



# Retail Banking Management

MFPE (POD - 1) INTCDE21IJ025

**Mentor :** Ayan Dasgupta

**Trainer :** Sanjay Srivatsa

**Coach :** Easvar Varadharajan

## Team Members

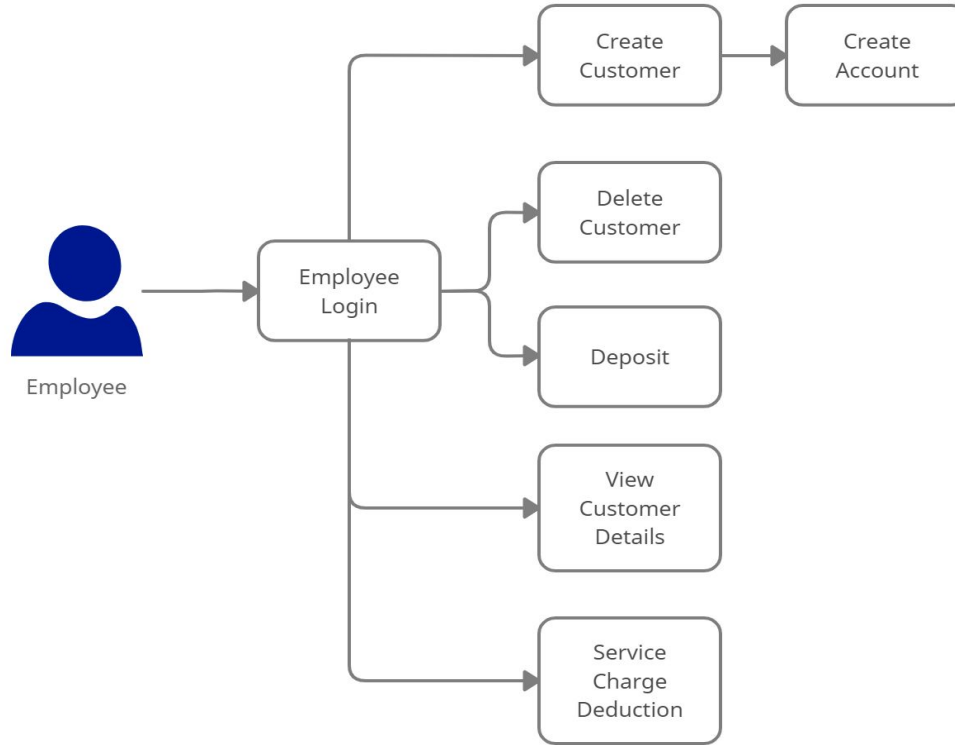
- Bavani S (916589)
- Harish K (916273)
- Mehul Kumar (916542)
- Vemula Chandana (916353)
- Yash Gupta (916190)

# Introduction

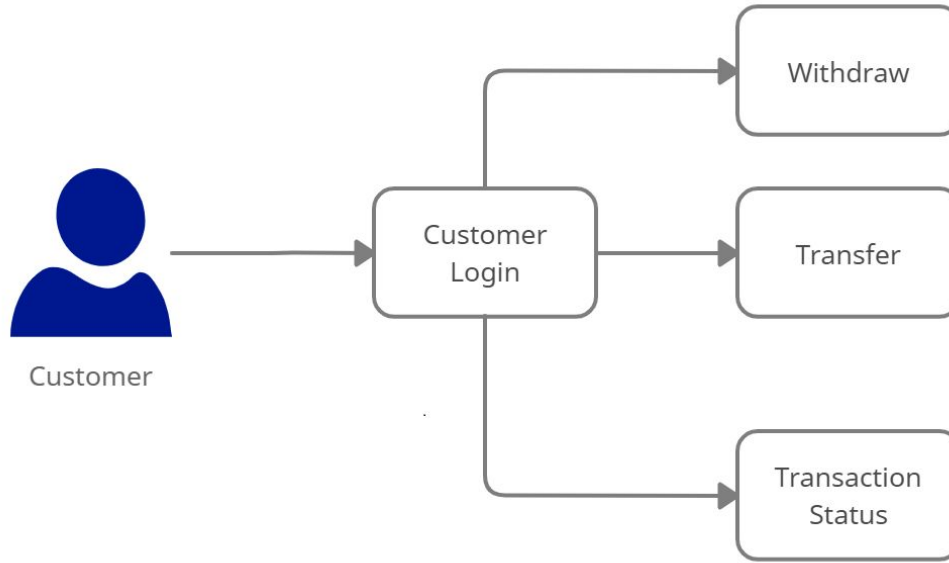
- The Retail Banking System provides **Microservice Architecture** based software solution.
- It provides **high Availability operations** for their **client & bank employees** without disruption.
- **Different Microservices combines together** to perform together as a **Unified application**.



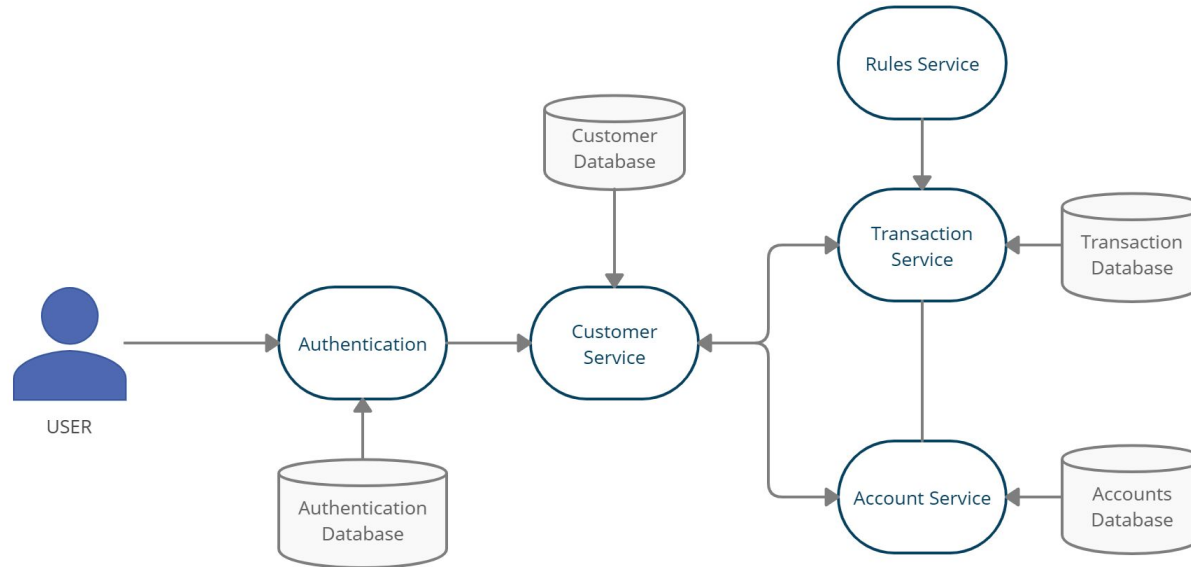
# Architecture Diagram for Employee



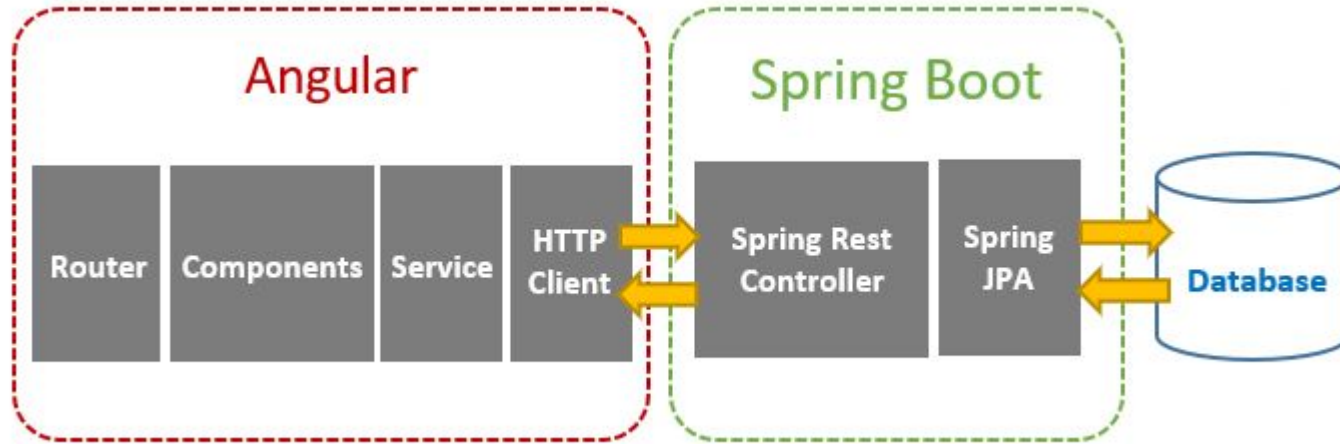
# Architecture Diagram for Customer



# WorkFlow:



# Client - Server Architecture



# Technologies Used

- ❖ Front End : **Angular 8**
- ❖ Backend : **Java, Spring Boot, Restful-Services**
- ❖ Database : **H2 - database**
- ❖ Tools : **Spring Tool Suite, Swagger, Eclemma, 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 **authorisation** based on **user's login**



# Authorization Microservice - Sample request

POST ▼ http://localhost:8084/auth-ms/login Params Send ▼

Authorization Headers (1) **Body** ● Pre-request Script Tests

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

```
1 {  
2   "userId": "EMPLOYEE101",  
3   "password": "emp",  
4   "role": "EMPLOYEE"  
5 }
```

Body Cookies Headers (13) Test Results Status: 202 Accepted

Pretty Raw Preview JSON ▼ ≡

```
1 {  
2   "userId": "EMPLOYEE101",  
3   "authToken": "eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJFTVBMT11FRTEwMSIsImV4cCI6MTYyNzU1OTY3MywiaWF0IjoxNjI3NTU5MDczfQ.eCva17280QAFQ6cV24Iret5z8GA920yeGEFbQhLQaJgI",  
4   "role": "EMPLOYEE"  
5 }
```

# Customer Microservice

- ❖ Customer microservice will performs **operations like** :
  - Creation and Updation of 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.eyJzdWUiOiJFTVBMT1lFRTEwMSIsImV4cCI6MTYyNz...			
	New key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Pretty

Raw

Preview

JSON

1

2

3

```
{  
  "message": "CUSTOMER DELETED"  
}
```

# 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	*** Bulk Edit
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiJ9.eyJzdWl0OiJFTVBMT1FRTEwMSIsImV4cCI6MTYyNz...		
<input checked="" type="checkbox"/>	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

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

POST localhost:8090/rules/serviceCharges Params Send

☒ Content-Type application/json

☒ Authorization eyJhbGciOiJIUzI1NiJ9.eyJzdWIIOiJFTVBMT1FRTEwMSIsImV4cCI6MTYyZm...

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": "mehul",
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

Code Commit :

Developer Tools > CodeCommit > Repositories

**Repositories** [Info](#)  [Notify](#) [Clone URL](#) [View repository](#) [Delete repository](#) [Create repository](#)

< 1 > 

	Name ▼	Description	Last modified ▼	Clone URL
<input type="radio"/>	mfpe-customer	-	6 hours ago	 HTTPS  SSH  HTTPS (GRC)
<input type="radio"/>	mfpe-transaction	-	6 hours ago	 HTTPS  SSH  HTTPS (GRC)
<input type="radio"/>	mfpe-account	Account module	6 hours ago	 HTTPS  SSH  HTTPS (GRC)
<input type="radio"/>	mfpe-rules	rules module	6 hours ago	 HTTPS  SSH  HTTPS (GRC)
<input type="radio"/>	mfpe-auth	authorization module	11 hours ago	 HTTPS  SSH  HTTPS (GRC)

# AWS Deployment

## Elastic Container Registry :

ECR > Repositories

Private | Public

Private repositories (5)

[View push commands](#) [Delete](#) [Edit](#) [Create repository](#)








	Repository name ▲	URI	Created at ▼	Tag immutability	Scan on push	Encryption type
<input type="radio"/>	<a href="#">ij025-pod1-account-ms-registry</a>	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-account-ms-registry	Jul 30, 2021 06:12:21 PM	Disabled	Disabled	AES-256
<input type="radio"/>	<a href="#">ij025-pod1-auth-ms-registry</a>	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-auth-ms-registry	Jul 30, 2021 01:01:50 PM	Disabled	Disabled	AES-256
<input type="radio"/>	<a href="#">ij025-pod1-customer-ms-registry</a>	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-customer-ms-registry	Jul 30, 2021 06:12:40 PM	Disabled	Disabled	AES-256
<input type="radio"/>	<a href="#">ij025-pod1-rules-ms-registry</a>	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-rules-ms-registry	Jul 30, 2021 06:12:49 PM	Disabled	Disabled	AES-256
<input type="radio"/>	<a href="#">ij025-pod1-transaction-ms-registry</a>	941516478045.dkr.ecr.ap-northeast-1.amazonaws.com/ij025-pod1-transaction-ms-registry	Jul 30, 2021 06:13:12 PM	Disabled	Disabled	AES-256


# AWS Deployment

## Code Build:

Developer Tools > CodeBuild > Build projects

### Build projects Info

  Notify ▼  Start build ▼  View details  Edit ▼  Delete build project  Create build project

Your projects ▼ < 1 > 

	Name ▼	Source provider	Repository	Latest build status	Description	Last Modified
<input type="radio"/>	ij025-pod1-account-ms-build-1	AWS CodeCommit	mfpe-account	✔ Succeeded	Account Service	7 hours ago
<input type="radio"/>	IJ025-POD1-Rules-MS-BUILD	AWS CodeCommit	mfpe-rules	✔ Succeeded	Rules Module	7 hours ago
<input type="radio"/>	IJ025-Pod1-Customer-ms-build-2	AWS CodeCommit	mfpe-customer	✔ Succeeded	Customer Module	7 hours ago
<input type="radio"/>	ij025-Pod1-transaction-ms-build	AWS CodeCommit	mfpe-transaction	✔ Succeeded	Transaction module	9 hours ago
<input type="radio"/>	ij025-pod1-auth-ms-build	AWS CodeCommit	mfpe-auth	✔ Succeeded	auth module	3 days ago

# AWS Deployment

## Load Balancer:

[Create Load Balancer](#) [Actions](#)

	Name	DNS name	State	VPC ID	Availability Zones	Type
<input checked="" type="checkbox"/>	ij025-pod1-loadbalancer	ij025-pod1-loadbalancer-491...	Active	vpc-7ca75a1a	ap-northeast-1a, ap-nor...	application

Load balancer: **ij025-pod1-loadbalancer**

Description

Listeners

Monitoring

Integrated services

Tags

**Basic Configuration**

<b>Name</b>	ij025-pod1-loadbalancer
<b>ARN</b>	arn:aws:elasticloadbalancing:ap-northeast-1:941516478045:loadbalancer/app/ij025-pod1-loadbalancer/1ea1498101553390 <a href="#">🔗</a>
<b>DNS name</b>	ij025-pod1-loadbalancer-491001860.ap-northeast-1.elb.amazonaws.com <a href="#">🔗</a> (A Record)
<b>State</b>	Active
<b>Type</b>	application

# AWS Deployment

Cluster:

## Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

Create Cluster

Get Started

View

list

card

view all



1 - 1 of 1



ij025-pod1 >

CloudWatch monitoring

✓ Default Monitoring

FARGATE

5

Services

5

Running tasks

0

Pending tasks

EC2

0

Services

0

Running tasks

0

Pending tasks

No data

CPUUtilization

No data

MemoryUtilization

0

EC2 container instances

EXTERNAL

# AWS Deployment

## Cluster Services:

### Cluster : ij025-pod1

Get a detailed view of the resources on your cluster.

**Cluster ARN** `arn:aws:ecs:ap-northeast-1:941516478045:cluster/ij025-pod1`

**Status** **ACTIVE**

**Registered container instances** 0

**Pending tasks count** 0 Fargate, 0 EC2, 0 External

**Running tasks count** 5 Fargate, 0 EC2, 0 External

**Active service count** 5 Fargate, 0 EC2, 0 External

**Draining service count** 0 Fargate, 0 EC2, 0 External

Services

Tasks

ECS Instances

Metrics

Scheduled Tasks

Tags

Capacity Providers

Create

Update

Delete

Actions ▾

Filter in this page

Launch type ALL ▾

Service type ALL ▾

<input type="checkbox"/>	Service Name	Status	Service type	Task Definition...	Desired tasks
<input type="checkbox"/>	<a href="#">ij025-Pod1-Customer-ms-service</a>	ACTIVE	REPLICA	<a href="#">ij025-Pod1-Cust...</a>	1
<input type="checkbox"/>	<a href="#">ij025-pod1-transaction-ms-service</a>	ACTIVE	REPLICA	<a href="#">ij025-pod1-trans...</a>	1
<input type="checkbox"/>	<a href="#">ij025-pod1-auth-ms-service</a>	ACTIVE	REPLICA	<a href="#">ij025-pod1-auth...</a>	1
<input type="checkbox"/>	<a href="#">IJ025-Pod1-Rules-ms-service</a>	ACTIVE	REPLICA	<a href="#">IJ025-Pod1-Rul...</a>	1
<input type="checkbox"/>	<a href="#">ij025-pod1-account-ms-service</a>	ACTIVE	REPLICA	<a href="#">ij025-pod1-acco...</a>	1



# Unit testing and Code Coverage:

The screenshot displays an IDE window for a project named 'Rules-MS'. The top toolbar includes 'File', 'Edit', 'Source', 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The 'Package Explorer' on the left shows the project structure, and the 'JUnit' tab displays a list of test results. The 'RulesController.java' file is open in the editor, showing a REST controller with a single endpoint. The 'Coverage' tab at the bottom provides a detailed breakdown of code coverage for the 'Rules-MS' element.

JUnit Test Results:

- Finished after 19.268 seconds
- Runs: 72/72
- Errors: 0
- Failures: 0

Test List:

- AccountInputTest [Runner: JUnit 5] (0.004 s)
- AuthenticationResponseTest [Runner: JUnit 5] (0.004 s)
- TransactionTest [Runner: JUnit 5] (0.031 s)
- RulesInputTest [Runner: JUnit 5] (0.016 s)
- ServiceResponseTest [Runner: JUnit 5] (0.001 s)
- ErrorDetailsTest [Runner: JUnit 5] (0.002 s)
- AppUserTest [Runner: JUnit 5] (0.012 s)
- MinimumBalanceTest [Runner: JUnit 5] (0.188 s)
- RulesServiceApplicationTests [Runner: JUnit 5] (0.455 s)
- ServletInitializerTest [Runner: JUnit 5] (2.538 s)
- RulesServiceImplTest [Runner: JUnit 5] (0.036 s)
- RulesControllerTest [Runner: JUnit 5] (4.271 s)

Code Coverage for Rules-MS:

Element	Coverage	Covered Instru...	Missed Instru...	Total Instru...
Rules-MS	91.6 %	1,607	148	1,755

```
1 package com.ruleservice.controller;
2
3 import java.util.List;
4
5 @RestController
6 @CrossOrigin()
7 public class RulesController {
8
9     private static final String INVALID = "Send Valid Details.";
10
11     @Autowired
12     public RulesService rulesService;
13
14     @Autowired
15     AccountFeign accountFeign;
16
17     @PostMapping("/evaluateMinBal")
18     public ResponseEntity<?> evaluate(@RequestBody RulesInput account) throws MinimumBalanceException {
19         if (account.getCurrentBalance() == 0) {
20             throw new MinimumBalanceException(INVALID);
21         } else {
22             boolean status = rulesService.evaluate(account);
23
24             return new ResponseEntity<Boolean>(status, HttpStatus.OK);
25         }
26     }
27 }
```



Video Sample:



Thank You ...!

