**ADJ BANKING SOFTWARE**

1. **Introduction**

ADJ Banking Software maintains client’s bank accounts and providing basic banking services such as deposit, withdraw, and change PIN. The group tried to recreate a real-life banking management services by providing a Python application of such which consist of two main parts: The administrative module and the ATM module. The administrative module is used and accessed by the System Administrators or Bank Tellers. It has the following functionalities: To open, close accounts and view account information. In opening accounts, the new account will be added to the database system. In closing accounts, an existing account will be deleted in the database. In view account information, an account number is required to be entered to view the particular account. On the other hand, the ATM module side can be used by the client in order to do deposit, withdraw, balance inquiry, or change pin transaction. First, the user must be authenticated by putting his/her card number (ATM card insertion in real-life ATM) together with the PIN number. All except balance inquiry will update the database after execution. In summary, the administrative module has open close account functionalities while ATM module has deposit, withdraw, change pin and check balance transaction.

1. **Objectives**

* To be able to implement a banking management software using Python programming by providing an application that can offer basic bank transactions.
* To be able to utilize SQLite as the database management system of the software.

1. **Functions**

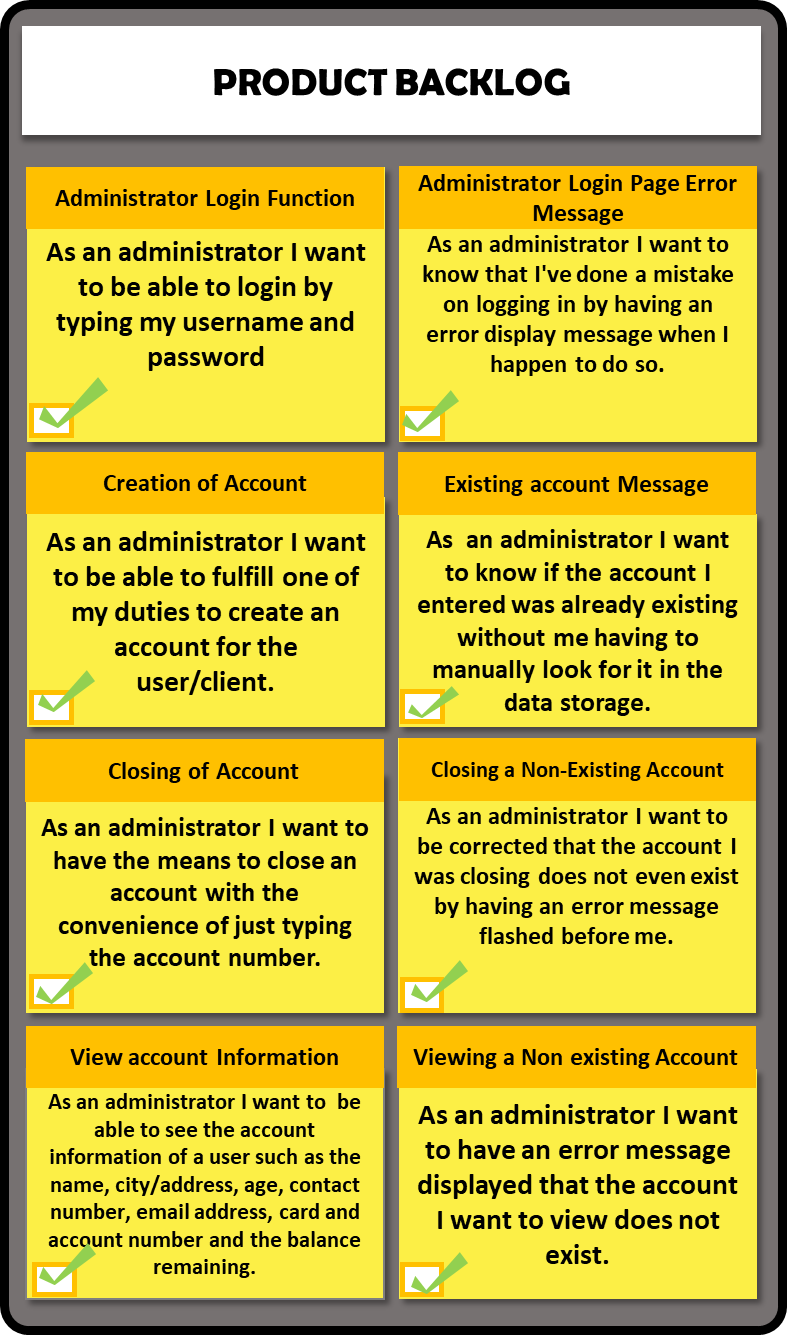
**Administrator**

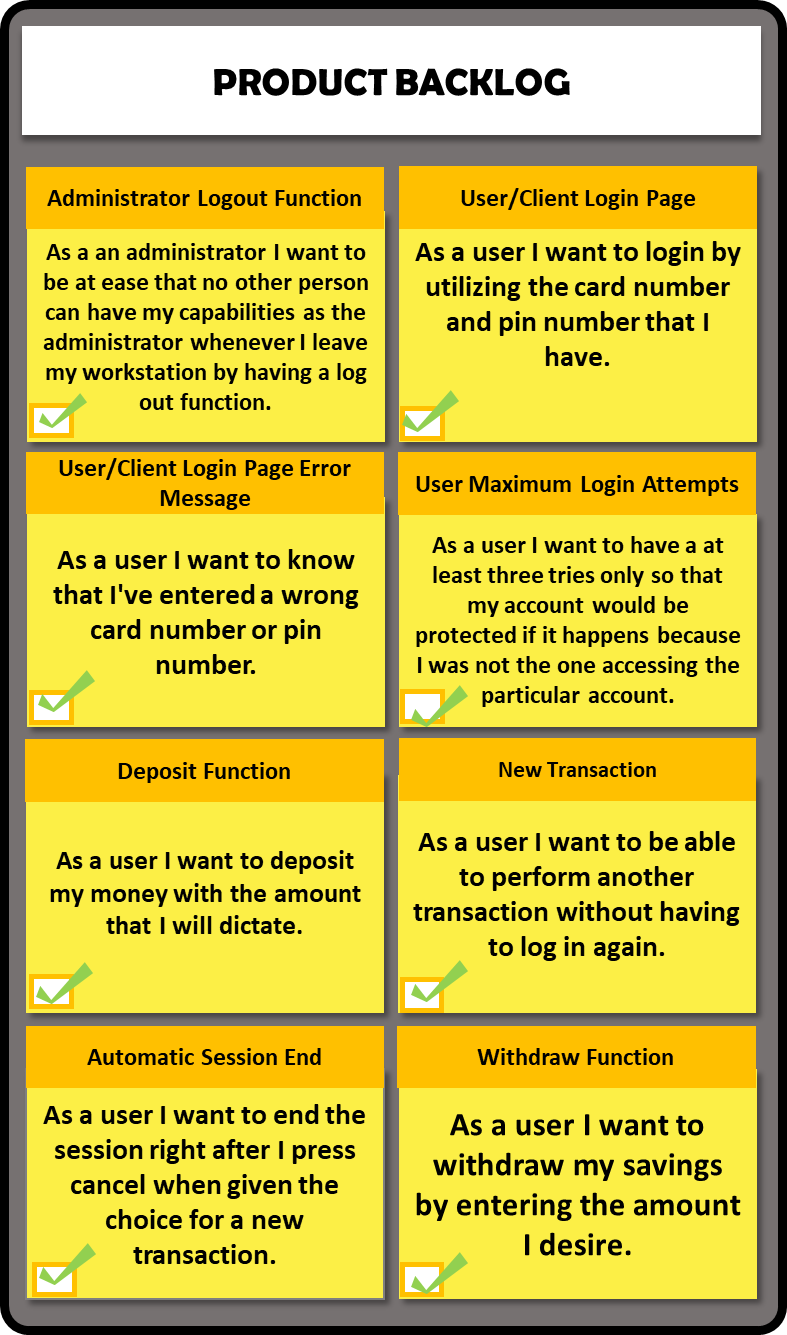
* Open Account - the new account will be added to the database system.
* Close Account - an existing account will be deleted in the database.
* View Account - an account number is required to be entered to view the particular account.

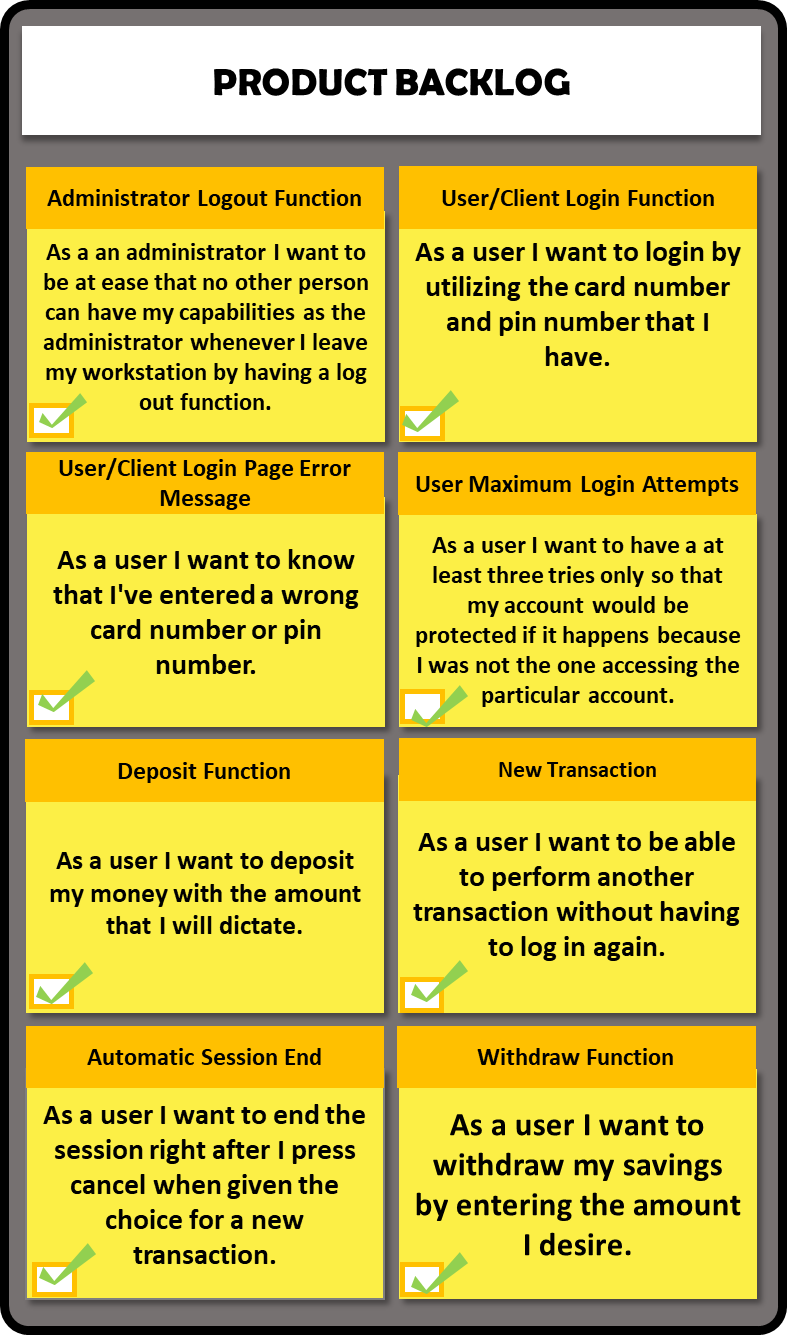
**Client (ATM MODULE)**

* Deposit – prompts the user to enter the desired amount to be deposited.
* Withdraw - prompts the user to enter the desired amount to be withdrawn.
* Balance inquiry – Displays the current balance of the user as determined in the SQLite database.
* Change Pin transaction – Requires the user to enter a new pin number. Reentering the new pin number is also required to make sure that the entered pin is the correct desired pin.

1. **Product Backlog**

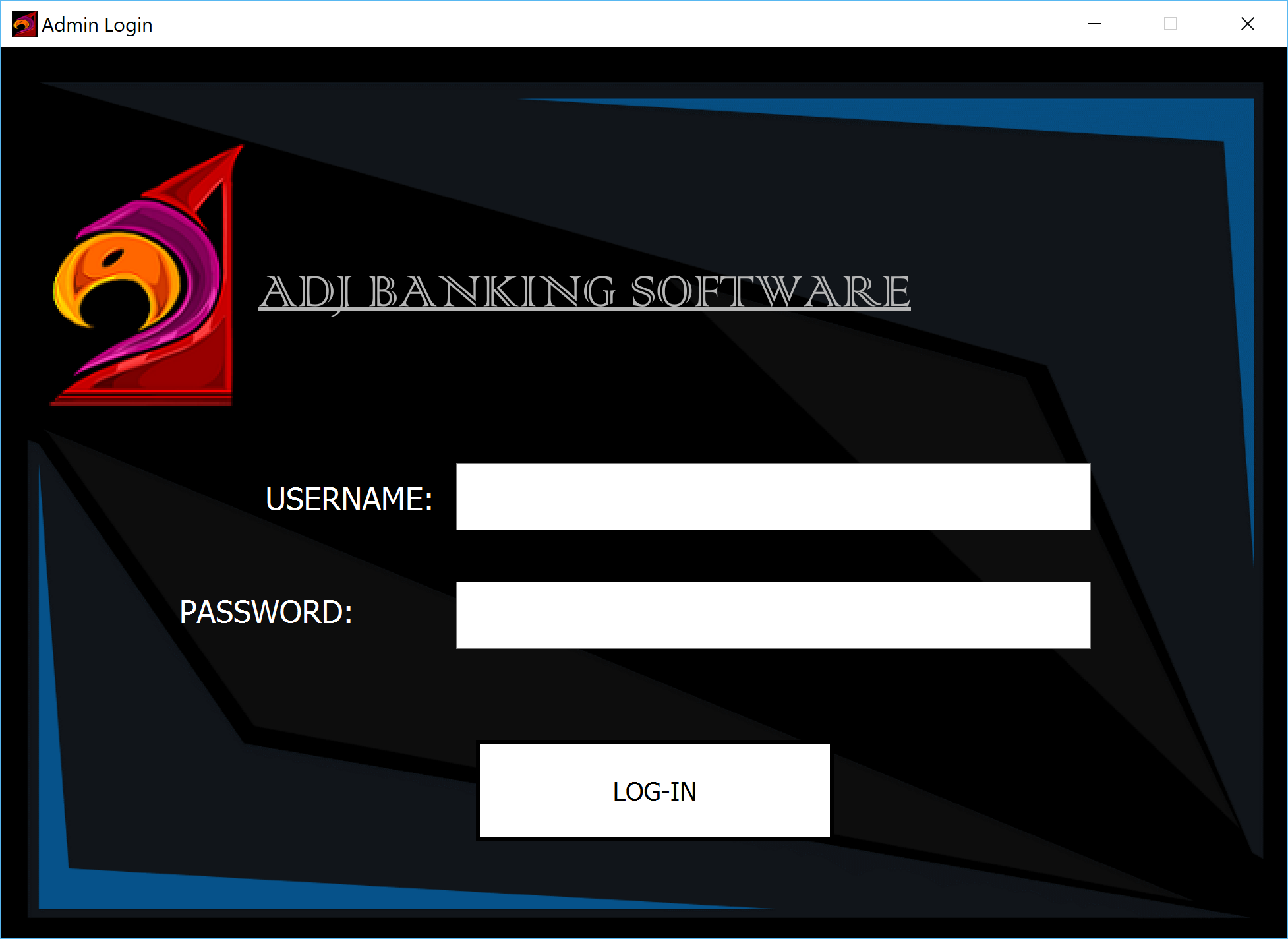
****

****

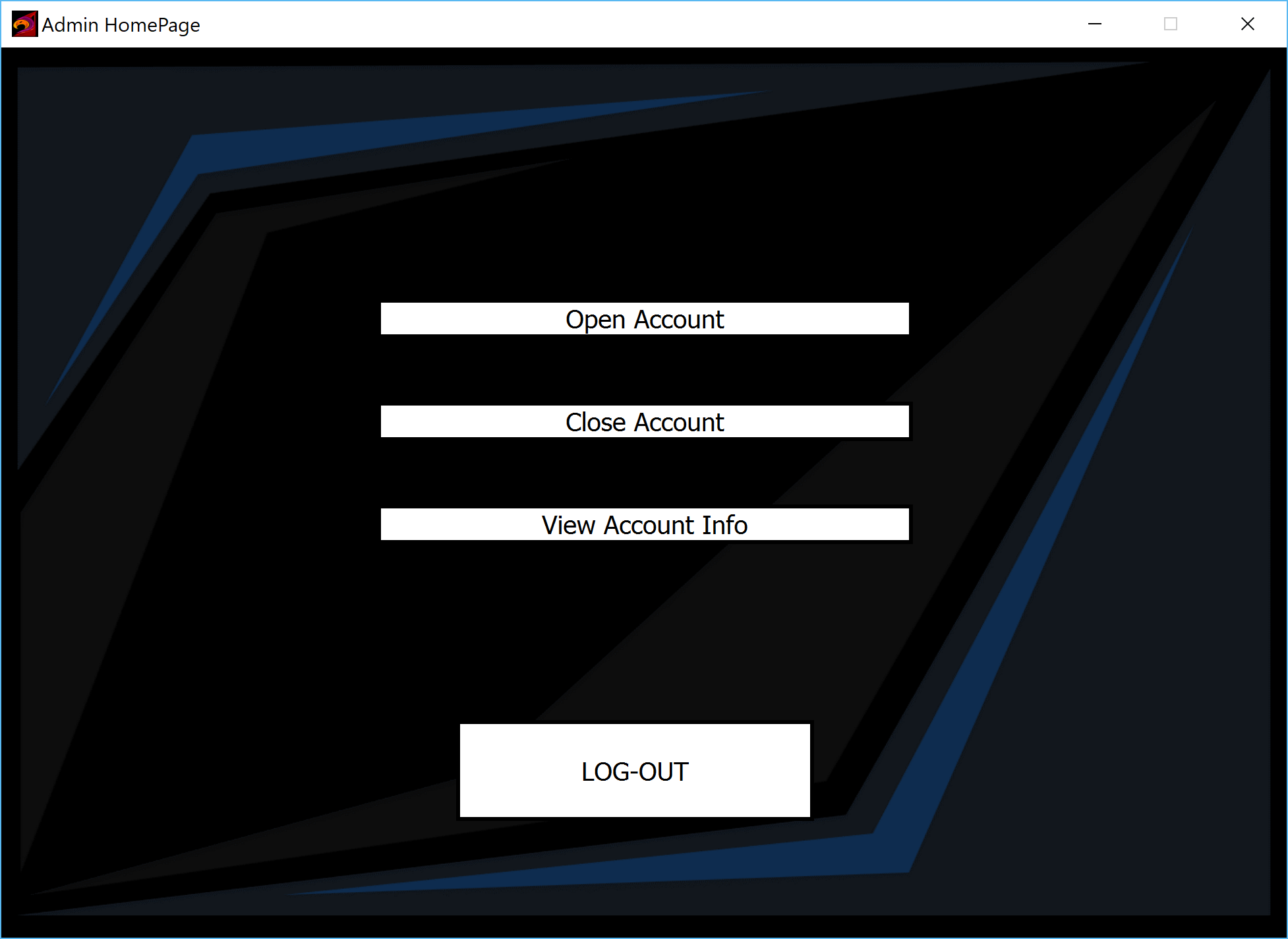
****

1. **Graphical User Interface**

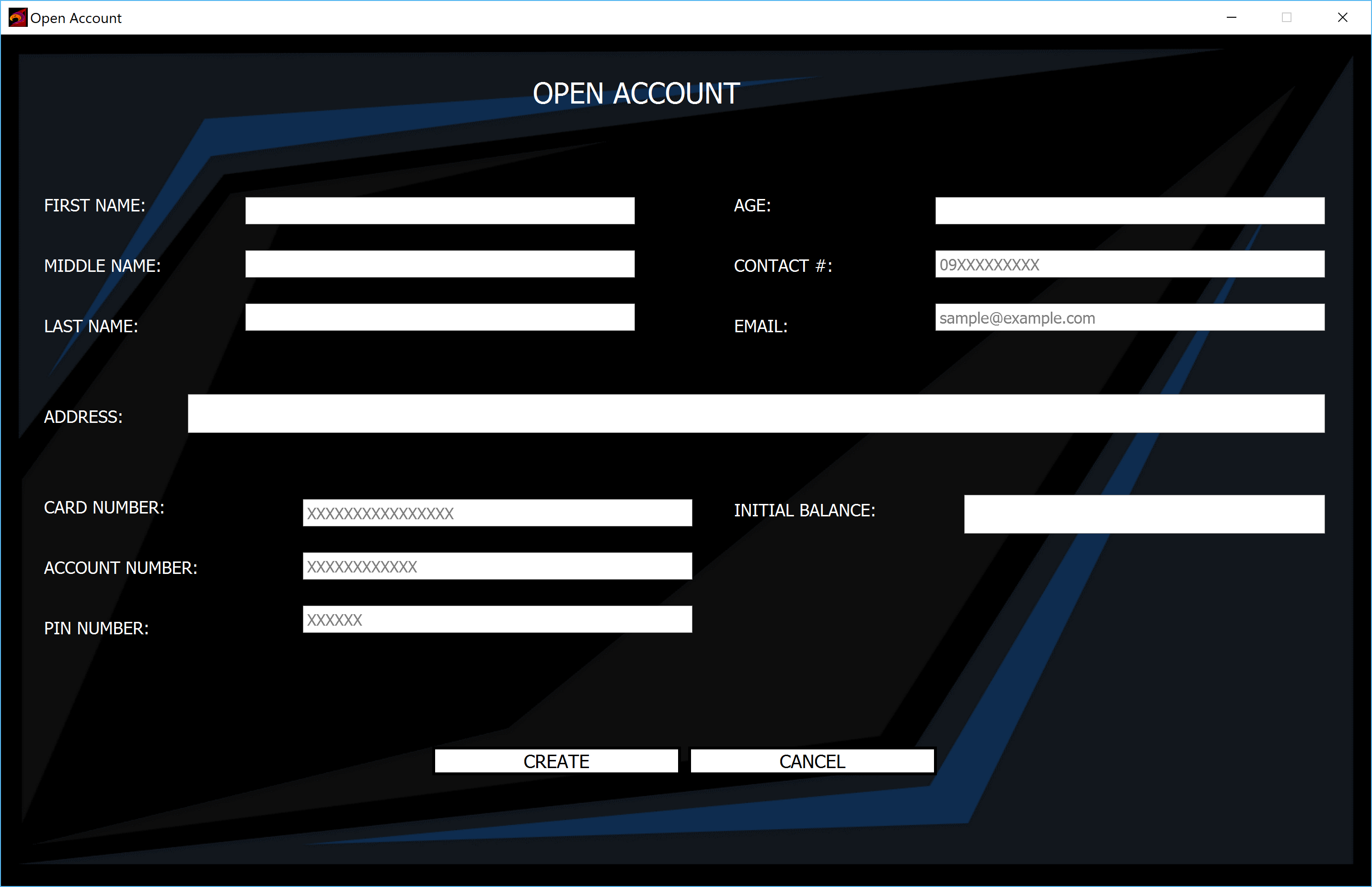
**Admin UI**



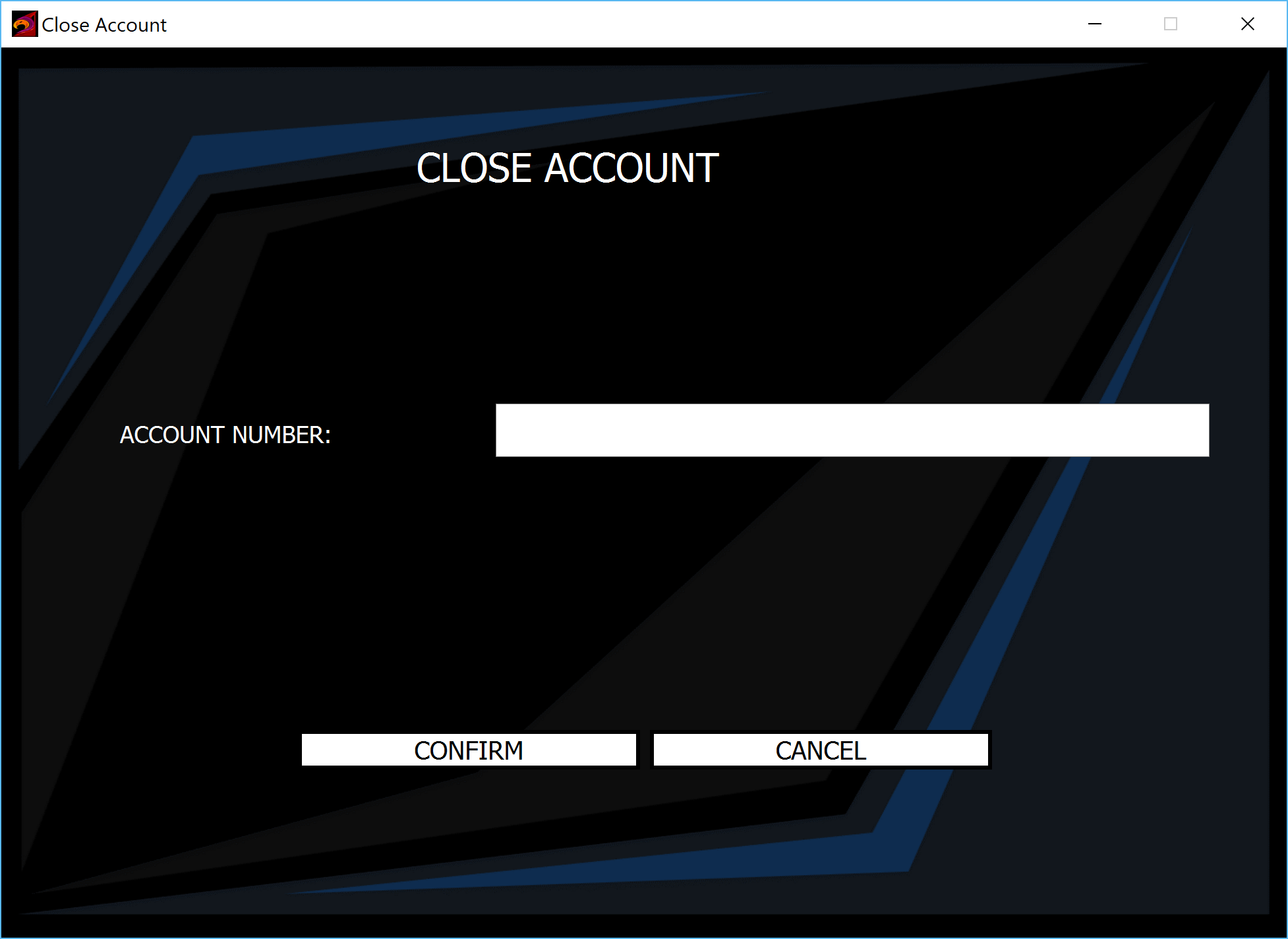
*Login Page (Administrator)*



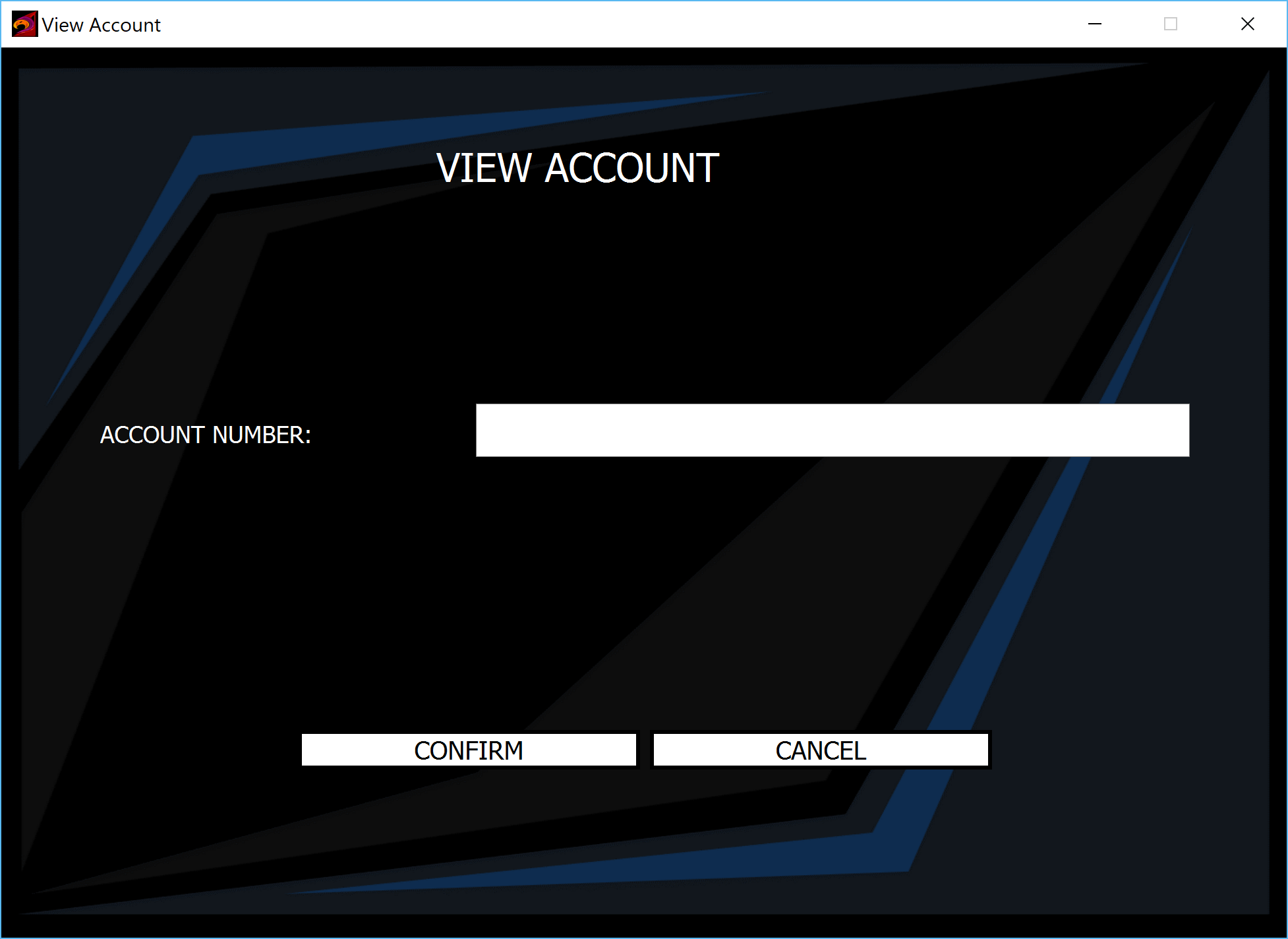
*Homepage (Administrator)*



*Open Account (Administrator)*

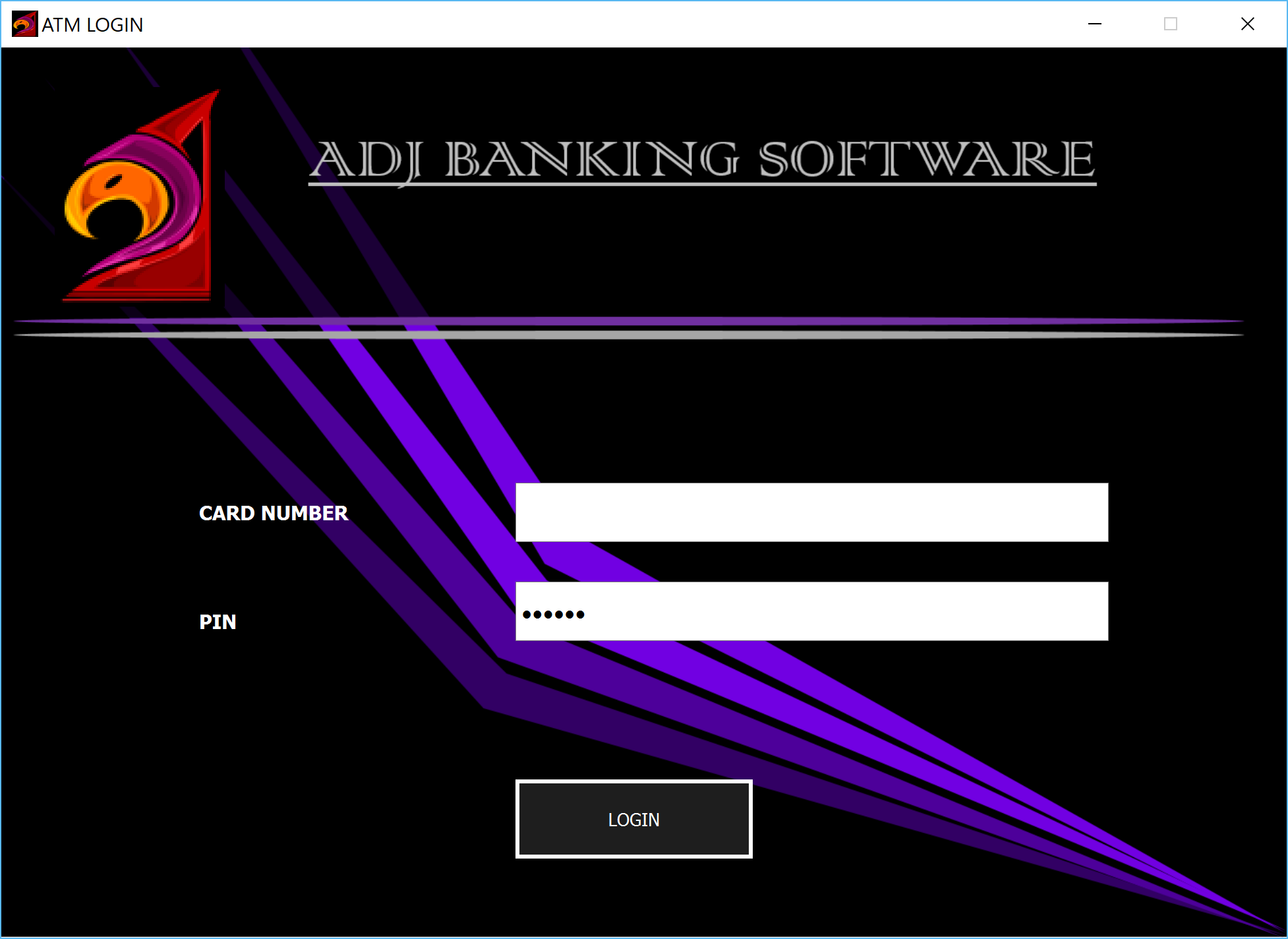


*Close Account (Administrator)*

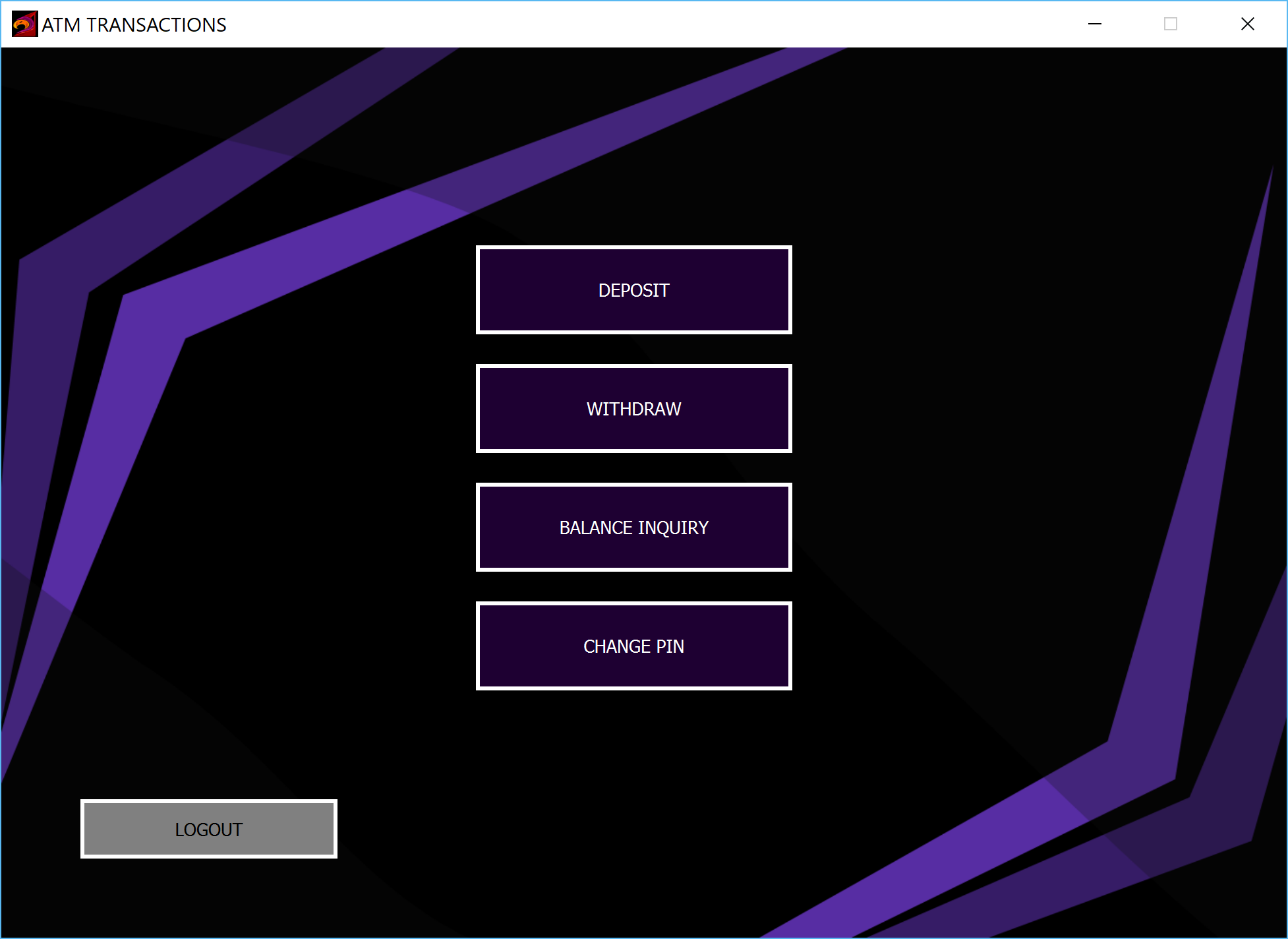


*View Account (Administrator)*

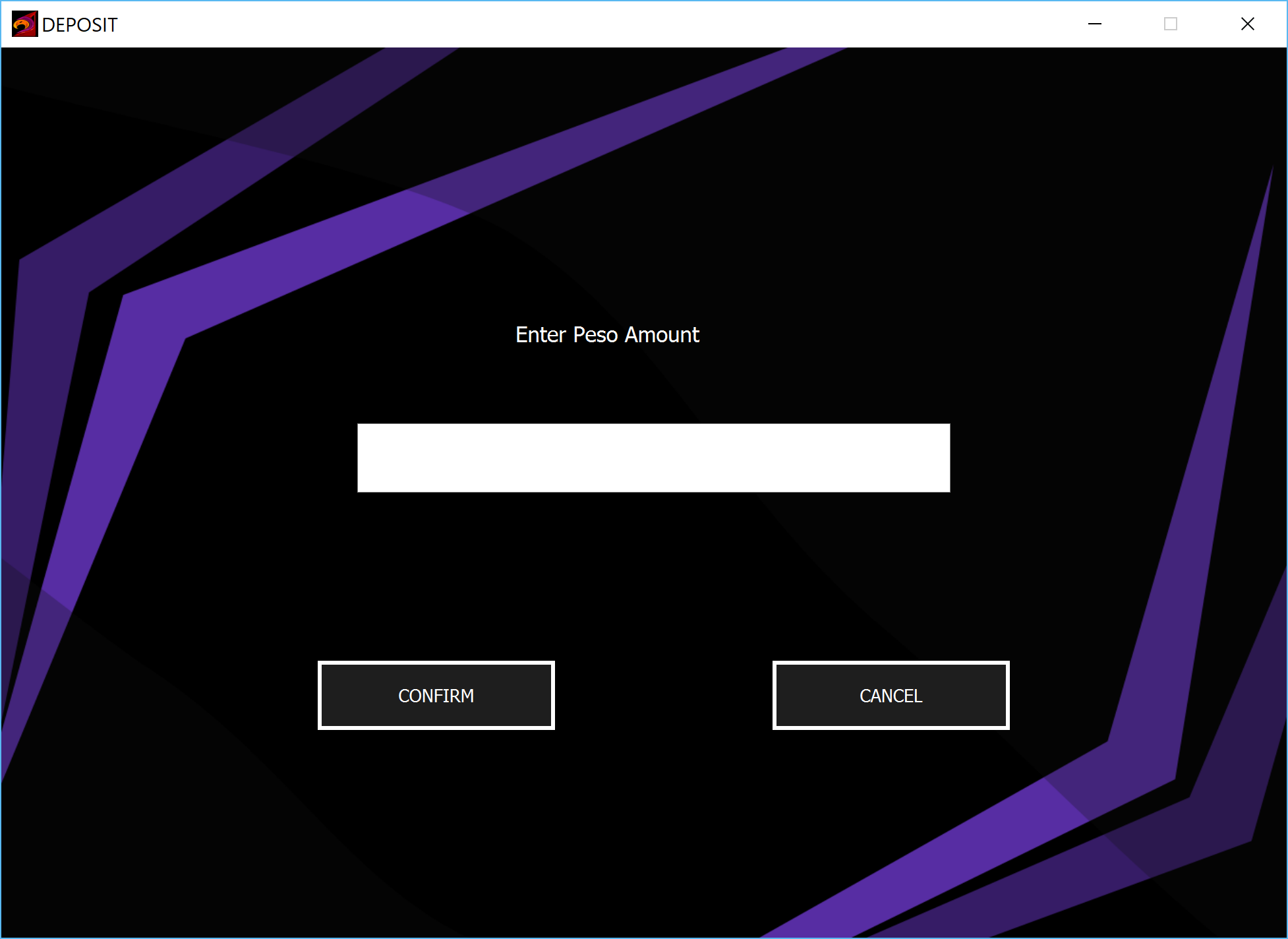
**Client UI**



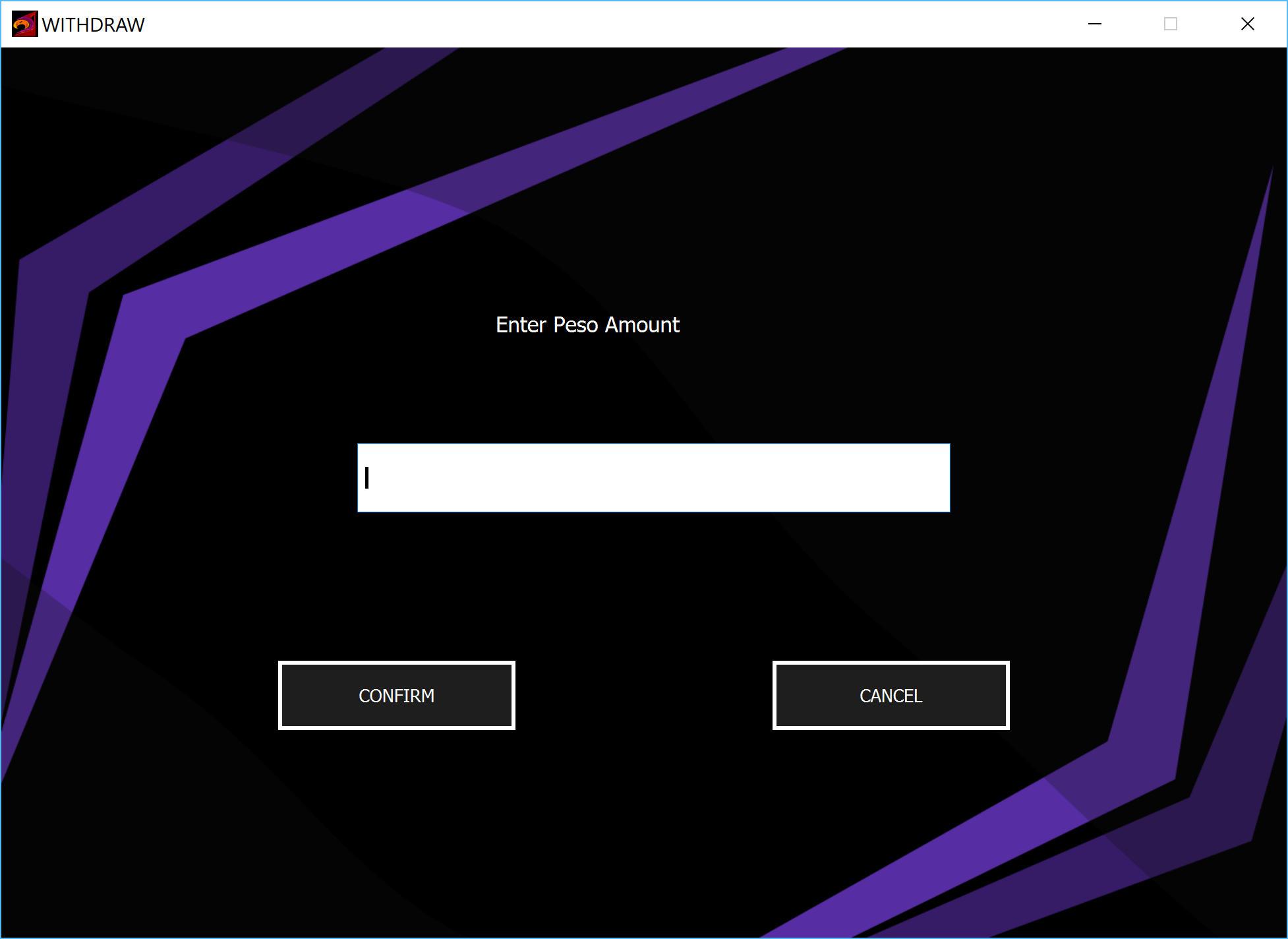
*Login Page (Client)*



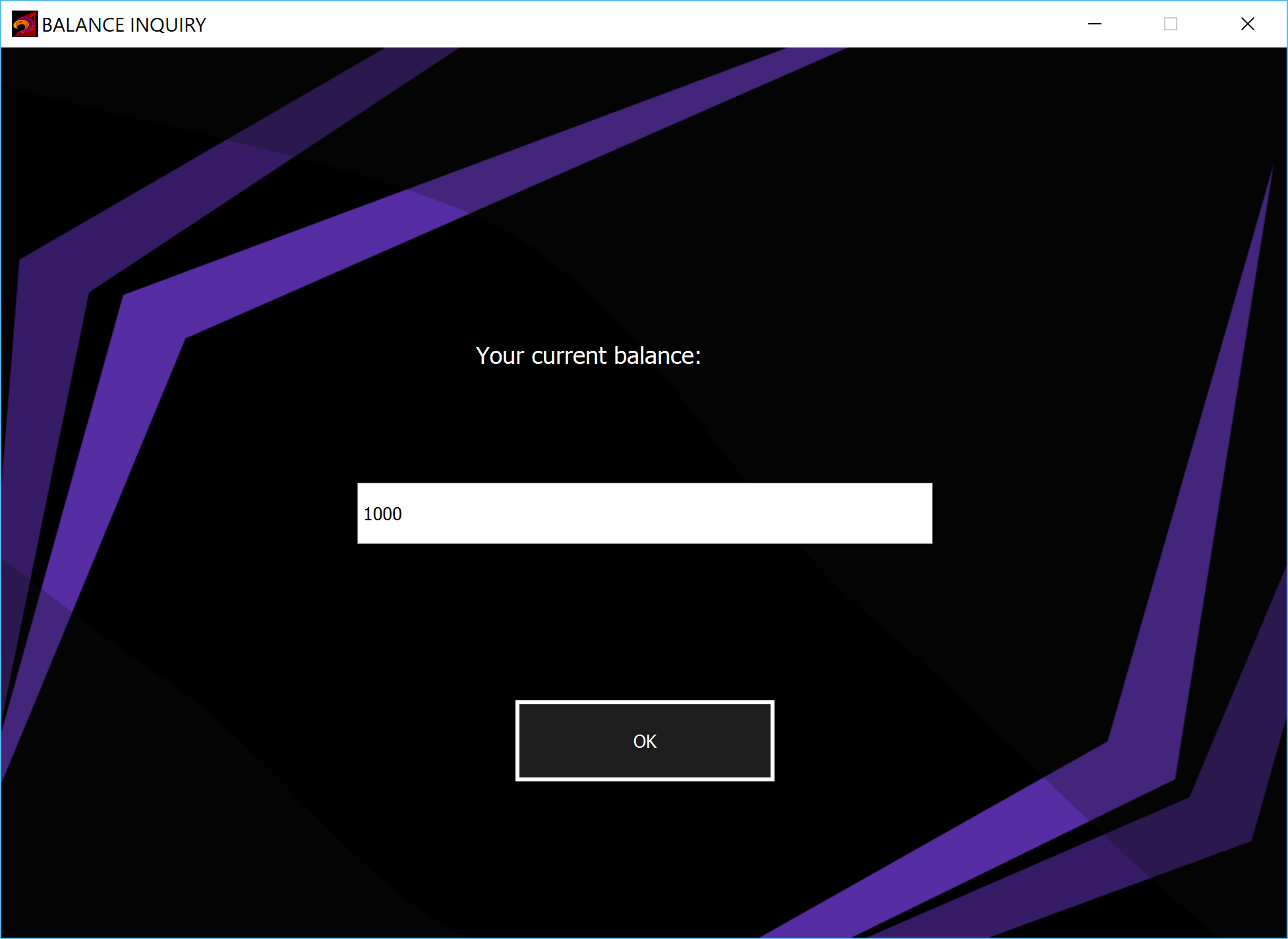
*Transactions (Client)*



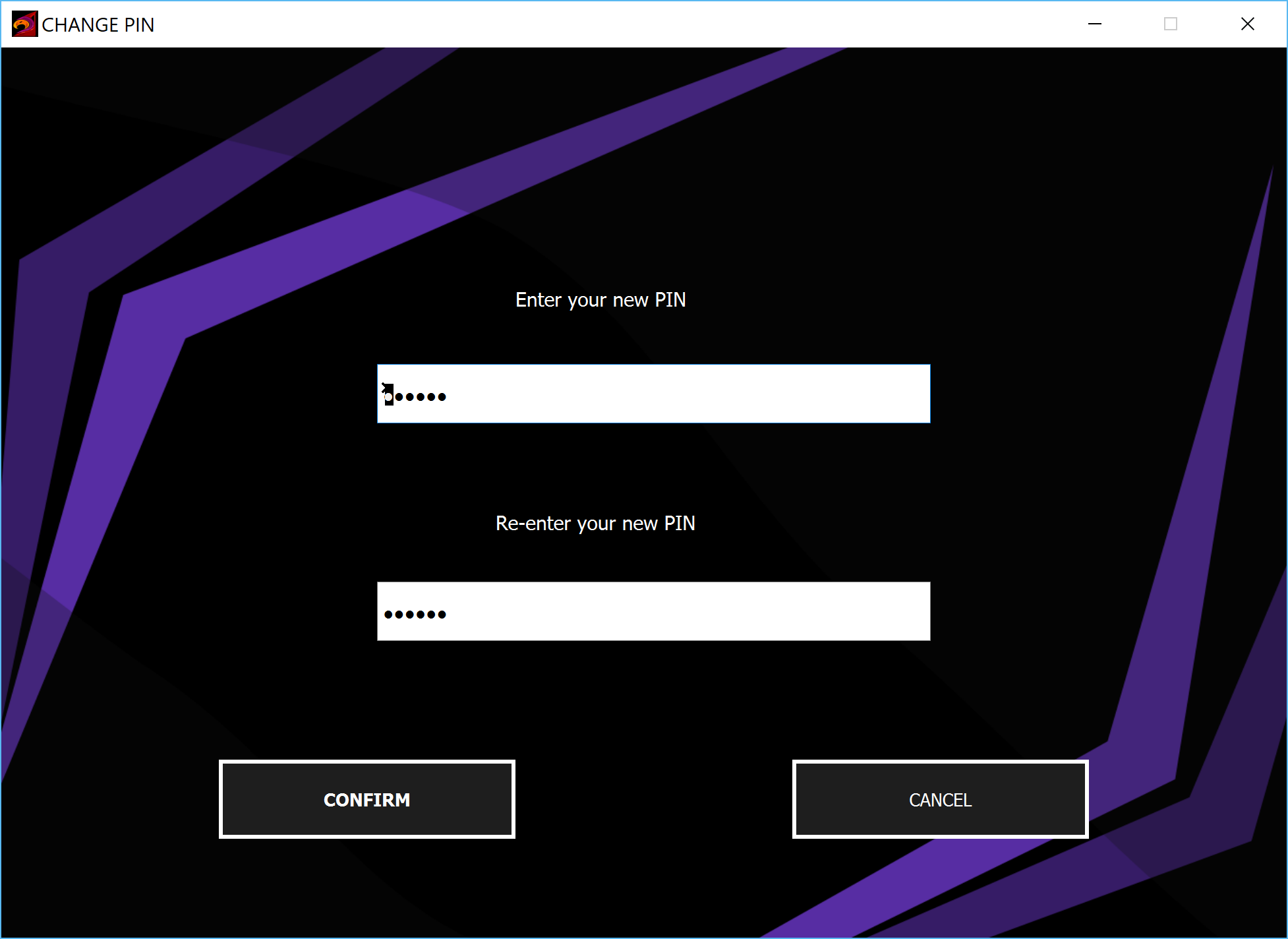
*Deposit (Client)*



*Withdraw (Client)*

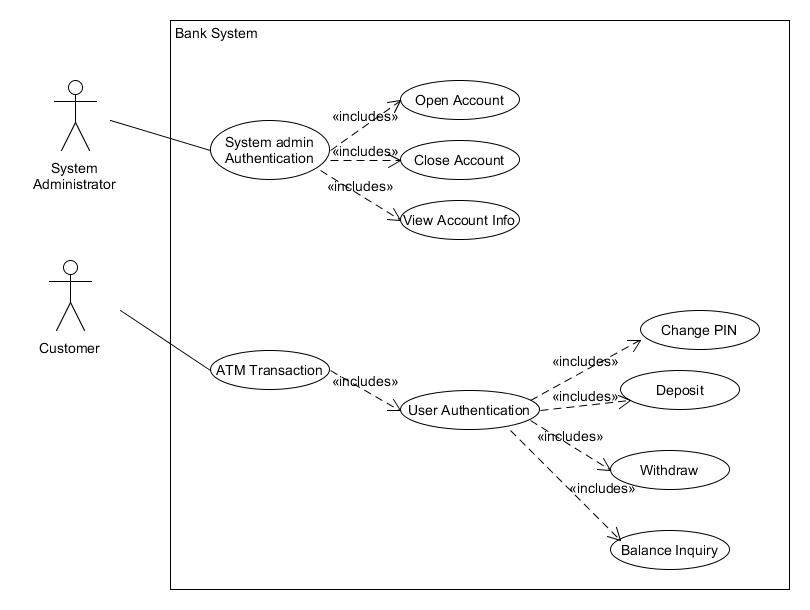


*Balance Inquiry (Client)*



*Change PIN (Client)*

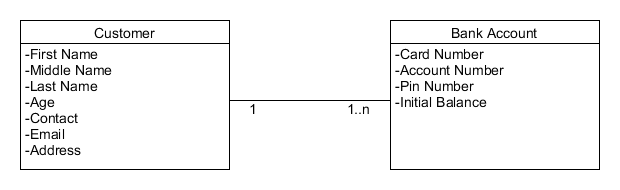
1. **Use Case Diagram**

****

The Use Case diagram above is the representation of the overall banking services of the software. It consists of two actors (system administrator, customer). Each actor own sets of functionalities represented by the ovals (use case). The system administrator is prompt for authentication before being able to open, close and view account info as well as the customer before giving the choices of transaction.

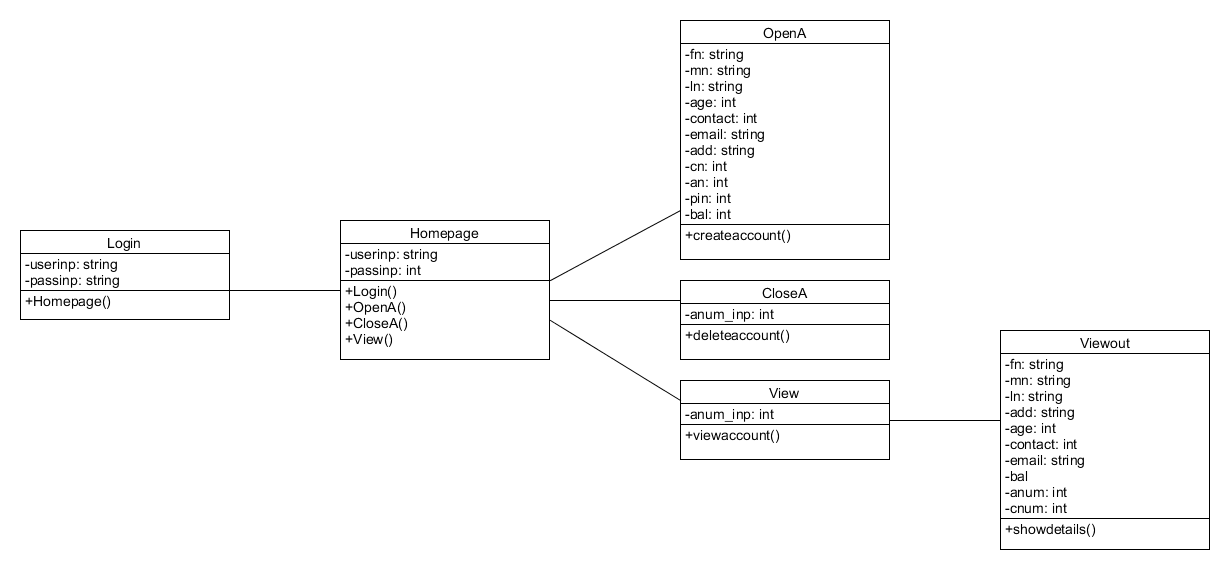
1. **Class Diagram**

**Conceptual**



The diagram above is in a business level type of representation. The following class names are only related to the real class names in the future codes are not necessarily the names to be implemented in the programming. Customer has the following attributes which will correspond to the attributes in the Bank Account.

**Detailed**

****

Detailed as the name suggest is comprised of additional contents in the class diagram. A class diagram composed of classes with attributes included with data types and methods. The administrator side has 4 methods( Login(), OpenA(), CloseA(), ViewA() that can be accessed only after authentication from the Login class.   
“A” represents accounts to minimize the name of the method to be used. Mostly the “string” and “integer” datatypes were needed for the whole diagram.

1. **Test Case Diagram**

ADMIN

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#1 – Admin Login | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_A1 | Authenticate System Administrator | System is on and displaying a login page | Input system admin username and password |  | System displays system administrator homepage |
| TC\_A2 | System rejects system admin username and/or password | System is on and displaying a login page | Input invalid system admin username and/or password |  | System displays an error screen and is ready to start a new session |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#2 – Creating an Account | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_B1 | System allows system admin to choose Create account option | The user must be logged-in as System Administrator | Choose Create account option |  | System displays Create account option session |
| TC\_B2 | Create non-existing account | Create account session is displayed | 1. Type-in user details  2. Type-in account details of non-existing account |  | System displays User account successfully created and goes back to homepage |
| TC\_B3 | Create existing account | Create account session is displayed | 1. Type-in user details  2. Type-in account details of existing account |  | System prompts User account already existed |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#3 – Closing an Account | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_C1 | System allows system admin to choose Close account option | The user must be logged-in as System Administrator | Choose Close account option |  | System displays Close account option session |
| TC\_C2 | Close existing account | Close account session is displayed | Enter valid account# |  | System displays User account successfully closed and goes back to homepage |
| TC\_C3 | Close non-existing account | Close account session is displayed | Enter invalid account# |  | System displays an error screen and is prompt to enter again |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#4 – View Account Info | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_D1 | System allows system admin to choose View Account Info | The user must be logged-in as System Administrator | Choose View Account Info option |  | System displays View Account Info session |
| TC\_D2 | View existing user account info | View Account Info session is displayed | Enter account# |  | System displays User account info successfully and goes back to homepage |
| TC\_D3 | View non-existing user account info | View Account Info session is displayed | Enter invalid account# |  | System displays an error screen and is prompt to enter again |

CLIENT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#5 – ATM Transaction (Login) | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_E1 | System authenticates the user | System is on and displaying a login page | Insert card (user card#) and PIN |  | System displays a menu of transaction types |
| TC\_E2 | System rejects invalid user card (card#) and/or PIN | System is on and displaying a login page | Insert invalid card (user card#) and/or PIN |  | System displays an error screen and is ready to start a new session |
| TC\_E3 | Customer exceeds max number of invalid PIN tries (3 entries) and system aborts session | System is on and displaying a login page | Enter 3rd try ofinvalid card (user card#) and/or PIN |  | System displays an error screen and session is terminated |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#6 – ATM Transaction (Deposit) | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_F1 | System allows customer to perform a Deposit transaction | Menu of transaction types is being displayed | Choose Deposit transaction |  | System displays the Deposit transaction session |
| TC\_F2 | System allows customer to enter a peso amount to deposit | System is displaying a request for the customer to type a peso amount | Enter a legitimate peso amount |  | System prints receipt showing amount and updated balance |
| TC\_F3 | System allows multiple transactions in one session | System is asking whether customer wants another transaction | Choose yes |  | System displays a menu of transaction types |
| TC\_F4 | Session ends when customer chooses not to do another transaction | System is asking whether customer wants another transaction | Choose no |  | System terminates and is ready to start a new session |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#7 – ATM Transaction (Withdrawal) | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_G1 | System allows customer to perform a Withdrawal transaction | Menu of transaction types is being displayed | Choose Withdrawal transaction |  | System displays the Withdrawal transaction session |
| TC\_G2 | System verifies that customer's balance is sufficient to fulfill the request | System is displaying a request for the customer to type a peso amount | Choose an amount that is not greater than the account balance |  | System prints receipt showing amount and updated balance |
| TC\_G3 | System verifies that customer's balance is insufficient to fulfill the request | System is displaying a request for the customer to type a peso amount | Choose an amount that is greater than the account balance |  | System displays an appropriate message and asks customer to choose a different amount |
| TC\_G4 | System verifies that customer's withdrawal amount is greater than per transaction limit amount (10000Php) | System is displaying a request for the customer to type a peso amount | Choose an amount that is greater than per transaction limit amount |  | System displays an appropriate message and asks customer to choose a different amount |
| TC\_G5 | System allows multiple transactions in one session | System is asking whether customer wants another transaction | Choose yes |  | System displays a menu of transaction types |
| TC\_G6 | Session ends when customer chooses not to do another transaction | System is asking whether customer wants another transaction | Choose no |  | System terminates and is ready to start a new session |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#8 – ATM Transaction (Balance Inquiry) | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_H1 | System allows customer to perform a Balance Inquiry transaction | Menu of transaction types is being displayed | Choose Balance Inquiry transaction |  | System prints receipt showing balance |
| TC\_H2 | System allows multiple transactions in one session | System is asking whether customer wants another transaction | Choose yes |  | System displays a menu of transaction types |
| TC\_H3 | Session ends when customer chooses not to do another transaction | System is asking whether customer wants another transaction | Choose no |  | System terminates and is ready to start a new session |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TC#9 – ATM Transaction (Change PIN) | | | | | |
| TC\_ID# | Test Objective | Precondition | Steps | Test Data | Expected Result |
| TC\_I1 | System allows customer to perform Change PIN | Menu of transaction types is being displayed | Choose Change PIN |  | System displays the Change PIN session |
| TC\_I2 | System verifies that customer's new PIN is valid | System is displaying a request for the customer to enter new PIN | Enter valid new PIN |  | System prompts the customer that current PIN is successfully changed |
| TC\_I3 | System verifies that customer's new PIN is the same as old PIN | System is displaying a request for the customer to enter new PIN | Enter invalid or old PIN |  | System displays an appropriate message and asks customer to choose a different PIN |
| TC\_I4 | System verifies that customer's new PIN is not the same with re-entered PIN | System is displaying a request for the customer to enter new PIN | Enter valid new PIN different with re-entered PIN |  | System displays an appropriate message and asks customer to re-enter the same PIN |
| TC\_I4 | System verifies that customer's new PIN is NULL | System is displaying a request for the customer to enter new PIN | Enter NULL PIN |  | System displays an appropriate message and asks customer to enter valid PIN. |
| TC\_I5 | System allows multiple transactions in one session | System is asking whether customer wants another transaction | Choose yes |  | System displays a menu of transaction types |
| TC\_I6 | Session ends when customer chooses not to do another transaction | System is asking whether customer wants another transaction | Choose no |  | System terminates and is ready to start a new session |

1. **Partial Codes**
2. **Sprint 1**

* main.py
* res.py
* admin\_log.py
* admin\_hp.py
* admin\_open.py
* admin\_close.py
* admin\_view.py
* admin\_viewout.py
* admin\_prompt.py
* admin\_prompt2.py
* admin\_error.py
* admin\_error2.py

1. **Sprint 2**

**Admin**

* SPLASH\_screen.py
* admin\_close.py
* admin\_error.py
* admin\_error2.py
* admin\_error3.py
* admin\_error4.py
* admin\_hp.py
* admin\_log.py
* admin\_open.py
* admin\_prompt.py
* admin\_prompt2.py
* admin\_view.py
* admin\_viewout.py
* data\_handler.py
* main.py
* reso.py
* test\_main.py

**Client**

* Atm\_Balance.py
* Atm\_ChangePin.py
* Atm\_ChangePin\_Blank.py
* Atm\_ChangePin\_Reenter.py
* Atm\_ChangePin\_Success.py
* Atm\_Deposit.py
* Atm\_Deposit\_Receipt.py
* Atm\_Login.py
* Atm\_Login\_Error.py
* Atm\_Login\_Error\_Max.py
* Atm\_Main.py
* Atm\_NewTransaction.py
* Atm\_Transactions.py
* Atm\_Withdraw.py
* Atm\_Withdraw\_Error\_Exceed.py
* Atm\_Withdraw\_Error\_Insuff.py
* Atm\_Withdraw\_Receipt.py
* SPLASH\_screen.py
* client\_data\_handles.py
* res1.py
* reso.py

1. **Sprint 3**
2. **Errors Encountered**

|  |  |
| --- | --- |
| **Date** | **Errors** |
| 9/21/18 | No errors encountered yet, the task was recently distributed. |
| 9/22/18 | Two account types (Checking and Savings) were initially mentioned but was corrected in the day 2 of sprint 1 to be savings account only.  The mockups lack display for error messages.  The team noticed that it is unnecessary to add the pin number in order for the admin to authenticate before being able to close an account. |
| 9/23/18 | Some of the documents were only uploaded in the github and not on the file exchange. |
| 9/24/18 | Accessing data from the csv file took several trial and errors but was accomplished later on. |
| 9/25/18 | The program was not running on the cmd when the created database was incorporated with the program. |
| 9/26/18 | There is no button that signifies the confirm action when the user is asked to try again which leaves the user to rely on the close button. The burn down chart must be set in such a way that the last connecting line in the graph falls into zero. |
| 9/27/18 | There is a problem with the program flow linking on the deposit transaction after a successful login. |
| 9/28/18 | Allowing the ESC button to be an alternative for the closing of error displayed message is unsuccessful. |
| 9/29/18 | Few trial and errors were taken in order to correct the way on how the deposit and withdraw function should work such as adding and subtracting respectively and updating it into the csv file.  Changing the background using the designer platform chosen takes a bit of work and needed caution, otherwise, the background design would not appear in the window. |
| 9/30/18 | There was an error noticed on fetching the current pin of the client as well as overwriting it in the csv file.  Due to the lack of information on the internet with the designer that is currently being utilized by the team regarding the splash screen makes the member unable to decide whether to include the mentioned feature. |
| 10/1/18 | Errors encountered were linking all client functions with one another. Some functions were not able to result with the desired function linked them.  An error with regards to the deposit and withdraw functions were noticed after a running the program. It crashes when there is no amount placed by the user on the text field. |
| 10/2/18 | There were times when implementing the program unit testing that there is an incorrect usage of assert methods to check for and report failures.  There was a piece of code related on the splash screen which ends the whole program. |

1. **Recommendation**
2. **Conclusion**