CORE JAVA PROJECT

ATM INTERFACE

By
Dhande Punam

❖ Introduction :-

Automated Teller Machine enables clients of a bank to have access to their account without going to the bank. This is achieved only by development the application using online concepts.

In the ATM interface project, the user has to select an option from the options displayed on the screen. The options are related to withdraw the money, deposit the money, check the balance, and exit.

To withdraw the money, we simply get the withdrawal amount from the user and remove that amount from the total balance and print the successful message.

To deposit the money, we simply get the deposit amount from the user, add it to the total balance and print the successful message.

To check balance, we simply print the total balance of the user

❖ AIM :-

- To achieve transactions in the account more securely and easily.
- To save the mini-statements of each transaction.
- To achieve Fastest way to deposit withdraw money.

- **Hardware Requirements :-**

Processor	Intel Core i3
Memory	32 MB Ram

- **Software Requirement :-**

Operating System	Window
Front End	Java
Back End	MySQL

❖ Modules :-

In this ATM INTERFACE Project there has 6 main operations.

1. Verification Process → verifying user by using username and pin
2. View Balance →User can view account balance.
3. Deposit →User can add money to existing account
4. Withdraw →User can withdraw money from your account
5. View Mini Statements →User can view all transaction history.

INITIAL SETUP :-

Software Editor :- Eclipse
Software Database : MySQL
Maven Project Name :- ATM Interface
Package Name :- com.project
Classes Name :- > MainClass
>Get_Record_Atm

❖ (IN DATABASE) :

creating and using database-

```
mysql> create database ATM_INTERFACE;
Query OK, 1 row affected (1.47 sec)

mysql> use ATM_INTERFACE;
Database changed
mysql>
```

❖ Creating new table for storing transactions:

```
ysql> create table transaction;
ERROR 4028 (HY000): A table must have at least one visible column.
ysql> create table transaction(deposit int,withdraw int,Available int,accno int,foreign key(accno) references atm(accno));
uery OK, 0 rows affected (2.05 sec)

ysql> insert into transaction values(5000,0,5000,1000);
uery OK, 1 row affected (0.19 sec)

ysql> select * from transaction;
+---------+----------+-----------+-------+
| deposit | withdraw | Available | accno |
+---------+----------+-----------+-------+
|    5000 |        0 |      5000 |  1000 |
+---------+----------+-----------+-------+
 row in set (0.00 sec)
```

❖ Creating describing table structure :

```
mysql> create table atm(accno int primary key,username varchar(20) not null,pin_no int not null,available int);
Query OK, 0 rows affected (4.80 sec)

mysql> desc atm;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| accno     | int         | NO   | PRI | NULL    |       |
| username  | varchar(20) | NO   |     | NULL    |       |
| pin_no    | int         | NO   |     | NULL    |       |
| available | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

❖ Inserting and Displaying records:

```
ysql> insert into atm values(1000,'punam',123,0);
Query OK, 1 row affected (0.46 sec)

ysql> insert into atm values(1001,'dhande',124,0);
Query OK, 1 row affected (0.19 sec)

ysql> select * from atm;
+-------+----------+--------+-----------+
| accno | username | pin_no | available |
+-------+----------+--------+-----------+
|  1000 | punam    |    123 |         0 |
|  1001 | dhande   |    124 |         0 |
+-------+----------+--------+-----------+
2 rows in set (0.00 sec)
```

THANK YOU