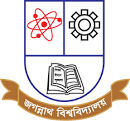
Lab 05



Course- Software Engineering Lab (CSEL-3206)

Submitted By

Submitted To

Dr. Md. Ashraf Uddin

Associate Professor

Jagannath University, Dhaka

Susmita Rani Saha

ID: B180305047

**EXPERIMENT : 1**

**AUTOMATED BANKING SYSTEM**

* **AIM :**

TO design and implement Electronic Cash Counter system through Class Diagram.

* **HARDWARE / SOFTWARE TOOLS :**

Star UML / UML Graph / Top cased.

* **PROBLEM DEFINITION :**

To develop an automated banking system, which is required to perform the

following functions :

* The customer logs into the system using card number and pin number.

The system checks for validation.

* The system queries the customer for the type of account either fixed

deposit or credit account. After getting the type of account the system

shows the balance left.

* The system queries the customer for the transaction type either

withdrawal or deposit and the required amount. The user enters the

amount and the transaction if carries out.

* **INTRODUCTION :**
* **Purpose:**
* The purpose is to describe the requirements involved in developing an Automated Banking System
* The intended audience is any person who wants:

1. To create account.
2. To withdraw or deposit either in fixed deposit or credit account.

* **Scope:**
* The product is titled Automated Banking System (ABS).
* The product will perform the following tasks:

1. Allow a new user to create an account, either fixed or credit account by entering the details and by depositing an initial amount.
2. Allow the existing user to enter his account details like card number, pin number and account type to view his balance.
3. Allow the existing user to deposit an amount by entering the amount to be deposited after the balance had been viewed.
4. Allow the existing user to withdraw an amount by entering the amount to be withdrawn after the balance had been viewed.
5. The primary benefits expected of the system are: user friendly, continuous connectivity without failure, fault tolerant and involves lesser manpower.

* **Definitions, Acronyms and Abbreviations:**

ABS: Automated Banking System.

* **References:**
* IEEE standard 830-1998 recommended practice for Software Requirements Specifications-Description.
* IEEE Software Requirements Specifications Template

http://www.cas.master.ca/~carette/SE3M04/2003/files/ srs\_template.doc

* **Overview:**

The SRS contains an analysis of the requirements necessary to help easy design.

The overall description provides interface requirements for the Banking system, product perspective, hardware interfaces, software interfaces, communication interface, memory constraints, product functions, user characteristics and other constraints.

Succeeding pages illustrate the characteristics of typical naïve users accessing the system along with legal and functional constraints enforced that affect banking system in any fashion.

* **THE OVERALL DESCRIPTION:**

**Product perspective:**

* **Hardware interfaces:**
* Hard disk: The database connectivity requires a hardware configuration that is on-line. This makes it necessary to have a fast database system (such as any RDBMS) running on high rpm hard-disk permitting complete data redundancy and backup systems to support the primary goal of reliability.
* The system must interface with the standard output device, keyboard and mouse to interact with this software.
* **Software interfaces:**
* Back End: MS Access 2007
* Front End: Microsoft Visual Basic 6.0
* **Operations:**
* The user can create a new account.
* The existing user can access his account and view his balance by entering his details.
* The user can deposit and withdraw money from his account.

**Product Functions:**

* **Creating a New Account:**

The user should provide his personal details to facilitate the bank clerk to create a new account. The user should provide:

* Customer Name.
* Customer address.
* Required account type.
* Pin Number.
* Initial deposit.
* **Operating with created account:**

The user should be able to operate with his new account after:

* Entering card number.
* Entering pin number.
* Entering the account type, transaction type and amount involved in the transaction.
* **User characteristics:**
* The intended users of this software need not have specific knowledge as to what is the internal operation of the system. Thus the end user is at a high level of abstraction that allows easier, faster operation and reduces the knowledge requirement of end user.
* The Product is absolutely user friendly, so the intended users can be the naïve users.
* The product does not expect the user to possess any technical background. Any person who knows to use the mouse and the keyboard can successfully use this product.
* **Constraints:**
* At the time of creating the new account, each user gives a pin number and is provided with a unique card number that must be used for further transactions. Hence the user is required to remember or store these numbers carefully.
* At the time of creating the new account, the initial deposit should not be less than the specified amount.
* **SPECIFIC REQUIREMENTS:**
* **Logical Database Requirements:**
* The system should contain databases that include all the necessary information for the product to function according to the requirements. These include relations such as Customer Details and Account Details.
* Customer details refer to the customer’s name and address. Account details of the customer include the card number, account type, transaction type and the pin number given by the user to be used at the time of the transaction at the bank.
* **FRONT – END DESCRIPTION:**

The front end for the Automated Banking System (ABS) is designed using Microsoft Visual Basic 6.0. The front end contains a user-friendly interface. The first form contains a welcome screen that provides an option for the user to either crate a new account or to operate through an existing account. The “create account” module contains a provision to create a new account after collecting the customer name, address and other details. The card number and pin number of the user is obtained every time there is a transaction. The user is requested to select the required type of transaction and the amount involved in the transaction.

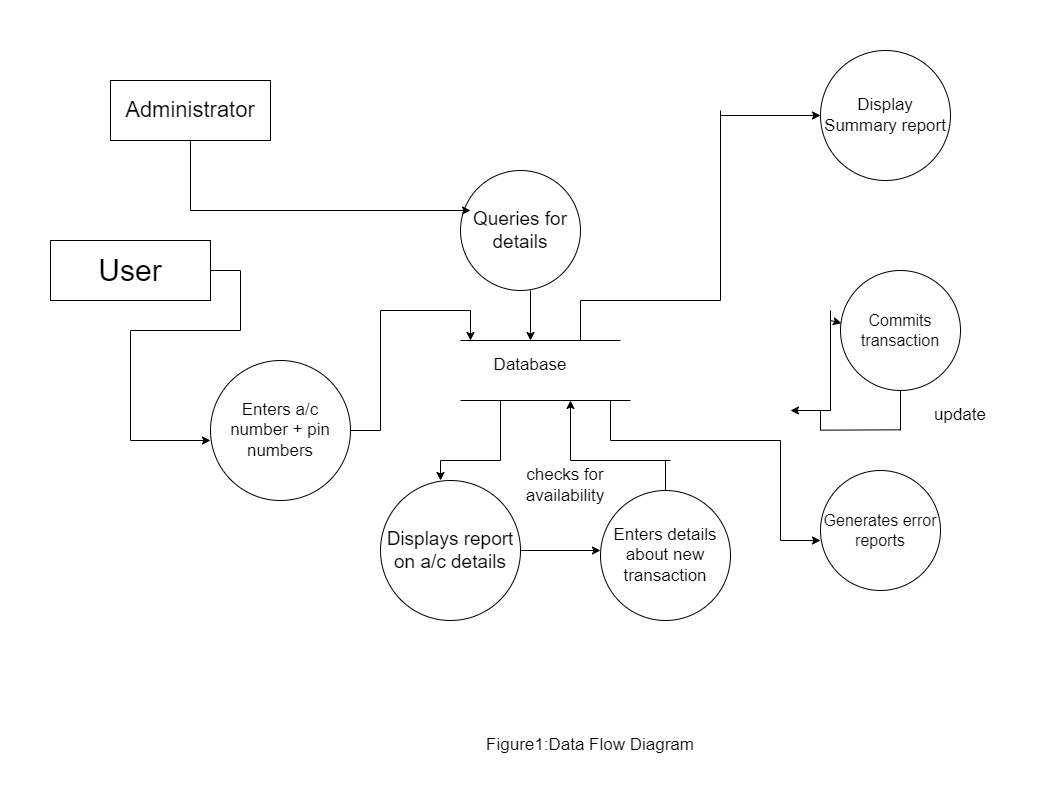
* **BACK – END DESCRIPTION:**

The Automated Banking System (ABS) database contains only one table. It correlates a unique card number, customer name, account type, pin number and the balance.

* **DATA STRUCTURES:**
* **ACCOUNT DETAILS:**

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **TYPE** | **CONSTRAINTS** |
| NAME | TEXT |  |
| ACC\_NO | AUTONUMBER | NOT NULL |
| AGE | NUMBER |  |
| GENDER | TEXT |  |
| EMAIL | TEXT |  |
| PHONE\_NO | NUMBER |  |
| PASSWORD | TEXT |  |
| BALANCE | NUMBER |  |

* **DATA FLOW DIAGRAM:**



* **PROCEDURE:**

The purpose of the class diagram is to model the static view of an application.

The class diagrams are the only diagrams which can be directly mapped with

object oriented languages and thus widely used at the time of construction. The

UML diagrams like activity diagram, sequence diagram can only give the

sequence flow of the application but class diagram is a bit different. So it is the

most popular UML diagram in the coder community. So the purpose of the class

diagram can be summarized as:

• Analysis and design of the static view of an application.

• Describe responsibilities of a system.

• Base for component and deployment diagrams.

• Forward and reverse engineering.

Class diagrams commonly contain the following things:

• Classes

• Interfaces

• Collaborations

• Dependency, generalization and association relationships

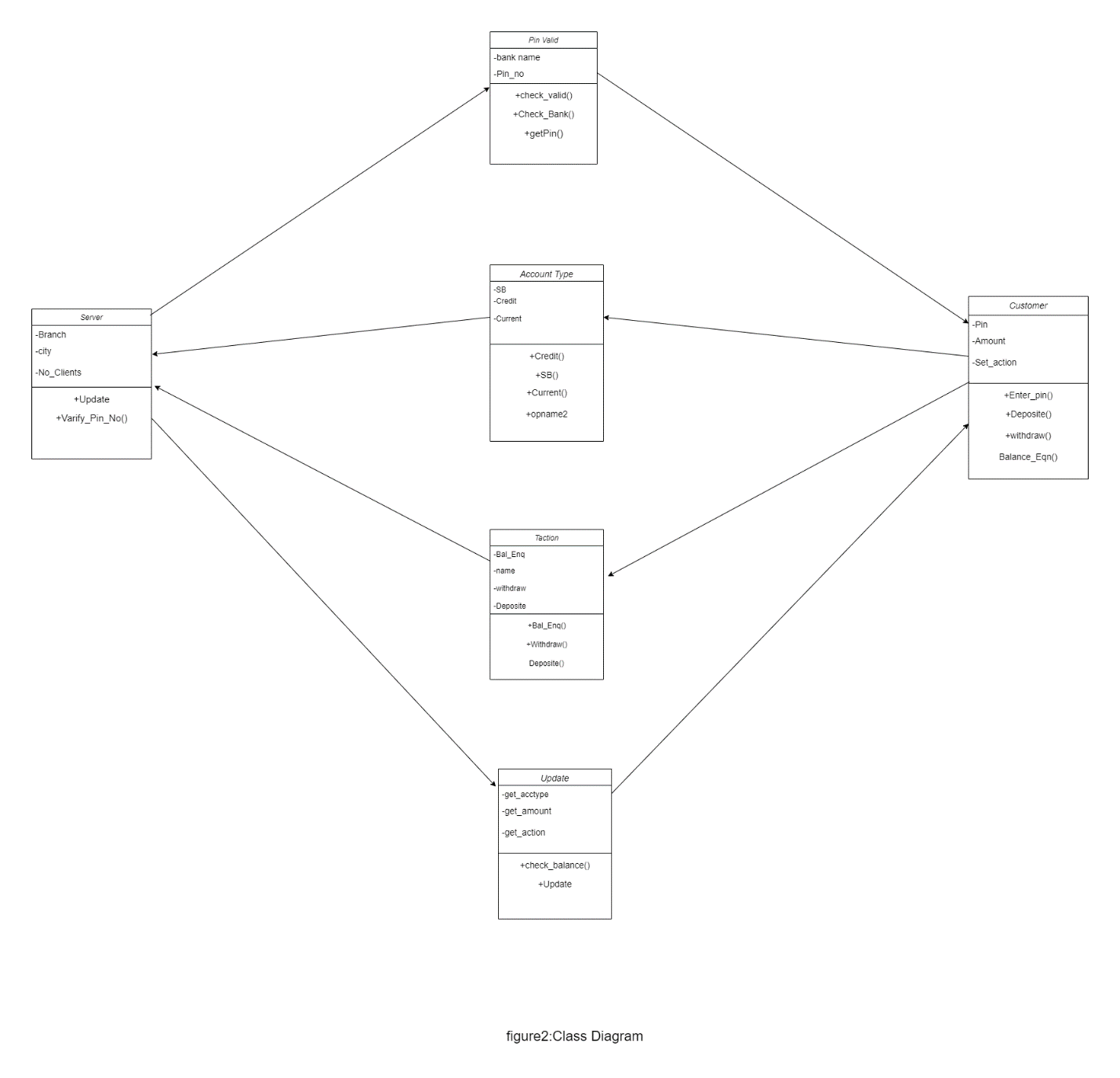
* **Steps:**

**Step1:** First Classes are created.

**Step2:** Named as Pin Valid, Account Type, Transaction, Update, Server, Customer classes are created.

**Step3:** Appropriate relationships are provided between them as association.

* **DIAGRAM:**



* **Inferences:**

1. Understand the concept of classes

2. Identify classes and attributes and operations for a class

3. Model the class diagram for the system

* **Applications:**
* Online transaction
* Online banking
* **NAME OF EXPERIMENT: Use case diagram for ATM System**
* **AIM:** To design and implement ATM System through Use Case Diagram.
* **Purpose:**

The purpose of use case diagram is to capture the dynamic aspect of a system. Because other four diagrams (activity, sequence, collaboration and State chart) are also having the same purpose. So we will look into some specific purpose which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified.

So in brief, the purposes of use case diagrams can be as follows:

• Used to gather requirements of a system.

• Used to get an outside view of a system.

• Identify external and internal factors influencing the system.

• Show the interacting among the requirements are actors.

* **Procedure:**

**Step1:** First an Actor is Created and named as User/Customer.

**Step2:** Secondly a system is created for ATM.

**Step3:** A use case Enter PIN, Withdraw money is created and connected with user as association relationship.

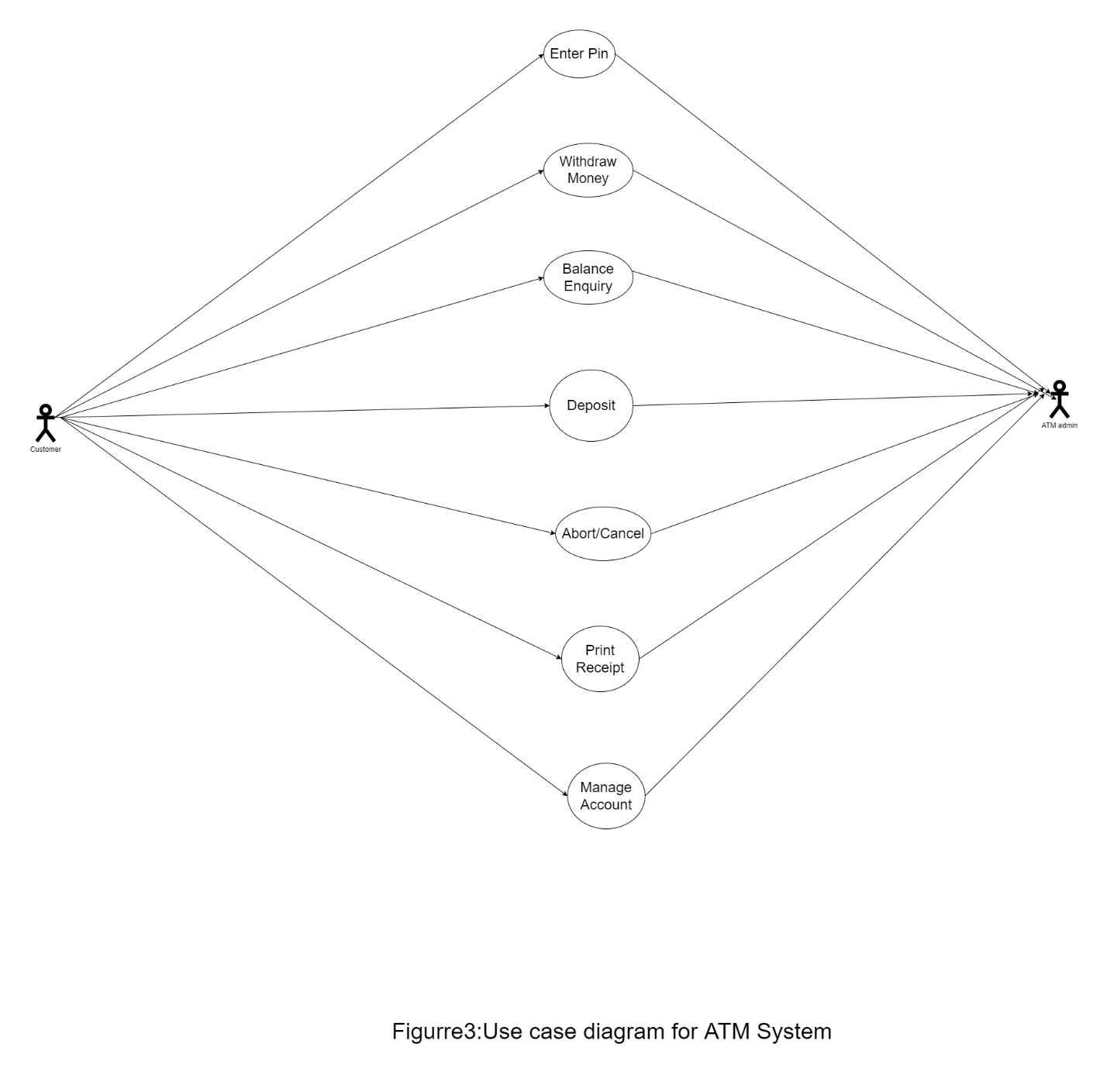
**Step4:** Similarly various use cases like Deposit money, Balance Enquiry, Manage Account etc. are created and appropriate relationships are associated with each of them.

* **Inferences:**

1. Identification of use cases.

2. Identification of actors.

* **DIAGRAM:**



* **Withdrawal UseCase:**

A withdrawal transaction asks the customer to choose a type of account to withdraw from (e.g. checking) from a menu of possible accounts, and to choose an amount from a menu of possible amounts. The system verifies that it has sufficient money on hand to satisfy the request before sending the transaction to the bank. (If not, the customer is informed and asked to enter a different amount.) If the transaction is approved by the bank, the appropriate amount of cash is dispensed by the machine before it issues a receipt. A withdrawal transaction can be cancelled by the customer pressing the Cancel key any time prior to choosing the amount.

* **INTERACTIONDIAGRAMS:**

We have two types of interaction diagrams in UML. One is sequence diagram and the other is a collaboration diagram. The sequence diagram captures the time sequence of message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.

So the following things are to be identified clearly before drawing the interaction diagram:

1. Objects taking part in the interaction.

2. Message flows among the objects.

3. The sequence in which the messages are flowing.

4. Object organization.

* **Purpose:**

1. To capture dynamic behaviour of a system.

2. To describe the message flow in the system.

3. To describe structural organization of the objects.

4. To describe interaction among objects.

* **Contents of a Sequence Diagram:**
* Objects
* Focus of control
* Messages
* Life line
* **Contents of a Collaboration Diagram:**
* Objects
* Links
* Messages
* **NAME OF EXPERIMENT: Sequence diagram for ATM System:**
* **AIM:** To design and implement ATM System through Sequence Diagram.
* **Procedure:**

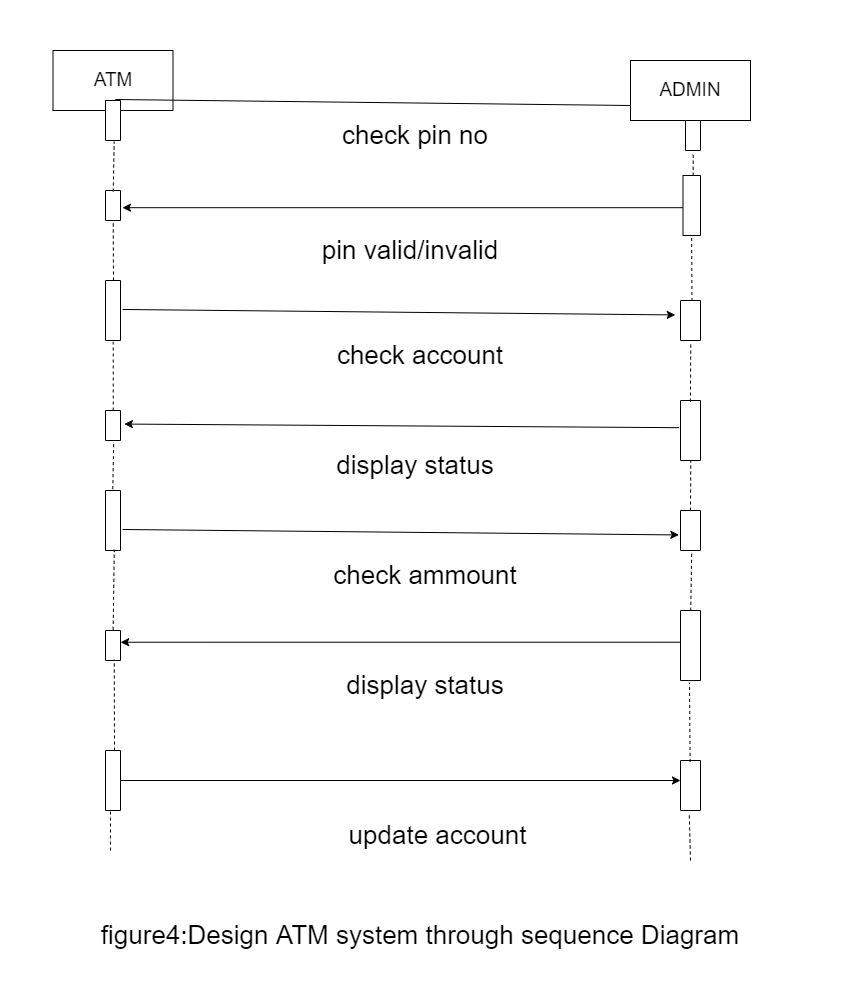
Step1: First an actor is created and named as user.

Step2: Secondly an object is created for Atm.

Step3: Timelines and lifelines are created automatically for them.

Step4: In sequence diagram interaction is done through time ordering of messages. So appropriate messages are passed between user and ATM is as shown in the figure.

* **DIAGRAM:**

****

* **NAME OF EXPERIMENT: collaboration for ATM System**
* **AIM:** To design and implement ATM System through Collaboration diagram.
* **Procedure:**

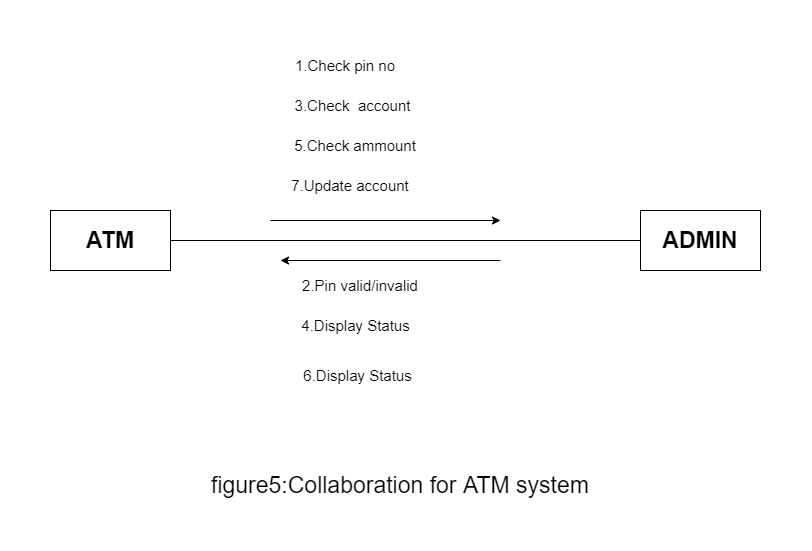
Step1: First an actor is created and named as user.

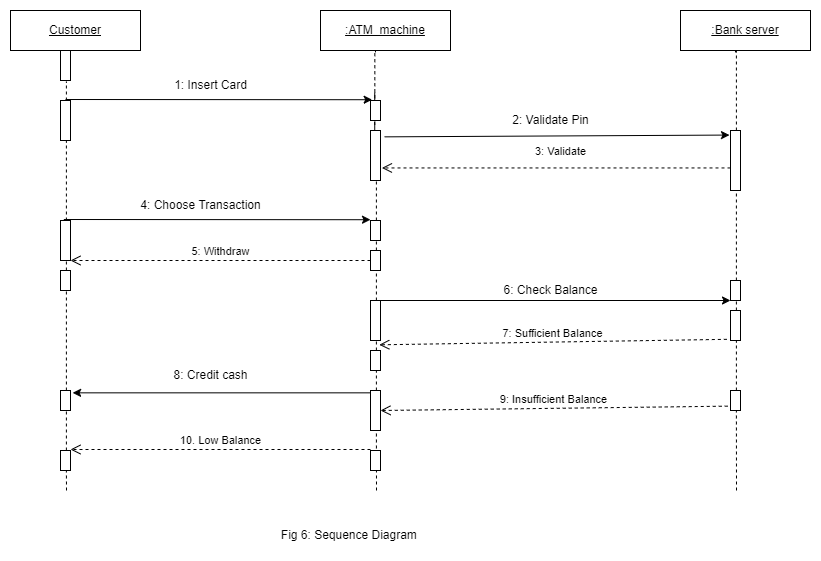
Step2: Secondly an object is created for ATM.

Step3: In collaboration diagram interaction is done through organization.

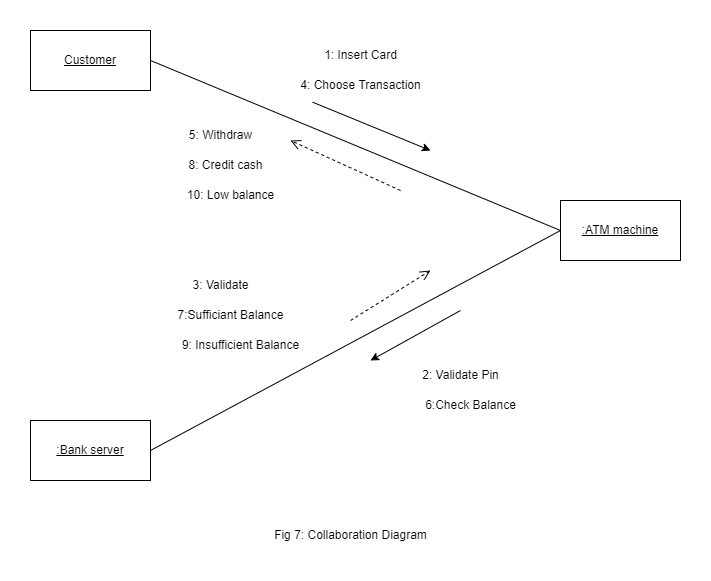
Step4: So appropriate messages are passed between user and ATM as shown in the figure.

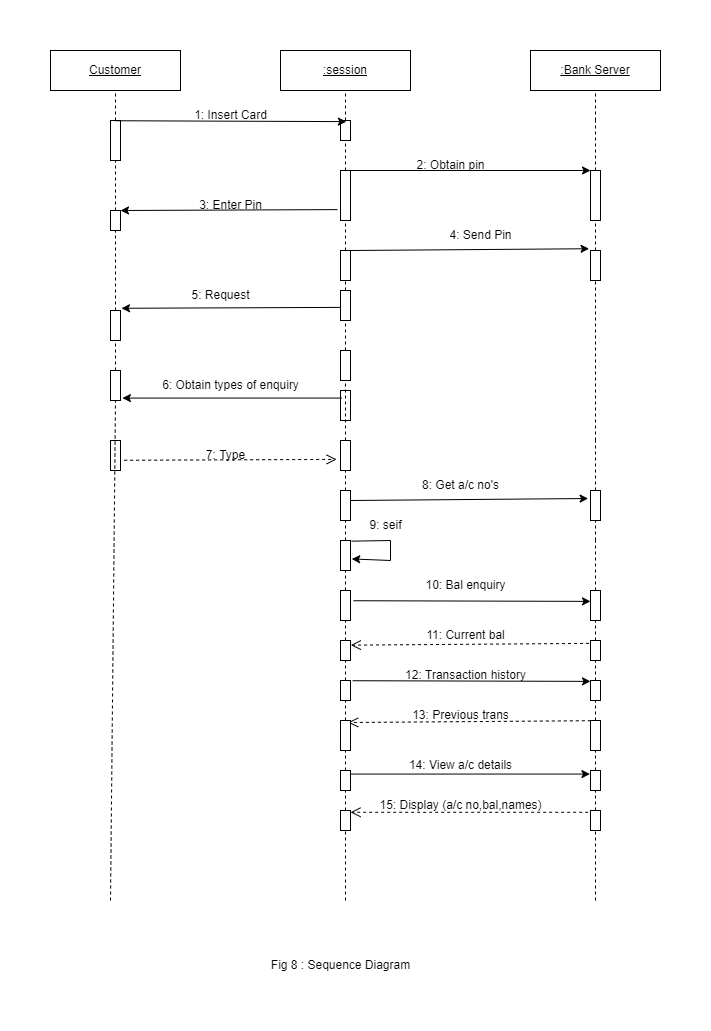
* **DIAGRAM:**

****

* **WITHDRAWL:**
* **SEQUENCE DIAGRAM:** 

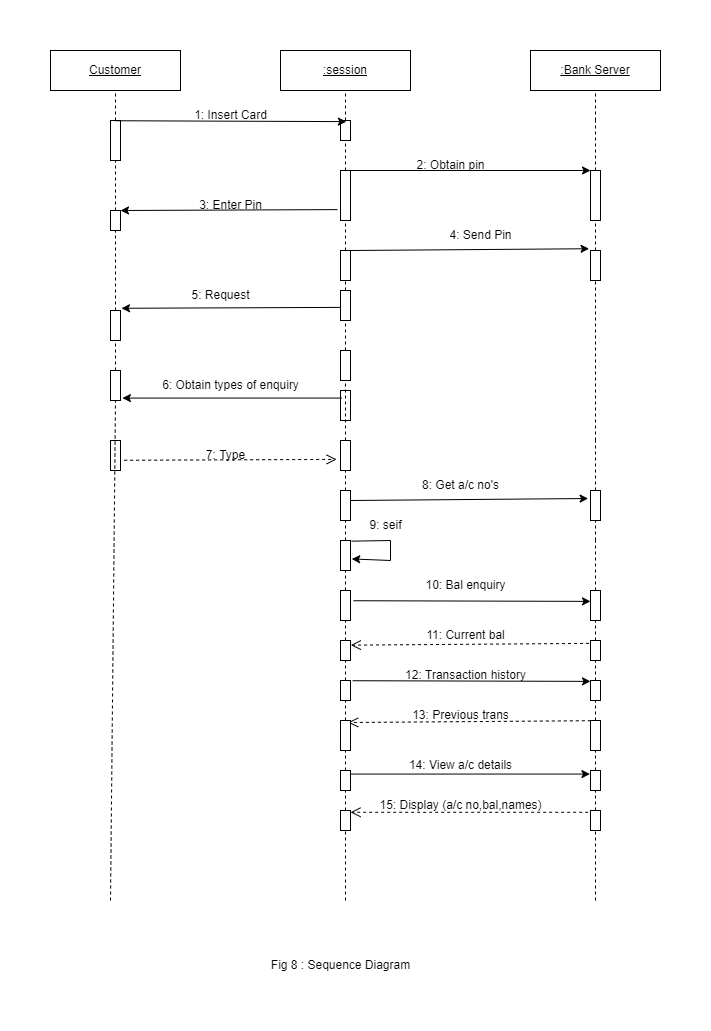
* **COLLABORATION DIAGRAM:**



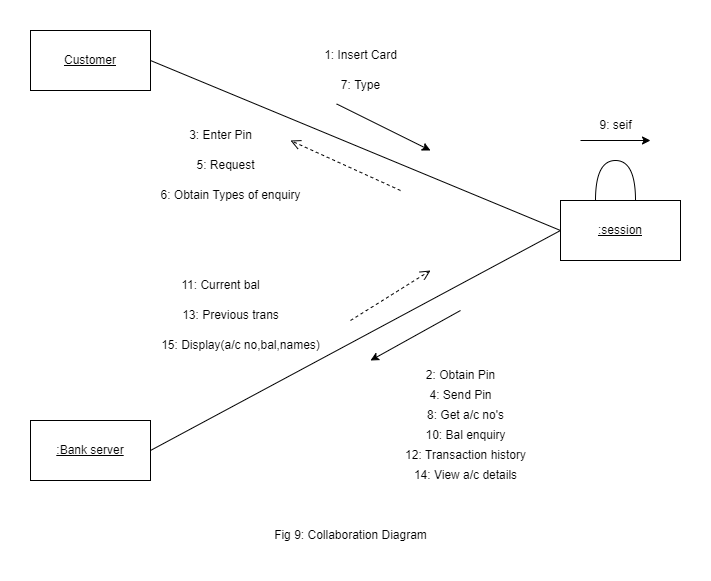


#### ENQUIRY:

* **SEQUENCE DIAGRAM:**

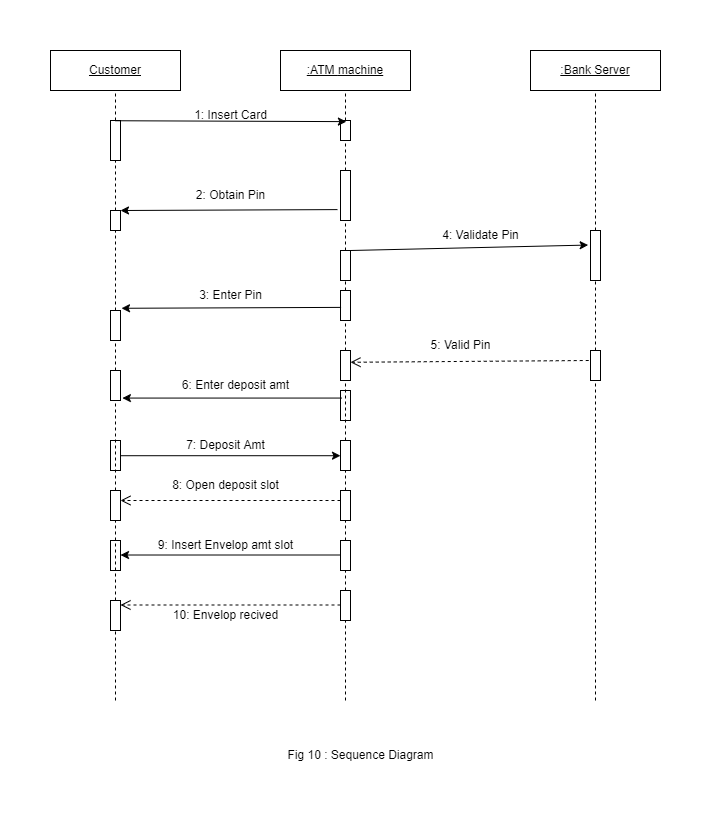


* **COLLABARATION DIAGRAM:**



##### DEPOSITUseCase:

* **SEQUENCE DIAGRAM:**



* **COLLABARATION DIAGRAM:**

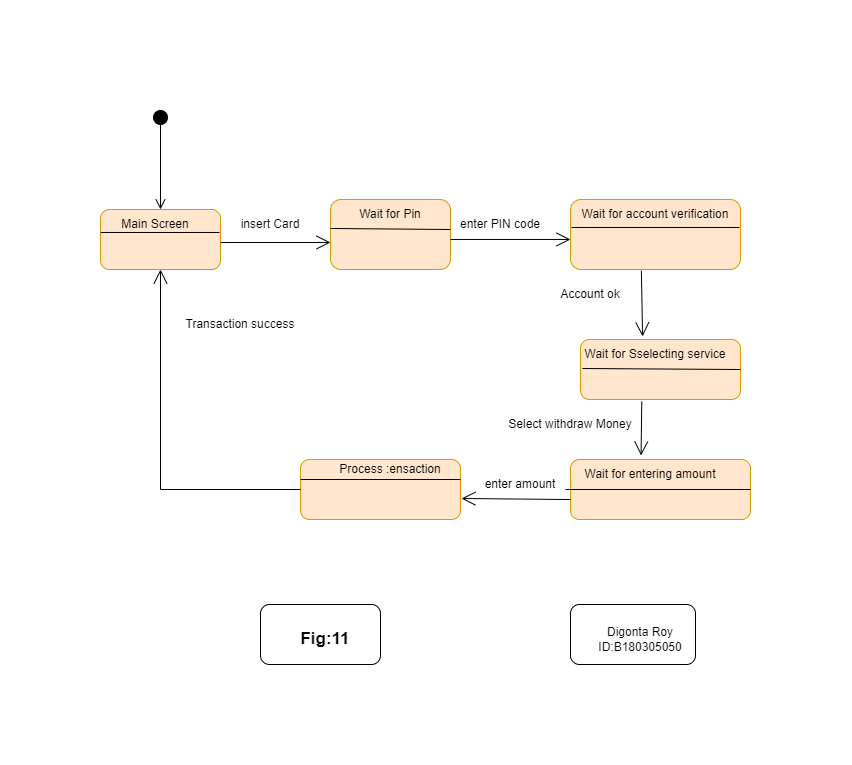


Fig11: Collaboration Diagram

##### STATECHART Diagram :

State Chart diagram 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. State chart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. But the main purpose is to model reactive system.

##### Contents

* Simply state and composite states
* Transitions, including events and actions
* **NAME OF EXPERIMENT: State chart diagram for ATM System**
* **AIM:** To design and implement ATM System through State Chart diagram.

##### Purpose:

Following are the main purposes of using State chart diagrams:

1. To model dynamic aspect of a system.
2. To model life time of a reactive system.
3. To describe different states of an object during its life time.
4. Defines a state machine to model states of an object.

##### Procedure:

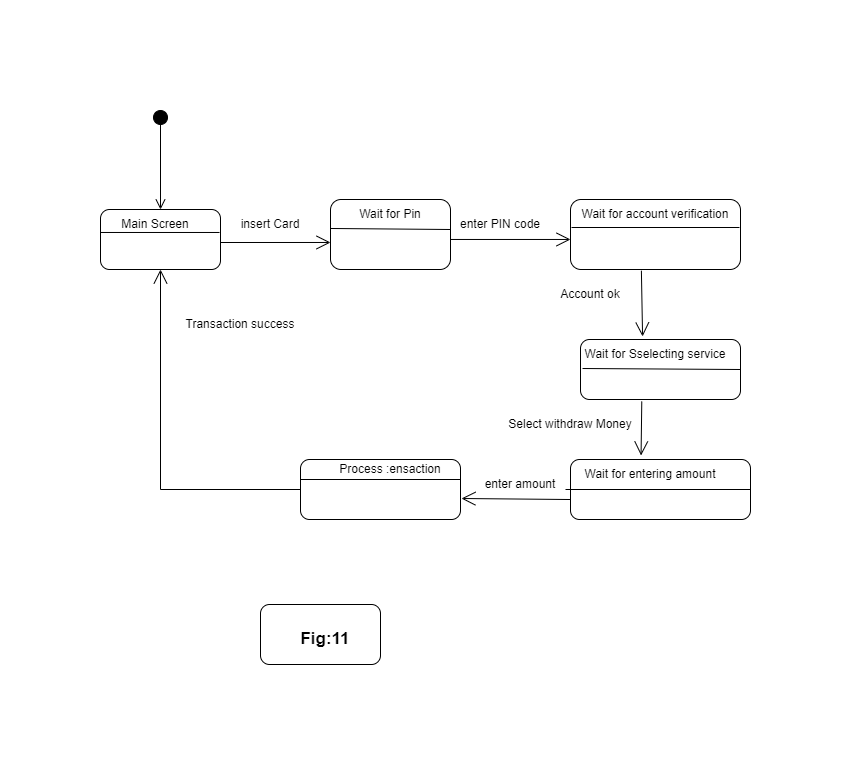
**Step1:** First after initial state control undergoes transition to ATM screen.

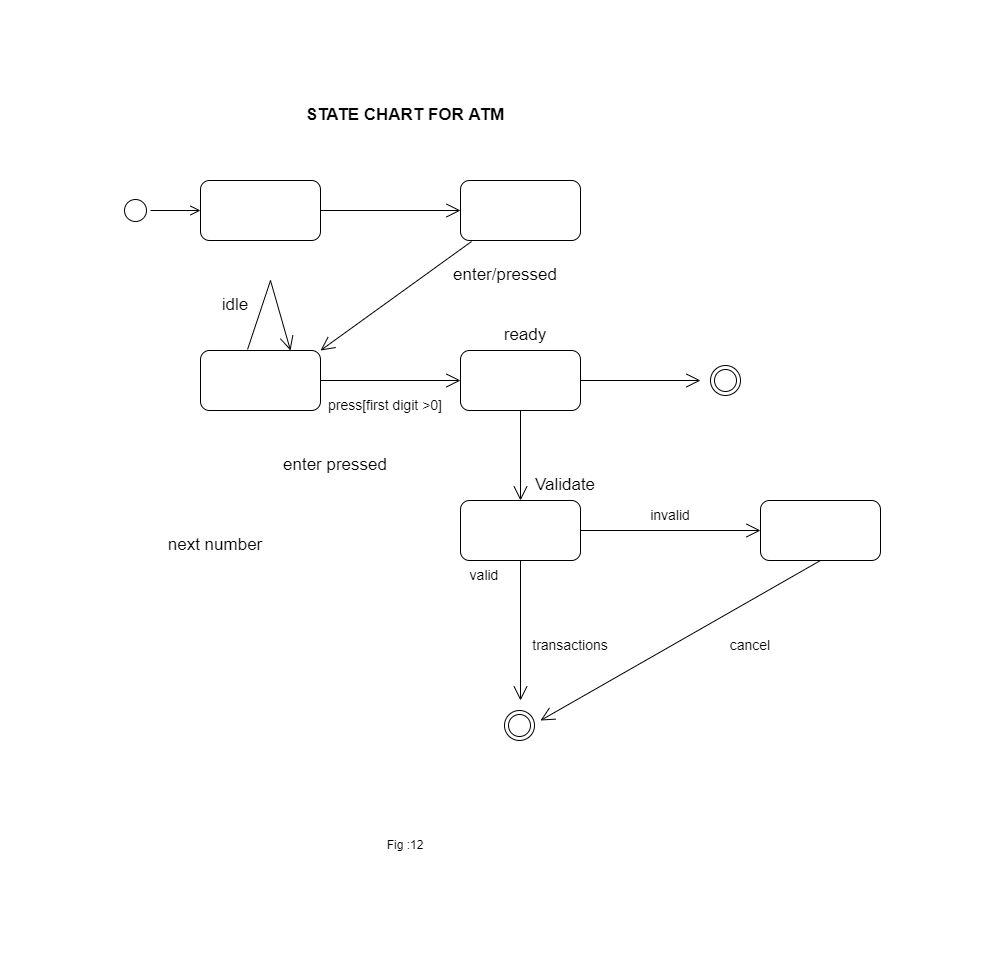
**Step2:** After inserting card it goes to the state wait for pin.

**Step3:** After entering pin it goes to the state account verification.

**Step4:**. In this way it undergoes transitions to various states and finally reaches the ATM screen state as shown in the fig.

#### DIAGRAM:

****



* **NAME OF EXPERIMENT: Activity diagram for ATM System**
* **AIM:** To design and implement ATM System through Activity Diagram**.**
* **THEORY:** An activity diagram shows the flow from activity to activity .An activity is an ongoing non atomic execution within a state machine .Activities ultimately results in some action, which is made up of executable atomic computations. We can use these diagrams to model the dynamic aspects of a system.

Activity diagram is basically a flow chart to represent the flow form one activity to another . The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow by using elements like fork, join etc.

##### Contents

Initial/Final State, Activity, Fork & Join, Branch, Swim lanes

##### Fork

A fork represents the splitting of a single flow of control into two or more concurrent Flow of control. A fork may have one incoming transition and two or more outgoing transitions, each of which represents an independent flow of control. Below fork the activities associated with each of these path continues in parallel.

##### Join

A join represents the synchronization of two or more concurrent flows of control. A join may have two or more incoming transition and one outgoing transition. Above the join the activities associated with each of these paths continues in parallel.

##### Branching

A branch specifies alternate paths takes based on some Boolean expression Branch is represented by diamond Branch may have one incoming transition and two or more outgoing one on each outgoing transition, you place a Boolean expression shouldn’t overlap but they should cover all possibilities.

##### Swim lane:

Swim lanes are useful when we model workflows of business processes to partition the activity states on an activity diagram into groups. Each group representing the business organization responsible for those activities, these groups are called Swim lanes .

##### Procedure:

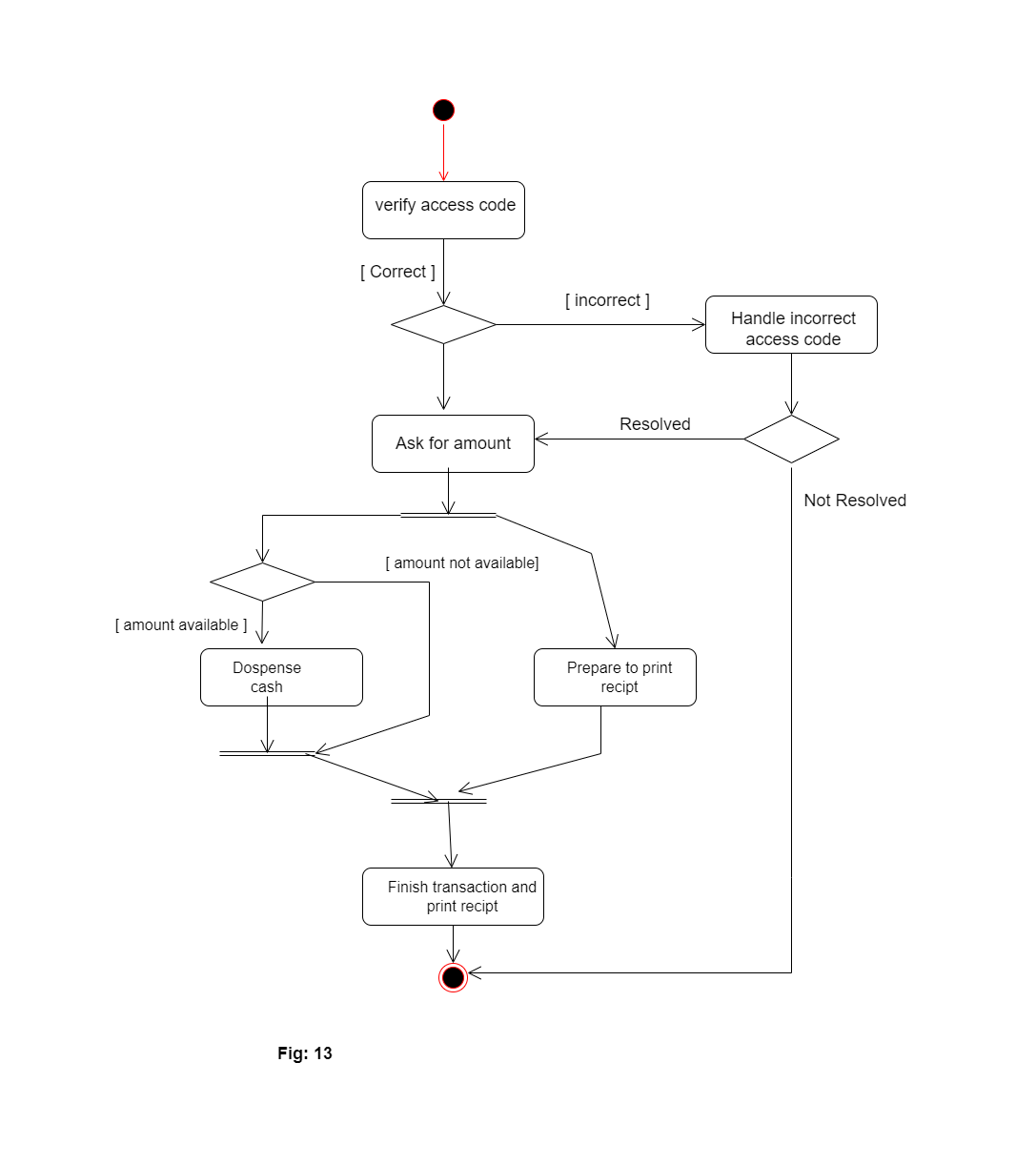
**Step1:** First initial state is created.

**Step2:** After that it goes to the action state insert card.

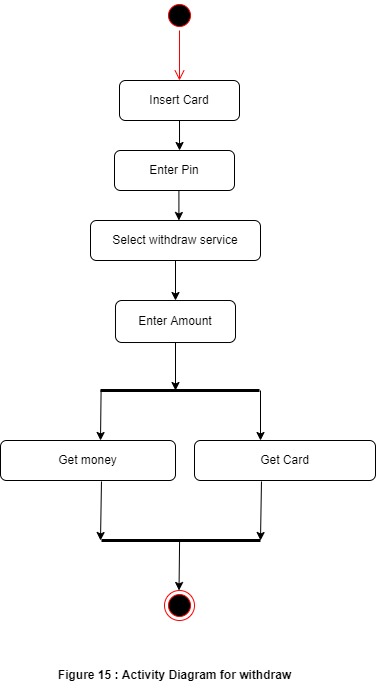
**Step3:** Next it undergoes transition to the state enter pin **Step4**: In this way it undergoes transitions to the various states. **Step5:** Use forking and joining wherever necessary.

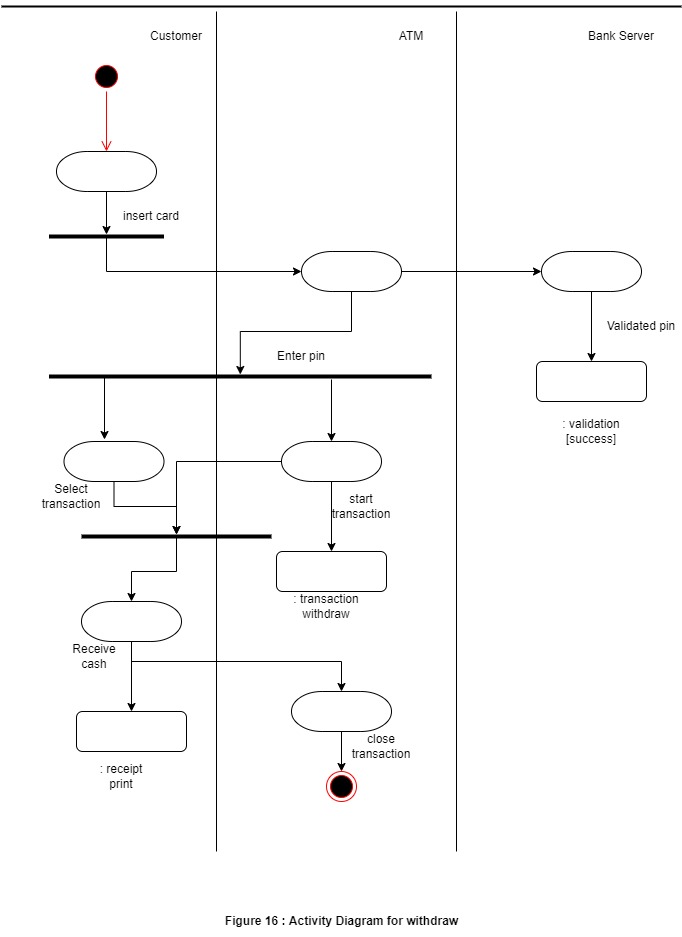
#### DIAGRAM:

##### Activity diagram for Transactions:

****

##### Activity diagram for Withdraw

****

****

##### Inferences:

1. Identify the action states of the objects .
2. Understand the transitions and events for various objects.

* **NAME OF EXPERIMENT: Component diagram for ATM System.**

**AIM:** To design and implement Component diagram for ATM System**. THEORY:**

Component diagrams are used to model physical aspects of a system. Physical aspects are the elements like executable, libraries, files, documents etc. which resides in a node. So component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

##### Purpose:

Component diagrams can be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

Before drawing a component diagram the following artifacts are to be identified clearly:

* + - Files used in the system.
    - Libraries and other artifacts relevant to the application.
    - Relationships among the artifacts.
    - Now after identifying the artifacts the following points needs to be followed:
    - Use a meaningful name to identify the component for which the diagram is to be drawn.
    - Prepare a mental layout before producing using tools.
    - Use notes for clarifying important points.

##### Contents

Components, Interfaces, Relationships

##### Procedure:-

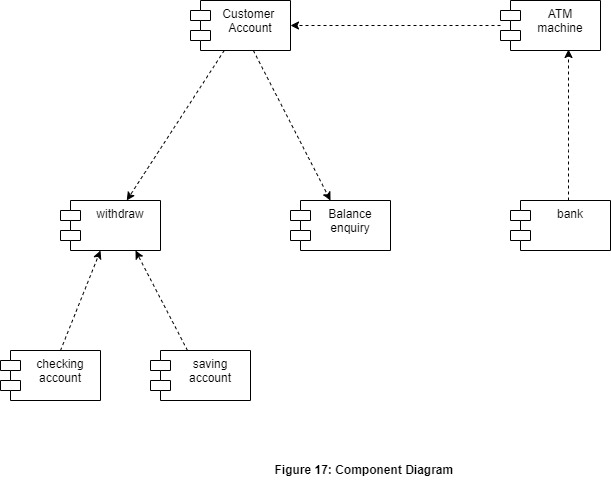
**Step1:** First user component is created.

**Step2:** ATM system package is created.

**Step3:** In it various components such as withdraw money, deposit money, check balance, transfer money etc. are created.

**Step4:** Association relationship is established between user and other components.

#### COMPONENT DIAGRAM:



* **NAME OF EXPERIMENT: Deployment diagram for ATM System.**

**AIM:** To design and implement ATM System through Deployment diagram.

##### Purpose:

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.

**Contents:** Nodes, Dependency & Association relationships

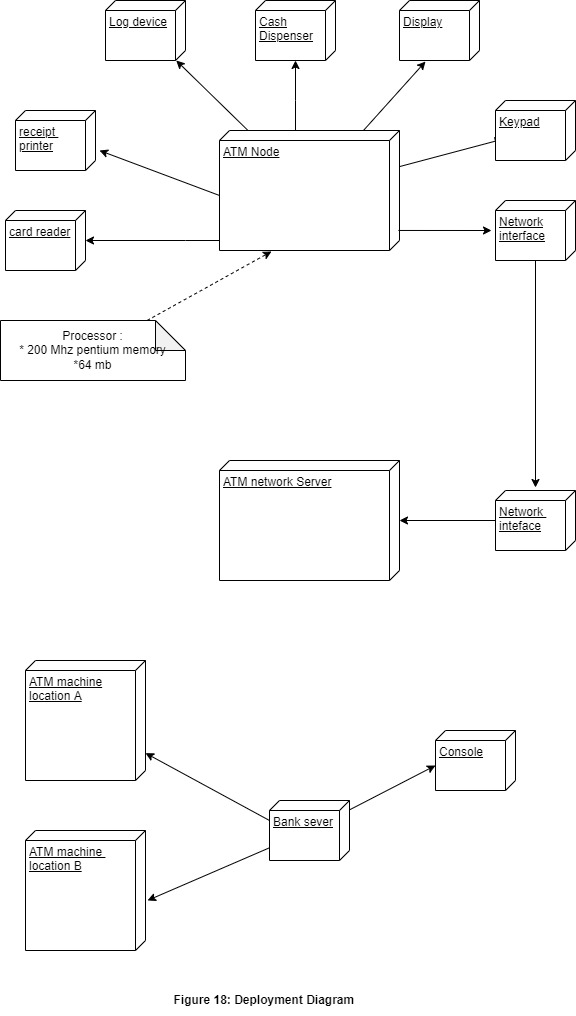
##### Procedure:-

**Step1:** First user node is created

**Step2:** various nodes withdraw money, deposit money, and check balance, transfer money etc. are created.

**Step4:** Association relationship is established between user and other nodes.

**Step5:** Dependency is established between deposit money and check balance.



* **NAME OF EXPERIMENT: Test Design for ATM** **AIM:** To design and Write the test cases for ATM **PURPOSE:**

Test Design is creating a set of inputs for given software that will provide a set of expected outputs. The idea is to ensure that the system is working good enough and it can be released with as few Problems as possible for the average user

## Test Desgin for ATM Machine

* 1. Machine is accepting ATM card
  2. Machine is rejecting expired card
  3. Successful entry of PIN number
  4. Unsuccessful operation due to enter wrong PIN number 3 times
  5. Successful selection of language
  6. Successful selection of account type
  7. Unsuccessful operation due to invalid account type
  8. Successful selection of amount to be withdraw
  9. Successful withdrawal.
  10. Expected message due to amount is greater than day limit
  11. Unsuccessful withdraw operation due to lack of money in ATM
  12. Expected message

Due to amount to withdraw is greater than possible balance.

* 1. Unsuccessful withdraw operation due to click cancel after insert card

##### Write the Test Cases for ATM

**Test Cases For Atm Machine**

1. Successful inspection of ATM card
2. Un successful operation due to insert card in wrong angle
3. Un successful operation due to invalid account Ex:other bank card or time expired Card
4. Successful entry of PIN number
5. un successful operation due to enter wrong PIN number 3times
6. Successful selection of language
7. Successful selection of account type
8. un successful operation due to invalid account type
9. Successful selection of withdrawl operation
10. Successful selection of amount to be withdrawl
11. Successful withdrawl operation
12. Unsuccessful withdrawl operation due to wrong denominations
13. Unsuccessful withdrawl operation due to amount is greater than day limit
14. Unsuccessful withdrawl operation due to lack of money in ATM
15. Unsuccessful withdrawl operation due to amount is greater than possible balance
16. Unsuccessful withdrawl operation due to transactions is greater than day limit
17. Unsuccessful withdrawl operation due to click cancel after insert card
18. Unsuccessful withdrawl operation due to click cancel after insert card & pin number
19. Unsuccessful withdrawl operation due to click cancel after insert card , pin number
    1. **RESULT**

Thus the requirements involved in developing an Automated Banking System was completed successfully.

**COURSE MANAGEMENT SYSTEM [CMS]**

**1. Problem Definition :**

The Course Management System is an application that permits students to compare, tamper and present student’s data in an important manner. The project develops student's information system for the Intranet Automation of student information management system Software, if the institution does not have a Course Information Management System, they face many problems specially about wasting time. It is hard to find data and to get any comment from students or teachers, the system has been to ease good Interaction/ communication facilities between the Students and administration. In doing so, the project helps a lot in doing faster duties in Institutions, there are some new ideas like Bulletin board system, Students comment, analysis skill student etc. We need some methods to complete that system using detailed Unified Modelling Language (UML Diagram), Dataflow Diagrams and Entity Relationship Diagrams (E-R Diagrams). The web pages about students are created dynamically based on the student's user ID, user password and links.

* Construct the design element for a course ware management system that can be used to manage courses and classes for an organisation that specialize in providing training.
* The organisation offers a variety of courses in a variety of areas such as learning management techniques and understanding different software languages and technologies.
* Each course is made up of set of topics.
* Tutors in the organisation are assigned courses to teach according to the area that is specialized in and their availability.
* The organisation publish and maintain a calendar of different courses and assign tutors every year.
* There is a group of Course Administrator in the organisation to manage the courses including course content, assign courses to tutor and define the course schedule.
* The training organisation aim to use the Course-ware Management System to get a better control and visibility to the management of courses as also to streamline the process of generating and managing the schedule of the different courses.

**ADVANTAGES OF PROPOSED SYSTEM**

* + It is very fast and clever.
  + No need more user admin, always extra manual effort.
  + Not easy for data loss.
  + Approaching students to learn the requirements of the institution.
  + Short time to knowledge and learning to use operate the system.
  + It needs a less number of used hardware device.
  + Recently about the relationship between teacher and student lesson topic.
  + Need short time to find any student information.

### 2 . Requirement Specification

**OUT LINE OF THE CMS**

Introduction Requirements System Use Cases

Test Plan and Test Cases Flow Chart

System Sequence Diagram System Architecture Database Tables Technologies to be Used References

Course Management System (CMS) is a web-based course application used by students and professors. The objective of this product is to provide interaction between students and teachers.

This product provides Online discussions, Online quizzes, Automatic group allocation Statistical analysis of students performance others

**Requirement of the Students :**

**Students must be able to:**

Login to the system

Sign up for access to a course

Download files within/related to their courses Taking quizzes of multiple choice questions

Access a Project Management Room (PMR) that will be only accessible to group members

**Upon logging into this room they can:**

view what other group members are currently logged into the room as well as email addresses

access a real-time chat room to communicate online with other group members

post and get files

post and get messages

log out of the project meeting room Logout of the whole system

## Requirements for Instructors

**Instructors must have all the abilities as the student additionally, they can:**

Create a course for their students to access Delete a course

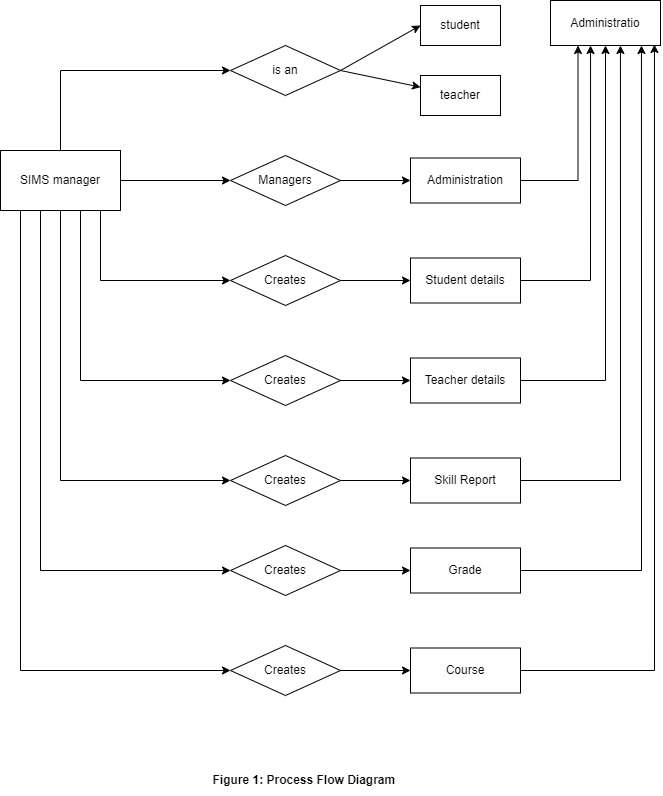
Create an interactive quiz that their students can take by using a simple interface where they provide the questions, choices, and correct answers.

Create a PMR that can will be accessible only by specified group members

Use Cases describe the sequence of events an actor, using a system, performs to complete a process. This necessitates identifying the actors as well as the processes they perform. The following table illustrates the actors identified along with the processes / use cases and sequence diagram.

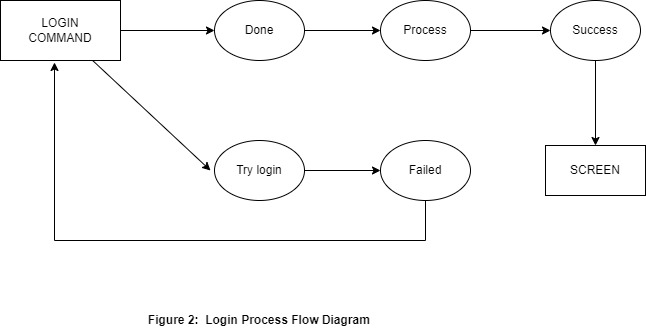
## **Design & Modelling**

**PROCESS FLOW DIAGRAM**



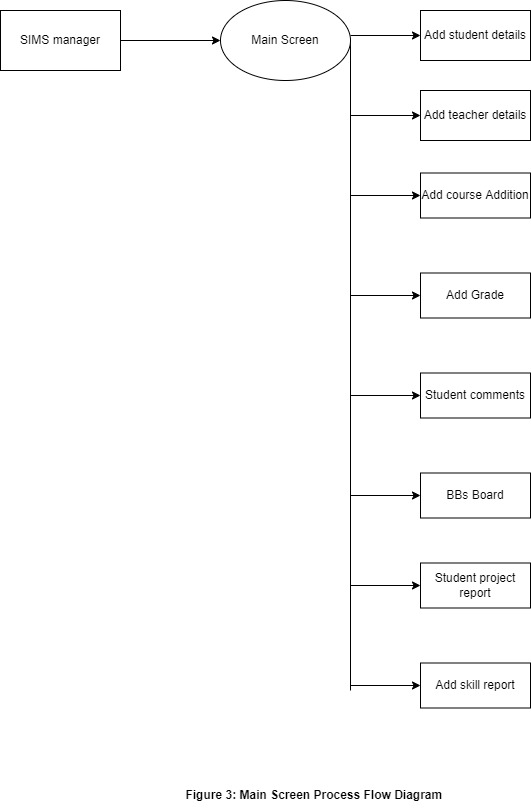
**Description:** In the above system the main task is to identify a criminal face. So the operator’s show as Administrator will create some of part in the SIMS system

**LOGIN PROCESS**

******

**Description:** The inputting process is allowing you to enter user ID and passwords to make user allow the software available to display system. After user finished the input details, the process will change to the next stage to check the main screen in the system. It displays a screen if matching occurs an otherwise error message if they are not matched the process asked you to login again back to the first stage.

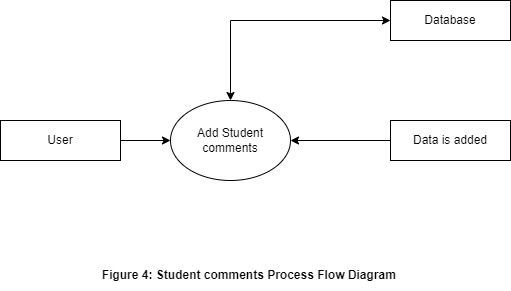
**MAIN SCREEN PROCESS**

****

**Description:** This process will explain any part of the SIMS system to administrator. The selection

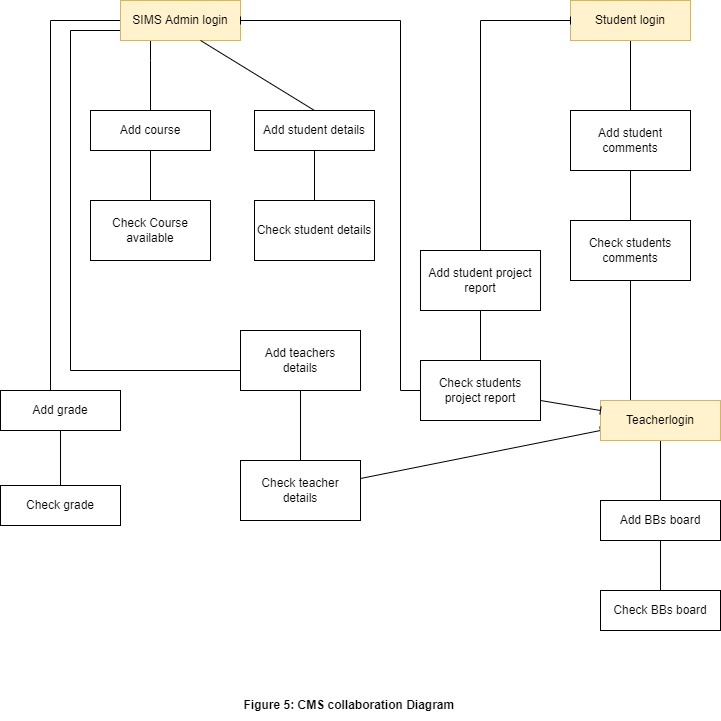
of the screen depends on the administrator and he can use any model he wants or select any part in our system. The different screens that are available are Add Student and teacher details, Add skill Report, Add student project report, check student comments, Add grade, Add course addition, Add BBs board.

**STUDENT COMMENTS**

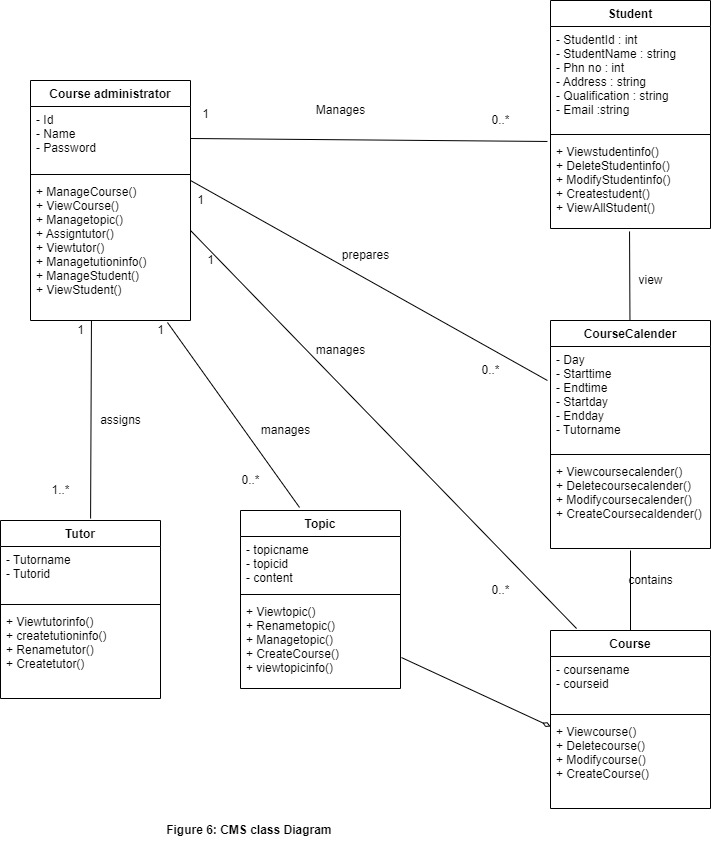
****

**Description:** The process allowed Students are also asked to add any comments or suggestions to solve any homework or questions for any subject to make a great team always related with science to get new ideas, to create a good actor in any education place. In the SIMS system we make easier students homework system to return lost time fast connect between each student's and make bigger mind students to more discussion any subject. The student can enter SIMS into using student comments so easy to use.

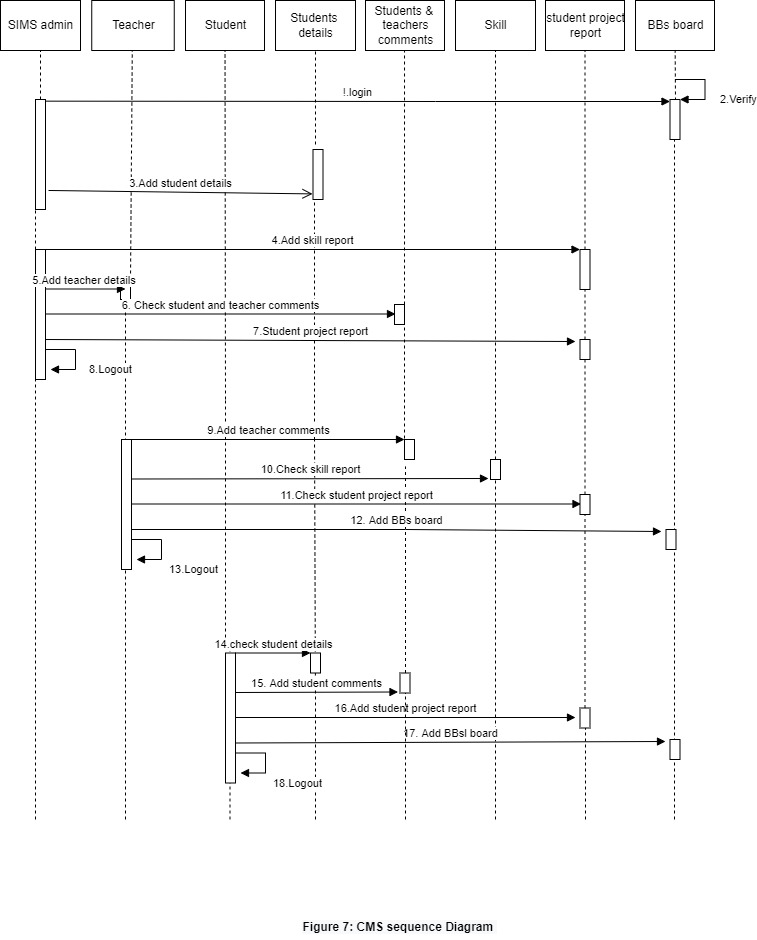
#### STATE CHART DIAGRAMS COLLABORATION DIAGRAM



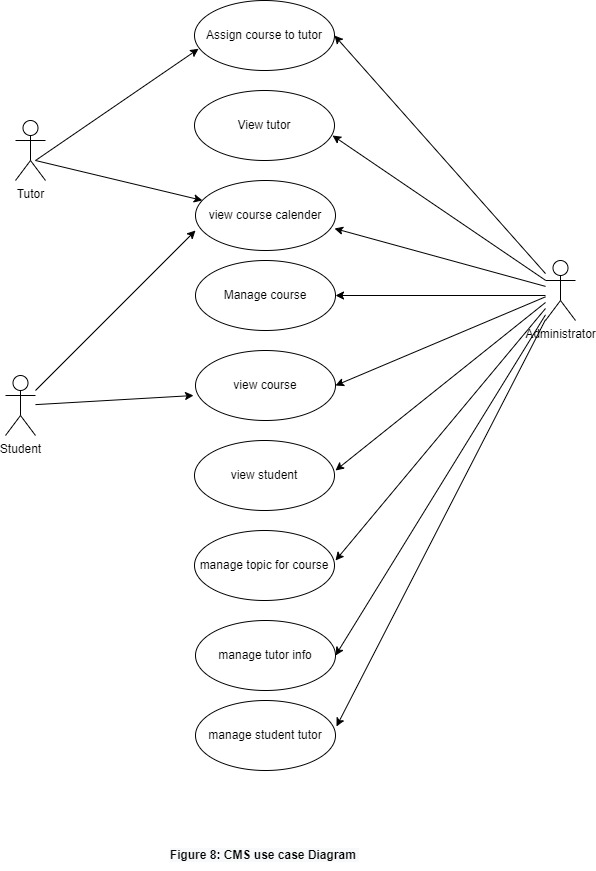
##### Class Diagram of CMS :

****

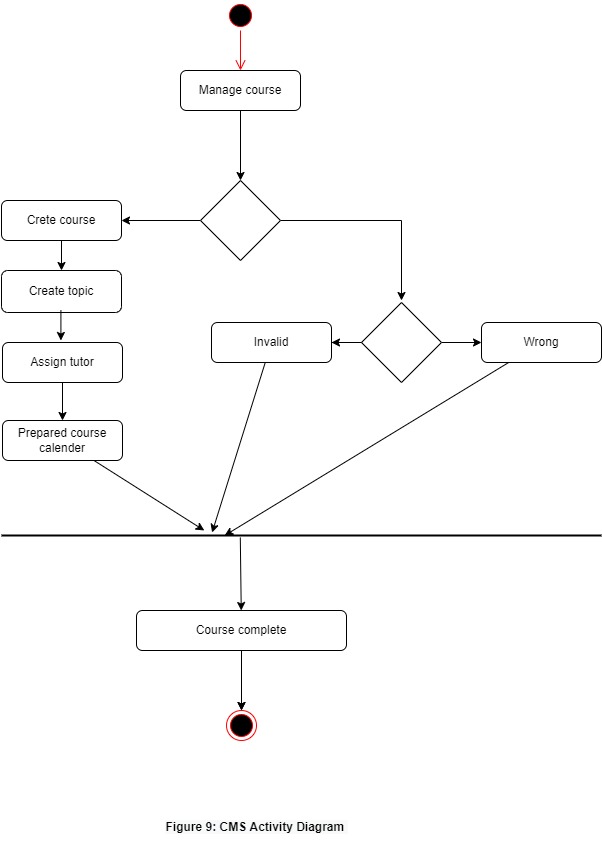
**SEQUENCE DIAGRAM**

****

### Use Case Diagram:-

****

### Activity Diagram

****

#### TESTING

1. **TESTING PHASE:** We permitted the completion of the system only after it has been thoroughly tested; the test is to deal with all requests without any denial of service application. At this stage in the implementation of real-time without any crashes, and therefore can refuse the package even at this stage. Making sure inquiries sent to the introduction of the expected results of the database.
2. **SYSTEM TESTING:** Testing is a helps guarantee to input data in our system and get results, work as new data are added. The proposed system tests in equivalent software that consists of its own phases of analysis, implementation, testing and maintenance. Make sure our design and functions are correct or not.
3. **UNIT TESTING:** Each module of the system was tested separately to uncover errors within its boundaries. The user can use interface a guide in the process.
4. **MODULE TESTING:** A module is composed of various programs related to that module. Module testing is done to check the module functionality and interaction between units within a module. It checks the functionality of each program with relation to other programs within the same module. It then tests the overall functionality of each module.

**5 . IMPLEMENTATION PHASE:** The implementation phase in our work in last and important phase. When a production system done or installed, first need to user training, make sure our model is correct for that system, ease to use for user, get our point to create our system, user documentation is delivered, then this phase is completed. Users can test the developed system when changes are made according to the needs. The testing phase involves the testing of the developed system using various kinds of data. Testing of data is prepared and system is tested using the test data. The system needed to be plugged into the network, then it could be accessed from anywhere, after a user logins into the system. The tasks that had to be done to implement the system is to create the database tables in the organization database domain. The administrator was granted his role so that the system could be accessed. Deliverables needs some steps like (User Training, Distributed User Documentation, Finalized System Documentation, Installed Production System, Post- Implementation Review Summary, and Methodology Compliance Form)

**6. CONCLUSION :** The design tools for Course Management System [CMS], and the

development, implementation, and database management system for online educational Institutions. The system also focus to make a good relationship between teacher and student to make as in the education system, The system provides an effective environment contribute to the follow-up basis of the performance of the students during the academic year and helps create an environment in constant communication among students, parents, teachers and school administration / university. The waste of computer resources, workers, and time associated with the existing manual system.

This project will provide the course management information system serves as a useful approach Database dialog box for the update function, advanced search options to the authorized person and continuing link between each student and discuss any questions and exchanged the ideas of students, it serves as a useful approach for users. It reduces the time it takes a user to add, update, delete, view and search for information.

**END**