

INSURANCE POLICY MANAGEMENT

PRESENTED BY

SWETHA M K

T R AKSHAY (TEAM LEAD)

TAMALIKA DEY

VEERAVALLI DEEPIKA

ABESH DAS

ABHISKEK PRABHU KANNAN

AISHWARYA P R

B MOUNIKA

DASARI MANJULA

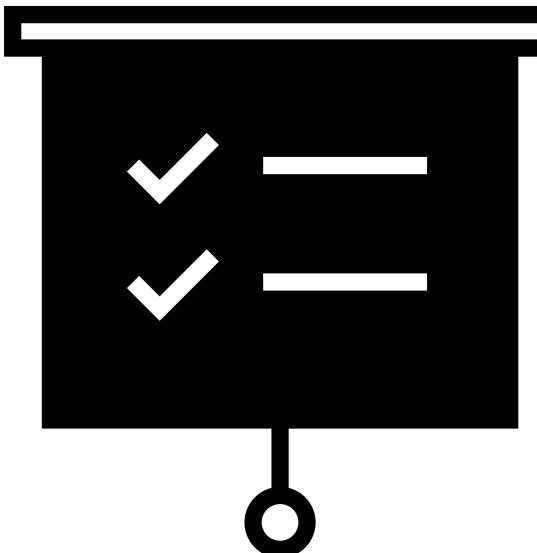
GANGAVARAPU AKHILA

230 JAVA SL KK - RLL - GROUP 4

DATE: 12-09-2023

INTRODUCTION

- In an uncertain world, where tomorrow's events are unpredictable and can affect our health and possessions, insurance stands as a shield of financial security.
- Recognizing the significance of insurance, Codebun has embarked on a journey to develop an innovative Insurance Management System. This system will cater to the needs of both administrators and customers, providing a seamless experience for managing insurance-related activities.
- The system is designed with an efficient Java Microservice backend using SpringCloud Architecture. The core roles within this system are the administrators and customers, each with specific functionalities and responsibilities.
- Underpinning the system's seamless operation are Feign Clients, facilitating smooth service-to-service communications within the microservices. Moreover, the system leverages the prowess of Resilience4J to ensure robust CircuitBreaker functionality, while Sleuth and Zipkin enhance traceability and monitoring, contributing to a comprehensive insurance management experience.



PROJECT OVERVIEW

- **Domain:** Insurance
- **Project Objective:** Streamline insurance management with a modern, microservices-driven platform for administrators and customers.



TOOLS AND TECHNOLOGIES

SERVER-SIDE TECHNOLOGIES:

- Spring Boot
- Spring Cloud
- Spring Data JPA

SERVER:

- Tomcat

DEVOPS TOOLS:

- Docker
- GitHub

API TESTING TOOLS

- Postman
- Boomerang API

DATABASE MANAGEMENT:

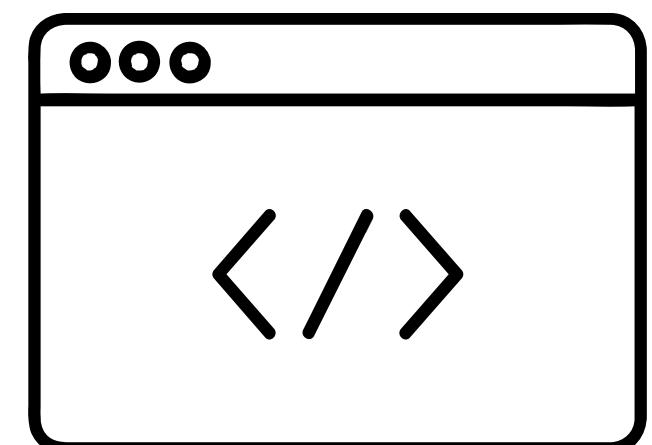
- MySQL
- MySQL Workbench

INTEGRATED DEVELOPMENT ENVIRONMENTS (IDE'S):

- Spring STS

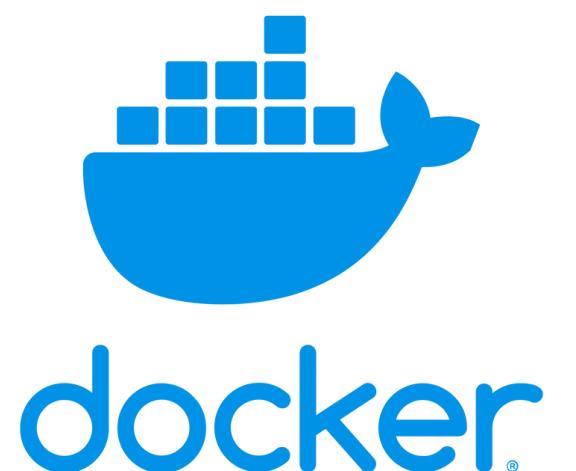
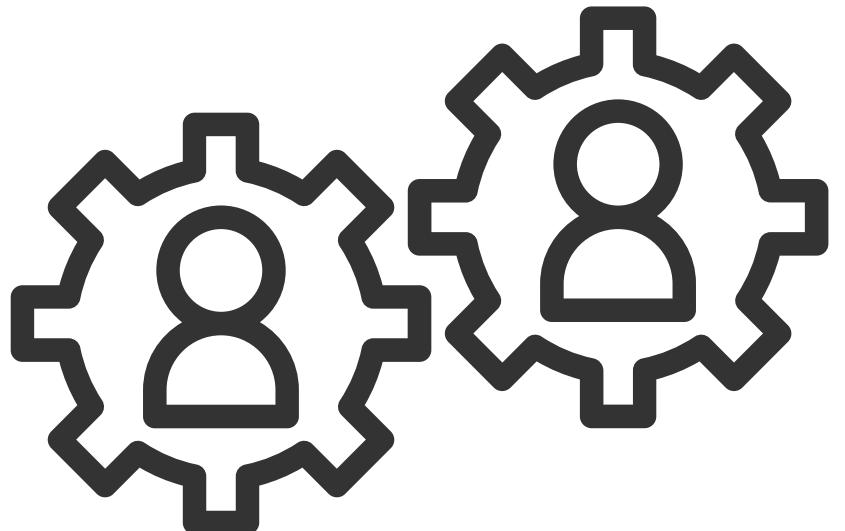
SPRING CLOUD COMPONENTS

- Spring Cloud Config
- Spring Cloud Netflix Eureka
- Feign
- Resilience4j
- Retry
- Sleuth
- Zipkin



FUNCTIONALITY HIGHLIGHTS

- 1. Robust Backend:** Java Spring Microservices
- 2. Role-Based Access:** Admin and Customer Roles
- 3. Service Communication:** Feign Client Integration
- 4. Fault Tolerance:** Resilience4J for Reliability
- 5. Error Tracking:** Sleuth & Zipkin
- 6. Configurability:** Central Spring Cloud Config
- 7. DevOps Integration:** Docker & GitHub
- 8. Database Management:** MySQL



LIST OF MICROSERVICES

Microservices with Databases

1. **Insurance-Address**
2. **Insurance-Admin (Feign-enabled)**
3. **Insurance-Contact-Form**
4. **Insurance-Customer (Feign-enabled)**
5. **Insurance-Customer-Queries**
6. **Insurance-Email**
7. **Insurance-Policy**
8. **Insurance-Policy-Categories**

Microservices for servers

1. **Insurance-Config-Server**
2. **Insurance-Eureka-Server**



TABLES PRESENT IN THE MICROSERVICES (CREATED BY ORM)

Insurance-Address

Database Name: insurance_address_db
Table: address

Attributes: Created by ORM

<u>address_id</u>	bigint not null
city	varchar(255)
country	varchar(255)
postal_code	varchar(255)
state	varchar(255)
street_address	varchar(255)

Insurance-Admin

Database Name: insurance_admin_db
Table: admin_user

Attributes: Created by ORM

<u>admin_id</u>	bigint AI PK
admin_email	varchar(255) not null
admin_name	varchar(255)
admin_password	varchar(255)

Insurance-Contact-Form

Database Name: insurance_contact_form_db
Table: contact_form

Attributes: Created by ORM

<u>form_id</u>	bigint AI PK
email	varchar(255)
full_name	varchar(255)
message	varchar(255)

Insurance-Customer-Queries

Database Name: insurance_question_db
Table: question

Attributes: Created by ORM

<u>question_id</u>	bigint AI PK
answer	varchar(255),
customer_id	bigint,
question	varchar(255),

Insurance-Customer

Database Name: insurance_customer_db
Table: customer

Attributes: Created by ORM

<u>customer_id</u>	bigint AI PK
customer_age	integer not null,
customer_email	varchar(255) not null,
customer_gender	varchar(255),
customer_image_url	varchar(255),
customer_mobile	varchar(255),
customer_name	varchar(255),
customer_password	varchar(255),

Insurance-Policy

Database Name: insurance_policy_db
Tables: policy, issue_policy

Attributes: Created by ORM

<u>policy_id</u>	bigint not null AI PK
policy_coverage_amount	decimal(19,2)
policy_coverage_limit	decimal(19,2)
policy_coverage_period_months	integer
policy_name	varchar(255)
policy_premium	decimal(19,2)

<u>issue_policy_id</u>	bigint not null AI PK
customer_id	bigint
policy_date_time	datetime
policy_id	bigint
policy_status	integer not null

Insurance-Policy-Category

Database Name: insurance_policy_category_db
Tables: policy_category, policy_category_policy_ids (Created by
@ElementCollection annotation, dynamically)

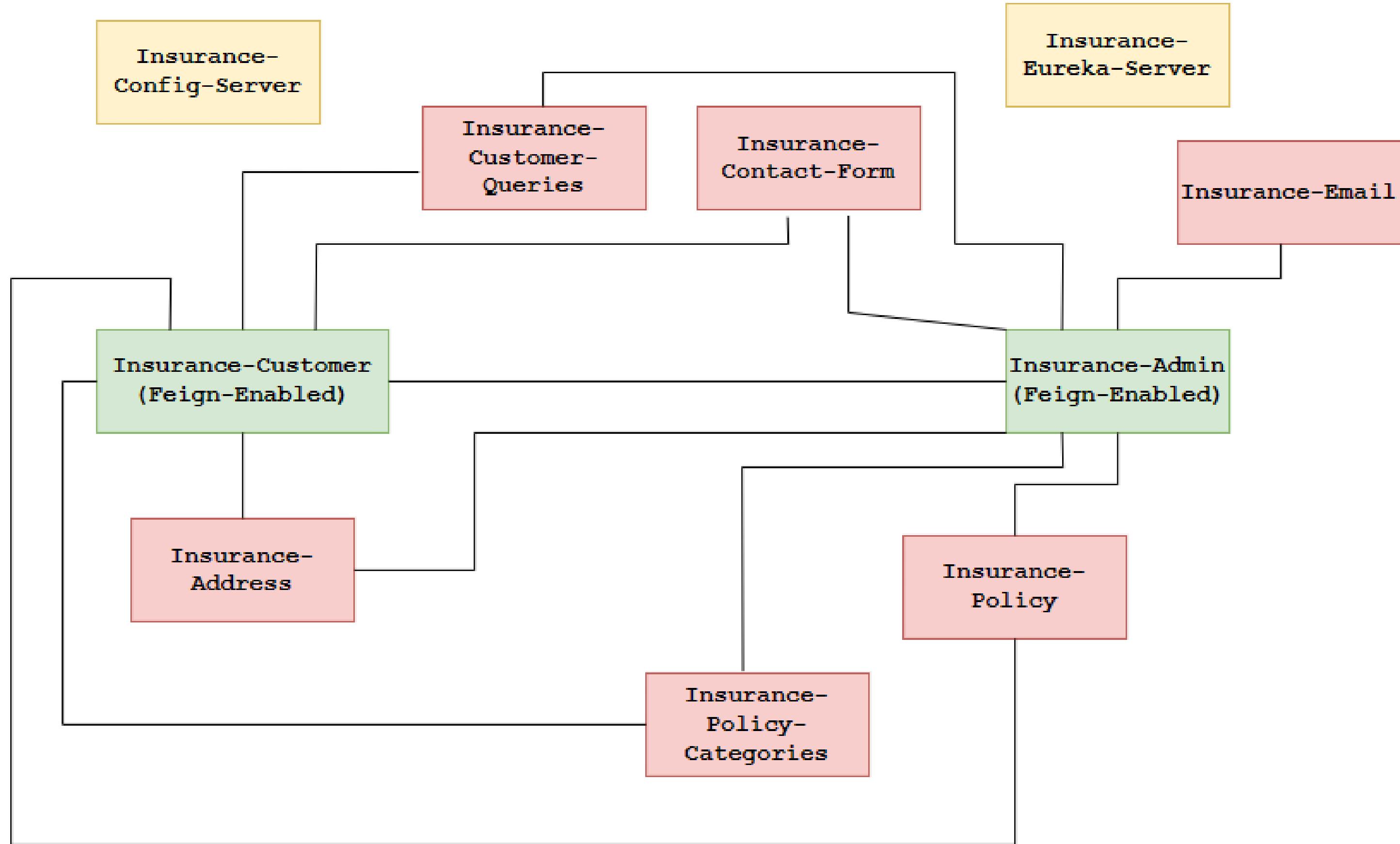
Attributes: Created by ORM

<u>category_id</u>	bigint not null auto_increment,
category_name	varchar(255)
policy_category_category_id	bigint not null,
policy_ids	bigint



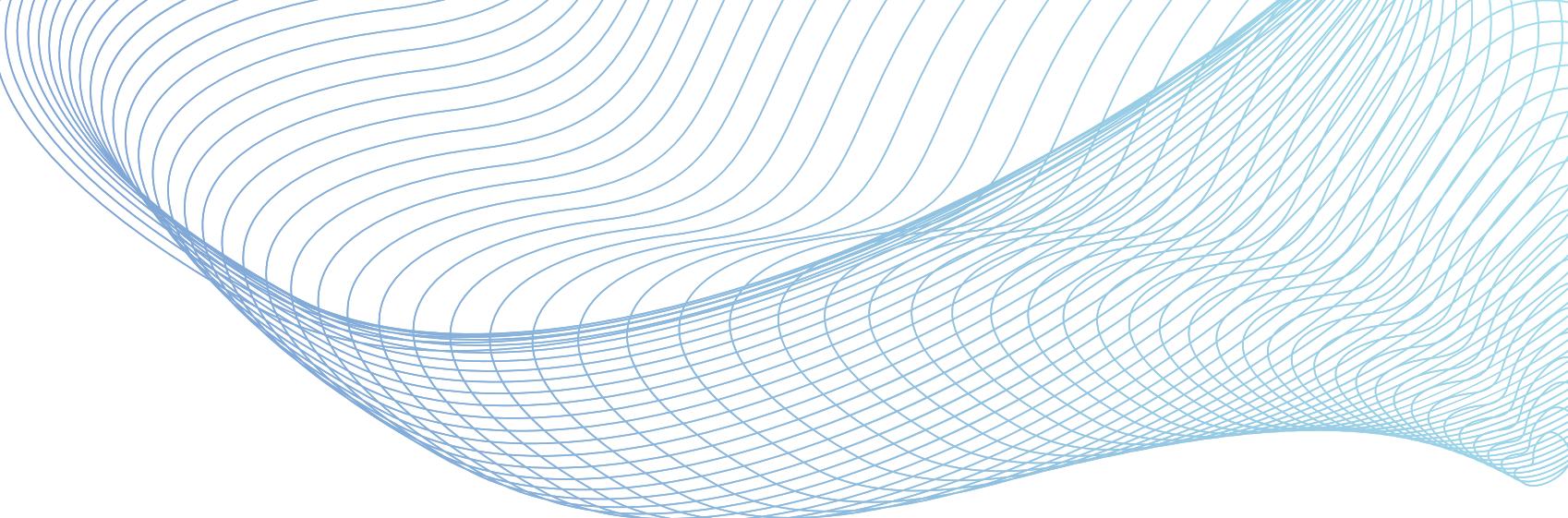
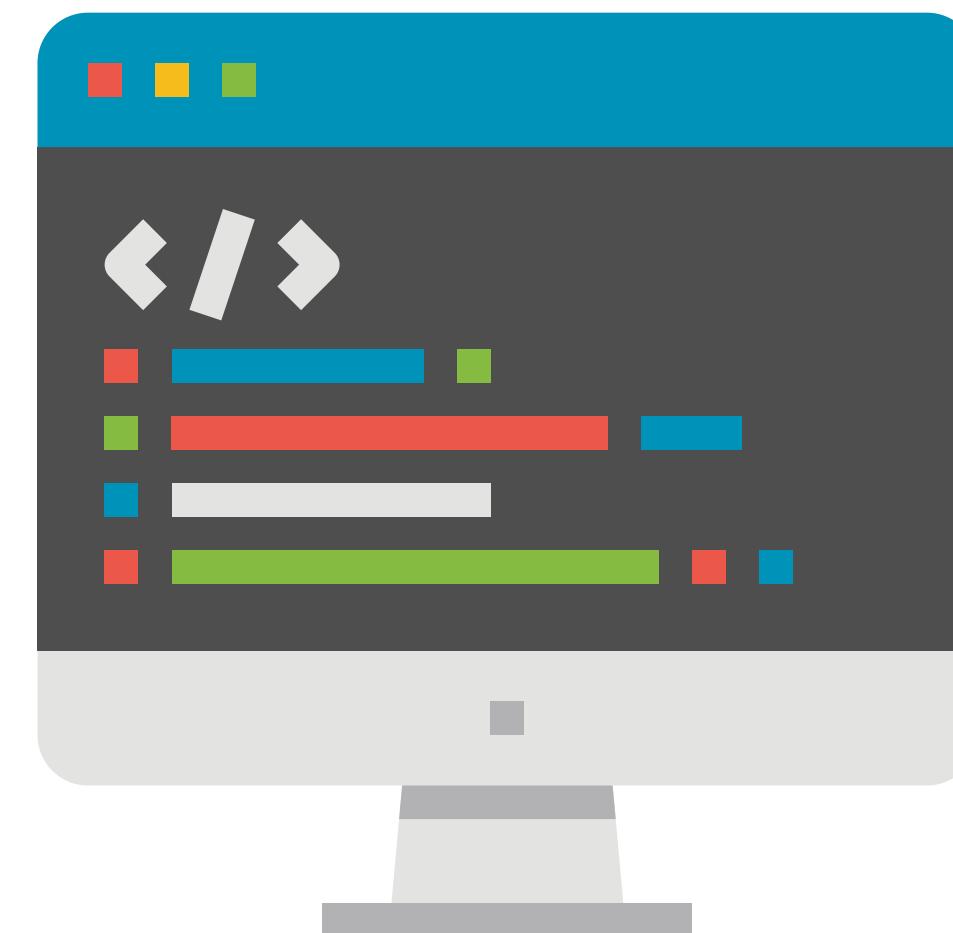
MySQL

MICROSERVICES ARCHITECTURE OVERVIEW





DEMONSTRATION



CONCLUSION

In conclusion, the Insurance Management System is a robust and innovative solution designed to streamline insurance-related activities. With an efficient backend, and integration of cutting-edge technologies, this system is poised to revolutionize insurance management. It offers enhanced functionality, seamless service, and scalability, making it a valuable asset for insurance administrators and customers alike. As we move forward, we are committed to continually enhancing and expanding this system to meet the evolving needs of the insurance industry.





THANK YOU

