**ICB BANKING SYSTEM**

**Requirements:**

* Login page for admin, customer and employee
* A customer can have a bank account which can be a current or a savings account.
* The interest rates and charges are calculated differently for current and savings accounts.
* A customer can add money, deduct money, check their balance, view their transaction history, and update his information.
* An employee can create a customer account, delete a customer account.
* A customer, employee and admin must be able to see the transactions

**MODELS:**

* Employee
  + employeeId
  + employeeName
  + employeeAddress
  + employeeEmail
  + employeePhoneNumber
* Admin
  + adminId
  + adminName
  + adminAddress
  + adminEmail
  + adminPhoneNumber
* Customer:
  + customerId
  + customerName
  + customerAddress
  + customerEmail
  + customerPhoneNumber
  + BankAccount
    - accountNumber
    - balance
    - type
    - charges
    - interest
    - transaction
      * amount
    - branch
      * head office
* Transaction
  + transactionId
  + amount
  + fromAccount
  + toAccount

**FUNCTIONALITY OF EACH MODEL:**

* **Customer:**
  + Should be able to add or remove amount in their account through transactions from one valid bank account to another.
  + Should be able to update customer details.
  + Login into customer account
* **Employee:**
  + Should be able to create new customer account or delete a customer account or edit a customer account.
  + Should be able to update any customer information and their own information.
  + Login into employee account
* **Admin**
  + Should be able to create, delete new customer and employee accounts as well as other admin accounts
  + Should be able to update any customer or employee information and their own information
  + Login into admin account
* **Transaction**
  + Should be able to be accessed by customer, employee and admin.

**Backend:  
Using: Spring boot**

**Using: REST API**

**Frontend:**

**Using: Angular**

**Endpoints:  
Customer:**

Update customer: POST: [http://localhost:8080/customer/update/{id}](http://localhost:8080/customer/update/%7bid%7d)

Delete a customer: POST: [http://localhost:8080/customer/delete/{id}](http://localhost:8080/customer/delete/%7bid%7d)

Store transaction: POST: http://localhost:8080/customer/{sendid}/transaction/{receiveId}

**Employee:**

Create an employee: POST: <http://localhost:8080/employee>

Get all employees: GET: <http://localhost:8080/employee>

Update employee: POST: [http://localhost:8080/employee/update/{id}](http://localhost:8080/employee/update/%7bid%7d)

View transactions of a customer: [http://localhost:8080/admin/view/transaction/{id}](http://localhost:8080/admin/view/transaction/%7bid%7d)

Create a customer: POST: <http://localhost:8080/employee/create/customer>

Get all customers: GET: <http://localhost:8080/employee/view/customer>

Get customer by id: GET: http://localhost:8080/employee/view/customer/{id}

**Admin:**

Create an employee: POST: <http://localhost:8080/admin/create/employee>

Create a customer: POST: <http://localhost:8080/admin/create/customer>

View all customers: GET: <http://localhost:8080/admin/view/customer>

View all employees: GET: <http://localhost:8080/admin/view/employee>

Delete an employee: POST: <http://localhost:8080/admin/delete/employee>

Delete a customer: POST: <http://localhost:8080/admin/delete/customer>

View transactions of a customer: [http://localhost:8080/admin/view/transaction/{id}](http://localhost:8080/admin/view/transaction/%7bid%7d)