**Project Title: Bank Accounts**

**Use Case / Project Description.**

**Create a system where a customer can login and view his bank account. Only one account per customer is expected. He should also be able to do transactions on his accounts. Only Basic transactions are expected – Deposit and Withdraw.**

**Project Scope:**

* Login screen – for customers
* Each customer is defined by a customerId and AccountNumber. Customer should be able to signup to the application using his customer Id and a password. Password to be saved in the database in hased format.
* Customer should be shown last 4 digits of his account number on login. Account number will be 12 digit, but customer will be shown only last 4 digits initially. He can then view the full account number by clicking on the same.
* It is important for the application to create REST APIs for each functionality from the backend and that be consumed from the front end. REST APIs are mandatory for the project.
* Pages for the application
  + Account Balance – which will show what is the current balance of the accounts
  + Transaction page – where customer can do transaction on the accounts
  + Account statement – where the customer can see the list of transactions done on the account.
* Application should be able to logout also have session managed
* Good to have Jenkins implemented
* Additionally if the system can let the customer download the account statement as a PDF, it will be an added advantage. Not mandatory

**Details on all the functionalities are described in further sections.**

**Technology Stack to be used:**

|  |  |  |
| --- | --- | --- |
| **#** | **Area** | **Tech choices available** |
| 1 | Backend | Java spring boot, Node JS (chose any 1) |
| 2 | Front end | React JS based web pages, HTML5 and CSS, JS based web pages |
| 3 | Database | Mysql, Postgresql, MongoDB (Chose any 1) |
| 4 | Deployment | Springboot – to be packaged as a runnable Jar file  Node JS – to be deployed on a node server |
| 5 | Devops | Maven build for Java projects  Node build for frontend and react js |

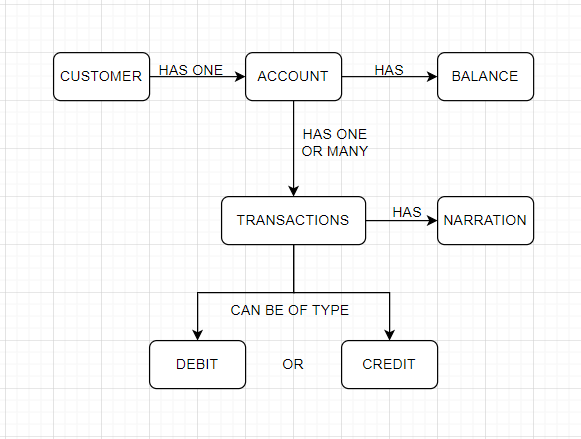
**Functional Overview**

**Objects and Relations**

To understand and develop the project better, we first should be able to understand the relations between the objects. For that the first step is to list down all the objects that are present In the system. In our project we have

* + Customer
  + Account
  + Balance
  + Transactions, (further classified as)
    - Debit
    - Credit

Relationships between the objects is as shown below



Object data types can be used as follows

|  |  |  |
| --- | --- | --- |
| **#** | **Object Name** | **Type** |
| 1 | Customer Id | Number (9 digit number) |
| 2 | Account number | Numeric (12 digit number) |
| 3 | Transaction Type | VARCHAR (DEBIT/CREDIT) |
| 5 | Transcation Amount | Number with 2 decimals |
| 6 | Transaction Narration | Varchar (upto 40 characters) |

**Pages to be developed**

**Login and Signup page:**

* There has to be a page where the customer can login / signup.
* Customer should be able to login using his customer id and password.
* He should be able to signup using his customer id and provide a password during signup. As soon as the customer signs up, he should be allocated a unique 12 digit account number (randomly generated) and also should be welcomed with Rs 100/- balance in that account.
* Password should be validated from the database. Password should not be stored in plain text in database, it should be hashed and stored.

**Accounts page:**

* On successful login, the customer should be taken to his accounts page where is 12 digit account number is shown in masked format (only the last 4 digits of the account number are to be shown). The customer should be able to view the full account number on clicking the account number.
* Along with the account number, the balance in that account should also be shown. Balance cannot be negative.
* There should be a link on the page “Do Transactions” which can take him to the transaction page.
* There should also be another link on the page “See Account statement” which should show the list of transactions done on that account.

**Transactions Page:**

When the customer enters the transaction page, he should be able to see the account balance and be asked to perform transaction. Following dataset has to be captured from the customer

Type of transaction – (Fund transfer / Cash deposit)

*Nature of transaction (Debit or credit)* – should be auto populated based on the type of transaction. If transaction type is Fund Transfer – than the nature will be debit. If the Type of transaction is Cash deposit – then nature of transaction will be credit. This field will be non editable

Amount of transaction – For Debit transactions, the amount cannot be more than account balance. For Credit transactions – the amount cannot be more than INR 10000/-

Beneficiary account number – in case of fund transfer transaction, the customer has to enter the account where the transaction has to be credited. The customers account will be debited and the entered account number has to get credited. This account number has to be a valid 12 digit account number from the database. Validation is required. In case of cash deposit transaction – this field should be auto populated and non editable as the account number of the same customer itself.

Transaction Narration (Remarks) – User defined remarks to identify his transaction. Free text where the user can enter upto 40 letters. No special characters allowed.

Once all the above data is populated on the page, the customer can click on Submit button which will confirm the transaction and the actual transaction should happen.

Below are the steps to perform actual transaction:

For cash deposit transaction -

It is a simple transaction, where the customer’s account number should be added the amount that was mentioned in the cash deposit transcation.

*Account Balance post Cash Deposit txn = Account balance pre Cash deposit txn + Amount of the Cash deposit txn.*

Also this will be captured as a transaction for the account statement.

For Fund transfer transaction:

It is a complex transaction where both debit and credit have to happen. In this case,

For the customer’s account (who is doing the transaction)

*Account balance post FT transaction = Account balance pre FT transaction – FT transaction amount*

For the beneficiary account (where the amount will get credited to)

*Account balance post FT transaction = Account balance pre FT transaction + FT transaction amount*

For a fund transfer transaction, there has to be two entries – 1 for Debit to customer account numbr and other for Credit to beneficiary account number in the transaction table.

**Account Statement Page:**

Here the customer should be able to see the list of transactions for that account. All DEBIT and CREDIT transactions have to be shown on this page. There can be on screen filters to show only CREDIT or DEBIT transcations.

Apart, as an additional functionality, the customer could also be able to download the account statement as a PDF.

**Steps to work on the project:**

1. Start with database design. Get the database design right.
2. Start with backend and front end development. Frontend should connect to backend via REST APIs only.