TECHNICAL ASSESSMENT NEUEDA

ATM

Table of Content

How to run the ATM application?

Please find below the steps to run the ATM application as a part of Neueda Technical Assessment.

A. Installation and cloning the application.

a. Download and install docker.
b. Now, clone the repository:
   Command: *git clone https://github.com/klavania/ATM_Assessment.git*
c. Start docker
d. Enter into the directory/folder where we have dockerfile and application war available:
   Command: *cd ATM_Assessment/*

B. Creating mysql image and running the container.

a. Mysql runs on port 3306 and needs an environment variable to set the password. This command will start a docker container using mysql image.
   Command: *docker run --name mysqldb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root -d mysql*
b. Enter into the “mysqldb” container to access mysql.
   Command: *docker container exec -it mysqldb bash*
c. To start mysql, we need to enter the username and password.
   Command: *mysql -uroot -proot*
d. Once we have mysql running in the container, fire the following queries. We can copy all the queries and run them at once.

```
create database atm;
use atm;
create table account(account_number int(10) primary key,pin int(4),
opening_balance int(10), over_draft int(10) );
insert into account values(123456789,1234,800,200);
insert into account values(987654321,4321,1230,150);
select * from account;
create table atm_balance(balance varchar(5),fifty int(4),twenty int(4),ten
int(4),five int(4));
insert into atm_balance values('atm',10,30,30,20);
```

select * from atm_balance;
```

e. Above queries will create a database along with a couple of tables; account, atm_balance with the data required to check all the project's functionality.
f. To exit mysql, type "quit". Later, exit the container by typing "exit" from the bash.
g. Once we are outside the container, check if our mysql container is running.
   Command: *docker container ls*

```
mysql> create database atm;
Query OK, 1 row affected (0.01 sec)

mysql> use atm;
Database changed
mysql> create table account(account_number int(10) primary key,pin int(4), opening_balance int(10), over_draft int(10) );
Query OK, 0 rows affected, 4 warnings (0.03 sec)

mysql> insert into account values(123456789,1234,800,200);
Query OK, 1 row affected (0.01 sec)

mysql> insert into account values(987654321,4321,1230,150);
Query OK, 1 row affected (0.00 sec)

mysql> select * from account;
+----------------+------+-----------------+------------+
| account_number | pin  | opening_balance | over_draft |
+----------------+------+-----------------+------------+
|      123456789 | 1234 |             800 |        200 |
|      987654321 | 4321 |            1230 |        150 |
+----------------+------+-----------------+------------+
2 rows in set (0.00 sec)

mysql> create table atm_balance(balance varchar(5),fifty int(4),twenty int(4),ten int(4),five int(4));
Query OK, 0 rows affected, 4 warnings (0.02 sec)

mysql> insert into atm_balance values('atm',10,30,30,20);
Query OK, 1 row affected (0.01 sec)

mysql> select * from atm_balance;
+---------+-------+--------+------+------+
| balance | fifty | twenty | ten  | five |
+---------+-------+--------+------+------+
| atm     |    10 |     30 |   30 |   20 |
+---------+-------+--------+------+------+
1 row in set (0.00 sec)

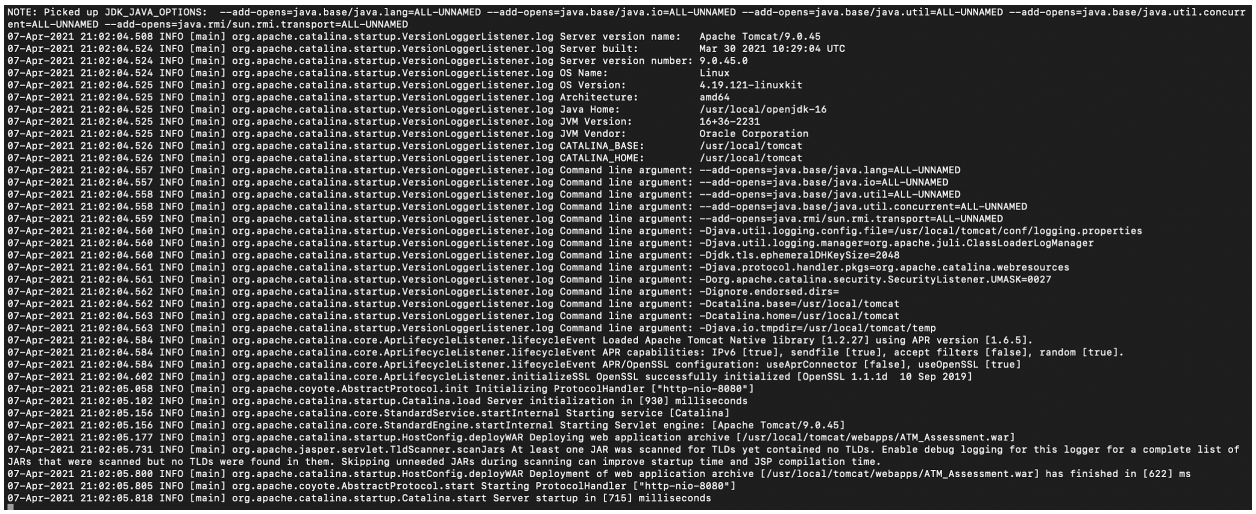
mysql> quit
Bye
root@9aaf08784162:/# exit
exit
```

C. Building application's image.

a. It's important to be in the directory where we have the dockerfile present. Check with the "ls" command.
b. To build the image of the application, run the docker build command and name the image. Note: this would require a "." (dot) at the end or after the build keyword.
   Command: *docker build -t atm_assess .*
```

c. Check if our application and mysql images are present.
Command: *docker image ls -a*

D. Running docker container for the ATM_Assessment application.
   a. We will run the container with the docker run command only assigning the port 80:8080. The **important** part in this command is to link the already running “mysqldb” container.
   Command: *docker run --name ATM_Assess_Container -p 80:8080 --link mysqldb atm_assess*



E. Let’s run the application now :)

URL: http://0.0.0.0/ATM_Assessment/

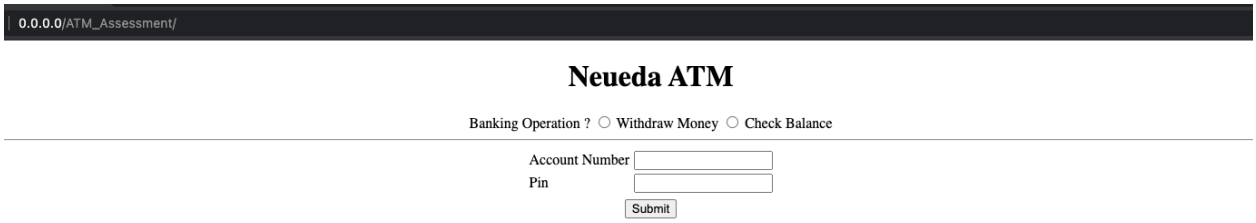
Project Implementation

- IDE: Eclipse
- Database: MySQL
- Language: Java
- Application Server: Tomcat
- Architecture: MVC
- Database Connectivity: JDBC
- Unit Testing Framework: JUnit
- Some external jars are used which are included in the project *WEB-INF/lib* folder.
- Deployment Tool: Docker

Application Functionalities

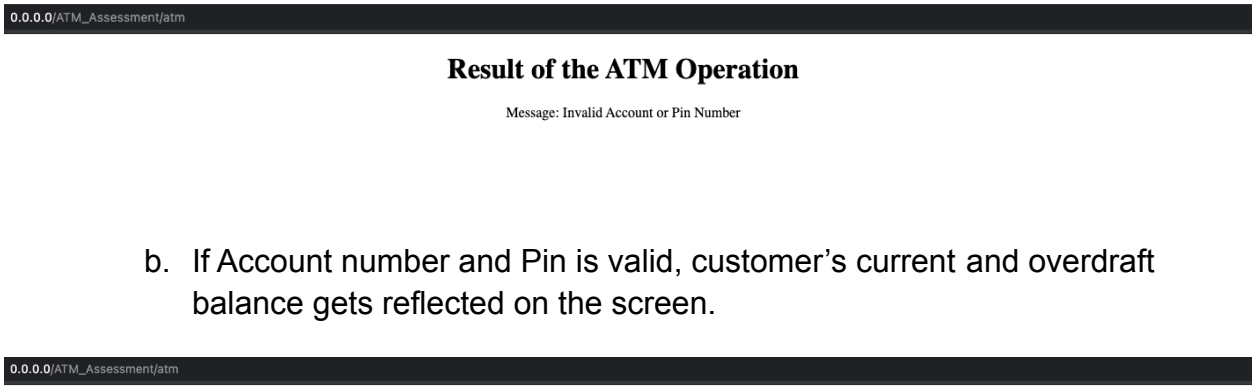
The following functional requirements are been taken care of:

A. Customer can choose between the two ATM operations:
   a. Check Balance
   b. Withdraw Money

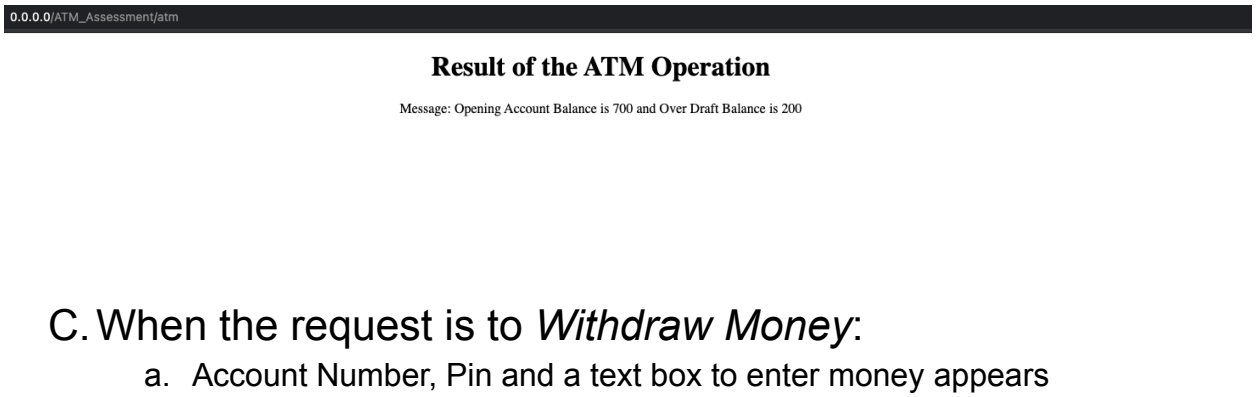


B. *Check Balance* has following functionalities:

   a. Enter Account Number and Pin. If it's incorrect, "Invalid Account or Pin Number" message pops up.

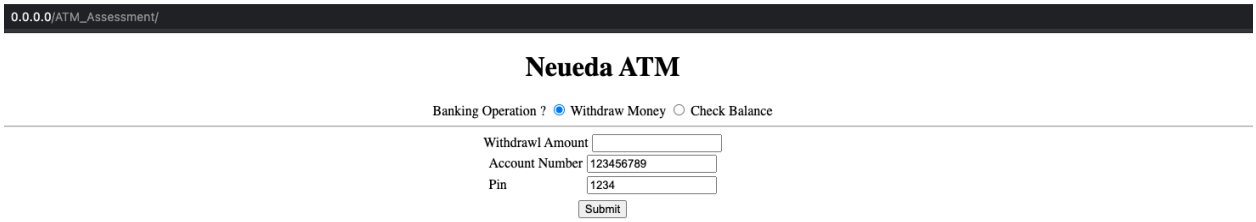


b. If Account number and Pin is valid, customer's current and overdraft balance gets reflected on the screen.

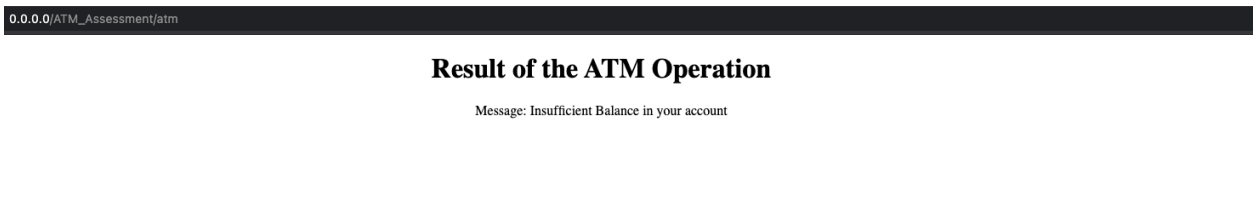


C. When the request is to *Withdraw Money*:

a. Account Number, Pin and a text box to enter money appears

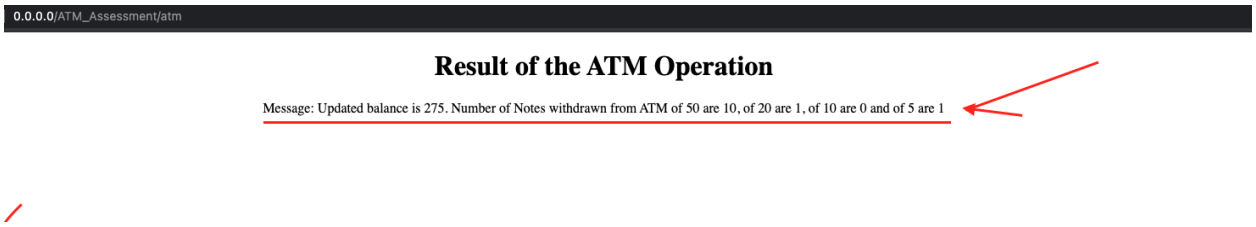


b. If the details are incorrect, "Invalid Account or Pin Number" message pops up.
c. If the customer does not have sufficient balance in the opening account as well as in the overdraft, the customer will not be able to withdraw money and the following message will be displayed "Insufficient Balance in your account".



d. If the balance is available in both current as well as in overdraft, a minimum number of notes will be drawn from the ATM. We can also see the notes withdrawn on the screen.

For an instance, if there is a request to withdraw €525, customer will see the notes distribution as shown in the screenshot below:



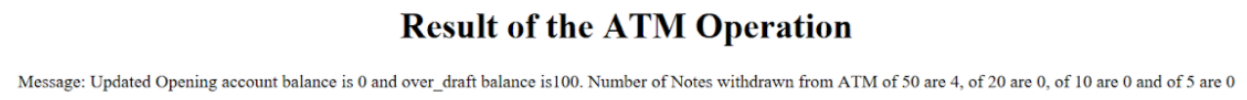
e. The request will be fulfilled from the opening account first and if the balance is lower than the requested amount, it will go into overdraft and the remaining money will be deducted from there.

   Opening Acc balance: 100

   Overdraft limit: 200

   Amount to be withdrawn: 200

   100 will be deducted from the opening balance and rest from the overdraft.



f. If there is no money left in the opening account, and the condition fulfills the overdraft. Money will be taken from overdraft itself.
g. If there is no money left in the ATM machine. Machine will return “Amount cannot be withdrawn”

0.0.0.0/ATM_Assessment/atm

Result of the ATM Operation

Message: Amount cannot be withdrawn.

h. If there is an invalid entry of the amount. The machine will again return an “Amount cannot be withdrawn” message.

   Amount requested: 71

   Since there is no 1 euro note available. “Amount cannot be withdrawn” message will be displayed.

This is the end of the document. Thank you!