GREATER NOIDA INSTITUTE OF TECHNOLOGY

GREATER NOIDA (COLLEGE CODE: 272)

Affiliated to Guru Gobind Singh Indraprastha University, New Delhi

Software Engineering Lab

Subject Code: ETCS-353

Submitted By: - **Submitted To: -**

Name: -

Roll no: - **(Asst. Prof, Deptt of CSE)**

Branch:-

LIST OF EXPERIMENTS

S. No.	Name of Experiments	Date of Experiment	Date of submission	Sign.
1.	To prepare a problem statement for Bank Management System			
2.	Do required analysis and develop SRS for Bank Management System			
3.	To draw a sample ER diagram for Bank Management System			
4.	Use Case Diagram of Bank Application Management.			
5.	Class and object diagram for Bank Application System.			
6.	Activity Diagram for ATM.			
7.	Sequence Diagram for Bank Application System.			
8.	Draw an activity diagram for bank application.			
9.	To draw a Sample DFD for Bank Management System			

EXPERIMENT NO. 1

AIM: To prepare problem statement for Bank Management System.

REQUIREMENTS:

Hardware Interfaces

- Pentium(R) 4 CPU 2.26 GHz, 1gb RAM
- Screen resolution of at least 800 x 600 required for proper and complete viewing of screens. Higher resolution would not be a problem.
- Secondary Memory: 500MB

Software Interfaces

- Any window-based operating system (Windows 95/98/2000/XP/NT)
- WordPad or Microsoft Word

Bank Management System

This system provides the basic services to manage bank accounts at a bank. Bank has many branches, each of which has an address and branch number. A client opens accounts at a branch. Each account is uniquely identified by an account number; it has a balance and a credit or overdraft limit. There are many types of accounts, including: a mortgage account, a chequing account, and a credit card account. It is also possible to have a joint account.

Information objective: bank management system would be able to maintain information and able to keep records of that particular event.

Although the basic type of services offered by a bank depends upon the type of bank and the country, services provided usually include: Taking deposits from their customers and issuing current or checking accounts and savings accounts to individuals and business. Extending loans to individuals and business, Cashing cheque. Facilitating money transactions such as wire transfer and cashiers cheque, Consumer & commercial financial advisory services, financial transaction can be performed through many different channels.

EXPERIMENT NO. 2

OVERALL DESCRIPTION

PRODUCT PRESPECTIVE

To develop a system that will overlook the activities going transaction the particular bank without manual processing. All transaction will be updated automatically by using the information stored in record.

SOFTWARE REQUIREMENT

Front end:

JAVA, HTML.

Back end:

MySQL.

Operating System:

Windows 7/8/8.1/10

HARDWARE REQUIREMENT

- Android version 2.3 ginger bread (minimum, android user's)
- 2GB ram
- 1.2 GHz processor
- Intel i5

FUNCTIONAL REQUIREMENT

Overview of Bank Management System

There are various job positions within bank institutions. These positions are as follows: -

1) Customer Service Position
2) Mid Management Position
3) Executive Level Position

Customer Service Position consists of a bank teller, bank marketing representative.
Mid Management Position consists of internal auditor, data processing officer etc.
Executive Level Position consists of loan officer, branch manager etc.

Bank Teller

Bank tellers are the first line of advertising for a bank.

Tellers must be friendly and have an attitude of service.

Tellers are responsible for basic account transactions such as servicing savings and checking accounts and providing account details to customers.

Bank Marketing Representative

A bank marketing representative is an individual who markets banking products to customers.

Products range from checking accounts and savings accounts to CD and special deposit accounts.

The bank marketing representative is also responsible for understanding the strategic plan for marketing various products and implementing plans for the bank.

Internal Auditor

Audit is extremely important in banking environments.

The internal audit position is considered a midlevel management position that ensures the bank is following compliance and regulatory laws concerning bank operations and procedures.

The internal auditor is the individual responsible for creating audit programs for all areas of operations.

Positions under the scope of internal auditor are Financial Auditor and Information Systems Auditor.

Information Systems Auditor monitors data processing, data security and disaster recovery strategies for the bank.

Branch Manager

Branch managers are assigned by midlevel or executive management to drive-through bank operation.

Branch managers create branch work schedules for personnel, provide a budget to executive management concerning resources and ensure that bank policies are followed at the branch level.

A branch manager can work from the main bank and have several branch banks assigned under her supervision.

Provides training, coaching, development and motivation

Loan Officer

Meets with applicants to obtain information for loan applications and to answer questions about the process.

Analyze applicants' financial status, credit, and property evaluations to determine feasibility of granting loans.

Explain to customers the different types of loans and credit options that are available, as well as the terms of those services.

Obtain and compile copies of loan applicants' credit histories, corporate financial statements, and other financial information.

Review and update credit and loan files.

Data Processing Officer

The data processing officer (DPO) is responsible for the operation, maintenance and security of the bank information systems and offline terminals or devices not attached to the system.

At the end of a daily processing cycle, the DPO ensures the general ledger account is balanced.

The officer also ensures that daily transaction exception reports are created for managerial review.

The DPO is responsible for creating security profiles for employees.

Scope of the project:

This project can be implemented in any bank by fulfilling basic requirements.

Conclusion: This Bank Management System will provide the transaction going inside the bank without manual processing. All information will be updated automatically by using the information stored in the system files.

SRS should address the following

The basic issues that the SRS shall address are the following:

a) Functionality. What is the software supposed to do?

b) External interfaces. How does the software interact with people, the system‘s hardware, other hardware, and other software?

c) Performance. What is the speed, availability, response time, recovery time of various software functions, etc.?

d) Attributes. What are the portability, correctness, maintainability, security, etc. considerations?

e) Design constraints imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.?

Characteristics of a good SRS

An SRS should be

a) Correct
b) Unambiguous
c) Complete
d) Consistent
e) Verifiable
f) Modifiable
g) Traceable

Correct - This is like motherhood and apple pie. Of course you want the specification to be correct. No one writes a specification that they know is incorrect. We like to say - "Correct and Ever Correcting." The discipline is keeping the specification up to date when you find things that are not correct.

Unambiguous - An SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation. Again, easier said than done. Spending time on this area prior to releasing the SRS can be a waste of time. But as you find ambiguities - fix them.

Complete - A simple judge of this is that is should be all that is needed by the software designers to create the software.

Consistent - The SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another.

Verifiable - Don't put in requirements like - "It should provide the user a fast response." Another of my favourites is - "The system should never crash." Instead, provide a quantitative requirement like: "Every key stroke should provide a user response within 100 milliseconds."

Modifiable - Having the same requirement in more than one place may not be wrong - but tends to make the document not maintainable.

Traceable - Often, this is not important in a non-politicized environment. However, in most organizations, it is sometimes useful to connect the requirements in the SRS to a higher level document. Why do we need this requirement?

EXPERIMENT NO. 3

Objective: - To draw a sample ER diagram for Bank Management System

REQUIREMENTS:-

Hardware Requirements:

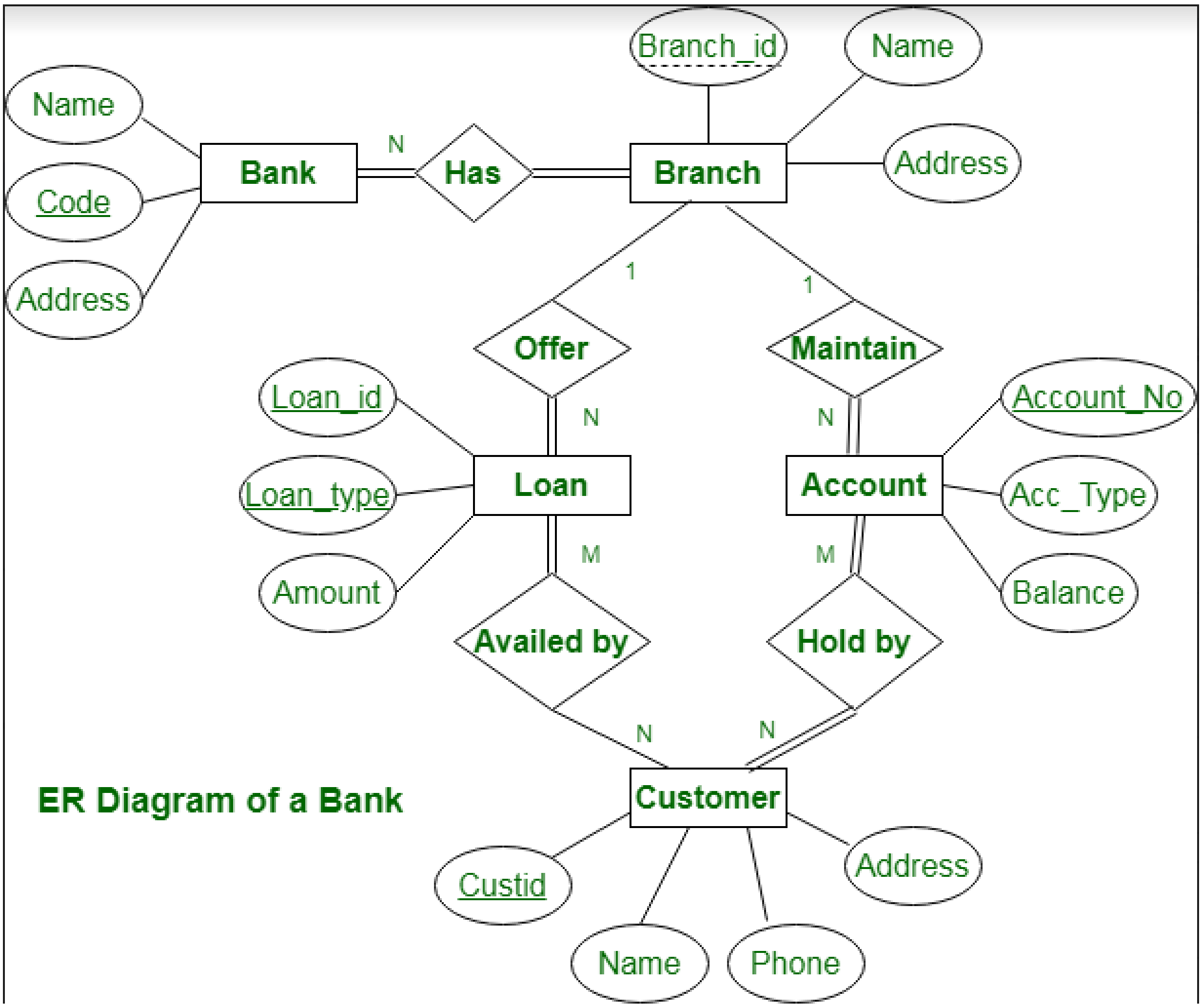
Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, coloured monitor.

Software Requirements:

Rational Rose, Windows XP,

THEORY

Entity Relationship Diagrams are a major data modelling tool and will help organize the data in


ER Diagram of a Bank

your project into entities and define the relationships between the entities. This process has proved to enable the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner

Conclusion: The entity relationship diagram For Bank Management System has been drawn successfully.

EXPERIMENT NO. 4

Objective: - Draw an use case diagram for bank application.

Hardware Requirements:

Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.

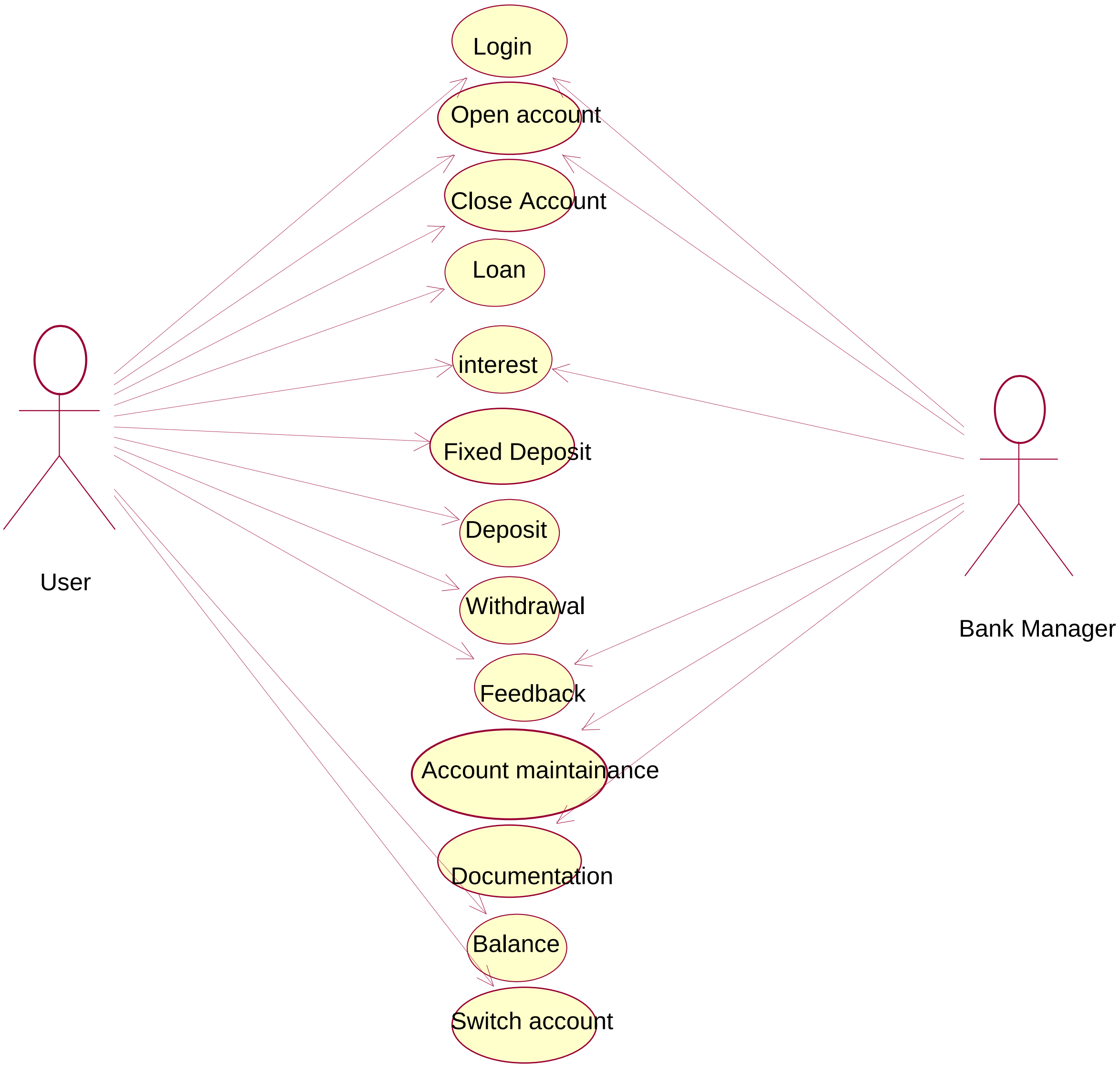
Software Requirements:

Rational Rose, Windows XP,

THEORY

According to the UML specification a use case diagram is “a diagram that shows the relationships among actors and use cases within a system.” Use case diagrams are often used to:

- Provide an overview of all or part of the usage requirements for a system or organization in the form of an essential model or a business model
- Communicate the scope of a development project
- Model your analysis of your usage requirements in the form of a system use case model



Conclusion: The use Case diagram For Bank Management System has been drawn successfully.

EXPERIMENT NO. 5

Objective: -Draw the class diagram and object model for banking Application System.

REQUIREMENTS:

Hardware Requirements:

- Pentium(R) 4 CPU 2.26 GHz, 128 MB RAM
- Screen resolution of at least 800 x 600 required for proper and complete viewing of screens. Higher resolution would not be a problem.
- CD ROM Driver

Software Requirements:

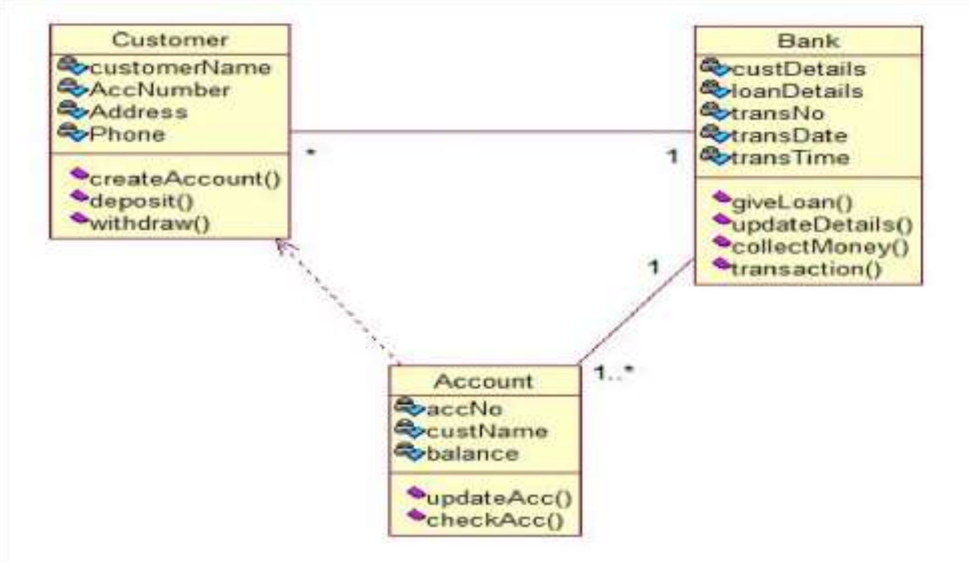
- Any window-based operating system(Windows98/2000/XP/NT)
- IBM Rational Rose Software

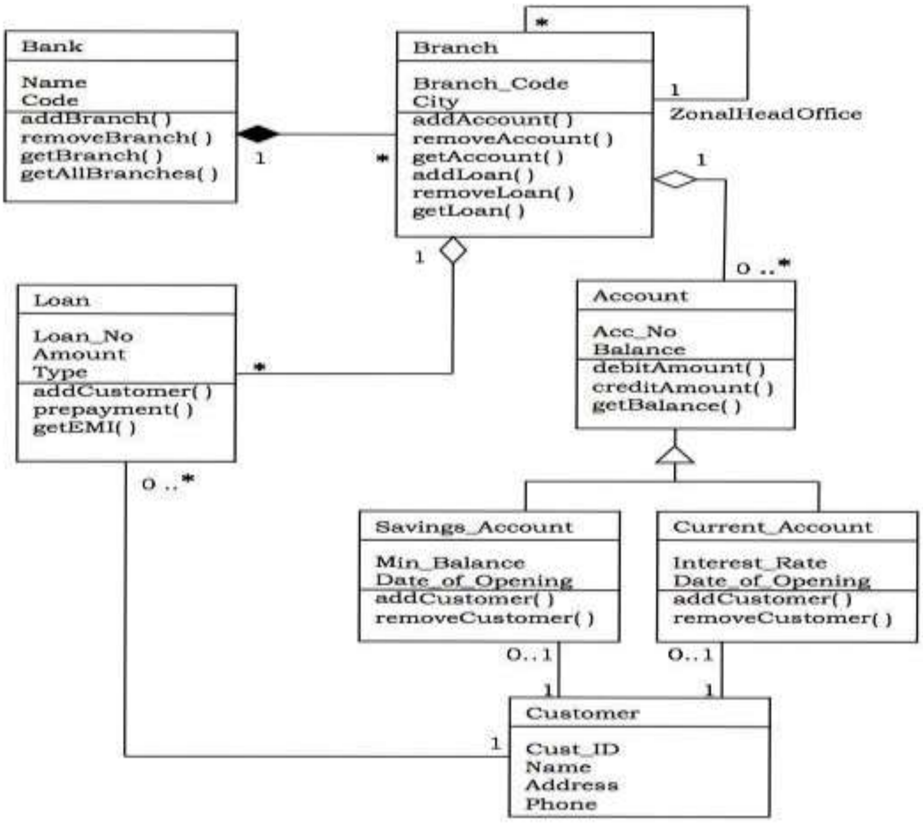
THEORY

A **class diagram** 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.

Class diagrams show the classes of the system, their inter-relationships, and the operations and attributes of the classes. Class diagrams are typically used, although not all at once, to:

- Explore domain concepts in the form of a domain model
- Analyze requirements in the form of a conceptual/analysis model
- Depict the detailed design of object-oriented or object-based software





Conclusion: The Class diagram For Bank Management System has been drawn successfully.

PROGRAM NO. 6

Objective: -Draw the activity diagram for ATM, discuss three activities for the same.

Hardware Requirements:

Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.

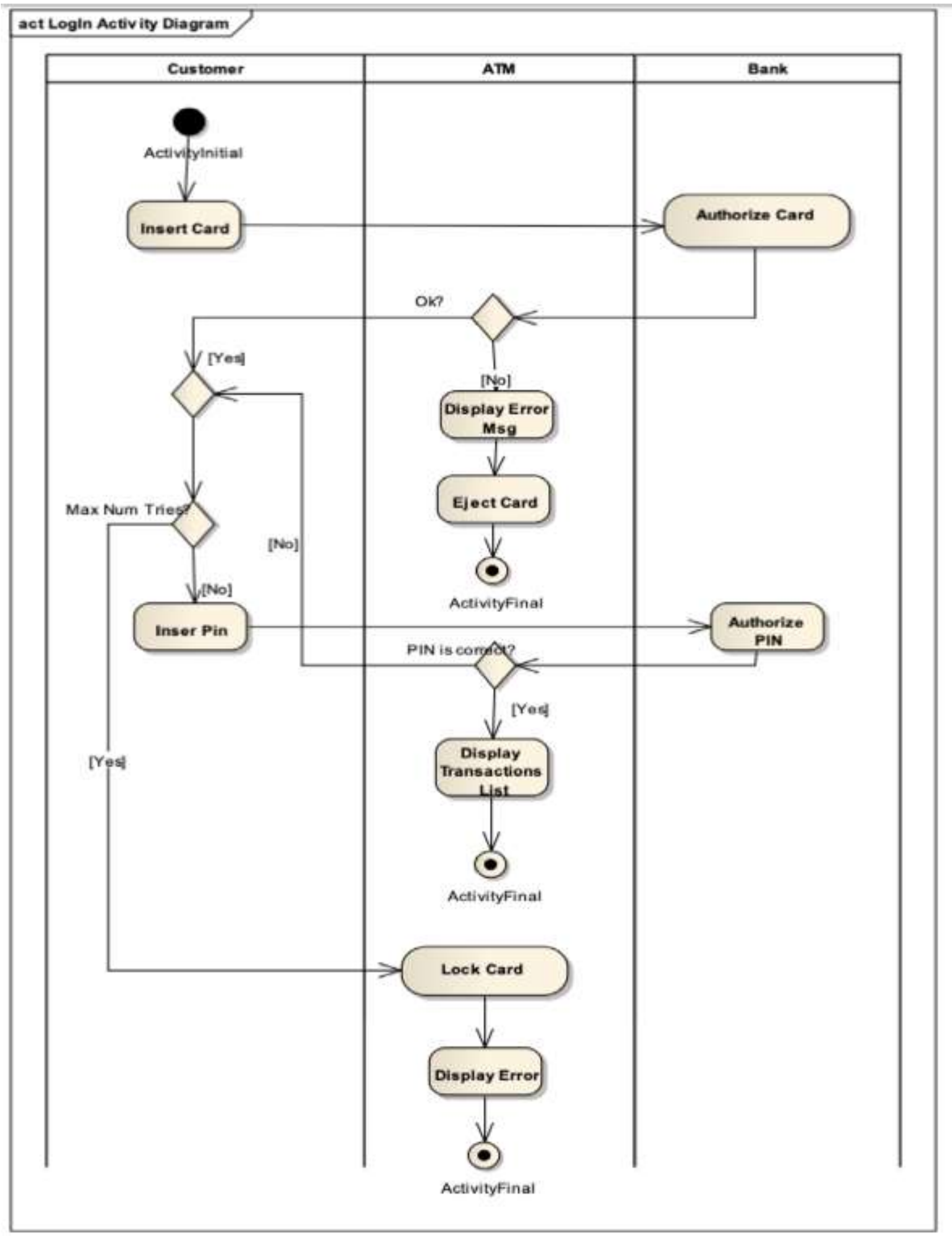
Software Requirements:

Rational Rose, Windows XP,

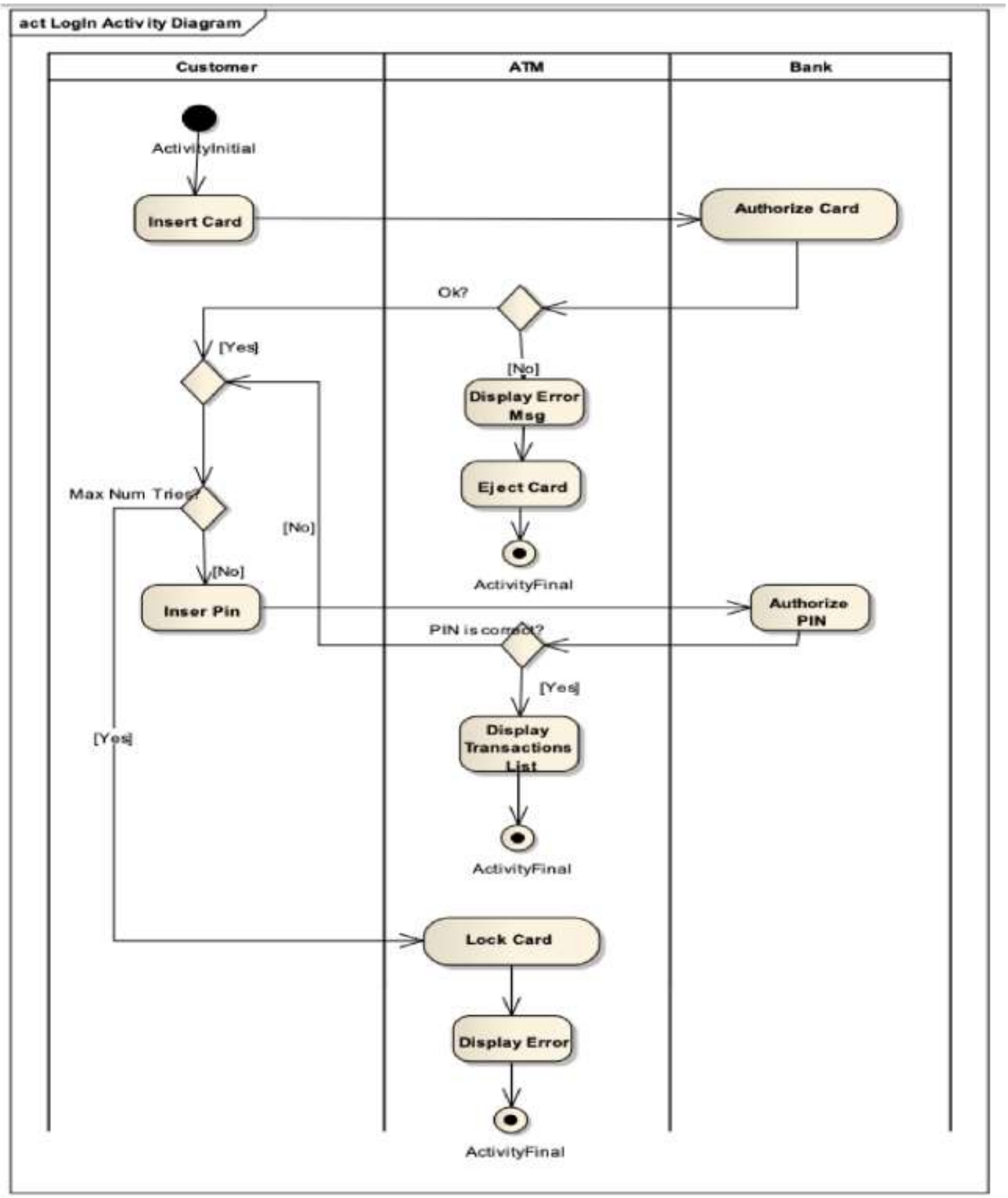
THEORY

UML 2 activity diagrams are typically used for business process modeling, for modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. Although UML activity diagrams could potentially model the internal logic of a complex operation it would be far better to simply rewrite the operation so that it is simple enough that you don't require an activity diagram. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structured development.

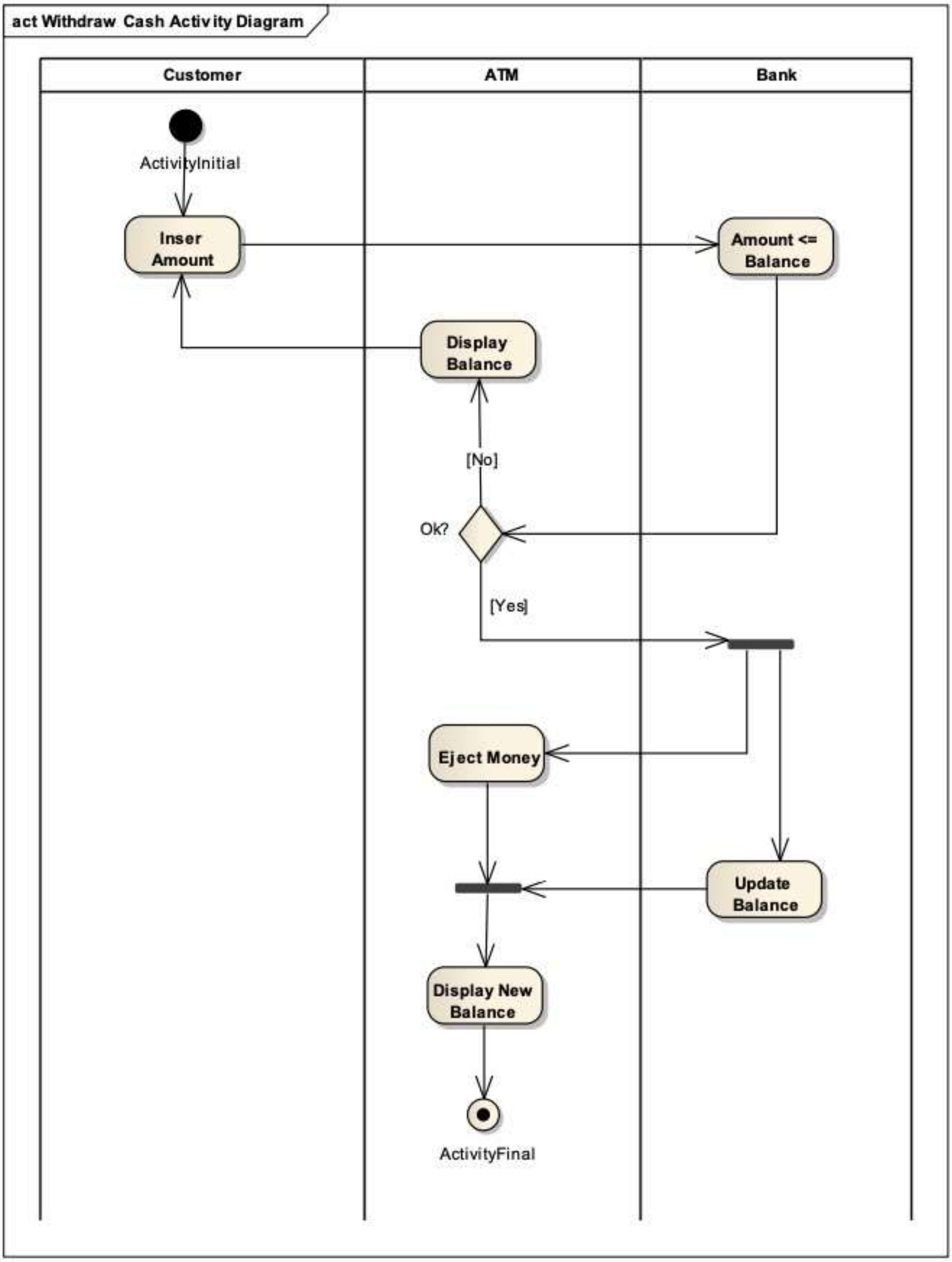
Activity1:


act Login Activ ity Diagram
Customer
ATM
Bank
ActivityInitial
Insert Card
Authorize Card
Ok?
[Yes]
[No]
Display Error Msg
Eject Card
ActivityFinal
Max Num Tries?
[No]
[No]
Inser Pin
Authorize PIN
PIN is correct?
[Yes]
Display Transactions List
[Yes]
ActivityFinal
Lock Card
Display Error
ActivityFinal

Activity 2:


act Login Activ ity Diagram
Customer
ATM
Bank
ActivityInitial
Insert Card
Authorize Card
Ok?
[Yes]
[No]
Display Error Msg
Eject Card
ActivityFinal
Max Num Tries?
[No]
[No]
Inser Pin
Authorize PIN
PIN is correct?
[Yes]
Display Transactions List
[Yes]
ActivityFinal
Lock Card
Display Error
ActivityFinal

Activity 3:



Conclusion: The Activity diagram For Bank Management System has been drawn successfully.

EXPERIMENT NO. 7

Objective: -Draw the sequence diagram for Bank Application System

Hardware Requirements:

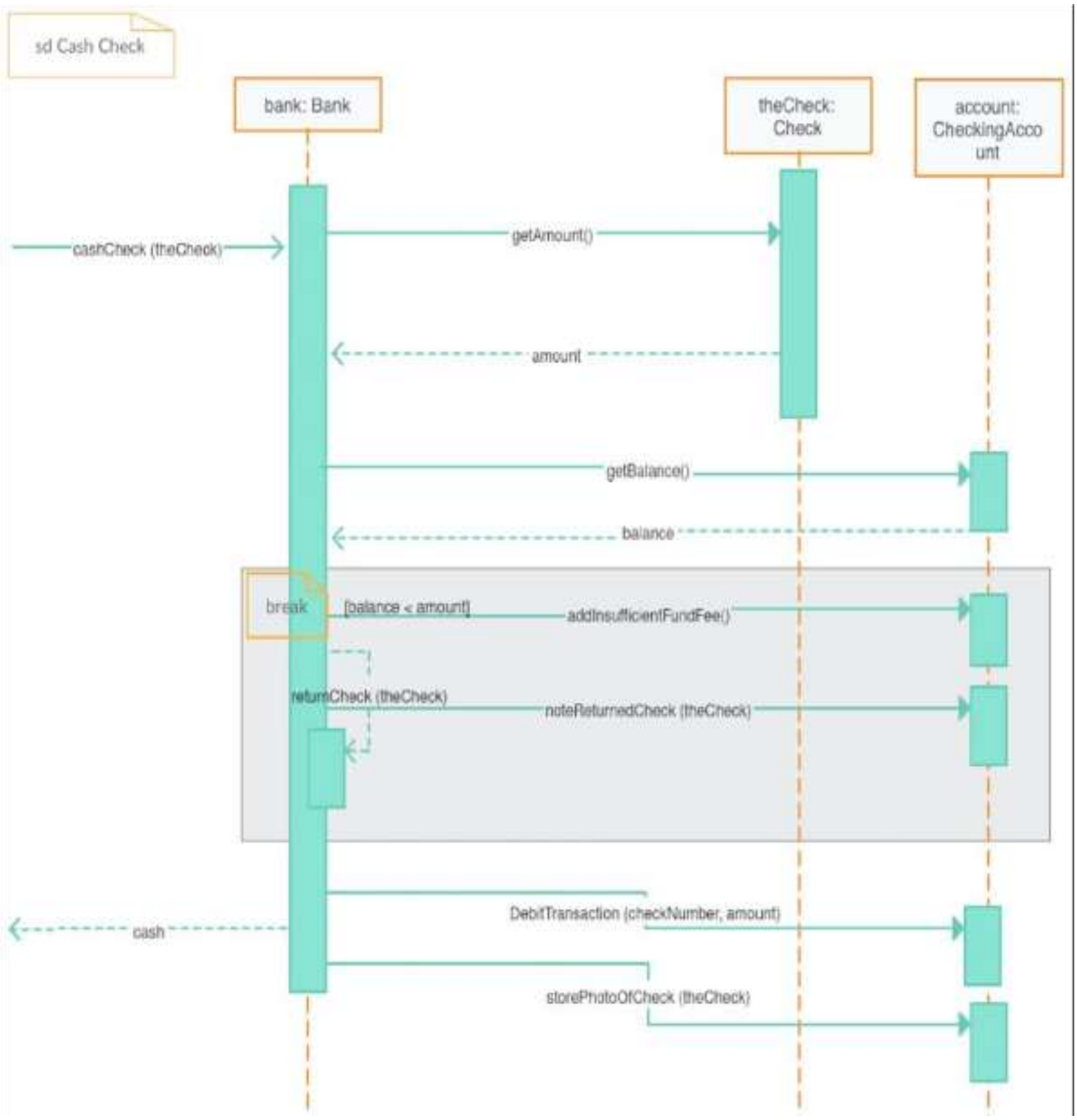
Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard and mouse, colored monitor.

Software Requirements:

Rational Rose, Windows XP

THEORY

UML sequence diagrams model the flow of logic within the system in a visual manner, enabling the user both to document and validate the logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Sequence diagrams, along with class diagrams and physical data models are the most important design-level models for modern application development.



Conclusion: The Sequence diagram For Bank Management System has been drawn successfully.

EXPERIMENT NO. 8

Objective: - Draw an activity diagram for bank application.

Hardware Requirements:

Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.

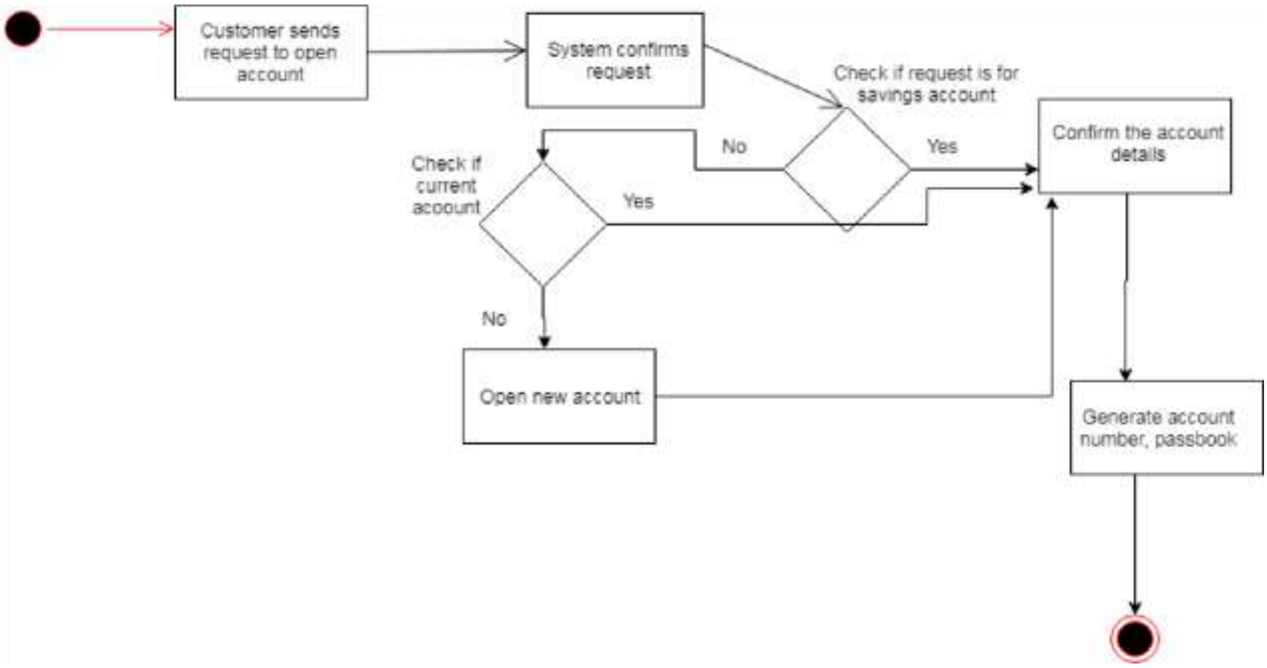
Software Requirements:

Rational Rose, Windows XP,

THEORY

UML 2 activity diagrams are typically used for business process modeling, for modeling the logic captured by a single use case or usage scenario, or for modeling the detailed logic of a business rule. Although UML activity diagrams could potentially model the internal logic of a complex operation it would be far better to simply rewrite the operation so that it is simple enough that you don't require an activity diagram. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams (DFDs) from structured development.

Result:-


Customer sends request to open account
System confirms request
Check if request is for savings account
No
Yes
Confirm the account details
Check if current acoount
Yes
No
Open new account
Generate account number, passbook

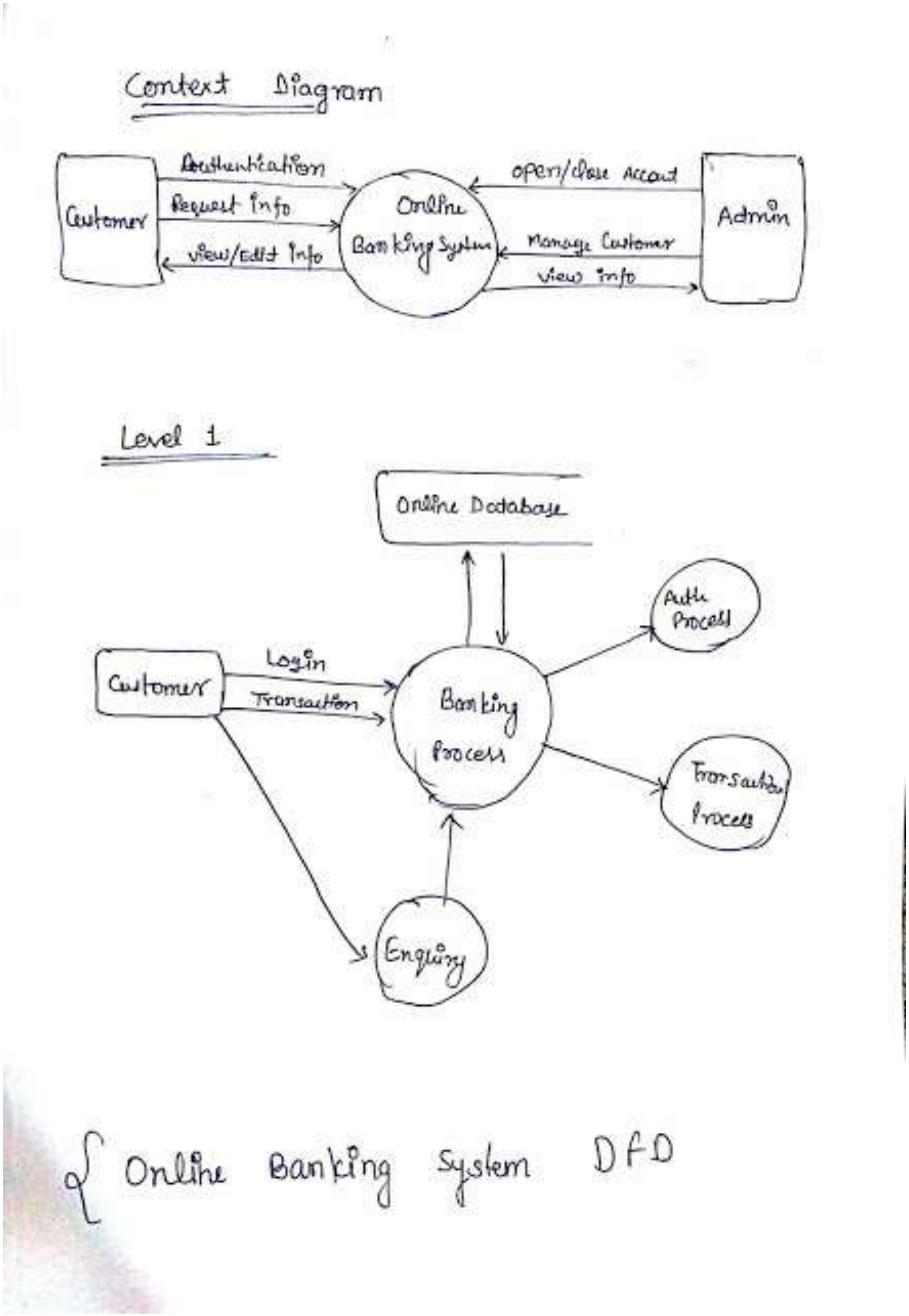
Conclusion: The Activity diagram For Bank Management System has been drawn successfully.

EXPERIMENT NO. 9

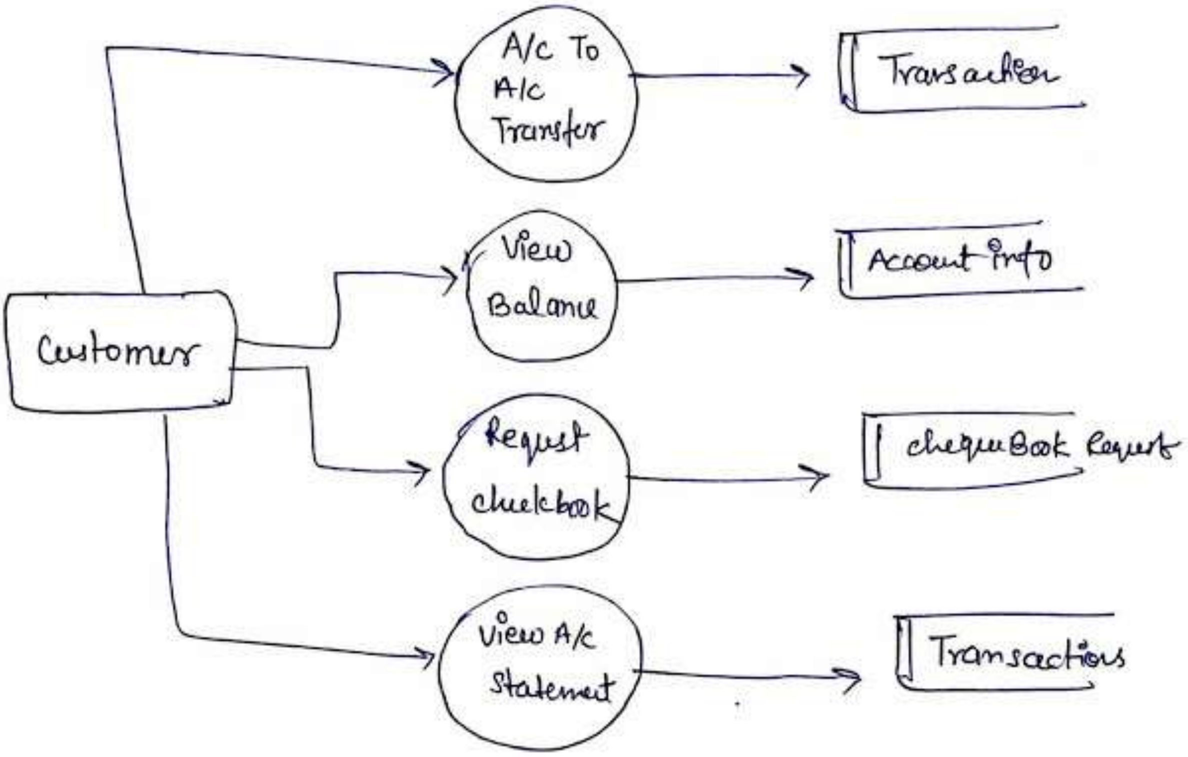
Objective: - To draw a Sample DFD for Bank Management System

Requirements:

1. SOFTWARE REQUIREMENT – Microsoft Word, MS Paint
2. HARDWARE REQUIREMENT – Computer, Keyboard, Mouse, CPU



Level 2 ⇒ DFD



Conclusion: The DFD For Bank Management System has been drawn successfully.