



# **Automated Banking Machine** *for* **MacBank Inc.**

SFWR 3M04  
McMaster University  
Department of Electrical and Computer Engineering

*Presented by*  
Khaled Chebaro, Yaser Jafar, Orin Pereira  
***KYO Engineering Consultants Inc.***

*on*  
***27/11/07***

# OUTLINE

- Introduction
- Request for proposal (RFP)
- Software Requirement Specification (SRS)
- Software Design Specification (SDS)
- Refinement of SDS to low-level design
- Implementation of the ABM system using JAVA
- Demo of the ABM system
- Lessons Learned

# INTRODUCTION

# INTRODUCTION

## ➤ Software Engineering

It is an application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

- **Knowledge**
- **Tools**
- **Methods**



- Software Requirements
- Performing Software Design
- Software Construction
- Software testing
- Software Maintenance Tasks.



# INTRODUCTION

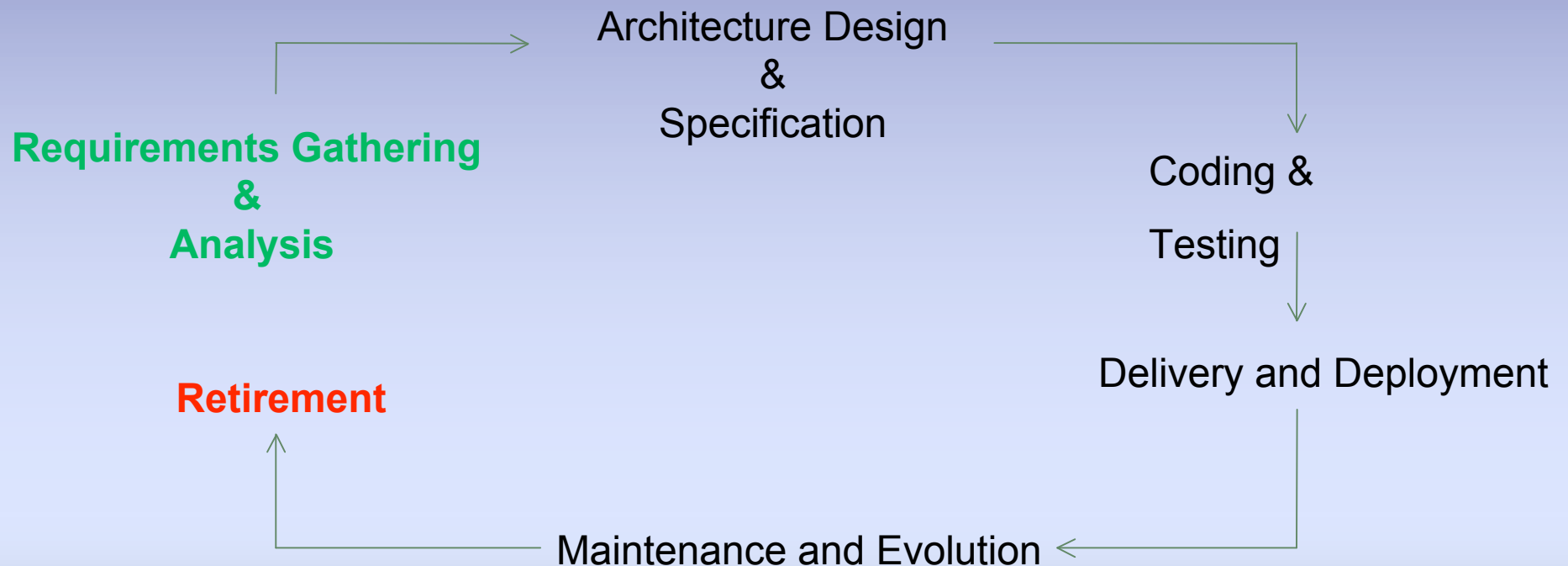
- **Importance of Software Engineering Discipline**
  - Found in products and situations where very high reliability is expected
  - Under demanding conditions,
    - Monitoring and Controlling of power plants
    - Keeping a modern aircraft in the air



# INTRODUCTION

## ➤ Software Life Cycle Process

Refers to the activities that are used for a software product:-



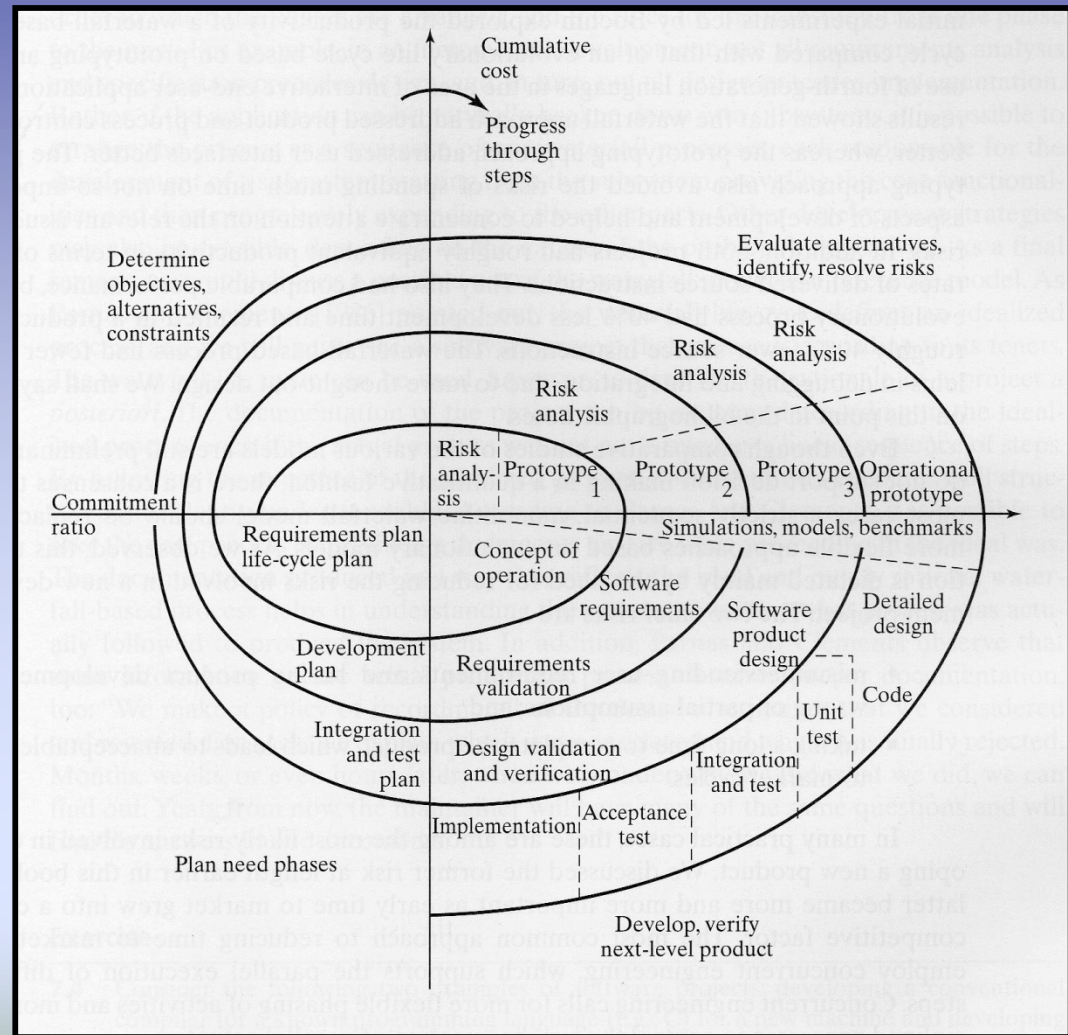
# INTRODUCTION

## ➤ Spiral Model

The spiral model is a development method used in information technology .

It combines the features of the prototyping model and the waterfall model.

The spiral model is intended for large, expensive, and complicated projects.



# **REQUEST FOR PROPOSAL (RFP)**

# REQUEST FOR PROPOSAL (RFP)

## ➤ Constraints obtained from Request for Proposal (RFP)

- GUI must be user friendly.
- Switch to standby mode for maintenance
- Notify Bank Staff when cash is less than \$10,000
- Disable user account after three incorrect PIN entries
- Insertion of the Bank card / entry of the card # is required to use the banking facility
- Withdrawal limit = \$500 per day
- Deposits have no limits and can be in the form of cash or cheques

# REQUEST FOR PROPOSAL (RFP)

## ➤ Types of accounts are:

- Chequing Account
- Savings Account
- Line of Credit
- Credit Card Account

## ➤ Banking Activities include:

- Withdrawing Money
- Depositing Money
- Transferring Money  
between different Accounts
- Paying Bills

# **SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

## ➤ Software Requirement Specification (SRS)

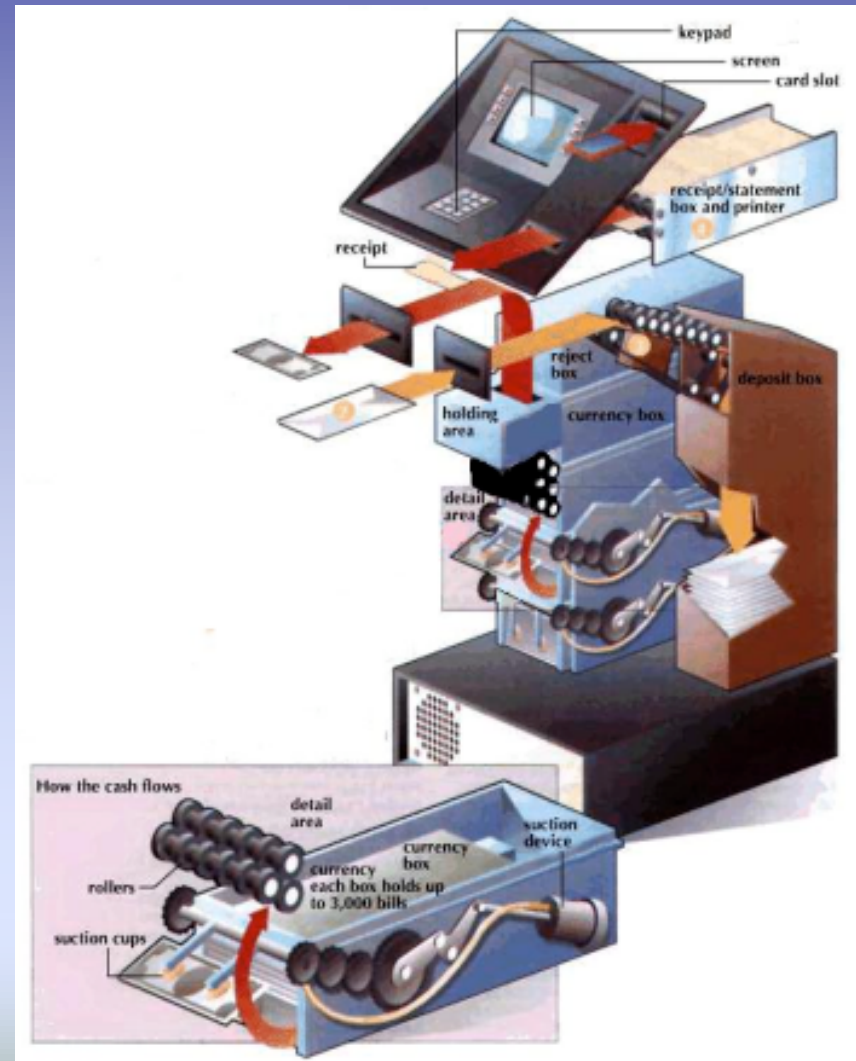
- Simplified Version of the IEEE
- A complete description of the behaviour of the system to be developed
- Includes the external Interfaces and functional specifications for every component



# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

## ➤ Hardware Used:

- Magnetic stripe Card Reader
- Customer Console (Display & Keyboard)
- Envelope Depositing Slot
- Cash Counter and dispenser
- Statement Printer
- Safe
- A key-operated switch for system Start-Up/Shut-down



# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

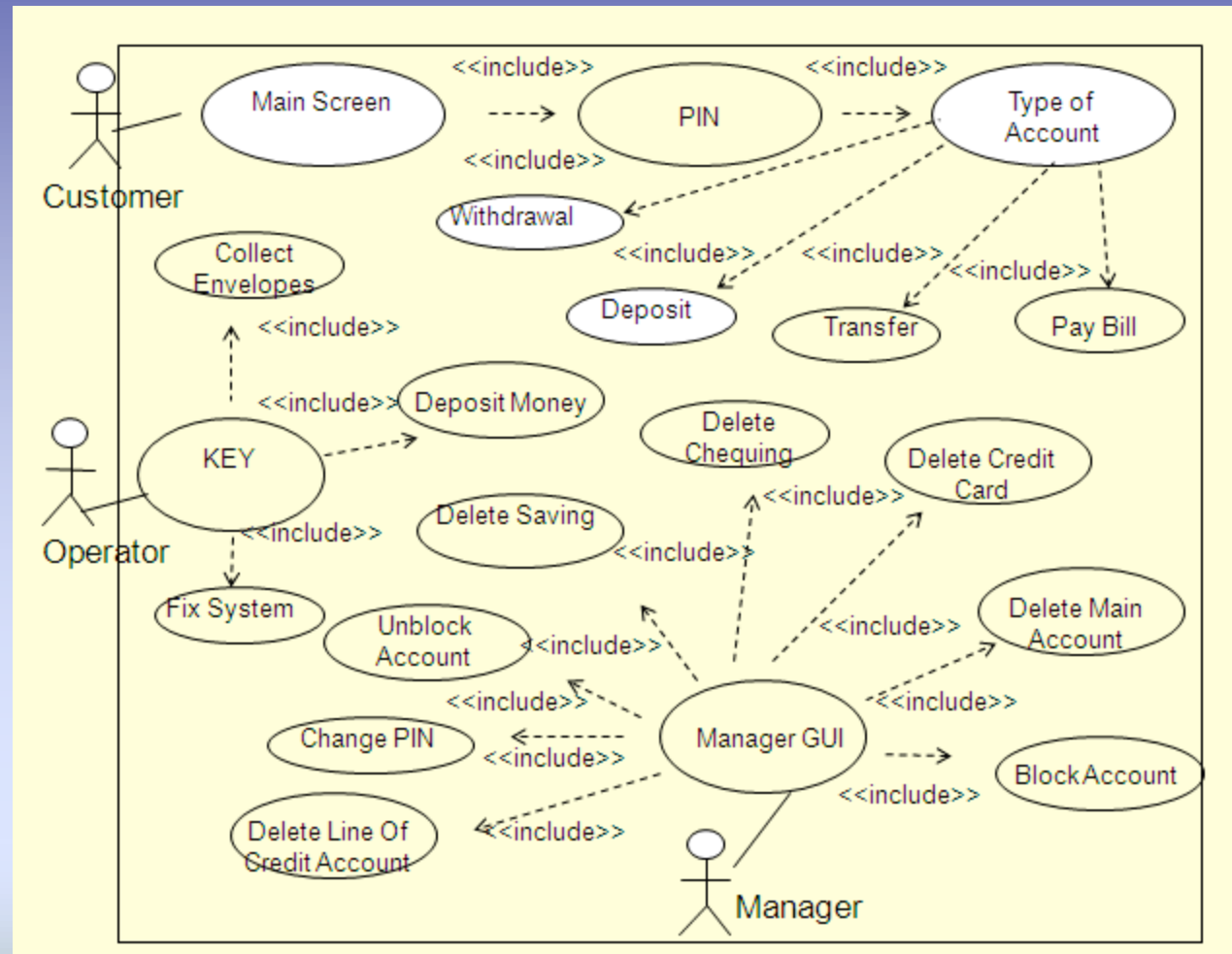
## ➤ Safety and Security

- Receipt must provide relevant information
- Three Incorrect PIN entries deactivates the user account.
- PIN appears as stars on the screen when entered
- Maximum Withdrawal of \$500 per day is allowed from any account

# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

## ➤ UML

It is a modeling language that includes a graphical notation used to create an abstract model of a system, referred to as a *UM model*.



# SOFTWARE REQUIREMENT SPECIFICATION (SRS)

## ➤ Software Quality Attributes

- **Usability**: User friendly
- **Reliability**: Improper usage shouldn't crash the program
- **Maintainability**: Well documented for other users to modify
- **Interoperability**: Synchronization with external bank system
- **Portability**: The source code should be executable in different versions of operating system

# **SOFTWARE DESIGN SPECIFICATION (SDS)**

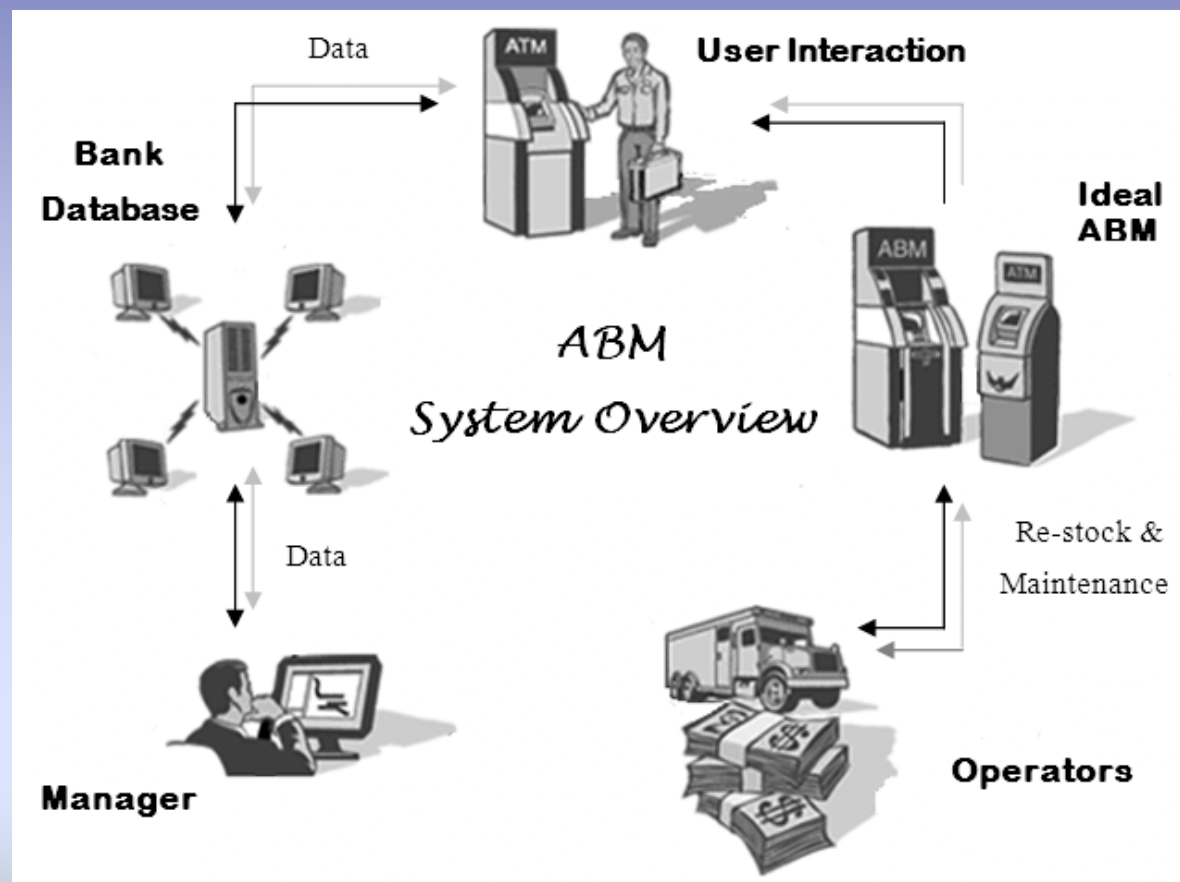
# SOFTWARE DESIGN SPECIFICATION (SDS)

## ➤ Software Design Specification (SDS)

- A high-level design document
- Includes the external interfaces and functional specifications for every component
- Complete set of scenarios that covers all the components' external functions
- Specifies the logic and structure of the main system program

# SOFTWARE DESIGN SPECIFICATION (SDS)

## ➤ System Overview



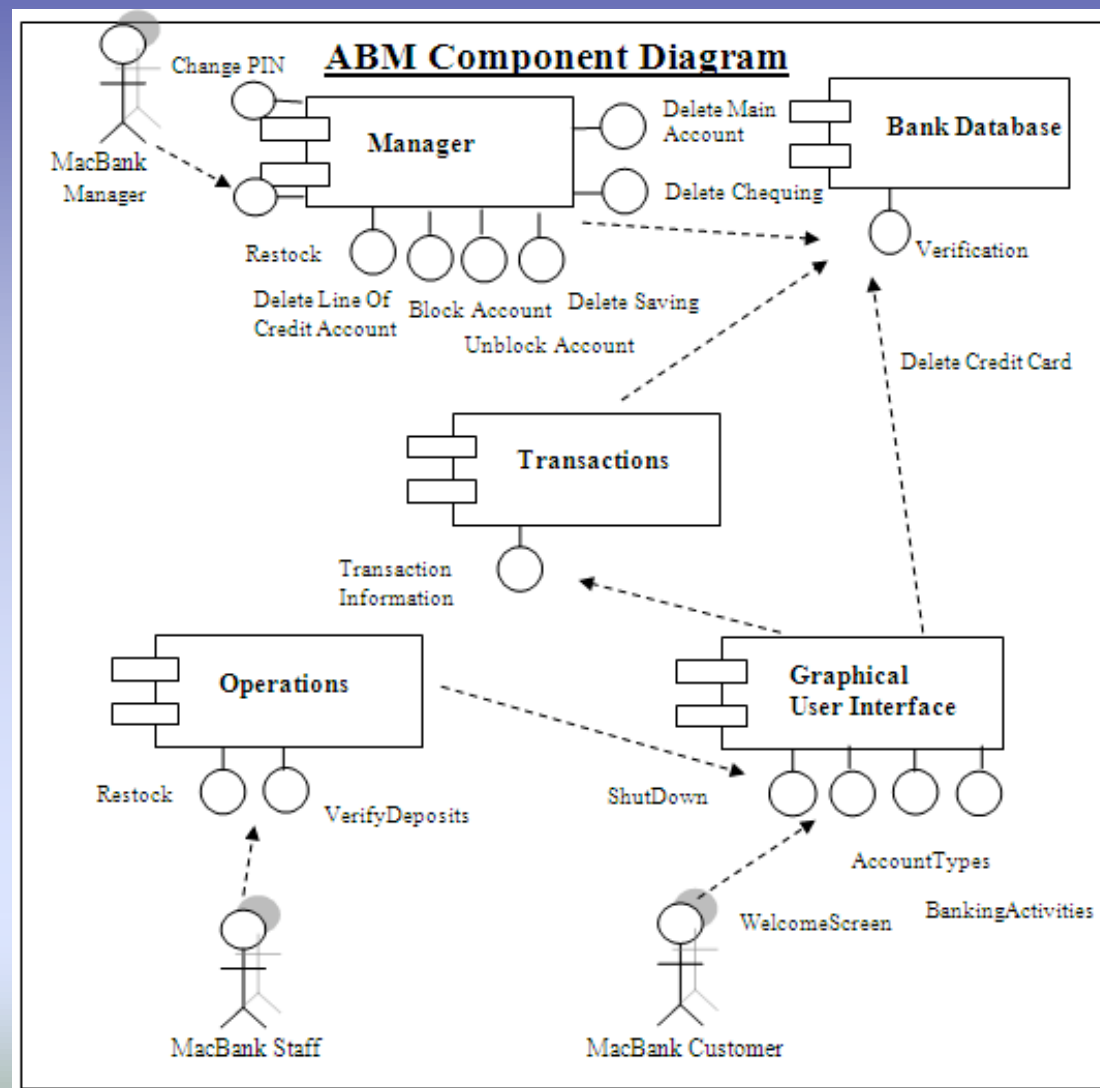
# SOFTWARE DESIGN SPECIFICATION (SDS)

## ➤ **Component Diagram**

Depicts how a software system is split up into physical components and shows the dependencies among these components.



# SOFTWARE DESIGN SPECIFICATION (SDS)



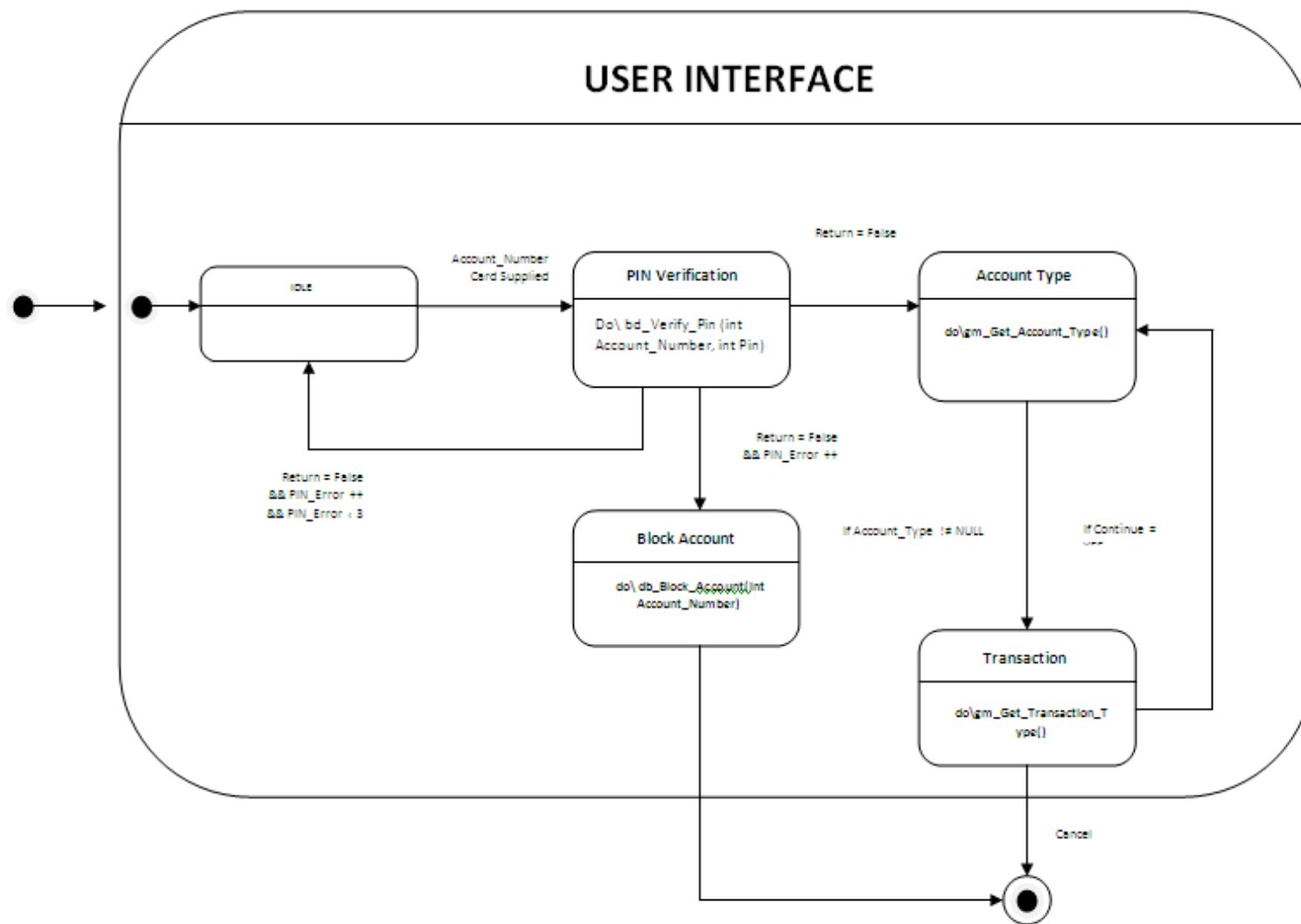
# SOFTWARE DESIGN SPECIFICATION (SDS)

## ➤ **State Chart**

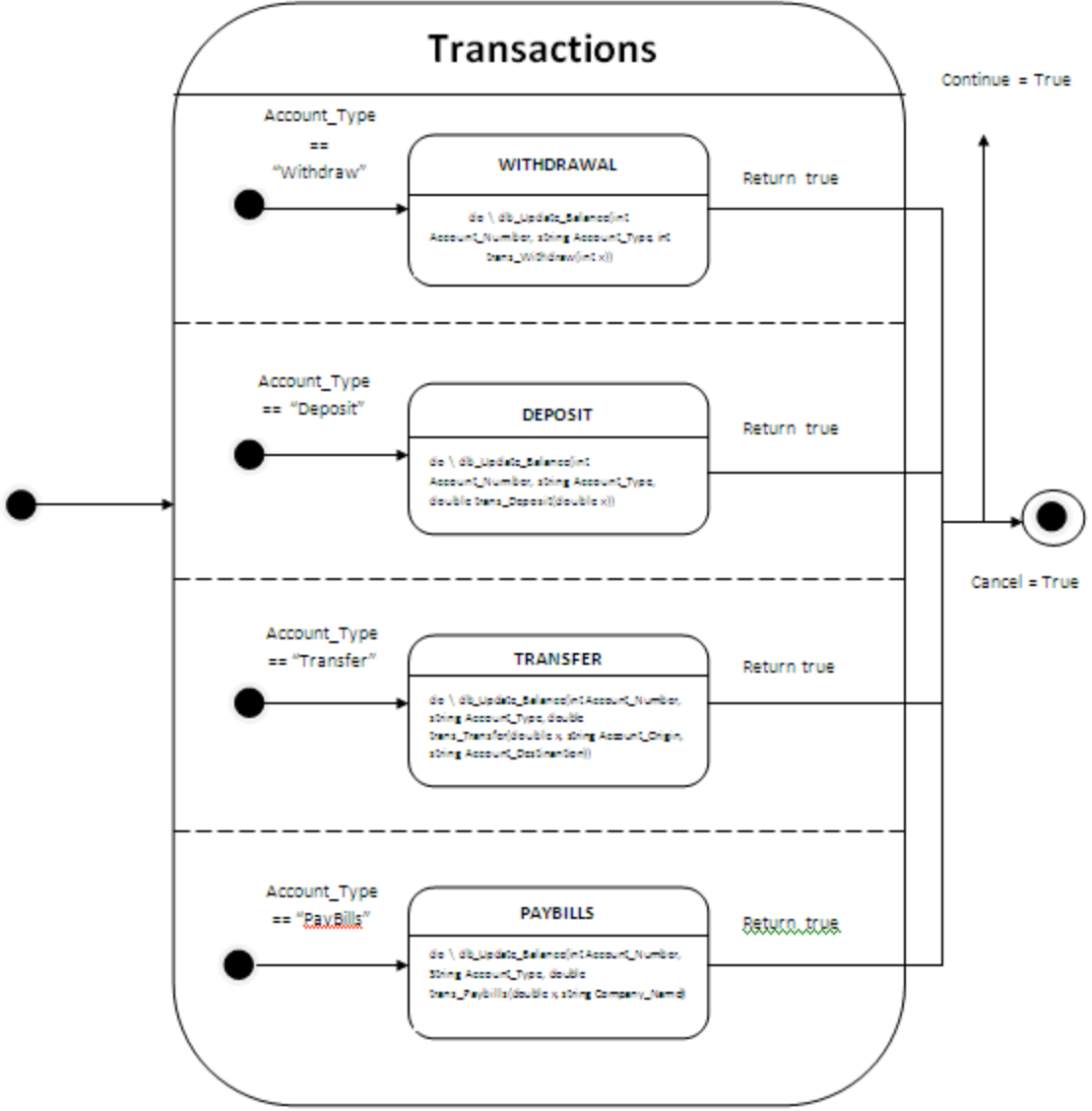
- Used to graphically represent finite state machines
- State transition tables are another possible representation

# **LOW LEVEL DESIGN of SDS**

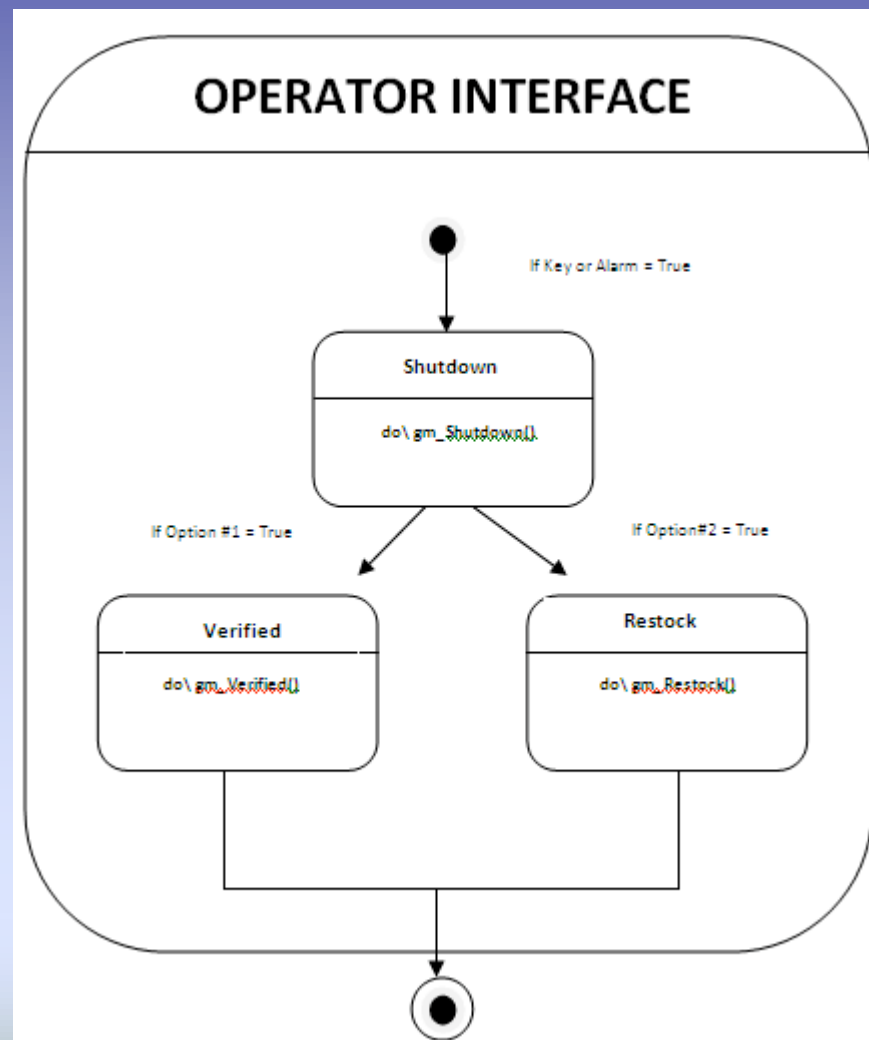
# LOW LEVEL SDS DESIGN



# LOW LEVEL SDS DESIGN



# LOW LEVEL SDS DESIGN



# **IMPLEMENTATION USING JAVA**

# IMPLEMENTATION USING JAVA

## ➤ JAVA

- Programming Language developed by Sun Microsystems.
- Object - Oriented Program (OOP)
- Can run on any Java virtual machine (JVM) regardless of computer architecture
- NetBeans IDE 5.5.1 used in development of Java desktop applications, and an integrated development environment (IDE)





# IMPLEMENTATION USING JAVA

## ➤ Data File Structure



- **AddCustomer** cardNumber pinNumber
- **AddAccount** cardNumber accountType accountNumber

Account types are: 0=Chequing 1=Savings 2=Line of Credit 3=Credit Card

- **AddCheque** cardNumber account amount
- **AddCash** cardNumber account amount
- **BlockAccount** cardNumber

# IMPLEMENTATION USING JAVA

## ➤ Data Structure Array

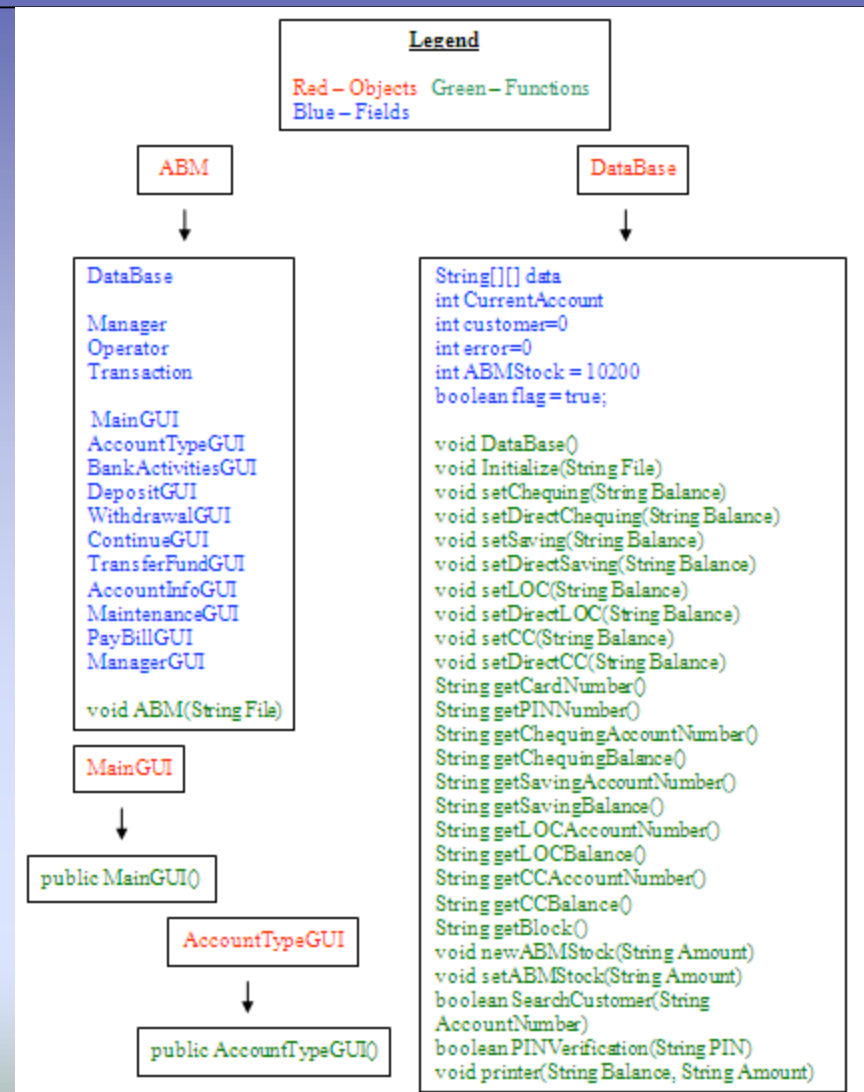
- Structure used to store relevant information for each customer
- It is a field in the database class

```
data[number][11]
```

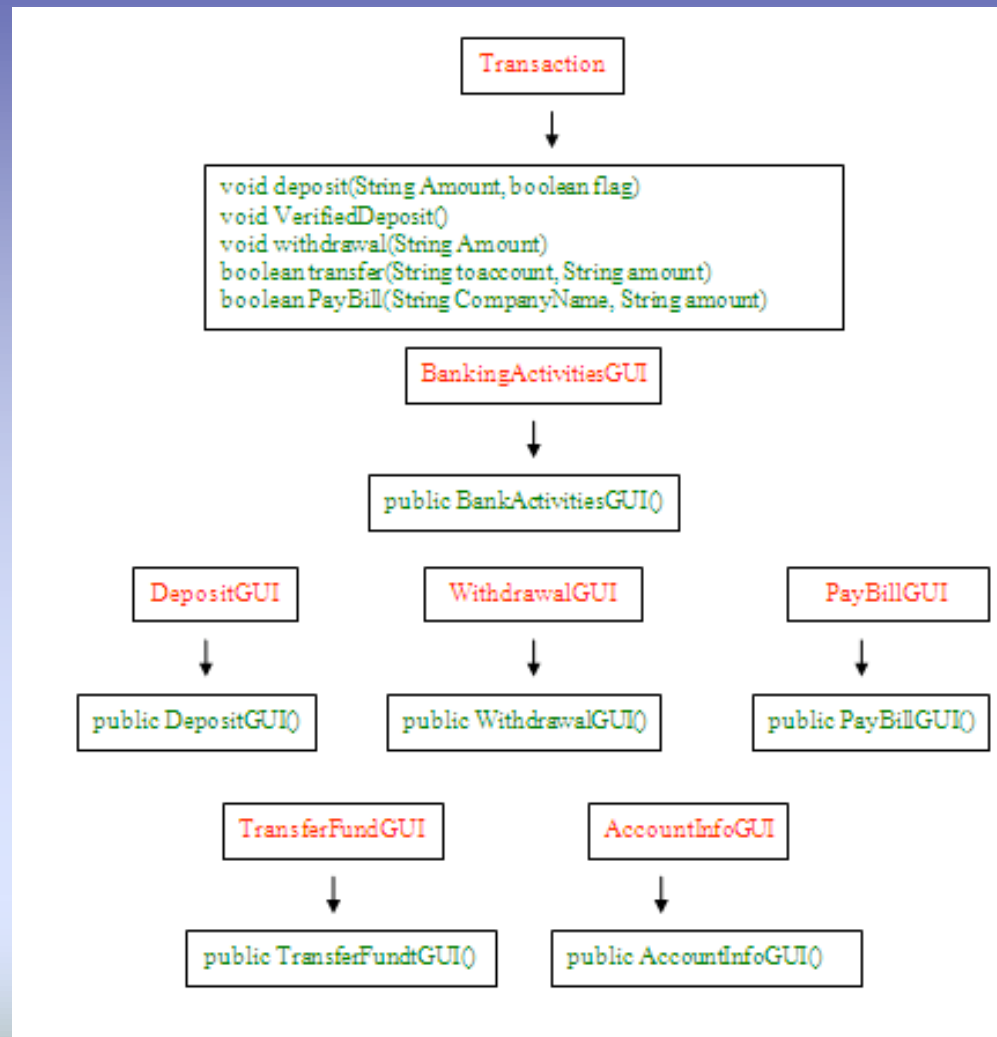
```
Array Format Index
```

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
CardAccount#	PIN	Chequing#	Chequing	Saving#	Saving	LOC#	Line of Credit	CC#	Credit Card	Block
12345678	1234	123456789	1000	234567891	1000	345678912	1000	456789123	100	0

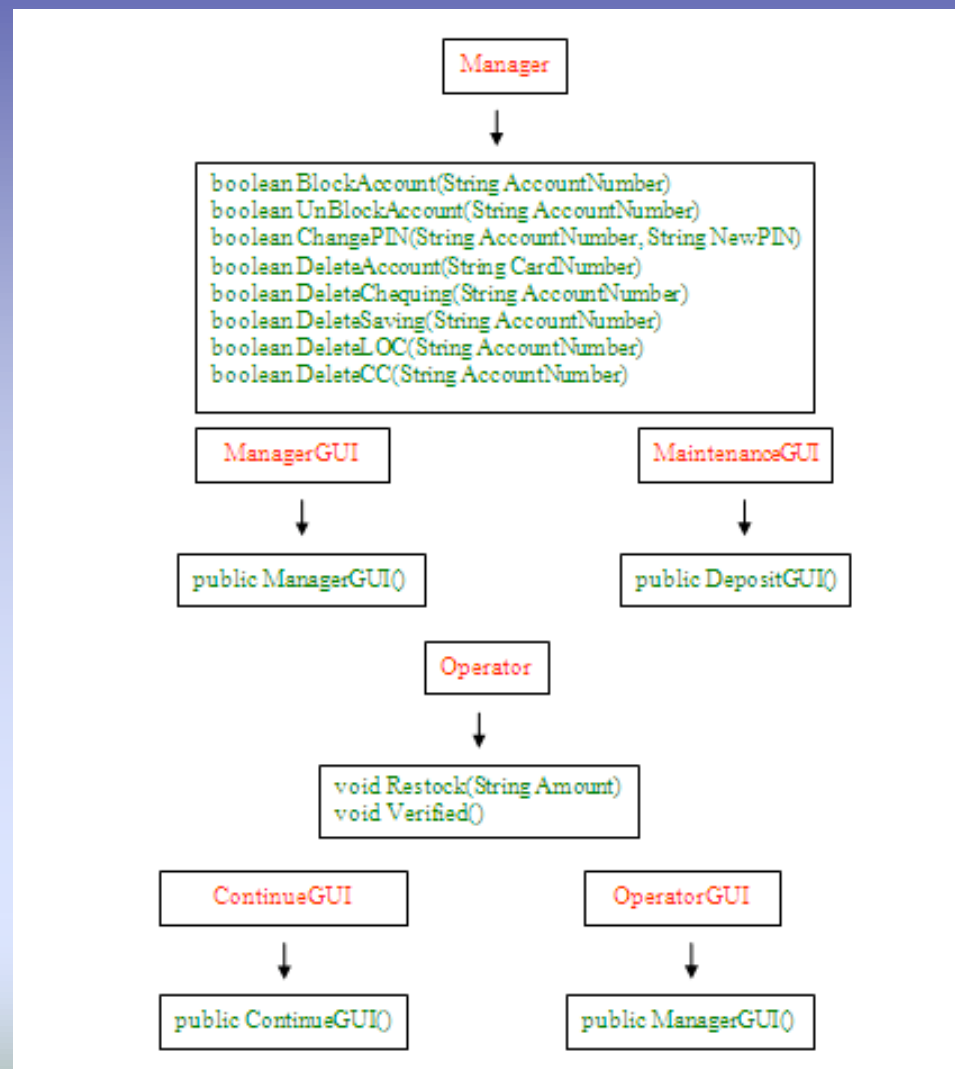
# IMPLEMENTATION USING JAVA



# IMPLEMENTATION USING JAVA



# IMPLEMENTATION USING JAVA



# DEMO

# LESSONS LEARNT

# LESSONS LEARNT

## ➤ Software Engineering is more than just programming

**Encompasses** →

- Computer Engineering
- Computer Science
- Management
- Mathematics
- Project Management
- Quality Management
- Software Ergonomics
- Systems Engineering



# LESSONS LEARNT

- Learnt to prepare documents using the IEEE standards
- Able to recognize various diagrams such as UML, State Charts
- Documentation of source code
- Improve Programming skills
- Enhanced Leadership qualities
- Team spirit
- Time Management
- Organization skills
- Presentation skills

# QUESTIONS ?

The background of the slide features a series of overlapping, wavy, horizontal bands in various shades of blue and green, creating a sense of depth and movement.

# THANK YOU