# Introduction:

Ping-source’s software engineers develop the next-generation technologies that change how billions of users connect, explore, and interact with information and one another. After recently joining Ping-Source , I have been given range of tasks during my probationary one-year period. There are two tasks - Task 1 is of building application for ATM machine and Task 2 is of building Simon game. I have planned to use Java for Task1 and C++ for task 2.

# Task 1:

An automated teller machine (ATM) or the automatic banking machine (ABM) is a banking subsystem (subject) that provides bank customers with access to financial transactions in a public space without the need for a cashier, clerk, or bank teller.

There are two roles from which ATM can be used :

* + - User/client
    - Technician

Both of actors interaction with ATM is managed with bank manager actor who stores record of both actors, user and technician.

**public** List<Accounts> getAccounts(){

//list of accounts from database

**return** Arrays.*asList*(**new** Accounts("Shariq","qwerty",10000),

**new** Accounts("Yasir","qwerty1",1000),

**new** Accounts("Anas","qwerty2",900));

}

**public** List<Technician> getTechnicians(){

//list of technicians from database

**return** Arrays.*asList*(**new** Technician(124431,"qwerty"),

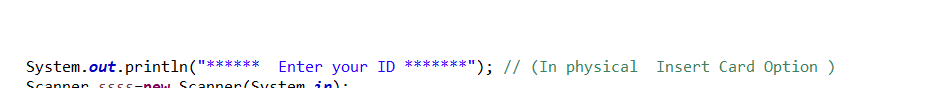
**new** Technician(242431,"abc123"),

**new** Technician(263118,"qwaszx"));

}

More record can be added as per need.

In a real world it user operates with ATM card and user personal pin. User enters card and pin to access its account using ATM and perform his or her desired tasks. User can withdraw , deposit funds to his or her account and check balance and transfer funds using ATM. In a real world user use plastic ATM card but I have consider user account name/ID as alternative for user ATM card for a software or console base application.



Also in a real world technician interacts with ATM directly but to show similar behavior I have user two options whenever ATM application starts , user and technician login. Technician can add more cash to ATM and also he can repair and do relevant services with ATM. Obviously we can show repair or any other function in a software application, technician can only ass cash to application.

## Part a:

Class Diagram for only User interaction:

Diagram

Description automatically generated

There are three classes, ATM machine class handles ATM functions and is main class of application. It can have zero or many customers which are in Accounts class. User can only be manage by 1 bank actor or manager which keeps record of accounts of clients and validate accounts whenever user interact with ATM machine.

**public** Accounts selectAccounts(String accountName,String accountPassword){

**for** (Accounts account:accounts){

**if**(account.isMatching(accountName, accountPassword)){

System.***out***.println("Matched, Welcome "+accountName+"Your balance is "+account.balance);

**return** account;

}

}

System.***out***.print("Not Matched");

**return** **null**;

}

Accounts class object is use for interaction of client with ATM. It keeps record of user balance when user interacts with ATM.

## Part b:

Class diagram with Technician actor:

Diagram

Description automatically generated

All classes are same with addition of Technician class. Technician can add more cash to ATM. ATM has pre define amount of cash when application starts, user cant withdraw more cash then available cash in Atm. ATM machine can manage by 1 technician. Bank can have many or more technicains.

## Part c:

Application is built in JAVA with OOP logic and implementation.

### Interaction:

When program starts there are two options:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you are user press 1 , if you are technician press 2

**User interaction:**

User enter his or her ID and pin to use his or her account.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome to ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\* Enter account holder name \*\*\*\*\*\*\*

Shariq

Enter account password

Qwerty

Matched, Welcome Shariq. Your balance is 10000.0

Then menu displays with 6 options.

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

**Balance inquiry:**

1

Your current balance is : 10000.0

**2. Withdraw:**

\*User cant withdraw cash if cash is not available in bank\*

Enter the amount to withdrraw

11000

Sorry !! Your request could not be proceed ... ATM is out of cash

Enter the amount to withdrraw

1000

Successfull withdraw. Your Remaining balance is: 9000.0

\*User cannot withdraw cash if amount is not available in his account.\*

Enter the amount to withdrraw

4000

You have insufficient balance

**3. Deposit:**

Initial balance of Shariq is 10000

Matched, Welcome Shariq. Your balance is 10000.0

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

3

Enter the amount to deposit

5000

Successfull deposit. Your balance is now: 15000.0

**4. Transfer:**

\*Initial balance of shariq is 10000 and Yasir is 1000, so if Shariq transfer 5000 to Yasir :\*

Matched, Welcome Shariq.Your balance is 10000.0

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

4

Enter the name of reciever account

Yasir

Account Found with name,Yasir

Enter the amount to transfer

5000

Successfull Transfer. Your Remaining balance is: 5000.0

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you are user press 1 , if you are technician press 2

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome to ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\* Enter account holder name \*\*\*\*\*\*\*

Yasir

Enter account password

qwerty1

Matched, Welcome Yasir. Your balance is 6000.0

\*Yasir balance becomes 6000 and Shariq becomes 5000.\*

**Technician interaction:**

\*Technician can add more cash to atm\*

Default available cash in ATM is 10000. User cant withdraw more cash than 9000 , but if technician add more cash he or she can then withdraw more cash.

\*Go through of above statement\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you are user press 1 , if you are technician press 2

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome to ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\* Enter account holder name \*\*\*\*\*\*\*

Shariq

Enter account password

qwerty

Matched, Welcome Shariq.Your balance is 10000.0

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

2

Enter the amount to withdrraw

10000

**Sorry !! Your request could not be proceed** ... **ATM is out of cash**

Menu **:**

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

6

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you are user press 1 , if you are technician press 2

2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM Maintanance\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\* Enter your ID \*\*\*\*\*\*\*

242431

Enter account password

abc123

Matched, Welcome

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Current ATM cash is 9000.0

To add cash press 1 , 0 to exit

1

Enter cash to add to ATM

5000

Updated ATm cash is 14000.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

If you are user press 1 , if you are technician press 2

1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Welcome to ATM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\* Enter account holder name \*\*\*\*\*\*\*

Shariq

Enter account password

qwerty

Matched, Welcome Shariq.Your balance is 10000.0

Menu :

1.Balance ENquiry

2.Withdrawl

3.Deposit

4.Transfer

5.Exit

6. Use ATM Again with new account

2

Enter the amount to withdrraw

10000

**Successfull withdraw.** Your Remaining balance is: 0.0

First shariq try to withdraw 10000 which is greater than available cash 9000, so it cant withdraw or withdrawal request was rejected, then technician added more 5000 cash then Shariq withdrawal request was completed.

# Task 2:

This Simon game is built with C++. In this Simon game, 4 colors that are , RED GREEN BLUE YELLOW appears in a pattern and disappears , user Input same pattern , if it matched another pattern appears with 1 more light , if it does not match final score of user displays.

## Part a:

First welcome screen displays with some information of how to play game.

**Welcome to the Simon color game!**

**R=red , G= greeen , B= blue , Y = yellow**

**Enter the patern in specific order it displays everytime, without space**

**Press any key to continue . . .**

When user press any keys, pattern appears and disappears.

Text

Description automatically generated

Then User is asked to enter pattern.

I User enter correct pattern then:

Text

Description automatically generated

New patterns appears with addition of one more color from colors, if user then inputs wrong pattern then:

Text

Description automatically generated

**One more play:**

1st PATERN: GBRG

2ND PATERN: GYBGY

3RD PATERN: GBGYRY

4TH PATERN: BRYGGGR

5TH PATERN: BYRGYRGRG

**CONCOLE RUN:**

**Welcome to the Simon color game!**

**R=red , G= greeen , B= blue , Y = yellow**

**Enter the patern in specific order it displays everytime, without space**

**Press any key to continue . . .**

**Enter patern**

**GBRG**

**Your guess is right, your score is 1**

**Enter patern GYBGY**

**Your guess is right, your score is 2**

**Enter patern GBGYRY**

**Your guess is right, your score is 3**

**Enter patern BRYGGGR**

**Your guess is right, your score is 4**

**Enter patern BYRGRGG**

**Your guess is wrong , your final score is 4**

**Thank you!!**

**Logic:**

There are 3 global vector strings:

std::vector<std::string> colour{ "R", "B", "G","Y" };

std::vector<std::string> pattern;

std::vector<string> userpatern;

1st vector is of available colors. 2nd vector is used to keep track of generated colors pattern. 3rd vector is sued to keep track of user input pattern.

First pattern is generated using display pattern function:

void displayPatern() {

pattern.clear();

srand(time(NULL));

for (int i = 0; i < r+3; i++) {

v1 = rand() % 4;

Sleep(300);

std::cout << colour[v1] << " ";

pattern.push\_back(colour[v1]);

}

Sleep(800);

cout << "\33[2K";

cout << "\n Enter patern ";

}

Whenever this function is call in main it clear the previous generated pattern and genreted new pattern with respect to **r** variable. **r** variable keeps track of each round user is in and it increment with every user right guess.

**\33[2K** erases the entire line your cursor is currently on. So that user cant see the generated pattern after sleep of 800ms.

Then user input pattern function is call:

void userInput() {

userpatern.clear();

string temp;

std::cin >> temp;

int n = temp.length();

for (int i = 0; i < temp.length(); i++) {

string s(1, temp[i]);

userpatern.push\_back(s);

}

}

Which first erase previous user pattern and adds user input string with breakdown of characters in user pattern vector.

Then compare pattern function is used to compare both patterns and to terminate or continue accordingly:

boolean comparePatern() {

if (userpatern == pattern)

return true;

return false;

}

## Part b:

Application is working find with wrong inputs and tested thoroughly.

Text

Description automatically generated

As I have use comparison vectors and string so there is no problem of index out of bound exception as there are dynamic data structures.