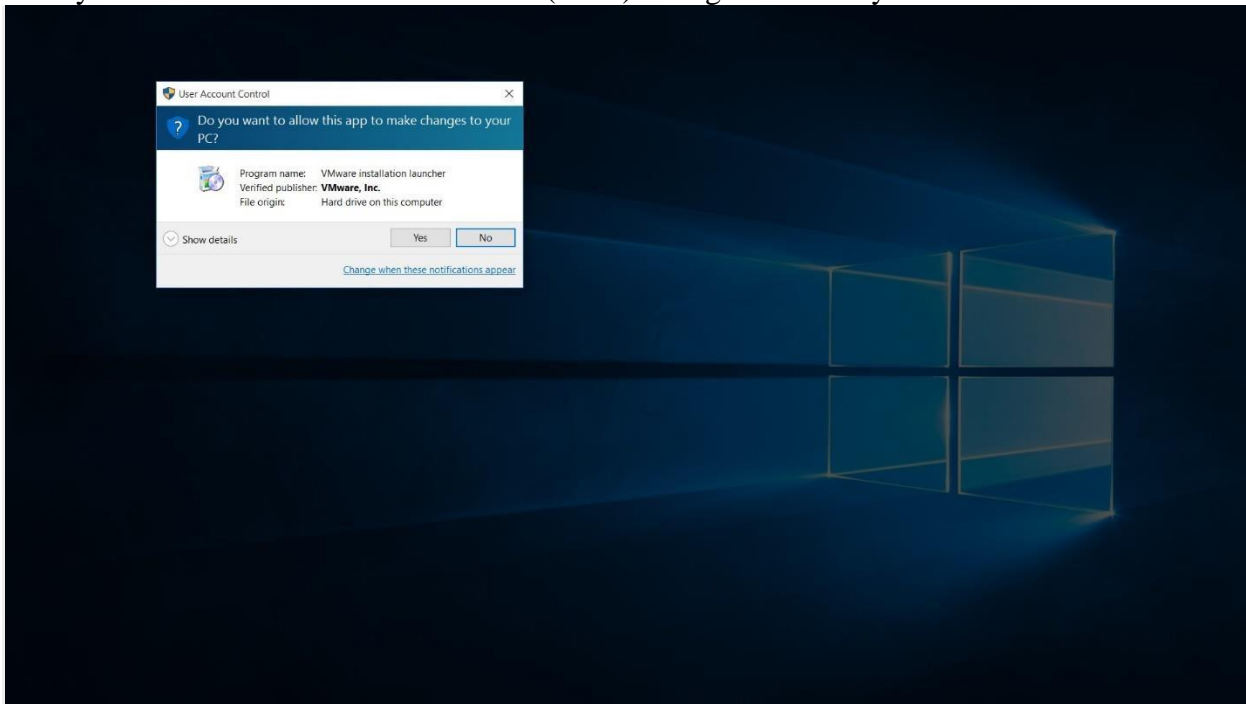
For demonstration purpose, I have placed the downloaded installer on my desktop. Find the installer on your system and double click to launch the application.



VMware workstation 15 pro for windows 10 installer file screenshot.

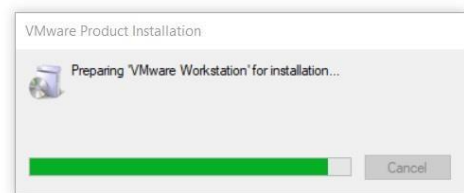
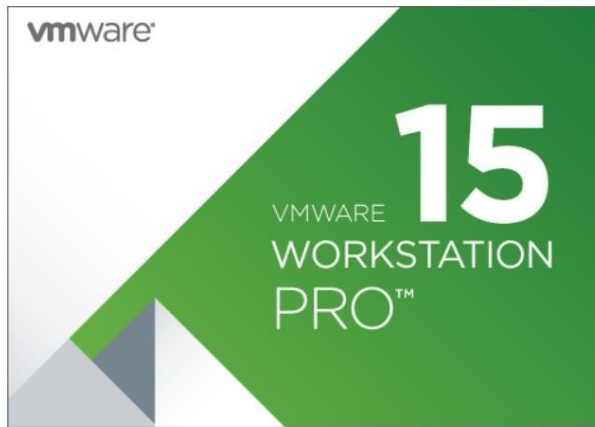
Step 4- User Access Control (UAC) Warning

Now you should see User Access Control (UAC) dialog box. Click yes to continue.



VMware Workstation 12 Pro installer windows 10 UAC screenshot

Initial Splash screen will appear. Wait for the process to complete.



VMware Workstation 15 Installation Splash Screen

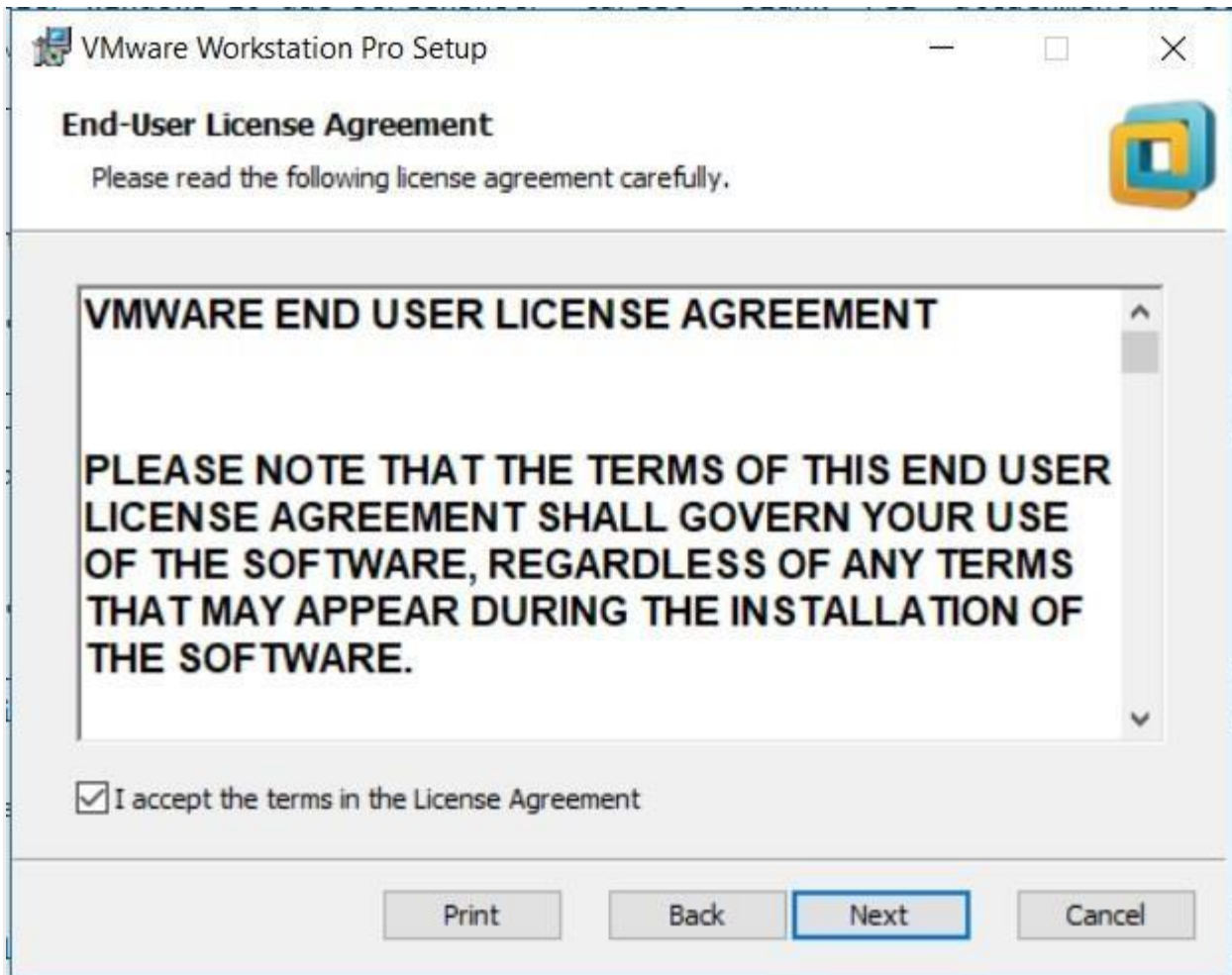
Step 5- VMware Workstation Setup wizard

Now you will see VMware Workstation setup wizard dialog box. Click next to continue.



Step 6- End User Licence Agreement

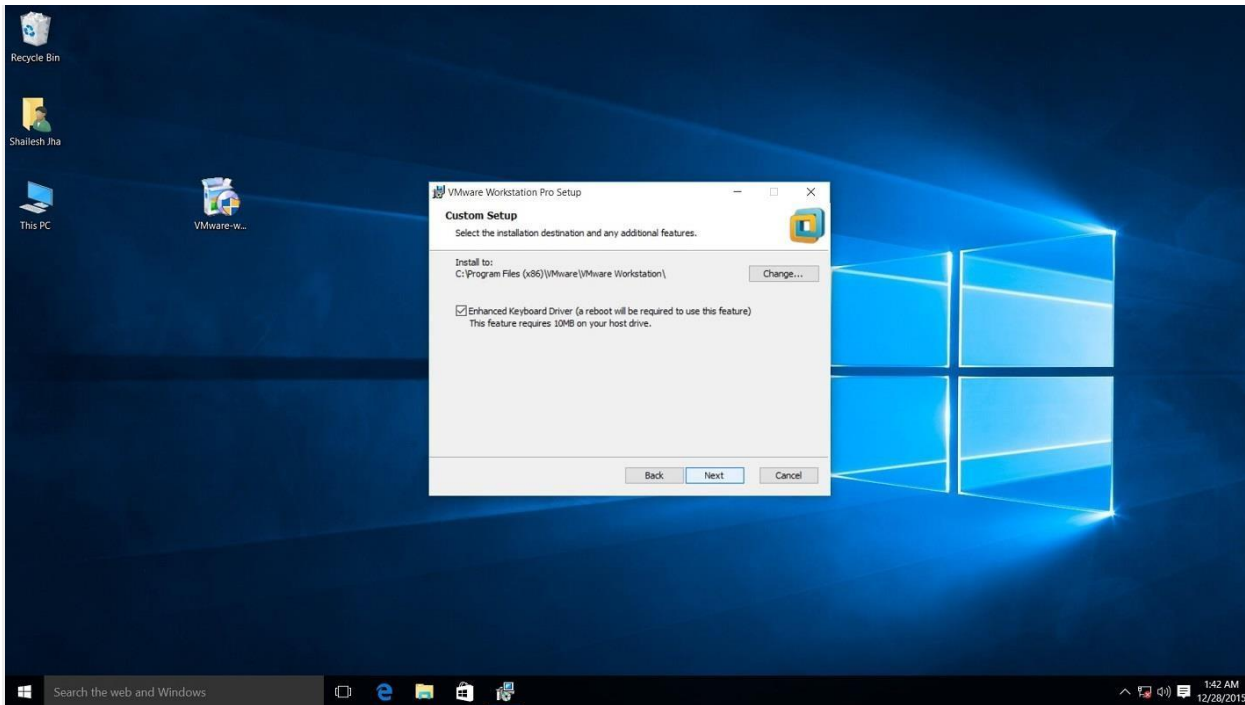
This time you should see End User Licence Agreement dialog box. Check "I accept the terms in the Licence Agreement" box and press next to continue.



VMware Workstation 15 Installation – End User Licence Agreement

Step 7- Custom Setup options

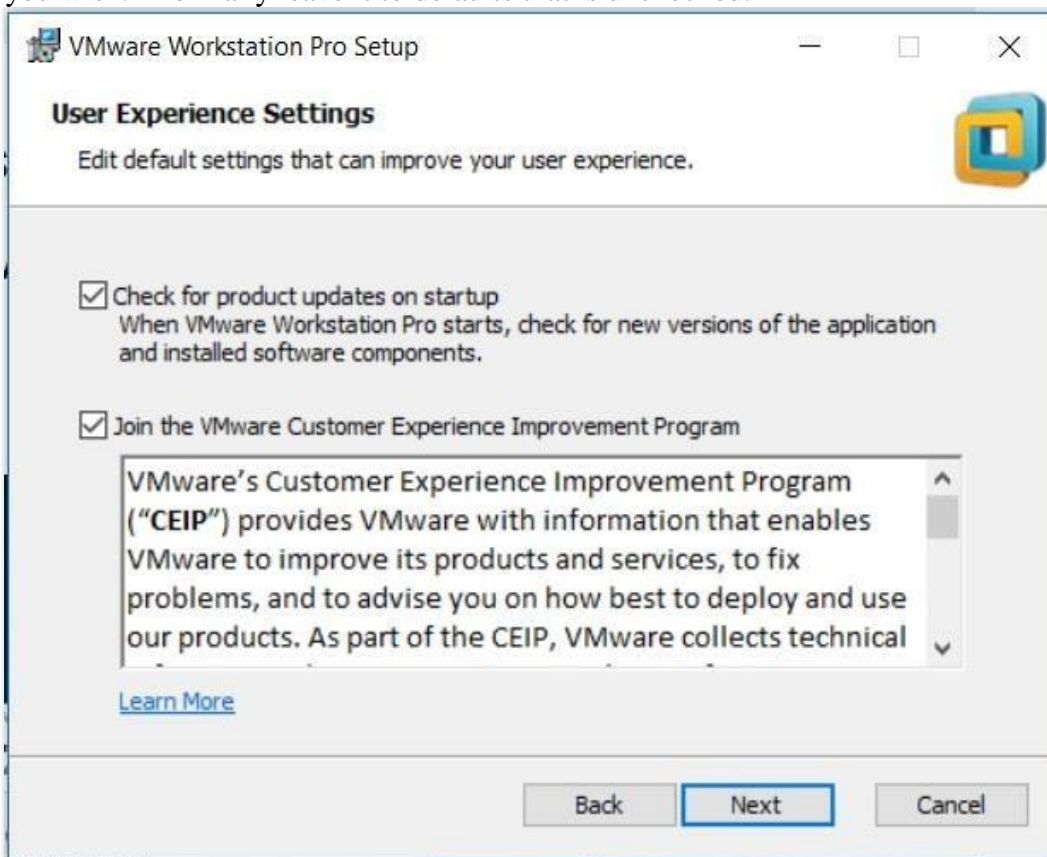
Select the folder in which you would like to install the application. There is no harm in leaving the defaults as it is. Also select Enhanced Keyboard Driver check box.



VMware Workstation 15 Pro installation – select installation folder

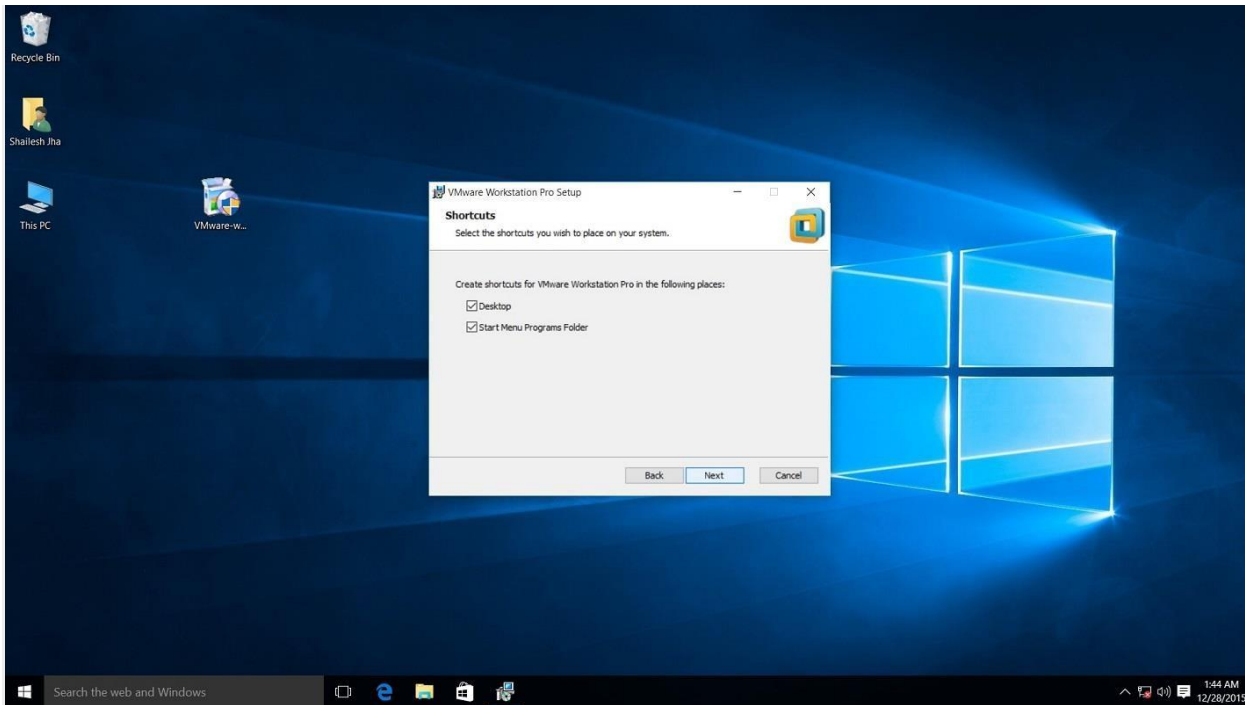
Step 8- User Experience Settings

Next you are asked to select “Check for Updates” and “Help improve VMware Workstation Pro”. Do as you wish. I normally leave it to defaults that is unchecked.



Step 9- Application Shortcuts preference

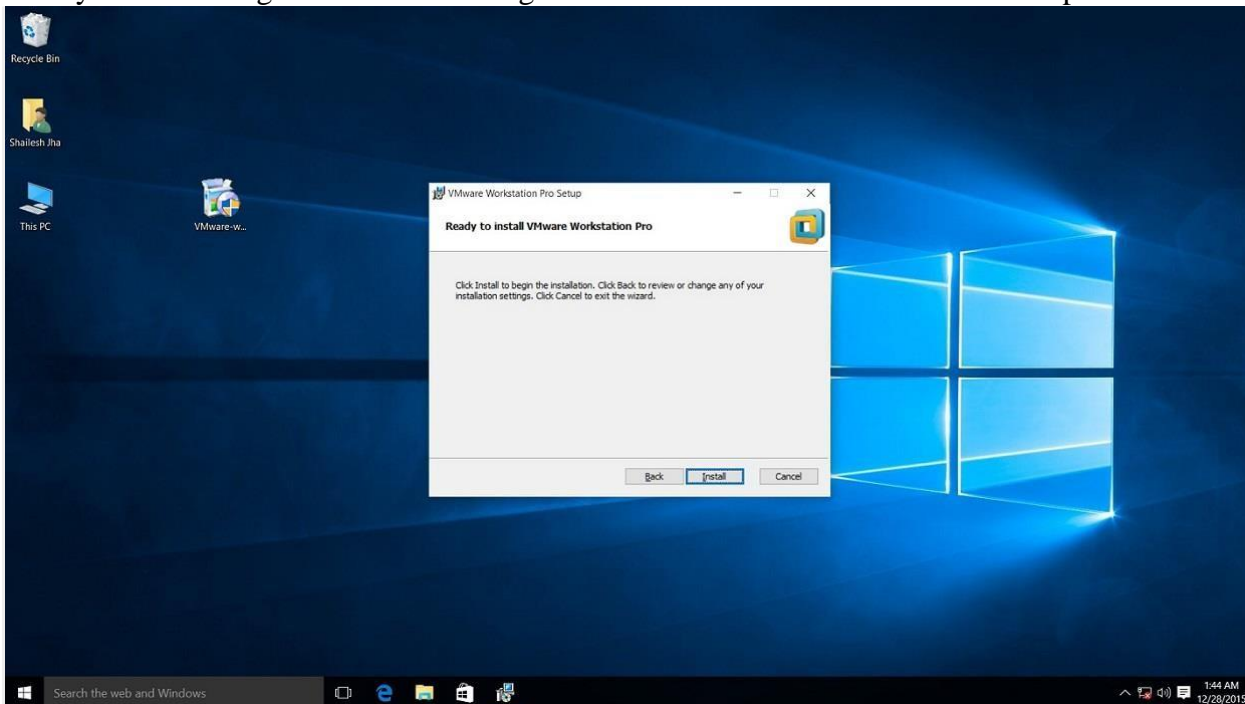
Next step is to select the place you want the shortcut icons to be placed on your system to launch the application. Please select both the options, desktop and start menu and click next.



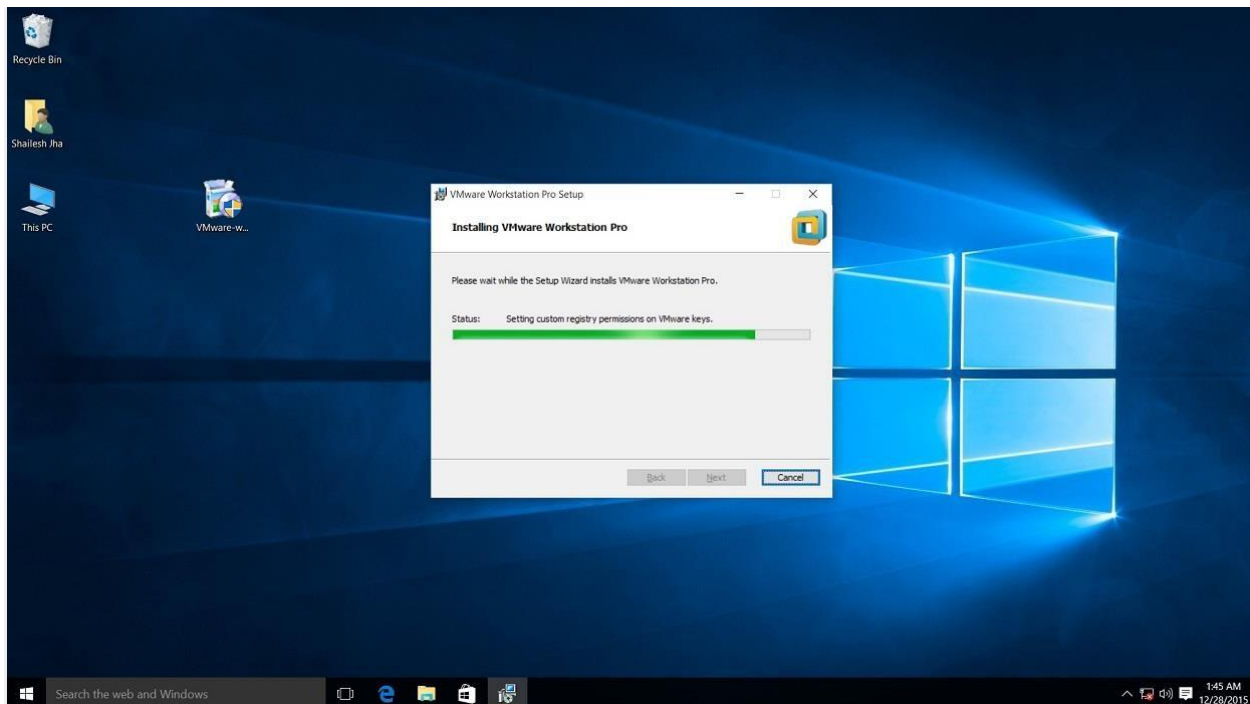
VMware workstation 15 pro installation shortcut selection checkbox screenshot.

Step 10- Installation begins

Now you see the begin installation dialog box. Click install to start the installation process.



Screenshot for VMware Workstation 15 pro installation begin confirmation dialog box on windows 10. Below screenshot shows Installation in progress. Wait for this to complete.

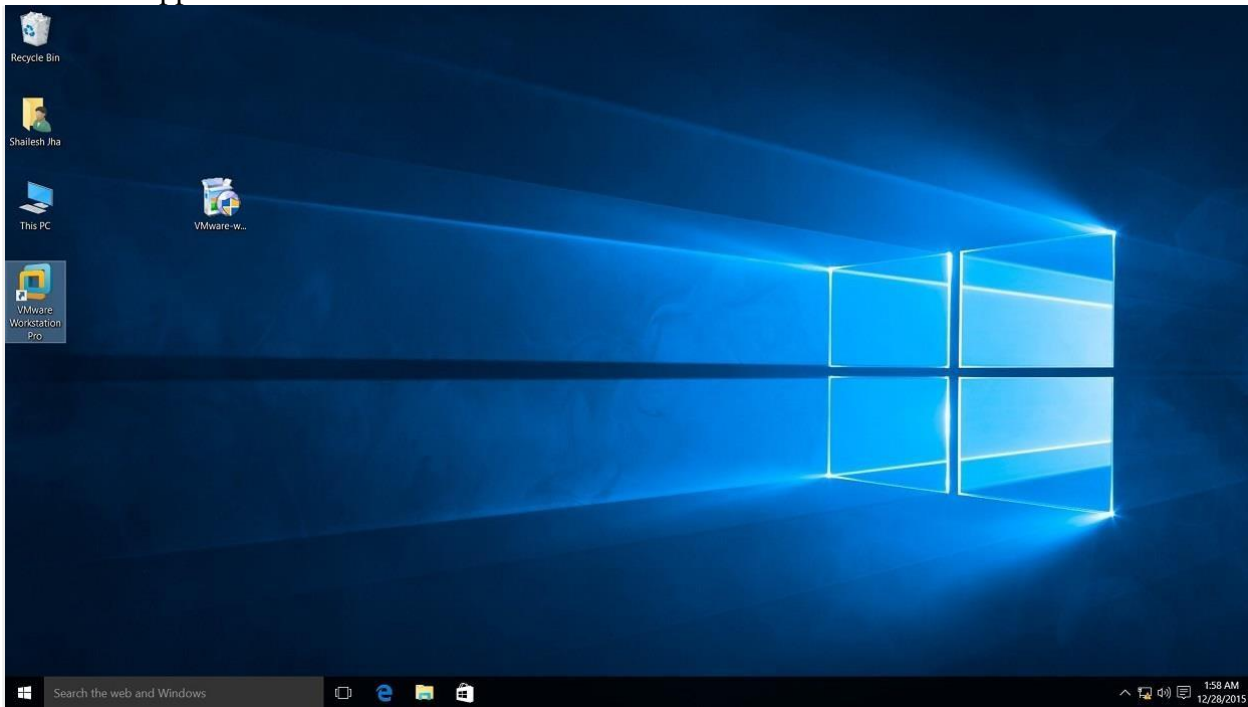


Screenshot for VMware Workstation 15 pro installation process. At the end you will see installation complete dialog box. Click finish and you are done with the installation process. You may be asked to restart your computer. Click on Yes to restart.



VMware Workstation 15 Installation – Installation Complete
Step 11- Launch VMware Workstation

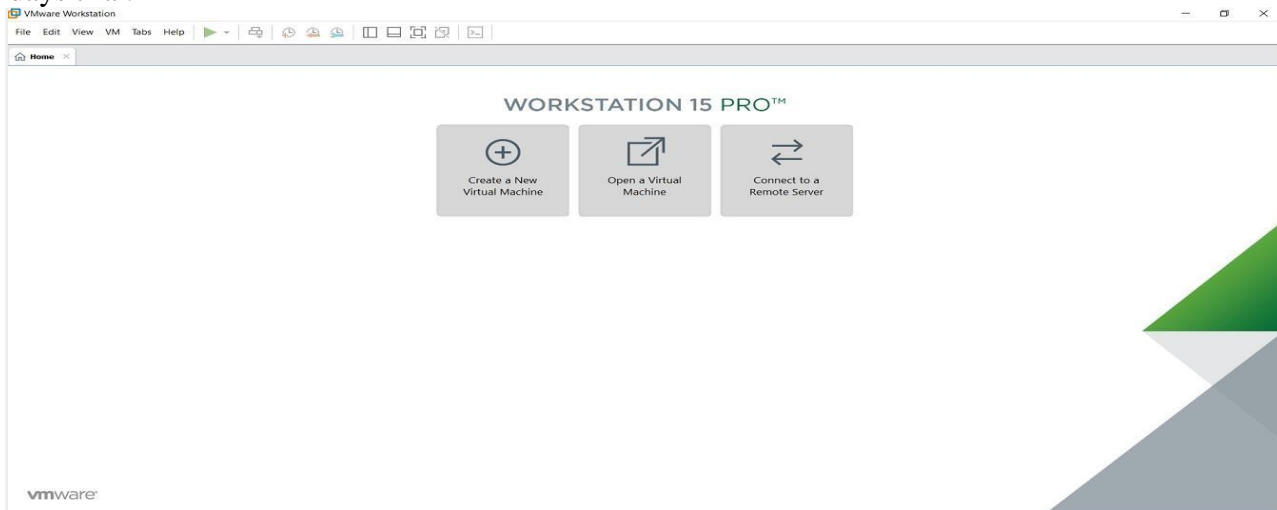
After the installation completes, you should see VMware Workstation icon on the desktop. Double click on it to launch the application.



Screenshot for VMware Workstation 15 Pro icon on windows 10 desktop.

Step 12- Licence Key

If you see the dialog box asking for licence key, click on trial or enter the licence key. Then what you have is the VMware Workstation 15 Pro running on your windows 10 desktop. If don't have the licencekey, you will have 30 days trial.



VMware Workstation 15 Pro home screen

Step 13- At some point if you decide to buy

At some point of time if you decide to buy the Licence key, you can enter the Licence key by going to **Help->Enter a Licence Key** You can enter the 25 character licence key in the dialog box shown below and click OK. Now you have the licence version of the software.

File Edit View VM Tabs Help

Solutions

Windows 10 20H2 x

Windows 10 20H2

- Power on this virtual machine
- Edit virtual machine settings

▼ Devices

Memory	4 GB
Processors	8
Hard Disk (NVMe)	30 GB
CD/DVD (SATA)	Using file C:\Use...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Display	Auto detect

▼ Description

Type here to enter a description of this virtual machine.

▼ Virtual Machine Details

State: Powered off
Configuration file: H:\Windows 10 20H2\Windows 10 20H2.vmx
Hardware compatibility: Workstation 16.x virtual machine
Primary IP address: Network information is not available

Windows 10 20H2 - VMware Workstation

File Edit View VM Tabs Help

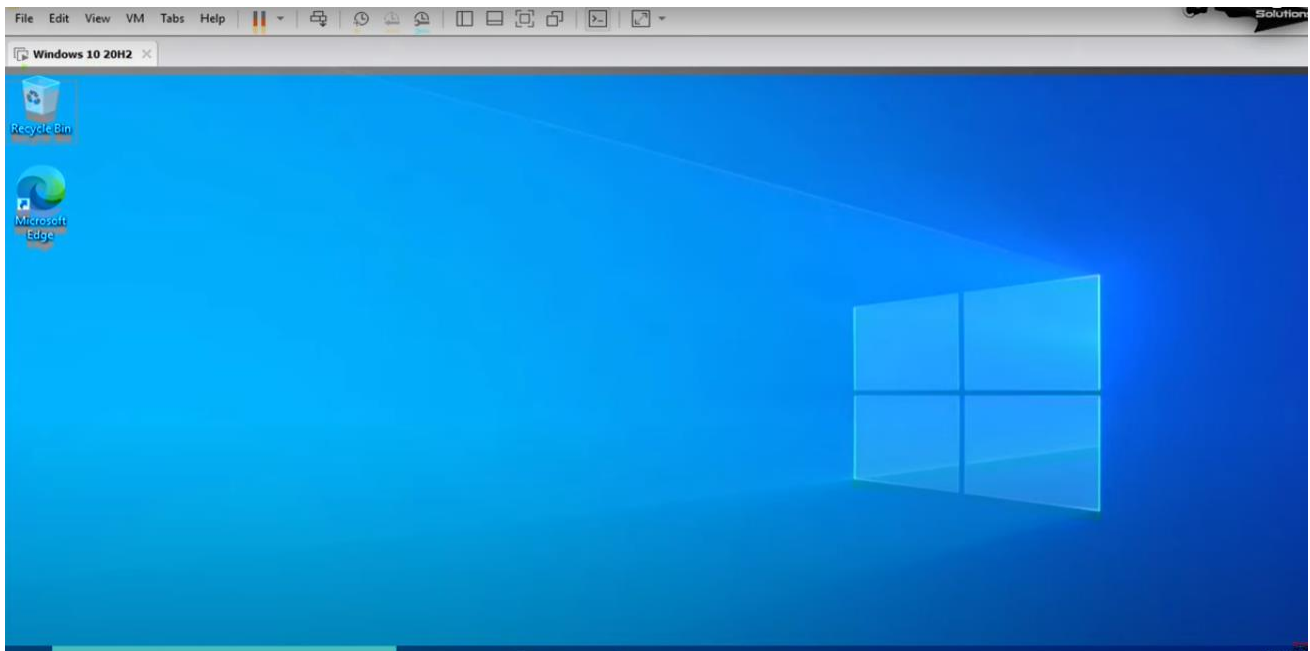
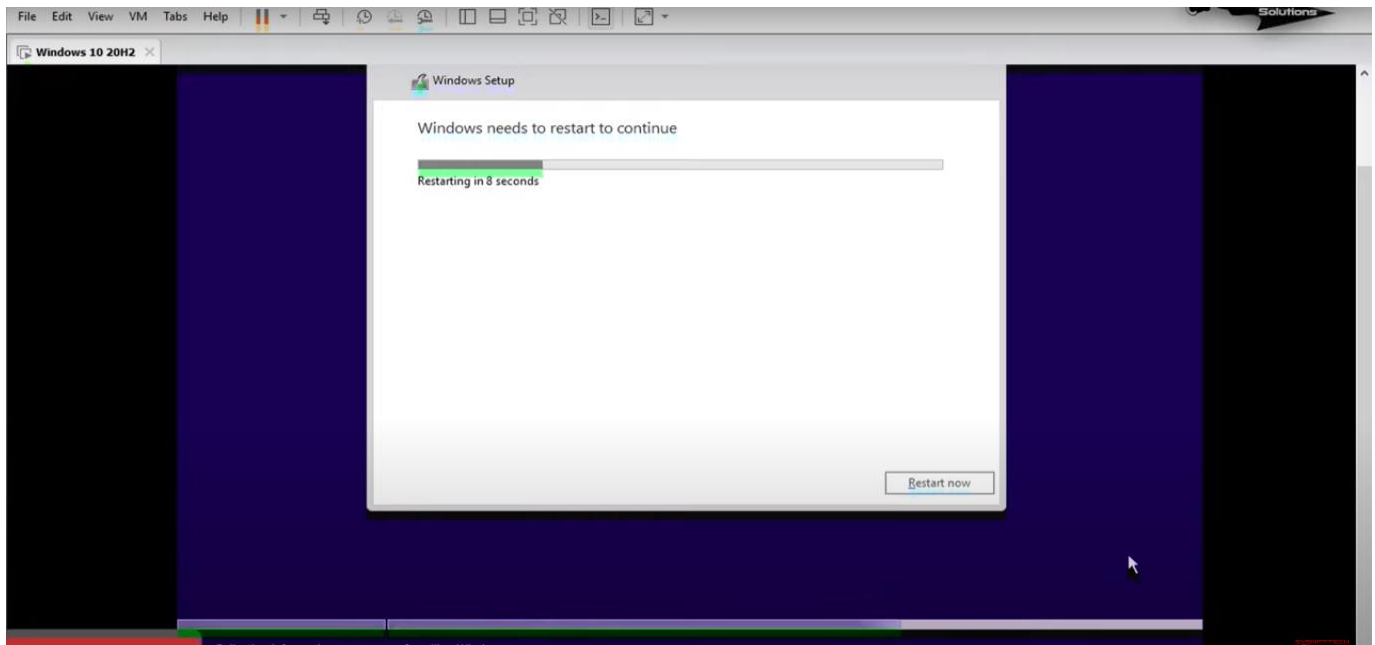
SYSNETTECH Solutions

Windows 10 20H2 x

Press any key to boot from CD or DVD.....

7x

Ctrl+Alt



RESULT:

EX.NO 2:

DATE :

INSTALL A C COMPILER IN THE VIRTUAL MACHINE AND EXECUTE A SAMPLE PROGRAM

AIM:

To Install a C compiler in the virtual machine and execute a sample C program.

PROCEDURE :

Step 1 : Open VMware software.

Step 2 : Create the virtual machine and startup the virtual machine.

Step 3: Install C compiler and type the following sample C program in Turbo c which is installed on virtual machine.

PROGRAM:

OUTPUT:

RESULT :

EX. NO 3:

DATE :

INSTALL GOOGLE APP ENGINE CREATE HELLO WORLD APP AND OTHER SIMPLE WEB APPLICATIONS USING PYTHON/JAVA.

AIM:

To create hello world app and other simple web applications using python/java.

ALGORITHM:

- 1. Install Google cloud SDK and python softwares.**
- 2. Authenticate the cloud SDK account by giving username and password.**
- 3. Create python and yaml files in you directory.**
- 4. Set the path in Google cloud shell and run the server.**
- 5. Type <http://localhost:8080> in the browser and the result will be displayed.**

PROGRAM:

OUTPUT:

RESULT :

EX. NO. 4

DATE:

USE GAE LAUNCHER TO LAUNCH THE WEB APPLICATIONS

AIM:

To launch GAE launcher to launch the web applications.

ALGORITHM:

1. Install Google App Engine Software.
2. Create the application name and open the directory that you have created.
3. Create the yaml and python files in your directory.
4. Run the Google App Engine to launch the application.

PROGRAM:

OUTPUT:

RESULT :

EX.NO 5:

DATE :

SIMULATE A CLOUD SCENARIO USING CLOUDSIM AND RUN A SCHEDULING ALGORITHM THAT IS NOT PRESENT IN CLOUDSIM

AIM : To simulate a cloud scenario using Cloud Sim and run a scheduling algorithm that is not present in Cloud Sim

PROCEDURE : The steps to be followed: How to use CloudSim in Eclipse
CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.

To use CloudSim in Eclipse:

1. Download CloudSim installable files from
<https://code.google.com/p/cloudsim/downloads/list> and unzip
2. Open Eclipse
3. Create a new Java Project: File -> New
4. Import an unpacked CloudSim project into the new Java Project
5. The first step is to initialize the CloudSim package by initializing the CloudSim library, as follows:
CloudSim.init(num_user, calendar, trace_flag)
6. Data centre's are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or space shared, the time zone and its price:
Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), s
7. The third step is to create a broker:
DatacenterBroker broker = createBroker();
8. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM's owner, mips, number of PEs amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:
VM vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())
9. Submit the VM list to the broker:
broker.submitVmList(vmlist)
10. Create a cloudlet with length, file size, output size, and utilisation model:
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel, utilizationMode
11. Submit the cloudlet list to the broker:

EX.NO 6:

DATE :

FILES TRANSFER FROM ONE VIRTUAL MACHINE TO ANOTHER VIRTUAL MACHINE.

AIM : To find a procedure to transfer the files from one virtual machine to another virtual machine.

PROCEDURE :

1. You can copy few (or more) lines with *copy & paste* mechanism.
For this you need to share clipboard between host OS and guest OS, installing **Guest Addition** on both the virtual machines (probably setting *bidirectional* and restarting them). You *copy* from *guest OS* in the clipboard that is shared with the *host OS*.
Then you *paste* from the *host OS* to the second *guest OS*.
2. You can enable **drag and drop** too with the same method (Click on the machine, settings, general, advanced, drag and drop: set to *bidirectional*)
3. You can have **common Shared Folders** on both virtual machines and use one of the directory shared as buffer to copy.
Installing **Guest Additions** you have the possibility to set Shared Folders too. As you put a file in a shared folder from *host OS* or from *guest OS*, is immediately visible to the other. (Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).
If you use the same folder shared on more machines you can exchange files directly copying them in this folder.
4. You can use **usual method to copy files between 2 different computer** with client-server application. (e.g. scp with sshd active for linux, winscp... you can get some info about SSH servers e.g. here)
You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).
Note: many Linux/Ubuntu distribution install sshd by default: you can see if it is running with `pgrep sshd` from a shell. You can install with `sudo apt-get install openssh-server`.
5. You can **mount part of the file system** of a virtual machine via NFS or SSHFS on the other, or you can **share file and directory** with Samba.
You may find interesting the article *Sharing files between guest and host without VirtualBox shared folders* with detailed step by step instructions.

You should remember that you are dialling with a little network of machines with different operative systems, and in particular:

- Each virtual machine has its own operative system running on and acts as a physical machine.
- Each virtual machine is an instance of a program *owned* by an *user* in the hosting operative system and should undergo the restrictions of the *user* in the *hosting OS*.
E.g Let us say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization). When each of them runs a virtual machine, for the hosting OS those virtual machines are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user. This is a restriction due to the *hosting OS*. It's easy to overcome it: it's enough to give authorization to read/write/execute to a directory or to choose a different directory in which both users can read/write/execute.
- Windows likes mouse and Linux fingers. :-)
I mean I suggest you to enable *Drag & drop* to be cosy with the Windows machines and the *Shared folders* or to be cosy with Linux.

When you will need to be fast with Linux **you will feel the need** of ssh-keygen an to Generate once SSH Keys to copy files on/from a remote machine without writing password anymore. In this way it functions \bash auto-completion remotely too!

PROGRAM:

OUTPUT:

RESULT :

EX.NO 7:

DATE :

TO LAUNCH VIRTUAL MACHINE USING TRYSTACK

AIM : To find a procedure to launch virtual machine using trystack.

PROCEDURE :

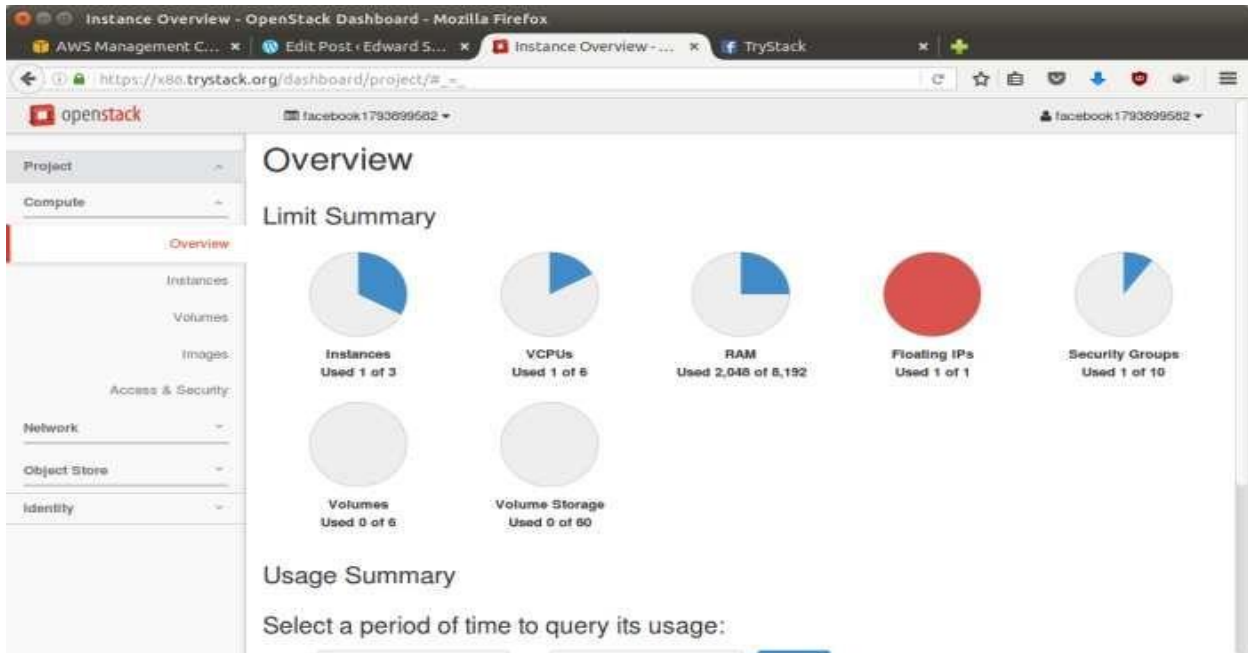
OpenStack is an open-source software cloud computing platform. OpenStack is primarily used for deploying an infrastructure as a service (IaaS) solution like Amazon Web Service (AWS). In other words, you can *make your own AWS* by using OpenStack. If you want to try out OpenStack, **TryStack** is the easiest and free way to do it.

In order to try OpenStack in TryStack, you must register yourself by joining **TryStack Facebook Group**. The acceptance of group needs a couple days because it's approved manually. After you have been accepted in the TryStack Group, you can log in TryStack.



TryStack.org Homepage

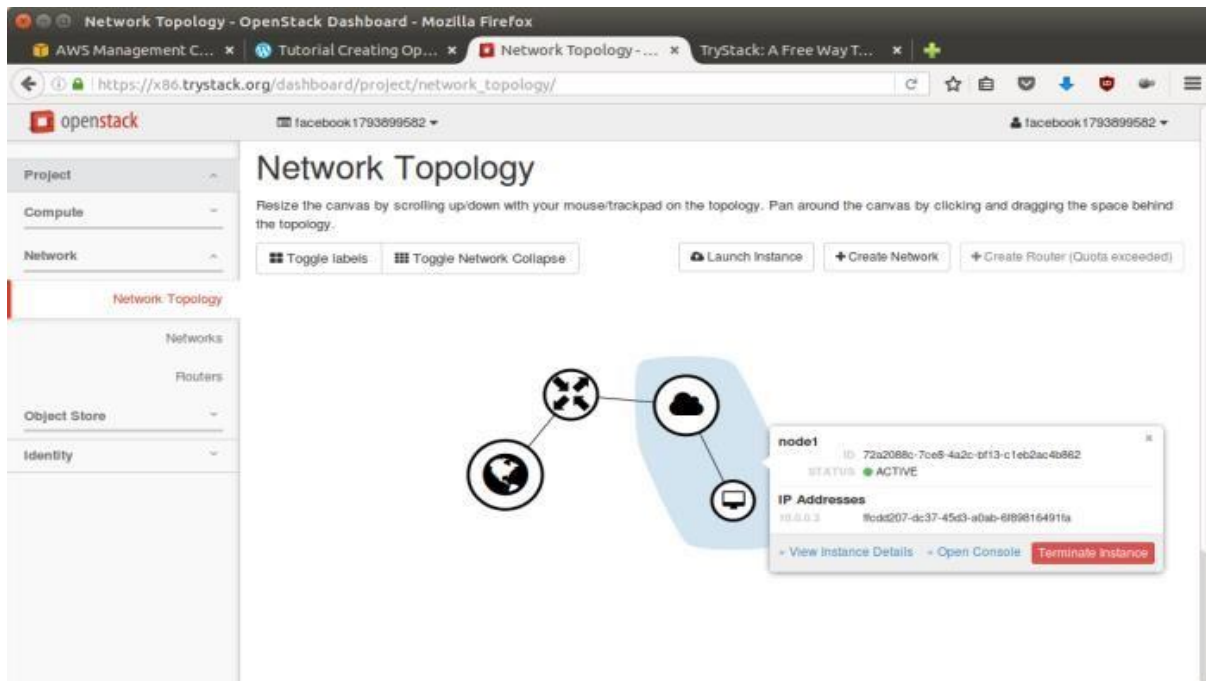
I assume that you already join to the Facebook Group and login to the dashboard. After you login to the TryStack, you will see the Compute Dashboard like:



OpenStack Compute Dashboard

Run an OpenStack instance.

The instance will be accessible through the internet (have a public IP address). The final topology will like:



Network topology

As you see from the image above, the instance will be connected to a local network and the local network will be connected to internet.

Step 1: Create Network

Network? Yes, the network in here is our own local network. So, your instances will be not mixed up with the others. You can imagine this as your own LAN (Local Area Network) in the cloud.

1. Go to **Network > Networks** and then click **Create Network**.
2. In **Network** tab, fill **Network Name** for example `internal` and then click **Next**.
3. In **Subnet** tab,
 1. Fill **Network Address** with appropriate CIDR, for example `192.168.1.0/24`. Use **privatenetwork CIDR block** as the best practice.
 2. Select **IP Version** with appropriate IP version, in this case `IPv4`.
 3. Click **Next**.
4. In **Subnet Details** tab, fill **DNS Name Servers** with `8.8.8.8` (Google DNS) and then click **Create**.

Step 2: Create Instance

Now, we will create an instance. The instance is a virtual machine in the cloud, like AWS EC2. You need the instance to connect to the network that we just created in the previous step.

1. Go to **Compute > Instances** and then click **Launch Instance**.
2. In **Details** tab,
 1. Fill **Instance Name**, for example `Ubuntu 1`.
 2. Select **Flavor**, for example `m1.medium`.
 3. Fill **Instance Count** with `1`.
 4. Select **Instance Boot Source** with **Boot from Image**.
 5. Select **Image Name** with **Ubuntu 14.04 amd64 (243.7 MB)** if you want install Ubuntu 14.04 in your virtual machine.
3. In **Access & Security** tab,
 1. Click **[+]** button of **Key Pair** to import key pair. This key pair is a public and private key that we will use to connect to the instance from our machine.
 2. In **Import Key Pair** dialog,
 1. Fill **Key Pair Name** with your machine name (for example `Edward-Key`).
 2. Fill **Public Key** with your **SSH public key** (usually is in `~/.ssh/id_rsa.pub`). See description in Import Key Pair dialog box for more information. If you are using Windows, you can use **Puttygen** to generate key pair.
 3. Click **Import key pair**.
 3. In **Security Groups**, mark/check **default**.
4. In **Networking** tab,
 1. In **Selected Networks**, select network that have been created in Step 1, for example `internal`.
5. Click **Launch**.
6. If you want to create multiple instances, you can repeat step 1-5. I created one more instance with instance name `Ubuntu 2`.

Step 3: Create Router In the step 1, we created our network, but it is isolated. It doesn't connect to the internet. To make our network has an internet connection, we need a router that running as the gateway to the internet.

1. Go to **Network > Routers** and then click **Create Router**.
2. Fill **Router Name** for example `router1` and then click **Create router**.
3. Click on your **router name link**, for example `router1`, **Router Details** page.
4. Click **Set Gateway** button in upper right:
 1. Select **External networks** with **external**.
 2. Then **OK**.
5. Click **Add Interface** button.
 1. Select **Subnet** with the network that you have been created in Step 1.
 2. Click **Add interface**.
6. Go to **Network > Network Topology**. You will see the network topology. In the example, there are two networks, i.e. external and internal, those are bridged by a router. There are instances those are joined to internal network.

Step 4: Configure Floating IP Address

Floating IP address is public IP address. It makes your instance accessible from the internet. When you launch your instance, the instance will have a private network IP, but no public IP. In OpenStack, the public IPs are collected in a pool and managed by admin (in our case is TryStack). You need to request a public (floating) IP address to be assigned to your instance.

1. Go to **Compute > Instance**.
2. In one of your instances, click **More > Associate Floating IP**.
3. In **IP Address**, click Plus [+].
4. Select **Pool** to **external** and then click **Allocate IP**.
5. Click **Associate**.
6. Now you will get a public IP, e.g. 8.21.28.120, for your instance.

Step 5: Configure Access & Security

OpenStack has a feature like a firewall. It can whitelist/blacklist your in/out connection. It is called *Security Group*.

1. Go to **Compute > Access & Security** and then open **Security Groups** tab.
2. In **default** row, click **Manage Rules**.
3. Click **Add Rule**, choose **ALL ICMP** rule to enable ping into your instance, and then click **Add**.
4. Click **Add Rule**, choose **HTTP** rule to open HTTP port (port 80), and then click **Add**.
5. Click **Add Rule**, choose **SSH** rule to open SSH port (port 22), and then click **Add**.
6. You can open other ports by creating new rules.

Step 6: SSH to Your Instance

Now, you can SSH your instances to the floating IP address that you got in the step 4. If you are using Ubuntu image, the SSH user will be `ubuntu`.

OUTPUT:

RESULT :

EX.NO 8:

DATE :

INSTALL HADOOP SINGLE NODE CLUSTER AND RUNSIMPLE APPLICATIONS LIKE WORDCOUNT

AIM :

To install hadoop single node cluster and run simple applications like wordcount.

PROCEDURE:

1. Hardware Requirement
 - * RAM — Min. 8GB, if you have SSD in your system then 4GB RAM would also work.
 - * CPU — Min. Quad core, with at least 1.80GHz
2. JRE 1.8 — Offline installer for JRE
3. Java Development Kit — 1.8
4. A Software for Un-Zipping like 7Zip or Win Rar
5. Download Hadoop zip

2. Unzip and Install Hadoop

After Downloading the Hadoop, we need to Unzip the hadoop-3.5.5.tar.gz file.

Now we can organize our Hadoop installation, we can create a folder and move the final extracted file in it.

Please note while creating folders, DO NOT ADD SPACES IN BETWEEN THE FOLDER NAME

3. Setting Up Environment Variables

Another important step in setting up a work environment is to set your Systems environment variable.

To edit environment variables, go to Control Panel > System > click on the “Advanced system settings” link

Alternatively, We can Right click on This PC icon and click on Properties and click on the “Advanced system settings” link Or, easiest way is to search for Environment Variable in search bar and there you go

INSTALLATION PROCEDURE:

Editing Hadoop files

Once we have configured the environment variables next step is to configure Hadoop. It has 3 parts:-

PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2

Name	Date modified	Type	Size
bin	20-09-2020 16:35	File folder	
data	20-09-2020 16:31	File folder	
etc	13-11-2018 20:45	File folder	
include	13-11-2018 20:45	File folder	
lib	13-11-2018 20:45	File folder	
libexec	13-11-2018 20:45	File folder	
logs	20-09-2020 16:41	File folder	
sbin	13-11-2018 20:45	File folder	
share	13-11-2018 20:45	File folder	
LICENSE	13-11-2018 20:45	TXT File	104 KB
NOTICE	13-11-2018 20:45	TXT File	16 KB
README	13-11-2018 20:45	TXT File	2 KB

Creating Folders

We need to create a folder data in the hadoop directory, and 2 sub folders namenode and datanode

Creating Data Folder

- Create **DATA folder** in the Hadoop directory

PC > Shashank (D:) > Shashank > Study > hadoop-2.9.2 > data

Name	Date modified	Type	Size
datanode	25-12-2020 15:34	File folder	
namenode	25-12-2020 15:34	File folder	

Creating Sub-folders

- Once DATA folder is created, we need to create 2 new folders namely, **namenode and datanode** inside the data folder
- These folders are important because files on HDFS resides inside the datanode.

Editing Configuration Files

Now we need to edit the following config files in hadoop for configuring it :-
(We can find these files in Hadoop -> etc -> hadoop)

- * core-site.xml
- * hdfs-site.xml
- * mapred-site.xml
- * yarn-site.xml
- * hadoop-env.cmd

Editing core-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>fs.defaultFS</name>
  <value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Editing hdfs-site.xml

Right click on the file, select edit and paste the following content within

```
<configuration></configuration>tags.
```

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

Also replace PATH~1 and PATH~2 with the path of namenode and datanode folder that we created recently(step 4.1).

```
<configuration>
<property>
  <name>dfs.replication</name>
  <value>1</value>
</property>
<property>
  <name>dfs.namenode.name.dir</name>
  <value>PATH~1\namenode</value>
  <final>true</final>
```

```
</property>
<property>
  <name>dfs.datanode.data.dir</name>
  <value>PATH~2\datanode</value>
  <final>true</final>
</property>
</configuration>
```

Editing mapred-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>mapreduce.framework.name</name>
  <value>yarn</value>
</property>
</configuration>
```

Editing yarn-site.xml

Right click on the file, select edit and paste the following content within <configuration> </configuration> tags.

Note:- Below part already has the configuration tag, we need to copy only the part inside it.

```
<configuration>
<property>
  <name>yarn.nodemanager.aux-services</name>
```

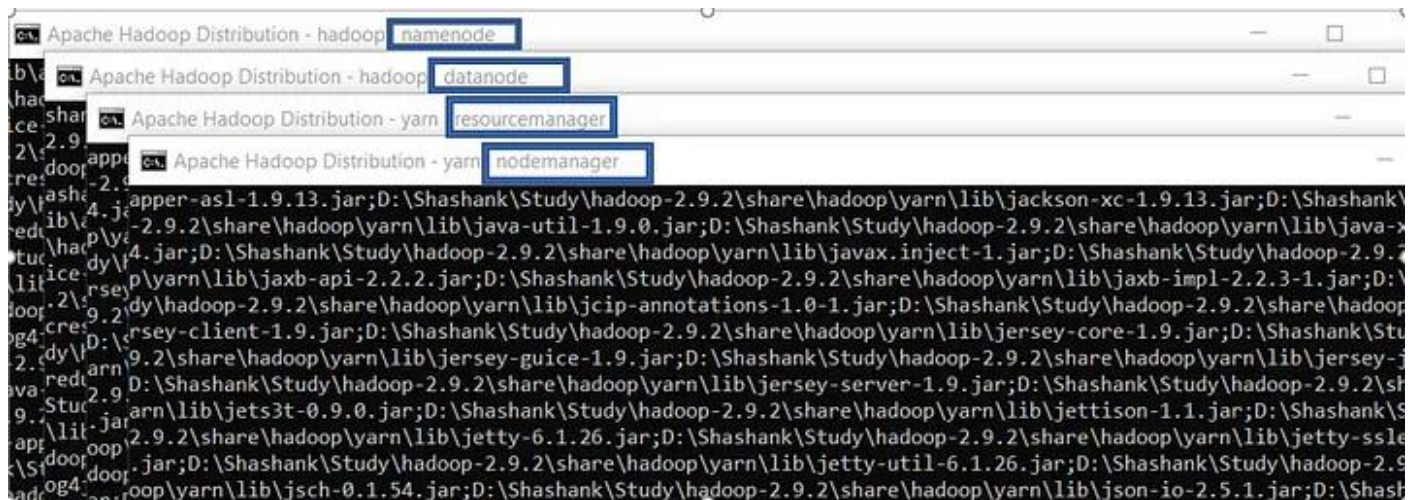


```
C:\Users\shash>start-all.cmd
This script is Deprecated. Instead use start-dfs.cmd and start-yarn.cmd
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.
starting yarn daemons
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\shash>
```

This will open 4 new cmd windows running 4 different Daemons of hadoop:-

- * Namenode
- * Datanode
- * Resourcemanager
- * Nodemanager



OUTPUT:

RESULT :

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)
Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

COMMON TO: DEPARTMENT OF INFORMATION TECHNOLOGY



**CS3591- COMPUTER NETWORKS
LABORATORY**

R2021

LAB MANUAL

INDEX

S.No.	Name of the Experiment	Page No	Date of Experiment	Remarks
1.	Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine			
2.	Write a HTTP web client program to download a web page using TCP sockets			
3.	Applications using TCP sockets like: Echo client and echo server, Chat and File Transfer			
4.	Simulation of DNS using UDP Sockets			
5.	Write a code simulating ARP RARP protocols			
6.	Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS			
7.	Study of TCP/UDP performance using Simulation tool			
8.	Simulation of Distance Vector/ Link State Routing algorithm			
9.	Performance Evaluation of Routing protocols using Simulation tool			
10.	Simulation of Error Detection Code (like CRC)			

EX.NO:1

DATE :

Learn to use commands like tcpdump, netstat, ifconfig, nslookup and traceroute. Capture ping and trace route PDUs using a network protocol analyzer and examine.

AIM:

To Learn to use commands like tcpdump, netstat, ifconfig, nslookup traceroute and ping.

PRE LAB DISCUSSION:

Tcpdump:

The tcpdump utility allows you to capture packets that flow within your network to assist in network troubleshooting. The following are several examples of using tcpdump with different options. Traffic is captured based on a specified filter.

Netstat

Netstat is a common command line TCP/IP networking available in most versions of Windows, Linux, UNIX and other operating systems.

Netstat provides information and statistics about protocols in use and current TCP/IP network connections.

ipconfig

ipconfig is a console application designed to run from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer.

From the command prompt, type **ipconfig** to run the utility with default options. The output of the default command contains the IP address, network mask, and gateway for all physical and virtual network adapter.

nslookup

The **nslookup** (which stands for *name server lookup*) command is a network utility program used to obtain information about internet servers. It finds name server information for domains by querying the Domain Name System.

Trace route:

Traceroute is a network diagnostic tool used to track the pathway taken by a packet on an IP network from source to destination. Traceroute also records the time taken for each hop the packet makes during its route to the destination

Commands:

Tcpdump:

Display traffic between 2 hosts:

To display all traffic between two hosts (represented by variables host1 and host2): # tcpdump host host1 and host2

Display traffic from a source or destination host only:

To display traffic from only a source (src) or destination (dst)

host: # tcpdump src host

tcpdump dst host

Display traffic for a specific protocol

Provide the protocol as an argument to display only traffic for a specific protocol, for example tcp, udp, icmp, arp

```
# tcpdump protocol
For example to display traffic only for the tcp traffic :
# tcpdump tcp
```

Filtering based on source or destination port

To filter based on a source or destination port:

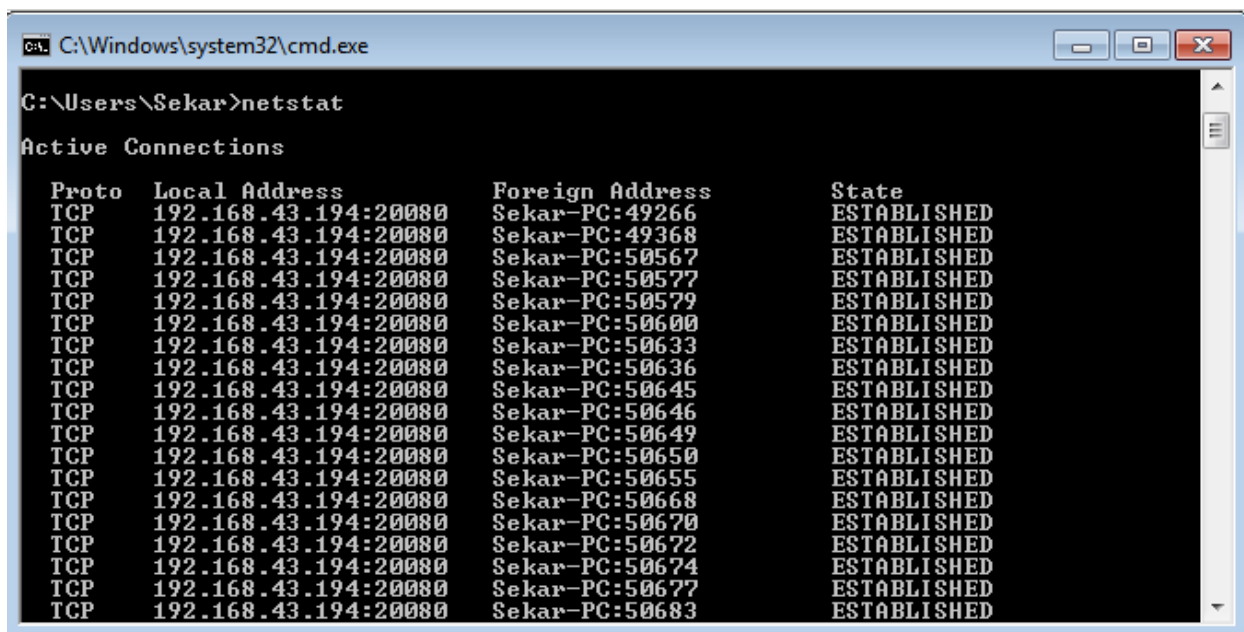
```
# tcpdump src port ftp
# tcpdump dst port
http
```

2. Netstat

Netstat is a common command line TCP/IP networking available in most versions of Windows, Linux, UNIX and other operating systems.

Netstat provides information and statistics about protocols in use and current TCP/IP network connections. The Windows help screen (analogous to a Linux or UNIX for netstat reads as follows: displays protocol statistics and current TCP/IP network connections.

```
#netstat
```



```
C:\Windows\system32\cmd.exe
C:\Users\Sekar>netstat
Active Connections
Proto Local Address Foreign Address State
TCP 192.168.43.194:20080 Sekar-PC:49266 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:49368 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50567 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50577 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50579 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50600 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50633 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50636 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50645 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50646 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50649 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50650 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50655 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50668 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50670 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50672 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50674 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50677 ESTABLISHED
TCP 192.168.43.194:20080 Sekar-PC:50683 ESTABLISHED
```

3. ipconfig

In Windows, **ipconfig** is a console application designed to run from the Windows command prompt. This utility allows you to get the IP address information of a Windows computer.

Using ipconfig

From the command prompt, type **ipconfig** to run the utility with default options. The output of the default command contains the IP address, network mask, and gateway for all physical and virtual network adapter.

```
#ipconfig
```

```
C:\Windows\system32\cmd.exe
C:\Users>ipconfig

Windows IP Configuration

Wireless LAN adapter Wireless Network Connection 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Wireless Network Connection 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . . . . . :

Wireless LAN adapter Wireless Network Connection:

    Connection-specific DNS Suffix . . . . . :
    IPv6 Address. . . . . : 2409:4072:616:44d0:61fd:d041:5a78:c2d8
    Temporary IPv6 Address. . . . . : 2409:4072:616:44d0:1093:b8ff:c0e:9b08
    Link-local IPv6 Address . . . . . : fe80::61fd:d041:5a78:c2d8%16
    IPv4 Address. . . . . : 192.168.43.194
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::d551:a02c:fa47:897c%16
```

4. nslookup

The **nslookup** (which stands for *name server lookup*) command is a network utility program used to obtain information about internet servers. It finds name server information for domains by querying the Domain Name System.

The nslookup command is a powerful tool for diagnosing DNS problems. You know you're experiencing a DNS problem when you can access a resource by specifying its IP address but not its DNS name.

```
#nslookup
```

5. Trace route:

Traceroute uses Internet Control Message Protocol (ICMP) echo packets with variable time to live (TTL) values. The response time of each hop is calculated. To guarantee accuracy, each hop is queried multiple times (usually three times) to better measure the response of that particular hop.

Traceroute is a network diagnostic tool used to track the pathway taken by a packet on an IP network from source to destination. Traceroute also records the time taken for each hop the packet makes during its route to the destination. Traceroute uses Internet Control Message Protocol (ICMP) echo packets with variable time to live (TTL) values.

The response time of each hop is calculated. To guarantee accuracy, each hop is queried multiple times (usually three times) to better measure the response of that particular hop. Traceroute sends packets with TTL values that gradually increase from packet to packet, starting with TTL value of one. Routers decrement TTL values of packets by one when routing and discard packets whose TTL value has reached zero, returning the ICMP error message ICMP Time Exceeded.

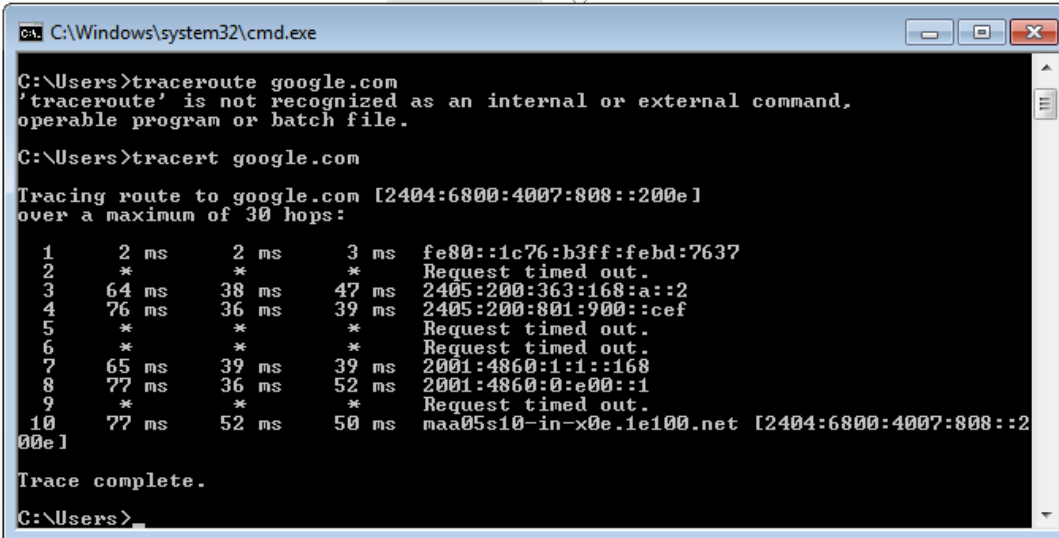
For the first set of packets, the first router receives the packet, decrements the TTL value and drops the packet because it then has TTL value zero. The router sends an ICMP Time Exceeded message back to the source. The next set of packets are given a TTL value of two, so the first router forwards the packets, but the second router drops them and replies with ICMP Time Exceeded.

Proceeding in this way, traceroute uses the returned ICMP Time Exceeded messages to build a list of routers that packets traverse, until the destination is reached and returns an ICMP Echo Reply message.

With the tracert command shown above, we're asking tracert to show us the path from the local computer all the way to the network device with the hostname

www.google.com.

```
#tracert google.com
```



```
C:\Windows\system32\cmd.exe
C:\Users>tracert google.com
'tracert' is not recognized as an internal or external command,
operable program or batch file.
C:\Users>tracert google.com
Tracing route to google.com [2404:6800:4007:808::200e]
over a maximum of 30 hops:
  0  0 ms  0 ms  0 ms  fe80::1c76:b3ff:febd:7637
  1  *      *      *      Request timed out.
  2  *      *      *      Request timed out.
  3  64 ms  38 ms  47 ms  2405:200:363:168:a::2
  4  76 ms  36 ms  39 ms  2405:200:801:900::cef
  5  *      *      *      Request timed out.
  6  *      *      *      Request timed out.
  7  65 ms  39 ms  39 ms  2001:4860:1:1::168
  8  77 ms  36 ms  52 ms  2001:4860:0:e00::1
  9  *      *      *      Request timed out.
 10  77 ms  52 ms  50 ms  maa05s10-in-x0e.1e100.net [2404:6800:4007:808::200e]
Trace complete.
C:\Users>
```

6. Ping:

The ping command sends an echo request to a host available on the network. Using this command, you can check if your remote host is responding well or not. Tracking and isolating hardware and software problems. Determining the status of the network and various foreign hosts. The ping command is usually used as a simple way to verify that a computer can communicate over the network with another computer or network device. The ping command operates by sending Internet Control Message Protocol (ICMP) Echo Request messages to the destination computer and waiting for a response

```
# ping 172.16.6.2
```



```
C:\Windows\system32\cmd.exe
er).
and has no effect on the type of service field in the IP Head
-r count      Record route for count hops (IPv4-only).
-s count      Timestamp for count hops (IPv4-only).
-j host-list  Loose source route along host-list (IPv4-only).
-k host-list  Strict source route along host-list (IPv4-only).
-w timeout    Timeout in milliseconds to wait for each reply.
-R           Use routing header to test reverse route also (IPv6-only).
-S srcaddr    Source address to use.
-4           Force using IPv4.
-6           Force using IPv6.

C:\Users>ping 172.16.6.2

Pinging 172.16.6.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 172.16.6.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users>
```

RESULT:

Ex.No: 2

DATE :

Write a HTTP web client program to download a web page using TCP sockets

AIM:

To write a java program for socket for HTTP for web page upload and download .

PRELAB DISCUSSION:

- HTTP means **H**yper**T**ext **T**ransfer **P**rotocol. HTTP is the underlying [protocol](#) used by the [World Wide Web](#) and this protocol defines how messages are formatted and transmitted, and what actions [Web servers](#) and [browsers](#) should take in response to various commands.
- For example, when you enter a [URL](#) in your browser, this actually sends an HTTP command to the Web server directing it to fetch and transmit the requested [Web page](#).
- The other main standard that controls how the World Wide Web works is [HTML](#), which covers how Web pages are formatted and displayed. HTTP functions as a [request–response](#) protocol in the client–server computing model.
- A [web browser](#), for example, may be the client and an application running on a computer [hosting a website](#) may be the server.
- The client submits an HTTP request message to the server. The server, which provides resources such as [HTML](#) files and other content, or performs other functions on behalf of the client, returns a response message to the client.
- The response contains completion status information about the request and may also contain requested content in its message body.

ALGORITHM:

Client:

1. Start.
2. Create socket and establish the connection with the server.
3. Read the image to be uploaded from the disk
4. Send the image read to the server
5. Terminate the connection
6. Stop.

Server:

1. Start
2. Create socket, bind IP address and port number with the created socket and make server a listening server.
3. Accept the connection request from the client
4. Receive the image sent by the client.
5. Display the image.
6. Close the connection.
7. Stop.

PROGRAM

OUTPUT

RESULT:

Ex.No: 3

DATE :

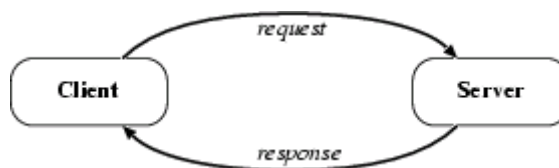
**Applications using TCP sockets like: Echo client and echo server,
Chat and File Transfer**

AIM

To write a java program for application using TCP Sockets Links

PRE LAB DISCUSSION:

- In the TCP Echo client a socket is created. Using the socket a connection is made to the server using the connect() function. After a connection is established, we send messages input from the user and display the data received from the server using send() and read() functions.
- In the TCP Echo server, we create a socket and bind to a advertised port number. After binding the process listens for incoming connections. Then an infinite loop is started to process the client requests for connections. After a connection is requested, it accepts the connection from the client machine and forks a new process.
- The new process receives data from the client using recv() function and echoes the same data using the send() function. Please note that this server is capable of handling multiple clients as it forks a new process for every client trying to connect to the server. TCP socket routines enable reliable IP communication using the transmission control protocol (TCP).
- The implementation of the [Transmission Control Protocol \(TCP\)](#) in the Network Component. TCP runs on top of the Internet Protocol (IP). TCP is a connection-oriented and reliable, full duplex protocol supporting a pair of byte streams, one for each direction.
- A TCP connection must be established before exchanging data. TCP retransmits data that do not reach the final destination due to errors or data corruption. Data is delivered in the sequence of its transmission



Echo client and echo server

ALGORITHM

Client

1. Start
2. Create the TCP socket
3. Establish connection with the server
4. Get the message to be echoed from the user

5. Send the message to the server
6. Receive the message echoed by the server
7. Display the message received from the server
8. Terminate the connection
9. Stop

Server

1. Start
2. Create TCP socket, make it a listening socket
3. Accept the connection request sent by the client for connection establishment
4. Receive the message sent by the client
5. Display the received message
6. Send the received message to the client from which it receives
7. Close the connection when client initiates termination and server becomes a listening server, waiting for clients.
8. Stop.

PROGRAM:

B.Chat

ALGORITHM

Client

1. Start
2. Create the UDP datagram socket
3. Get the request message to be sent from the user
4. Send the request message to the server
5. If the request message is "END" go to step 10
6. Wait for the reply message from the server
7. Receive the reply message sent by the server
8. Display the reply message received from the server
9. Repeat the steps from 3 to 8
10. Stop

Server

1. Start
2. Create UDP datagram socket, make it a listening socket
3. Receive the request message sent by the client
4. If the received message is "END" go to step 10
5. Retrieve the client's IP address from the request message received
6. Display the received message
7. Get the reply message from the user
8. Send the reply message to the client
9. Repeat the steps from 3 to 8.
10. Stop.

PROGRAM

OUTPUT

RESULT:

C. File

Transfer

AIM:

To write a java program for file transfer using TCP Sockets.

Algorithm

Server

1. Import java packages and create class file server.
2. Create a new server socket and bind it to the port.
3. Accept the client connection
4. Get the file name and stored into the BufferedReader.
5. Create a new object class file and realine.
6. If file is exists then FileReader read the content until EOF is reached.
7. Stop the program.

Client

1. Import java packages and create class file server.
2. Create a new server socket and bind it to the port.
3. Now connection is established.
4. The object of a BufferedReader class is used for storing data content which has been retrieved from socket object.
5. The connection is closed.
6. Stop the program.

PROGRAM

OUTPUT

RESULT

Ex.No: 4

DATE :

Simulation of DNS using UDP Sockets

AIM

To write a java program for DNS application

PRE LAB DISCUSSION:

- The Domain Name System (DNS) is a [hierarchical](#) decentralized naming system for computers, services, or other resources connected to the [Internet](#) or a private network. It associates various information with [domain names](#) assigned to each of the participating entities.
- The domain name space refers a hierarchy in the internet naming structure. This hierarchy has multiple levels (from 0 to 127), with a root at the top. The following diagram shows the domain name space hierarchy.
- Name server contains the DNS database. This database comprises of various names and their corresponding IP addresses. Since it is not possible for a single server to maintain entire DNS database, therefore, the information is distributed among many DNS servers.
- Types of Name Servers
- Root Server is the top level server which consists of the entire DNS tree. It does not contain the information about domains but delegates the authority to the other server
- Primary Server stores a file about its zone. It has authority to create, maintain, and update the zone file.
- Secondary Server transfers complete information about a zone from another server which may be primary or secondary server. The secondary server does not have authority to create or update a zone file.
- DNS is a TCP/IP protocol used on different platforms. The domain name space is divided into three different sections: generic domains, country domains, and inverse domain.
- The main function of DNS is to translate domain names into IP Addresses, which computers can understand. It also provides a list of mail servers which accept Emails for each domain name. Each domain name in DNS will nominate a set of name servers to be authoritative for its DNS records.

ALGORITHM

Server

1. Start
2. Create UDP datagram socket
3. Create a table that maps host name and IP address
4. Receive the host name from the client
5. Retrieve the client's IP address from the received datagram
6. Get the IP address mapped for the host name from the table.
7. Display the host name and corresponding IP address

8. Send the IP address for the requested host name to the client
9. Stop.

Client

1. Start
2. Create UDP datagram socket.
3. Get the host name from the client
4. Send the host name to the server
5. Wait for the reply from the server
6. Receive the reply datagram and read the IP address for the requested host name
7. Display the IP address.
8. Stop.

PROGRAM

OUTPUT

RESULT

Ex.No:5

DATE :

Write a code simulating ARP /RARP protocols

AIM:

To write a java program for simulating ARP and RARP protocols using TCP.

PRE LAB DISCUSSION:

- Address Resolution Protocol (ARP) is a low-level network protocol for translating network layer addresses into link layer addresses. ARP lies between layers 2 and 3 of the OSI model, although ARP was not included in the OSI framework and allows computers to introduce each other across a network prior to communication. Because protocols are basic network communication units, address resolution is dependent on protocols such as ARP, which is the only reliable method of handling required tasks.
- The Address Resolution Protocol (ARP) is a communication protocol used for discovering the link layer address, such as a MAC address, associated with a given internet layer address,
- When configuring a new network computer, each system is assigned an Internet Protocol (IP) address for primary identification and communication. A computer also has a unique media access control (MAC) address identity. Manufacturers embed the MAC address in the local area network (LAN) card. The MAC address is also known as the computer's physical address.
- Address Resolution Protocol (ARP) is used to resolve an IPv4 address (32 bit Logical Address) to the physical address (48 bit MAC Address). Network Applications at the Application Layer use IPv4 Address to communicate with another device.
- Reverse Address Resolution Protocol (RARP) is a network protocol used to resolve a data link layer address to the corresponding network layer address. For example, RARP is used to resolve a Ethernet MAC address to an IP address.
- The client broadcasts a RARP packet with an ethernet broadcast address, and it's own physical address in the data portion. The server responds by telling the client it's IP address. Note there is no name sent. Also note there is no security.
- Media Access Control (MAC) addresses need to be individually configured on the servers by an administrator. RARP is limited to serving only IP addresses. Reverse ARP differs from the Inverse Address Resolution Protocol which is designed to obtain the IP address associated with a local Frame Relay data link connection identifier. In ARP is not used in Ethernet.

ALGORITHM:

Client

1. Start the program
2. Create socket and establish connection with the server.
3. Get the IP address to be converted into MAC address from the user.
4. Send this IP address to server.
5. Receive the MAC address for the IP address from the server.
6. Display the received MAC address
7. Terminate the connection

Server

1. Start the program
2. Create the socket, bind the socket created with IP address and port number and make it a listening socket.
3. Accept the connection request when it is requested by the client.
4. Server maintains the table in which IP and corresponding MAC addresses are stored.
5. Receive the IP address sent by the client.
6. Retrieve the corresponding MAC address for the IP address and send it to the client.
7. Close the connection with the client and now the server becomes a listening server waiting for the connection request from other clients
8. Stop

PROGRAM

OUTPUT

RESULT

Ex.No: 6

DATE :

Study of Network simulator (NS) and Simulation of Congestion Control Algorithms using NS

AIM:

To Study Network simulator (NS).and Simulation of Congestion Control Algorithms using NS

PRE LAB DISCUSSION:

NET WORK SIMULATOR (NS2)

Ns Overview

- Ns Status
- Periodical release (ns-2.26, Feb 2003)
- Platform support
- FreeBSD, Linux, Solaris, Windows and Mac

Ns functionalities

Routing, Transportation, Traffic sources, Queuing disciplines, QoS

Congestion Control Algorithms

- Slow start
- Additive increase/multiplicative decrease
- Fast retransmit and Fast recovery

Case Study: A simple Wireless network.

Ad hoc routing, mobile IP, sensor-MAC
Tracing, visualization and various utilities
NS(Network Simulators)

Most of the commercial simulators are GUI driven, while some network simulators are CLI driven. The network model / configuration describes the state of the network (nodes, routers, switches, links) and the events (data transmissions, packet error etc.). An important output of simulations are the trace files. Trace files log every packet, every event that occurred in the simulation and are used for analysis. Network simulators can also provide other tools to facilitate visual analysis of trends and potential trouble spots.

Most network simulators use discrete event simulation, in which a list of pending "events" is stored, and those events are processed in order, with some events triggering future events—such as the event of the arrival of a packet at one node triggering the event of the arrival of that packet at a downstream node.

Simulation of networks is a very complex task. For example, if congestion is high, then estimation of the average occupancy is challenging because of high variance. To estimate the likelihood of a buffer overflow in a network, the time required for an accurate answer can be extremely large. Specialized techniques such as "control variates" and "importance sampling" have been developed to speed simulation.

Examples of network simulators

There are many both free/open-source and proprietary network simulators. Examples of notable network simulation software are, ordered after how often they are mentioned in research papers:

1. ns (open source)
2. OPNET (proprietary software)
3. NetSim (proprietary software)

Uses of network simulators

Network simulators serve a variety of needs. Compared to the cost and time involved in setting up an entire test bed containing multiple networked computers, routers and data links, network simulators are relatively fast and inexpensive. They allow engineers, researchers to test scenarios that might be particularly difficult or expensive to emulate using real hardware - for instance, simulating a scenario with several nodes or experimenting with a new protocol in the network. Network simulators are particularly useful in allowing researchers to test new networking protocols or changes to existing protocols in a controlled and reproducible environment. A typical network simulator encompasses a wide range of networking technologies and can help the users to build complex networks from basic building blocks such as a variety of nodes and links. With the help of simulators, one can design hierarchical networks using various types of nodes like computers, hubs, bridges, routers, switches, links, mobile units etc.

Various types of Wide Area Network (WAN) technologies like TCP, ATM, IP etc. and Local Area Network (LAN) technologies like Ethernet, token rings etc., can all be simulated with a typical simulator and the user can test, analyze various standard results apart from devising some novel protocol or strategy for routing etc. Network simulators are also widely used to simulate battlefield networks in Network-centric warfare.

There are a wide variety of network simulators, ranging from the very simple to the very complex. Minimally, a network simulator must enable a user to represent a network topology, specifying the nodes on the network, the links between those nodes and the traffic between the nodes. More complicated systems may allow the user to specify everything about the protocols used to handle traffic in a network. Graphical applications allow users to easily visualize the workings of their simulated environment. Text-based applications may provide a less intuitive interface, but may permit more advanced forms of customization.

Packet loss

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is distinguished as one of the three main error types encountered in digital communications; the other two being bit error and spurious packets caused due to noise.

Packets can be lost in a network because they may be dropped when a queue in the network node overflows. The amount of packet loss during the steady state is another important property of a congestion control scheme. The larger the value of packet loss, the more difficult

it is for transport layer protocols to maintain high bandwidths, the sensitivity to loss of individual packets, as well as to frequency and patterns of loss among longer packet sequences is strongly dependent on the application itself.

Throughput

Throughput is the main performance measure characteristic, and most widely used. In communication networks, such as Ethernet or packet radio, throughput or network throughput is the average rate of successful message delivery over a communication channel. Throughput is usually measured in bits per second (bit/s or bps), and sometimes in data packets per second or data packets per time slot. This measures how soon the receiver is able to get a certain amount of data sent by the sender. It is determined as the ratio of the total data received to the end-to-end delay. Throughput is an important factor which directly impacts the network performance.

Delay

Delay is the time elapsed while a packet travels from one point e.g., source premise or network ingress to destination premise or network egress. The larger the value of delay, the more difficult it is for transport layer protocols to maintain high bandwidths. We will calculate end-to-end delay

Queue Length

A queuing system in networks can be described as packets arriving for service, waiting for service if it is not immediate, and if having waited for service, leaving the system after being served. Thus queue length is very important characteristic to determine that how well the active queue management of the congestion control algorithm has been working.

Congestion control Algorithms

Slow-start is used in conjunction with other [algorithms](#) to avoid sending more data than the network is capable of transmitting, that is, to avoid causing [network congestion](#). The [additive increase/multiplicative decrease](#) (AIMD) algorithm is a feedback control algorithm. AIMD combines linear growth of the congestion window with an exponential reduction when a congestion takes place. Multiple flows using AIMD congestion control will eventually converge to use equal amounts of a contended link. Fast Retransmit is an enhancement to [TCP](#) that reduces the time a sender waits before retransmitting a lost [segment](#).

PROGRAM

OUTPUT

RESULT

Ex.No: 7

DATE :

Study of TCP/UDP performance using Simulation tool.

AIM:

To simulate the performance of TCP/UDP using NS2.

PRE LAB DISCUSSION:

- TCP is reliable protocol. That is, the receiver always sends either positive or negative acknowledgement about the data packet to the sender, so that the sender always has bright clue about whether the data packet is reached the destination or it needs to resend it.
- TCP ensures that the data reaches intended destination in the same order it was sent.
- TCP is connection oriented. TCP requires that connection between two remote points be established before sending actual data.
- TCP provides error-checking and recovery mechanism.
- TCP provides end-to-end communication.
- TCP provides flow control and quality of service.
- TCP operates in Client/Server point-to-point mode.
- TCP provides full duplex server, i.e. it can perform roles of both receiver and sender.
- The User Datagram Protocol (UDP) is simplest Transport Layer communication protocol available of the TCP/IP protocol suite. It involves minimum amount of communication mechanism. UDP is said to be an unreliable transport protocol but it uses IP services which provides best effort delivery mechanism.UDP is used when acknowledgement of data does not hold any significance.
- UDP is good protocol for data flowing in one direction.
- UDP is simple and suitable for query based communications.
- UDP is not connection oriented.
- UDP does not provide congestion control mechanism.
- UDP does not guarantee ordered delivery of data.
- UDP is stateless.
- UDP is suitable protocol for streaming applications such as VoIP, multimedia streaming.

TCP Performance

Algorithm

1. Create a Simulator object.
2. Set routing as dynamic.
3. Open the trace and nam trace files.
4. Define the finish procedure.
5. Create nodes and the links between them.
6. Create the agents and attach them to the nodes.
7. Create the applications and attach them to the tcp agent.

8. Connect tcp and tcp sink.
9. Run the simulation.

PROGRAM

OUTPUT

UDP Performance

ALGORITHM :

1. Create a Simulator object.
2. Set routing as dynamic.
3. Open the trace and nam trace files.
4. Define the finish procedure.
5. Create nodes and the links between them.
6. Create the agents and attach them to the nodes.
7. Create the applications and attach them to the UDP agent.
8. Connect udp and null agents.
9. Run the simulation.

PROGRAM

OUTPUT

RESULT

Ex.No: 8

DATE :

Simulation of Distance Vector/ Link State Routing algorithm.

AIM:

To simulate the Distance vector and link state routing protocols using NS2.

PRE LAB DISCUSSION:

LINK STATE ROUTING

Routing is the process of selecting best paths in a network. In the past, the term routing was also used to mean forwarding network traffic among networks. However this latter function is much better described as simply forwarding. Routing is performed for many kinds of networks, including the telephone network (circuit switching), electronic data networks (such as the Internet), and transportation networks. This article is concerned primarily with routing in electronic data networks using packet switching technology.

In packet switching networks, routing directs packet forwarding (the transit of logically addressed network packets from their source toward their ultimate destination) through intermediate nodes. Intermediate nodes are typically network hardware devices such as routers, bridges, gateways, firewalls, or switches. General-purpose computers can also forward packets and perform routing, though they are not specialized hardware and may suffer from limited performance. The routing process usually directs forwarding on the basis of routing tables which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time. Multipath routing techniques enable the use of multiple alternative paths.

In case of overlapping/equal routes, the following elements are considered in order to decide which routes get installed into the routing table (sorted by priority):

1. *Prefix-Length*: where longer subnet masks are preferred (independent of whether it is within a routing protocol or over different routing protocol)
2. *Metric*: where a lower metric/cost is preferred (only valid within one and the same routing protocol)
3. *Administrative distance*: where a lower distance is preferred (only valid between different routing protocols)

Routing, in a more narrow sense of the term, is often contrasted with bridging in its assumption that network addresses are structured and that similar addresses imply proximity within the network. Structured addresses allow a single routing table entry to represent the route to a group of devices. In large networks, structured addressing (routing, in the narrow sense) outperforms unstructured addressing (bridging). Routing has become the dominant form of addressing on the Internet. Bridging is still widely used within localized environments.

b. Flooding

Flooding is a simple routing algorithm in which every incoming packet is sent through every outgoing link except the one it arrived on. Flooding is used in bridging and in systems such as Usenet and peer-to-peer file sharing and as part of some routing protocols, including OSPF, DVMRP, and those used in ad-hoc wireless networks. There are generally two types of flooding available, Uncontrolled Flooding and Controlled Flooding. Uncontrolled Flooding is the fatal law of flooding. All nodes have neighbours and route packets indefinitely. More than two neighbours creates a broadcast storm.

Controlled Flooding has its own two algorithms to make it reliable, SNCF (Sequence Number Controlled Flooding) and RPF (Reverse Path Flooding). In SNCF, the node attaches its own address and sequence number to the packet, since every node has a memory of addresses and sequence numbers. If it receives a packet in memory, it drops it immediately while in RPF, the node will only send the packet forward. If it is received from the next node, it sends it back to the sender.

Distance vector Routing:

In computer communication theory relating to packet-switched networks, a **distance-vector routing protocol** is one of the two major classes of routing protocols, the other major class being the link-state protocol. Distance-vector routing protocols use the Bellman–Ford algorithm, Ford–Fulkerson algorithm, or DUAL FSM (in the case of Cisco Systems's protocols) to calculate paths.

A distance-vector routing protocol requires that a router informs its neighbors of topology changes periodically. Compared to link-state protocols, which require a router to inform all the nodes in a network of topology changes, distance-vector routing protocols have less computational complexity and message overhead.

The term *distance vector* refers to the fact that the protocol manipulates *vectors* (arrays) of distances to other nodes in the network. The vector distance algorithm was the original ARPANET routing algorithm and was also used in the internet under the name of RIP (Routing Information Protocol). Examples of distance-vector routing protocols include RIPv1 and RIPv2 and IGRP.

Method

Routers using distance-vector protocol do not have knowledge of the entire path to a destination. Instead they use two methods:

1. Direction in which router or exit interface a packet should be forwarded.
2. Distance from its destination

Distance-vector protocols are based on calculating the direction and distance to any link in a network. "Direction" usually means the next hop address and the exit interface. "Distance" is a measure of the cost to reach a certain node. The least cost route between any two nodes is the route with minimum distance. Each node maintains a vector (table) of minimum distance to

every node. The cost of reaching a destination is calculated using various route metrics. RIP uses the hop count of the destination whereas IGRP takes into account other information such as node delay and available bandwidth.

Updates are performed periodically in a distance-vector protocol where all or part of a router's routing table is sent to all its neighbors that are configured to use the same distance-vector routing protocol. RIP supports cross-platform distance vector routing whereas IGRP is a Cisco Systems proprietary distance vector routing protocol. Once a router has this information it is able to amend its own routing table to reflect the changes and then inform its neighbors of the changes. This process has been described as 'routing by rumor' because routers are relying on the information they receive from other routers and cannot determine if the information is actually valid and true. There are a number of features which can be used to help with instability and inaccurate routing information.

EGP and BGP are not pure distance-vector routing protocols because a distance-vector protocol calculates routes based only on link costs whereas in BGP, for example, the local route preference value takes priority over the link cost.

Count-to-infinity problem

The Bellman–Ford algorithm does not prevent routing loops from happening and suffers from the **count-to-infinity problem**. The core of the count-to-infinity problem is that if A tells B that it has a path somewhere, there is no way for B to know if the path has B as a part of it. To see the problem clearly, imagine a subnet connected like A–B–C–D–E–F, and let the metric between the routers be "number of jumps". Now suppose that A is taken offline. In the vector-update-process B notices that the route to A, which was distance 1, is down – B does not receive the vector update from A. The problem is, B also gets an update from C, and C is still not aware of the fact that A is down – so it tells B that A is only two jumps from C (C to B to A), which is false. This slowly propagates through the network until it reaches infinity (in which case the algorithm corrects itself, due to the relaxation property of Bellman–Ford).

ALGORITHM:

1. Create a Simulator object.
2. Set routing as dynamic.
3. Open the trace and nam trace files.
4. Define the finish procedure.
5. Create nodes and the links between them.
6. Create the agents and attach them to the nodes.
7. Create the applications and attach them to the udp agent.
8. Connect udp and null.
9. At 1 sec the link between node 1 and 2 is broken.
10. At 2 sec the link is up again.
11. Run the simulation.

LINK STATE ROUTING PROTOCOL

PROGRAM

OUTPUT

DISTANCE VECTOR ROUTING ALGORITHM

ALGORITHM:

1. Create a simulator object
2. Set routing protocol to Distance Vector routing
3. Trace packets on all links onto NAM trace and text trace file
4. Define finish procedure to close files, flush tracing and run NAM
5. Create eight nodes
6. Specify the link characteristics between nodes
7. Describe their layout topology as a octagon
8. Add UDP agent for node n1
9. Create CBR traffic on top of UDP and set traffic parameters.
10. Add a sink agent to node n4
11. Connect source and the sink
12. Schedule events as follows:
 - a. Start traffic flow at 0.5
 - b. Down the link n3-n4 at 1.0
 - c. Up the link n3-n4 at 2.0
 - d. Stop traffic at 3.0
 - e. Call finish procedure at 5.0
13. Start the scheduler
14. Observe the traffic route when link is up and down
15. View the simulated events and trace file analyze it
16. Stop

PROGRAM

OUTPUT

RESULT

Ex.No:9

DATE :

Performance Evaluation of Routing protocols using Simulation tool.

(a) UNICAST ROUTING PROTOCOL

AIM:

To write a ns2 program for implementing unicast routing protocol.

PRE LAB DISCUSSION:

- When a device has multiple paths to reach a destination, it always selects one path by preferring it over others. This selection process is termed as Routing. Routing is done by special network devices called routers or it can be done by means of software processes.
- The software based routers have limited functionality and limited scope. A router is always configured with some default route. A default route tells the router where to forward a packet if there is no route found for specific destination.
- In case there are multiple paths existing to reach the same destination, a router can make a decision based on the following information. Routes can be statically configured or dynamically learnt. One route can be configured to be preferred over others. Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with a specified destination. Routing unicast data over the internet is called unicast routing.
- It is the simplest form of routing because the destination is already known. Hence the router just has to look up the routing table and forward the packet to the next hop.
- Multicasting in a computer network is a group communication, where a sender(s) send data to multiple receivers simultaneously. It supports one-to-many and many-to-many data transmission across LANs or WANs. Through the process of multicasting, the communication and processing overhead of sending the same data packet or data frame is minimized.
- Multicast IP Routing protocols are used to distribute data (for example, audio/video streaming broadcasts) to multiple recipients. Using multicast, a source can send a single copy of data to a single multicast address, which is then distributed to an entire group of recipients.
- The key difference between broadcast and multicast is that in the broadcast the packet is delivered to all the hosts connected to the network whereas, in a multicast packet is delivered to intended recipients only.
- Multicast Message. Multicasting identifies logical groups of computers. A single message can then be sent to the group. Multicast Message. Multicasting uses the Internet Group Management Protocol (IGMP) to identify groups and group members.

ALGORITHM:

1. Start the program.
2. Declare the global variables ns for creating a new simulator.
3. Set the color for packets.
4. Open the network animator file in the name of file2 in the write mode.
5. Open the trace file in the name of file 1 in the write mode.
6. Set the unicast routing protocol to transfer the packets in network.
7. Create the required no of nodes.
8. Create the duplex-link between the nodes including the delay time,bandwidth and dropping queue mechanism.
9. Give the position for the links between the nodes.
10. Set a tcp reno connection for source node.
11. Set the destination node using tcp sink.
12. Setup a ftp connection over the tcp connection.
13. Down the connection between any nodes at a particular time.
14. Reconnect the downed connection at a particular time.
15. Define the finish procedure.
16. In the definition of the finish procedure declare the global variables ns, file1, and file2.
17. Close the trace file and name file and execute the network animation file.
18. At the particular time call the finish procedure.
19. Stop the program.

PROGRAM

OUTPUT

RESULT

(b) MULTICASTING ROUTING PROTOCOL

AIM:

To write a ns2 program for implementing multicasting routing protocol.

ALGORITHM:

1. Start the program.
2. Declare the global variables ns for creating a new simulator.
3. Set the color for packets.
4. Open the network animator file in the name of file2 in the write mode.
5. Open the trace file in the name of file 1 in the write mode.
6. Set the multicast routing protocol to transfer the packets in network.
7. Create the multicast capable no of nodes.
8. Create the duplex-link between the nodes including the delay time,bandwidth and dropping queue mechanism.
9. Give the position for the links between the nodes.
10. Set a udp connection for source node.
11. Set the destination node ,port and random false for the source and destination files.
12. Setup a traffic generator CBR for the source and destination files.
13. Down the connection between any nodes at a particular time.
14. Create the receive agent for joining and leaving if the nodes in the group.
15. Define the finish procedure.
16. In the definition of the finish procedure declare the global variables.
17. Close the trace file and namefile and execute the network animation file.
18. At the particular time call the finish procedure.
19. Stop the program.

PROGRAM

OUTPUT

RESULT

Ex.No:10

DATE :

Simulation of Error Detection Code (like CRC)

AIM:

To implement error checking code using java.

PRE LAB DISSCUSSION:

The cyclic redundancy check, or CRC, is a technique for detecting errors in digital data, but not for making corrections when errors are detected. It is used primarily in data transmission.

In the CRC method, a certain number of check bits, often called a checksum, are appended to the message being transmitted. The receiver can determine whether or not the check bits agree with the data, to ascertain with a certain degree of probability whether or not an error occurred in transmission.

CRC involves binary division of the data bits being sent by a predetermined divisor agreed upon by the communicating system. The divisor is generated using polynomials. So, CRC is also called polynomial code checksum.

CRC uses Generator Polynomial which is available on both sender and receiver side. An example generator polynomial is of the form like $x^3 + x + 1$. This generator polynomial represents key 1011. Another example is $x^2 + 1$ that represents key 101.

Sender Side (Generation of Encoded Data from Data and Generator Polynomial (or Key)):

- The binary data is first augmented by adding $k-1$ zeros in the end of the data
- Use modulo-2 binary division to divide binary data by the key and store remainder of division.
- Append the remainder at the end of the data to form the encoded data and send the same

Receiver Side (Check if there are errors introduced in transmission)

Perform modulo-2 division again and if remainder is 0, then there are no errors.

Modulo 2 Division:

- The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. Just that instead of subtraction, we use XOR here.
- In each step, a copy of the divisor (or data) is XORed with the k bits of the dividend (or key).
- The result of the XOR operation (remainder) is $(n-1)$ bits, which is used for the next step after 1 extra bit is pulled down to make it n bits long.
- When there are no bits left to pull down, we have a result. The $(n-1)$ -bit remainder which is appended at the sender side.

ALGORITHM:

1. Start the Program
2. Given a bit string, append 0s to the end of it (the number of 0s is the same as the degree of the generator polynomial) let $B(x)$ be the polynomial corresponding to B.
3. Divide $B(x)$ by some agreed on polynomial $G(x)$ (generator polynomial) and determine the remainder $R(x)$. This division is to be done using Modulo 2 Division.
4. Define $T(x) = B(x) - R(x)$
5. $(T(x)/G(x) \Rightarrow \text{remainder } 0)$
6. Transmit T, the bit string corresponding to $T(x)$.
7. Let T' represent the bit stream the receiver gets and $T'(x)$ the associated polynomial. The receiver divides $T'(x)$ by $G(x)$. If there is a 0 remainder, the receiver concludes $T = T'$ and no error occurred otherwise, the receiver concludes an error occurred and requires a retransmission
8. Stop the Program

PROGRAM

OUTPUT

RESULT

MADHA ENGINEERING COLLEGE

(Affiliated to Anna University and Approved by AICTE, New Delhi)
Madha Nagar, Kundrathur, Chennai-600069

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CS3501- COMPILER DESIGN

R 2021

LAB MANUAL

INDEX

S.No.	Name of the Experiment	Page No	Date of Experiment	Remarks
1	Implementation Of Symbol Table			
2	Develop A Lexical Analyzer To Recognize A Few Patterns In C			
3	Implementation Of Lexical Analyzer Using Lex Tool			
4	Generate Yacc Specification For A Few Syntactic Categories			
5	Implementation Of Three Address Code			
6	Implementation Of Type Checking			
7	Implement The Back End Of The Compiler Which Takes The Three Address Code And Produces The 8086 Assembly Language			
8	Implementation Of Simple Code Optimization Techniques			

Ex.No:01

DATE :

IMPLEMENTATION OF SYMBOL TABLE

AIM:

To write a C program to implement a symbol table.

ALGORITHM:

- 1) Start the program.
- 2) Get the input from the user with the terminating symbol '\$'.
- 3) Allocate memory for the variable by dynamic memory allocation function.
- 4) If the next character of the symbol is an operator then only the memory is allocated.
- 5) While reading, the input symbol is inserted into symbol table along with its memory address.
- 6) The steps are repeated till '\$' is reached.
- 7) To reach a variable, enter the variable to the searched and symbol table has been checked for corresponding variable, the variable along with its address is displayed as result.
- 8) Stop the program.

PROGRAM:

OUTPUT:

RESULT:

Ex.No:02

DATE :

DEVELOP A LEXICAL ANALYZER TO RECOGNIZE A FEW PATTERNS IN C

AIM:

To write and execute a C program to implement the lexical analyzer.

ALGORITHM:

1. Start the program.
2. Declare the file pointer and necessary variables.
3. Open the input file in the read mode.
4. Use the analyze function to analyze the input program and store the identifiers, keywords and operator on idhd, keyhd, ophd respectively.
5. Stores the tokens in data structure linked lists.
6. Increment the line number of each token and its occurrences.
7. Using the show function print the linked lists in a tabular format.
8. Stop the program.

PROGRAM:

OUTPUT:

RESULT:

Ex.No:03

DATE :

IMPLEMENTATION OF LEXICAL ANALYZER USING LEX TOOL

AIM:

To implement the lexical analyzer using lex tool for a subset of C language.

ALGORITHM:

1. Start the program.
2. Declare necessary variables and creates token representation using Regular.
3. Print the pre processor or directives, keywords by analysis of the input program.
4. In the program check whether there are arguments.
5. Declare a file and open it as read mode.
6. Read the file and if any taken in source program matches with RE that all returned as integer value.
7. Print the token identified using YYdex() function.
8. Stop the program

PROGRAM:

OUTPUT:

RESULT:

Ex.No:4

DATE :

GENERATE YACC SPECIFICATION FOR A FEW SYNTACTIC CATEGORIES

A) PROGRAM TO RECOGNIZE A VALID ARITHMETIC EXPRESSION THAT USES OPERATOR +, -, * AND /.

ALGORITHM:

1. Accept the token generated in lex part as input
2. Specify order of procedure
3. Define rules with end point
4. Parse input string from standard input by calling yyparse() by main function
5. Print the result of any rules matches
6. If none of results defined matches print "invalid expression"

COMPILATION STEPS:

```
$ lex exp.l
$ yacc -d exp.y
$ cc lex.yy.c y.tab.c -ll -ly
$ ./a.out
```

PROGRAM:

OUTPUT:

B) PROGRAM TO RECOGNIZE A VALID VARIABLE WHICH STARTS WITH A LETTER FOLLOWED BY ANY NUMBER OF LETTERS OR DIGITS.

PROGRAM:

OUTPUT:

C) IMPLEMENTATION OF CALCULATOR USING LEX AND YACC

PROGRAM:

OUTPUT:

RESULT:

Ex.No:05

DATE :

IMPLEMENTATION OF THREE ADDRESS CODE

AIM:

To Generate three address code using LEX and YACC.

ALGORITHM:

1. Start the program.
2. Include the header file.●
3. In int code.l,declare the variable line no as integer and assign it to be equal to '1'.
4. Start the int code.l with declarative section.
5. In translation rules section define keywords ,data types and integer along with their actions .
6. Start the main block. In main block check the statement
7. 1.declarative 2.assignment 3.conditional 4.if and else 5.While assignment.
8. Perform the actions of that particular block.
9. In main program declare the parameters argc as int and *argv[] as char.
10. In main program open file in read mode.
11. Print the output in a file.
12. End the program.

PROGRAM:

OUTPUT:

RESULT:

Ex.No:06

DATE :

IMPLEMENTATION OF TYPE CHECKING

AIM:

To write a C program for implementing type checking for given expression

ALGORITHM:

1. Start a program.
2. Include all the header files.
3. Initialize all the functions and variables.
4. Get the expression from the user and separate into the tokens.
5. After separation, specify the identifiers, operators and number.
6. Print the output.
7. Stop the program.

PROGRAM:

OUTPUT:

RESULT:

Ex.No:07

DATE :

**IMPLEMENT THE BACK END OF THE COMPILER WHICH
TAKES THE THREE ADDRESS CODE AND PRODUCES
THE 8086 ASSEMBLY LANGUAGE**

AIM:

To write a C program to implement Simple Code Generator.

ALGORITHM:

1. Start
2. Get address code sequence.
3. Determine current location of 3 using address (for 1st operand).
4. If current location not already exist generate move (B,O).
5. Update address of A(for 2nd operand).
6. If current value of B and () is null,exist.
7. If they generate operator () A,3 ADPR.
8. Store the move instruction in memory
9. Stop

PROGRAM:

OUTPUT:

RESULT:

Ex.No:08

DATE :

IMPLEMENTATION OF SIMPLE CODE OPTIMIZATION TECHNIQUES

AIM:

To write a C program to implement Code Optimization Techniques.

ALGORITHM:

Input: Set of 'L' values with corresponding 'R' values.

Output: Intermediate code & Optimized code after eliminating common expressions.

PROGRAM:

OUTPUT:

RESULT: