

SVR ENGINEERING COLLEGE

AYYALURUMETTA(V),NANDYAL,KURNOOLDT.A
NDHRAPRADESH-518502



2018-2019

LABORATORYMANUAL

OF
**OBJECT ORIENTED ANALYSIS AND DESIGN &
SOFTWARE TESTING LABORATORY**
(15A05509)

(R-15REGULATION)

Preparedby

MRS.B.BINDUKALA

Asst. Professor For

B.Tech III YEAR-ISEM.(CSE)

**DEPARTMENTOFCOMPUTERSCIENCEANDENGINEERING
SVRENGINEERINGCOLLEGE**

(AFFILIATED TOJNTUAAANANTHAPURAM-AICITE-INDIA)
AYYALURUMETTA(V),NANDYAL,KURNOOLDT.A
NDHRAPRADESH-518502

LABMANUALCONTENT
OBJECT ORIENTED ANALYSIS AND DESIGN
& SOFTWARE TESTING LABORATORY
(15A05509)

Institute Vision & Mission, Department Vision & Mission

1. PO, PEO & PSO Statements.
2. List of Experiments
3. CO-PO Attainment
4. Experiment Code and Outputs

1. Institute Vision & Mission, Department Vision & Mission

Institute Vision:

To produce Competent Engineering Graduates & Managers with a strong base of Technical & Managerial Knowledge and the Complementary Skills needed to be Successful Professional Engineers & Managers.

Institute Mission:

To fulfill the vision by imparting Quality Technical & Management Education to the Aspiring Students, by creating Effective Teaching/Learning Environment and providing State – of the – Art Infrastructure and Resources.

Department Vision:

To produce Industry ready Software Engineers to meet the challenges of 21st Century.

Department Mission:

- Impart core knowledge and necessary skills in Computer Science and Engineering through innovative teaching and learning methodology.
- Inculcate critical thinking, ethics, lifelong learning and creativity needed for industry and society.
- Cultivate the students with all-round competencies, for career, higher education and self-employability.

2. PO, PEO& PSO Statements

PROGRAMME OUTCOMES (POs)

PO-1: Engineering knowledge - Apply the knowledge of mathematics, science, engineering fundamentals of Computer Science& Engineering to solve complex real-life engineering problems related to CSE.

PO-2: Problem analysis - Identify, formulate, review research literature, and analyze complex engineering problems related to CSE and reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO-3: Design/development of solutions - Design solutions for complex engineering problems related to CSE and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, cultural, societal and environmental considerations.

PO-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.

PO-5: Modern tool usage - Select/Create and apply appropriate techniques, resources and modern engineering and IT tools and technologies for rapidly changing computing needs, including prediction and modeling to complex engineering activities, with an understanding of the limitations.

PO-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 CSE professional engineering practice.

PO-7: Environment and Sustainability - Understand the impact of the CSE professional engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.

PO-8: Ethics - Apply ethical principles and commit to professional ethics and responsibilities and norms of the relevant engineering practices.

PO-9: Individual and team work - Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO-10: Communication - Communicate effectively on complex engineering activities with the engineering community and with the society-at-large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, give and receive clear instructions.

PO-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.

PO-12: Life-long learning - Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadcast context of technological changes.

Program Educational Objectives (PEOs):

PEO 1: Graduates will be prepared for analyzing, designing, developing and testing the software solutions and products with creativity and sustainability.

PEO 2: Graduates will be skilled in the use of modern tools for critical problem solving and analyzing industrial and societal requirements.

PEO 3: Graduates will be prepared with managerial and leadership skills for career and starting up own firms.

Program Specific Outcomes (PSOs):

PSO 1: Develop creative solutions by adapting emerging technologies / tools for real time applications.

PSO 2: Apply the acquired knowledge to develop software solutions and innovative mobile apps for various automation applications

2.1 Subject Time Table

SVRENGINEERING COLLEGE::NANDYAL									
DEPARTMENT OF CSE									
B.BINDUKALA						III-I			
Day/ Time	9:30 AM	10:20 AM	11:30 AM	12:20 PM-	LUNCH BREAK	02:00 PM	02:50 PM	03:40 PM	
	10:20 AM	11:10 AM	12:20 PM	01:10 PM		02:50 PM	03:40 PM	04:30 PM	
MON									
TUE									
WED									
THU						OOAD&ST LAB			
FRI									
SAT									

SVR ENGINEERING COLLEGE				
Department:	COMPUTERSCIENCE & ENGINEERING			
Course Outcome Attainment - Internal Assessments				
Name of the faculty :	B. BINDUKALA /K. AMARENDHRANATH		Academic Year:	2018-19
Branch & Section:	CSE		Exam:	EXTERNAL LAB
Course:	Object Oriented Analysis and Design & Software Testing Laboratory		Semester:	III-II SEM
Course Outcomes	Internal Lab		Internal Lab	University Exam
15A05509.1	3		3	3
15A05509.2	3		3	3
15A05509.3	3		3	3
15A05509.4	3		3	3
15A05509.5	3		3	3
Course Outcomes				Attainment Level
15A05509.1	Demonstrate fundamentals of UML Tools.			3
15A05509.2	Develop Structural diagrams for modeling complex systems.			3
15A05509.3	Develop Behavioral diagrams for modeling complex systems.			3
15A05509.4	Describe SRS and test cases for Banking applications and Library management system reporting bugs.			3
15A05509.5	Demonstrate the Win Runner Testing Tool and its implementation			3
Average Attainment				3
Overall Course Attainment				3

SVR ENGINEERING COLLEGE																																																	
DEPARTMENT			COMPUTERSCIENCE & ENGINEERING																																														
PROGRAM OUTCOME ATTAINMENT																																																	
Name of Faculty:			B. BINDUKALA /K. AMARENDHRANATH					Academic Year			2018-19																																						
Branch & Section:			COMPUTERSCIENCE & ENGINEERING					SUB CODE:			15A05509																																						
Course:			Object Oriented Analysis and Design & Software Testing Laboratory					Semester:			III-II																																						
<table border="1"> <thead> <tr> <th colspan="5">COURSE OUTCOME ATTAINMENT</th> </tr> <tr> <th>Course outcome attainment</th> <th>Internal lab</th> <th></th> <th>Internal lab</th> <th>External lab</th> </tr> </thead> <tbody> <tr> <td>15A05509.1</td> <td>3</td> <td></td> <td>3</td> <td>3</td> </tr> <tr> <td>15A05509.2</td> <td>3</td> <td></td> <td>3</td> <td>3</td> </tr> <tr> <td>15A05509.3</td> <td>3</td> <td></td> <td>3</td> <td>3</td> </tr> <tr> <td>15A05509.4</td> <td>3</td> <td></td> <td>3</td> <td>3</td> </tr> <tr> <td>15A05509.5</td> <td>3</td> <td></td> <td>3</td> <td>3</td> </tr> </tbody> </table>															COURSE OUTCOME ATTAINMENT					Course outcome attainment	Internal lab		Internal lab	External lab	15A05509.1	3		3	3	15A05509.2	3		3	3	15A05509.3	3		3	3	15A05509.4	3		3	3	15A05509.5	3		3	3
COURSE OUTCOME ATTAINMENT																																																	
Course outcome attainment	Internal lab		Internal lab	External lab																																													
15A05509.1	3		3	3																																													
15A05509.2	3		3	3																																													
15A05509.3	3		3	3																																													
15A05509.4	3		3	3																																													
15A05509.5	3		3	3																																													
COURSE OUTCOMES AND PROGRAM OUTCOMES MAPPING																																																	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2																																			
15A05509.1	3	3	2	2	1	2			1		1	2	3	1																																			
15A05509.2	3	2	1	1		1		1				1	2	2																																			
15A05509.3	3	3	2	2	2	1	1			1			2	1																																			
15A05509.4	3	3	1	2		2			2		2	2	2	2																																			
15A05509.5	3	2	2	2		2		2		1			3	1																																			
AVERAGE	3.0	2.6	1.6	1.8	1.5	1.6	1.0	1.5	1.5	1.0	1.5	1.7	2.4	1.4																																			

PO- ATTAINMENT															
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
INTERNAL	15A05509.1	9	9	6	6	3	6			3		3	6	9	3
	15A05509.2	9	6	3	3		3		3				3	6	6
	15A05509.3	9	9	6	6	6	3	3			3			6	3
	15A05509.4	9	9	3	6		6			6		6	6	6	6
	15A05509.5	9	6	6	6		6		6		3			9	3
UNIVERSITY	15A05509.1	9	9	6	6	3	6			3		3	6	9	3
	15A05509.2	9	6	3	3		3		3				3	6	6
	15A05509.3	9	9	6	6	6	3	3			3			6	3
	15A05509.4	9	9	3	6		6			6		6	6	6	6
	15A05509.5	9	6	6	6		6		6		3			9	3
OVERALL	15A05509.1	3	3	3	3	3	3			3		3	3	3	3
	15A05509.2	3	3	3	3		3		3				3	3	3
	15A05509.3	3	3	3	3	3	3	3			3			3	3
	15A05509.4	3	3	3	3		3			3		3	3	3	3
	15A05509.5	3	3	3	3		3		3		3			3	3
Attainment		3	3	3	3	3	3	3	3	3	3	3	3	3	3
Faculty: B. BINDUKALA									Head of the Department						

LIST OF EXPERIMENTS:

SYLLABUS

(15A05509) OBJECT ORIENTED ANALYSIS AND DESIGN & SOFTWARE TESTING **LABORATORY**

UML PROGRAMS

1. Use Case diagram
2. Class Diagram
3. Sequence Diagram
4. Collaboration Diagram
5. State Diagram
6. Activity Diagram
7. Component Diagram
8. Deployment Diagram
9. Test Design.

Problems that may be considered are

1. College Information System.
2. Hostel Management.
3. ATM system.

Testing Lab Programs

1. Write programs in „C“ Language to demonstrate the working of the following constructs: i) do...while ii) while....do iii) if...else iv) switch v) for
2. A program written in „C“ language for Matrix Multiplication fails|| Introspect the causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the various bugs
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any testing tool (e.g. Win runner)
7. Study of any web testing tool (e.g. Selenium)
8. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9. Study of any test management tool (e.g. Test Director)
10. Study of any open source-testing tool (e.g. Test Link)

Suggested Software Tools: Rational Rose, Visual paradigm, Turbo C (Editor), Testing Tools like Win Runner tool, Selenium, Bugzilla, Test Director, Test Link

Introduction

In late 1960's people were concentrating on Procedure Oriented Languages such as **COBOL, FORTRAN, PASCAL...etc.** Later on they preferred **Object Oriented Languages**. In the middle of 1970-80 three Scientists named as BOOCH, RUMBAUGH and JACOBSON found a new language named as **Unified Modeling Language**. It encompasses the Designing of the System/Program. It is a **De facto** language.

What is UML?

- Is a **language**. It is not simply a notation for drawing diagrams, but a complete language for capturing knowledge (semantics) about a subject and expressing knowledge (syntax) regarding the subject for the purpose of communication.
- Applies to **modeling** and systems. Modeling involves a focus on understanding a subject (system) and capturing and being able to communicate in this knowledge.
- It is the result of **unifying** the information systems and technology industry's best engineering practices (principles, techniques, methods and tools).
- used for both database and software modeling

Overview of the UML

- The UML is a language for
 - visualizing
 - specifying
 - constructing
 - documentingThe artifacts of a software-intensive system

Visual modeling (visualizing)

- A picture is worth a thousand words!
 - Uses standard graphical notations
 - Semi-formal
 - Captures Business Process from enterprise information systems to distributed

Web-based applications and even to hard real time embedded systems

Specifying

- building models that are: Precise, Unambiguous, Complete
- UML symbols are based on well-defined syntax and semantics.
- UML addresses the specification of all important analysis, design, and implementation decisions.

Constructing

- Models are related to OO programming languages.
- Round-trip engineering requires tool and human intervention to avoid information loss
 - Forward engineering—direct mapping of a UML model into code.
 - Reverse engineering—reconstruction of a UML model from an implementation.

Documenting

- Architecture, Requirements, Tests, Activities (Project planning, Release management)

Conceptual Model of the UML

To understand the UML, you need to form a conceptual model of the language, and this requires learning three major elements.

Elements:

1. Basic building blocks
2. Rules
3. Common Mechanisms

Basic Building Blocks of the UML

The vocabulary of the UML encompasses three kinds of building blocks:

- Things
- Relationships
- Diagrams

1. Structural Things

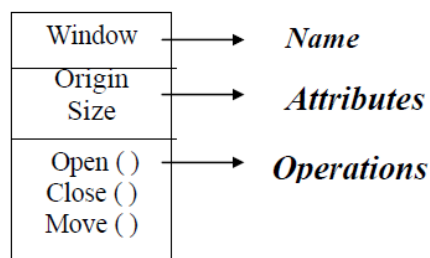
- These are the Nouns and Static parts of the model.
- These are representing conceptual or physical elements.

There are seven kinds of structural things:

1. Class
2. Interface
3. Collaboration
4. UseCase
5. ActiveClass
6. Component
7. Node

1. Class

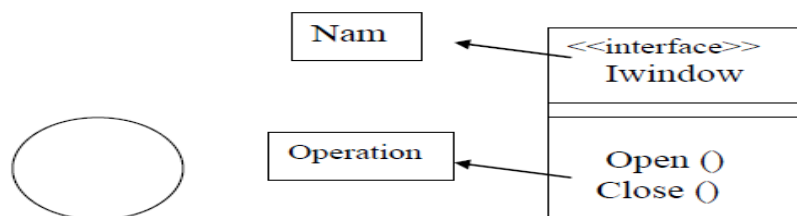
Is a description of a set of objects that share the same attributes, operations, methods, relationships and semantics.



A Simple Class

2. Interface

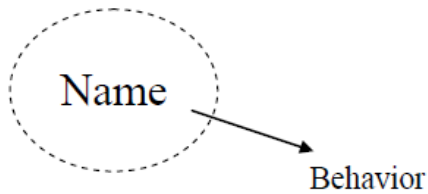
A collection of operations that specify a service (for a resource or an action) of a class or component. It describes the externally visible behavior of that element.



A Simple Interface

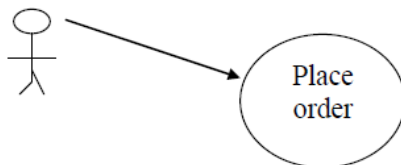
3. Collaboration

- Define an interaction among two or more classes.
- Define a society of roles and other elements.
- Provide cooperative behavior.
- Capture structural and behavioral dimensions.
- UML uses `_pattern` as a synonym (careful)



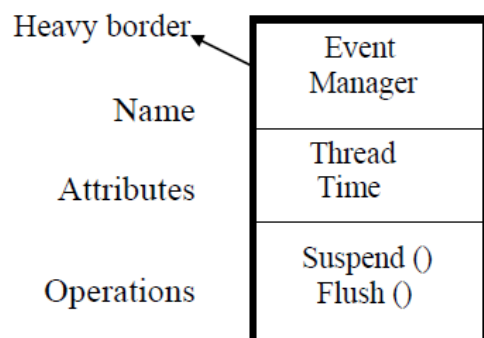
4. Use case

- A sequence of actions that produce an observable result for a specific actor.
- A set of scenarios tied together by a common user goal.
- Provides a structure for behavioral things.
- Realized through a collaboration (usually realized by a set of actors and the system to be built).



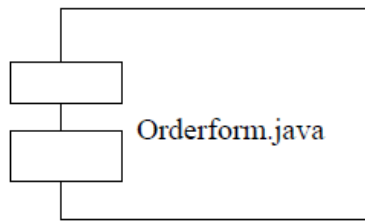
5. Active Class

- Special class whose objects own one or more processes or threads.
- Can initiate control activity



6. Component

- Replaceable part of a system.
- Components can be packaged logically.
- Conform to a set of interfaces.
- Provide the realization of an interface.
- Represents a physical module of code



7. Node

- Element that exists at run time.
- Represents a computational resource.
- Generally has memory and processing power.



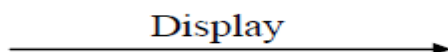
2. Behavioral Things

- These are Verbs of UML models.
- These are Dynamic parts of UML models: “behavior over time and space”
- Usually connected to structural things in UML.

There are two kinds of Behavioral Things:

1. Interaction

- Is a behavior of a set of objects comprising a set of message exchanges within a particular context to accomplish a specific purpose.



2. State Machine

- Is a behavior that specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together with its responses to those events.

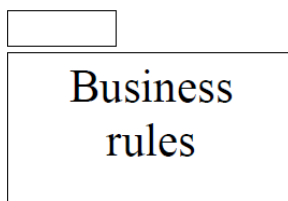


3.Grouping Things

- These are the organizational parts of the UML models.
- There is only one primary kind of grouping thing:

1.Packages

- General purpose mechanism for organizing elements into groups.
- Purely conceptual; only exists at development time.
- Contains behavioral and structural things.
- Can be nested.
- Variations of packages are: Frameworks, models, & subsystems.



4.Annotation Things

- These are Explanatory parts of UML models
- These are the Comments regarding other UML elements (usually called adornments in UML)

There is only one primary kind of annotation thing:

1.Note

A note is simply a symbol for rendering constraints and comments attached to an element or collection of elements.

Is a best expressed in informal or formal text.



Relationships

There are four kinds of relationships:

1. Dependency
2. Association
3. Generalization
4. Realization

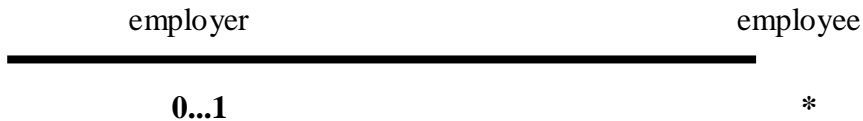
- These relationships tie things together.
- It is a semantic connection among elements.
- These relationships are the basic relational building blocks of the UML

1. Dependency

Is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing)

2. Association

Is a structural relationship that describes a set of links, a link being a connection among objects.



Aggregation

» Is a special kind of association. It represents a structural relationship between the whole and its parts.

» Represented by black diamond.

3. Generalization

Is a specialization/generalization relationship in which objects of the specialized element (the child) are more specific than the objects of the generalized element

Diagrams

- A diagram is the graphical presentation of a set of elements.
- Represented by a connected graph: Vertices are things; Arcs are behaviors.

UML includes nine diagrams:

- ClassDiagram;
- ObjectDiagram
- UsecaseDiagram
- SequenceDiagram;
- CollaborationDiagram
- StatechartDiagram
- ActivityDiagram
- ComponentDiagram
- DeploymentDiagram

Both Sequence and Collaboration diagrams are called Interaction Diagrams.

COLLEGE INFORMATION SYSTEM

AIM

To develop a College Information System using Rational Rose Software and to implement the software in Java.

PROBLEM ANALYSIS AND PROJECT PLANNING

A College Information System (CIS) is a software application for educational establishments to manage university data. College Information System provide capabilities for entering student test and other assessment scores, building schedules, tracking student attendance, and managing many other university-related data needs in a school, college or university.

PROBLEM STATEMENT

- a. Effective for Administration Purpose
- b. Cheap

c. Better Service

UML DIAGRAMS:

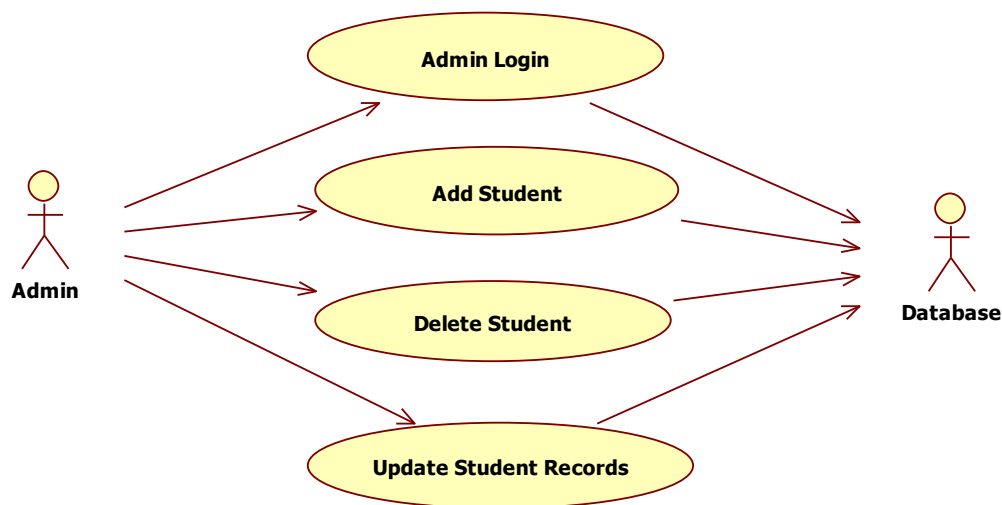
The following UML diagrams describe the process involved in the online system

1. Use case diagram
2. Class diagram
3. Sequence diagram
4. Collaboration diagram
5. Activity diagram
6. Component diagram
7. Activity diagram
8. Deployment Diagram

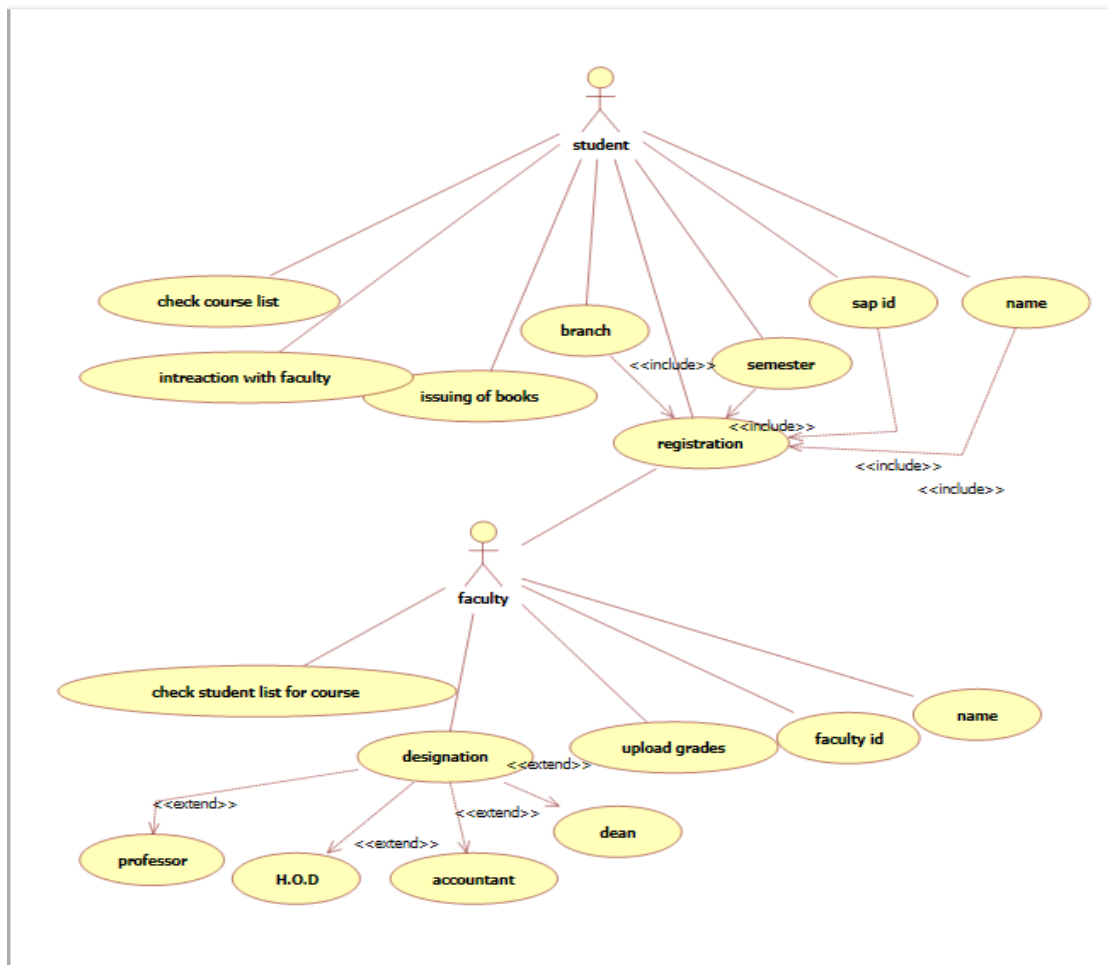
USE CASE DIAGRAM:

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

For Administrator:



College



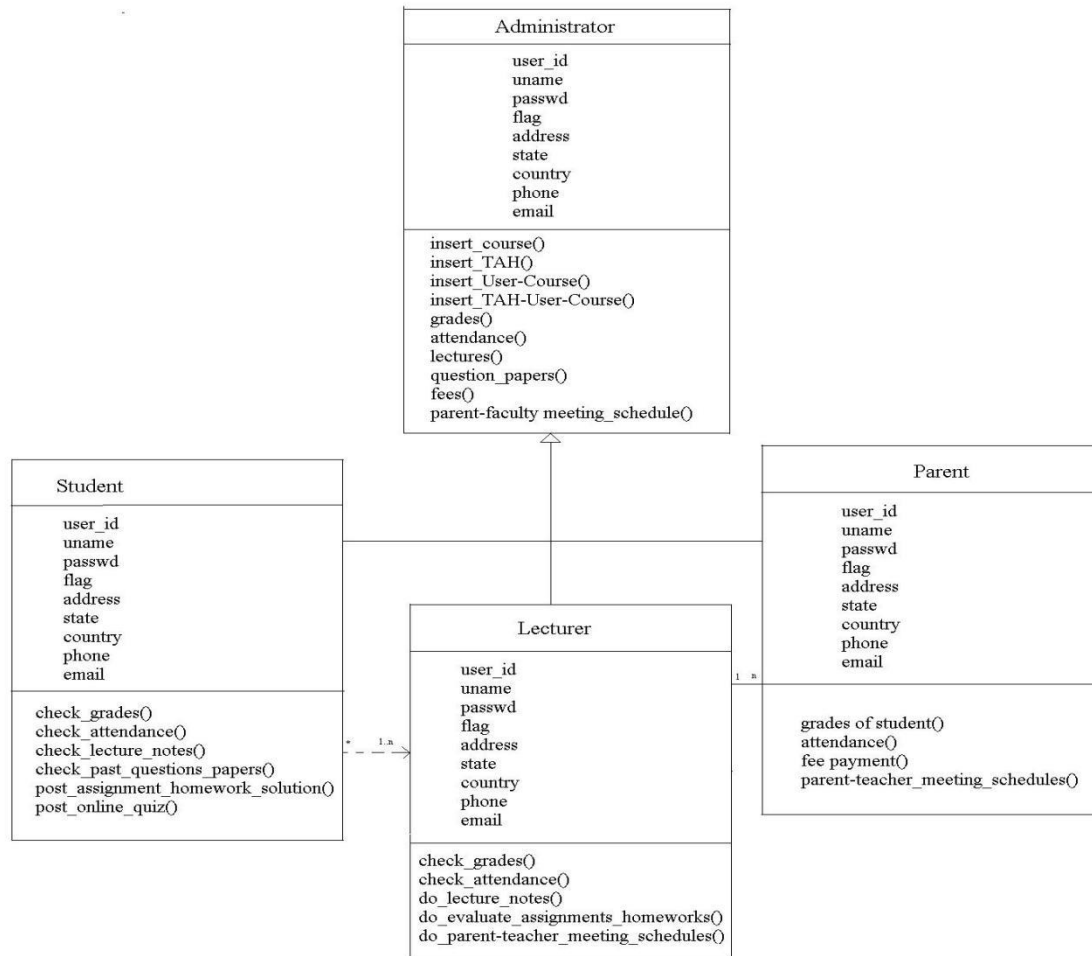
DOCUMENTATION OF USE CASE DIAGRAM

The actors in this use case diagram are Admin, Student, Database. The use cases are the activities performed by actors.

- Admin register login, and store the student records details in database.
- Student Register from the Student Login process.
- Then the database is searched for details and verified.
- Database stores the details and returns acknowledgement

CLASS DIAGRAM:

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



DOCUMENTATION OF CLASS DIAGRAM

This class diagram has three classes Login, Student details and Update details in database.

a. **Students** – is the class name. Its attributes are name, Address, DOB, Gender, College, Subjects, Semester, Year, Degree, and Branch. The operations Performed in the students class, Store database and Update.

b. **Administration**– is the class name. Its attributes are Login, Password and database. The operations performed are Student Details store in database and send acknowledgement.

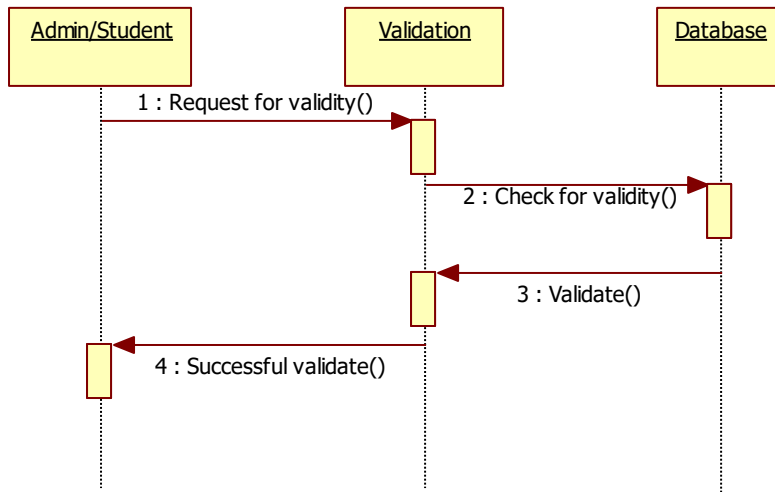
c. **Database** – is the class name. The operations performed are storing Search and storing the values.

SEQUENCE DIAGRAM:

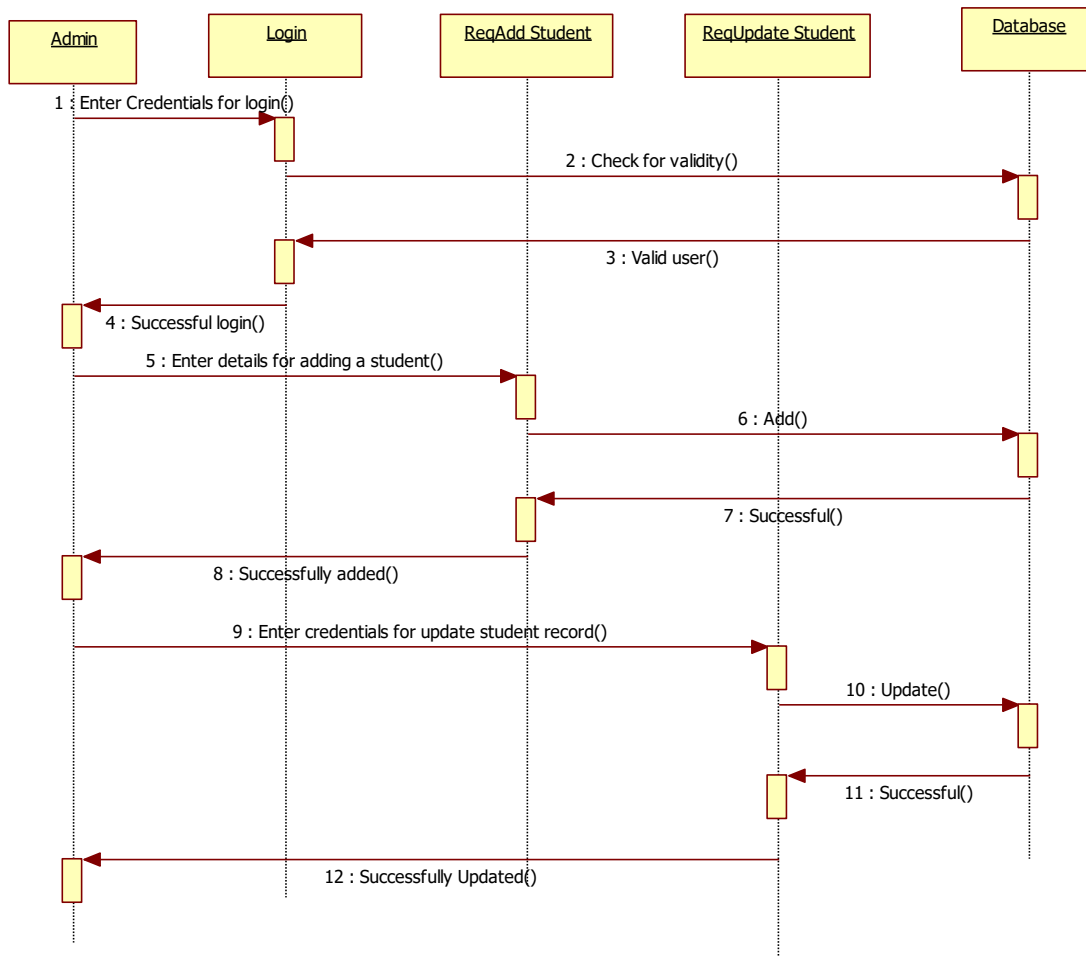
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

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

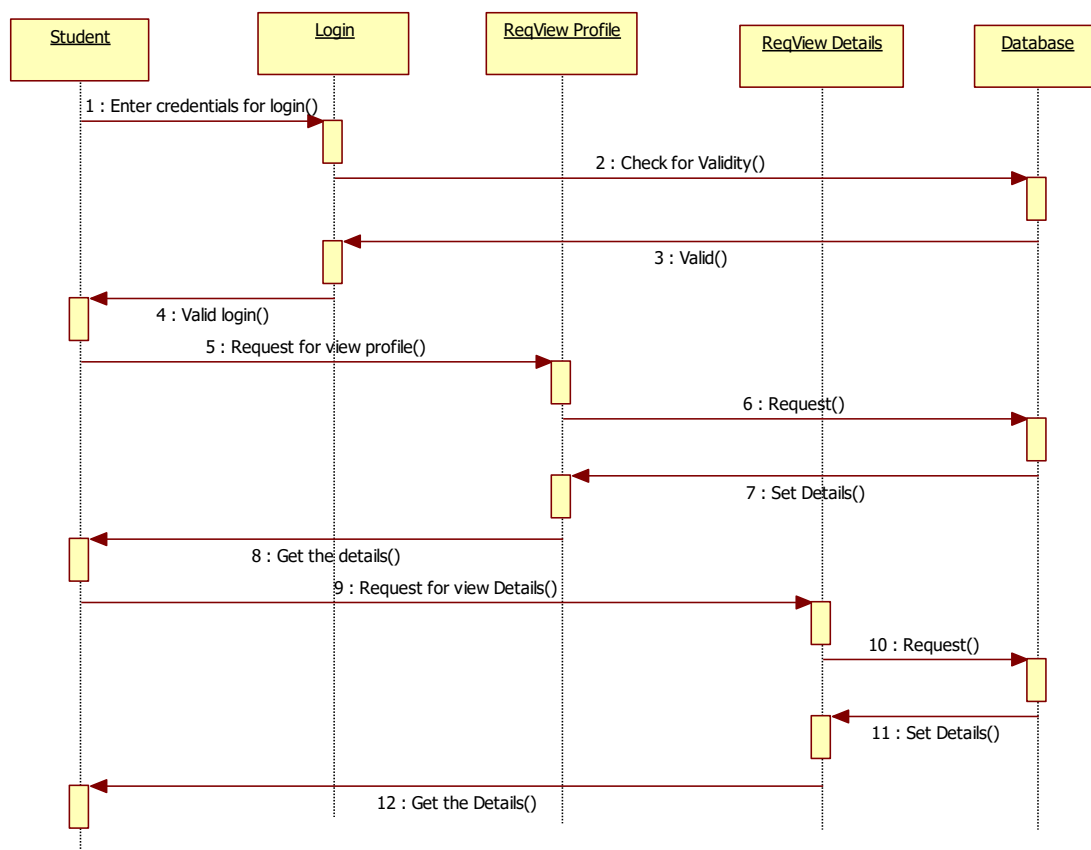
For Validity:



For Administrator:



For Student:



DOCUMENTATION OF SEQUENCE DIAGRAM

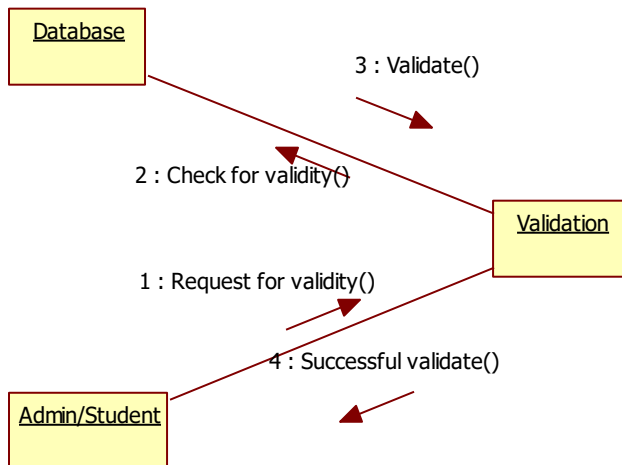
The sequence diagram describes the sequence of steps to show

- The Admin login and registering for add Student Details.
- The verification done by the interface and sending acknowledgement for registration.
- Searching the database with login and displaying it for maintenance.

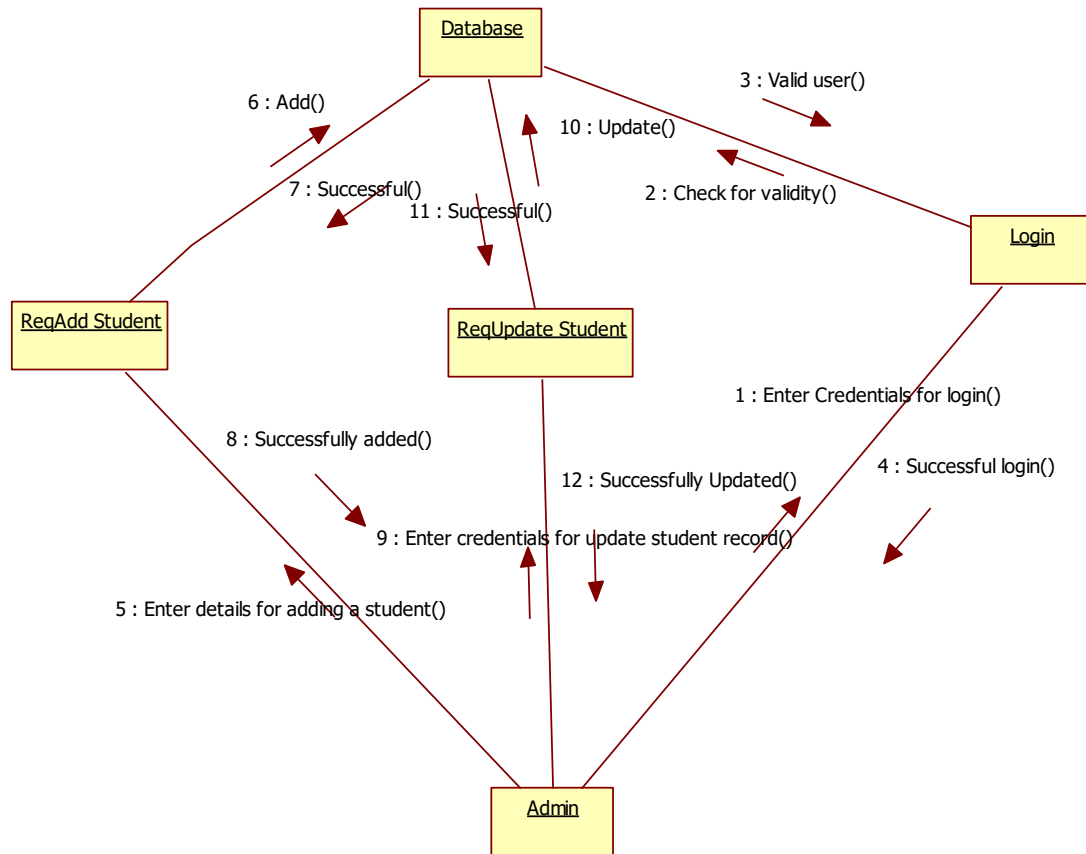
COLLABRATION DIAGRAM:

A collaboration diagram, also called a communication diagram or interaction diagram. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice. 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

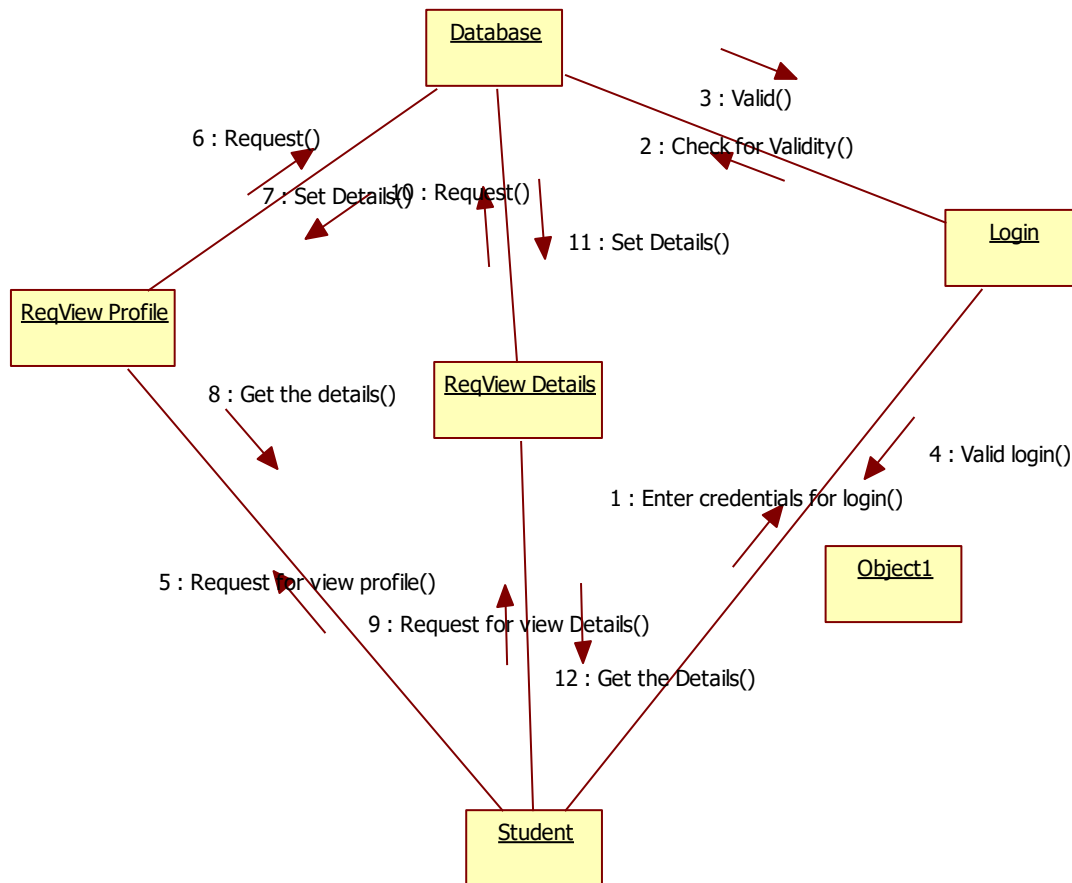
For Validity:



For Administrator:



For Student:



DOCUMENTATION OF COLLABRATION DIAGRAM

The collaboration diagram is to show how the Student registers and the authorities maintain the details of the registered students in the Information system. Here the sequence is numbered according to the flow of execution.

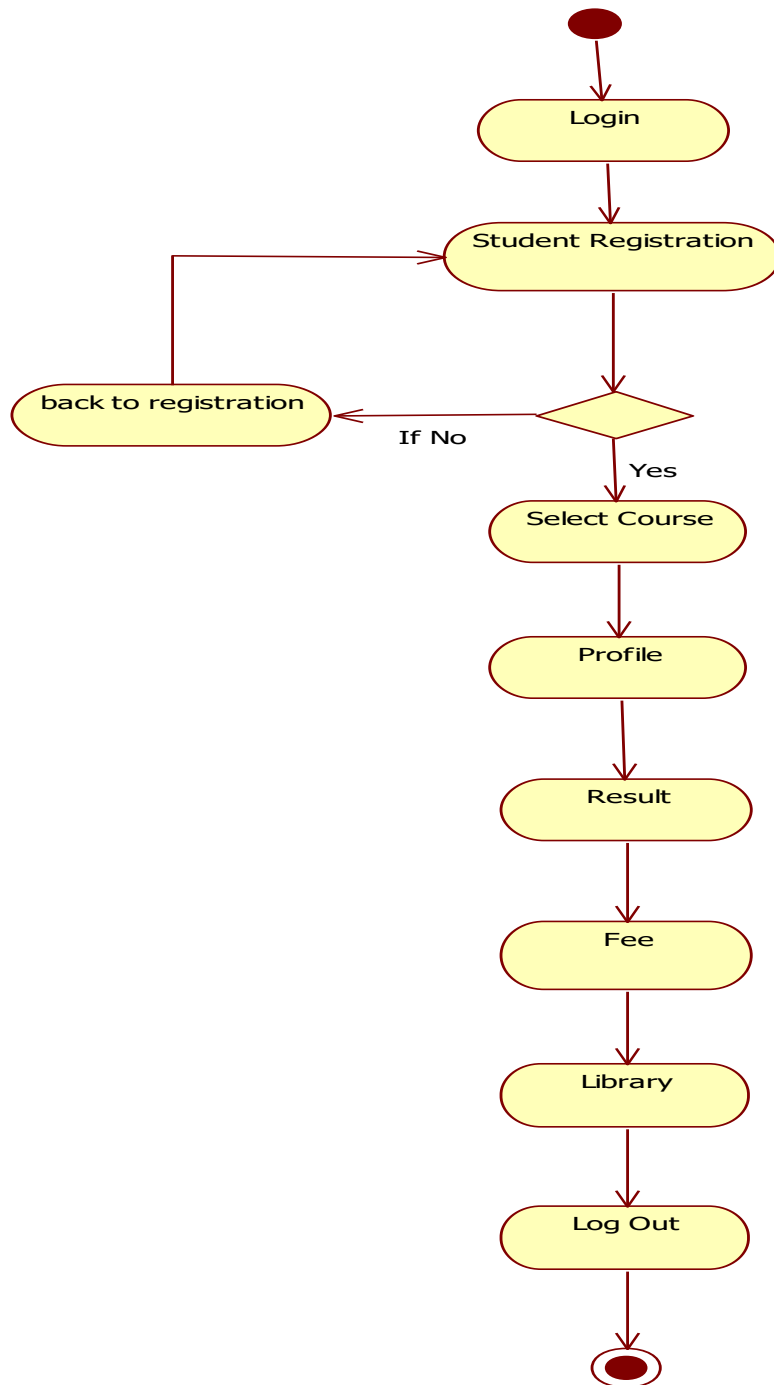
ACTIVITY DIAGRAM:

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

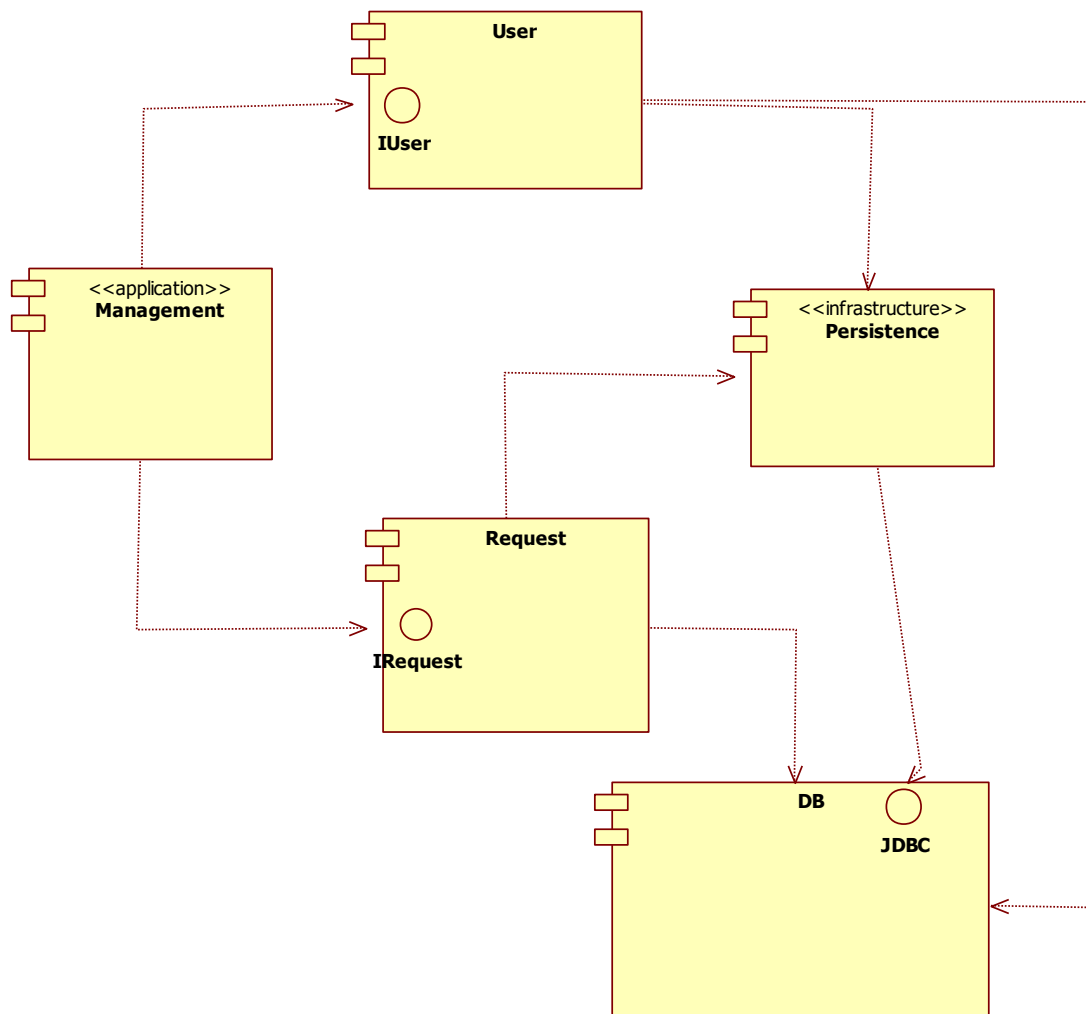
DOCUMENTATION OF ACTIVITY DIAGRAM

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

- a. The student details are Add and stored in database.
- b. Select the course from the given Course by student.
- c. Search Profile and Result with login and if data present in the database.
- d. The searched data is displayed if available and then Log Out.

**COMPONENT DIAGRAM:**

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

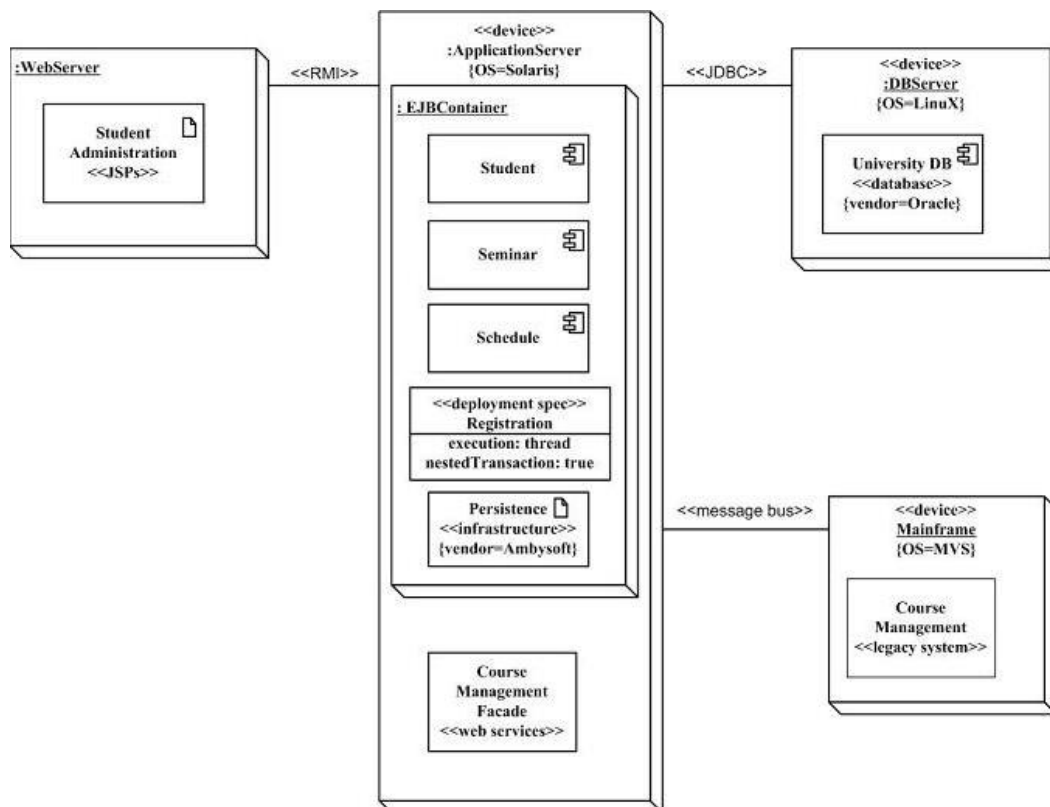


DOCUMENTATION OF COMPONENT DIAGRAM

The main component in this component diagram is Student Information system. And register, User and Manage, Request details are the components comes under the main component.

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-dimentional box. Dependencies are represented by communication association.



DOCUMENTATION OF DEPLOYMENT DIAGRAM

The processor in this deployment diagram is the University Information System which is the main part and the Student are the Admin, verify and search which are the some of the main activities performed in the system.

Hostel Management

AIM

To develop a **Hostel Management** using Rational Rose Software and to implement the software in Java.

PROBLEM ANALYSIS AND PROJECT PLANNING

Hostel Management System is a Customize and user-friendly software for Hostel. It has been designed to automate, manage and look after the over-all processing of even very large hostel. It is capable of managing Enquiry details, Student Details, Payment Details etc. Hostel Management System is a Customize and user-friendly software for

Hostel which provide hostel information, hostel room information, hostel accounts information.

Hostel Management Software System is offering a maximum of stability, cost-effectiveness and usability. It provides the most flexible and adaptable standards management system software solutions for hostel.

PROBLEM STATEMENT

- a. Effective for Administration Purpose
- b. Cheap
- c. Better Service

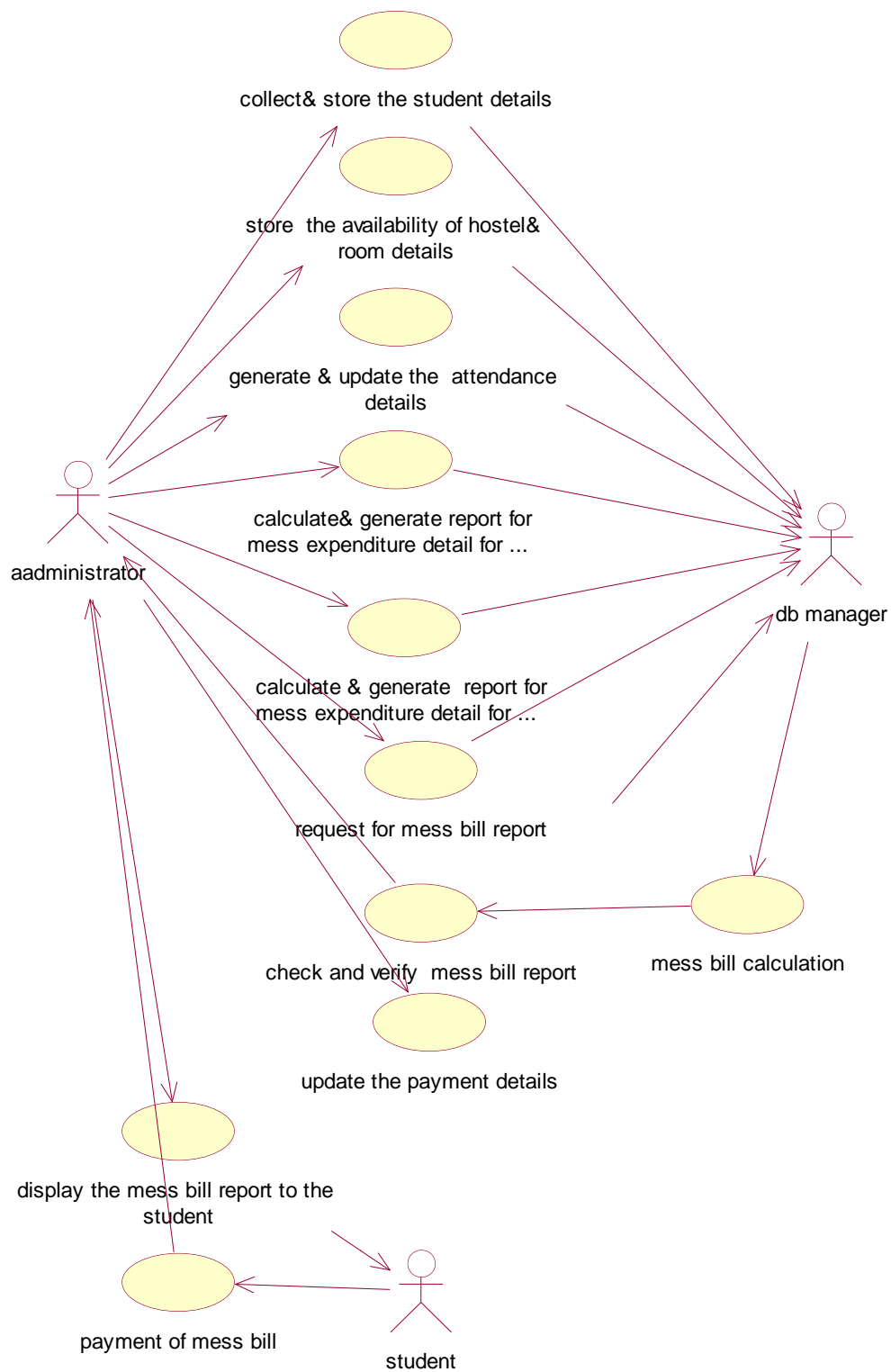
UML DIAGRAMS:

The following UML diagrams describe the process involved in the online system

- 1. Use case diagram
- 2. Class diagram
- 3. Sequence diagram
- 4. Collaboration diagram
- 5. Activity diagram
- 6. Component diagram
- 7. Activity diagram
- 8. Deployment Diagram

USE CASE DIAGRAM:

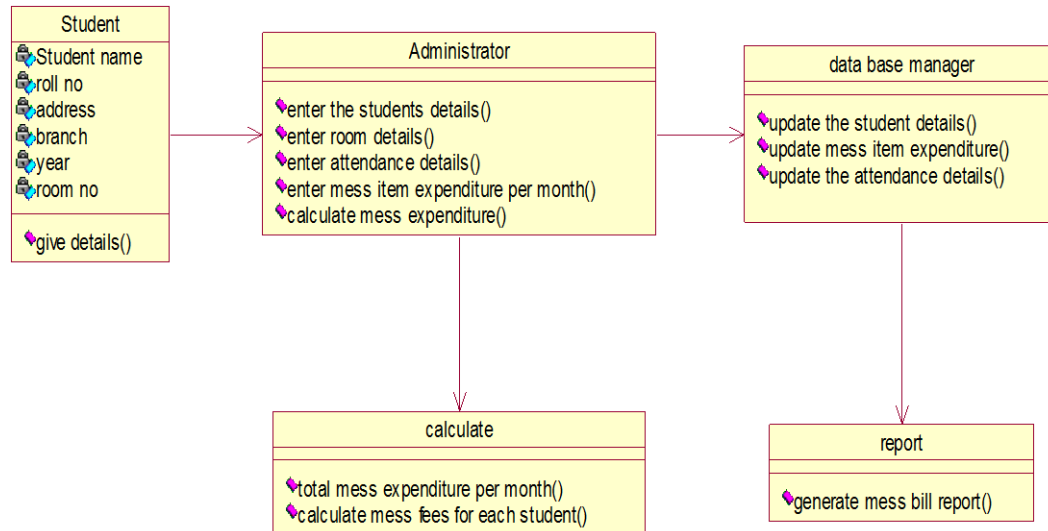
A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It is represented using ellipse. Actor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



CLASS DIAGRAM:

A class diagram in the unified modeling language (UML) is a type of static structure

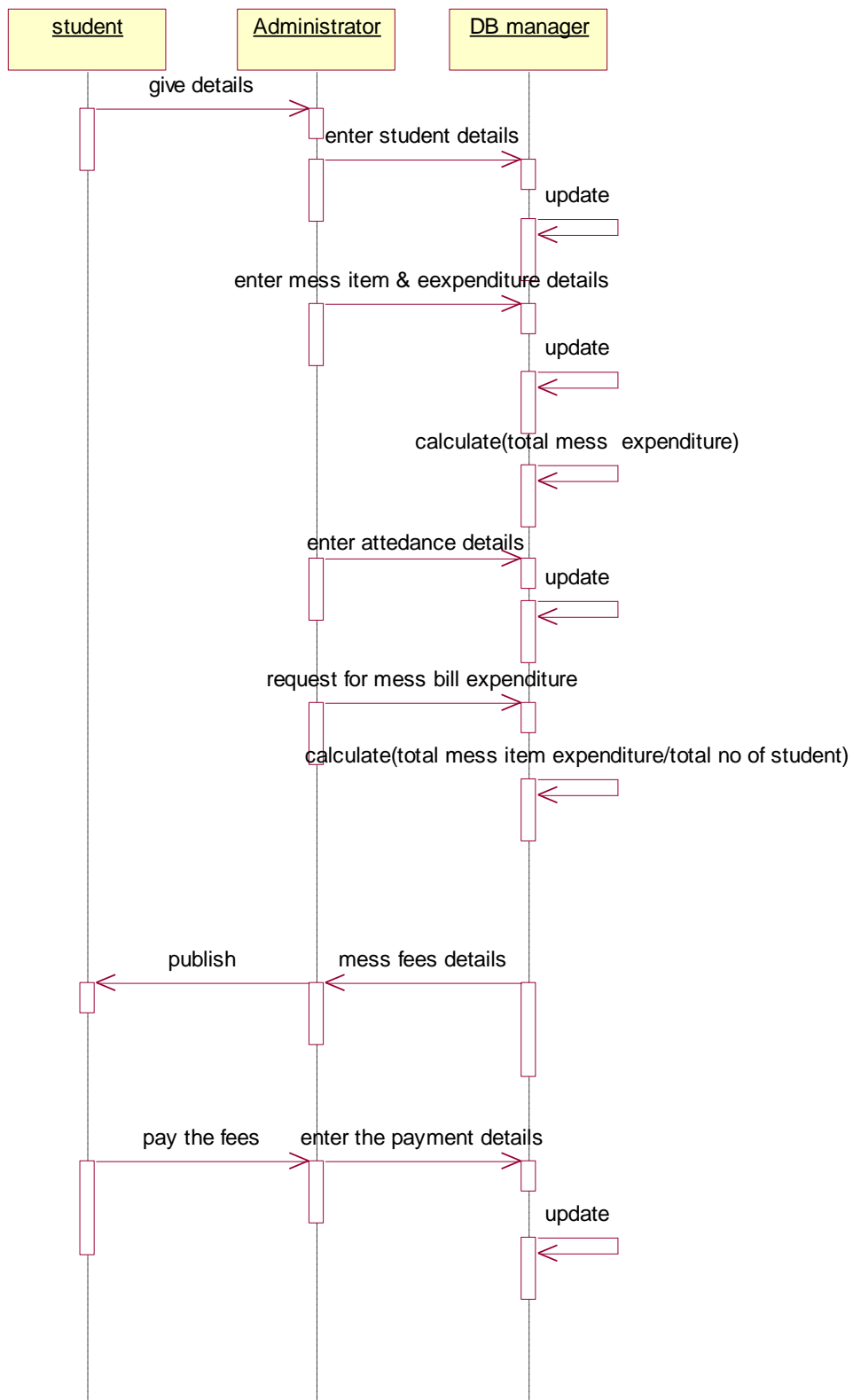
diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. It is represented using a rectangle with three compartments. Top compartment have the class name, middle compartment the attributes and the bottom compartment with operations.



SEQUENCE DIAGRAM:

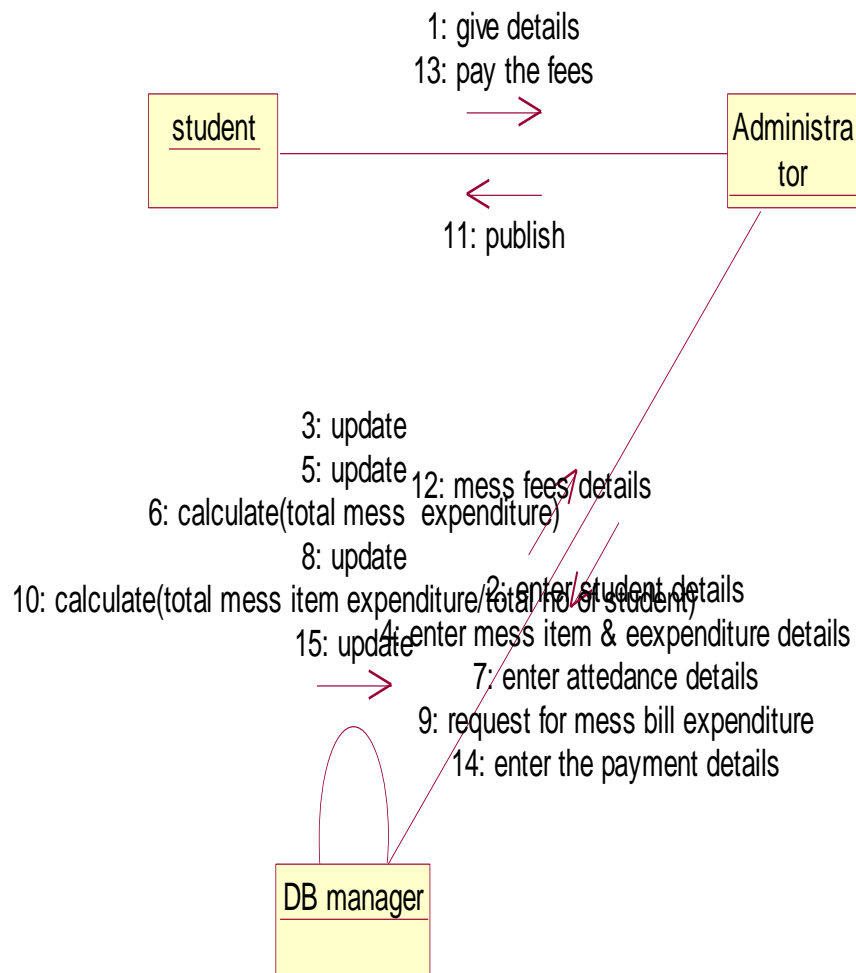
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. There are two dimensions.

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



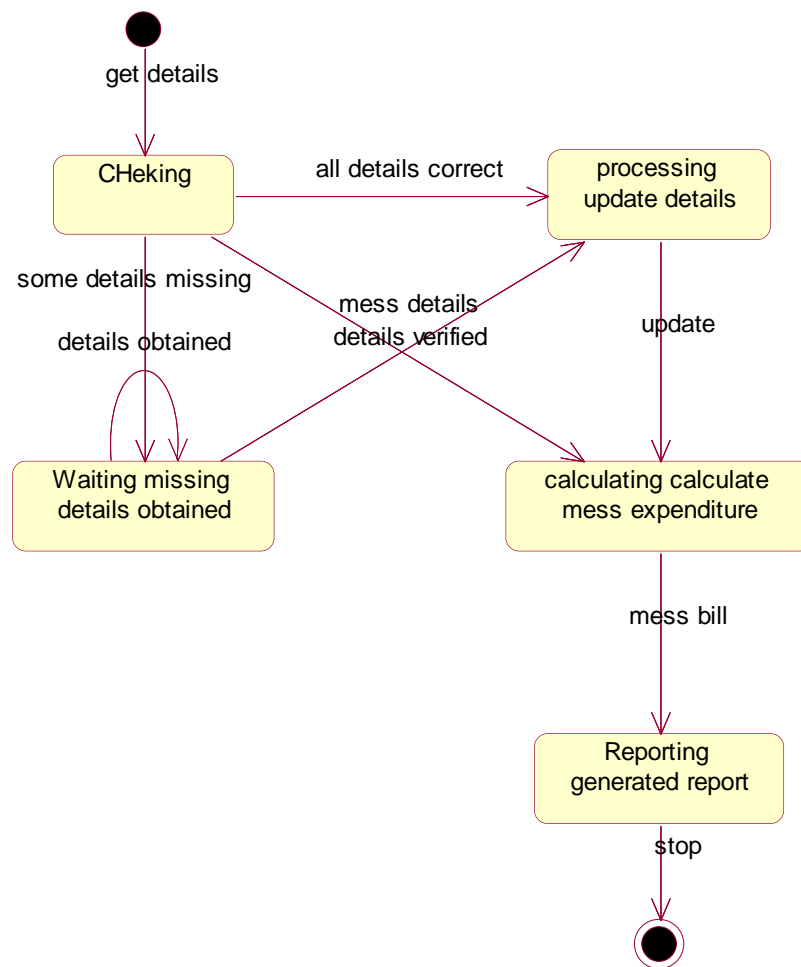
COLLABRATION DIAGRAM:

A collaboration diagram, also called a communication diagram or interaction diagram. A sophisticated modeling tool can easily convert a collaboration diagram into a sequence diagram and the vice. 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



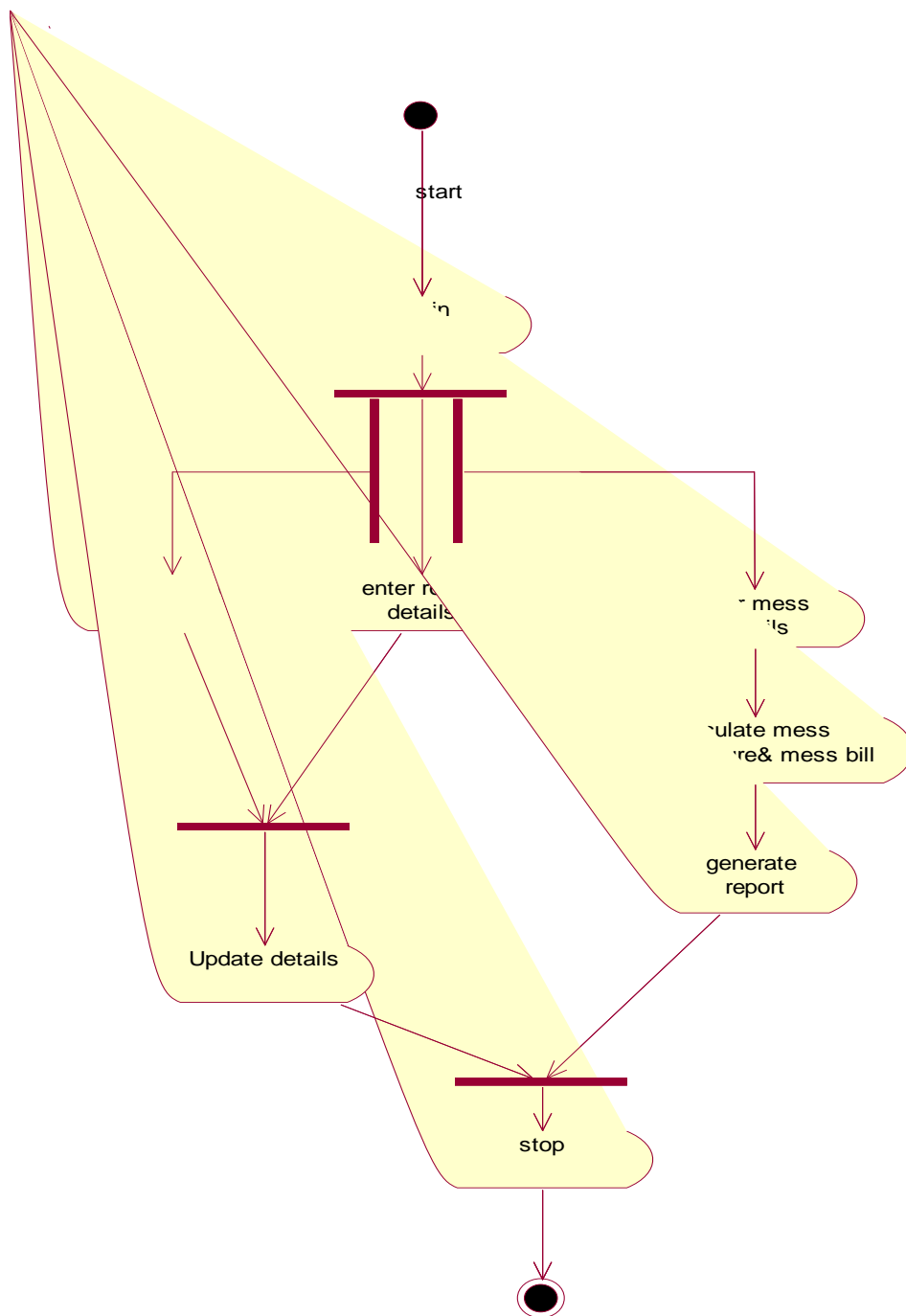
State Diagram:

State diagram 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 hoe the objects state changes as a result of events that reach the object.



ACTIVITY DIAGRAM:

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



ATM SYSTEM

Aim:

To create a system to perform Bank ATM transaction

Theory

Problem analysis and project planning

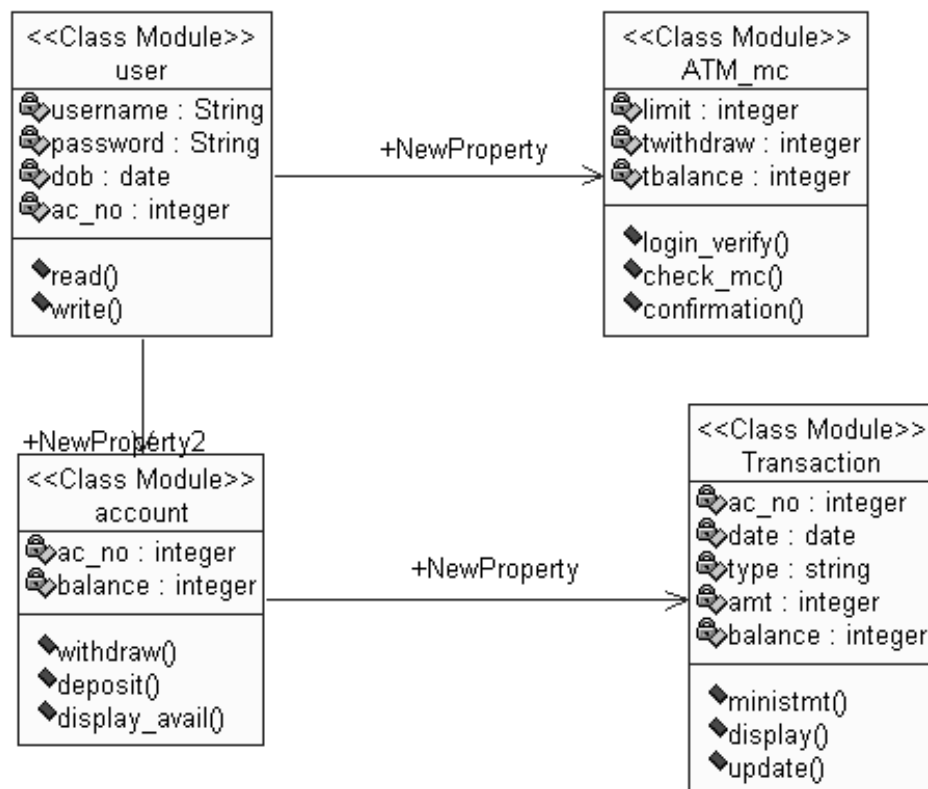
Introduction

Banking is one of the common and day to day attribute of life. Nowadays it is totally different from that existed a few years ago banking has become completely computerized new facilities such as credit cards, debit cards & ATM has been introduced. ATM is automatic teller machine which is basically used to withdraw money from an account.

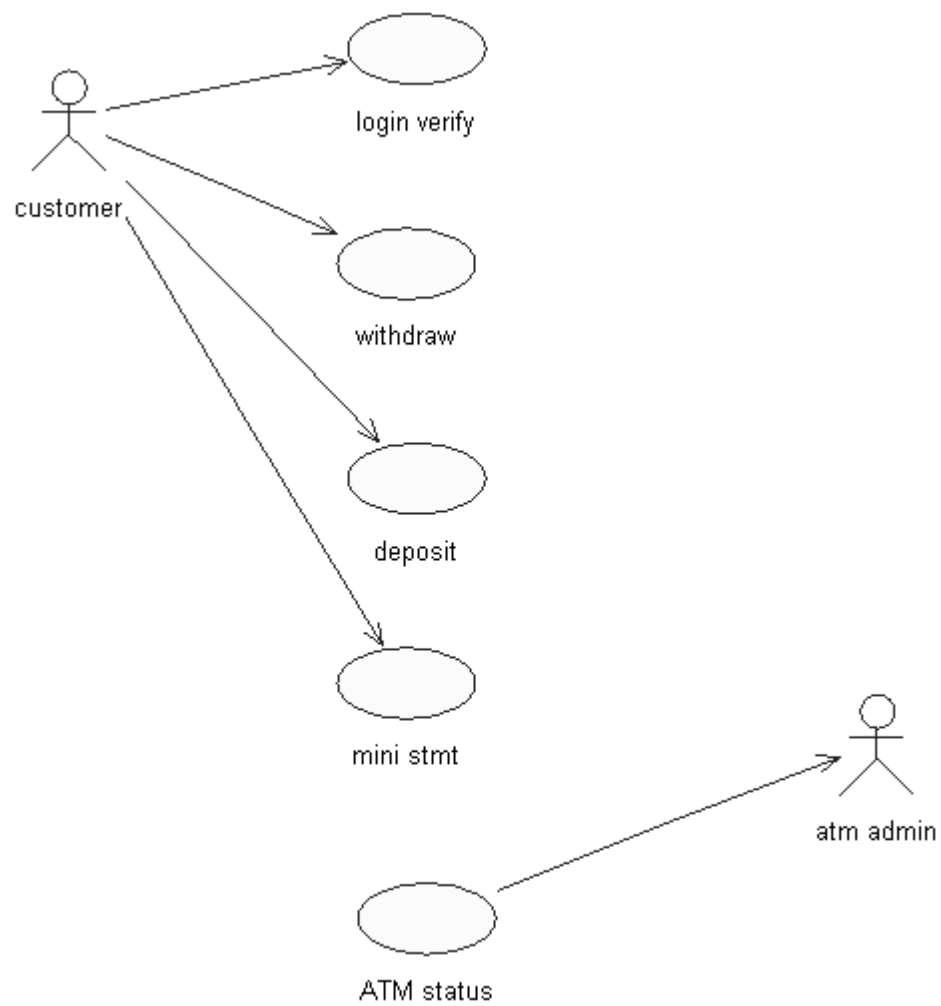
Diagrams:

1. Class diagram.
2. Use case diagram.
3. Activity diagram.
4. Sequence diagram.
5. Collaboration diagram.
6. Component diagram.
7. Deployment diagram.
8. State transition diagram

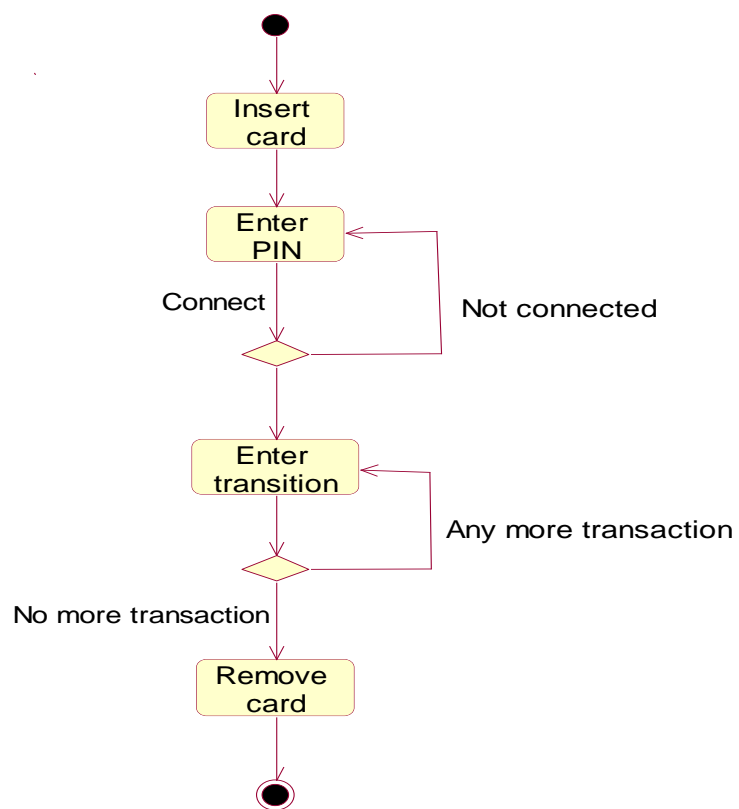
Class diagram for ATM system



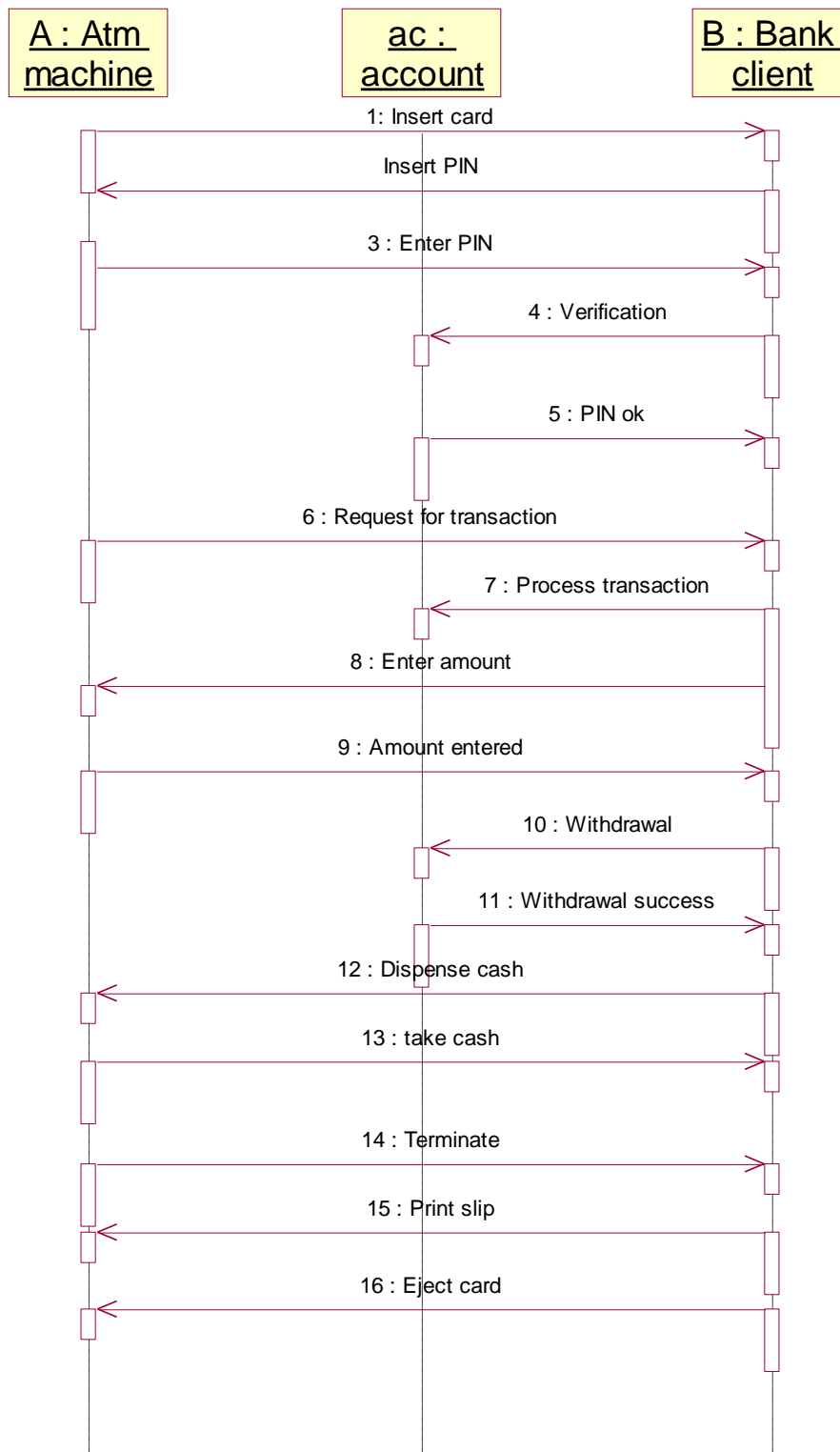
Use Case Diagram



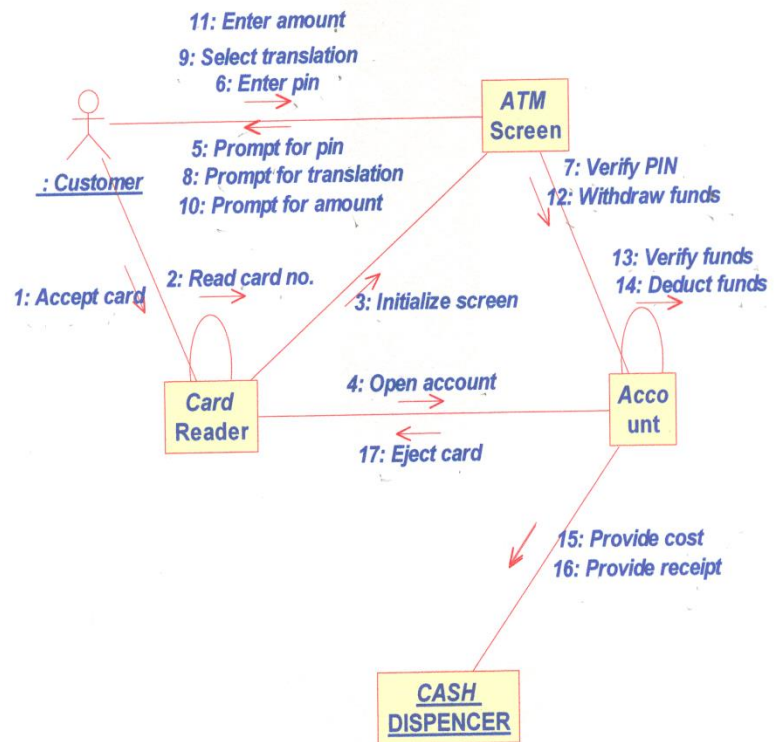
Activity Diagram



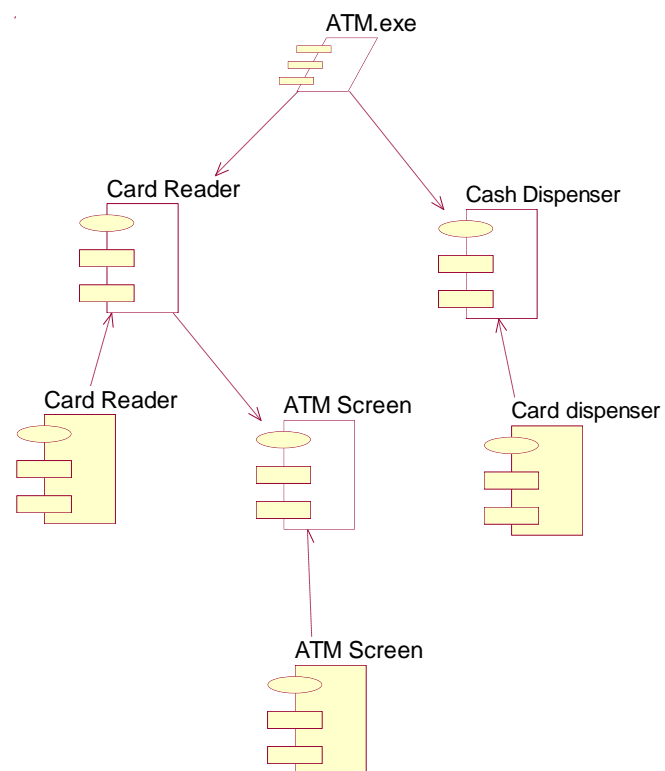
Sequence Diagram



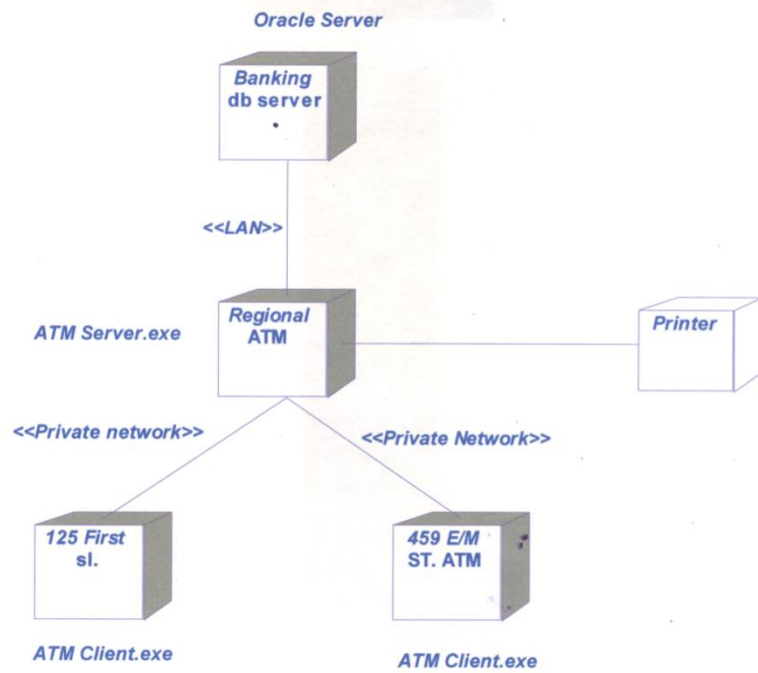
Collaboration Diagram



Component Diagram



Deployment Diagram



State Diagram

