



**VV COLLEGE OF ENGINEERING
TISAIYANVILAI**

LAB MANUAL

STUDENT NAME :

REGISTER NUMBER :

SUBJECT CODE : CCS356

SUBJECT NAME : OBJECT ORIENTED SOFTWARE ENGINEERING
LABORATORY

DEGREE /BRANCH : BE / CSE

YEAR / SEM : III / 06

ACADEMIC YEAR : 2023 - 2024



V V COLLEGE OF ENGINEERING

(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)

V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

College Vision and Mission Statement

Vision

“Emerge as a premier technical institution of global standards, producing enterprising, knowledgeable engineers and entrepreneurs.”

Mission

- Impart quality and contemporary technical education for rural students.
- Have the state of the art infrastructure and equipment for quality learning.
- Enable knowledge with ethics, values and social responsibilities.
- Inculcate innovation and creativity among students for contribution to society.

Vision and Mission of the Department of Computer Science and Engineering

Vision

“Produce competent and intellectual computer science graduates by empowering them to compete globally towards professional excellence”.

Mission

- Provide resources, environment and continuing learning processes for better exposure in latest and contemporary technologies in Computer Science and Engineering.
- Encourage creativity and innovation and the development of self-employment through knowledge and skills, for contribution to society
- Provide quality education in Computer Science and Engineering by creating a platform to enable coding, problem solving, design, development, testing and implementation of solutions for the benefit of society.

I. PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of Computer Science Engineering can

- Apply their technical competence in computer science to solve real world problems, with technical and people leadership.
- Conduct cutting edge research and develop solutions on problems of social relevance.
- Work in a business environment, exhibiting team skills, work ethics, adaptability and lifelong learning.

II. PROGRAM OUTCOMES (POs)

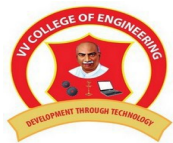
1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

III. PROGRAM SPECIFIC OUTCOMES (PSOs)

The Students will be able to

- Exhibit design and programming skills to build and automate business solutions using cutting edge technologies.
- Strong theoretical foundation leading to excellence and excitement towards research, to provide elegant solutions to complex problems.
- Ability to work effectively with various engineering fields as a team to design, build and develop system applications.



V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LABORATORY

LIST OF EXPERIMENTS – R2021

PRACTICAL SUBJECT NAME	OBJECT ORIENTED SOFTWARE ENGINEERING
PRACTICAL SUBJECT CODE	CCS356
SEMESTER/ YEAR	06 / THIRD
TOTAL HOURS	30
DEGREE / DEPARTMENT	BE / COMPUTER SCIENCE AND ENGINEERING
STAFF IN-CHARGE	Mrs. M. JANCYRANI MALLI
LAB INSTRUCTOR	Mr. AROCKIA STANLY
REGULATION	2021

CO1	Compare various Software Development Life cycle Models
CO2	Evaluate project management approaches as well as cost and schedule estimation strategies
CO3	Perform formal analysis on specifications
CO4	Use UML diagrams for analysis and design
CO5	Architect and design using architectural styles and design patterns and test the system



V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S.No	Name of the Experiment	CO Mapping	PO Mapping
1	CASE Tools	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
2	Passport Automation Sysem	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
3	Book Bank	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
4	Exam Registration	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
5	Stock Maintenance System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
6	Online Course Reservation System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
7	Airline / Railway Reservation System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
8	Software Personnel Management System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
9	Credit Card Processing	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
10	E-Book Management System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
11	Recruitment system	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
12	Foreign Trading System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
13	Conference Management System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
14	BPO Management System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
15	LibraryManagement System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3
16	Student Information System	CO1,CO2,CO3, CO4,CO5	PO3, PO5, PO12 PSO1-PSO3



V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RUBRICS FOR ASSESSING LABORATORY

Sl. No.	Criteria	Total Marks	Excellent (25)	Good (20)	Average (10)	Poor (5)
			91% - 100%	71% - 90%	50% - 70%	<50%
1	Preparation	25	Gives clear idea about the aim and having good capability of executing experiments.	Capability of executing experiments but no proper clarification about the objective.	Gives clear idea about the target and has less capability of executing experiments.	Gives indistinct idea about the target and has less capability of executing experiments & who feel difficult to follow the objectives.
2	Viva	25	Have executed the experiments in an efficient way & make credible and unbiased judgments regarding the experiments.	Executed the experiments with less efficient & has partial judgments regarding the experiments.	Executed the experiments with less efficiency and has no judgments regarding experiments .	Incomplete experiments & lack of judgments regarding experiments.
3	Performance	25	Followed all the instructions given in the procedure and submitted the manual on time.	Followed all the instructions given in the procedure with some assisting.	Followed some of the instructions given in the procedure & late in submission of manual.	Unable to follow the instructions given in the procedure & late in submission of manual.



V V COLLEGE OF ENGINEERING
(Approved By AICTE, New Delhi and Affiliated To Anna University Chennai)
V V Nagar, Arasoor, Tisaiyanvilai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering		
Preparation	25	
Viva	25	
Performance	25	
Total	75	
Lab Incharge	Date	

INDEX

S.No	DATE	NAME OF THE EXPERIMENT	SIGN
1		CASE Tools	
2		Passport Automation Sysem	
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		LibraryManagement System	
16		Student Information System	

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

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.

1. 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.

2. Class diagram:

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.

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 run time system. It shows the configuration of run time 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 OF ELEMENTS:

These are explanatory parts of the 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 in 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 the asset of links. A link is being connected among objects. Graphically association is represented as a solid line possibly including labels.

Generalization:

It is a specialized relationship in which the specialized elements are substitutable for objects 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 the structure and behavior of the 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 these building blocks are put together.
- Some common mechanisms that apply throughout the language.

Once you have grasped these ideas, you may be able to read. UML creates some basic ones. As you gain more experience in applying conceptual models 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 the 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 an “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 shows 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 workflow 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 the 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 diagrams. 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 a situation during the life of an object, which satisfies some condition or waits for an event.

Start state:

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

Object flow:

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

Synchronization:

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

End state:

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

Sequence diagram:

Description:

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

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

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

Object:

An object has state, behavior and identity. An object is not based is referred to as an instance. The various objects in e-mail client system are:

- ❖ User
- ❖ Website
- ❖ Login
- ❖ Groups

Message icon:

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

Collaboration diagram:

Description:

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

Collaboration diagram shows object s, their links and their messages. They can also contain simple class instances and class utility instances.

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

Toggling between the sequence and collaboration diagrams

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

CONCLUSION:

Ex no: 2

PASSPORT AUTOMATION SYSTEM

Date:

AIM:

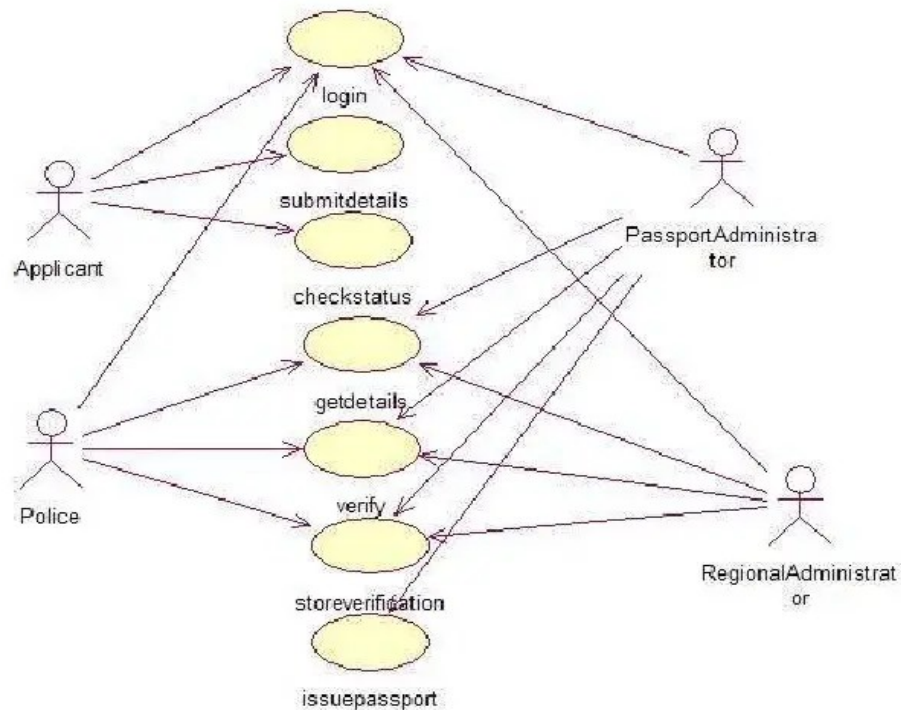
To create an automated system to perform the Passport Automation System using Argo UML.

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.

USE CASE DIAGRAM

1. The actors in use case diagram are Applicant, regional administrator, database, passport Administrator, Police.
2. The use cases are Login, givedetails, logout, collectdetails, verification, issue.
3. The actors use the use case are denoted by the arrow
4. The login use case checks the username and password for applicant, regional administrator, passport administrator and police.
5. The submit details use case is used by the applicant for submitting his details
6. The check status use case is used by the applicant for checking the status of the application process.
7. The get details, verify and store verification use case is used by passport administrator, regional administrator, and police.
8. The details use case is used for getting the details form the database for verification

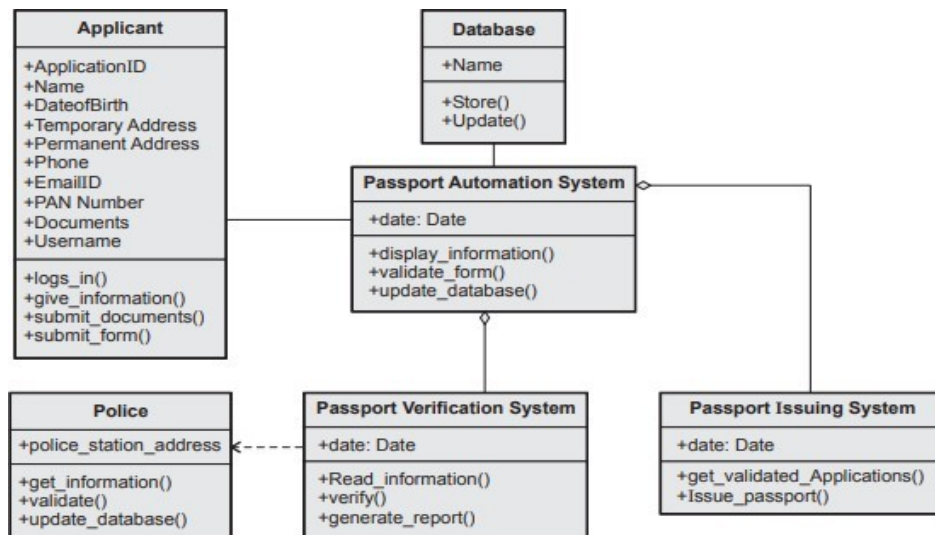


CLASS DIAGRAM

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.

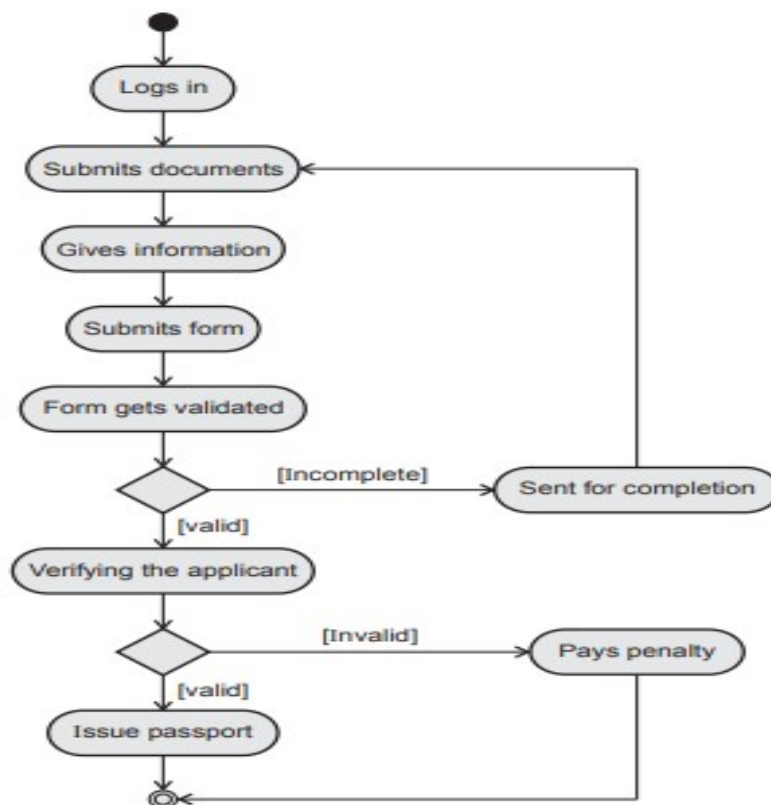
The Passport Automation system class diagram consists of four classes
Passport Automation System

1. Applicant
2. Database
3. Police
4. Passport verification system
5. Passport issuing system



ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of step wise 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.



SEQUENCE DIAGRAM

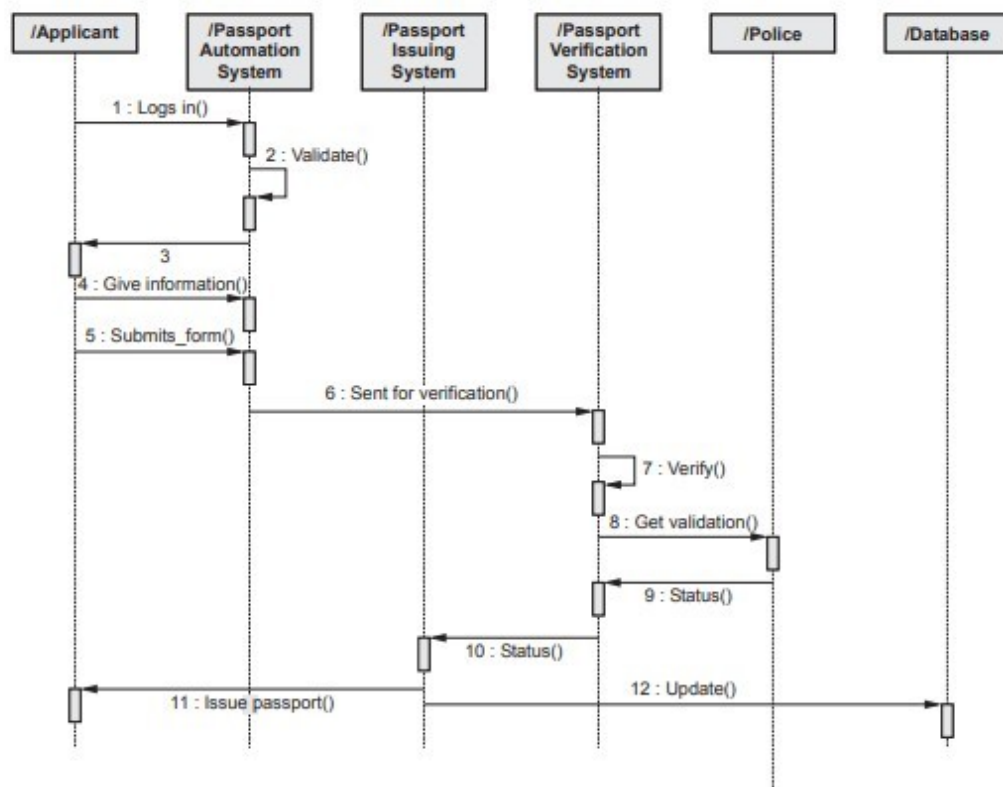
A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions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

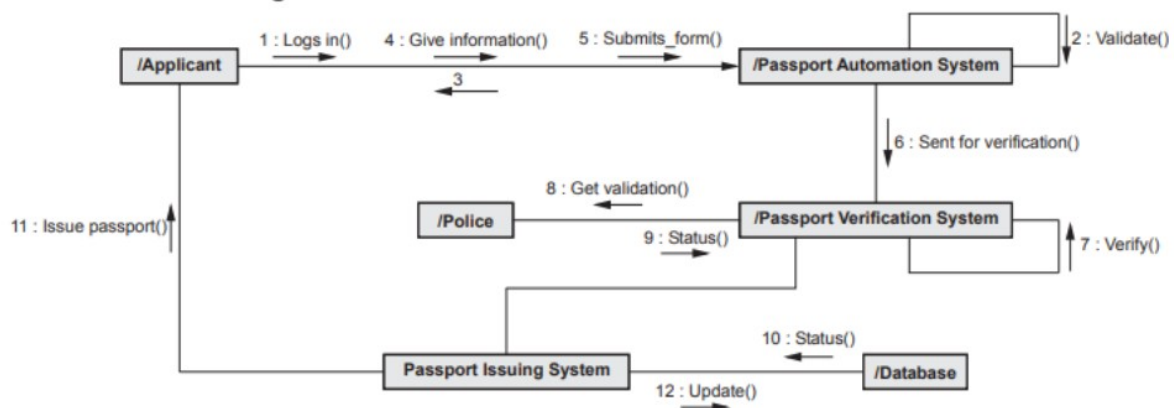
Sequence diagram



COLLABORATION DIAGRAM:

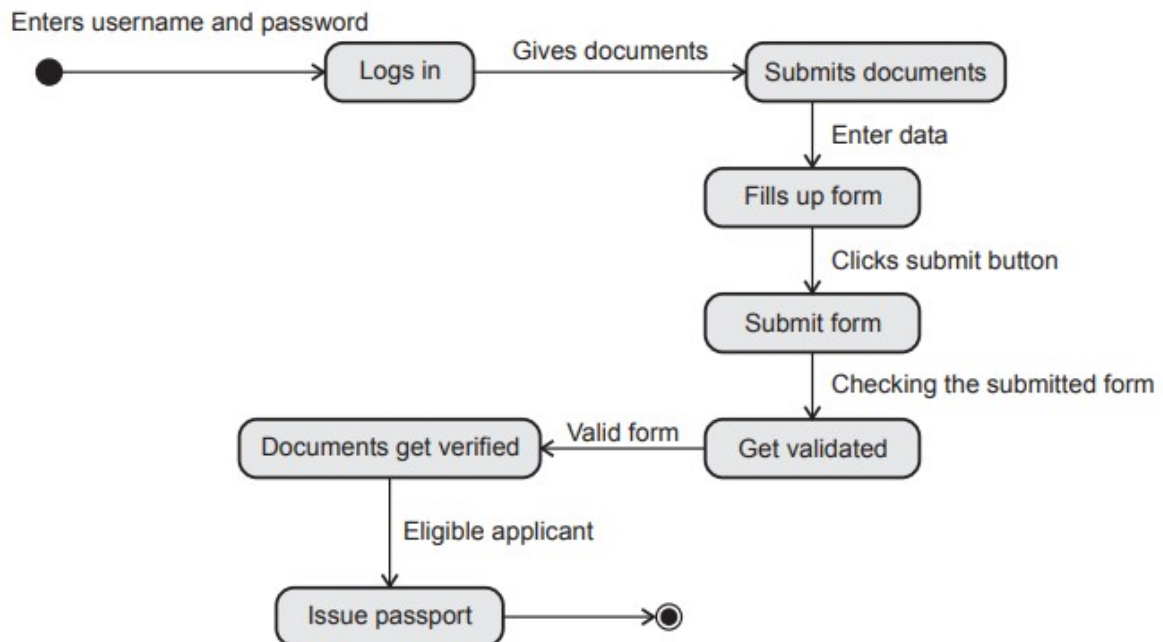
The diagrams show the process done by the administrator to the Passport Automation system. The applicant has to enter his details. The application details entered are verified by the administrator and the applicant is approved if the details

match then the passport is dispatch, otherwise an appropriate error message is displayed.



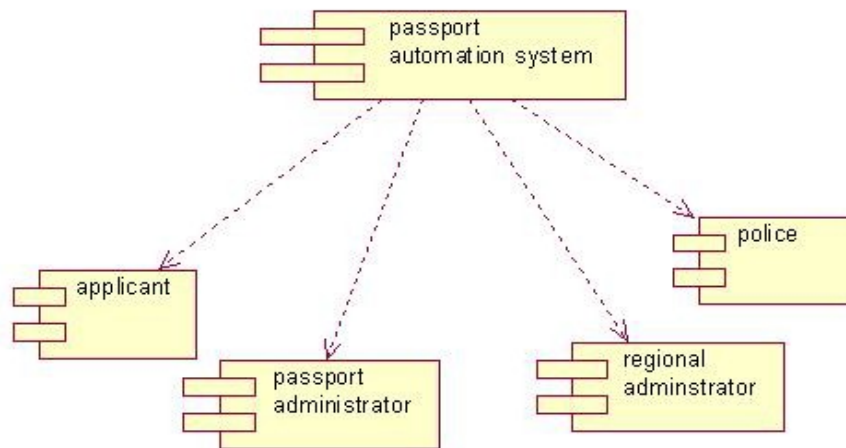
STATE CHART DIAGRAM

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.



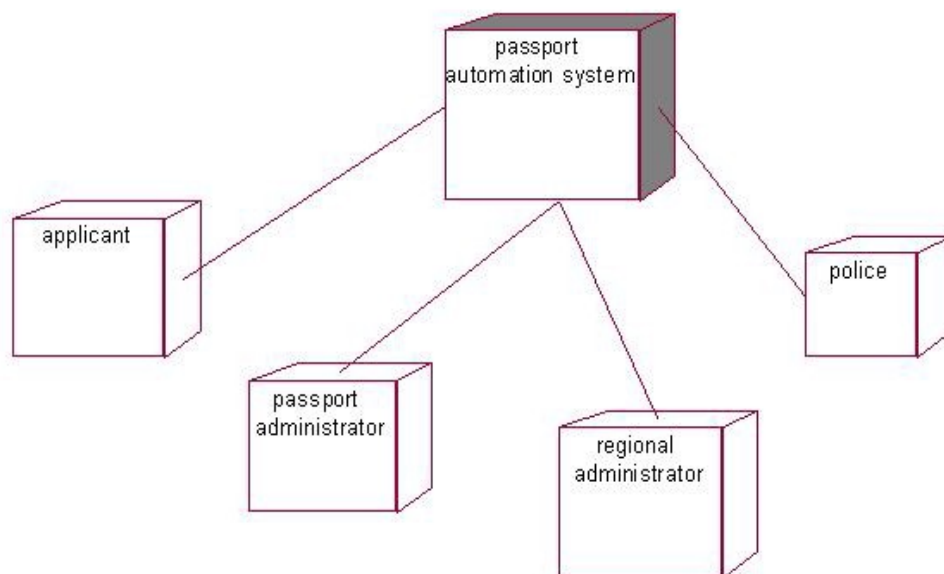
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 3

BOOK BANK SYSTEM

Date:

AIM:

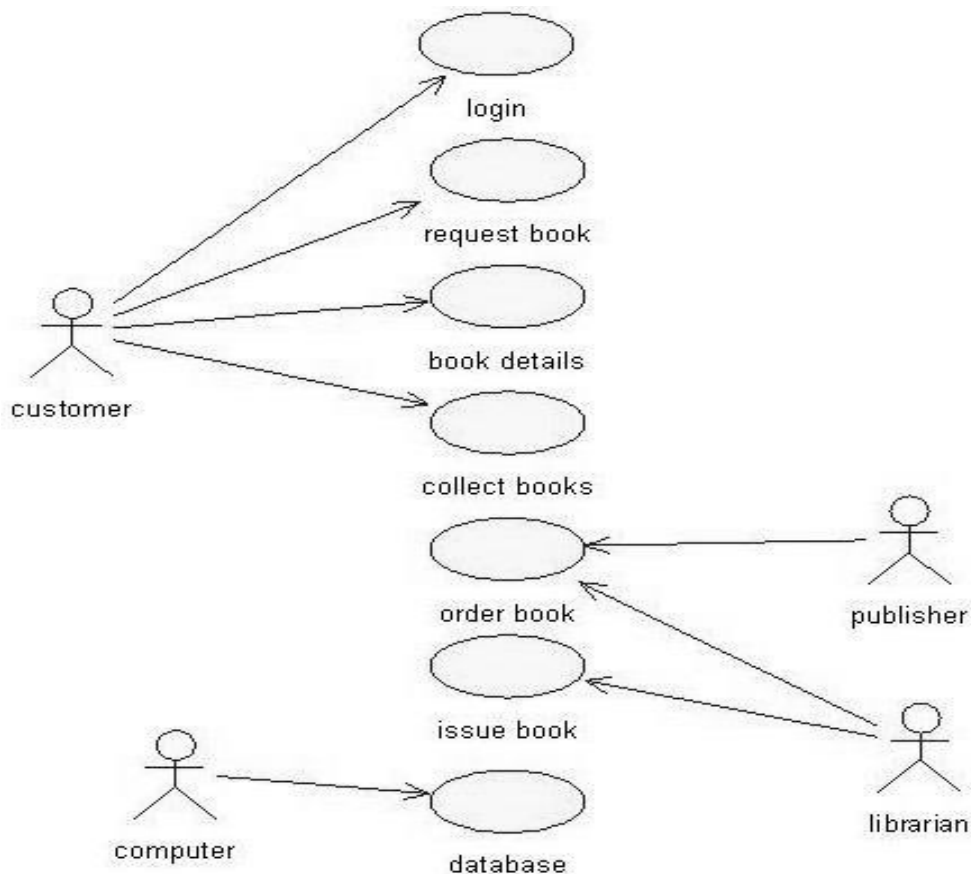
To create a system to perform book bank operation using Argo UML.

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 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.

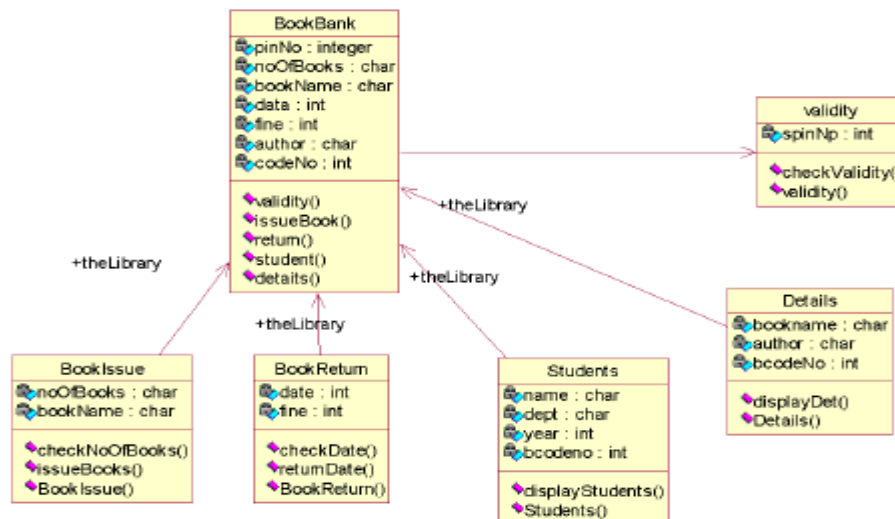
USE CASE DIAGRAM

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.



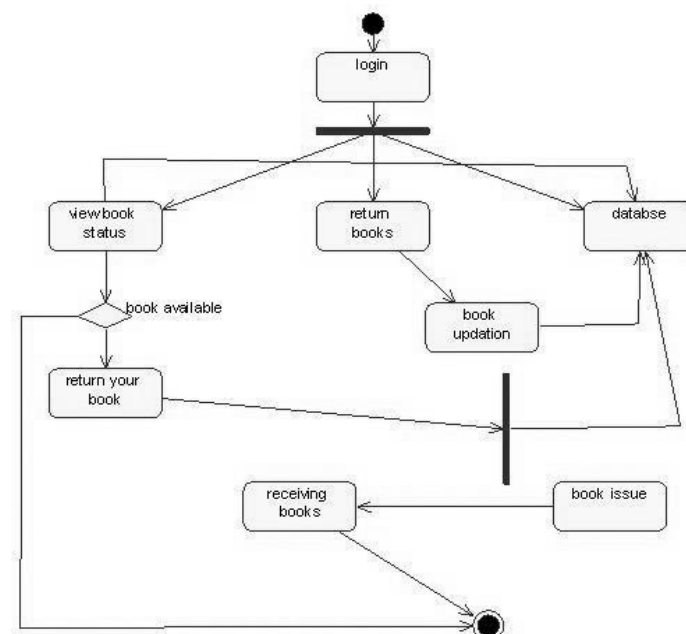
CLASS DIAGRAM

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.



ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of step wise 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.

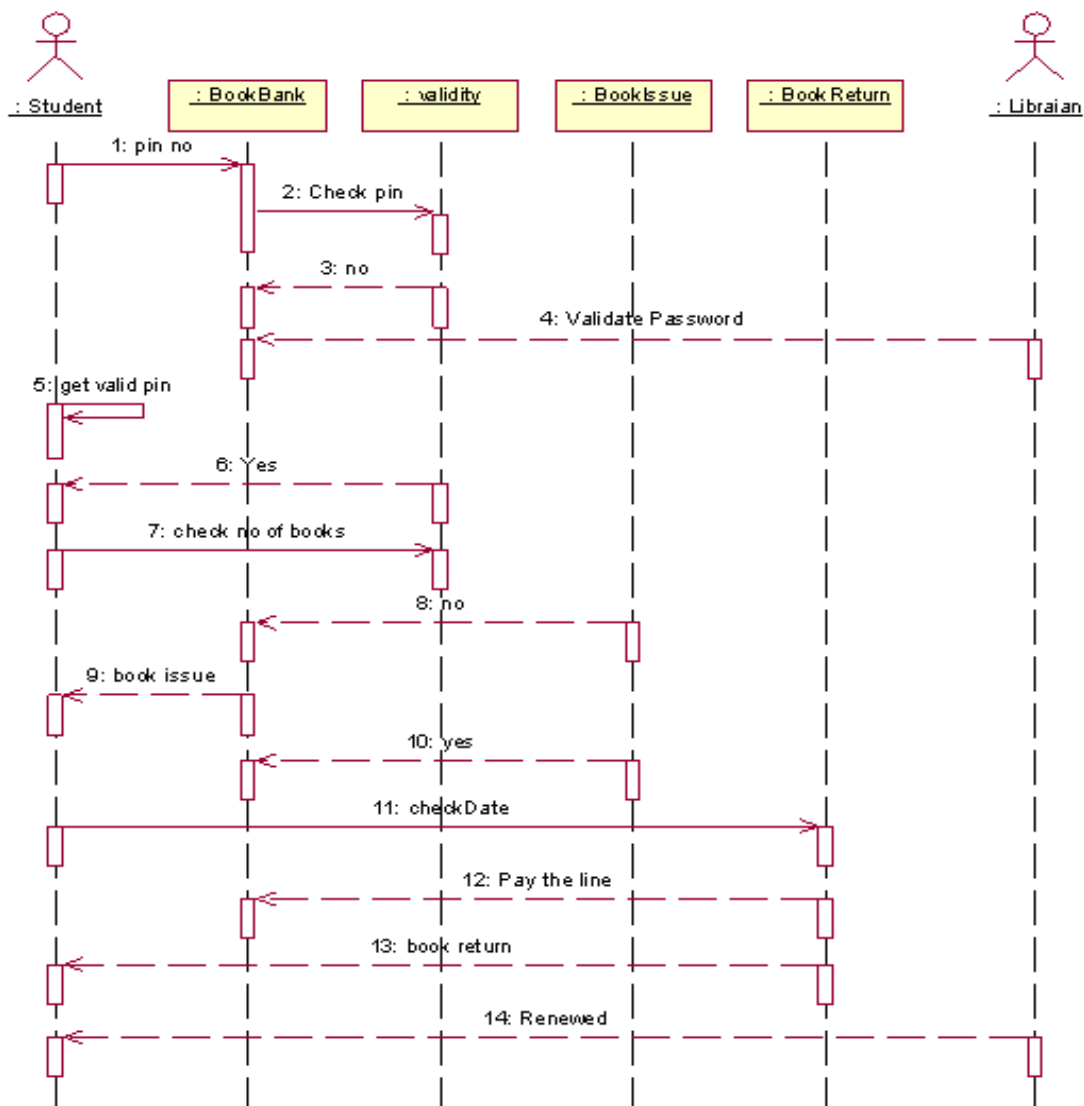


SEQUENCE DIAGRAM

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.

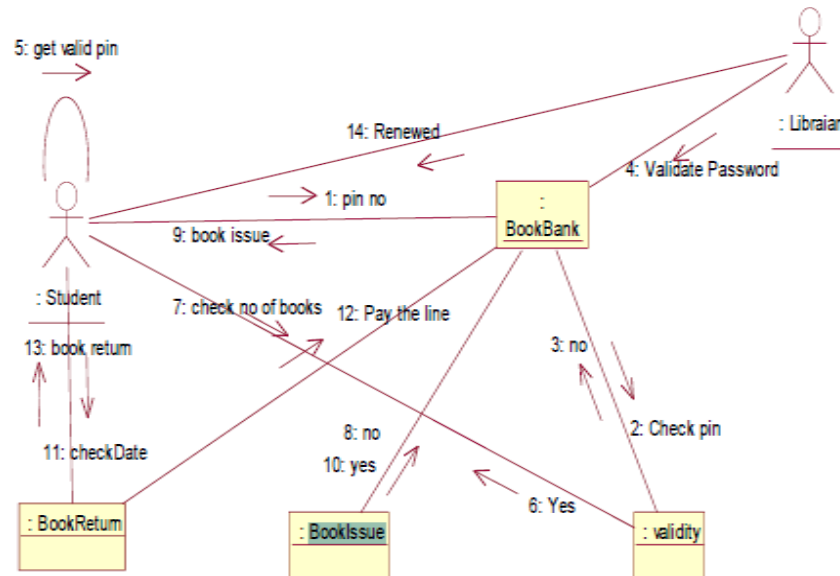
An event also is considered to be any action by an object that sends information. The event line represents a message sent from one object to another, in which the “from” object is requesting an operation be performed by the “to” object. The “to” object performs the operation using a method that the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



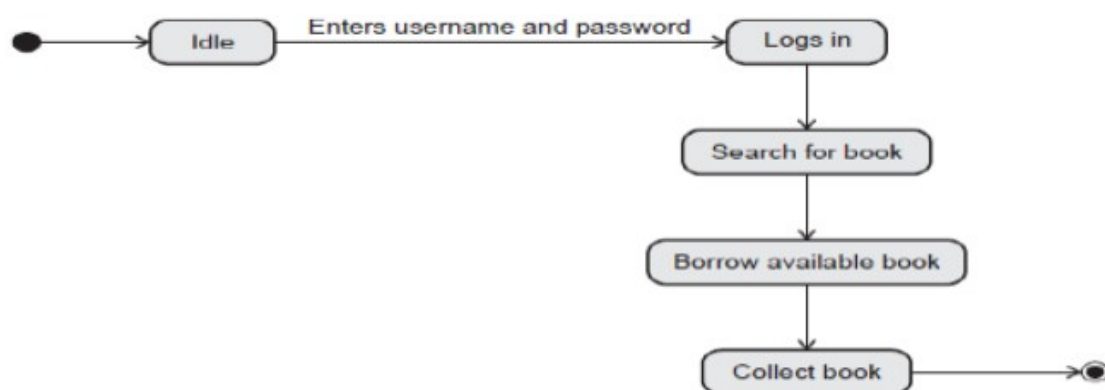
COLLABORATION DIAGRAM

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.



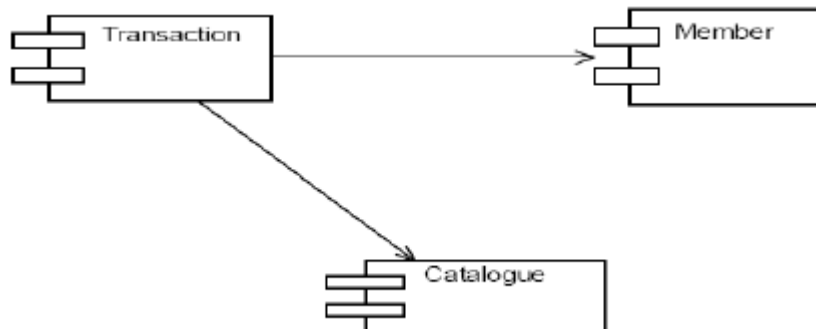
STATE CHART DIAGRAM

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.



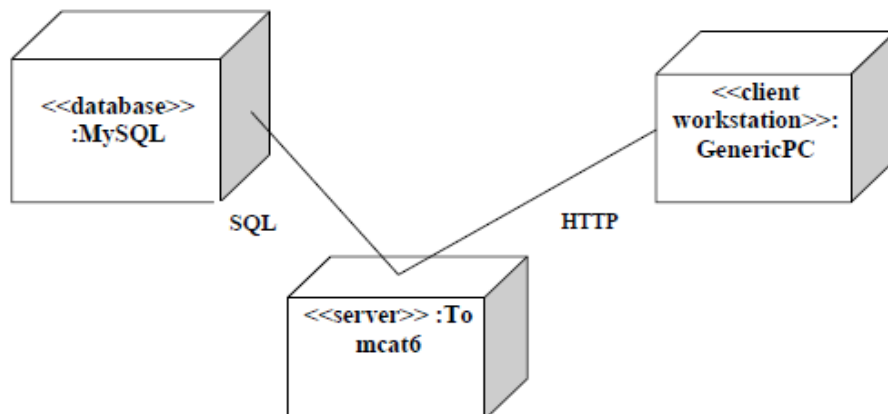
COMPONENT DIAGRAM

Component diagrams are used to visualize the organization and relationships



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 4

EXAM REGISTRATION SYSTEM

Date:

AIM:

To create a system to perform the Exam Registration system using Argo UML.

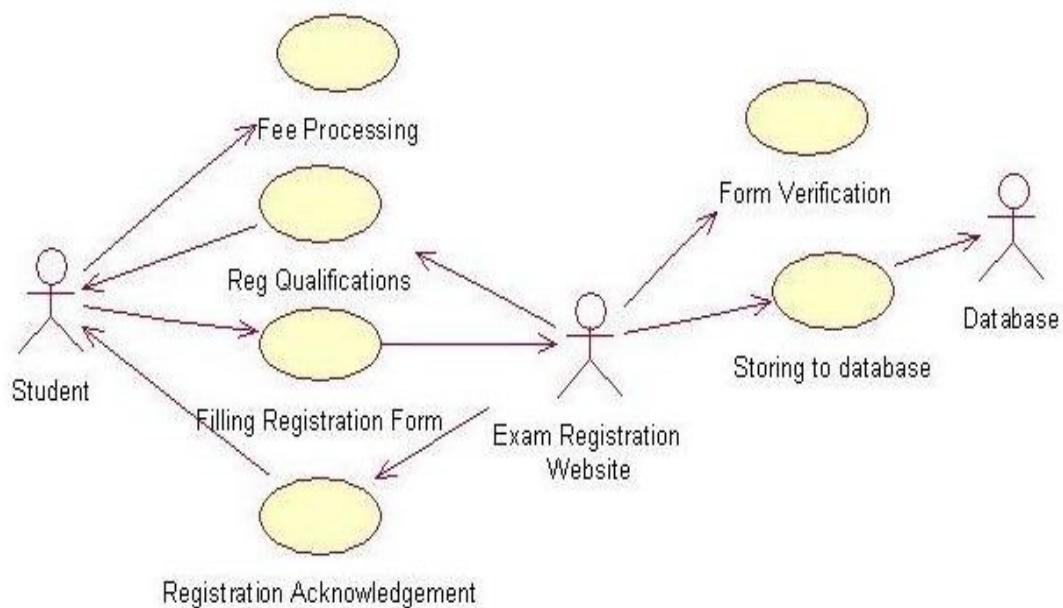
PROBLEM STATEMENT:

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.

USE CASE DIAGRAM:

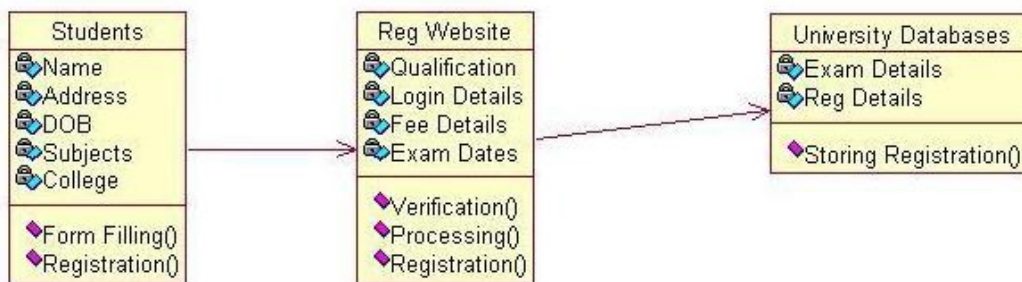
The Exam Registration use cases in our system are:

1. Fee Processing
2. Qualification
3. Registration Form
4. Registration Acknowledgment
5. Form Verification
6. Database



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.



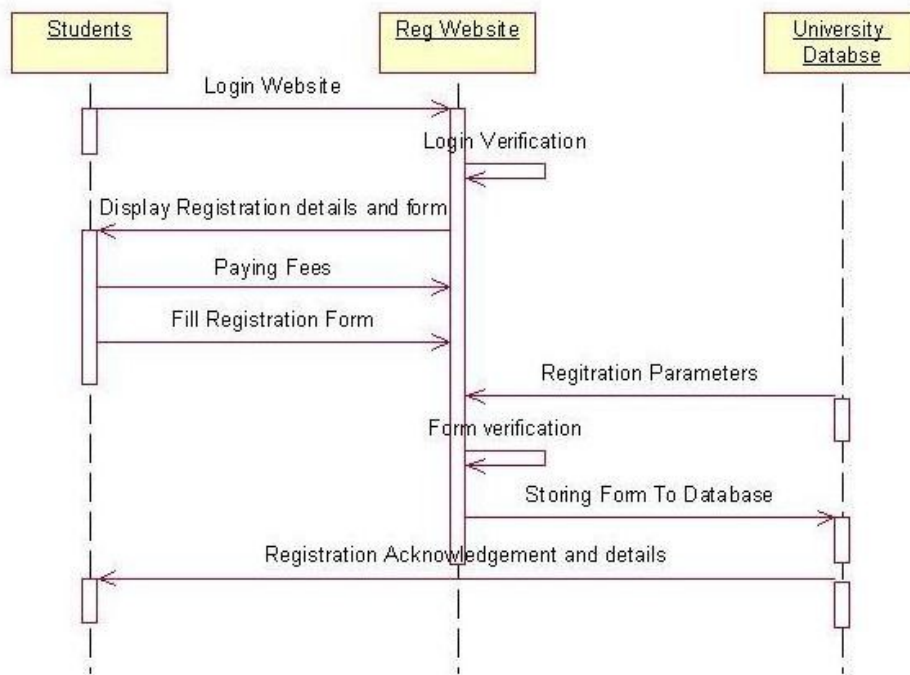
ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of step wise 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.



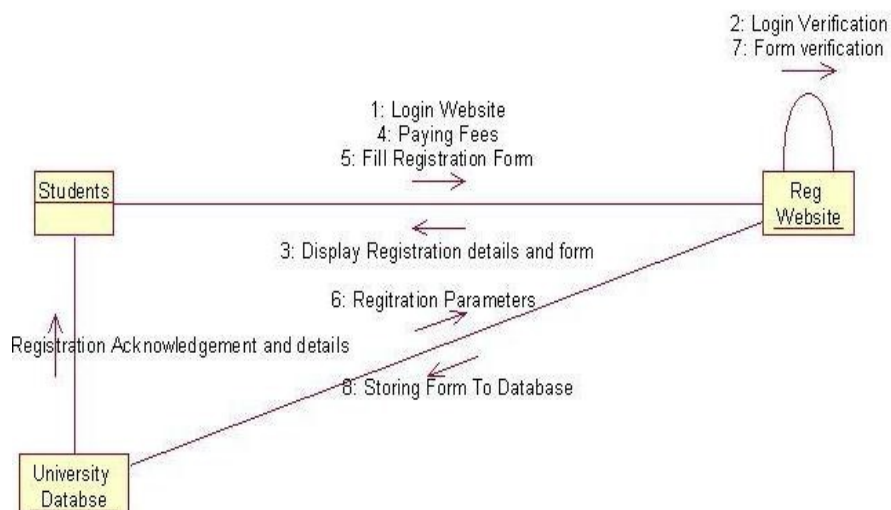
SEQUENCE DIAGRAM:

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.



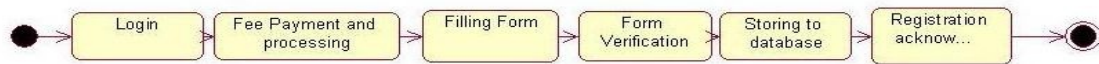
COLLABORATION DIAGRAM

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.



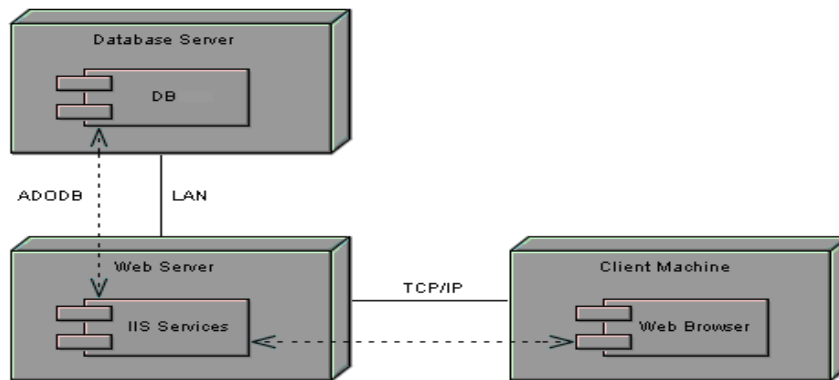
STATE CHART DIAGRAM

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.



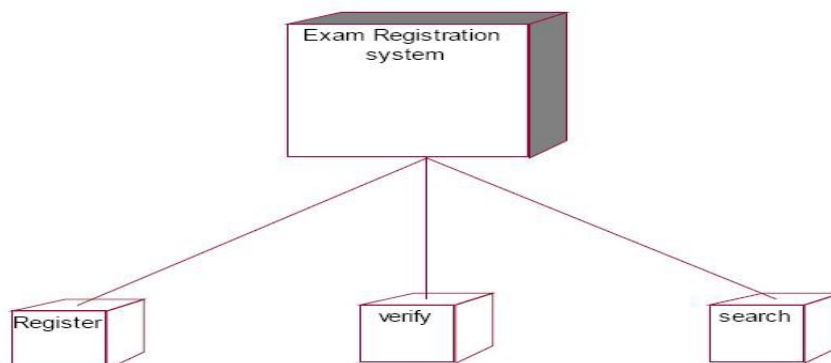
COMPONENT DIAGRAM:

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



DEPLOYMENT DIAGRAM:

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



RESULT:

Ex. No: 5

STOCK MAINTENANCE

Date:

AIM:

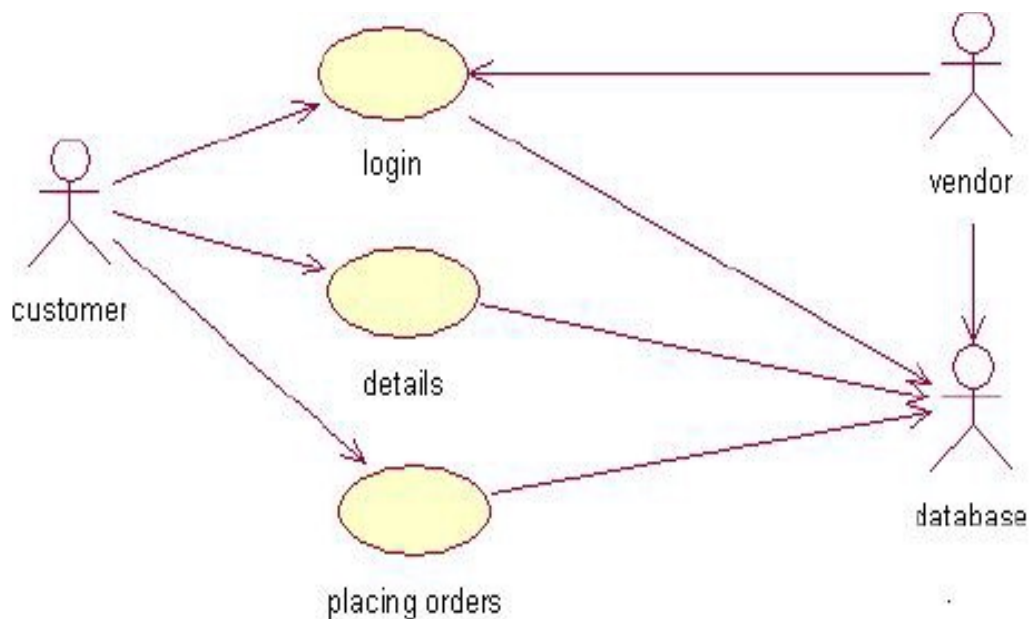
To create a system to perform the Stock Maintenance using Argo UML.

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.

USE CASE DIAGRAM

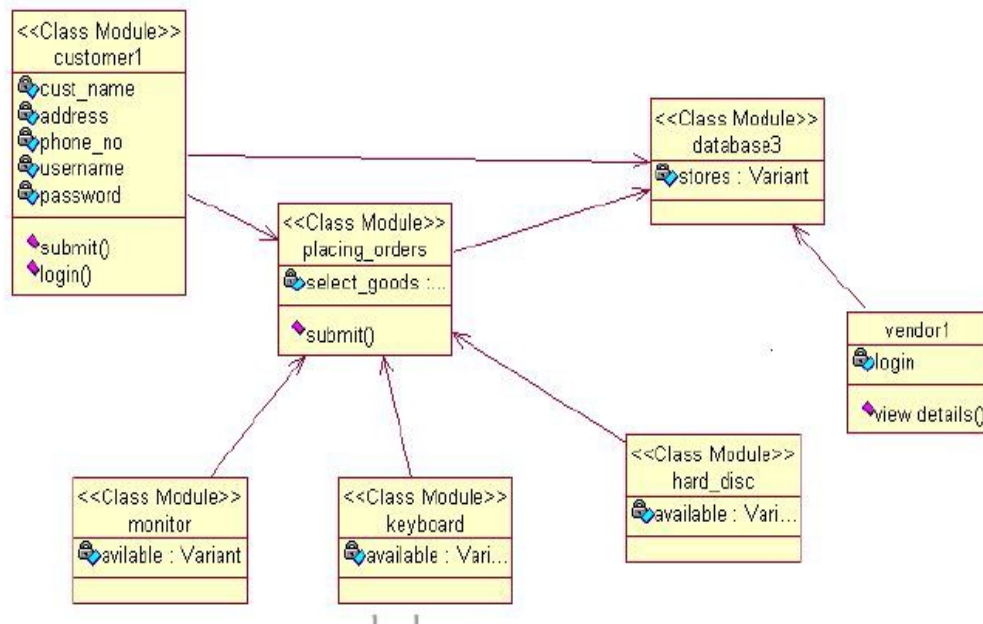
The functionality of a system can be described in a number of different use-cases, each of which represents a specific flow of events in a system. It is a graph of actors, a set of use-cases enclosed in a boundary, communication, associations between the actors and the use-cases, and generalization among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



CLASS DIAGRAM

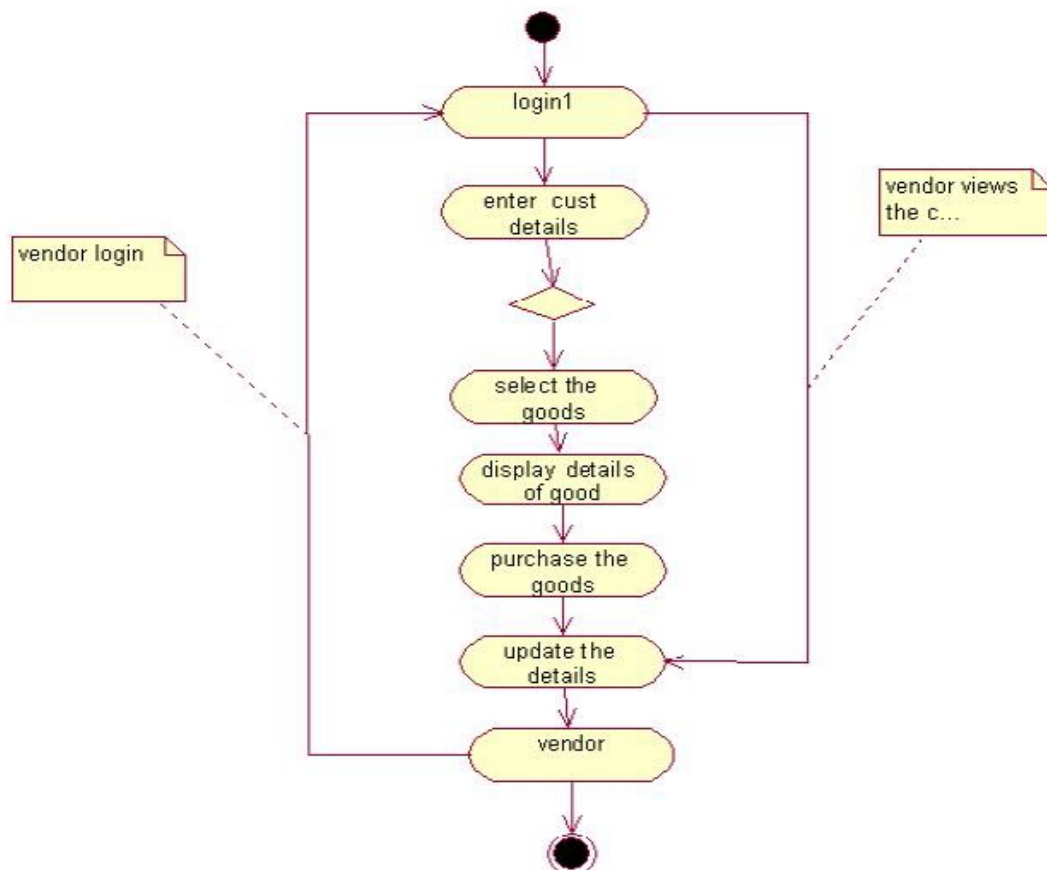
A class diagram describes the type of objects in system and various kinds of relationships that exists among them.

Class diagrams and collaboration diagrams are alternate representations of object models.



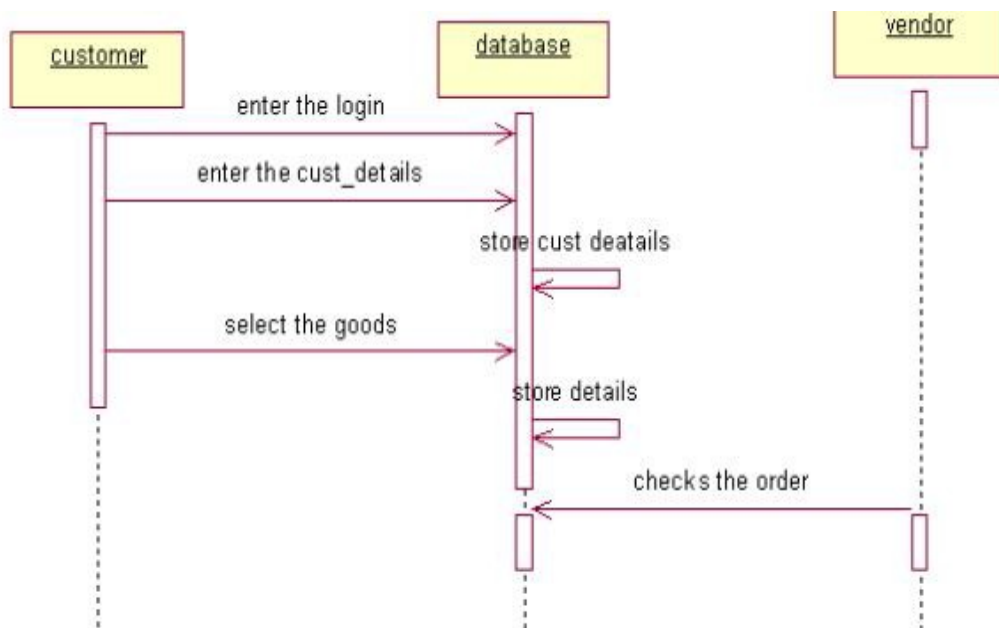
ACTIVITY DIAGRAM

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.



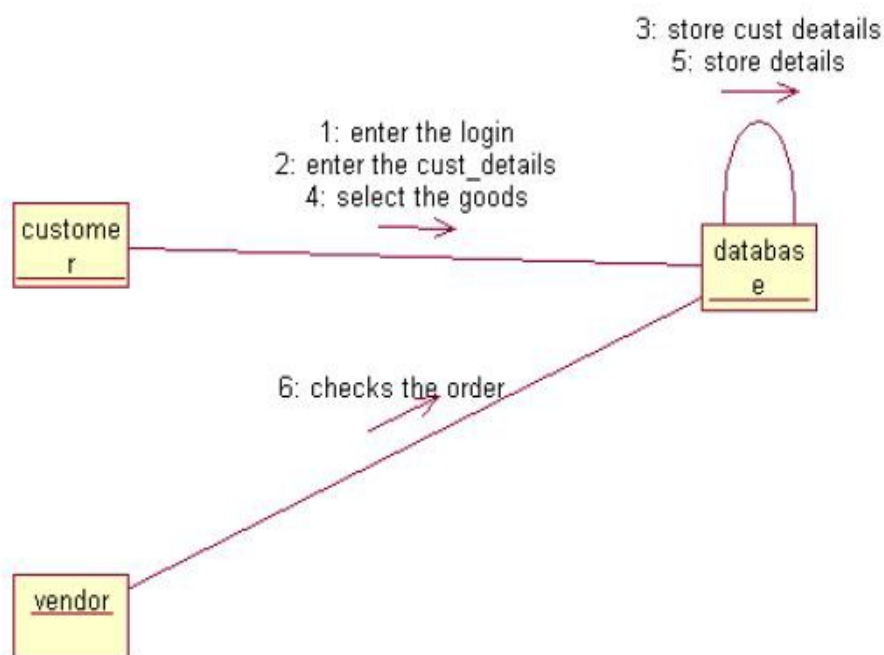
SEQUENCE DIAGRAM

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



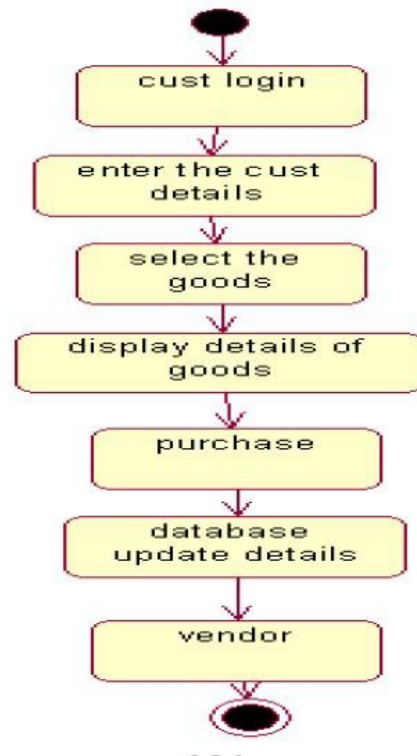
COLLABORATION DIAGRAM

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.



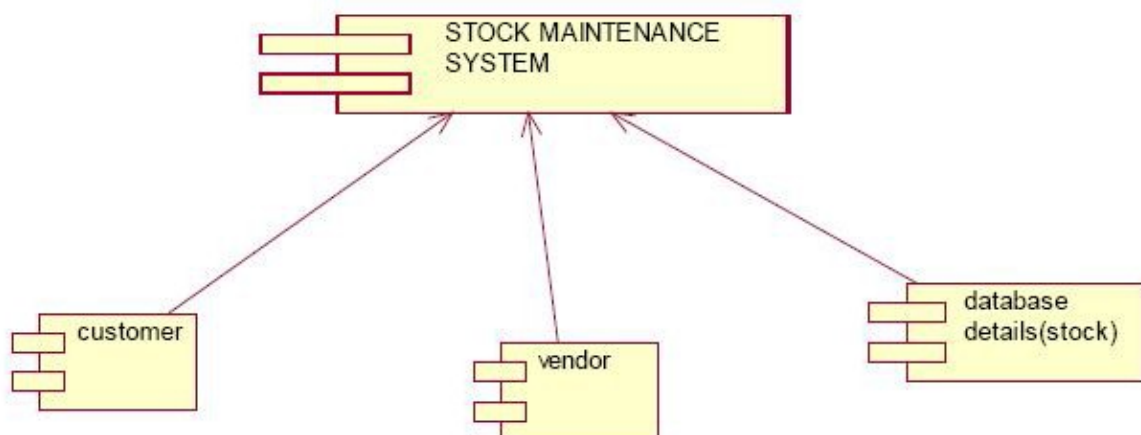
STATE CHART DIAGRAM

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.



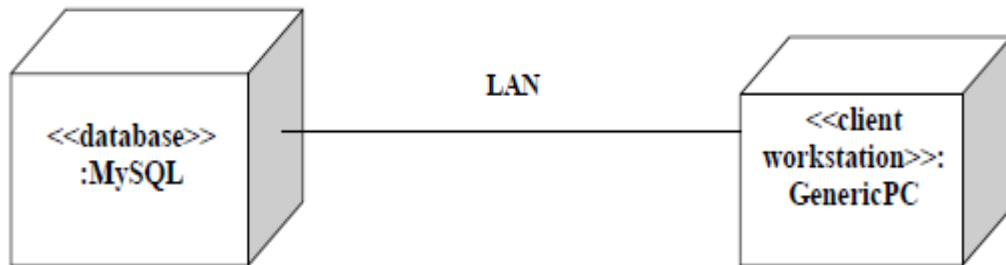
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 6

ONLINE COURSE RESERVATION SYSTEM

Date:

AIM

To design an object oriented model for Online Course Reservation System using Argo UML.

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.

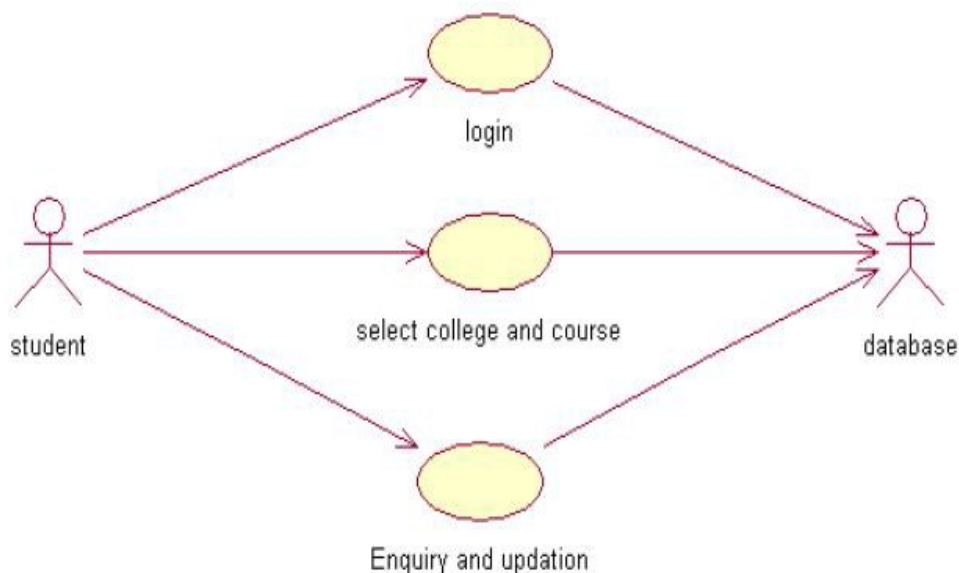
b. The system should maintain a list of professor who is teaching the course. At the end of the course the student must be provided with the certificate for the completion of the course.

USE CASE DIAGRAM

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

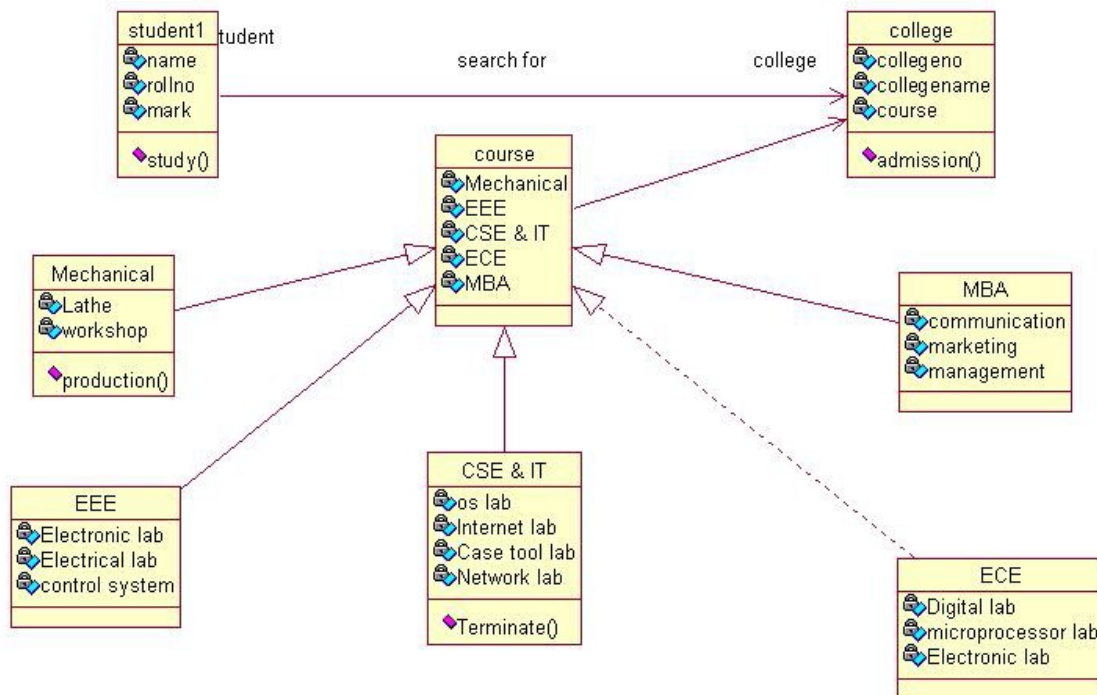
b. Use case is a set of scenarios together by a common user goal.

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



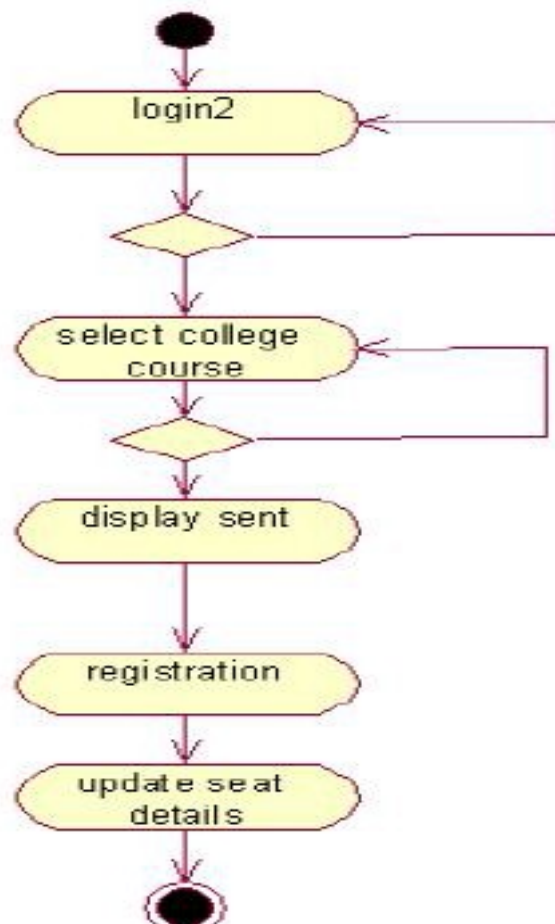
CLASS DIAGRAM:

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



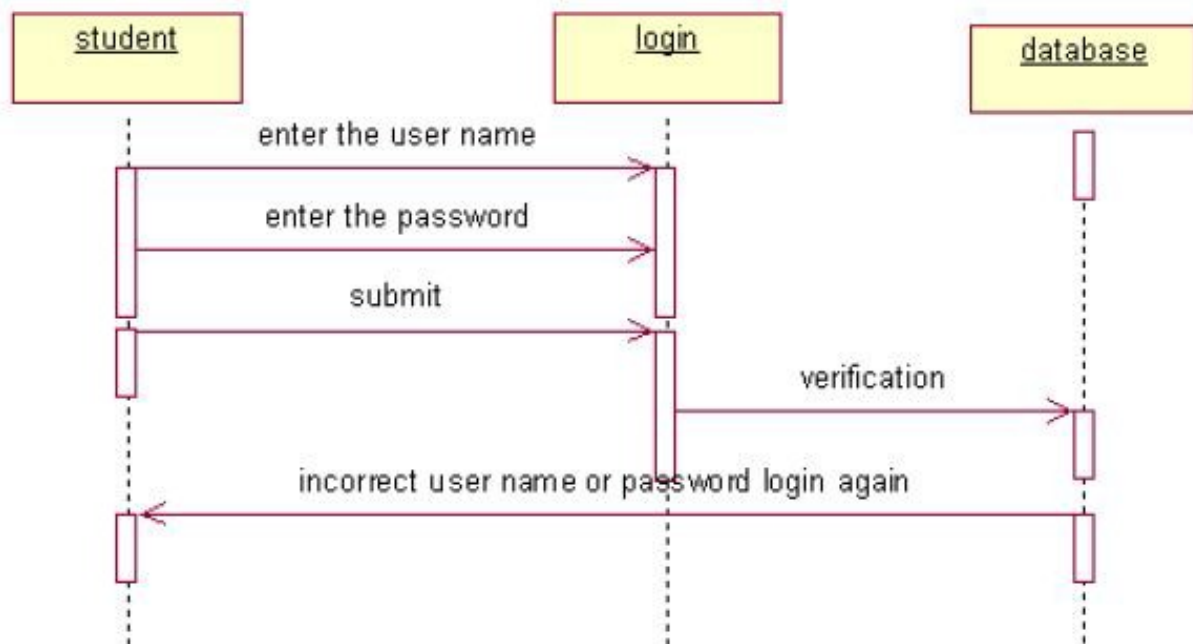
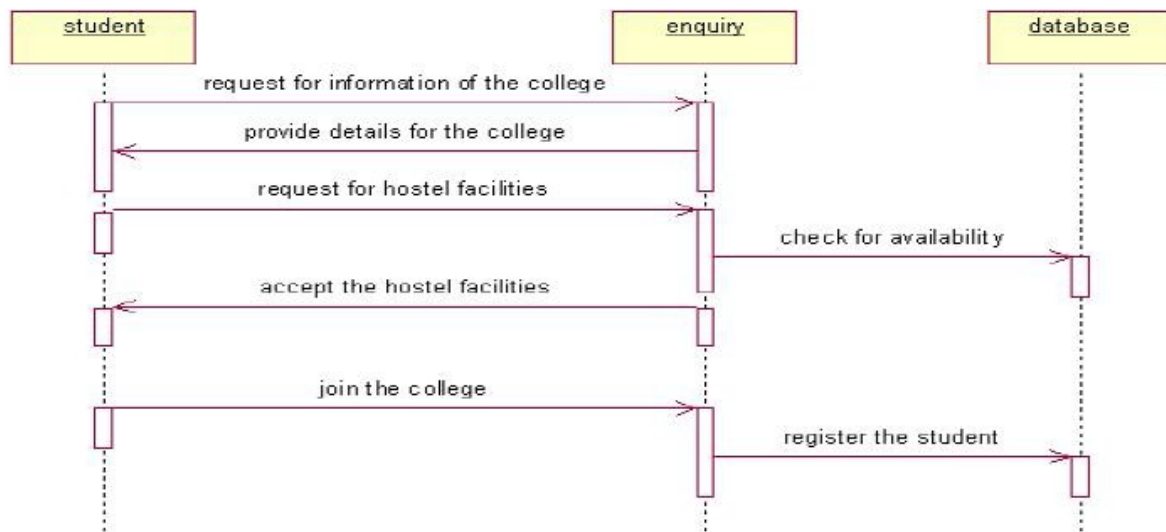
ACTIVITY DIAGRAM

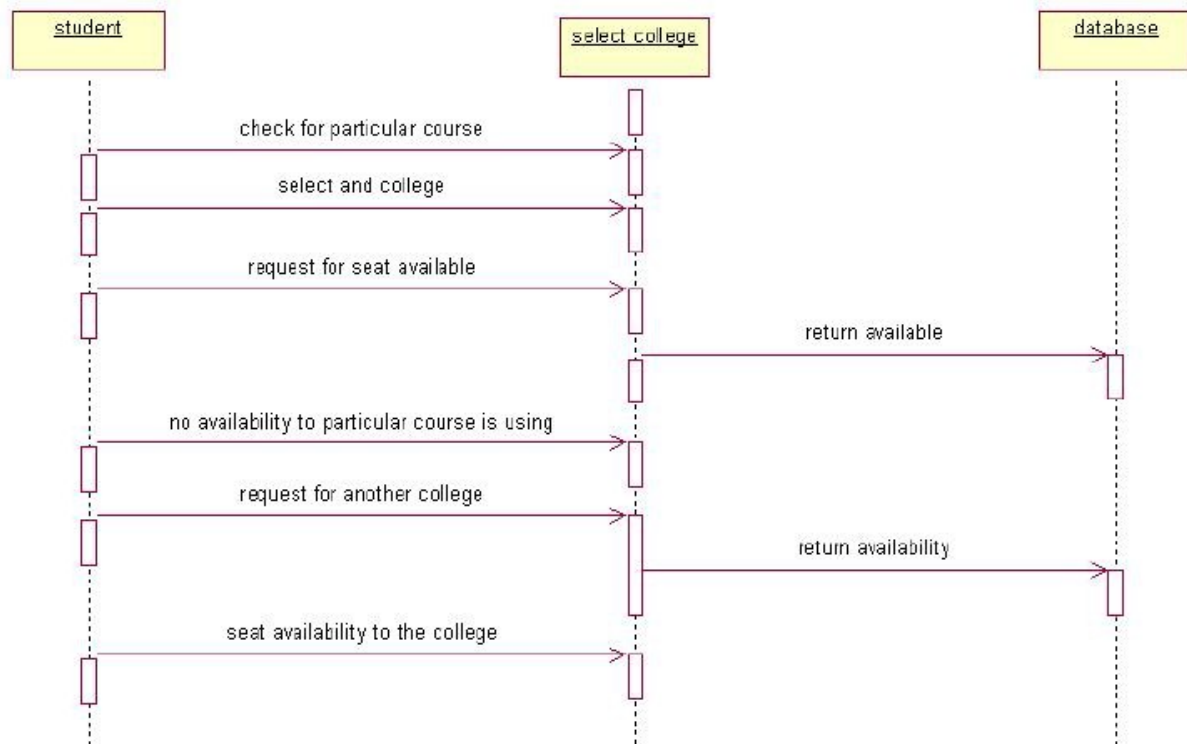
It includes all the activities of particular project and various steps using join and forks.



SEQUENCE DIAGRAM

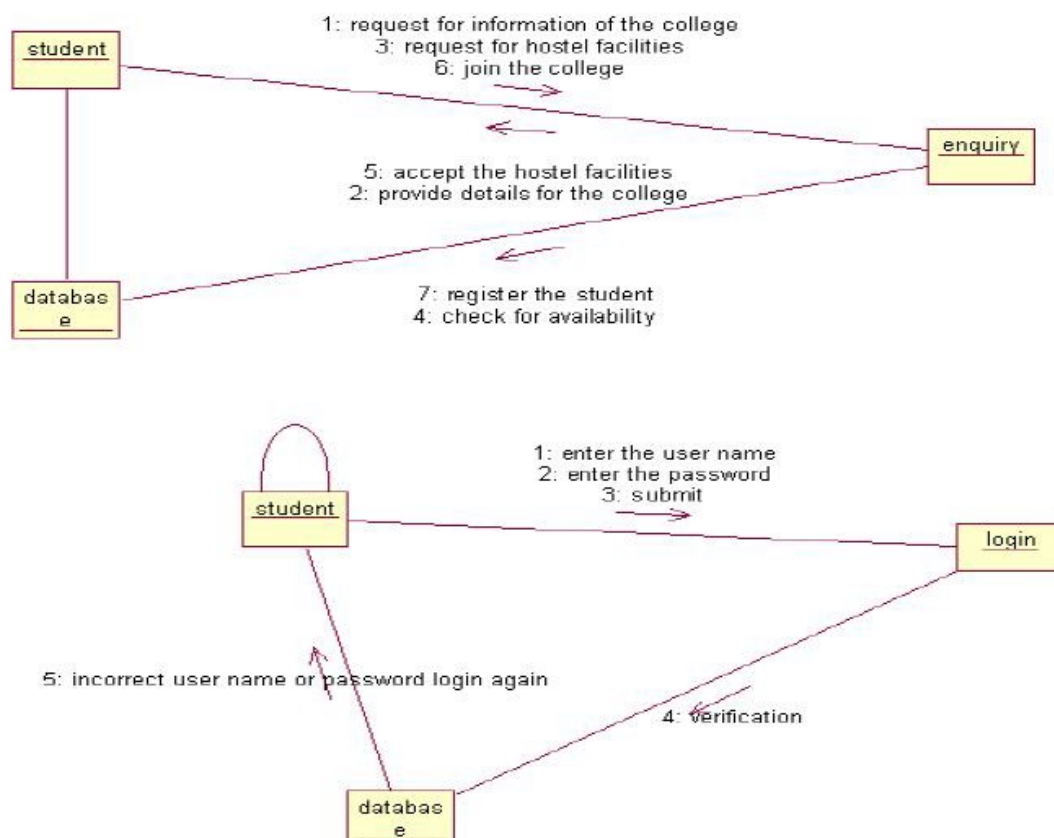
A sequence diagram is one that includes the object of the projects and tells the lifetimes and also various action performed between objects.

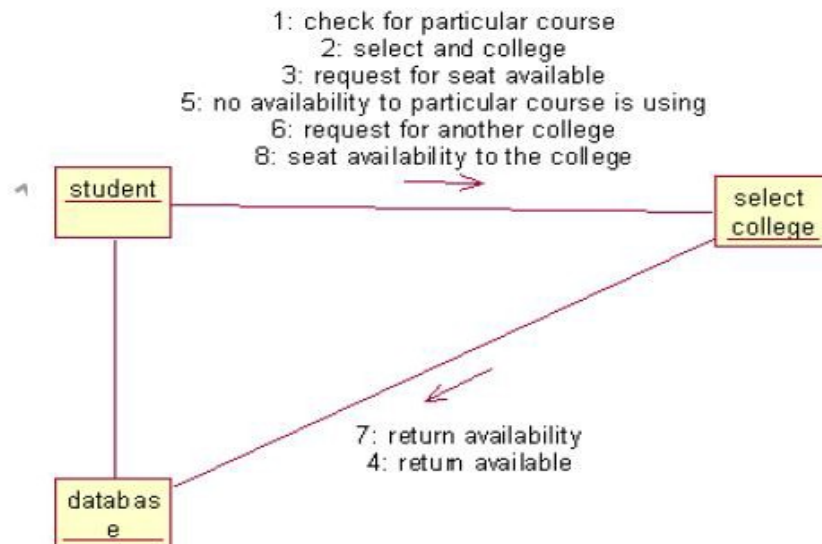




COLLABORATION DIAGRAM

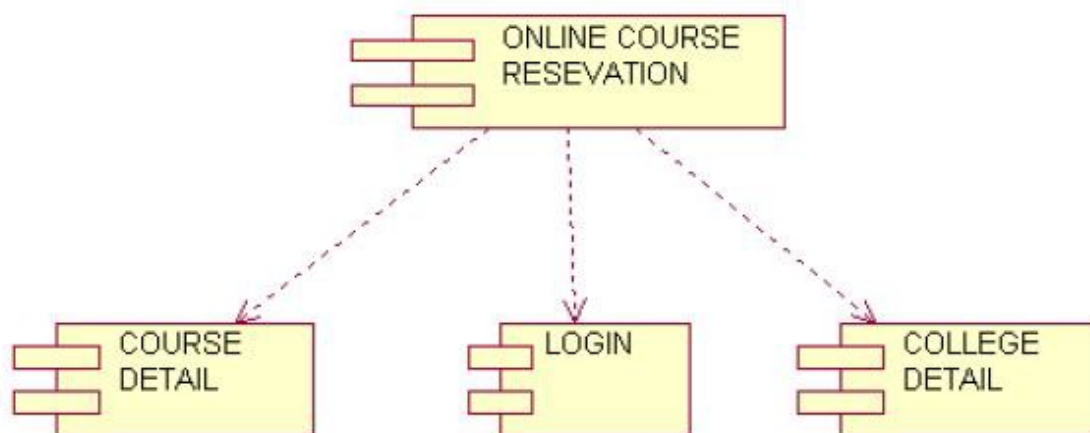
It is same as the sequence diagram that involved the project with the only difference that we give the project with the only difference that we give sequence number to each process.





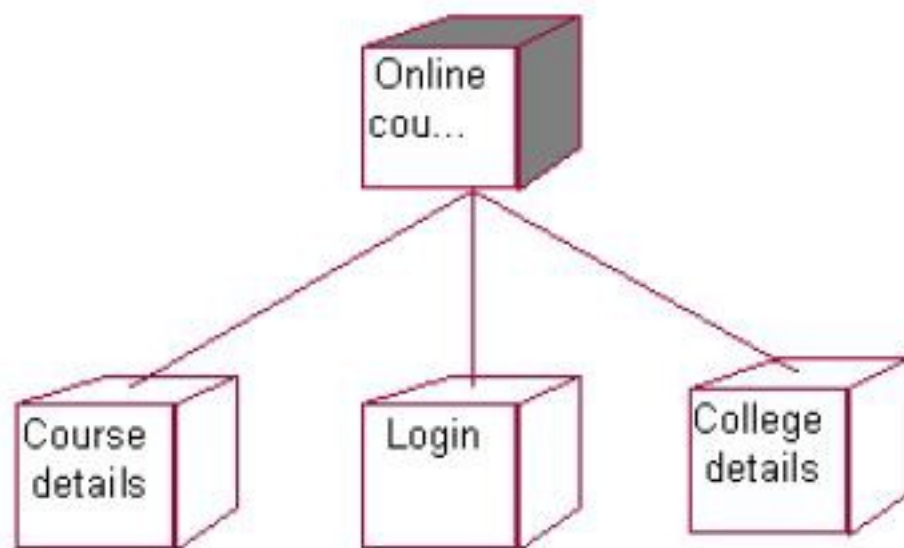
COMPONENT DIAGRAM

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



RESULT:

Ex. No: 7

AIRLINE / RAILWAY RESERVATION SYSTEM

Date:

AIM

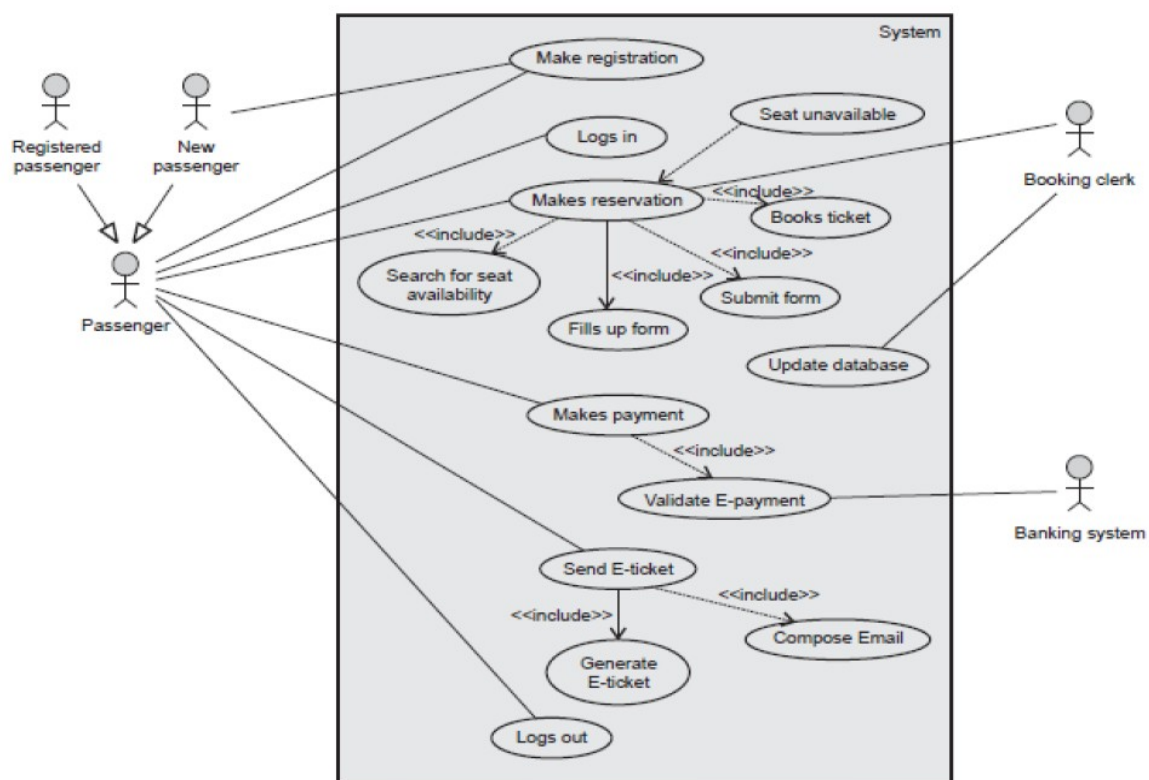
To develop the Airline/Railway reservation System using Argo UML

PROBLEM ANALYSIS

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 canceling the process.

USE CASE DIAGRAM

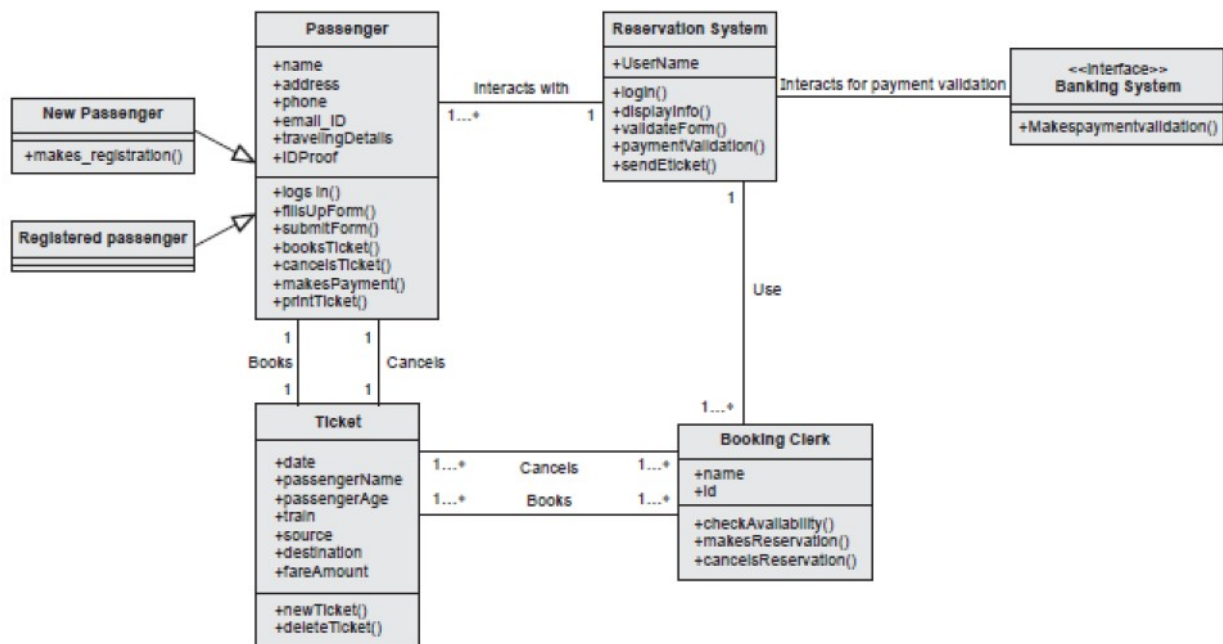
The passenger can view the status of the reserved tickets. So the passenger can confirm his/her travel.



CLASS DIAGRAM

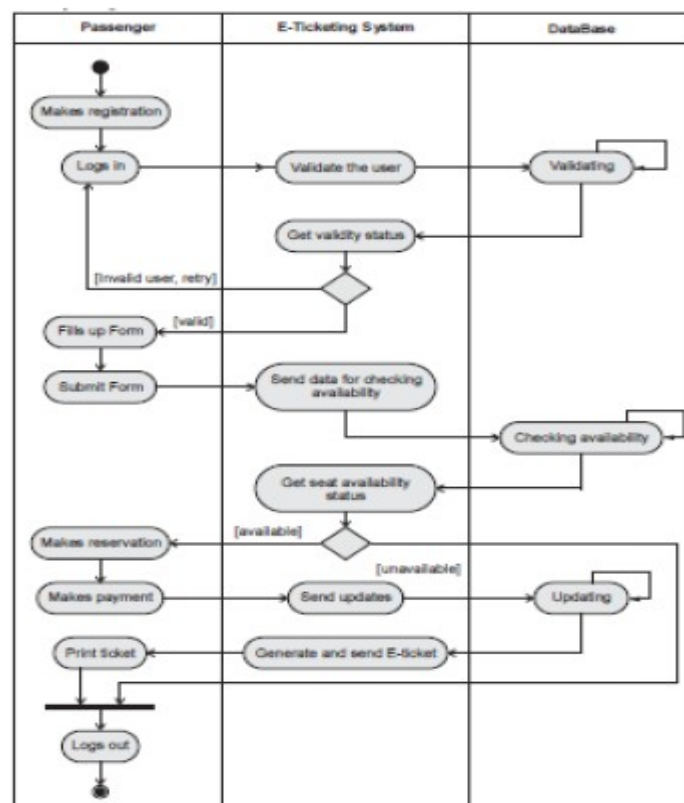
The online ticket reservation system makes use of the following classes:

1. Passenger
2. Reservation System
3. Ticket
4. Booking Clerk



ACTIVITY DIAGRAM

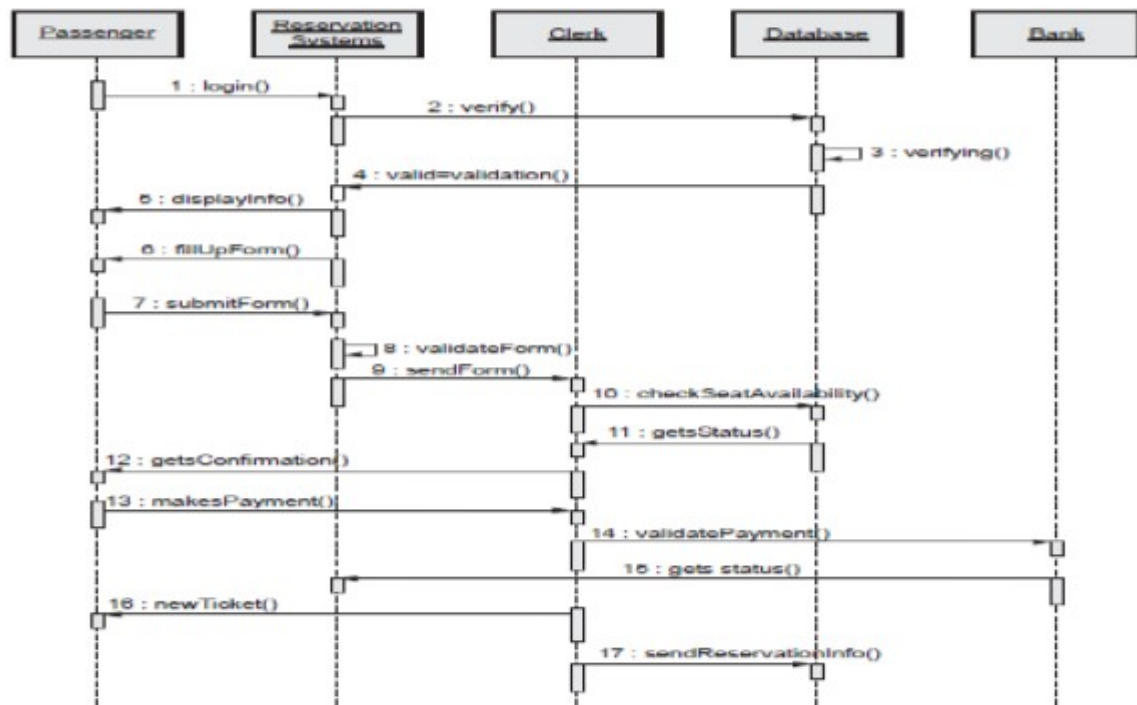
Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.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



SEQUENCE DIAGRAM

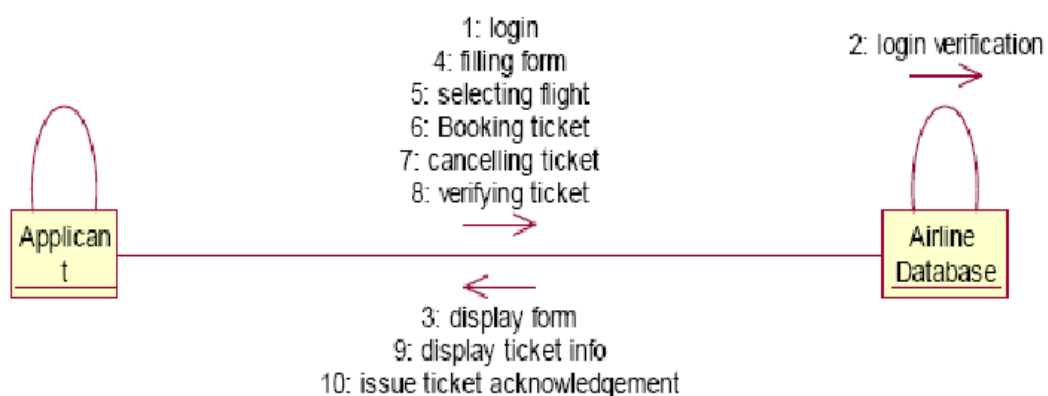
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

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



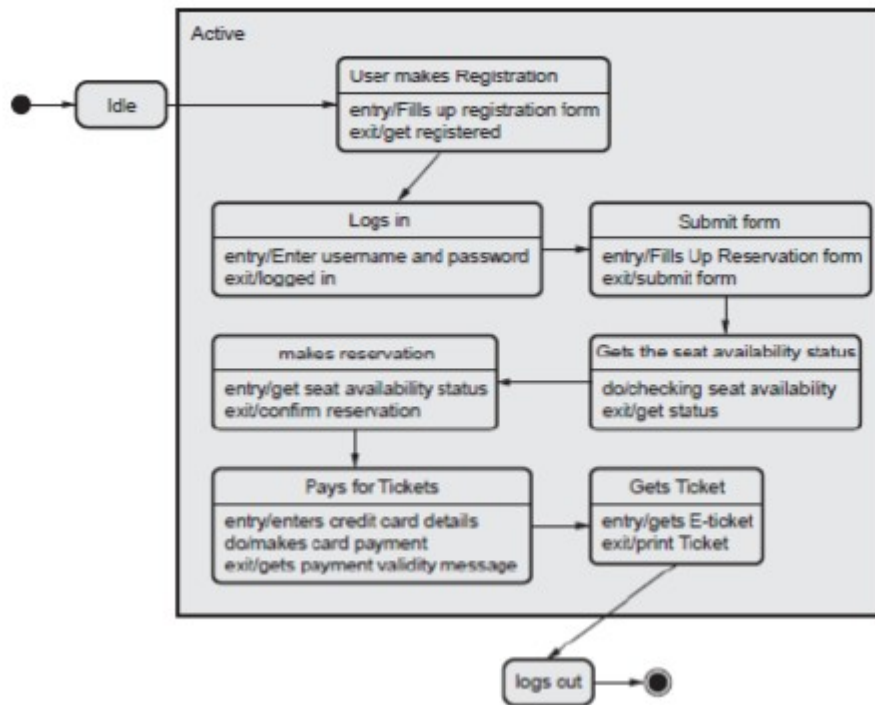
COLLABORATION DIAGRAM

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 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.



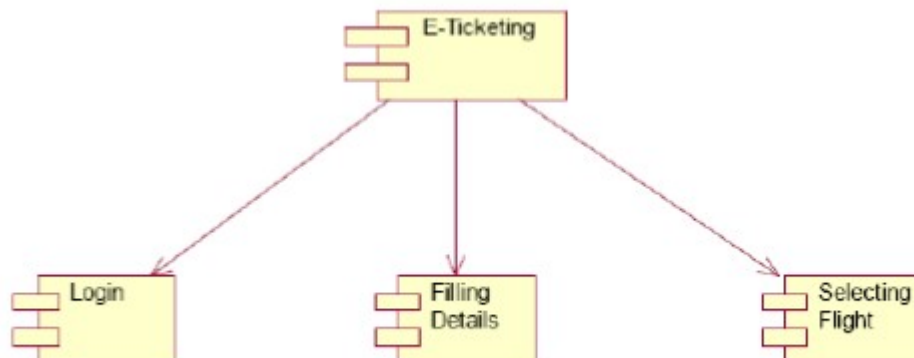
STATE CHART DIAGRAM

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.



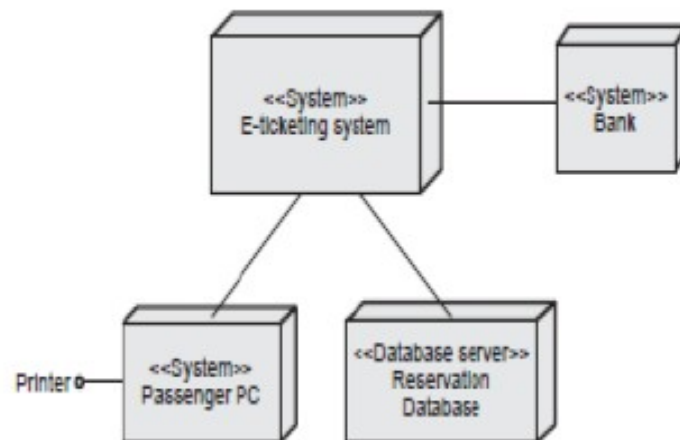
COMPONENT DIAGRAM

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.



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 8

SOFTWARE PERSONNEL MANAGEMENT SYSTEM

Date:

AIM:

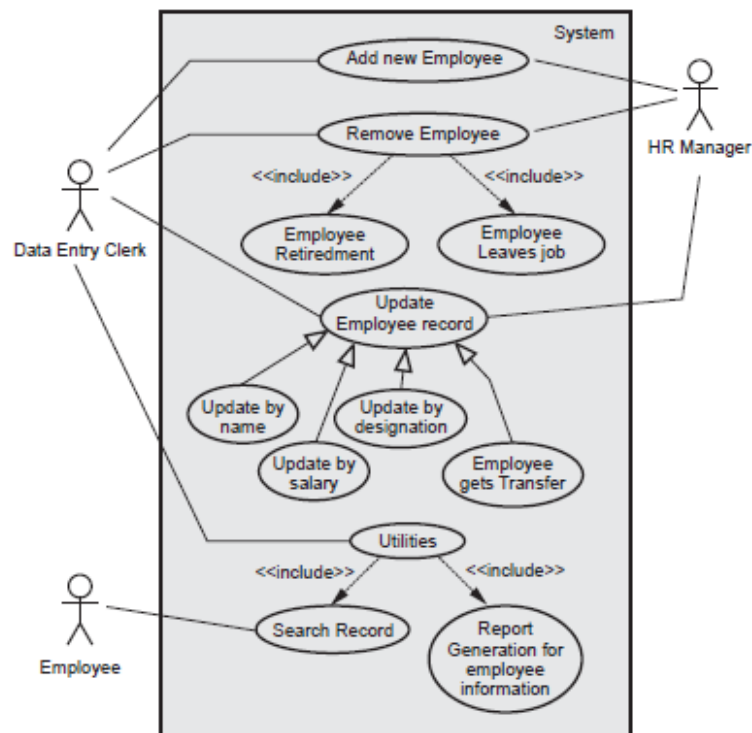
To implement a software for Software Personnel Management System using Argo UML.

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.

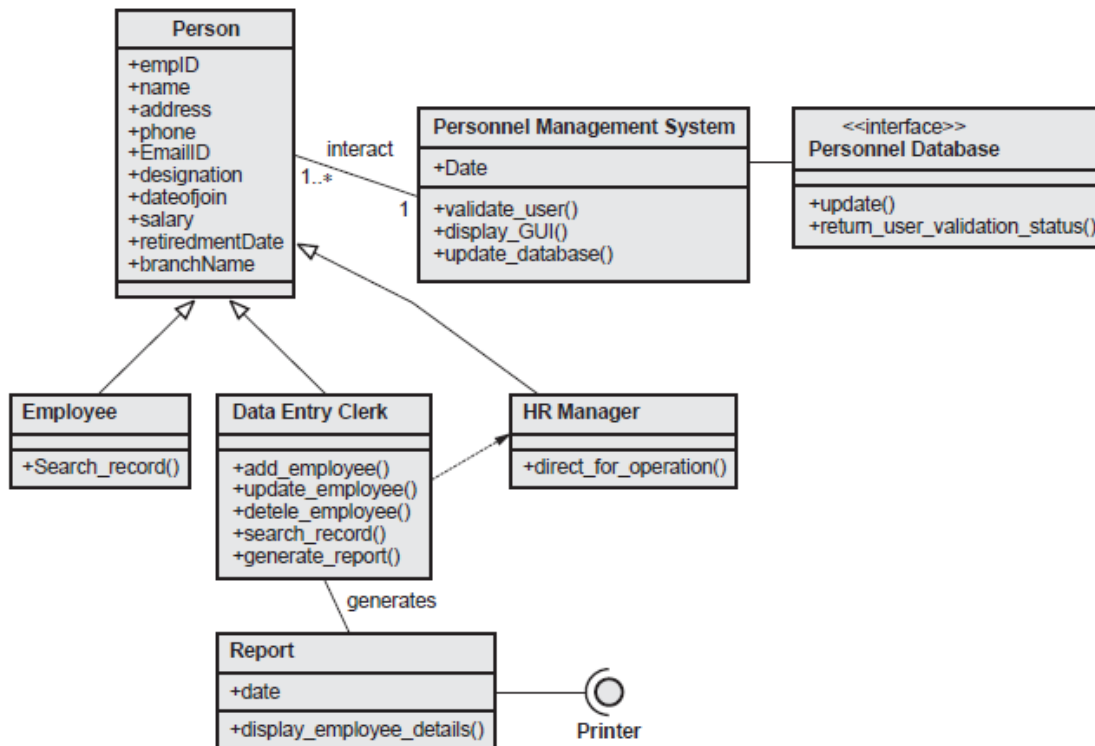
USE CASE DIAGRAM

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.



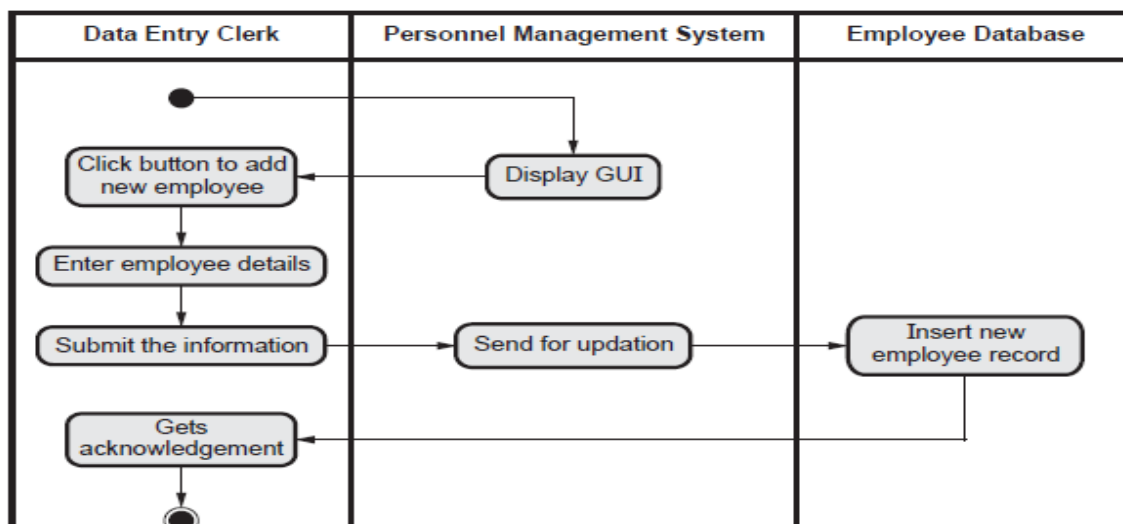
CLASS DIAGRAM

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.



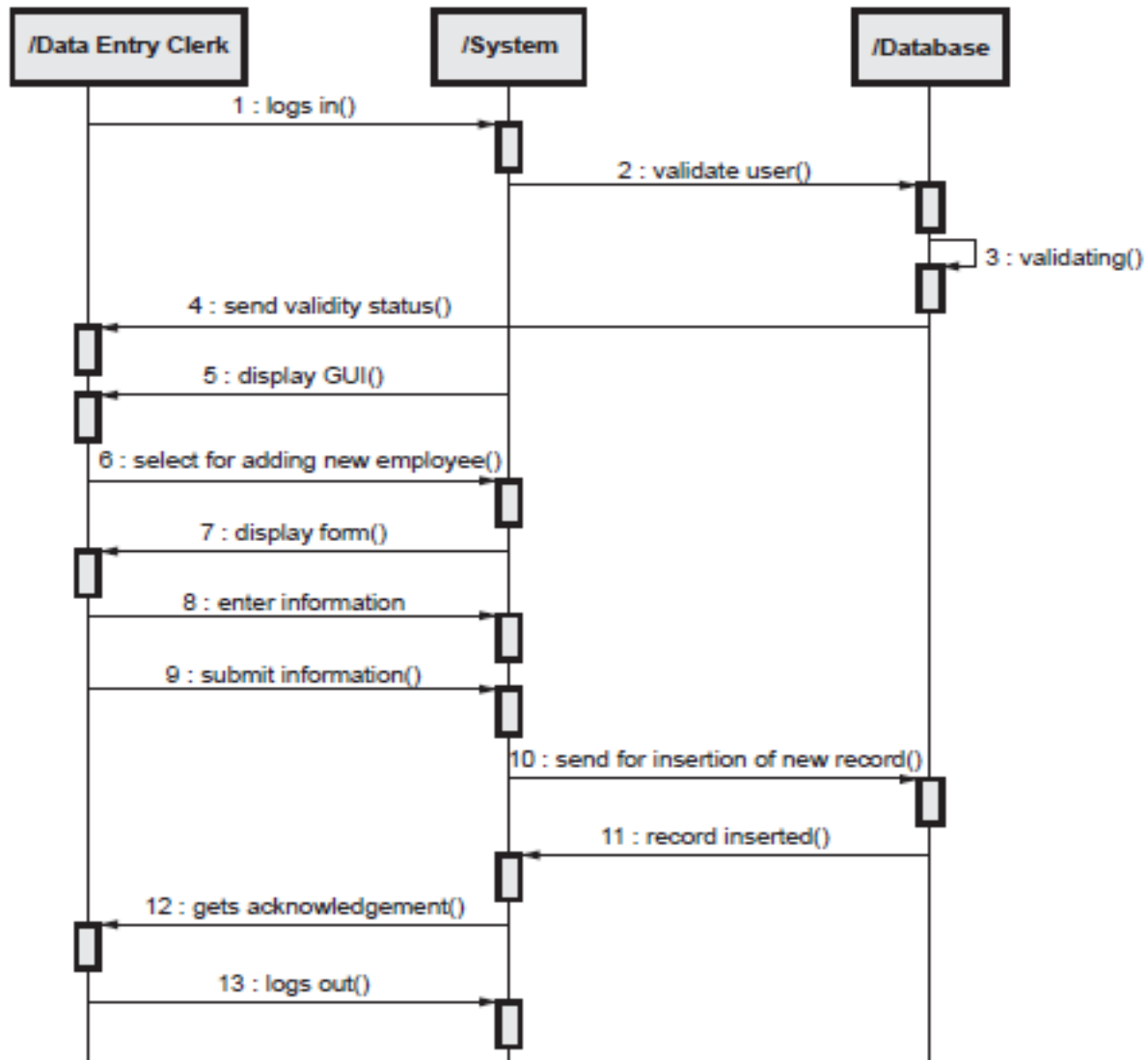
ACTIVITY DIAGRAM

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.



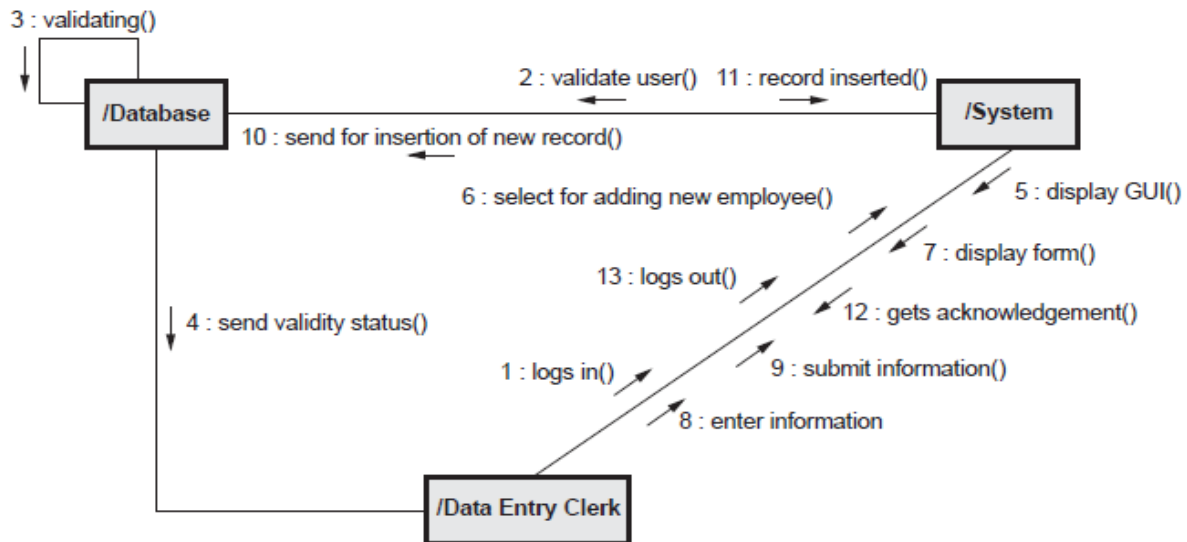
SEQUENCE DIAGRAM

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.



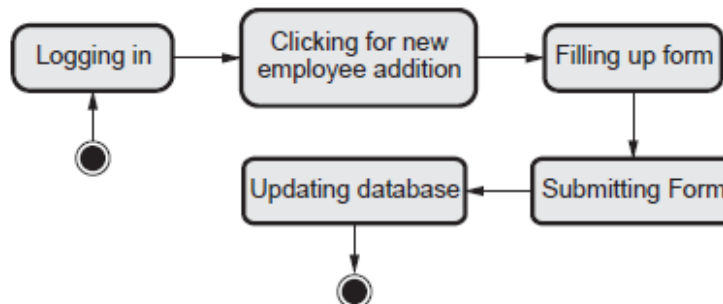
COLLABORATION DIAGRAM

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 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.



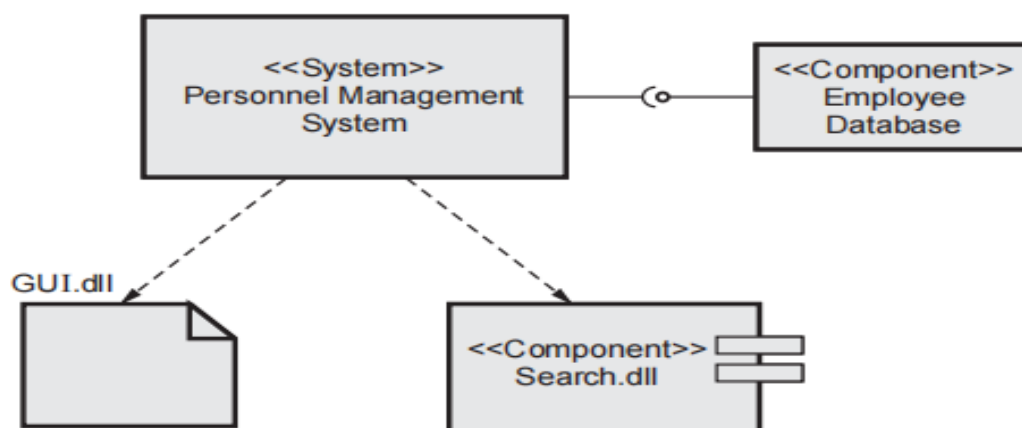
STATE CHART DIAGRAM

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.



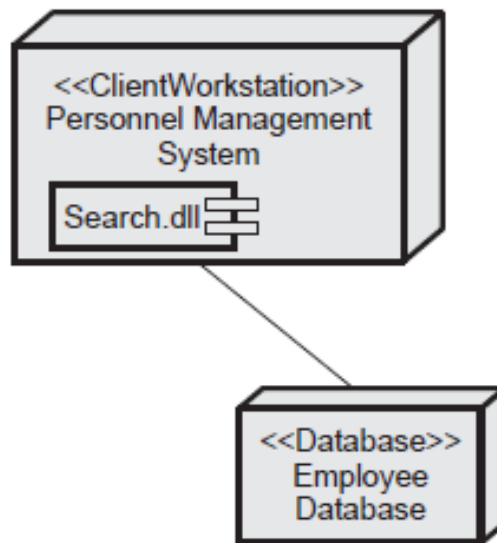
COMPONENT DIAGRAM

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



DEPLOYMENT 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.



RESULT:

Ex. No: 9

CREDIT CARD PROCESSING

Date:

AIM:

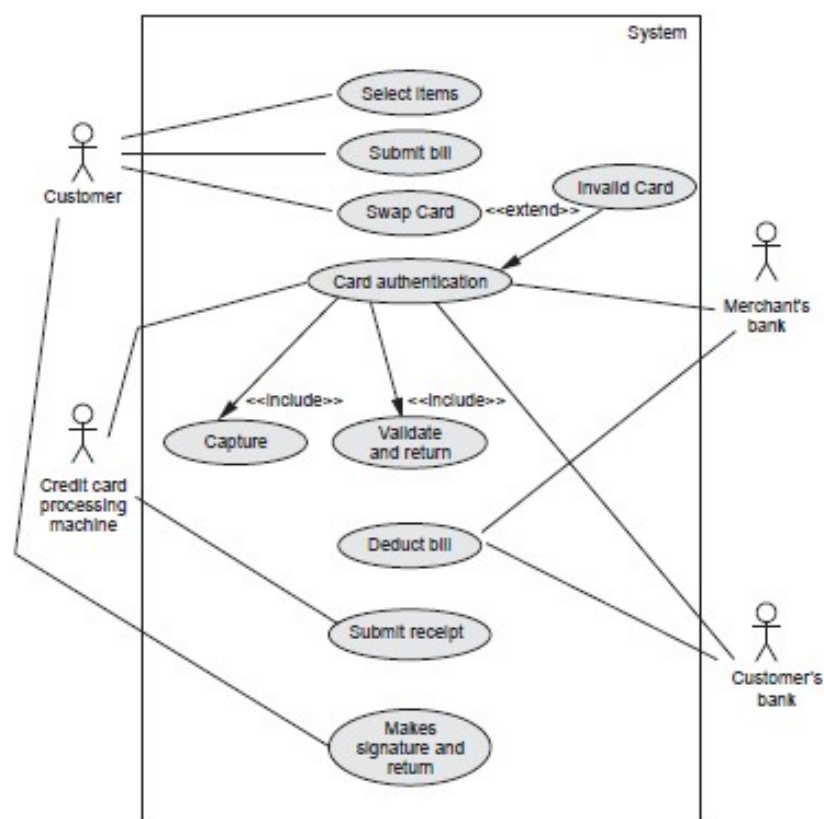
To create a system to perform the credit card processing using Argo UML..

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:

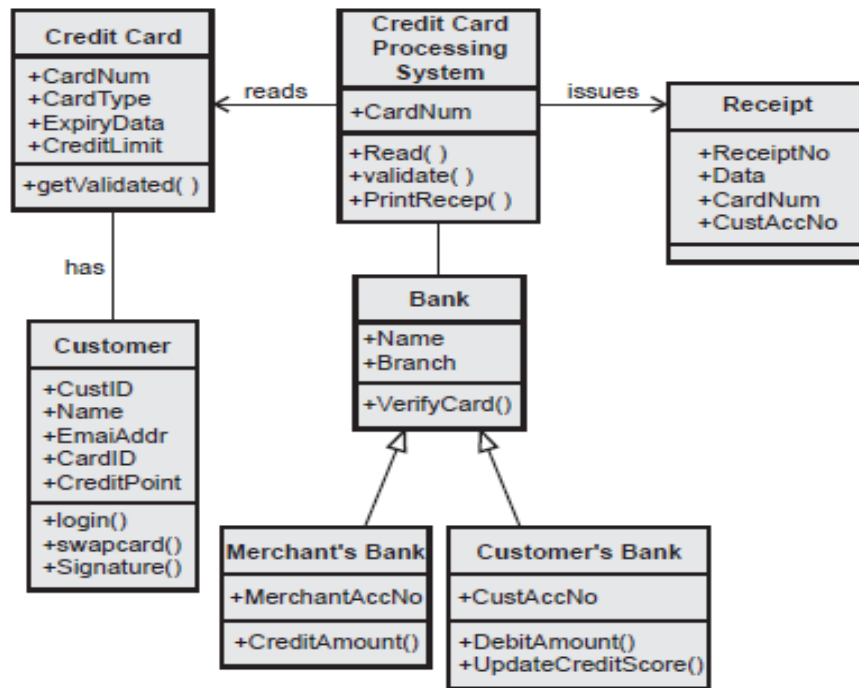
USE CASE DIAGRAM:

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).



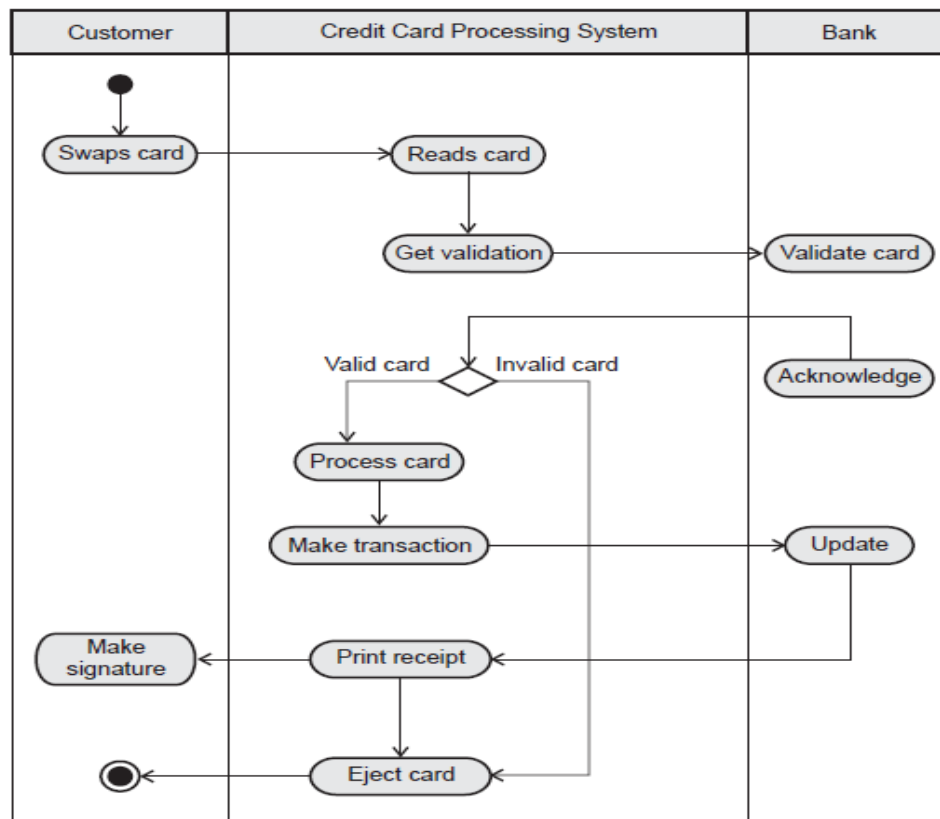
CLASS DIAGRAM:

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.



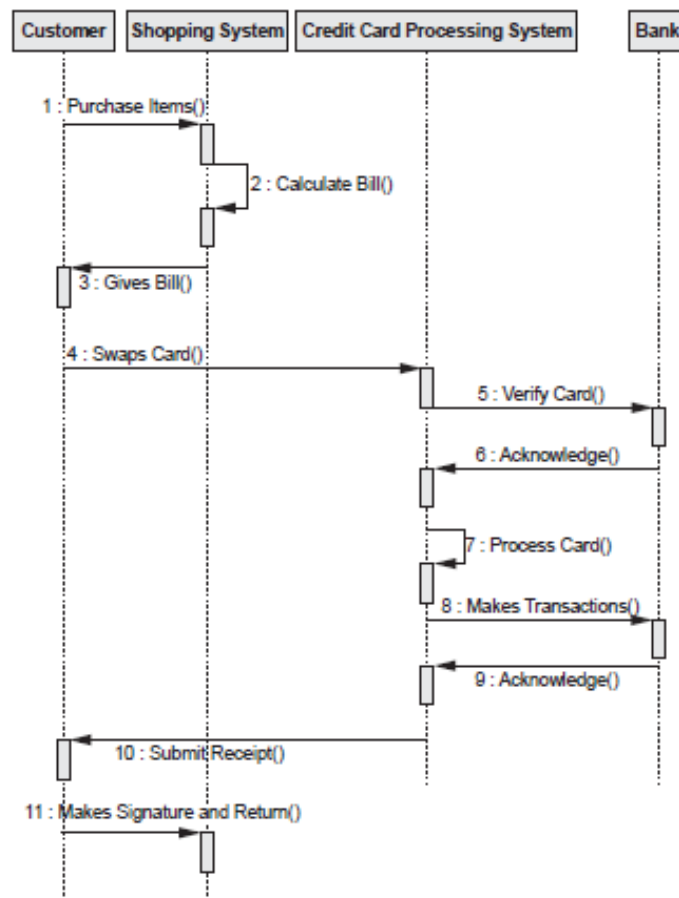
ACTIVITY DIAGRAM

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.



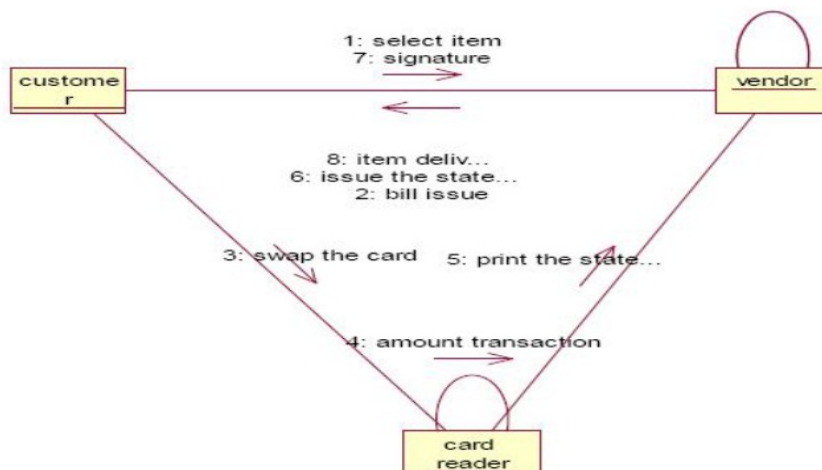
SEQUENCE DIAGRAM

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



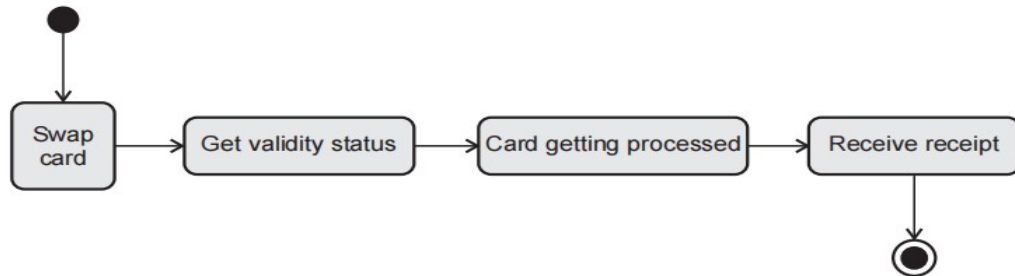
COLLABORATION DIAGRAM

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.



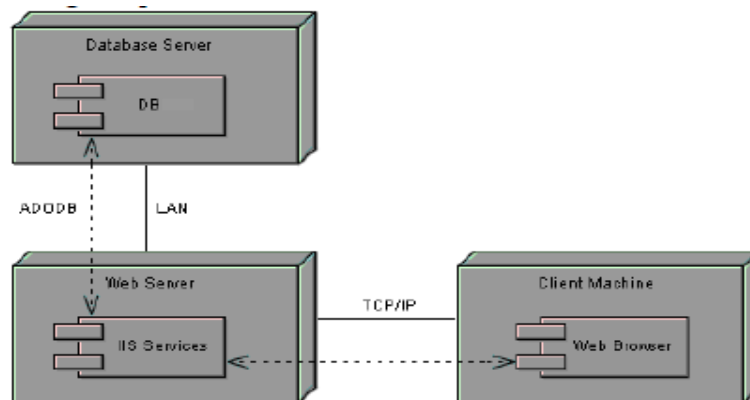
STATE CHART DIAGRAM

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.



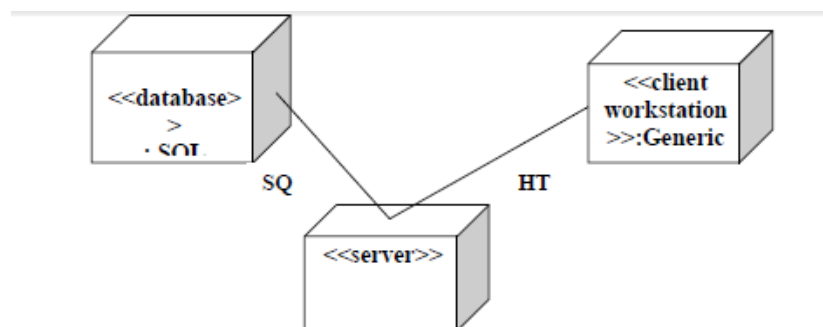
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 10

E-BOOK MANAGEMENT SYSTEM

Date:

AIM:

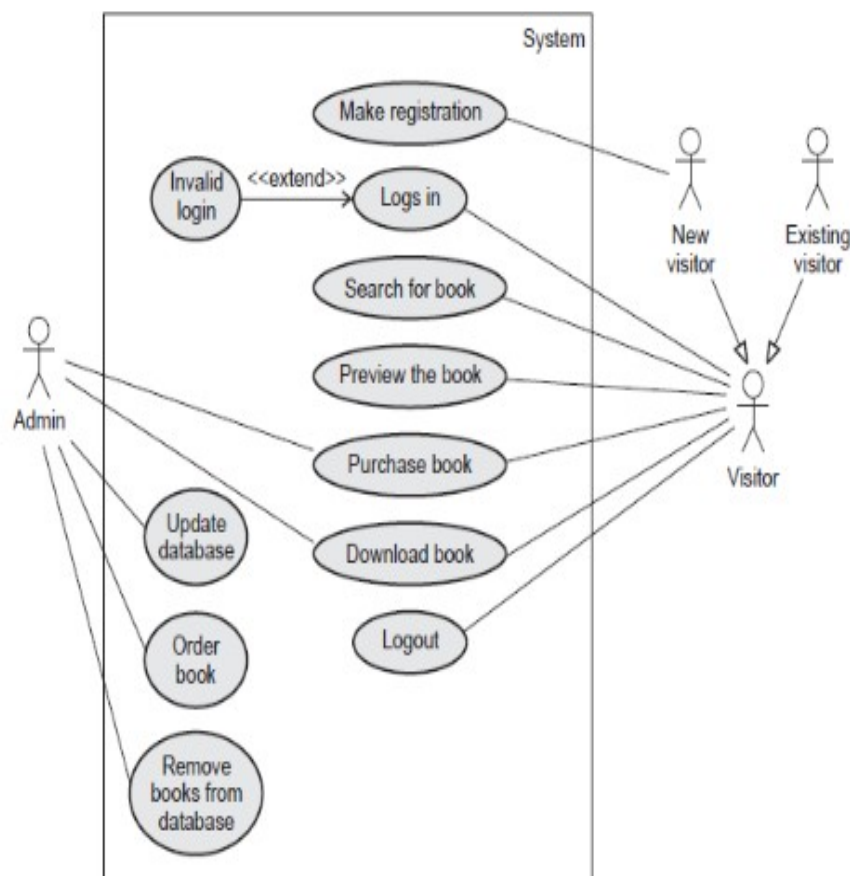
To create a system to perform E- book Management System using Argo UML.

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.

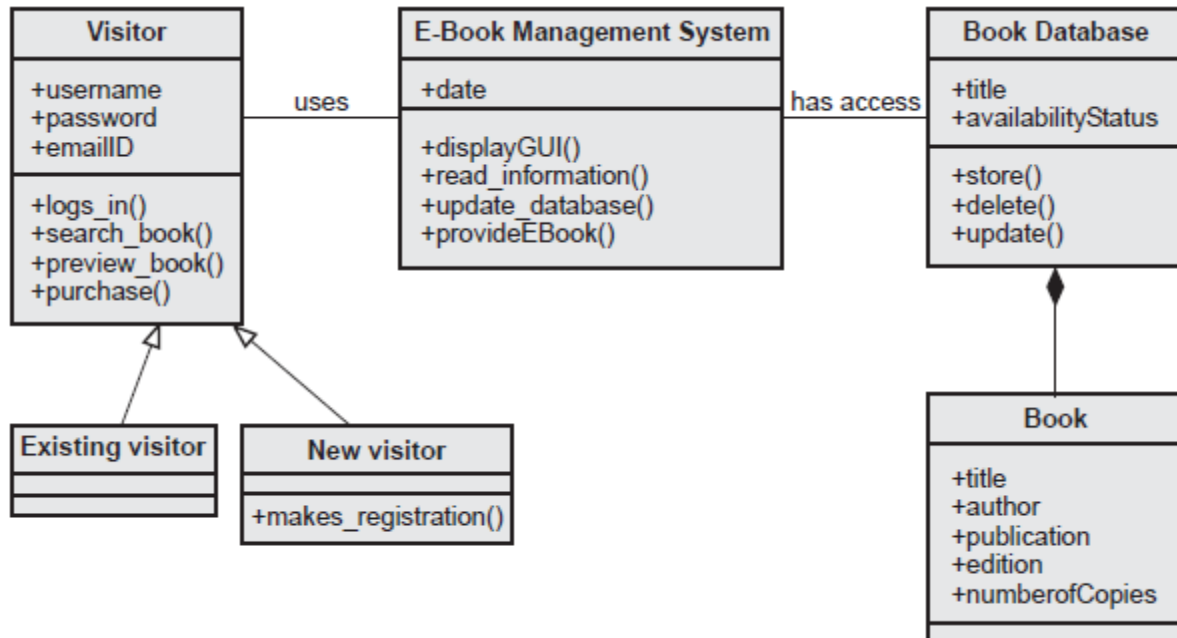
USE-CASE DIAGRAM

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.



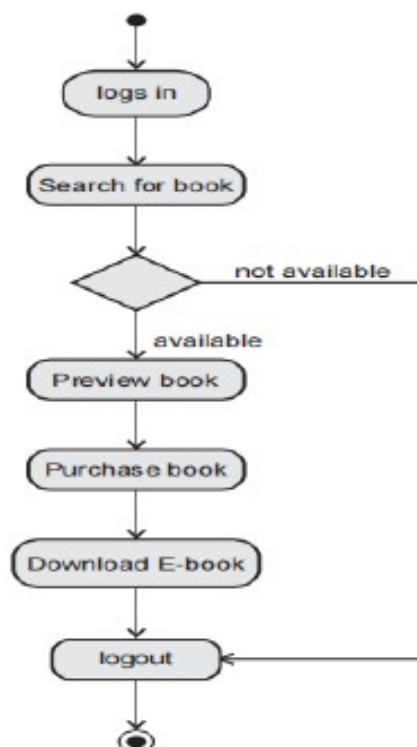
CLASS DIAGRAM

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.



ACTIVITY DIAGRAM

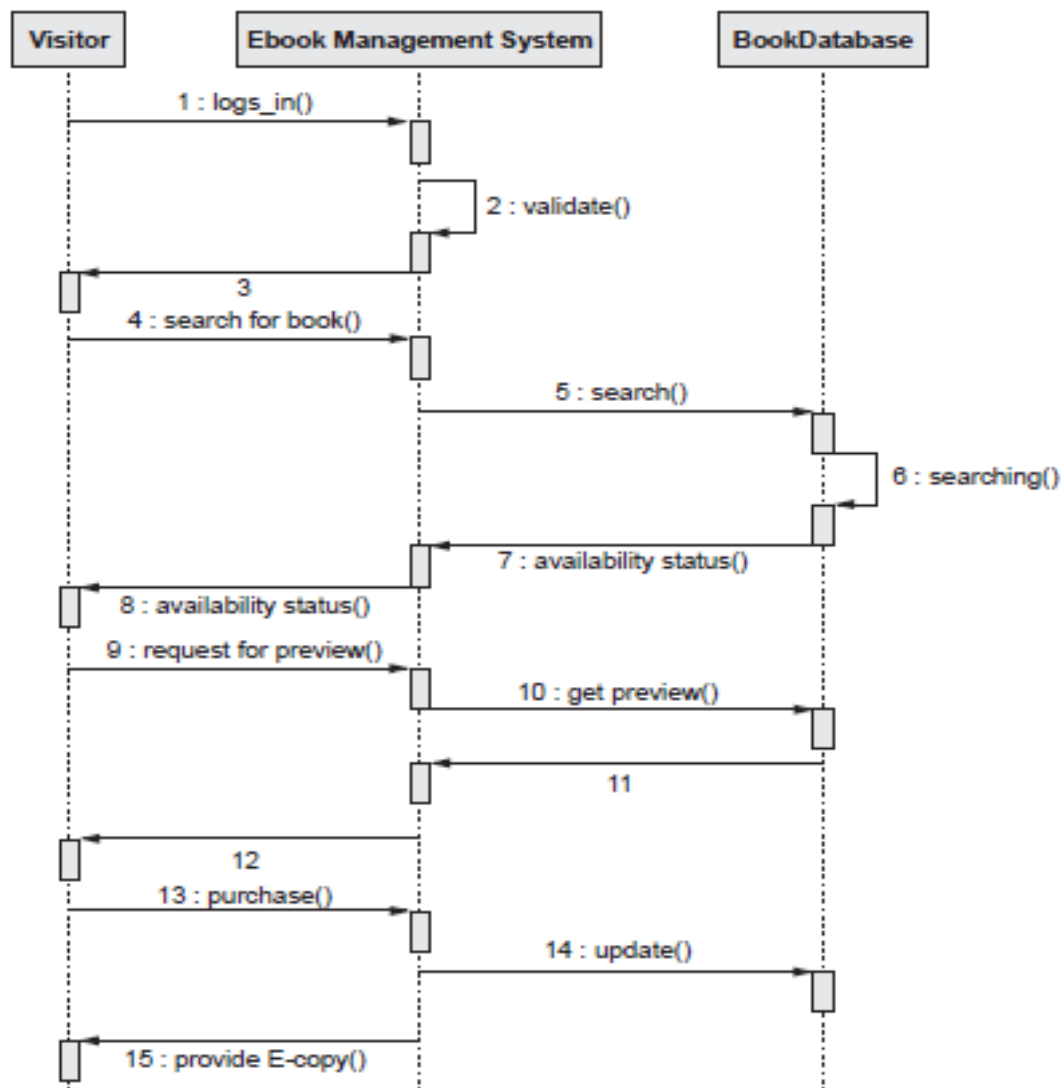
The activity diagram notation is an action, partition, fork join and object node. Most of the notation is self explanatory, two subtle points. Once an action finished, there is an automatic outgoing transaction. The diagram can show both control flow and data flow.



SEQUENCE DIAGRAM:

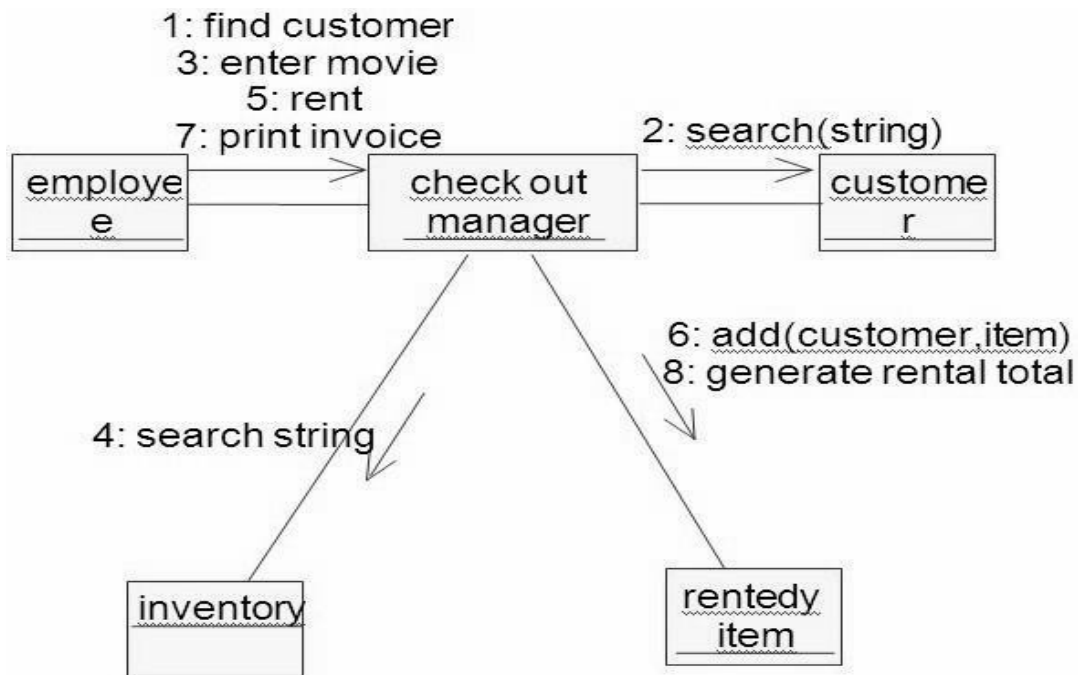
A sequence diagram represents the sequence and interactions of a given USE-CASE or scenario. Sequence diagrams can capture most of the information about the system. Most object to object interactions and operations are considered events and events include signals, inputs, decisions, interrupts, transitions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



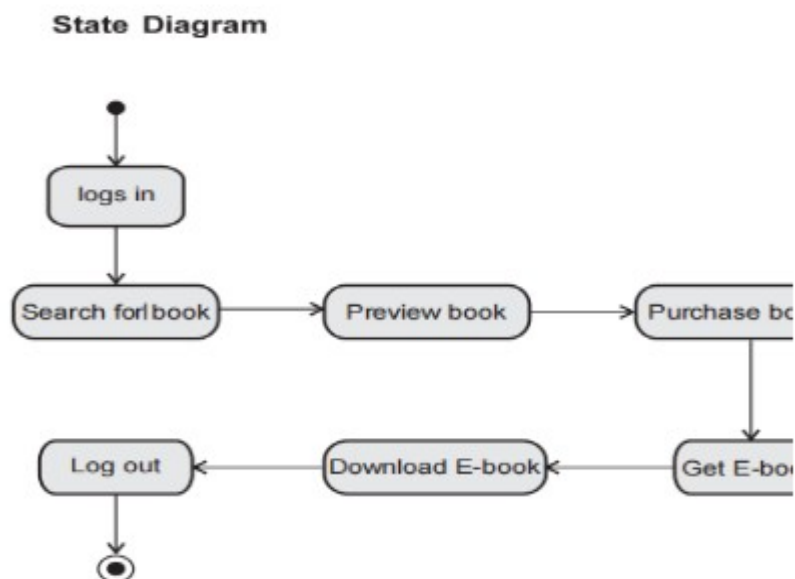
COLLABORATION DIAGRAM

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.



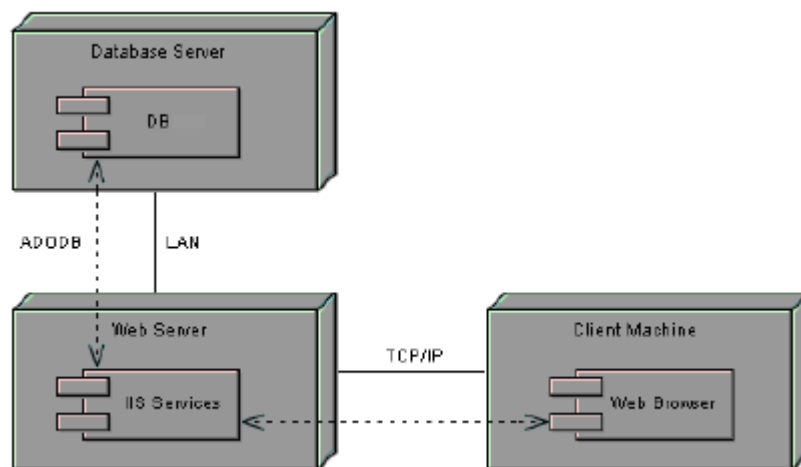
STATE CHART DIAGRAM

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.



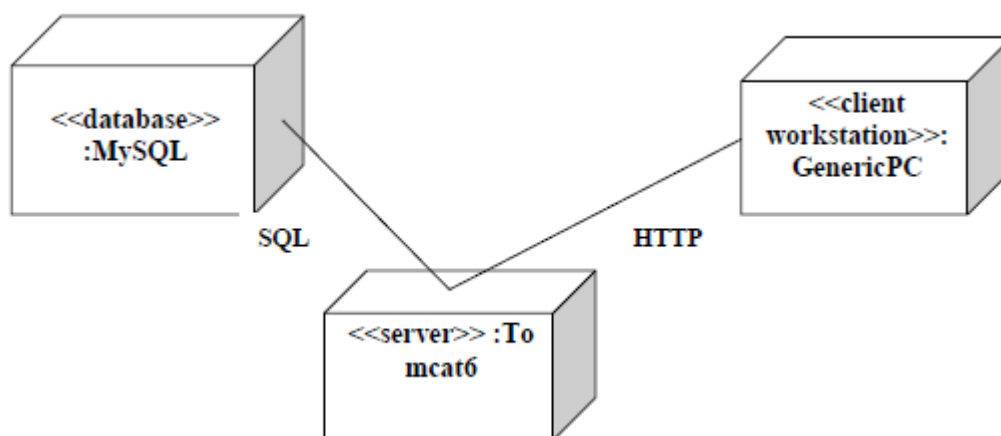
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 11

RECRUITMENT SYSTEM

Date:

AIM:

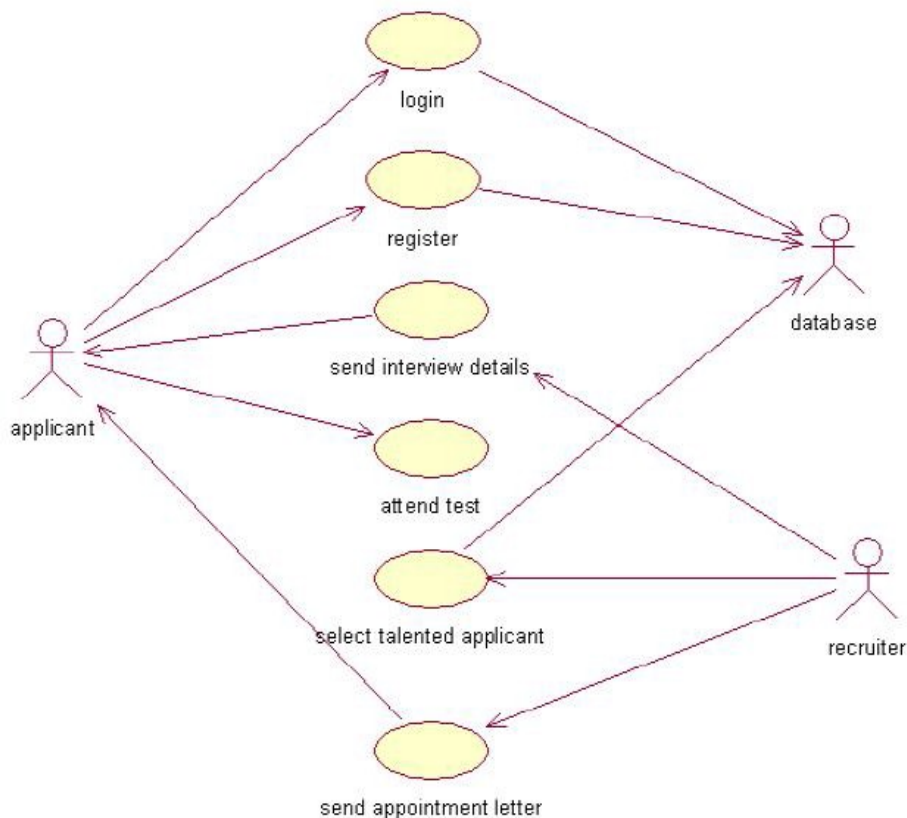
To create an automated system to perform the Recruitment System Process using Argo UML.

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.

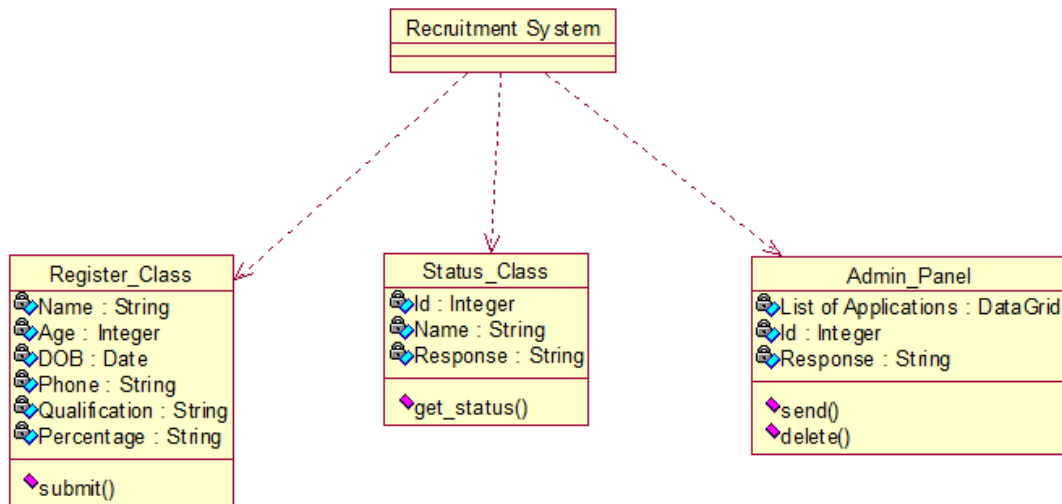
USE CASE DIAGRAM:

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.



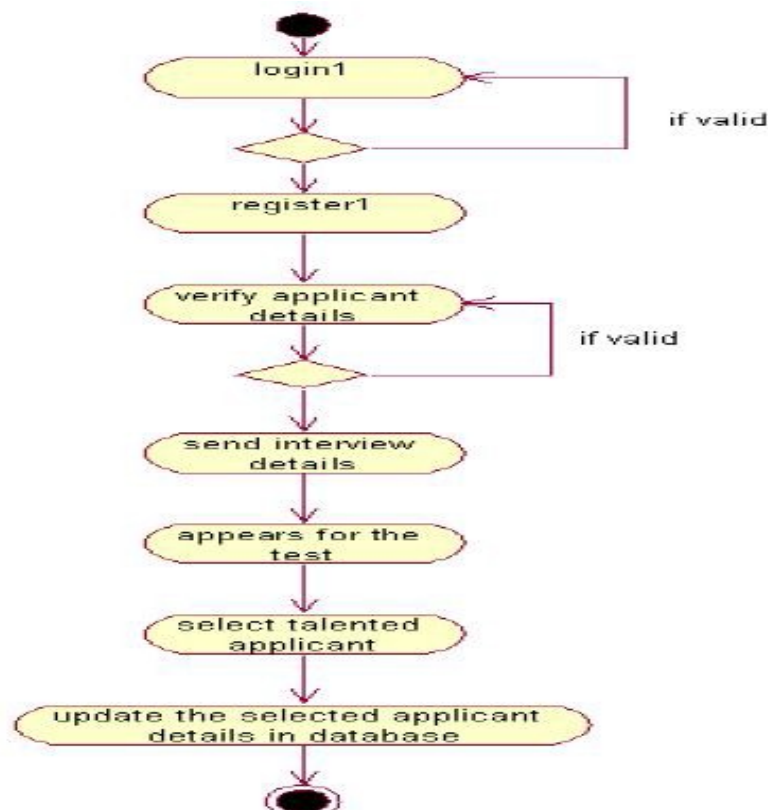
CLASS DIAGRAM:

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.



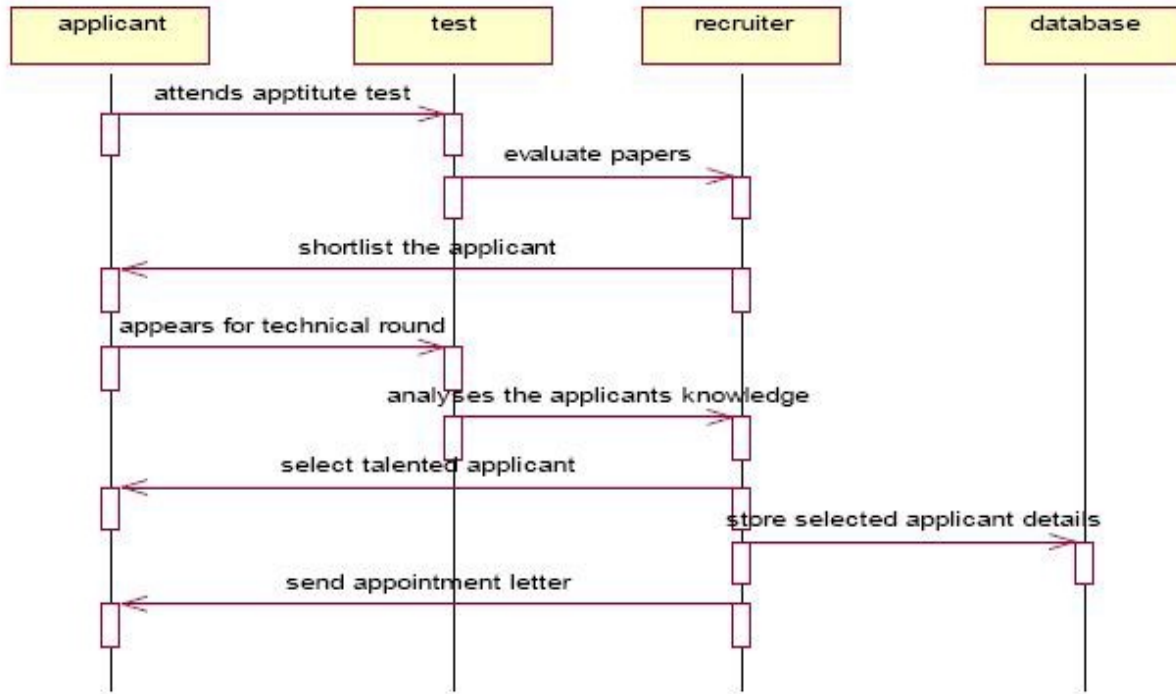
ACTIVITY DIAGRAM:

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.



SEQUENCE DIAGRAM:

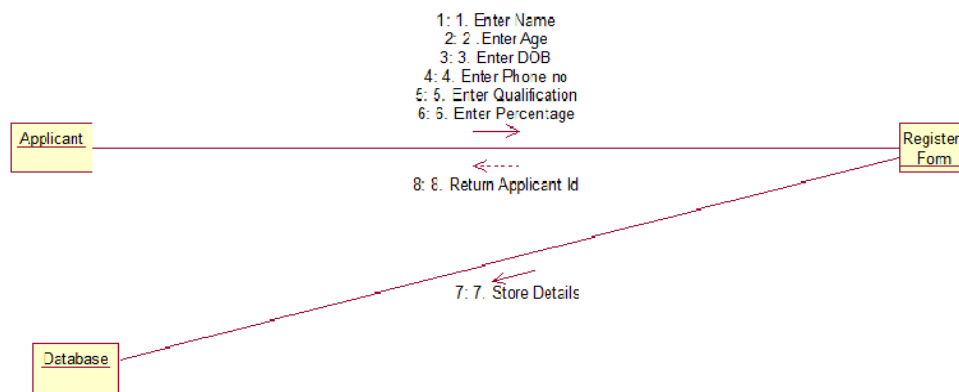
A sequence diagram illustrates a kind of format in which each object interacts via message. It is generalized between two or more specialized diagrams.



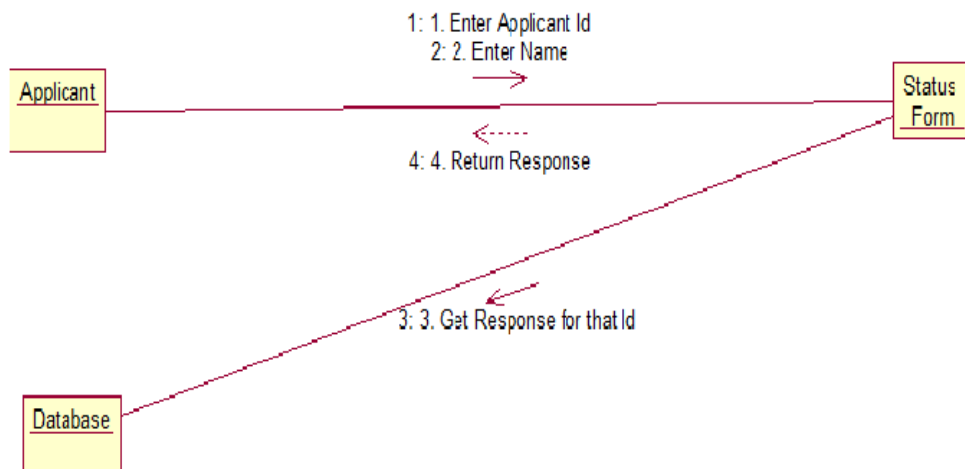
COLLABORATION DIAGRAM:

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

Collaboration Diagram for Register

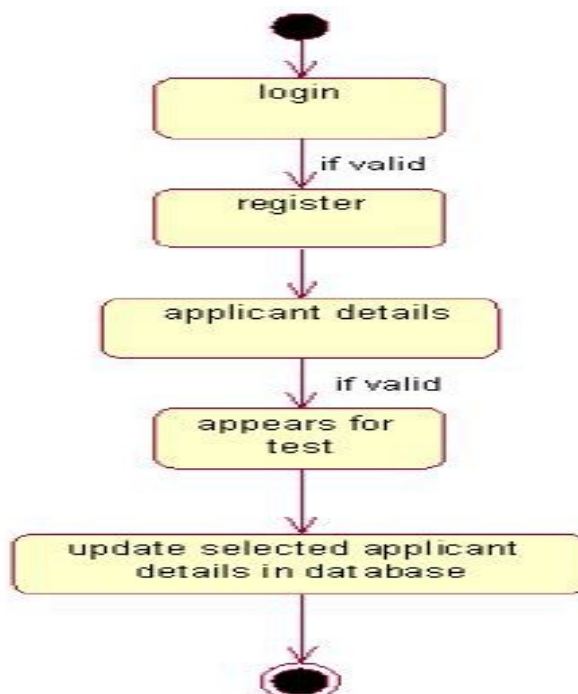


Collaboration Diagram for Status



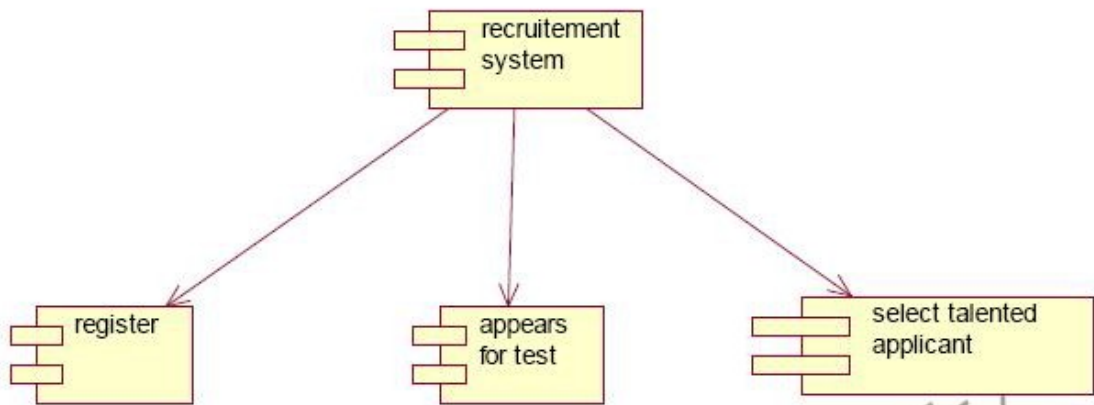
STATE CHART DIAGRAM

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.



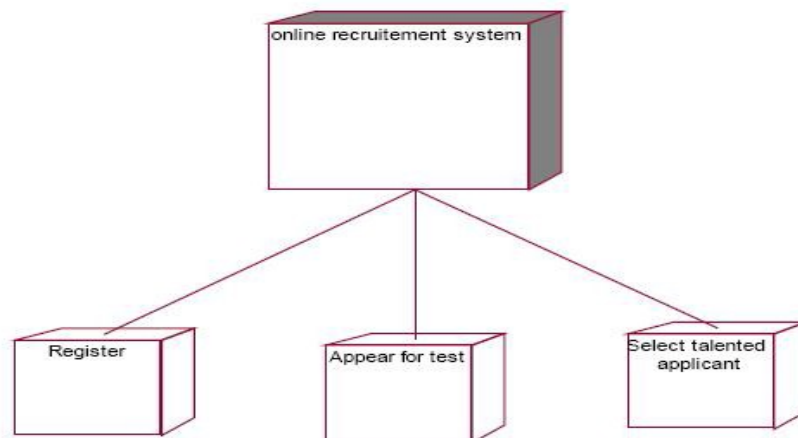
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT:

Ex. No: 12

FOREIGN TRADING SYSTEM

Date:

AIM

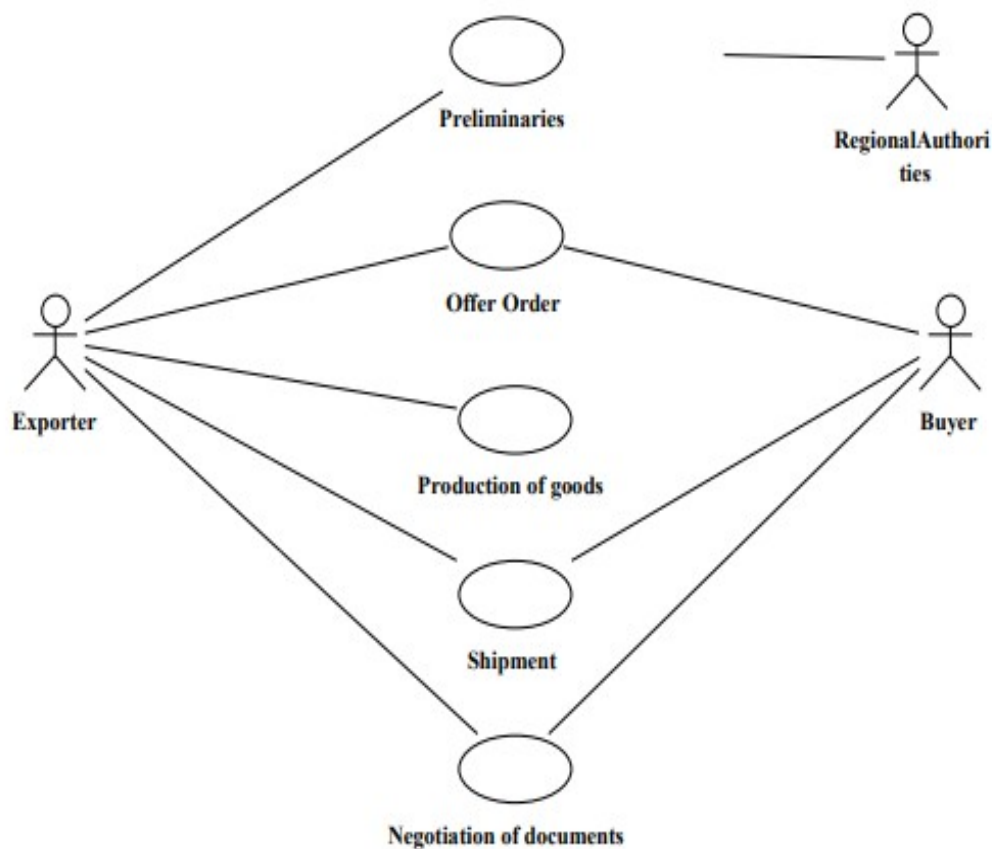
To design a project Foreign Trading System using Argo UML.

PROBLEM STATEMENT

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

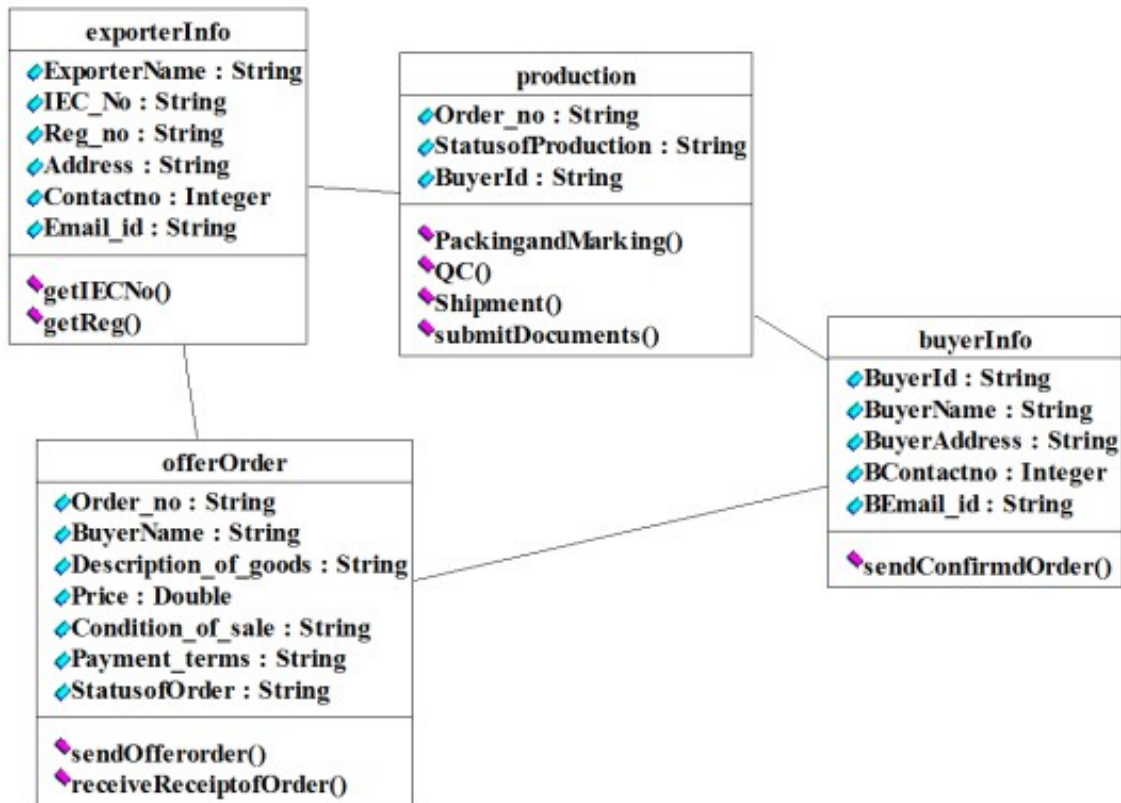
USE CASE DIAGRAM:

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



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.

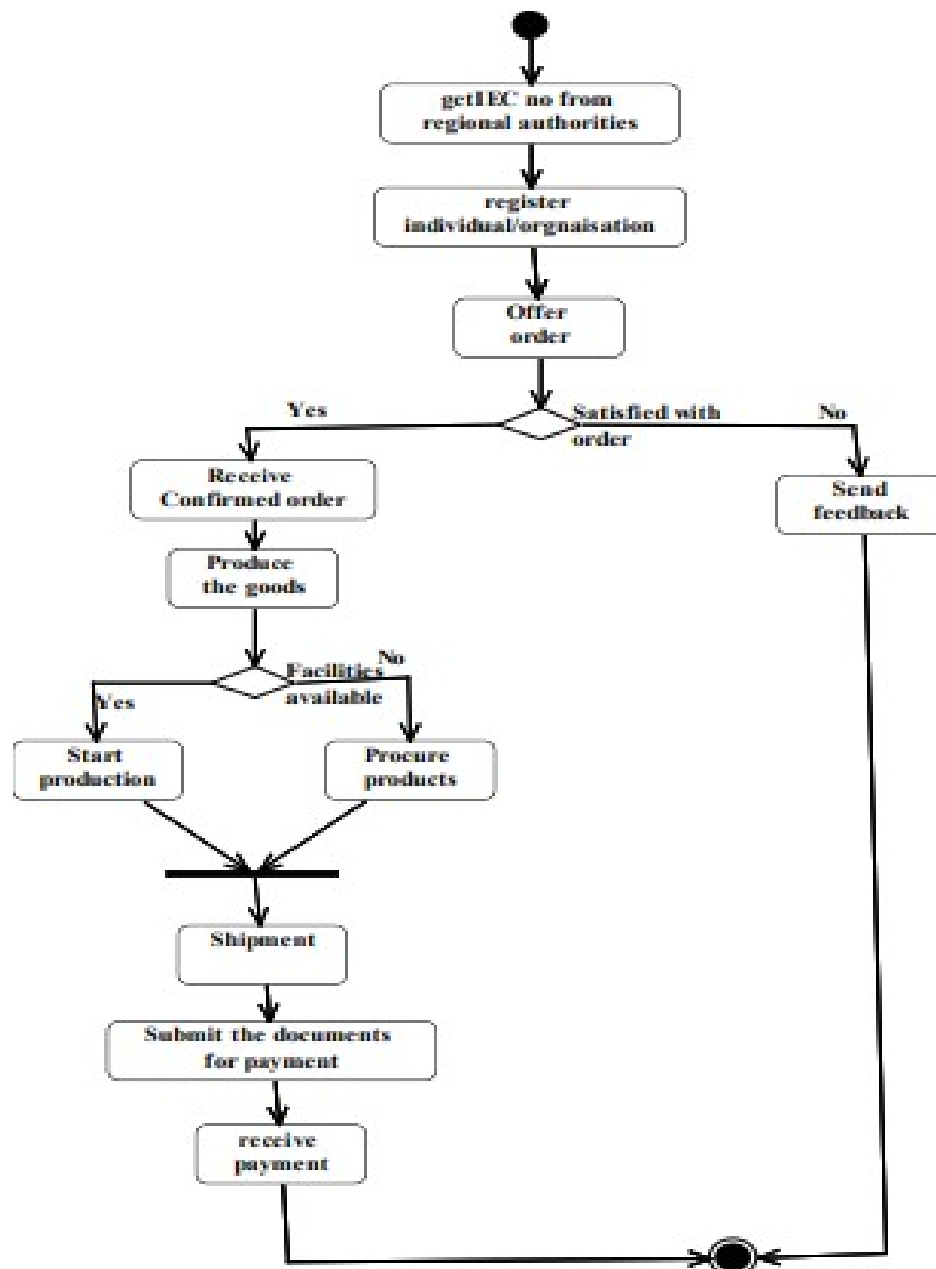


ACTIVITY DIAGRAM

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

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

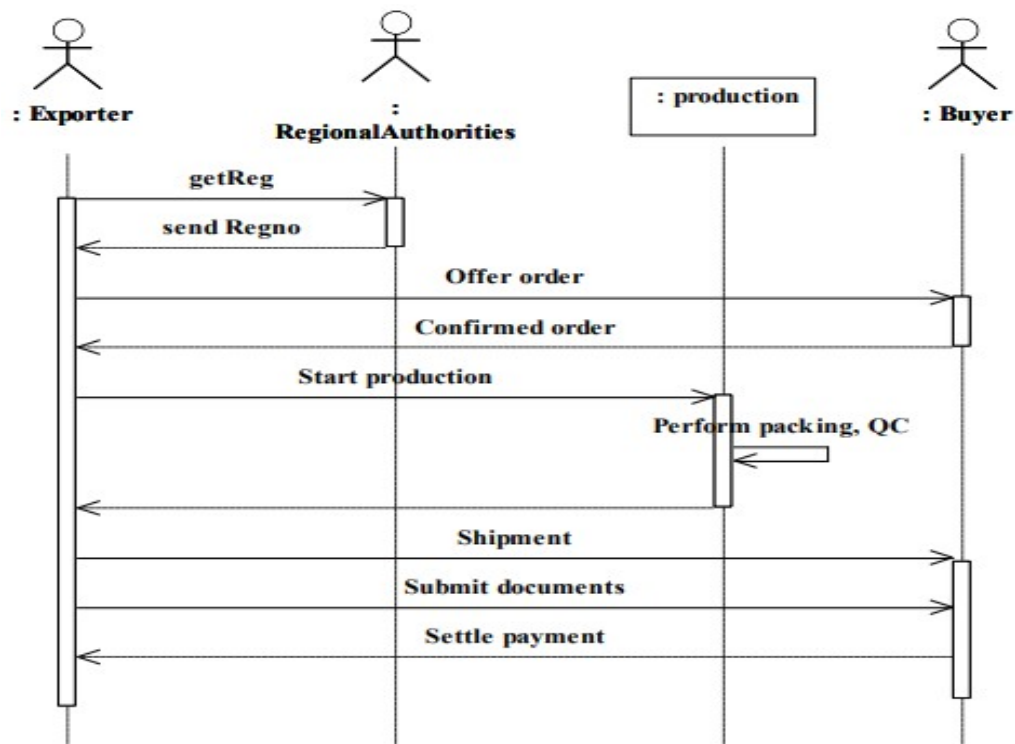
- The first action represents the trader logs in to the system.
- The second action is the place where the trader places the order.
- The decision state is the state where the trader decides to place the order.
- If the trader places the order, fill the form for the required commodities.
- The next activity is that the administrator provides the bill for those commodities.
- The trader pays for the bill and logout from the system.



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

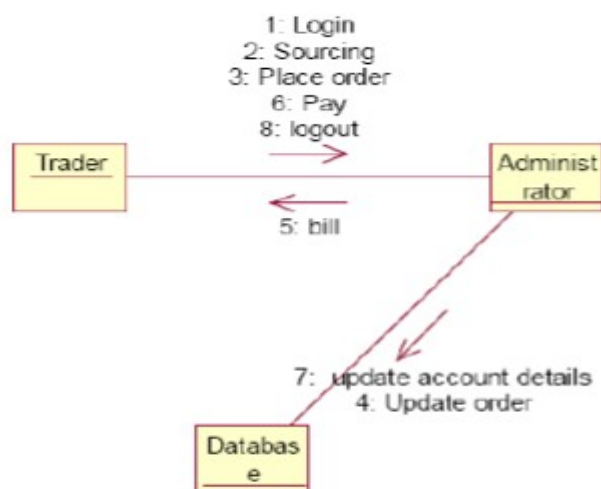
1. Vertical dimension-represents time.
2. Horizontal dimension-represent different object



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.

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.

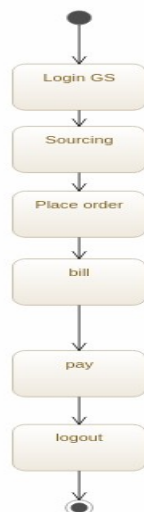


STATE CHART DIAGRAM

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.

The state diagram represents the following states.

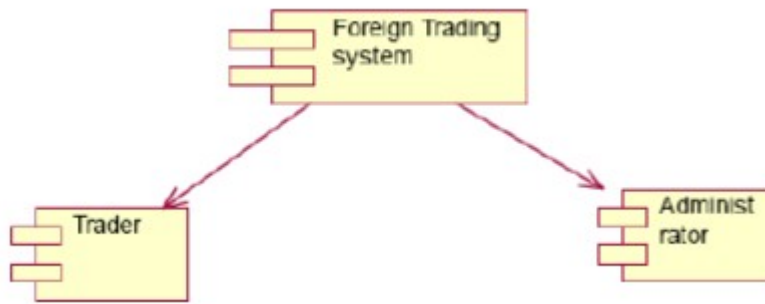
- The trader logs in the register in the first state and performs sourcing in the second state.
- The trader places the order in the third state.
- The trader receives the bill in the fourth state and pay the required amount in fifth state.
- The trader logs out from the system in the sixth state



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.

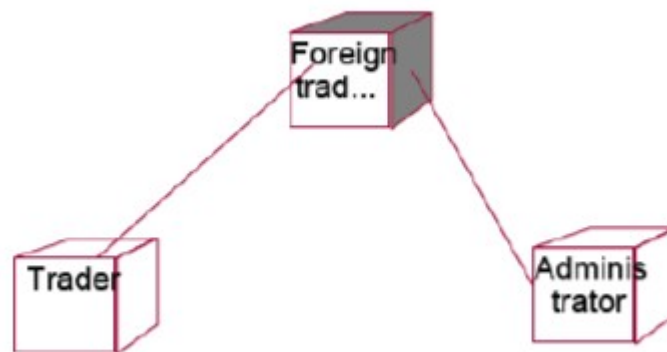
The main component in the component diagram is foreign trading system. The trader who come to do the trading process and administrator who manages all the other processes is the sub components.



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.

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.



RESULT

Ex. No: 13

CONFERENCE MANAGEMENT SYSTEM

Date:

AIM

To develop a project on Conference management system using Argo UML.

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 acknowledgment 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 acknowledgment to the candidate.
- Based on the selection, the best candidate is selected.
- Finally the candidate registers all details.

UML DIAGRAMS

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

USE CASE DIAGRAM

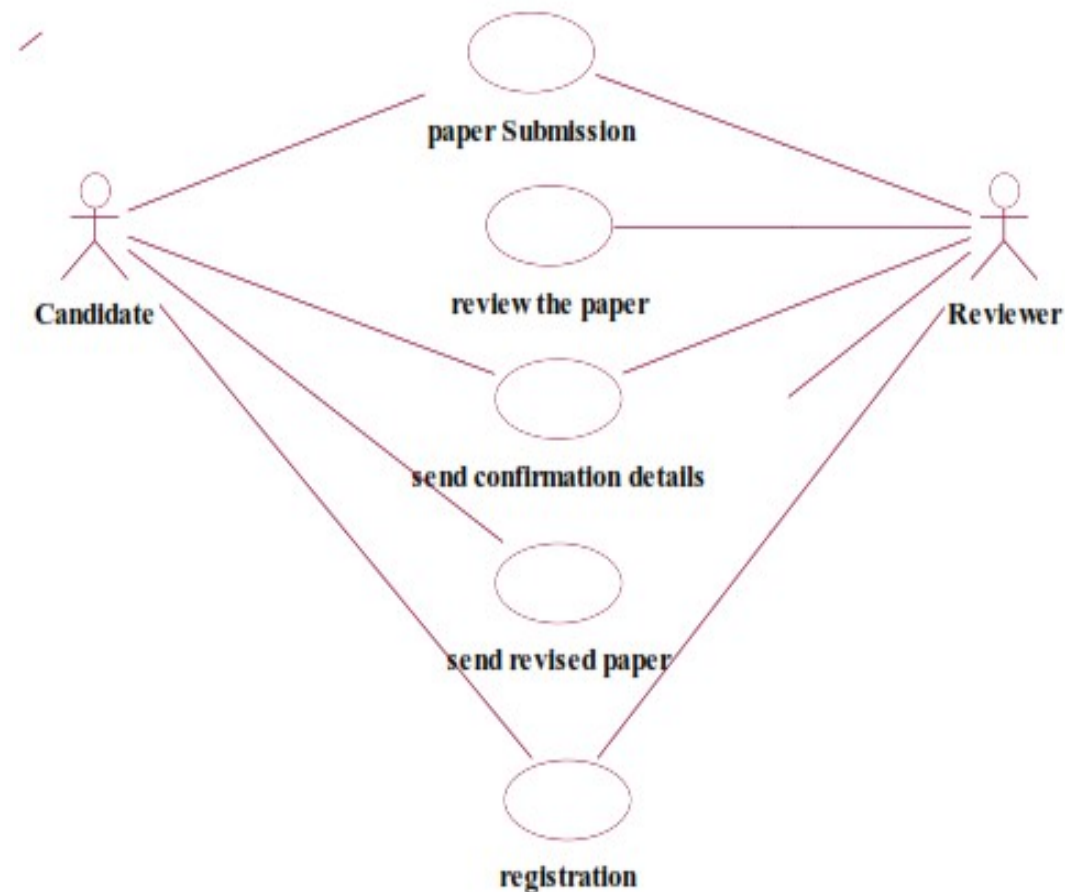
A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using ellipse. Actor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

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 acknowledgment 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.

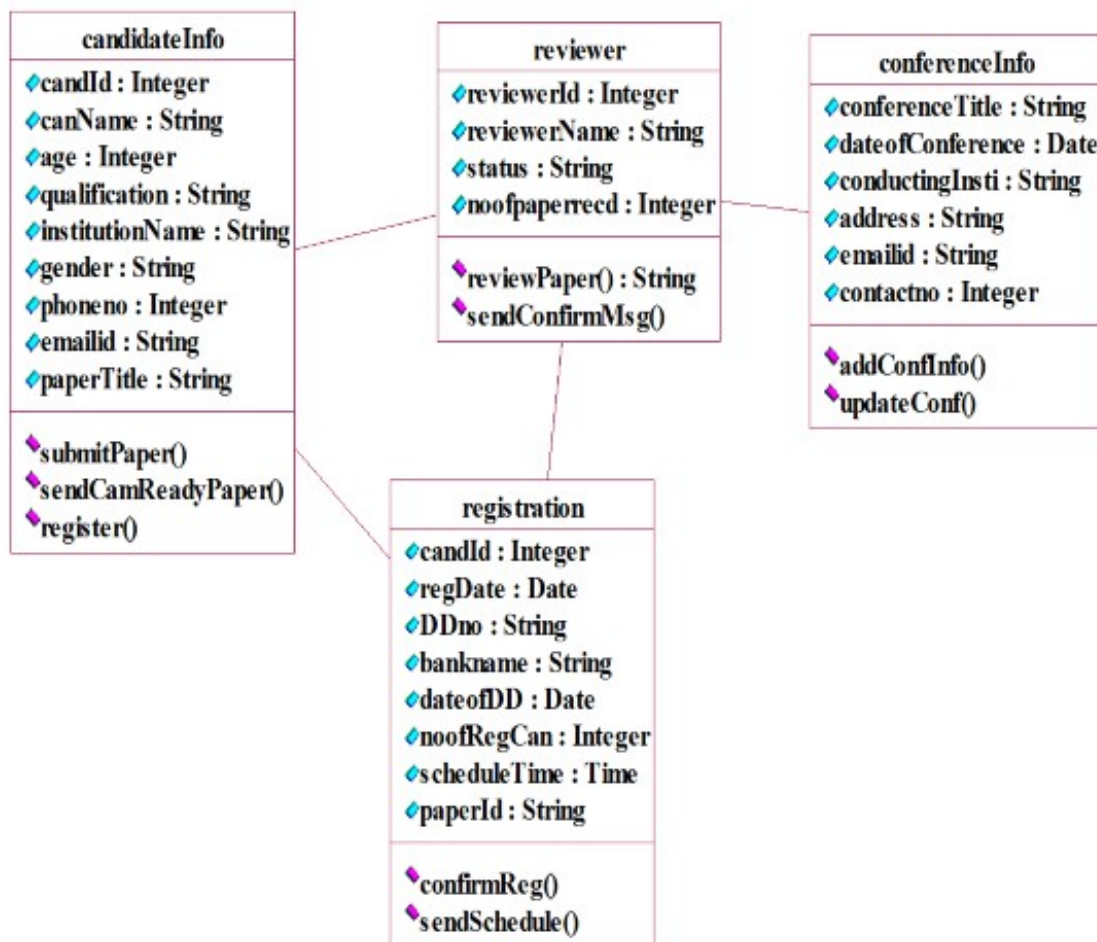


CLASS DIAGRAM

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.

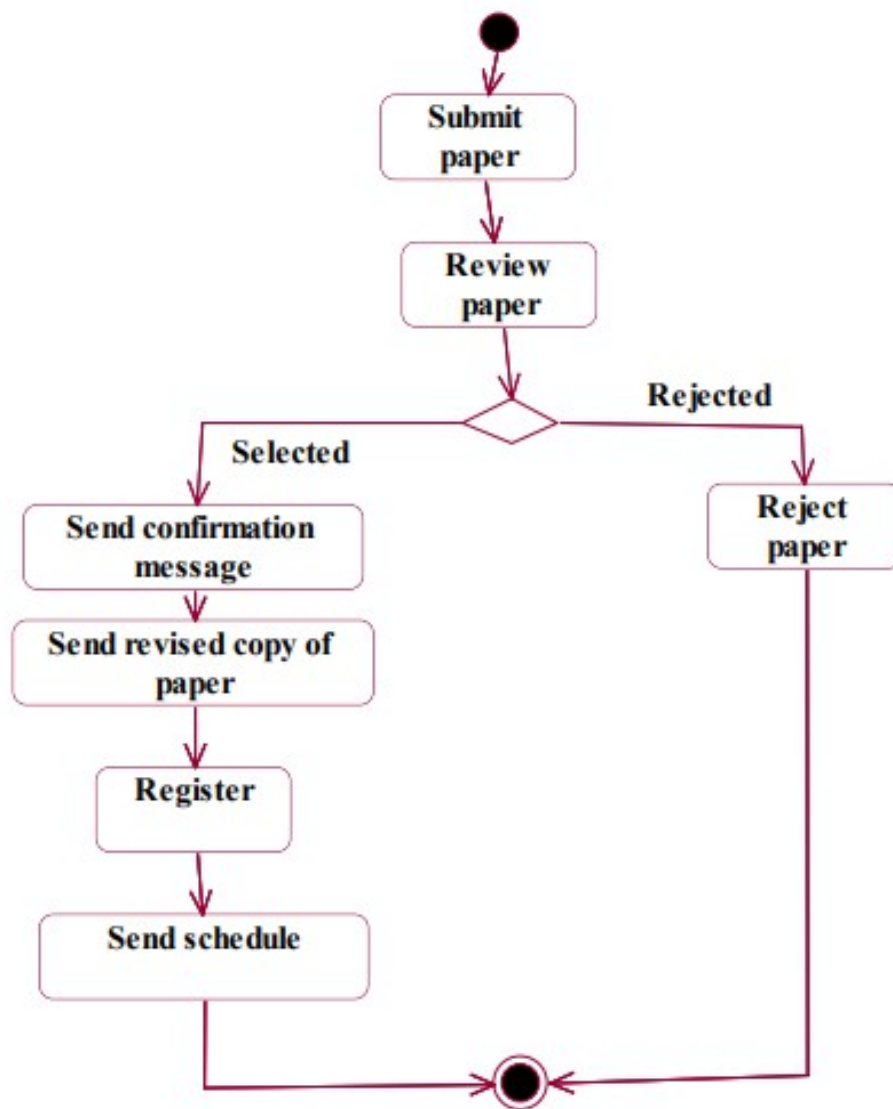
This class diagram has three classes candidate, reviewer and database.

- **Candidate** – Its attributes are name ,college name , 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 .



ACTIVITY DIAGRAM

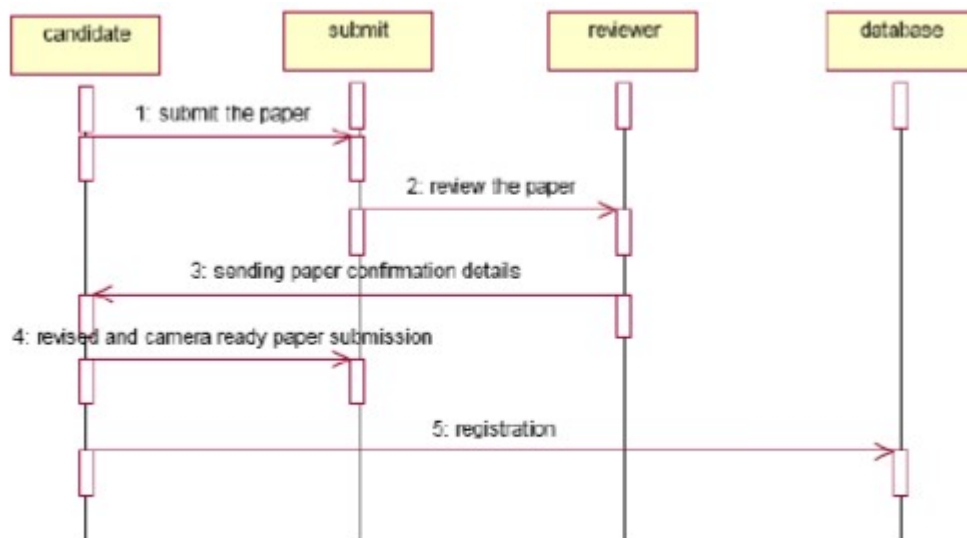
Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.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



SEQUENCE DIAGRAM

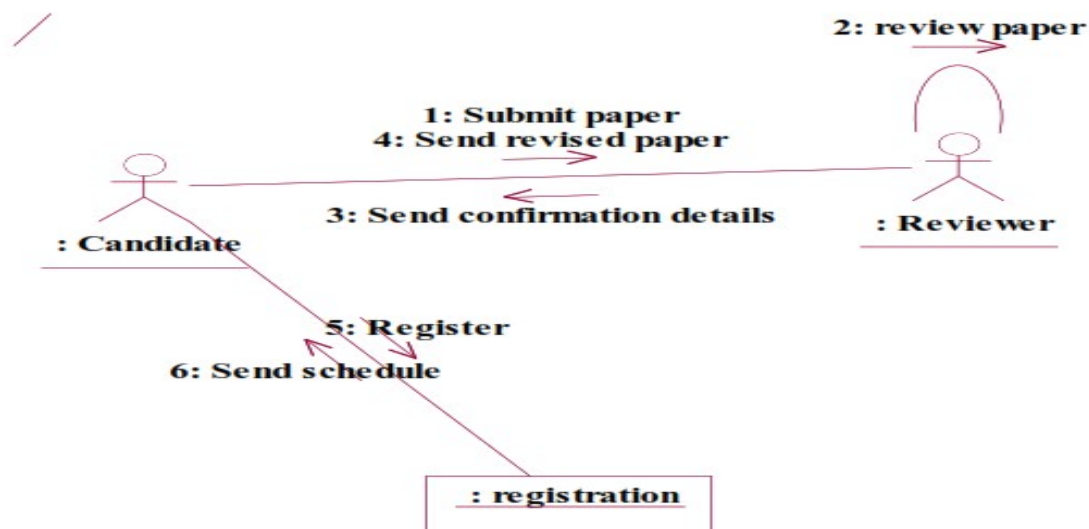
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

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



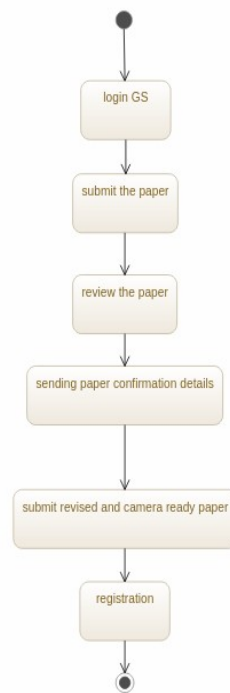
COLLABORATION DIAGRAM

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

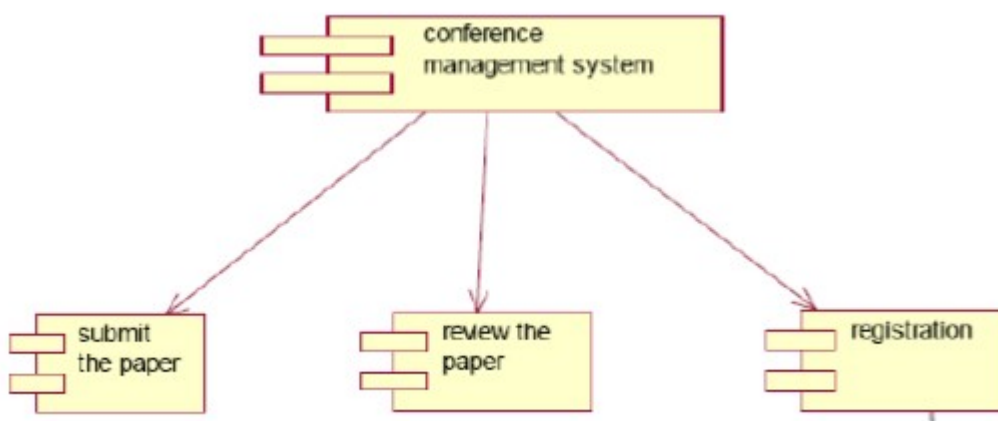
The purpose of state chart diagram is to understand the algorithm involved in performing a method. It is also called as state diagram. A state is represented as a round box, which may contain one or more compartments. An initial state is represented as small dot. A final state is represented as circle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



COMPONENT DIAGRAM

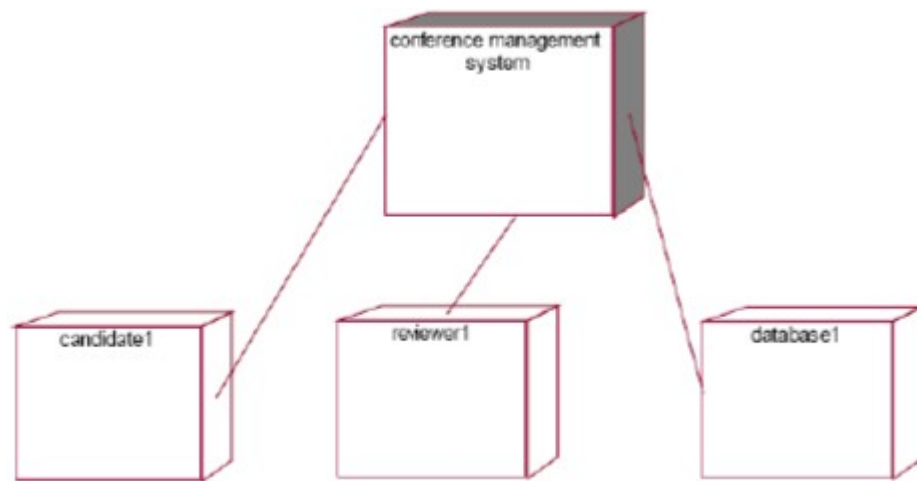
The component diagram's main purpose is to show the structural relationships between the components of a system. It is represented by boxed figure. Dependencies are represented by communication association.

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.



RESULT

Ex. No: 14

BPO MANAGEMENT SYSTEM

Date:

AIM:

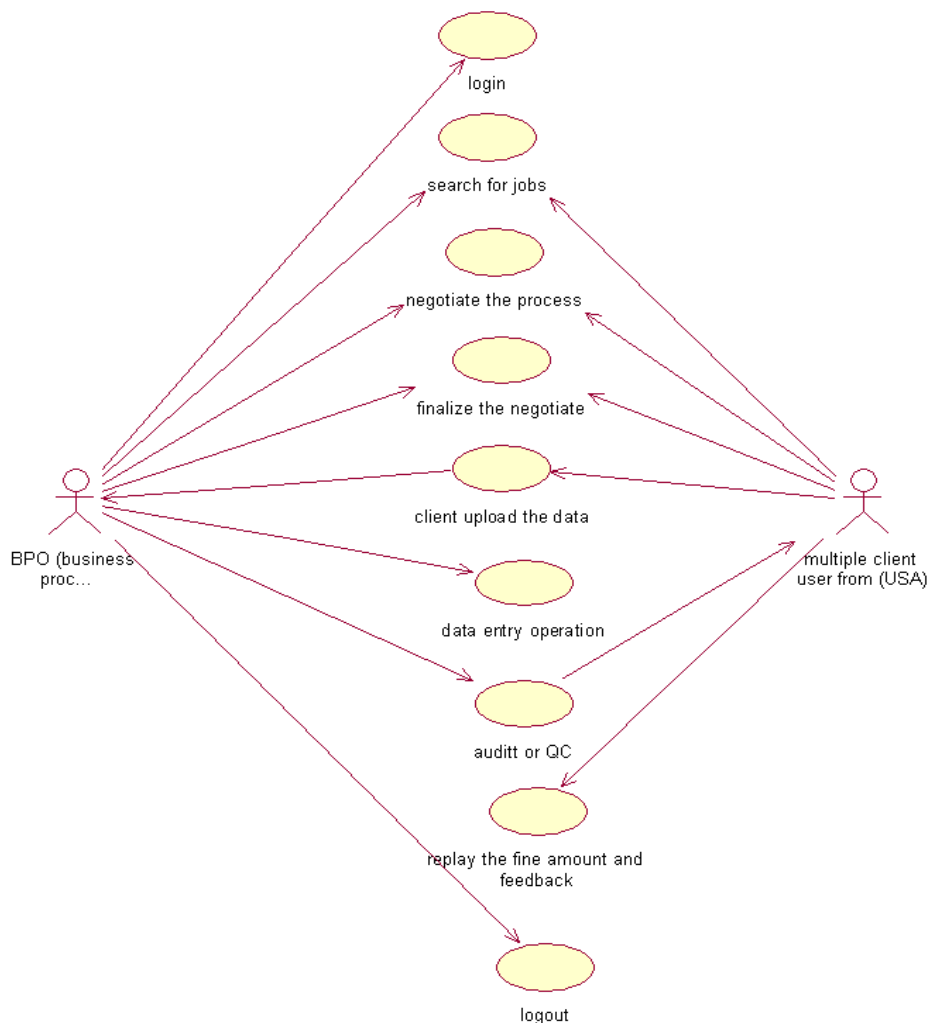
To implement a software for BPO management system using Argo UML.

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.

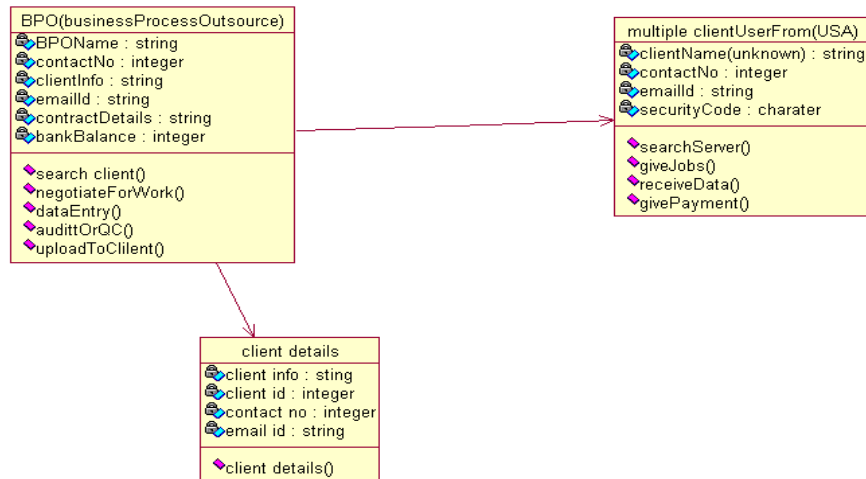
USE CASE DIAGRAM

The BPO management system use cases are:



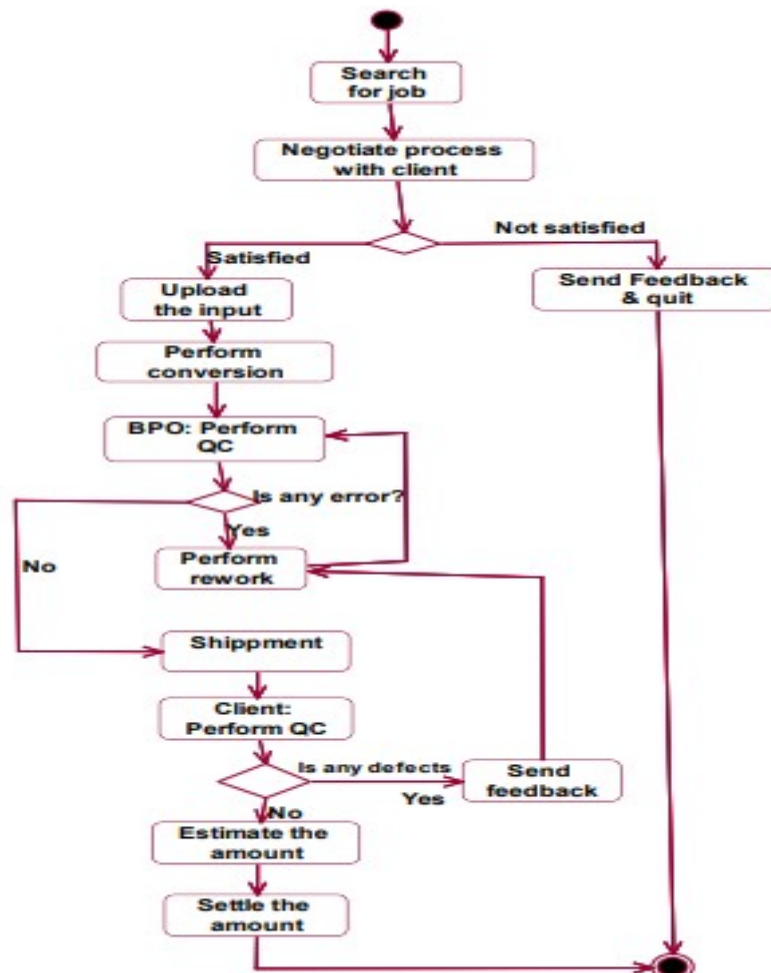
CLASS DIAGRAM

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.



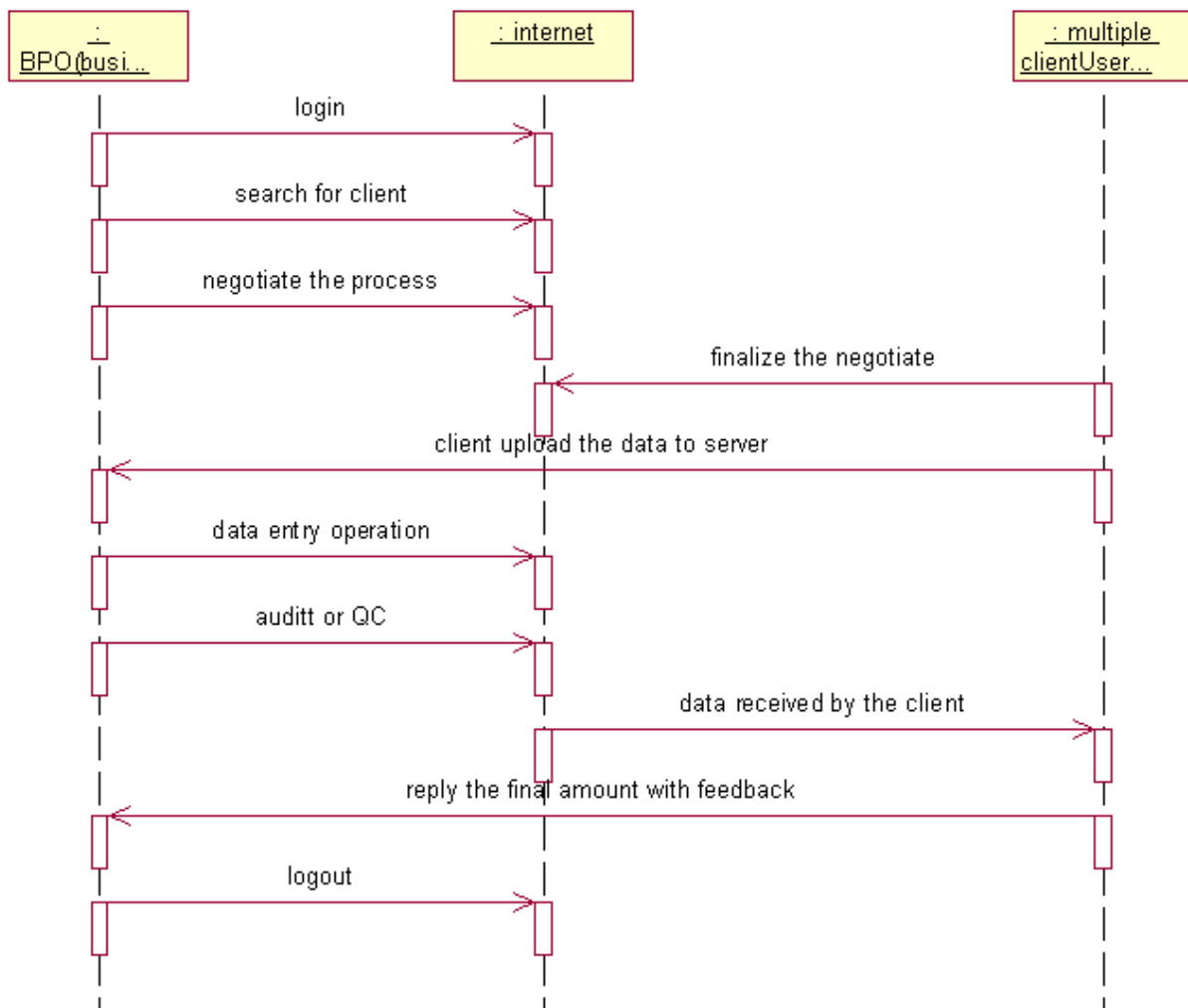
ACTIVITY DIAGRAM

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.



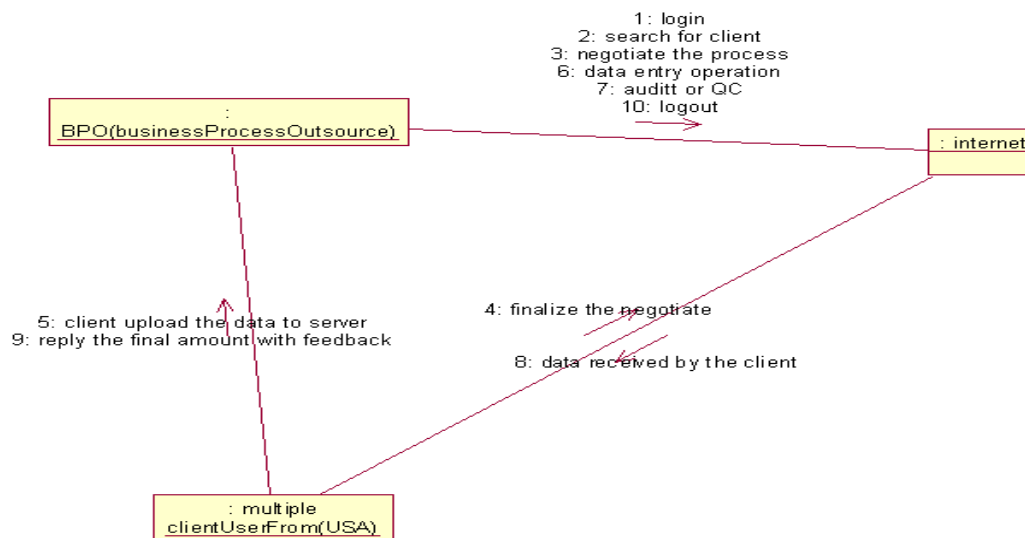
SEQUENCE DIAGRAM

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



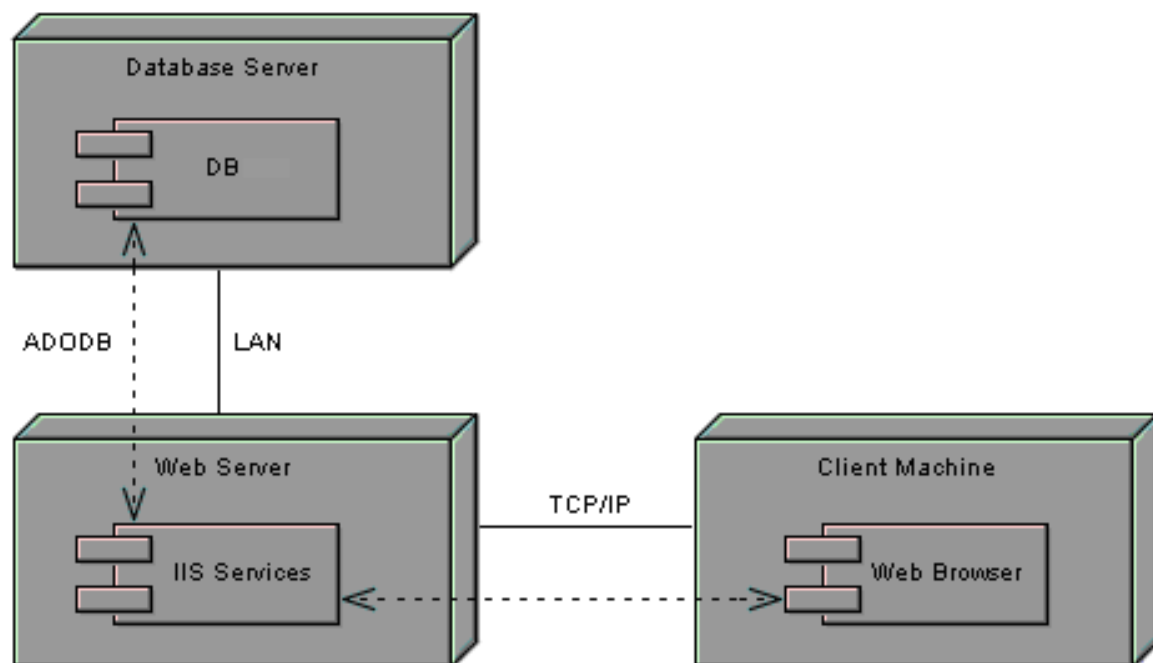
COLLABORATION DIAGRAM

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.



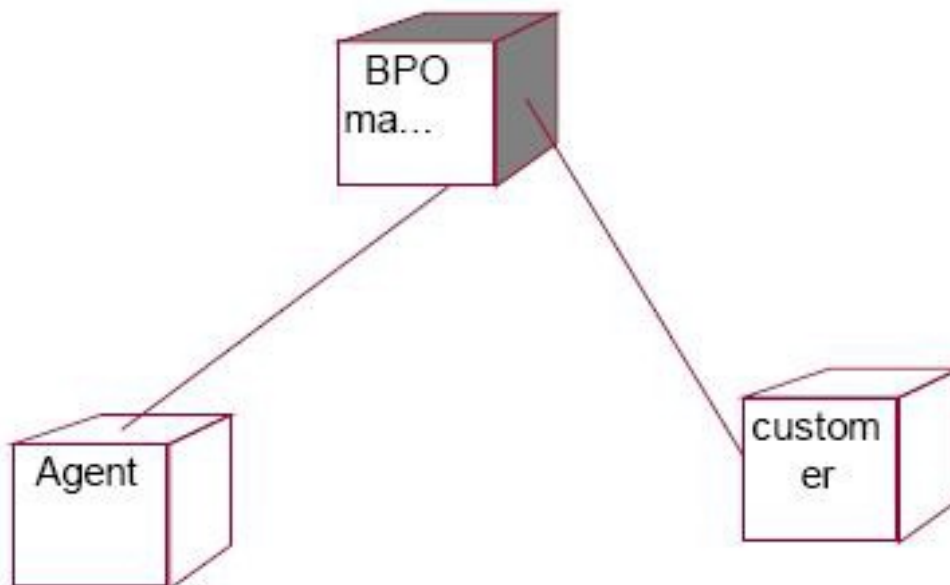
COMPONENT DIAGRAM

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



DEPLOYMENT DIAGRAM

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



RESULT :

Ex. No: 15

LIBRARY MANAGEMENT SYSTEM

Date:

AIM

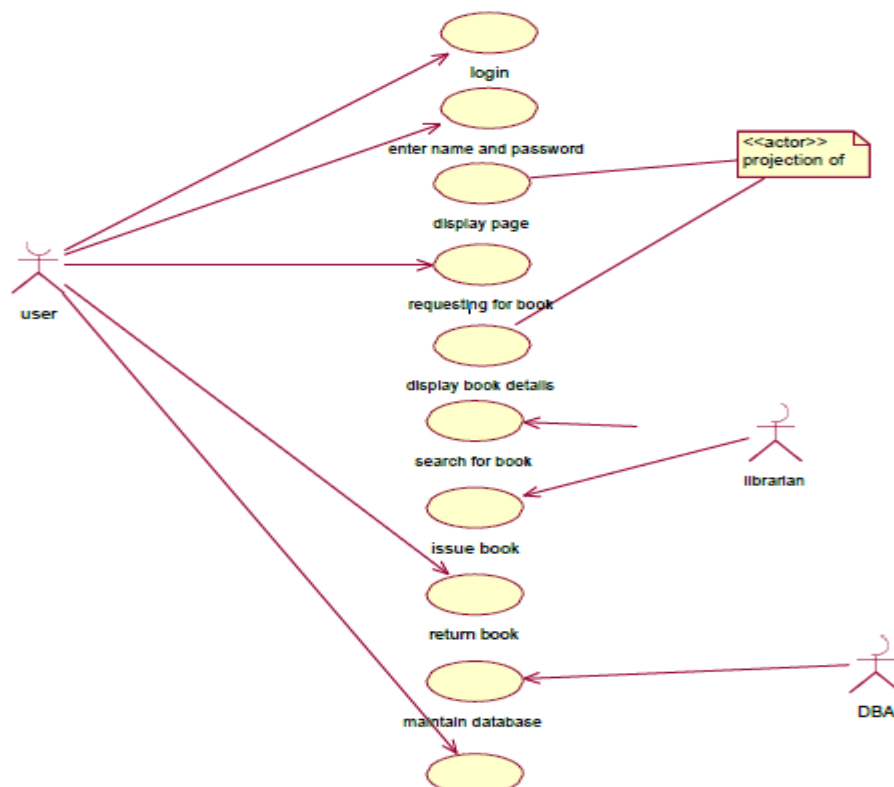
To design an object oriented model for Library Management System using Argo UML.

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.

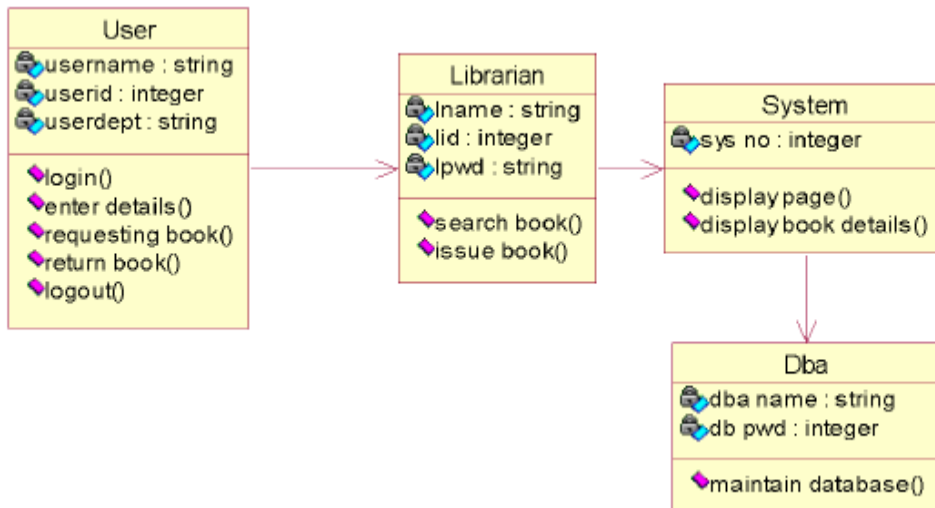
USE CASE DIAGRAM

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.



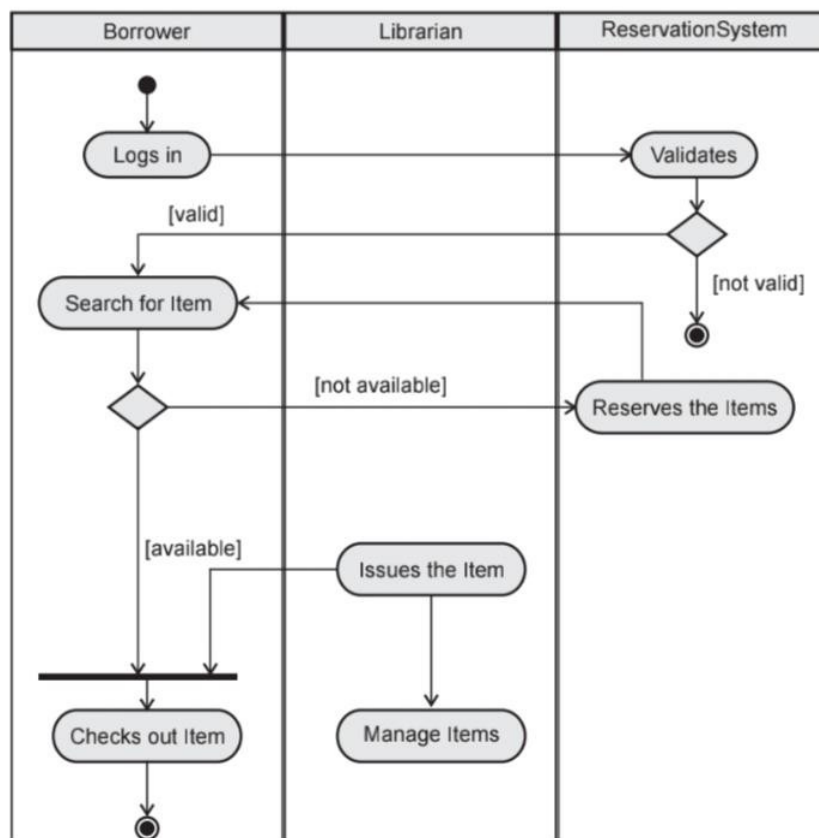
CLASS DIAGRAM

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.



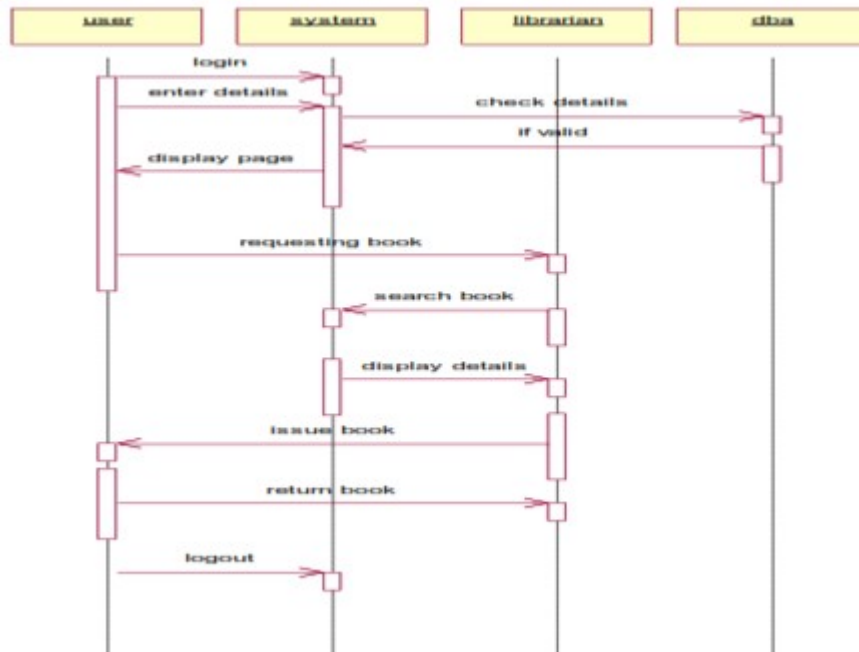
ACTIVITY DIAGRAM

Activity diagram are graphical representation of workflows of step wise 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.



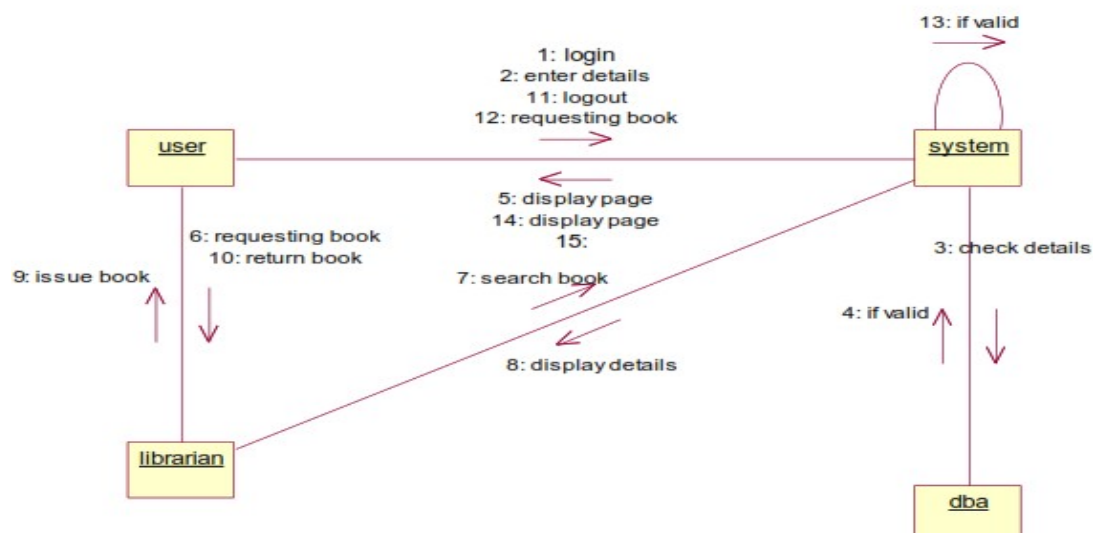
SEQUENCE DIAGRAM

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



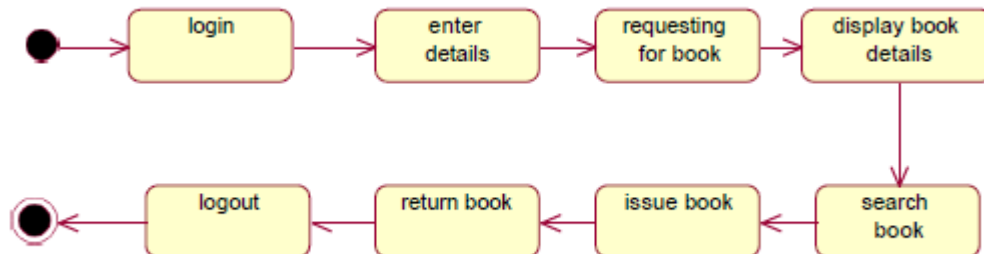
COLLABORATION DIAGRAM

Like sequence diagram collaboration diagrams are also called as interaction diagram. Collaboration diagram convey the same information 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.



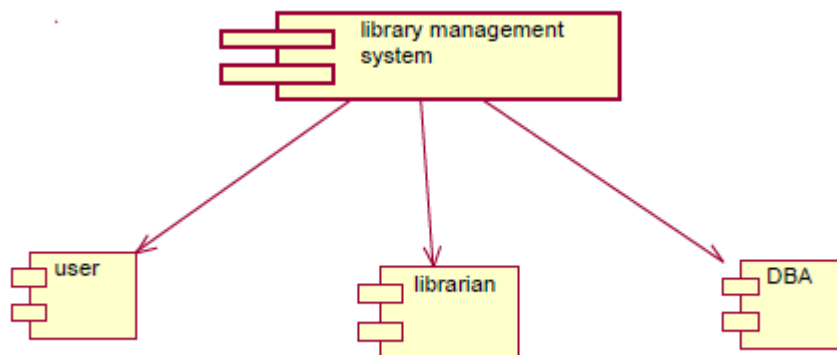
STATE CHART DIAGRAM

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.



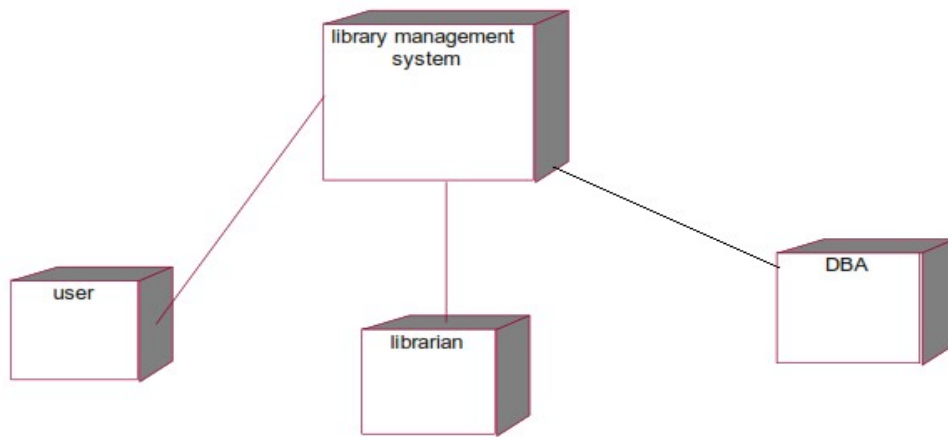
COMPONENT DIAGRAM

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.



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.



RESULT

Ex. No: 16

STUDENT INFORMATION SYSTEM

Date:

AIM

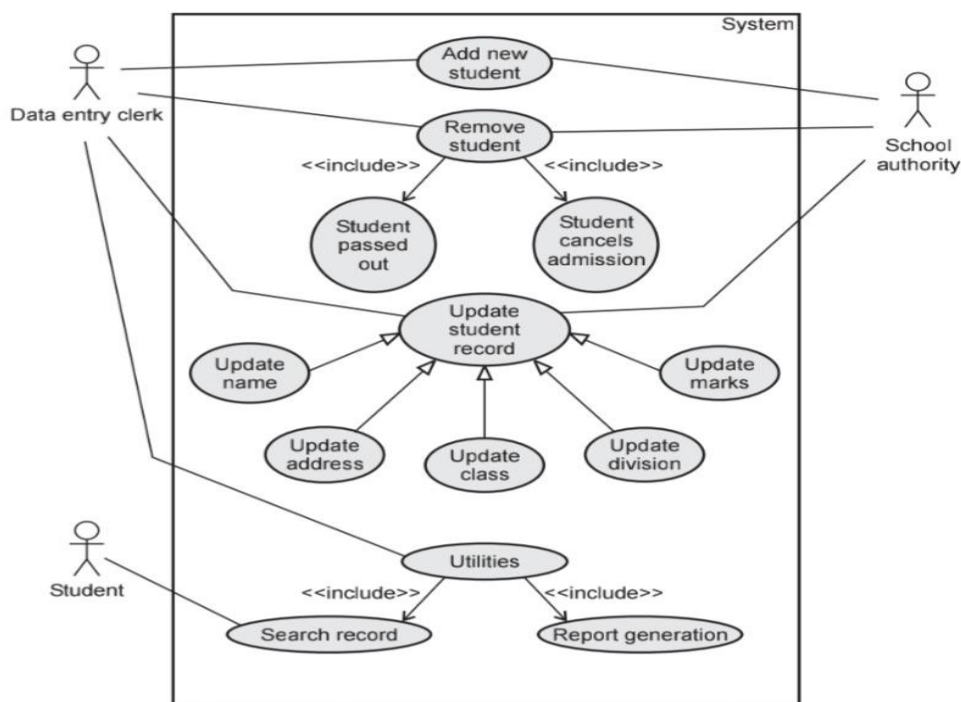
To design an object oriented model for Student information system using Argo UML.

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.

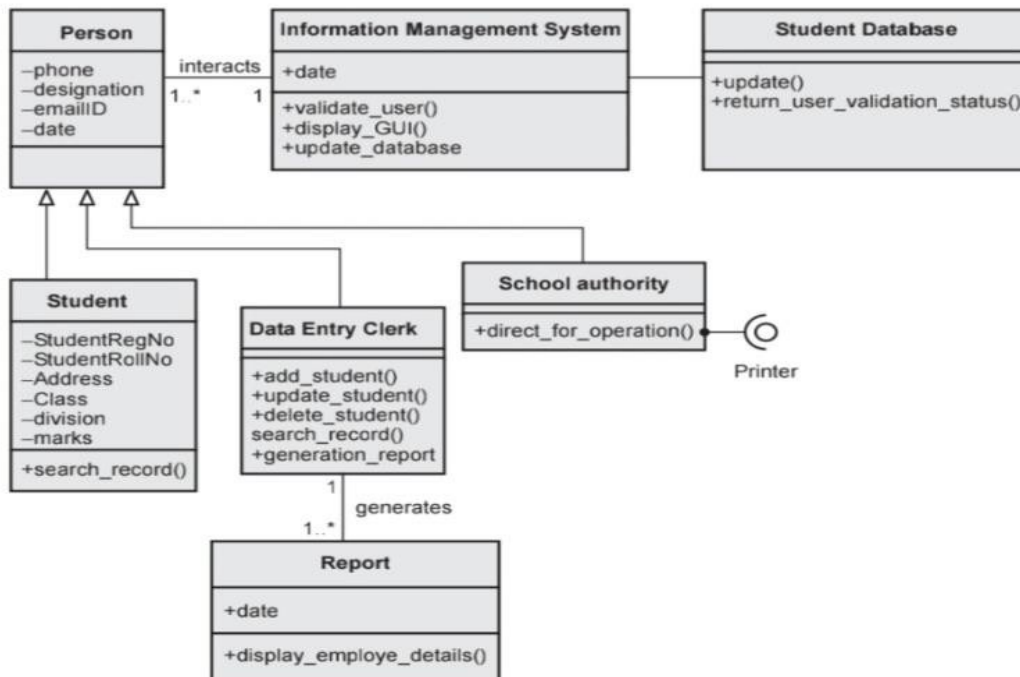
USE CASE DIAGRAM

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.



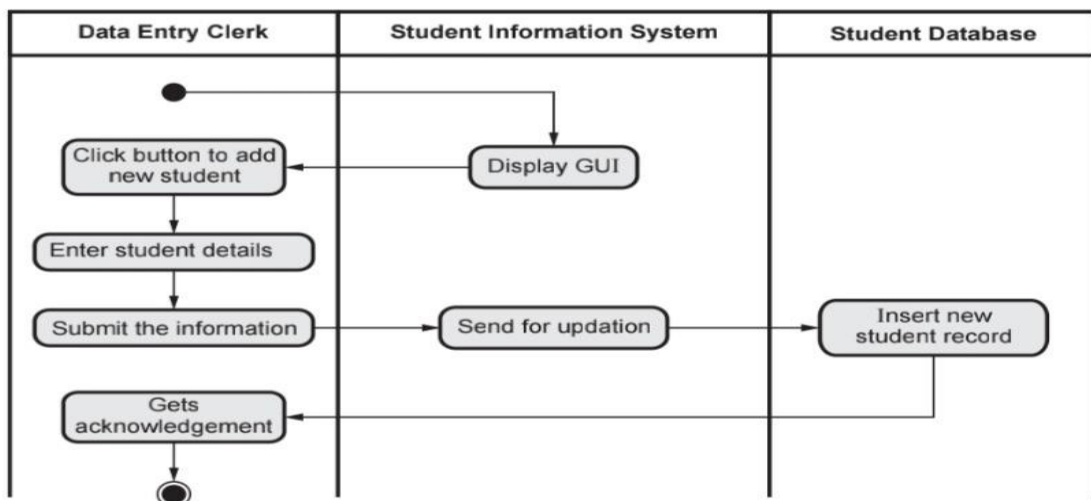
CLASS DIAGRAM

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.



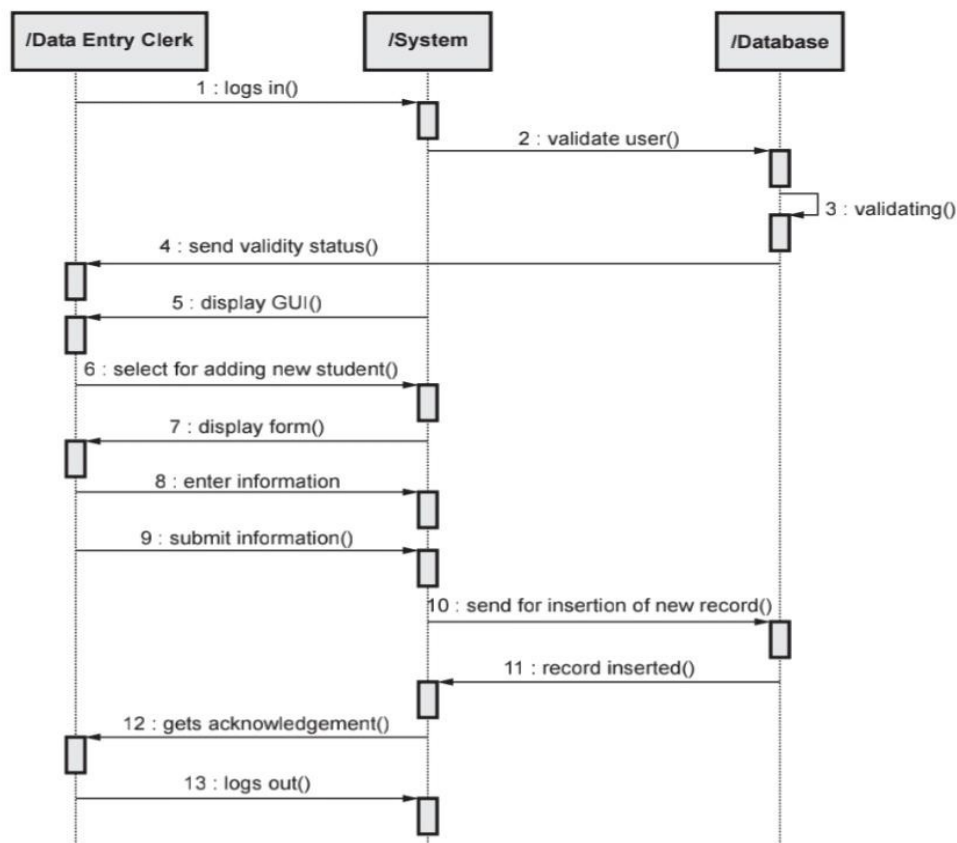
ACTIVITY DIAGRAM

Activity diagram are graphical representation of step wise 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.



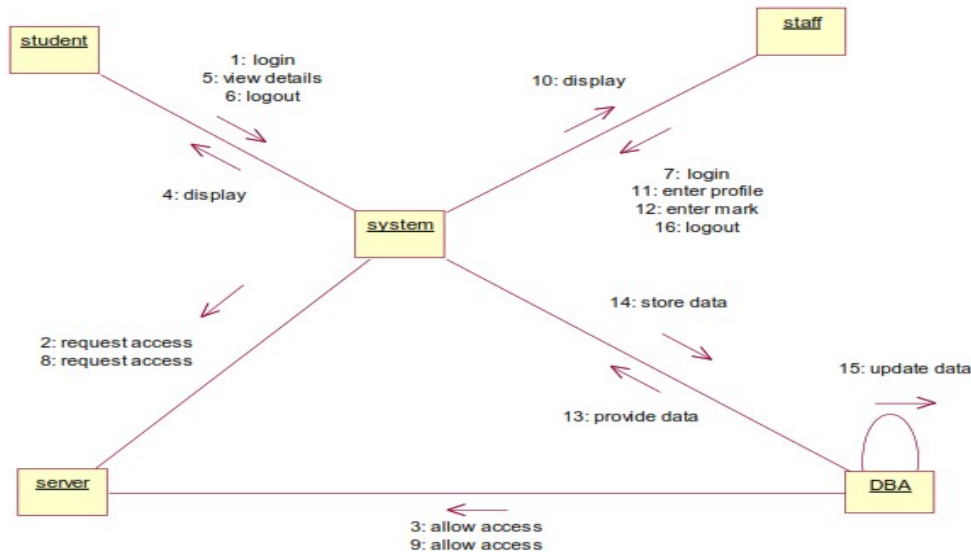
SEQUENCE DIAGRAM

A Sequence diagram represent the sequence and interaction of a given use case 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.



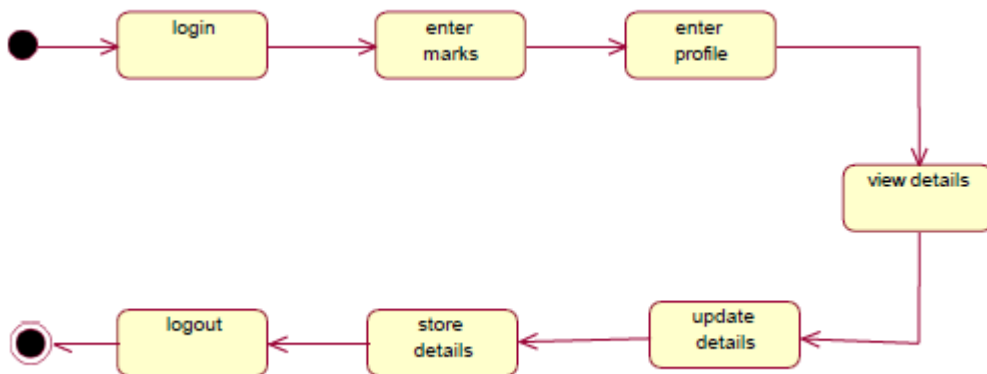
COLLABORATION DIAGRAM

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.



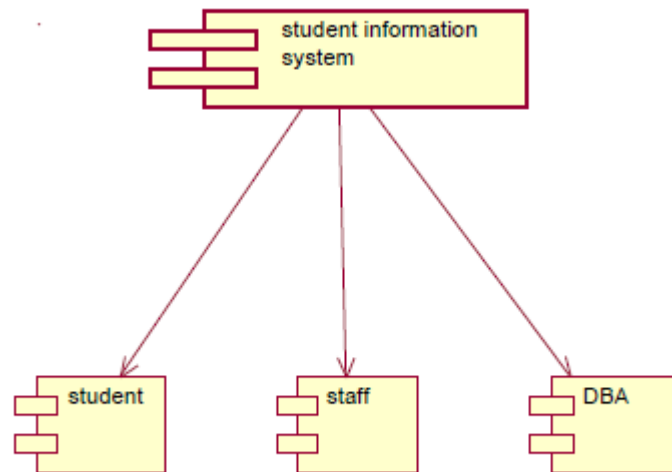
STATE CHART DIAGRAM

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.



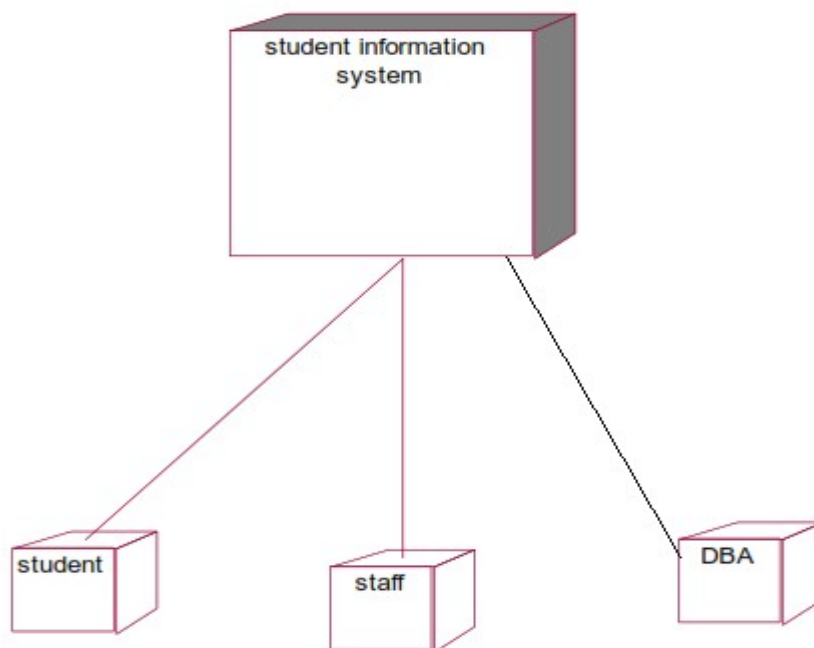
COMPONENT DIAGRAM

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.



DEPLOYMENT DIAGRAM

Deployment diagram shows the configuration of run time 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.



RESULT

VIVA QUESTIONS

1. Definition of UML.

- ❖ UML stands for Unified Modelling Language.
- ❖ It is a generic developmental modelling language used for analysis, design and implementation of software systems.

2. List the uses of UML

UML is used to create static structure diagrams based on a variety of engineering practices that have proven to be successful in the creation of complex systems.

3. What are the 7 software Development Life Cycle?

SDLC comprises seven different stages: planning, analysis, design, development, testing, implementation, and maintenance.

4. What is Water Fall Model?

Waterfall Model is a sequential model that divides software development into pre-defined phases.

5. What is Use case diagram in software?

- ❖ A use case diagram is a graphical depiction of a user's possible interactions with a system.
- ❖ The use cases are represented by either circles or ellipses.

6. What is meant by work product?

A work product is a description of content elements that are used to define anything used, produced, or modified by a task.

7. What are functional and non-functional requirements in software engineering?

- ❖ Functional requirements define a system or its component i.e. how the system must work.
- ❖ Non-functional requirements define the quality attribute of a system i.e. how the system should perform.

8. What is a model in software engineering?

Software modeling should address the entire software design including interfaces, interactions with other software, and all the software methods.

9. Relationship in class diagram?

There are six main types of relationships between classes: inheritance , realization / implementation , composition , aggregation , association, and dependency.

10. What are the two kinds of UML interaction diagrams?

- ❖ We have two types of interaction diagrams in UML.
- ❖ One is the sequence diagram and the other is the collaboration diagram.

11. Difference between sequence diagram and collaboration diagram.

SEQUENCE DIAGRAM

Clearly identify the sequence of message
Space Consume

COLLABORATION DIAGRAM

More difficult to identify the sequence of message
Space economical

12. What is the relationship between activity diagram and sequence diagram?

Activity diagrams show the workflow and how objects interact with each other, and sequence diagrams look at the messages sent between objects over time.

13. define task in software engineering

- ❖ In software engineering, tasks and work products are used to organize and track the development process.
- ❖ Tasks are specific activities that need to be completed as part of the software development process.
- ❖ They can include activities such as requirements gathering, design, coding, testing, and deployment.

14. What does FURPS stand for?

FURPS is an acronym for Functionality, Usability, Reliability, Performance and Supportability of computer systems software, but is now appropriate for the classification of many more associated systems and services.

15. How do you identify actors in a case study?

- Actors are not specific individuals or instances, but rather general types or categories of users or systems.
- To define actors, you need to ask yourself who or what initiates, participates in, or benefits from the use cases of the system.

16. What is the acquisition of knowledge?

- ❖ Knowledge acquisition refers to the process of acquiring, assimilating, and integrating new knowledge and information.
- ❖ It involves actively seeking and obtaining knowledge through various means such as reading, research, training, and learning experiences.

17.What is aggregation?

An aggregation is a special type of association in which objects are assembled or configured together to create a more complex object.

18. What is system design ?

System design is the designing the software/application as a whole [high level] that may include analysis, modeling, architecture, Components, Infrastructure etc.

19. What are metrics in software?

1. A software metric is a measurement of quantifiable or countable software characteristics.
2. Software metrics are essential for various purposes, including measuring software performance, planning work items, and measuring productivity.

20.Definition of OOSE

- ❖ (Object-Oriented Software Engineering) An object-oriented analysis and design method developed by Ivar Jacobsen.
- ❖ OOSE is known for its high-level design capabilities.

21. What is workflow in software engineering?

- 1) A Workflow is defined as a sequence of tasks that processes a set of data through a specific path from initiation to completion.
- 2) Workflows are the paths that describe how something goes from being undone to done, or raw to processed.

22. What is an artifact in software engineering?

An artifact is a byproduct of software development that helps describe the architecture, design and function of software.

23.What is object design?

The objected-oriented design is the set of defined rules/concepts to implement the functionalities within a software.

24.What is brainstorming?

Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.

25. What is swim lane diagram in software engineering?

- 1) Swim lane diagrams are flowcharts that show a process from start to finish.
- 2) These diagrams also show who is responsible for each step in the process.

26.What are the roles of project?

- 1) Project Manager
- 2) Sponsor
- 3) Business Analyst
- 4) Project planning
- 5) Communication

27. What is testing and its uses?

- 1) Testing is the practice of making objective judgments regarding the extent to which the. system (device) meets, exceeds or fails to meet stated objectives.
- 2) There are two fundamental purposes of testing: verifying procurement specifications and managing risk.

28. What are testing tools?

A testing tool is a product that supports one or more test activities, including test planning, requirements gathering, building, running tests, logging defects, and test analysis.

29. What is planning in software testing?

Planning is the process of establishing the goals and objectives of a project and determining the resources and actions needed to achieve those goals.

30.Difference between coupling and cohesion.

Coupling

Coupling is the indication of the relationship between modules.

Coupling shows the relative independence among the modules.

Coupling is a degree to which a component / module is connected to other modules.

Cohesion

Cohesion is the indication of relationship with in the module.

Cohesion shows the modules relative functional strength.

Cohesion is a degree (quality) to which a component / module focuses on the single thing.