



Retail Banking System

MFPE (POD - 1) ADM21JF010

Mentor : Karthikeyan Balakrishnan

Trainer : Saurabh Kumar Sharma

Coach : Madhumathi Malarvannan

Team Members

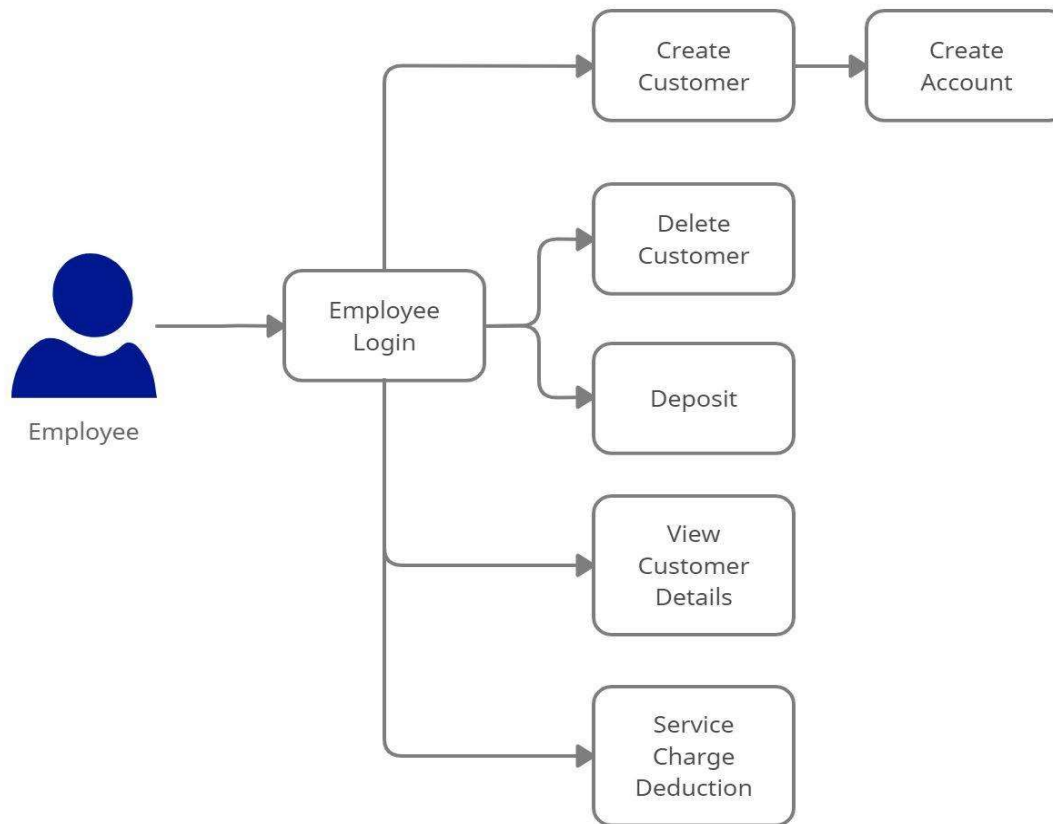
- Akash Yadav(2064581)
- Sharath Nair(2063655)
- Brahmanaidu V(2063357)
- Sekhar Kuppili (2063130)
- Yashraj Sisode (2062867)

Introduction

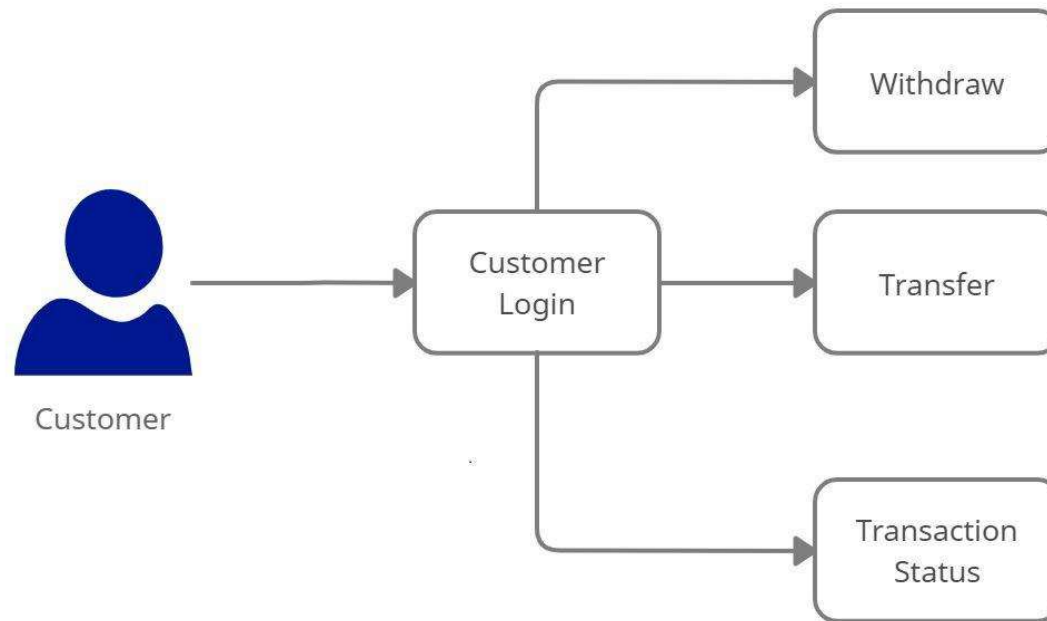
- The Retail Banking System provides **Microservice Architecture** based software solution.
- It provides **high Availability** operations for their client & bank employees without disruption.
- Different **Microservices** combines together to perform together as a **Unified** application.



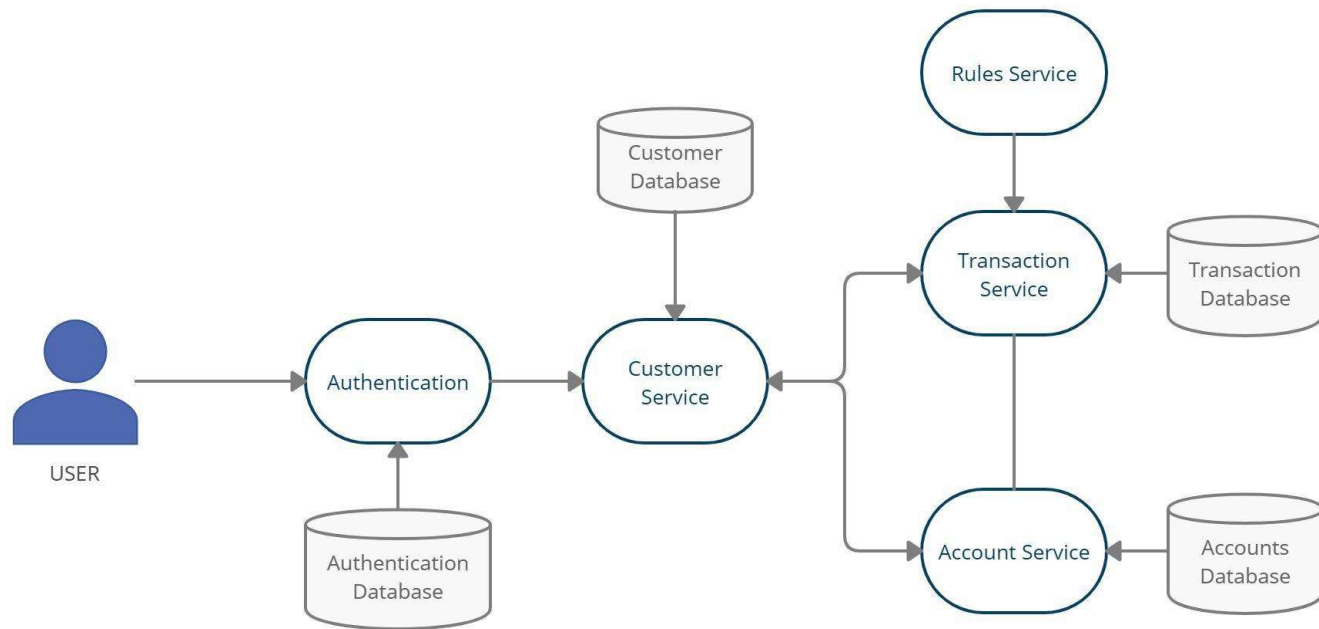
Architecture Diagram for Employee



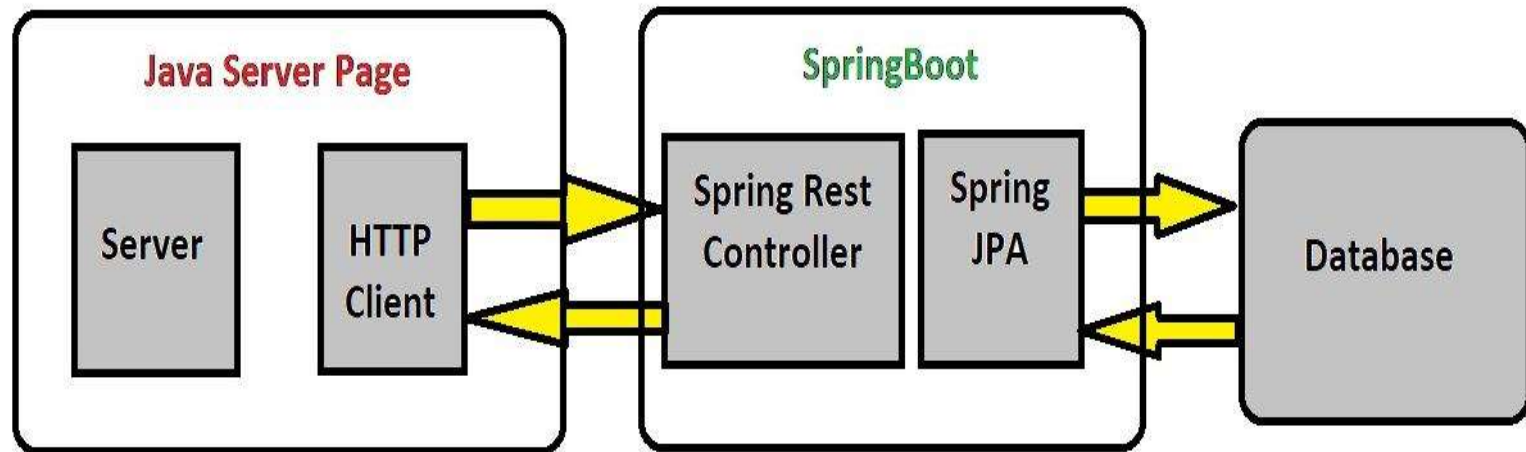
Architecture Diagram for Customer



WorkFlow:



Client - Server Architecture



Technologies Used

- ❖ Front End : Java Server Page
- ❖ Backend : Java, Spring Boot, Restful-Services
- ❖ Database : H2 - database
- ❖ Tools : Spring Tool Suite, Swagger, Postman, AWS

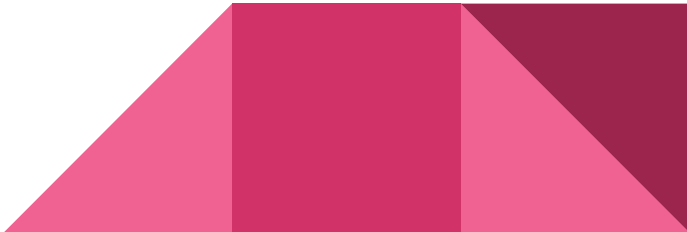


Working Microservices:

- ❖ Authorization microservice
- ❖ Customer microservice
- ❖ Account microservice
- ❖ Transaction microservice
- ❖ Rules microservice



Authentication Microservice

- ❖ Authentication Microservice will perform **operations** like:
 - **Login &**
 - **Logout**
 - ❖ Authentication will provide the **JWT** token after user's validation.
 - ❖ Providing the **authorization** based on **user's login**
- 

Authorization Microservice - Sample request

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8084/auth-ms/login
- Params:** (empty)
- Body:** JSON (application/json)

```
1 {  
2   "userid": "EMPLOYEE101",  
3   "password": "emp",  
4   "role": "EMPLOYEE"  
5 }
```
- Status:** 202 Accepted
- Response Body:** JSON (application/json)

```
1 {  
2   "userid": "EMPLOYEE101",  
3   "authToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50b3R5IiwiaWF0IjoxNjU5MDczfQ.eCva17280QAFQ6cV24Iret5z8GA920yeGEFbQhLQaJg",  
4   "role": "EMPLOYEE"  
5 }
```

Customer Microservice

- ❖ Customer microservice will performs **operations like** :
 - Creating Customer Details
 - Fetching Customer Details
 - Deletion of Customer
- ❖ It even interacts with authentication and account microservices for the purpose of sharing details.




Customer Microservice - Sample request

DELETE ☐ http://localhost:8085/customer/deleteCustomer/CUSTOMER101 Params [Send](#) ☐

Authorization Headers (2) Body ☒ Pre-request Script Tests

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> Content-Type	application/json			
<input checked="" type="checkbox"/> Authorization	eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT1IFRTEwMSIsImV4cCI6MTYyNz...			
New key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview JSON ☐ 

```
1 {  
2   "message": "CUSTOMER DELETED"  
3 }
```

Account Microservice

- ❖ Account Microservice will perform **following operations**:
 - **Customer Account Creation**
 - **Fetching Account Details for Each Customer**
 - **Deposit and Withdraw from customer's account.**
- ❖ **Interacts with transaction microservice for validating the account details and transferring the amount.**



Account Microservice - Sample request

GET localhost:8086/account-ms/getAccount/1000000002 Params Send

Authorization Headers (2) Body Pre-request Script Tests

Key	Value	Description
Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpzZW50L3VudC01FRTEwMSIsImV4cCI6MTYyNz...	
Content-Type	application/json	
New key	Value	Description

Body Cookies Headers (6) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 {
2   "accountId": 1000000002,
3   "customerId": "CUSTOMER102",
4   "currentBalance": 40000,
5   "accountType": "Current",
6   "openingDate": "2021-10-08T18:30:00.000+00:00",
7   "ownerName": "Albert Einstein",
8   "transactions": null
9 }
```

Transaction Microservice

- ❖ Transaction Microservices will perform **operations like**:
 - **Fetching transactions history** for individual account.
 - **Validating the account details** for individual customer.
 - **Checks minimum balance** for the transfer and withdraw account.
- ❖ Transactions microservice will **interact** with **Account microservice** to actually **complete the transactions within the accounts**.



Transaction Microservice - Sample request

The screenshot displays a REST client interface for a POST request. The URL bar shows 'localhost:8087/transaction-ms/deposit' with a 'Send' button to the right. Below the URL bar, tabs for 'Authorization', 'Headers (2)', 'Body', 'Pre-request Script', and 'Tests' are visible, with 'Body' currently selected. The 'Body' tab shows a JSON payload:

```
{  "accountId":1000000002,  "amount":5000}
```

. Below the body tab, there are tabs for 'Body', 'Cookies', 'Headers (7)', and 'Test Results', with 'Body' selected. The 'Body' tab shows a response of

```
true
```

. The status bar at the bottom right indicates 'Status: 200 OK'.

POST localhost:8087/transaction-ms/deposit

Params Send

Authorization Headers (2) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "accountId":1000000002,
3   "amount":5000
4 }
```

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 true
```


Rules Microservice

- ❖ Rules Microservice will be responsible for **evaluating rules**:
 - **Minimum Account Balance &**
 - **Service Charge Deduction**
- ❖ **Interacts with account and transaction microservice** for checking rules while **transferring** and **withdrawing** from **source account**.



Rules Microservice - Sample request

The screenshot displays a REST client interface for a POST request to `localhost:8090/rules/serviceCharges`. The request headers include `Content-Type: application/json` and an `Authorization` token. The response status is `200 OK`. The response body is a JSON array of three account objects, each with fields for `accountId`, `customerId`, `currentBalance`, `accountType`, `openingDate`, `ownerName`, and `transactions`.

POST `localhost:8090/rules/serviceCharges` Params Send

☒ Content-Type `application/json`

☒ Authorization `eyJhbGciOiJIUzI1NiIsInR5cGU6IjY9eyJzdWIIOjFTVBMT1lFRTEwMSIsImV4cCI6MTYyYnZ...`

New key Value Description

Body Cookies Headers (7) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 [
2   {
3     "accountId": 1000000001,
4     "customerId": "CUSTOMER104",
5     "currentBalance": 5000,
6     "accountType": "Savings",
7     "openingDate": "2021-07-27T00:00:00.000+00:00",
8     "ownerName": "meul",
9     "transactions": null
10  },
11  {
12    "accountId": 1000000002,
13    "customerId": "CUSTOMER103",
14    "currentBalance": 50000,
15    "accountType": "Savings",
16    "openingDate": "2016-09-29T00:00:00.000+00:00",
17    "ownerName": "meul",
18    "transactions": null
19  },
20  {
21    "accountId": 1000000003,
22    "customerId": "CUSTOMER101",
23    "currentBalance": 38500,
24    "accountType": "Savings",
25    "openingDate": "2017-10-27T00:00:00.000+00:00",
26    "ownerName": "meul",
27    "transactions": null
28  }
29 ]
```

AWS Deployment

Task Definition

New ECS Experience
Tell us what you think

Amazon ECS

Clusters

Task Definitions


Account Settings

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions 



Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of your task, what resources they will use, how they are linked together, and which host ports they will use. [Learn more](#)

Create new Task Definition

Create new revision

Actions ▾

Last updated on February 24, 2022 11:08:15 AM (0m ago)  

Status: **ACTIVE** INACTIVE

< 1-12 > Page size 50 ▾

<input type="checkbox"/>	Task Definition	Latest revision status
<input type="checkbox"/>	pod1-authentication	ACTIVE
<input type="checkbox"/>	pod3-authorization	ACTIVE
<input type="checkbox"/>	pod3-insuranceclaim	ACTIVE
<input type="checkbox"/>	pod3-ip-treatment-offerings	ACTIVE
<input type="checkbox"/>	pod3-ip-treatments	ACTIVE
<input type="checkbox"/>	pod4-auth-task-definition	ACTIVE
<input type="checkbox"/>	pod4-consumermicroservice-task	ACTIVE
<input type="checkbox"/>	pod4-policytaskdefinition	ACTIVE
<input type="checkbox"/>	pod4-quotemicroservice	ACTIVE
<input type="checkbox"/>	rahul-auth-task	ACTIVE
<input type="checkbox"/>	sharath-auto-pod1-test	ACTIVE

Video Sample:

Link:

<https://drive.google.com/file/d/1yJYUHcQva0kM8FZU0zRTBUjw5cChOCWD/view?usp=sharing>

Thank You ...!

