**Bank Balance and Dispensing System**

1. **Introduction**

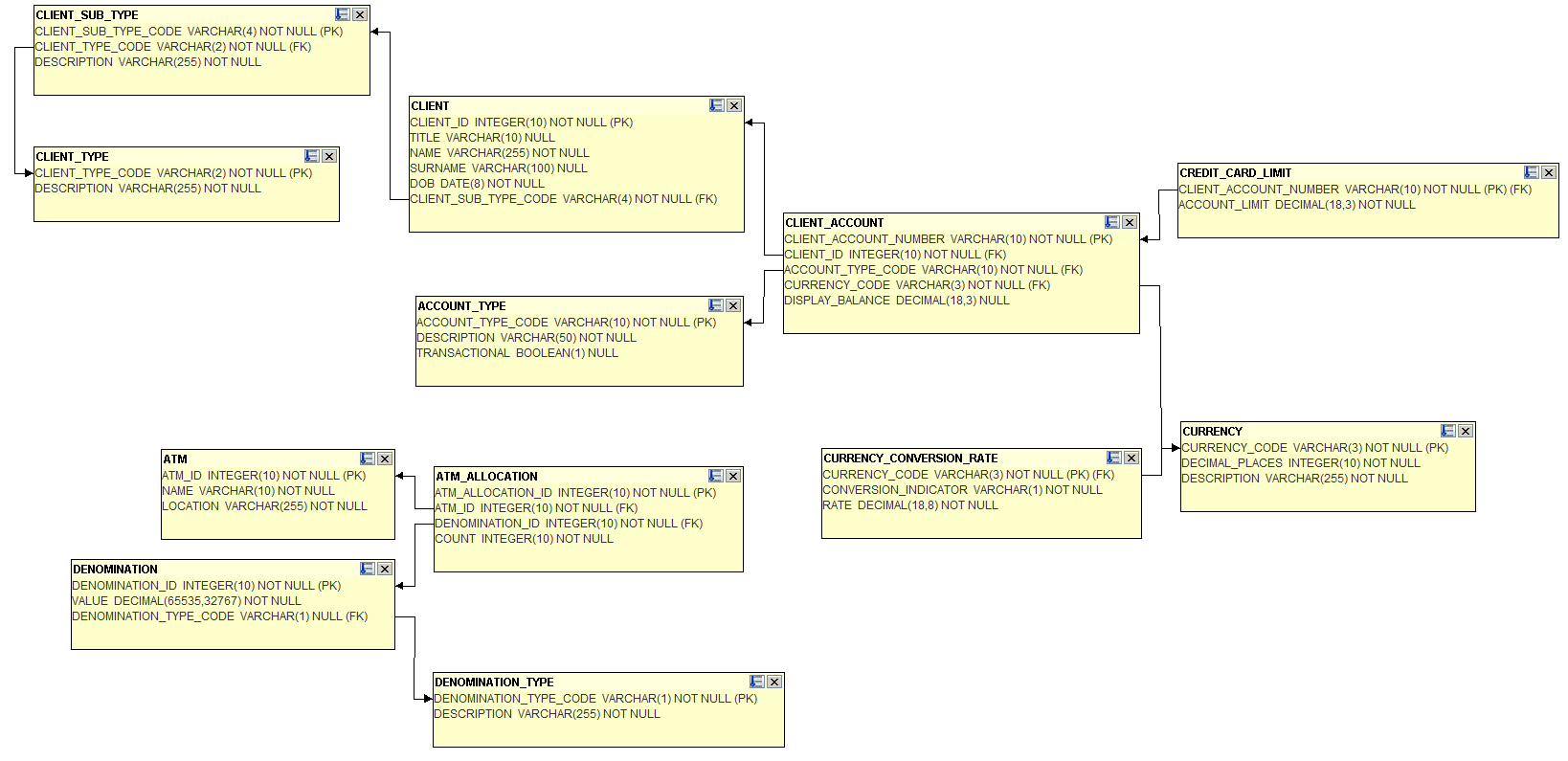
The goal of the Bank Balance and Dispensing System is to calculate and display the financial position to a client on an ATM screen. In addition, a client must also be able to make a request for a cash withdrawal.

1. **Scope**

The scope of the project is as follows:

|  |  |
| --- | --- |
| **In Scope** | **Out of Scope** |
| Display transactional accounts with balances | Login / Authentication |
| Display currency accounts with converted Rand values | User interface |
| Withdraw (Calculation of notes to be dispensed) | Any points marked as out of scope |
| Reporting – Find the transactional account per client with the highest balance |  |
| Reporting – Calculate aggregate financial position per client |  |
| A readme document that outlines any special instructions that should be considered when reviewing the assignment.  If standard ports and JDBC URLs are used and no custom tasks need to be executed, this can be omitted. |  |

1. **ERD**



1. **Requirements**
   1. **Use Case Summary**
   2. **Use Cases**
      1. **Display transactional accounts with balances**

|  |  |
| --- | --- |
| Use case | Display transactional balances |
| Goal in context | To view available balances |
| Level | User |
| Primary Actor | Bank client at ATM terminal |
| Precondition | Client successfully logged on to ATM terminal with bank card |
| Success end condition | Client is able to view all transactional accounts with the available balances on each account in descending order with the highest balance displaying first and the lowest balance displaying last |
| Basic flow of events | 1. Client logs in to ATM terminal using the card and pin pad and selects the “View Transactional Balances” option (out of scope). 2. The ATM terminal sends a request to the server with the client id and a timestamp (**out of scope)** 3. The server receives and interprets the request 4. The server determines the current display balances, sorts the details and returns the response to the ATM terminal 5. The ATM terminal receives and interprets the response from the server (out of scope) 6. The ATM terminal displays the balances to the user **(out of scope)** |
| Extension | 1. The client does not have any qualifying accounts The system displays an error message |
| Error Message | No accounts to display |
| Note | The UI needs to be aware of how to format the amount |

* + 1. **Display currency accounts with converted Rand values**

|  |  |
| --- | --- |
| Use case | Display currency account balances with converted Rand values |
| Goal in context | To view available balances of currency accounts with converted Rand values |
| Level | User |
| Primary Actor | Bank client at ATM terminal |
| Precondition | Client successfully logged on to ATM terminal with bank card |
| Success end condition | Client is able to view all transactional accounts with the available balances on each account in ascending order with the highest balance in Rand displaying first and the lowest balance displaying last |
| Basic flow of events | 1. Client logs in to ATM terminal using the card and pin pad and selects the “View Currency Account Balances” option **(out of scope)**. 2. The ATM terminal sends a request to the server with the client id and a time stamp **(out of scope)** 3. The server determines the current display balances    1. Determine current balance    2. Determine conversion rate    3. Determine multiplication rate for display purposes    4. Calculate Rand amount    5. Sort results by Rand amount    6. The service returns the result to the ATM terminal 4. The ATM terminal receives and interprets the response from the server **(out of scope)** 5. The ATM terminal displays the balances to the user **(out of scope)** |
| Extension | 1. The client does not have any qualifying accounts The system displays an error message |
| Error Message | No accounts to display |
| Note | The UI needs to be aware of how to format the amount fields |

* + 1. **Withdraw cash**

|  |  |
| --- | --- |
| Use case | Allow client to withdraw cash from an account |
| Goal in context | A client is allowed to withdraw money from a transactional account if he has a positive balance or is allowed to go into a negative balance if the account is a cheque account up to a maximum of R 10 000. |
| Level | User |
| Primary Actor | Bank client at ATM terminal |
| Precondition | Client successfully logged on to ATM terminal with bank card and has selected a transactional account to withdraw money from |
| Success end condition | The client account balance has been adjusted with the withdrawal amount and the ATM has been notified of the number of notes per denomination to dispense |
| Basic flow of events | 1. Client logs in to ATM via using the card and pin pad and selects the “Withdraw” **(out of scope)** 2. The is presented with a selection of transactional accounts and selects the appropriate account to proceed to the next screen **(out of scope)** 3. The client enters the amount required and selects submit **(out of scope)**. 4. The ATM sends a request to the server with the ATM id, client id, account number, required amount, and a time stamp **(out of scope)** 5. The server receives and interprets the request from the ATM terminal 6. The server determines the number of notes to dispense per denomination    1. Determine current balance    2. Determine if the requested amount is available for withdrawal based on balance and limit rules    3. Determine if the amount specified can be provided using only bank notes (no coins)    4. Determine the number of each denomination to dispense using the minimum number of notes using the available notes in the ATM    5. The system adjusts the clients account balance with the amount requested    6. The service returns the number of notes per denomination to dispense, to the ATM terminal 7. The ATM terminal receives and interprets the response from the server **(out of scope)** 8. The ATM dispenses the cash **(out of scope)** |
| Extension | 1. The client does not have enough available funds *The system displays an error message “Insufficient funds”* 2. The ATM does not have enough cash to dispense the required amount *The system displays an error message “Amount not available, would you like to draw X” where X is the next available amount lower than the requested amount.* 3. The system does not find the ATM in the system or does not find an allocation for the specified ATM *The system displays an error message “ATM not registered or unfunded”* |
| Error Message | Insufficient funds  Amount not available, would you like to draw X  ATM not registered or unfunded |

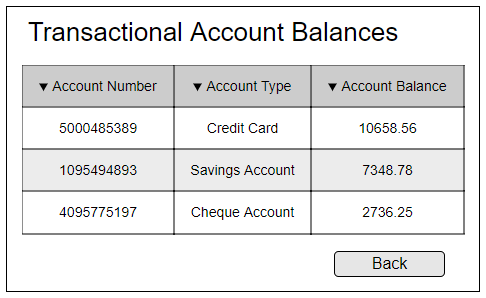
* + 1. **Reporting – Find the transactional account per client with the highest balance**

|  |  |
| --- | --- |
| Use case | Find the transactional account per client with the highest balance |
| Goal in context | A report needs to be generated that displays a list of all clients along with the account details of the client’s account with the highest balance |
| Level | System |
| Primary Actor | System |
| Success end condition | Successful report generation |
| Basic flow of events | On a monthly basis a report is generated and emailed to [support@thebank.com](mailto:support@thebank.com) **(out of scope)** |
| Note | Create a SQL script that can be executed that can be executed against the DB that will cater for the criteria specified above and return the fields below:   * Client Id * Client Surname * Client Account Number * Account Description * Display Balance |

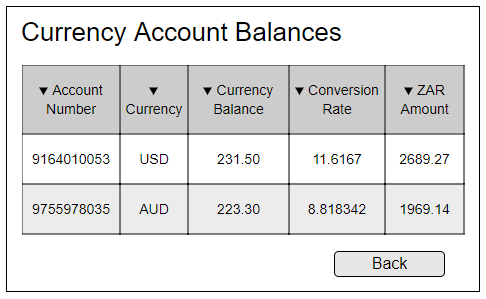
* + 1. **Reporting – Calculate aggregate financial position per client**

|  |  |
| --- | --- |
| Use case | Calculate aggregate financial position per client |
| Goal in context | A report needs to be generated that displays the net position per client based on all accounts held |
| Level | System |
| Primary Actor | System |
| Success end condition | Successful report generation |
| Basic flow of events | On a monthly basis a report is generated and emailed to [support@thebank.com](mailto:support@thebank.com) **(out of scope)** |
| Note | Create a SQL script that can be executed that can be executed against the DB that will cater for the criteria specified above and return the fields below:   * Client (Client Title + Client Name + Client Surname) * Loan Balance (Aggregate of all loan amounts) * Transactional Balance (Aggregate of all transactional accounts) * Net Position (Net position across all accounts) |

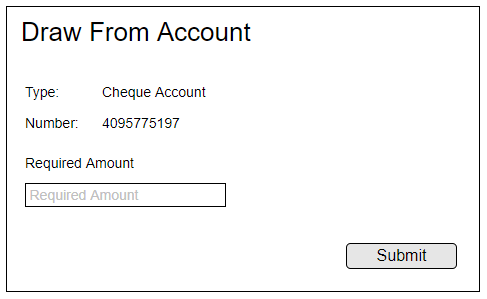
1. **Screens**
   * 1. **Display transactional accounts with balances**



* + 1. **Display currency accounts with converted Rand values**



* + 1. **Withdraw cash**



1. **Additional Information**

Scripts are provided for the creation of an H2 in memory DB and the service is to run as a Spring Boot application.

Data is provided for reference purposes and may be enhanced at time of review.

