Table of Contents

[Login page 3](#_Toc72683201)

[Login page validations 3](#_Toc72683202)

[Create new bank account 4](#_Toc72683203)

[Create bank account validations 4](#_Toc72683204)

[Search for an account 5](#_Toc72683205)

[Search account validations 5](#_Toc72683206)

[Update and delete account 6](#_Toc72683207)

[View all accounts 7](#_Toc72683208)

[Deposit money 7](#_Toc72683209)

[Deposit money validations 8](#_Toc72683210)

[Withdraw Money 8](#_Toc72683211)

[Withdraw money validations 9](#_Toc72683212)

[View transaction details 10](#_Toc72683213)

[Manager and cashier permissions 10](#_Toc72683214)

**Table of Figures**

[Figure 1:Empty username or password 3](#_Toc72683215)

[Figure 2: Wrong credential message 3](#_Toc72683216)

[Figure 3: Create new account 4](#_Toc72683217)

[Figure 4: Information required warning message 4](#_Toc72683218)

[Figure 5: Search account 5](#_Toc72683219)

[Figure 6: Empty Account Number 5](#_Toc72683220)

[Figure 7: Account not Found 6](#_Toc72683221)

[Figure 8: Modify account information 6](#_Toc72683222)

[Figure 9: Delete an account 7](#_Toc72683223)

[Figure 10: All bank accounts 7](#_Toc72683224)

[Figure 11:Deposit money 8](#_Toc72683225)

[Figure 12: Account not found to deposit money 8](#_Toc72683226)

[Figure 13: Withdraw money 9](#_Toc72683227)

[Figure 14: Account not found to deposit money 9](#_Toc72683228)

[Figure 15: Insufficient balance message 9](#_Toc72683229)

[Figure 16: View all transactions 10](#_Toc72683230)

[Figure 17: View transactions done by an account 10](#_Toc72683231)

[Figure 18: Manager view 11](#_Toc72683232)

[Figure 19: Cashier view 11](#_Toc72683233)

[Figure 20: Cashier accounts 12](#_Toc72683234)

[Figure 21: Search accounts 13](#_Toc72683235)

[Figure 22: Cashier account not found 13](#_Toc72683236)

# Login page

Whenever a user clicks the “Login” button in the user interface the system will access the checkPassword(String username, String password) function of the UserDAO class through the btnContActionPerformed(java.awt.event.ActionEvent evt) function of WelcomeScreen class. checkPassword(String username, String password) function will look for the that the user has provided credentials in AccountUsers.txt.

## Login page validations

If username or password is empty, it will ask the user to fill the required information.

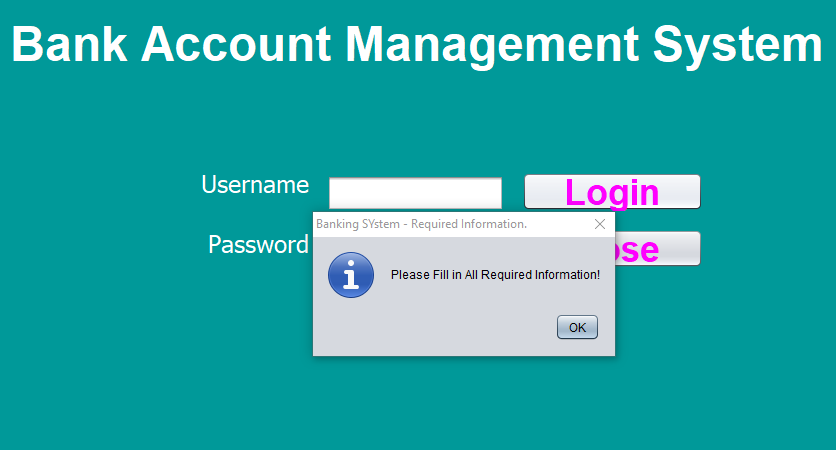


Figure 1:Empty username or password

If the user enters an incorrect username or password, the system will display the following error.

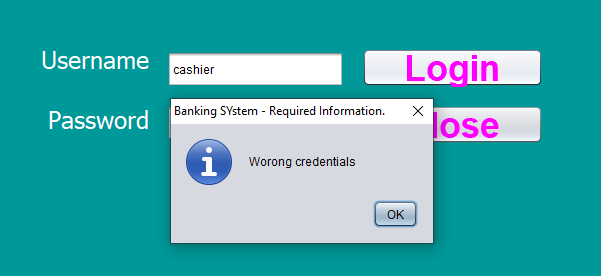


Figure 2: Wrong credential message

# Create new bank account

After inserting all the required information, the manager can click “Open” button to open a new bank account. When the user clicks the Open button the system will access the createAccount(Account account) function in AccountDAO class through the btnOpenAcctActionPerformed(java.awt.event.ActionEvent evt) function of the ApplicationMainMenu class. Also, createAccount(Account account) will take account object as a parameter.

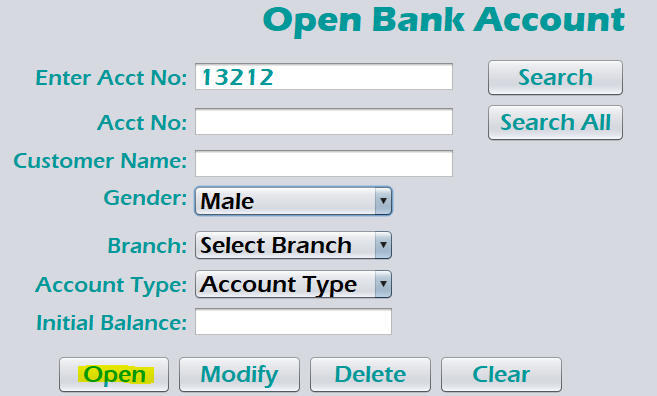


Figure 3: Create new account

## Create bank account validations

All the required information should be provided to create an account.

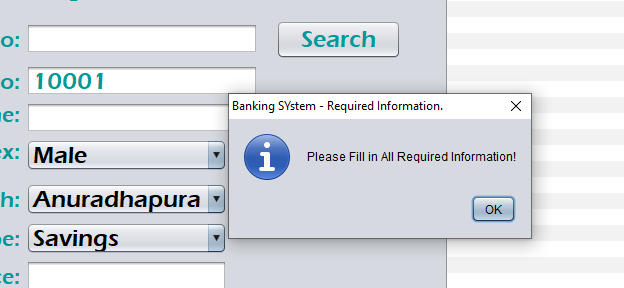


Figure 4: Information required warning message

# Search for an account

Search button can be used to search for an account. When the user clicks the “Search” button the system will access the selectAccount(String selectedAccount) function of AccountDAO class through the btnSearchAcctActionPerformed(java.awt.event.ActionEvent evt) function of the ApplicationMainMenu class. selectAccount(String selectedAccount) function will return Account object.

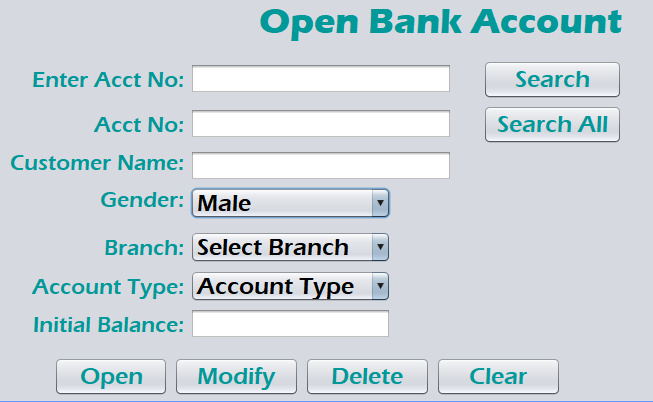


Figure 5: Search account

## Search account validations

If the user does not enter an account number to search, the system will display the following message.

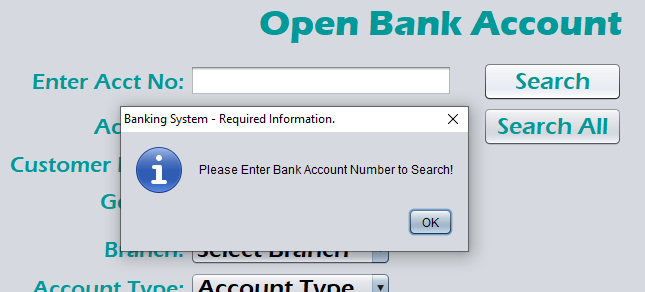


Figure 6: Empty Account Number

If the user enters a wrong account number, the system will display “Account not found” message.

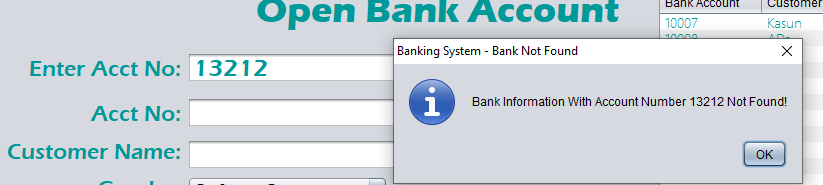


Figure 7: Account not Found

# Update and delete account

After searching for an account, the user can modify the account information. When the user clicks the “Modify” button, the system access the updateAccount(Account account) function of AccountDAO class using the of btnModifyAcctActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class.

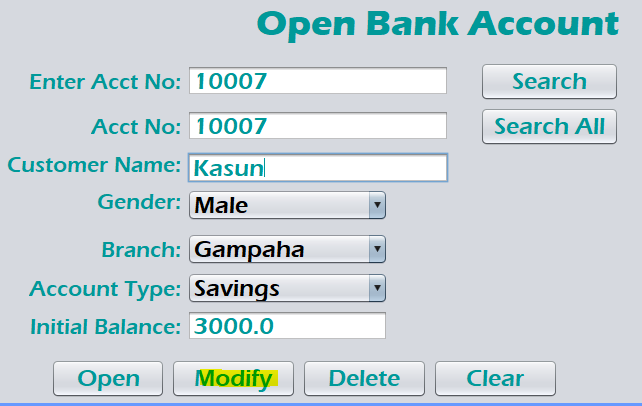
. 

Figure 8: Modify account information

After searching for an account, the manager can delete the selected account. When the user clicks the “Delete” button, the system access the deleteAccount(Account account) function of AccountDAO class using the of btnDeleteAcctActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class.

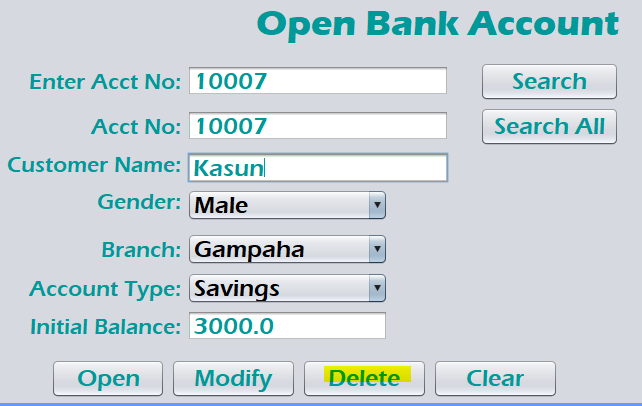


Figure 9: Delete an account

# View all accounts

All the accounts created can be seen in the right side of the application window. When the user clicks the “Search All” button, the system access the viewAllAccounts() function of AccountDAO class using the of btnSearchAcct1ActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class.

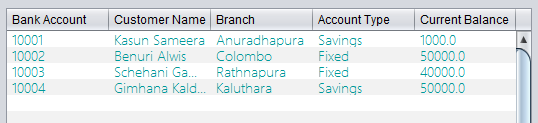


Figure 10: All bank accounts

# Deposit money

Click deposit money from the navigation pane to navigate to deposit money window. An account must be selected before depositing money therefore, search for an account and add the depositing amount. When the user clicks the “Deposit” button, the system access the makeTransaction(Transaction transaction) function of TransactionDAO class using the of btnDepositActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class.

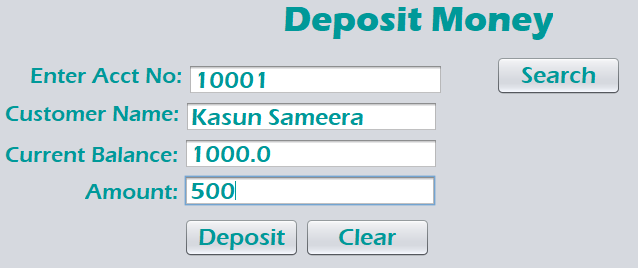


Figure 11:Deposit money

## Deposit money validations

If the user enters a wrong account number, the system will display account not found message.

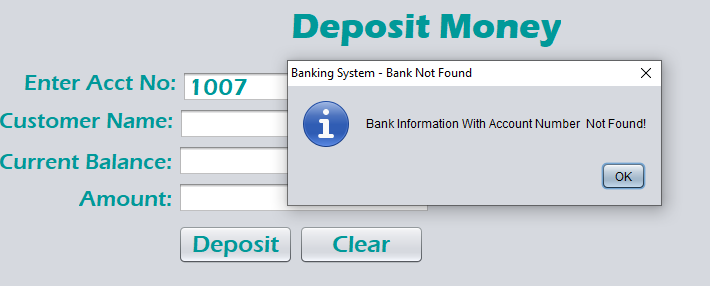


Figure 12: Account not found to deposit money

# Withdraw Money

Click withdraw money from the navigation pane to navigate to withdraw money window. An account must be selected before withdrawing money therefore, search for an account and add the withdraw amount. When the user clicks the “Withdraw” button, the system access the makeTransaction(Transaction transaction) function of TransactionDAO class using the of btnWithdrawActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class.

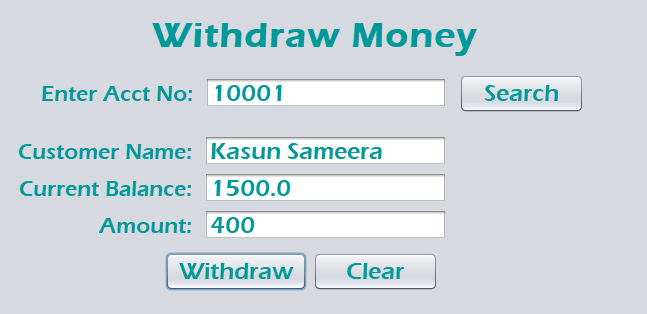


Figure 13: Withdraw money

## Withdraw money validations

If the user enters a wrong account number to withdraw money, the system will display account not found message.

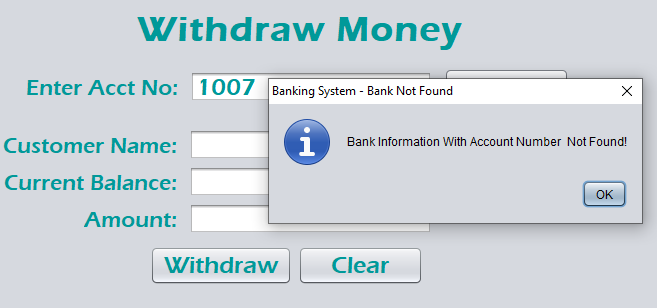


Figure 14: Account not found to deposit money

If the user tris to withdraw an amount more than the current account balance the system will pop up an error massage.

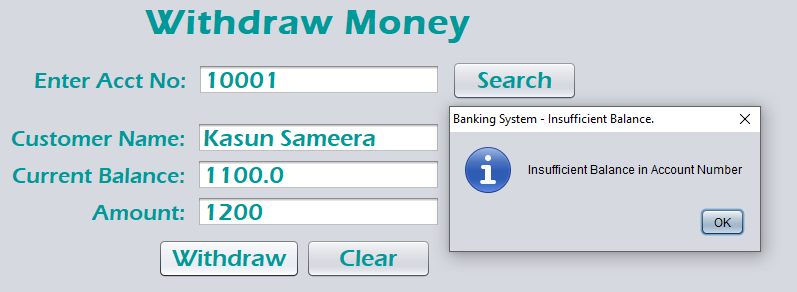


Figure 15: Insufficient balance message

# View transaction details

Navigate to “Bank Transaction” section to view details about the transactions done. Click on “View All” button to view all the transactions done within the day. When the user clicks the “View All” button, the system access the viewAllTransactions() function of TransactionDAO class using the of btnViewAllActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, viewAllTransactions() function will return a list of transactions.

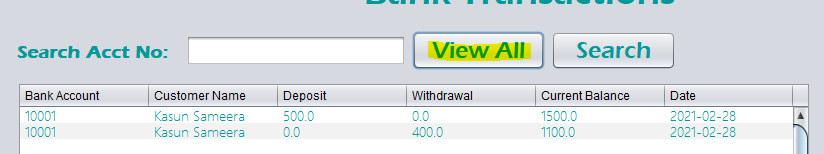


Figure 16: View all transactions

Enter an account number on the search bar and click “Search” button to get all transactions done from a selected account. When the user clicks the “Search” button, the system access the selectTransactionsById(String selectedAccount) function of TransactionDAO class using the of btnViewAllActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. selectTransactionsById(String selectedAccount) will take the account number as a parameter and return all the transactions done by that account

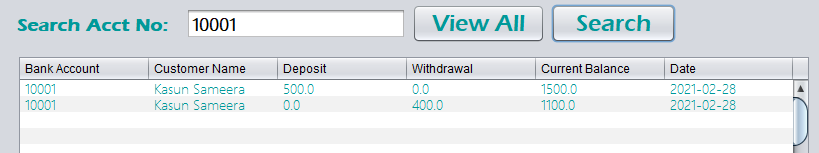


Figure 17: View transactions done by an account

# Manager and cashier permissions

Cashier has all the permissions same as the manager except creating and deleting bank accounts. Therefore, when a user continues as a cashier, create account and delete account buttons are not visible.

When the manager clicks the “Create” button in the user interface the system will access the createUser(User user) function of the UserDAO class through the btnCreateUserActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, createUser(User user) function will take cashier object as a parameter and create a new account in the AccountUsers.txt.

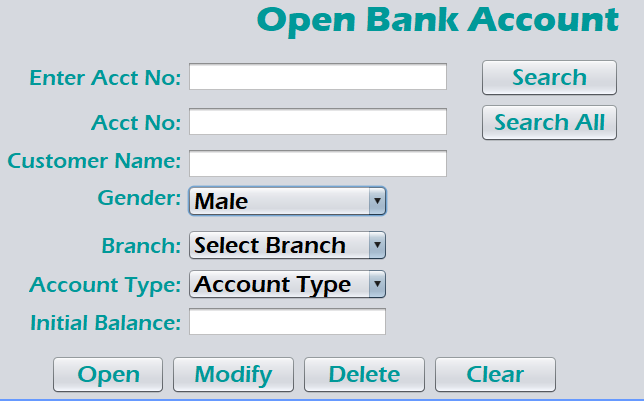


Figure 18: Manager view

When the manager/ cashier clicks the “Modify” button in the user interface the system will access the updateCashier(User cashier) function of the UserDAO class through the btnModifyUserActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, updateCashier(User cashier) function will take cashier object as a parameter and update the selected account in the AccountUsers.txt.

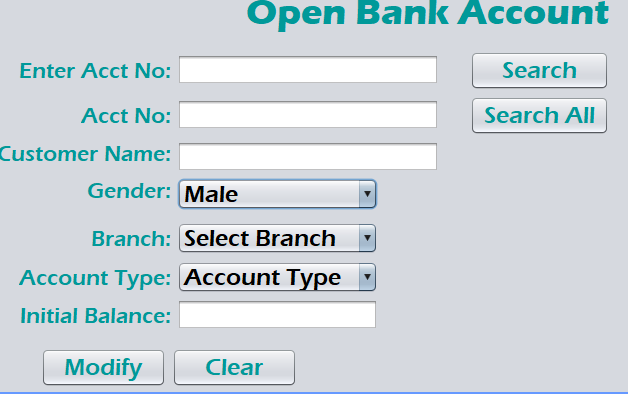


Figure 19: Cashier view

Manager can create, update, delete, search and view all cashier accounts casher accounts. When the manager clicks the “Delete” button in the user interface the system will access the deleteAccount(String username) function of the UserDAO class through the btnDeleteUserActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, deleteAccount(String username) function will take the username of a cashier as a parameter and update the selected account in the AccountUsers.txt.

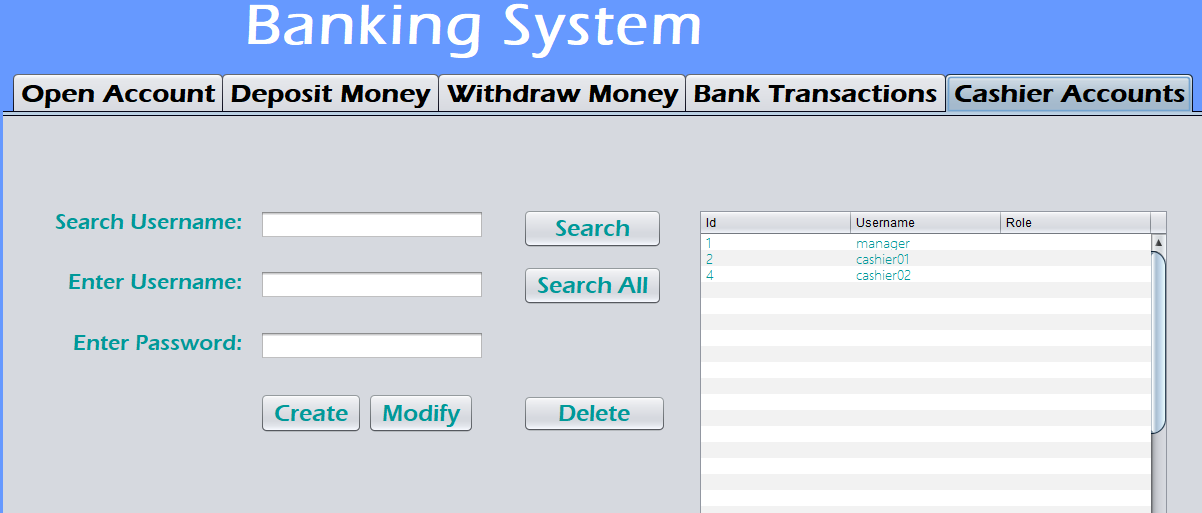


Figure 20: Cashier accounts

Both the manager and the cashiers can search for accounts. When the manager/ cashier clicks the “Search” button in the user interface the system will access the selectAccount(String username) function of the UserDAO class through the btnSearchAcct3ActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, selectAccount(String username) function will take cashier username as a parameter and will return a cashier object.

When the manager/ cashier clicks the “Search All” button in the user interface the system will access the viewAllUsers() function of the UserDAO class through the btnSearchAcct2ActionPerformed(java.awt.event.ActionEvent evt) function of ApplicationMainMenu class. Also, viewAllUsers() function will return a list of users form the AccountUsers.txt

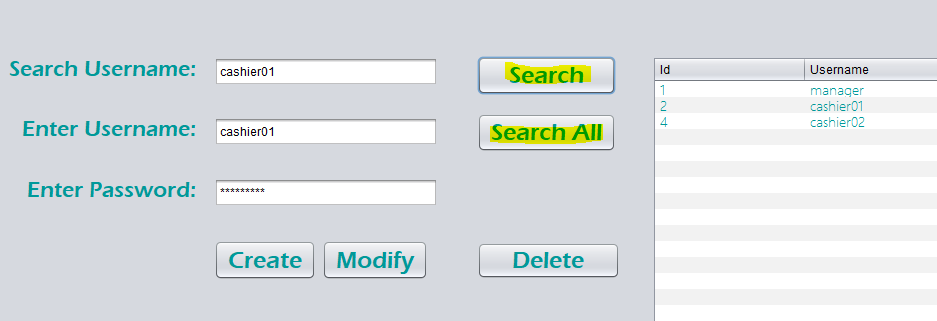


Figure 21: Search accounts

“Search All” button will fetch all the account records to the table. However, “Search” button will only fetch the details of a specific account. If the cashier account does not exist, the system will display the following error.

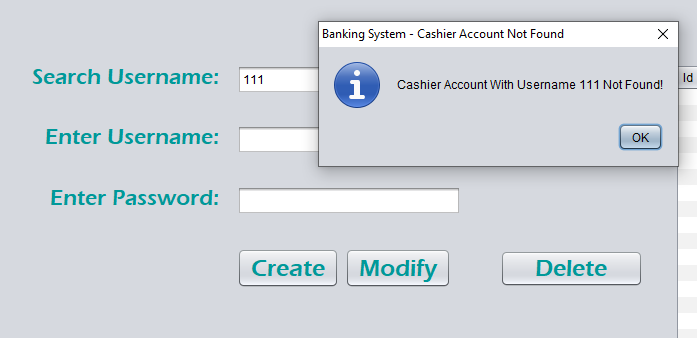


Figure 22: Cashier account not found