Name :Gajendran G
Email :gajendrangnanasekaran@gmail.com
Phone:9788430933,9944654778

Summary:

- 5.8 years of experience in Enterprise Java application development
- Possess insightful knowledge on **Java, Spring/Springboot, Apache Kafka, Elastic search & CI/CD, Microservice architecture**
- Experienced in evaluating and identifying open source products/solutions.
- Good in developing distributed and scalable applications using various technologies
- Experienced in Domain Driven Design, agile and scrum based development
- Good in training the juniors

Skills:

Programming Languages: Java ,C, C++,HTML.
Frameworks/Libraries: Spring Boot/Spring Cloud, JPA/Hibernate(ORM),Kafka, ElasticSearch, Keycloak,Redis/Hazelcast,Log4j
Tools: Eclipse, Git, Maven,Jira,Bitbucket.
Platforms: Linux.
Cloud: AWS.

Education:

2011,MCA. from Bharathidasan University, Tamil Nadu.
2008,B.Sc. (IT) from Annai College of Arts and Science, Kumbakonam, Tamil Nadu**.**

Isha Foundation - Sr. Java developer Nov 2015 - till date, Coimbatore, India

- **Common Cross cutting functionalities:**
- Developed and used following functionalities in most of the projects
- Implemented validation using **JSR 303** and also developed **custom annotations**.
- Implemented entity auditing using **Javers** to track the domain objects changes
- Implemented custom **JSONB data type** using Hibernate user type and handled JSONB operations
- Used **MapStruct** to simplifies the implementation of mappings between java beans (DTO to Business Object and vice versa)
- Used **Swagger** to expose API description
- Used build/deploy tools such as **Git, Maven, Jenkins, Docker** for Continuous Integration & Deployment (**CI/CD**)
- Parallel to development, developed test cases using **Junit** and **Mockito**
- Integrated **Keycloak SSO** and **authorization** functionalities.

Contact Data Integration 2.0:

- ***Contact MDM***: As a part of the team, developed contact Master Data Management (MDM) repository.
  - Integrated contact data scattered across multiple systems, departments and countries.
  - As a developer designed and developed ETL contact microservices using **Springboot** and **Kafka**
  - Exposed **REST APIs** to get single source of truth of a contact
- **Match Engine**: This component is responsible for performing Contact Deduplication. It provides recommendations about the contact matches. The challenge here was to match contacts data which was not very clean and was prone to typo errors.
- As part of the team:
  - Designed highly scalable, Modular, Real-time solution.
  - Developed four Modules/Stages each for **Indexing**, **Bucketing**, **Matching** & **Persisting**.
- **Searching**: Developed contact search component which is customizable, scalable and fast. Developed search APIs with **Elasticsearch** that supports several search requirements despite the quality of the data being not very clean.
- **Inference Service**: This service is responsible for inferring missing or incomplete attributes of a Contact (like Gender, Address, etc).

Transaction Store:

As part of the team, designed and developed near real-time transaction **Data Lake**. The transactions are donations, programs, volunteering information and much more.

- Defined data source format and distributed to the stakeholders.
- Used **Kafka Connect** and defined data connector configurations to pull source system data from databases like MySQL, PostgresSQL.
- Developed multiple **Kafka Consumers** to extract data from different Kafka topics
- Developed consumer configuration manager that manages starting and stopping of consumers. Also it manages the scalability of consumers through configuration.
- Developed microservice to retrieve transactional data.

Sangam:

This software provides a 360 degree view of donors who engages with the organization across departments. A unified view of a donors’ engagements and transactions is created by interacting with other in-house applications like programs, donation management system, volunteer management system, etc. An analytics engine is built on top of this to data to selectively nurture donors interest.

- Designed and developed sangam Aggregator microservice to provide a unified view of donors’ engagements and transactions.
- Used **Eureka** for registering and deregistering the microservices.
- Used **Config Server** for centralized configuration.
- Used **Zuul** as the microservice gateway.

- Actively involved in discussion to build **BigQuery** based analytical engine to be built on top of sangam data.

Tally Facade (Finance integration with TallyERP)

Tally ERP doesn’t provide a way to integrate third-party applications. So designed and developed Tally facade application using pipeline architecture. As part of this, the received message goes through stages like validation and XML conversion before posted to Tally ERP.

- Designed and developed a **Kafka Rest Proxy** to receive messages from third-party applications. Received messages are sent Kafka request topic.
- Developed validator consumer that validates JSON Schema and data validation. On error, the messages are sent to the error topic with proper error codes.
- Developed message translator consumer to convert JSON messages and XML message using **Thymleaf** template, as it is required by the Tally ERP. On error, error message will be generated and sent to the error topic.
- Developed Tally facade consumer that consumes the XML message and post it to Tally ERP. On Success, the response are sent to response topic and on Error, sent to error topic.
- Involved in design discussions of response and error path which were developed by the rest of the team

Varicon Technologies -Java developer April 2014 - oct 2015, Coimbatore, India

Purchase Automation System (PAS)

PAS developed for automating the purchase process of enterprises level organisation.It reduces manual process involved between various departments within an organization. Its web service based product and it provide RESTFull API’s for PAS functionalities to simplify the purchase process.

- As a developer Implemented RESTFul API's using **Apache-Jersey** Rest based on JAX-RS standards.
- Used **Spring Framework** for Dependency Injection.
- Implemented data persistence using JPA and Hibernate ORM as persistence provider.