Problem Statement – Banking Application

Although the basic type of services provided by banks varies depending on the type of bank and the country, services typically include: accepting deposits from customers and issuing current or checking accounts and savings accounts to individuals and businesses. Loans to individuals and businesses, Cashing a check Facilitating money transfers such as wire transfers and cashiers cheques, as well as consumer and commercial financial advisory services, financial transactions can be carried out through a variety of channels.

**Motivation to building banking project:**

To create a system that will monitor the activities of a specific bank's transactions without the need for manual processing. All transactions will be updated automatically based on the information stored in the record. The main goal of this project is to create a system that can handle the overall tasks that occur within the institutions with minimal effort.

The application shall have different type of users

1. Customer
2. Staff
3. Approver

Where each type of the user is given dedicated role.

**Technology List**

|  |  |
| --- | --- |
| **Type** | **Name** |
| Backend | SpringBoot |
| Frontend | React , hook and redux |
| Cloud | AWS |
| Architecture | Microservices |

**Port Numbers**

|  |  |
| --- | --- |
| **Service Name** | **Port Number** |
| MS1 | 8100-8105 |
| MS2 | 8200-8205 |
| MS3 | 8300-8305 |
| Eureka | 8761 |
| API Gateway | 8765 |
| MySQL | 3306 |
| MongoDB | 27017 |
| Angular Application | 4200 |

**Package names**

|  |  |
| --- | --- |
| **Layer Type** | **Name** |
| App | com.learning |
| Service | com.learning.service |
| Contracts/interfaces | com.learning.contracts |
| Repository | com.learning.repo |
| JPA | com.learning.jpa |
| Entity | com.learning.entity |
| Controllers | com.learning.controller |
| Cross Cutting/AOP | com.learning.aop |
| Others | com.learning.\* |

**Scope of work – Backend**

The application shall Registration, Login, and showing Food items when user logs in to the system, the user should have a proper authentication and authorization system.

The application shall allow registration, login of customer, staff and super admin.

1. The customer shall be able to Create Account, Add Beneficiary, Modify Beneficiary, Transfer Money, View Statement, Reset Forget password
2. The staff should be able to view Account statement, Approve Beneficiary, Approve Account, Enable or disable Customer
3. The super admin can create Staff and enable or disable

Given backend & the end points to be built:

|  |  |  |
| --- | --- | --- |
| **Method** | **End point** | **Description + Payload** |
| POST | /api/customer/register | To register the user with basic details like  **Input:**  {  username:String,  fullname:String,  password: String  }  **Return code:** 201  Return:  {  id: Number,  username:String,  fullname:String,  password: String  }  **Description:** The end point shall save the customer record to the DBMS (Mysql), and as acknowledgement the actual object is given(JSON).  **DBMS:** Mysql |
| POST | /api/customer/authenticate | To validate the customer is registered in the system  **Input:**  {  username: String,  password: String  }  **Return:**  “JWT Token”  **Return Code:** 200  When user details are wrong Return **Code:** 403 (Forbidden)  **DBMS:** Mysql |
| POST | /api/customer/:customerID/  acocunt | To create account for the customer  Input payload  {  accountType:Enum(SB/CA),  accountBalance:Number,  approved: String - no (default)  }  When successful  **Return code:** 200  {  accountType:Enum(SB/CA),  accountBalance:Number,  approved: no,  accountNumber: Number,  dateOfCreation: date/time,  customerId: Number  }  When not successful  {  Message: “Account cannot be created”  }  **Return code:** 403  **DBMS:** MongoDB/Mysql |
| PUT | /api/customer/:customerID/  account/:accountNo | **Role:** Staff  To approve the account which is create by customer by  Input payload  {  accountNumber: Number,  approved: yes  }  When successful  **Return code:** 200  {  accountNumber: Number,  approved: yes  }  When not successful  {  message: “Please check Account Number”  }  **Return code:** 403  **DBMS:** MongoDB/Mysql |
| GET | /api/customer/:customerID/  acocunt | To get all the accounts which are opened by the customer the end point should return an array of account, balance, and type, and status.  [  {  accountNumber: Integer,  accountType:Enum(SB/CA),  accountBalance:Number,  accountStatus: Enum(Enable/Disable)  },  {  accountNumber: Integer,  accountType:Enum(SB/CA),  accountBalance:Number,  accountStatus: Enum(Enable/Disable)  },  ]  **Return code:** 200  **DBMS:** MongoDB |
| GET |  | To return customer by specifying id  When valid  {  username: String,  fullName : String,  phone: String,  pan:String,  aadhar: String  }  When Id not found:  {  Message: ”Sorry Customer With <ID> not found”  }  **Return Code:** 200  **DBMS:** Mysql |
| PUT | /api/customer/:customerID | Should update the user customer in the payload which shall match the username and updated the existing customer with the new details.  When valid  {  customerId: Number,  fullname: String,  phone:String,  pan: String,  aadhar : String,  secretQuestion: String,  secretAnswer: String,  pan: multi part/image,  aarchar : multi part/ image  }  **Return Code:** 200  When Id not found:  {  Message: ”Sorry customer With <ID> not found”  }  **DBMS:** Mysql |
| GET | /api/customer/:customerID  /account/:accountID | Should get the customer account with the specified account number :accountID  When Valid :    {  accountNumber: Integer,  accountType:Enum(SB/CA),  accountBalance:Number,  accountStatus:Enum(Enable/Disable),  transaction: [  {  date: date/time,  reference: String,  amount: number,  db-cr: Enum(DB/CR)  },  {  date: date/time,  reference: String,  amount: number,  db-cr: Enum(DB/CR)  } ,  …  ]  }  **Return Code:** 200  When Id not found:  {  Message:”Sorry Account With <ID> not found”  }  **DBMS:** MongoDB |
| POST | /api/customer/  :customerID  /beneficiary | Should add the beneficiary for the customer with valid account number  {  accountNumber: Number,  accountType: Enum(SB/CA),  approved: no  }  When Valid :  {  Message: ”Beneficiary with <Account Number> added”  }  **Note:** By default active : Yes  **Return Code:** 200  When Id not found:  {  Message: ”Sorry Beneficiary With <Account Number> not added”  }  **DBMS:** MongoDB |
| GET | /api/customer/  :customerID  /beneficiary | Should get all the beneficiary for the given customer id  When Valid :  [  {  beneficiaryAccountNo: Number,  beneficiaryName: String,  active: Enum (Yes/No)  },  {  beneficiaryAccountNo: Number,  beneficiaryName: String,  active: Enum (Yes/No)  },  …  ]  **Return Code:** 200  When Id not found:  []  **DBMS:** Mysql/MongoDB |
| DELETE | /api/customer/  :customerID  /beneficiary/:beneficiaryID | Role: Customer  When Valid :  {  message: “Beneficiary Deleted Successfully”  }  **Return Code:** 200  When Not Valid:  {  message: “Beneficiary Not Delete”  }  **DBMS:** MySql |
| PUT | /api/customer/  /transfer | Role: Customer  To transfer the amount from one account to another account  payload {  fromAccNumber: Number,  toAccNumber: Number,  amount: Number,  reason: String,  by:customer  }  **Return Code:** 200  When Not Valid:  {  message: ”From/To Account Number Not Valid”  }  **DBMS:** MongoDb |
| GET | /api/customer/  /:username/forgot/  question/answer | Role: Customer  {  message: “Details validated”  }  **Return Code:** 200  When Not Valid:  {  Message: ”Sorry your secret details are not matching”  }  **DBMS:** Mysql |
| PUT | /api/customer/  /:username/forgot | Role: Customer  {  username:String,  password:String  }  **Return Code:** 200  {  message: ”new password updated”  }  When Not Valid:  {  Message: ”Sorry password not updated”  }  **DBMS:** Mysql |
| POST | /api/staff/authenticate | To validate the staff should be able to login to the system with validate username and password  **Input:**  {  username: String,  password: String  }  **Return:**  “JWT Token”  **Return Code:** 200  When user details are wrong Return **Code:** 403 (Forbidden)  **DBMS:** Mysql |
| GET | /api/staff/account/  :accountNo | Role: Staff  To get the statement of particular account, given the account id  Return Value  {  accountNo:Number,  customerName: String,  balance: Number,  transaction:[  {  date: date,  reference: String,  amount:Number,  transactionType: Enum(CR/DB)  },  {  date: date,  reference: String,  amount:Number,  transactionType: Enum(CR/DB)  }  …  ]  }  **Return Code:** 200  When Not Found  {  message:”account not found”  }  **DBMS:** MongoDB |
| GET | /api/staff/beneficiary | Role: Staff  List all the beneficiary to be approved  **Return Code:** 200  [  {  fromCustomer: Number  beneficiaryAcNo: Number,  beneficiaryAddedDate: date,  approved: String(No)  },  {  fromCustomerAcNo: Number  toBeneficiaryAcNo: Number,  beneficiaryAddedDate: date,  approved: String(No)  },  ..  ]  **Return Code:** 200  When not no beneficiary found to be approved  []  **DBMS:** Mysql |
| PUT | /api/staff/beneficiary | **Role:** Staff  Approve the beneficiary which were added by customer  Payload  {  fromCustomer: Number  beneficiaryAcNo: Number,  beneficiaryAddedDate: date,  approved: String (Yes)  }  **Return Code:** 200  {  fromCustomer: Number  beneficiaryAcNo: Number,  beneficiaryAddedDate: date,  approved: String (Yes)  }  **Return Code:** 200  When error while approving  {  Message: “Sorry beneficiary not approved”  }  **DBMS:** Mysql |
| GET | /api/staff/accounts/approve | **Role:** Staff  List all the accounts to be approved  **Return Code:** 200  [  {  accType:String,  customerName: String,  accNo: Number,  dateCreated: date,  approved:no  },  {  accType:String,  customerName: String,  accNo: Number,  dateCreated: date,  approved:no  }  …  ]  **DBMS:** Mysql |
| PUT | /api/staff/accounts/approve | **Role:** Staff  Approve the list of accounts added by the customer, so that that the transaction can be done  Payload  {  accType:String,  customerName: String,  accNo: Number,  dateCreated: date,  approved:String (Yes),  staffUserName:String  }  **Return Code:** 200  When error while approving  {  message: “Approving of account was not successful”  }  **DBMS:** Mysql |
| GET | /api/staff/customer | **Role:** Staff  List all the customer  [  {  customerId:Number,  customerName: String,  status: Enum(Enable/Disable)  },  {  customerId:Number,  customerName: String,  status: Enum(Enable/Disable)  }  …  ]  **Return Code:** 200  **Note:**  1. By default when customer registers status: Enable  When no customer are there  []  **DBMS:** Mysql |
| PUT | /api/staff/customer | Enable or disable the customer, based on that the customer should be able to login  {  customerId:Number,  status: Enum(Enable/Disable)  }  Return Code: 200  When Error  {  message:”Customer status not changed”  }  **DBMS:** Mysql |
| GET | /api/staff/customer  /:customerID | **Role:** Staff  Get customer with the id  {  customerId:Number,  customerName: String,  status: Enum(Enable/Disable),  created:date,  }  **Return Code:** 200  **Note:**  Return Code: 403  {  message:”Customer Not Found”  }  **DBMS:** Mysql |
| PUT | /api/staff/  /transfer | Role: Staff  To transfer the amount from one account to another account  payload {  fromAccNumber: Number,  toAccNumber: Number,  amount: Number,  reason: String,  by:String (Staff)  }  **Return Code:** 200  When Not Valid:  {  message: ”From/To Account Number Not Valid”  }  **DBMS:** MongoDb |
| POST | /api/admin/authenticate | To validate the admin is registered in the system  **Input:**  {  username: String,  password: String  }  **Return:**  “JWT Token”  **Return Code:** 200  When user details are wrong Return  **Code:** 403 (Forbidden)  Note: There will not be registration, in the DB there shall be role called “admin”  **DBMS:** Mysql |
| POST | /api/admin/staff | Create staff in the system  {  staffFullName: String,  staffUserName: String,  staffPassword: String  }  **Return Code:** 200  When user name already exists or any error  **Code:** 403 (Forbidden)  **DBMS:** Mysql |
| GET | /api/admin/staff | **Role:** Admin  List all the Staff  [  {  staffId:Number,  staffName: String,  status: Enum(Enable/Disable)  },  {  staffId:Number,  staffName: String,  status: Enum(Enable/Disable)  }…  ]  **Return Code:** 200  When no staffs are there  []  **DBMS:** Mysql |
| PUT | /api/admin/staff | Enable or disable the staff, based on that the staff should be able to login  {  staffId:Number,  status: Enum(Enable/Disable)  }  Return Code: 200  When Error  {  message: “Staff status not changed”  }  **DBMS:** Mysql |

**Scope of work – Frontend**

The application shall allow user to

1. Customer - Home Page
2. Customer - Registration
3. Customer - Login
4. Customer – Forgot Password
5. Customer - Logout
6. Customer - View dashboard
7. Customer – Create Account
8. Customer – Add Beneficiary
9. Customer – Remove Beneficiary
10. Customer – Transfer Amount
11. Customer - Update Profile
12. Customer - View Account Statement
13. Staff - Login
14. Staff – Logout
15. Staff – Approve/Reject Customer Account
16. Staff - Enable / Block Customer
17. Staff – Credit/Debit Amount
18. Staff – Approve/Modify Beneficiary
19. Staff – View Account Statement
20. Approver – Create Staff
21. Approver – Enable /Disable Staff

**User Stories**

1. Customer - Home Page

**As a customer,**

**I want to see the home page of the web application**

**So that I can login to the banking**

**Graphical user interface, application

Description automatically generated**

1. Customer - Registration

**As a customer,**

**I want to get registered into the banking app,**

**So that I can login to the system to create an account and do transaction**

**Acceptance Criteria:**

Given when the application is loaded

Then Navigate to the register page which shall be located in the home screen “Register Here”

And Enter user name (which shall be unique)

And Enter full name

And Enter password

And Enter confirm password (both should be the same)

When Click on “Register” button

Then A confirmation message to be shown “You Registered successfully”

**Home Screen**

**Graphical user interface, application

Description automatically generated**

Registration Screen

Graphical user interface, application

Description automatically generated

1. Customer - Login

As a customer,

I want to be able to login to the web app

So that I can view the dashboard which has list of accounts which is created.

**Acceptance Criteria:**

Given when the application is loaded

When Navigate to the login page (Click on “Login” button on home page)

And Enter valid username

And Enter valid password

And Click Login button

Then customer is taken to home page where customer can list of accounts he/she has

And option to click on profile

And Create Account, Add Beneficiary, Modify Beneficiary, Transfer Money, View Statement

**Graphical user interface, application

Description automatically generated**

1. Customer – Forgot Password

As a customer,

I want to be able to give my security question and answer

So that I can reset my forgot password

**Acceptance Criteria (1):**

Given when the application is loaded

When user is on the home page

And clicks on “Forgot Password’

Then a new screen to be shown when use enter username, select security question and enter security answer

When the details are valid

Then the user to be taken to another screen where user name should be shown

And new password and confirm password field should be entered

When clicked on “Update” button

Then password and confirm password fields should be matched

When matched then the database should set with the new password

And user to be taken to home page

**Acceptance Criteria (2):**

Given when the application is loaded

When user is on the home page

And clicks on “Forgot Password’

Then a new screen to be shown when use enter username, select security question and enter security answer

When the details are not valid

Then the user to be taken to another screen an error message to be shown

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Customer - Logout

**As a customer,**

**I want to be able to log out from the web app,**

**So that I can end my current session.**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

When Navigate to the Logout button ( Click on “Logout” on top right corner)

Then End the current customer session

And Navigate to Home page of the application

1. Customer - View dashboard

**As a customer,**

**I want to be able to view the my accounts**

**So that I can do transaction and manage accounts**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of options like Create Account, Add Beneficiary, Remove Beneficiary, Transfer Money, View Statement

And on Right side of the screen the list of accounts the user has

And on top of the screen to Logout

And view profile

Graphical user interface, text, application

Description automatically generated

1. Customer – Create Account

**As a customer,**

**I want to be able to create an account**

**So that I can do transaction like transfer**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “Create Account”

Then Create Account Form to be shown which shall ask for initial deposit, and type of account

When clicked on submit the account should be create

And message to be given “Account Creation under process”

Note: the account activation shall be done by admin

Graphical user interface, text, application

Description automatically generated

1. Customer – Add Beneficiary

**As a customer,**

**I want to be able to add the Beneficiary**

**So that I transfer the amount**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “Add Beneficiary”

Then Add Beneficiary form to be shown which shall ask for Account Number, Confirm Account Number, Type of the account

When clicked on “Submit” button

Then a message to be shown “Beneficiary added, and is validating”

Graphical user interface, text

Description automatically generated

1. Customer – Remove Beneficiary

**As a customer,**

**I want to be able to remove the Beneficiary**

**So that I don’t have to make any transaction with the beneficiary**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “Remove Beneficiary”

Then Remove Beneficiary form to be shown which shall list all the beneficiary added by me earlier

When clicked on (-) minus icon which is next to name

Then the beneficiary to be deleted

And a message to be shown “Beneficiary successfully deleted”

And with this Beneficiary not transaction shall happen

Graphical user interface

Description automatically generated with low confidence

1. Customer – Transfer Amount

**As a customer,**

**I want to be able view all the accounts**

**So that I can transfer amount to another account or I can receive**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “Transfer Money”

Then all the account which the user has to be shown in a tabular format

And there shall be a drop down of the account who’s Beneficiary are approved

When selected the Beneficiary

And entered the amount

And entered the reason for the transfer

And the amount entered is less than the balance amount of the selected account

When clicked on “Transfer” button

Then the transfer should successfully be taken

And a message to be shown “Transaction Successful”

Graphical user interface, text, application

Description automatically generated

1. Customer – Update Profile Amount

**As a customer,**

**I want to be able view update my profile information**

**So that I can store my PAN, AADHAR, Secret Question and Password**

**Acceptance Criteria:**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “Profile” link on top of the screen

Then a new page should be shown where customer id should not be editable, FullName, Phone, PAN, AADHAR, etc

When user enters full name, PAN, AADHAR, attachment of PAN, AADHAR, Select security question, Security Answer

And clicked on “update” button

Then message to be shown “Profile updated successfully”

Note: any of these fields are option

Graphical user interface

Description automatically generated

1. Customer – View Account Statement

**As a customer,**

**I want to be able view my account statement**

**So that I should know how my transaction’s had happened in the past**

**Acceptance Criteria(1):**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “View Statement”

Then a new page should be shown where list of accounts which user has to be shown in the drop down upon selecting the account

And clicking on the Show button

Then the account statement to be shown in the descending order of the date with the type of transaction

And end user should be able to download the statement in the PDF / Excel format

**Acceptance Criteria(2):**

Given when the application is loaded

And the customer is logged in

Then show dashboard which has list of accounts

When clicked on “View More” on the dashboard

Then the account statement to be shown in the descending order of the date with the type of transaction

And end user should be able to download the statement in the PDF / Excel format

Graphical user interface

Description automatically generated

1. Staff – Login

**As a staff,**

**I want to be able to login to the web app**

**So that I can view account statement, approve Account, Approve Beneficiary , Enable or Disable Customers**

**Acceptance Criteria:**

Given when the application is loaded

When Clicked on “Staff Corner” which is located on top right corner

Then staff login page should be shown

And Enter valid username

And Enter valid password

And Click Login button

Then staff is taken to home page where staff is given various choices like “By Account Number”, “Approve Beneficiary”, “Approve Account”, “Enable/Disable Account”

Graphical user interface, application

Description automatically generated

Text

Description automatically generated

1. Staff – Logout

**As a staff,**

**I want to be able to log out from the web app,**

**So that I can end my current session.**

**Acceptance Criteria:**

Given when the application is loaded

And the staff is logged in

When Navigate to the Logout button ( Click on “Logout” on top right corner)

Then End the current customer session

And Navigate to Home page of the application

1. Staff – Approve/Reject Customer Account

**As a staff,**

**I want to be able enter account number,**

**So that I can view account transaction**

**Acceptance Criteria**

Given when the application is loaded

And the staff is logged in

Then show dashboard

When clicked on “By Account Number”

Then a new page should be shown where application shall accept valid account number

When clicked on GO Button

Then the account statement to be shown in the descending order of the date with the type of transaction

And customer Name, and current balance to be shown

Table

Description automatically generated

1. Staff - Enable / Block Customer

**As a staff,**

**I want to be able to view all the customers**

**So that I can enable or disable the login**

**Acceptance Criteria**

Given when the application is loaded

And the staff is logged in

Then show dashboard

When clicked on “Enable / Block Customer”

When a new page shall open and list all the customers who are registered to the system

And staff is provided with a toggle button next to customer Id

When disabled

Then customer should not be able to login

When enabled

Then customer should be able to login

Table

Description automatically generated

1. Staff – Credit/Debit Amount

**As a staff,**

**I want to be able to enter account number (from/to), amount, reason**

**So that I can transfer amount on behalf of customer**

**Acceptance Criteria**

Given when the application is loaded

And the staff is logged in

Then show dashboard

When clicked on “Transfer” on top of the screen

Then a new form should be shown which shall take from Account number, To Account Number, Amount, Remarks

When the details are valid

And clicked on “Transfer” button

Then the transaction should be successful

Text

Description automatically generated with medium confidence

1. Staff – Approve Beneficiary

**As a staff,**

**I want to be able to see all pending approved Beneficiary**

**So that I can approve the Beneficiary post validating details**

**Acceptance Criteria**

Given when the application is loaded

And the staff is logged in

Then show dashboard

When clicked on “Approve Beneficiary”

Then list of beneficiary to be shown in a form to table

When clicked on Approve against the Name of the Customer Name

Then an acknowledgement to be shown “Beneficiary Approve”

Note: Any transaction to be done only post approval Staff / Customer

Graphical user interface, table

Description automatically generated

1. Staff – View Account Statement

**As a staff,**

**I want to be able to see transaction of account**

**So that I can provide better understanding of transaction**

**Acceptance Criteria**

Given when the application is loaded

And the staff is logged in

Then show dashboard

When clicked on “By Account Number”

Then a new form to be opened

And there shall be a text box which should ask for valid account number

When entered and clicked “Go”

Then the transaction of the particular account number to be shown in descending order of date

Table

Description automatically generated

1. Approver – Create Staff

**As a super admin,**

**I want to be able to create staff**

**So that staff manages the bank**

**Acceptance Criteria**

Given when the application is loaded

When clicked on Staff Corner

When entered the user name as [admin@admin.com](mailto:admin@admin.com), and password: secret@123

Then a form to be shown to create a staff asking Staff Name, Staff User Name (unique), password, confirm password

When clicked on Create button

Then Message to be shown “Staff Created Successfully”

Graphical user interface

Description automatically generated

1. Approver – Enable /Disable Staff

**As a super admin,**

**I want to be able to suspend the staff**

**So that I can restrict**

**Acceptance Criteria**

Given when the application is loaded

When clicked on Staff Corner

When entered the user name as [admin@admin.com](mailto:admin@admin.com), and password: secret@123

Then dashboard to be shown

When clicked on View Staffs

Then list of staffs to be shown

And against the username there shall be a toggle button

When toggled the user should be enable or disabled to login

Table

Description automatically generated