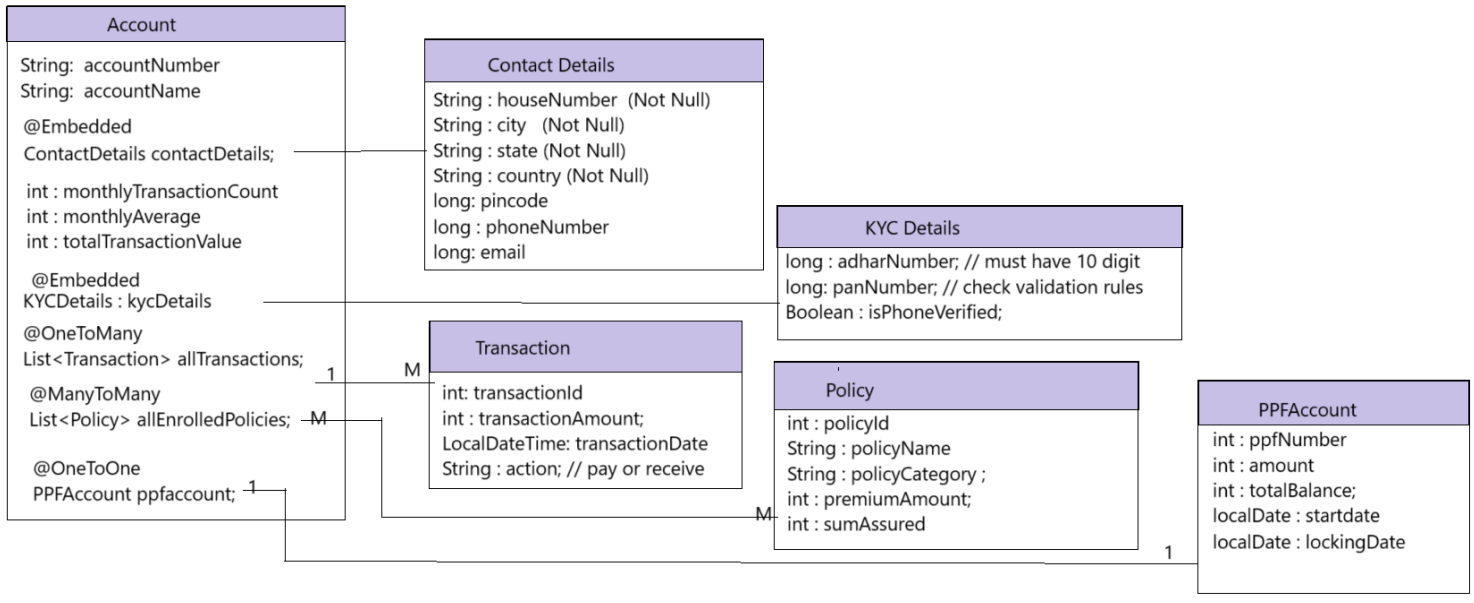
Spring Assignment

Objective : Assignment is based on Banking Business requirement , where participants have to design the application and database schema from the starch.

After completion of this assignment participant will be able to perform following technical objectives.

1) Establish various business relationships through Spring data JPA
2) Invoking api's (Endpoints) using Spring Boot RestController , mapping using requests , setting request URLs , request body and request params.
3) Executing queries through Spring Data JPA
4) Implementing DTO objects and Setting Response and Request Entities.
5) Implementing Server-side validation.
6) Implementing Custom Exception Handling.
7) Implementing non-functional requirements such as *(Optional Part, based on Curriculum)*
    a. Spring Security using JWT
    b. Logging
    c. Swagger UI
8) Implementation of Hateoas *(Optional Part, based on Curriculum)*
9) Implementation of Docker *(Optional Part, based on Curriculum)*
10) Deploying Application on Cloud *(Optional Part, based on Curriculum)*

Class Diagram



Important Business Highlights

1) One Account holder can have many policies such as health policy , car policy , home policy , life insurance policy etc. Similarly one policy plan can be purchased by many accounts.
2) One Account holder can perform various Transactions, each transaction is unique to that account. There Is One to many relationship between Account & Transaction
3) One Account may opt for PPF account where an account has to submit certain amount of fund every month, which cannot be withdrawn within certain locking period of time, the amount submitted every month must be added in a totalBalance;
4) Validation rule for PAN Number
    a. First six characters
    b. Followed by 4 digit numbers
    c. Followed any single character out of given list [A,B,C,D,E,F]
    Example : WEHUID1234B , POIUYT1111F , ~~TGBYHN1221H , QWE10A~~
5) Account Name , Policy Name , city , state & country must contains only characters with min length : 2
6) Validate emailId & phoneNumber based on standard validation rules

Business rules

1) Account cannot perform any transaction unless phone Number is verified, otherwise custom Exception “PhoneNumberVerificationException” should be raised.
2) Account cannot transfer/withdraw amount which is more than (total balance + 3000), otherwise “BalanceInsufficientException ” should be raised.
3) Account cannot withdraw amount from PPF Account (if avaliable), before lockingDate, otherwise “PPFLockingDateException” should be raised.
    a. As well as BalanceInsufficientException also apply on ppf account.

Functional Requirements

1) Add new account based on following inputs

Input
```
{
  accountName: "Ramesh Kumar"
}
```

Output
```
{
  accountNumber : 1234
  accountName : "Ramesh Kumar"
}
```

2) After creation of account update contact details based on following input

```
{
  houseNumber: "ABC-123"
  city : "New Delhi"
  State : "Delhi"
 Country : "India"
 Pincode : "110058"
}
```

3) Update account's phone number & email based on separate endpoints for both.
4) Update KYC details based on following input

```
{
  adharNumber : 123456789
  panNumber : abcdef1234b
}
```

5) Perform various (deposit & withdrawal transactions) on account
6) Add new Policy (no update/delete operation on Policy)
    a. Implement following Get Mappings on Policy
        i. Get Policy by id
        ii. Get Policy by policy category
7) Account can purchase any policy
    a. Note : while purchasing that policy , premium amount should be deducted from account total balance , otherwise "BalanceInsufficientException" be raised and policy should not be added
8) Account can opt for PPF Account opening based on certain amount.

Implement following Get Mappings on Account

1) Get Account based on accountNumber
2) Get Accounts based on city
3) Get Total Balance of all accounts in a city & state
4) Get Accounts where KYC is not null or null
5) Get accounts where isPhoneVerified is true or false
6) Get all transactions of an accounts based on month / given date or / date range
7) Get All Policy details opted by an Account
8) Get details of PPF account if opted by an Account
9) Implementing withdrawal amount from PPF account.
10) Getting the list of all Accounts having PPF account in a state

Output:

Account Number	PhoneNumber	Account Holder Name	Balance	Total Transaction Value	PPF Number	PPF Amount	PPF balance

Hint: use Inner Join