

LOHITH KUMAR NEERUKONDA

Software Engineer



(469) 774-0075

lohithkumarneerukonda@gmail.com

Southfield, MI

[Linkedin](#)

[Github](#)

SUMMARY

- Results-oriented, hands-on Micro Services specialist with around **3 years** of IT experience including software lifecycle experience - requirements, solution (Cloud & On-prem) architecture, design & development, testing, governance, delivery, and management of mission-critical enterprise applications using distributed systems and manage the integration of heterogeneous enterprise systems at scale.
- Worked in Self Service model where the Agile team/developer is responsible for the design, development, testing, deployment, and maintenance of the application.
- Expertise in building applications using **Spring Boot, Spring AOP, Spring MVC, Spring Scheduling, and Spring JMS**.
- Cloud computing experience using Amazon Web Services: **EC2, S3, Elastic Container Service, ECR, Route 53, Auto Scaling Groups, Lambda, SQS, Bedrock**.
- Having Hands-on experience in creating the **Docker** file for building the docker images for REST services.
- Experience in creating the **JMeter** scripts for measuring the REST services performance.
- Hands-on experience in improving the REST services performance by implementing **Asynchronization and Caching**.
- Experience in design, develop and deployment of **Micro Services** on AWS environment.
- Developed front-end dynamic views and templates using JSP, and HTML/CSS/JavaScript, enhancing the overall user experience.
- Developed **CICD Pipelines for Continuous integration** using Maven, Jenkins, and GitHub.
- Implemented **Hystrix Circuit Breaker** pattern to allow the service to continue operating when a downstream service fails.
- Implemented **Mustache** templates for dynamic content generation.
- Experience in querying/monitoring service in **DataDog** and Developed **Kafka** producer to publish messages.
- Experience in **Agile & Waterfall** SDLC Methodologies.
- Expertise in **J2EE** Object-Relational Mapping frameworks like JPA.
- Good knowledge in working with databases like **MySQL**, PostgreSQL, and Oracle.
- Used **Junit, Mockito and Hamcrest** for unit testing.

TECHNICAL SKILLS

Java & J2EE Technologies	Java 8, 11 & 17
Frameworks	Spring Boot Framework, Spring Scheduling, Spring AOP(AspectJ), Spring Batch, Spring-JMS
DAO Frameworks	JDBC, DAO, Spring JDBC Template
Frontend Technologies	HTML, CSS, JavaScript, Nodejs
Tools & Others	UML, Slf4j, Log4j2, Lucidchart, VISIO, Check Style, Sonar, Jenkins, Maven, Junit, Kubernetes, Tableau, Apache Kafka , ActiveMQ, RabbitMQ, Postman, Mustache, Hibernate, Cursor, Vercel, Open AI
Cloud	AWS Cloud, Microsoft Azure
Databases	SQL Server, MySQL , PostgreSQL
Version Control	Bitbucket, Git, CVS, GitHub

EDUCATION

Master of Science in Information Systems from University of North Texas, Denton, TX	01/2023 – 05/2024
Bachelor's of Engineering in IT from Andhra University, Visakhapatnam, Andhra Pradesh, India.	07/2018 – 05/2022

CERTIFICATIONS

- Microsoft Certified: Azure Developer Associate [Certification](#)

PROFESSIONAL EXPERIENCE

Client: Anywhere Real Estate Inc., New York, NY (Remote)

06/2024 – Present

Project: AI Capabilities for Real Estate Automation

Role: Java Developer

Developed a suite of AI-powered REST services utilizing AWS Bedrock to streamline real estate processes. The core feature of this project involved automating property description generation through image and data analysis. Key highlights include:

- **Tag and Caption Generation:** Implemented REST services that analyze property images uploaded by agents/photographers, leveraging AWS Bedrock to generate relevant tags and captions automatically.
- **Property Description Automation:** Designed a service to process these tags, captions, and listing details, automatically generating property descriptions, significantly reducing manual effort for coordinators.

This AI-driven solution enhanced productivity and minimized manual intervention of coordinators in processing the order.

Responsibilities:

- Developed and maintained **RESTful API** for property description generation.
- Leveraged **AWS Bedrock AI** service to analyze property images and generate tags and caption.
- Developed microservices using Java and Spring Boot and built the **Docker** image and deployed it in ECS container.
- Developed a component to get the image from AWS S3 and used for tags and caption generation.
- Created service to generate property description using the listing details, tags and captions.
- Designed and maintained database schemas to efficiently by using Flyway.
- Utilized the Spring async feature for processing the images in parallel. It helped increase performance by 30%.
- Conducted performance testing for measuring the throughput using **JMeter** and **Grafana**.
- Participated in **Agile methodologies**, including sprint planning, daily stand-ups, and retrospectives, to iteratively deliver features.
- Implemented comprehensive **logging** for all API interactions and processing steps to facilitate troubleshooting and monitoring.

Environment: Java 17, **Spring Boot**, Spring Rest Framework, **Spring AOP**, **Slf4j**, Junit, **Mockito**, JIRA, **Agile**, AWS ECS, AWS Cloud, **S3**, **AWS Bedrock**, **Docker**, MySQL, Maven, JMeter, Grafana.

Company: Shell/ Greenlots, Bay Area, CA (Remote)

05/2023 – 01/2024

Project: Notification Service

Role: Systems Engineer

Developed a REST service that efficiently manages notification requests by integrating **Twilio** and **SendGrid** for reliable message delivery. The application processes request details, retrieves user information from an external **User Service**, and sends personalized notifications based on user preferences. Key features include dynamic message generation using **Mustache** templates. Developed a comprehensive **logging** system for auditing and retry mechanisms for handling delivery failures.

Additionally, the application enhanced user engagement through app push notifications and included user preferences management, along with optimized scheduling for timely updates, improving communication efficiency and delivering a better user experience.

Responsibilities:

- Designed multiple microservices to handle notification requests, ensuring scalability and performance.
- Developed front-end dynamic views and templates using JSP, and HTML/CSS/JavaScript, enhancing the overall user experience.
- Created custom controllers, view resolvers, and model attributes to efficiently handle user requests, responses, and interactions.
- Implemented robust form validation and data binding mechanisms using **Spring MVC** validators to ensure data integrity and seamless user input handling.
- Collaborated with the external user service to retrieve user information and preferences, ensuring accurate and timely notification delivery.
- Integrated Twilio and SendGrid for reliable message delivery.
- Utilized Mustache templates to implement dynamic message generation, allowing personalized notifications based on user data.
- Created a Quartz scheduler to retry sending failed notifications at specified intervals.
- Improved performance by introducing in-memory caching and asynchronous processing.
- Implemented a CI/CD pipeline using Docker to automate build, test, and deployment processes.
- Developed integration and unit test cases using JUnit and Mockito frameworks.

Environment: Java 11, Spring Rest Framework, Spring JPA, Twilio, SendGrid, App Push Notifications, Mustache, MySQL, AWS S3, SQS, Maven, JPA, REST API, Docker, **Jira**, GitHub, Bitbucket, Sonar, **Apache Tomcat**, **HTML**, **CSS**, **JavaScript**, **JSP**.

Company: Infosys, Hyderabad, India
Project: Device Confidence
Role: Systems Engineer

07/2022 – 12/2022
Hyderabad, India

The **Device Confidence API** will create an additional layer of security between clients and services for high-risk transactions. It will leverage an external API to validate customer and device information from the mobile provider's customer data against the organization's customer repository. Based on the results, it will determine whether the device can be trusted and used for OTP services.

Converted device-confidence service business rules to Drools rules and stored in ContetentOne system. Spring scheduler in Device Confidence service will reload the rules every 4hrs. So new/modified rules will be active without application re-deployment. Used Maven for building and managing dependencies of the application. Delivery ownership of the Rest Services and coordination with different client teams to make integrated releases.

Responsibilities:

- Developed a **REST** service using the Spring Framework and integrated it with various private and public REST services.
- Improved the REST service performance by making **parallel REST service calls and caching**.
- Implemented the **Hystrix Circuit Breaker** pattern to ensure the service continues operating during downstream service failures, preventing cascading failures and allowing recovery time.
- Created a Docker file to build the images and deployed them in AWS ECS.
- Publishing the device confidence score to **Kafka**, which is then used by the risk assessment service.
- Developed comprehensive integration tests using JUnit and Mockito to ensure seamless interaction between services.
- Utilized Maven for building and managing application dependencies.
- Maintained the token rules file in **S3** buckets and accessed by the **microservice**.

Environment: Java8, Spring Rest Framework, **Spring IOC**, **Spring-AOP**, **Apache Kafka**, Maven, JSON, AWS Cloud, JSON, Apache Tomcat, GitHub.

Company: Infosys, Hyderabad, India
Project: Pathfinder Rest Services and Risk Assessment Service
Role: Systems Engineer

01/2022 – 06/2022
Hyderabad, India

Pathfinder Service API:

The **Pathfinder Service API** is a service designed to optimize customer navigation after successful primary and secondary authentication. By analyzing various factors such as the customer's enrolled platforms, individual preferences, and segmentation data, the service determines the most appropriate navigation flow tailored to each user's unique needs. Customer segmentation is conducted at periodic intervals, allowing the service to adapt dynamically to changes in customer behavior and preferences. This means that as customer interactions evolve, the navigation flow can be adjusted to ensure it remains relevant and engaging.

Risk Assessment Service:

The **Risk Assessment Service** used to calculate the risk score based on the various parameters like customer IP address, browser type, last login timestamp, transaction type, based on the risk score, service will return different statuses like ACCEPT, CHALLENGE and BLOCK.

Responsibilities:

- Developed a **REST** Microservice using the Spring Framework and integrated with various private and public REST services.
- Improved the performance of REST services by implementing parallel REST service calls, significantly reducing response time and enhancing user experience.
- Used Maven for building and managing application dependencies.
- Utilized **GitHub** for efficient version control for smooth collaboration and code management across the development team.
- Implemented testing using JUnit and Mockito to ensure code functionality.
- Developed **JMeter** performance scripts and Grafana dashboards for viewing the performance results.

Environment: Java8, Spring Rest Framework, Spring IOC, Spring-AOP, Maven, JSON, AWS Cloud, OAuth, RabbitMQ, GitHub, Apache Tomcat.