Employee Management System (EMS)

Team Members:

Name	Area
Ajay Bondugula Ajay.Bondugula@Intinfotech.com	Department service
Bhagath Sunkara Bhagath.Sunkara@Intinfotech.com	Organization service
Kishor Arya Kishor.Arya@Intinfotech.com	Employee service
Pankaj Kapse Pankaj.Kapse@Intinfotech.com	Devops
Sachin Prabhu Sachin.Prabhu@Intinfotech.com	Devops
SyedS Alam SyedS.Alam@Intinfotech.com	Frontend

Requirement:

Create a project using microservices based architecture. Implemented the program using Java spring boot. Create a frontend using React/Angular framework. The project should include required component testing and other best practices. Deploy the application to one of the three major cloud providers.

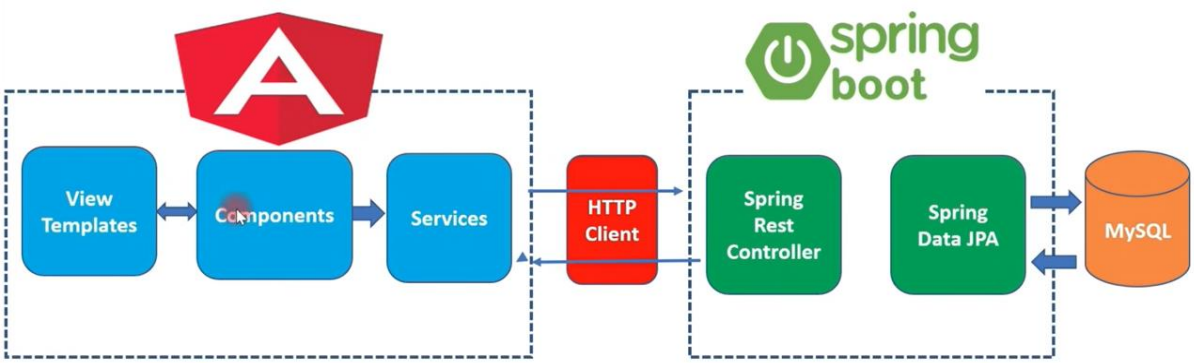
Overview:

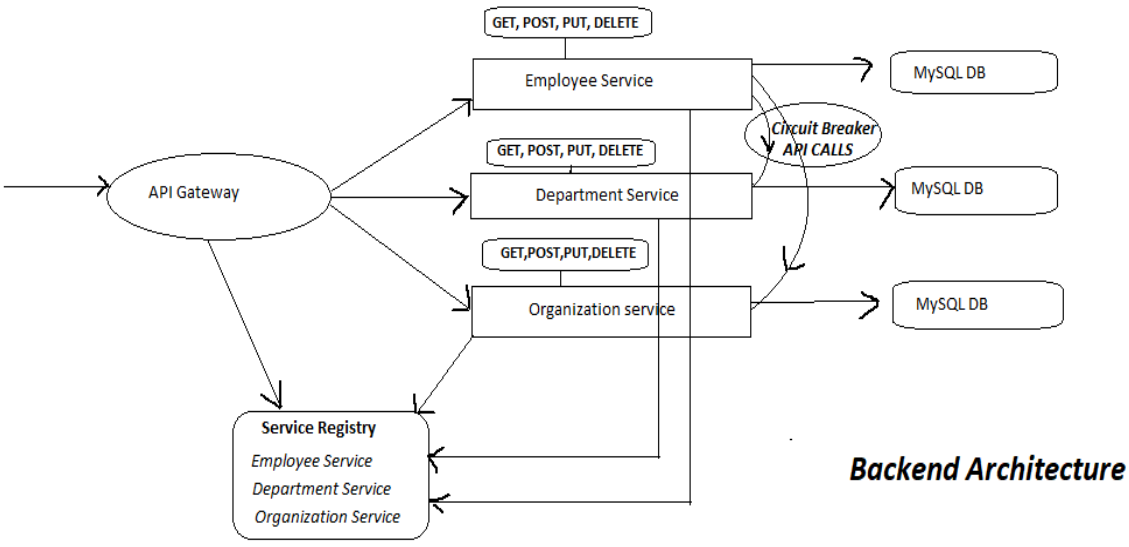
We have created an Employee Management System (EMS) using Java spring-boot. We have implemented three micro-services. We are currently progressing with the application frontend design using Angular. We are implementing testing in the project. As backend is already implemented, we will be deploying this code to cloud over next 4-5 days.

Tools and Technologies

Technologies	Spring boot
	Microservices
	Spring Cloud
	data JPA
	MySQL Database
	GCP Cloud
	AngularJS
Tools	STS
	MySQL Workbench
	Web browse
	Post man
	GCP Web platform

Architecture:





Backend Architecture

Micro-services:

Employee Service:

a module containing the first of our sample microservices that allows to perform CRUD operation on in-memory repository of employees and storing the details of Employees.

Department Service:

a module containing the second of our sample microservices that allows to perform CRUD operation on in-memory repository of departments. It communicates with employee-service.

Organization Service:

a module containing the third of our sample microservices that allows to perform CRUD operation on in-memory repository of organizations. It communicates with both employee-service and organization-service.

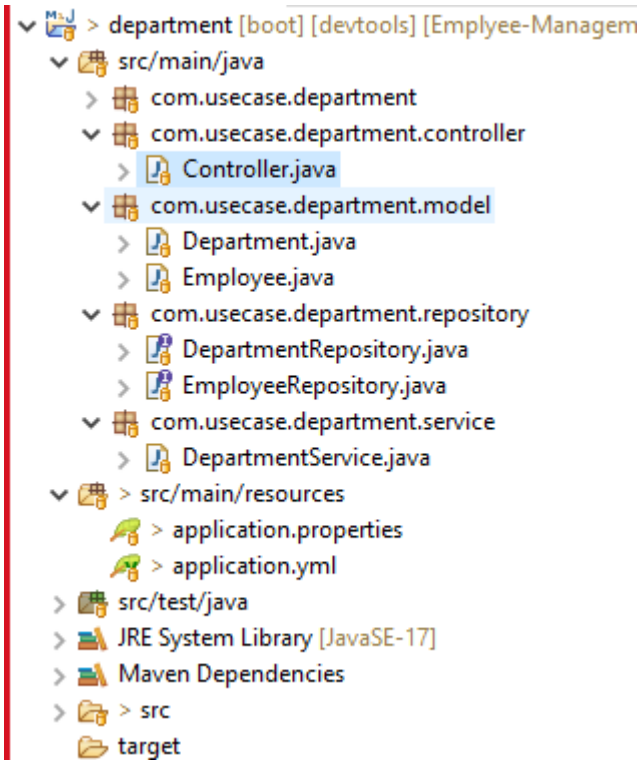
Gateway service:

a module that Spring Cloud Netflix Zuul for running Spring Boot application that acts as a proxy/gateway in our architecture.

Eureka Server:

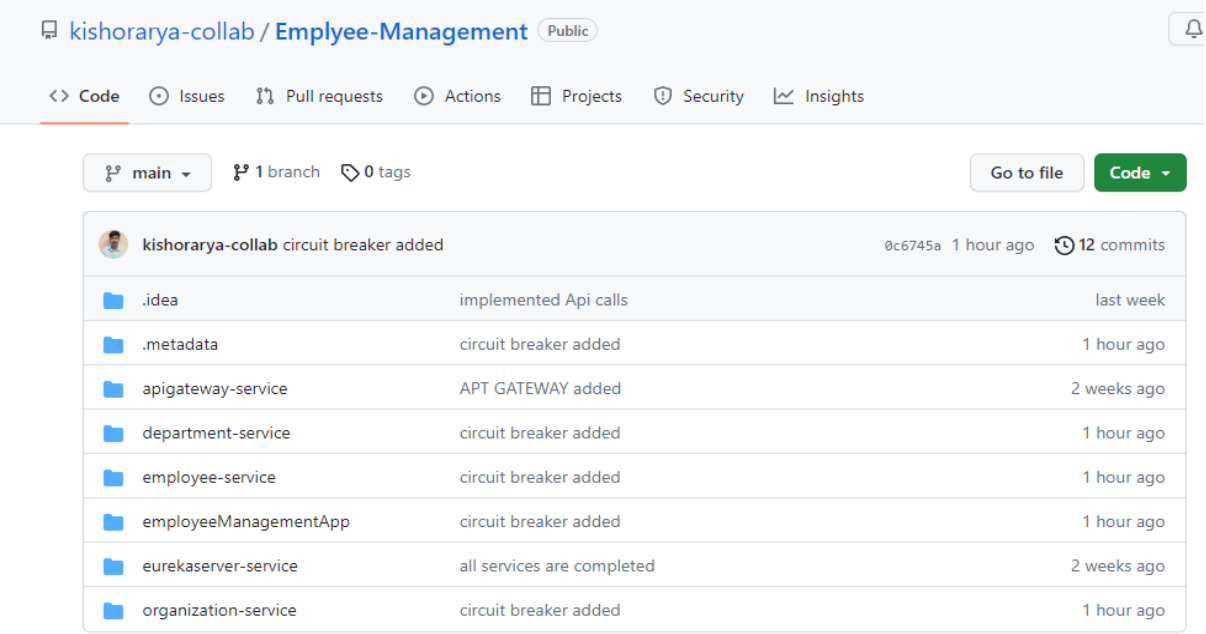
a module that registers all instances of all the client microservices.

Services Structure:



Version Control:

We have used **Github** for version control and given access to all the team members.



Given the concerns raised by Vinay over public repository, we will use **Google Cloud Source Repositories** as alternative.

Google Cloud Platform:

We are going to use Google cloud platform.

Springboot –

DB -