**Sana Ramakrishnan**

**Cleveland, OH**

**+1 754-946-4873**

[**Roxon@theappliedthought.com**](mailto:Roxon@theappliedthought.com)

[**linkedin.com/in/sudarsana-r-8a381b20a**](https://www.linkedin.com/in/sudarsana-r-8a381b20a)

**Professional Summary:**

* Java Consultant with 7+ years of Extensive experience Designing, Developing and Testing Enterprise applications using Java/J2EE technologies for various domains.
* Solid experience in complete Software Development Life Cycle (SDLC) which includes requirement collection, design, coding, development, and testing. Followed various Agile methodology like Test Driven Development (TDD), Behavior Driven Development (BDD).
* Good knowledge in Core Java with a strong understanding of Multithreading, Exception handling, Stream, API, Lambda Expressions, and Functional Interfaces.
* Specifically worked on designing and developing the processes using core java concepts multithreading to make it asynchronous.
* In-depth knowledge in Assessment and maintenance of Microservices architecture applications using Spring Boot 3.0, Spring Cloud 2022.0.2, Spring Security 5.x, Spring MVC 5.x, Spring AOP 5.x, Spring JDBC 5.
* Expertise in developing messaging modules by using Spring JMS 5.0, Kafka and ActiveMQ 5.
* Extensive experience in Object Relational Mapping (ORM) using Hibernate and Spring Data JPA 2.2.x. Working knowledge of Java Web Services with real-time knowledge using SOAP, RESTful, and using POSTMAN for testing the web services.
* Worked on application layer Security using Spring-Security 5.3, SAML 2.0, LDAP 2, OAuth 2.
* Expertise in client scripting language and server-side scripting languages like HTML5, CSS3, JavaScript ES6, jQuery 3.4, AJAX, JSP, JSF, Angular 2/6/8 and Node.js v16.x.
* Designed Single Page Application using Angular 2, Angular 8, and processed JSON data to render on UI.
* Good experience working with MEAN (MongoDB, Express, Angular, Node.js) stack for developing applications.
* Worked with JFreeCharts, D3.js for visualization.
* Extensive experience working with RESTful Web Services as well as Node Rest Framework for backend and using MongoDB.
* Experience in deploying applications on servers like Apache Tomcat 8/7, WebLogic 12.x, and WildFly 19.0.
* Experienced using build tools such as Maven 3.x, Gradle 6.x, Apache Ant 1.10.x.
* Worked on scripting Docker files and deploying it onto Docker containers. Hands-on experience in using Kubernetes to orchestrate the deployment, scaling, and management of Docker Containers.
* Implemented Continuous Integration and Continuous Delivery.
* Managed Jenkins 2.x and Performed maintenance and troubleshooting of build/deployment systems.
* Experience with Amazon Web Services like EC2, S3, Lambda, SQS, EBS, IAM, Cloud Watch, Security groups, API Gateway, Simple Notification Services.
* Developed test cases with JUnit, Mockito for Unit testing, Log4j for logging and JIRA as defect tracking system.
* Knowledge on different databases like MySQL 8.0, Microsoft SQL Server 2019, Oracle, and with NoSQL databases like MongoDB, DynamoDB. Worked with various IDEs such as Eclipse, IntelliJ IDEA 2022.3, Spring Tools 4, Visual Studio Code 1.74.x.
* Team player and Quick learner with strong analytical and strong Interpersonal communication skills ability to work independently as well as in a group.

**Technical Skills:**

**Programming Languages :** Java 1.8, SQL, C, Python 3.x, bloblang.

**Software Methodologies :** Agile, Scrum, TDD, SDLC.

**J2EE Technologies & Frameworks :** JSP (2.x), JSF, Struts2, Servlets 3.1, JDBC.

**Web Technologies :** HTML5, AJAX, Bootstrap, JavaScript, jQuery,

XML, CSS, JSON, Node.js, Angular, React

**Java Frameworks :** Spring 4.3, Spring Security 3.2, Spring MVC, Spring

Data JPA, Spring Batch, Spring Cloud, Spring AOP, Spring Boot.

**Web Services :** SOAP, RestAPI, FastAPI.

**Design Patterns :** Singleton, Factory Design Pattern, Proxy design

Pattern, Data Transfer Object, Data Access Object,

MVC.

**Application / Web Servers :** Web Logic Server 12c, Apache Tomcat (8.5/8.0)

**Relational Databases :** Oracle 9i/10g/12c, SQL Server 2014/2016, MySQL,

PostgreSQL

**NoSQL Databases :** MongoDB, DynamoDB.

**Monitoring Tools :** Grafana, Prometheus.

**Messaging :** JMS MQ, Active MQ, Rabbit MQ

**Build Tools :** Maven 3.3.9, Ant 1.10, Gradle 3.4.1, Jenkins

**Version and Source Control :** GIT, GITHUB, JIRA.

**Testing Tools :** JUnit, Mockito, Log4J, Mock Framework, pytest

**Cloud Technologies :** AWS, EC2, S3, IAM, Elastic Load Balancing, AKS,

Snowflake

**Professional Experience:**

**Applied Thought Inc Sept 2021 - Present**

**UnitedHealth Group Inc / Optum – REMOTE**

**Full Stack Java Developer**

Optum is committed to making health care work better, leading the way to better experiences and better health. This involves collecting clinical data and processing it with the various lab vendors on board. The clinical source data is obtained from the clinical source systems and sends the processed information to the downstream systems.

**Responsibilities:**

* Worked on all tasks using Agile methodology and was involved in design, development and maintaining of the application.
* Developed Microservice based applications which made development and deployment quicker. This involved using Spring Boot 2.x and benthos.
* Connected to RestAPI endpoints for communication with the vendor claims. Developed Rest microservices using Spring Boot 2x and benthos. Used Postman 10.9 for testing the Rest endpoints. Also worked on authenticating and authorizing the API's using OAuth2.0 and JSON Web Tokens.
* Worked on a Kafka based messaging system. Used Kafka topics on GCP clusters for streaming the data. Implemented Benthos 4.11 yaml for communicating with Kafka topics.
* Worked on Google Cloud Platform (GCP) like stack driven monitoring, cloud load balancing, cloud storage and cloud deployment manager.
* Implements multithreading in the Spring Boot application in order to communicate between different Kafka partitions in a Kafka topic.
* Connected with SQL tables and Snowflake tables using Spring Boot and Benthos applications respectively. Querying was done on the tables for getting the success and failed messages.
* Performed caching by Using Redis cache to cache the data and avoid the duplicates. The cached data was also used to implement retry mechanisms for checking the authenticity of the claims.
* Built API's with Node.js for communication with the UI module of the Application. Also worked on Error handling and performed testing on the same.
* Worked with a UI module which was developed using Angular12 and JavaScript. Used Material UI for implementing tables and used Chart JS for developing chats for the JSON received from the API. Worked with Node.js to connect to the PostgreSQL which contained the claims data.
* Worked with updating the application and with new features of Angular12 like the ng-templates, form validation and also Implemented client-side Interface using Angular12. Also, Incorporated authentication for the application in Node.js using JSON web token.
* Used AWS EC2 for deploying the Application and testing the lower environments such as Dev, Test and Stage.
* Object storage service AWS S3 were used for storing and retrieving the claims objects and other large files.
* Used AWS Lambda for creating and performed debugging of serverless applications and used AWS IAM for creating new accounts, roles and groups.
* Performed unit testing using pytest and Junit. Used Junit to test and generate test cases for the Application.
* Worked with Jenkins for building the application's image in docker. The docker images get built when the changes get pushed to the Github repository. Also used Maven, Jenkins, SonarQube and JaCoCo for continuous integration.
* Worked on deploying, scaling and maintaining Microservices for all the benthos yaml and worked with Kafkasource which triggers the pods when the messages appear on the kafka topics the application is connected to the pods.
* Deployed the benthos yaml on to the Azure Kubernetes Service (AKS). Configured the pods to connect to the Snowflake tables and to the redis.
* Monitoring of the message streaming between the Microservices was done using Prometheus and Grafana. The grafana dashboard helps us to track the messages flowing between the services.

**Environment:** Java/JDK 11, Spring Boot 2.x, Restful Web Services, Spring Data JPA, Kafka, Benthos 4.11, Hibernate 5, MySQL 8.x, Maven, HTML5, CSS3, Typescript, JSON, Angular12, Node.js v12, Chart Js, MaterialUI, Snowflake, Redis, Jenkins, Docker, Kubernetes, AWS, Azure, Tomcat 8.x.

**Applied Thought Inc. June 2020 – Aug 2021**

**ERC Enhanced Recovery Center, Jacksonville, Florida**

ERC is a debt collection support provider which is a web-based cloud service. It eliminates work needed for managing the invoices and communicating with the debtor by automating collection operation. This application is built in such a way that it is accessible via web and mobile.

**Responsibilities:**

* Involved in complete project development life cycle SDLC including designing, maintaining, and implementing application in Agile Scrum methodology with Pair Programming technique
* Implemented Microservices Architecture for making applications independent and to deploy it quickly. It involved using Spring Boot 2.x and Spring Cloud.
* Performed tuning of the Microservice based Spring Boot application and used Spring batch for the bulk transaction processing.
* Developed Java Web Services to get customer data from various external application systems.
* Consumed RESTful Web Services using Spring REST for communicating with external applications.
* Used Spring Security to provide JWT (JSON Web Token) authentication, authorization, and provide Single Sign On(SSO) for Application.
* Developed the Session Beans for business logic and Message Driven beans for asynchronous flow of messages
* Used JMS, Kafka to implement Messaging Queue between the services.
* Designed Dynamic and browser compatible pages using JavaScript, jQuery, HTML, CSS.
* Creating a single page application with loading multiple views using route services to make dynamic by using Angular and Node.js.
* Used Node.js to run Grunt tasks and build the project and Implemented modules into Node.js to integrate with designs and requirements.
* Created and managed S3 buckets and utilized S3 bucket and Glacier for storage and backup on AWS.
* Used dynamic data visualization tool D3.js for visualizing the records and payments history and generated records
* Used Jenkins for continuous integration and end-to-end build and deployment automation of Microservices
* Implemented AWS CodePipeline and Created Cloud formation JSON templates in Terraform for infrastructure as code.
* Used AWS CDK for building the application on Cloud and imported AWS CloudFormation templates. It was combined with AWS SAM (Serverless Application Model) to build and test the serverless application
* Worked on NoSQL database DynamoDB for storing and used AWS CloudWatch for monitoring the services and Cloudwatch logs for Monitoring the application.
* Used AWS Lambda for creating and performed debugging of serverless applications and used AWS IAM for creating new accounts, roles and groups and also engaged in enabling lambda function for dynamic creation of roles.
* Worked in containerizing Java applications using Docker and deploying them on OpenShift clusters orchestrated by Kubernetes.
* Hands-on experience in managing Docker images, containers, and Kubernetes resources within OpenShift environments.
* Wrote Test Cases for Service layer and Rest controllers using JUnit 5 and Mockito and also used SonarQube and JaCoCo for measuring the code coverage.
* Developed a logging framework to help debug issues/errors using Log4j.
* Developed using Agile Designed and coded application components in an Agile environment utilizing a test-driven development approach.
* Worked on multithreading and data structure concepts in Core Java for transaction management.
* Developed microservices using Spring Boot and used Spring data JPA to extract patient-related data from MySQL database. Also, worked on migration of monolithic to microservices on the existing modules.
* Messaging services have been implemented using RabbitMQ. This also includes communication between the microservice.
* Implemented logging in the application using Spring Framework AOP Module to know the application status, and Spring Security and Oauth2 for providing security.
* Developed Dynamic single page application using Angular 6 for doctor service to interact with the user's data and worked with component's HTML template and styled with CSS.
* Created API endpoints using Node.js and exposed it for another system to use it. Also worked on structuring the JS code and optimizing it.
* Worked on the CICD process for automated delivery and Created Jenkins pipeline for build and deploy.
* Enabled Amazon IAM service to grant permission and resources to make calls to EC2 and Elastic Load Balancing APIs.
* Used Git for source control, JIRA for tracking and monitoring of work assignments and Log4j for logging of errors

**Environment:**

Java 8, J2EE, Spring Boot 2, Microservices, Hibernate 5, Amazon Web Services, REST, Oracle 12c/11g, RabbitMQ, Maven, HTML5, CSS3, Angular 6/7, JUnit 5.x, JIRA8, Log4j, Jenkins, Docker, GIT. Java/JDK 8, Spring Boot 1.x, Spring Rest, Spring Cloud, Spring Security, Spring Batch JMS, Kafka, Amazon S3, MySQL 8.x, Maven, JavaScript ES6, AJAX, JSON, jQuery, Node.js v12, D3.js, Jenkins, Tomcat 8.x, OpenShift, Kubernetes.

**ZVST Cloud Technologies Jan 2019 - June 2020**

**Client: Ohio Mutual Insurance Group, Bucyrus, OH**

**Java Developer**

Ohio Mutual Insurance is simply one whose interests are aligned with its policyholders. We created a Cloud-based solution based on modern technology. This is a single page application to rate, quote, issue and manage multi line policies, manage billing and receivables, agencies & commission, and Claims. Many features were included to promote user efficiency, reducing average handling time.

**Responsibilities:**

* Performed tasks that are related to design, build, test according to Agile Methodology and followed Scrum
* Used Spring framework's MVC framework to better control the presentation, business, and navigational aspects of the applications.
* Effective use of hibernate ORM tool in the business layer in communicating with the Spring DAO and developed mapping files and POJO java classes using ORM tool
* Worked on Core Java collections API like Lists, Sets, Maps and implemented them in services
* Also worked on multithreading for transaction management and performed caching along the different services.
* Implemented ActiveMQ producers using Spring JMS and also implemented JMS Message Listeners on ActiveMQ
* Designed the front-end screens using Angular2. Developed services, models, and developed HTML templates with various Angular directives.
* Used Node.js Framework to set up middleware to respond to the HTTP request and MongoDB for persisting information from API developed.
* Implemented AWS CodePipeline and Created Cloud formation JSON templates in Terraform for infrastructure as code. Also worked on AWS CodeBuild for build and test.
* Used AWS CDK for building the application on Cloud and imported AWS CloudFormation templates. It was combined with AWS SAM (Serverless Application Model) to build and test the serverless application.
* Developed unit test cases using JUnit and Mockito and configured them through Maven surefire plugins to be part of continuous integration.
* Managed version control using GIT and actively participated in meetings to identify the scope, requirements, and architecture of the project.

**Environment:** Java 8, Tomcat 7, Spring Tool Suite, Spring Boot 2.x, Hibernate 5.x, Oracle 12g, HTML5, CSS3, jQuery Angular2, ActiveMQ, Jenkins, Log4j, Maven, Docker, SAML, Spring Eureka, WildFly 19.0, GIT.

**Amplify Systems, Salem, NH December 2017 – Jan 2019**

**Client: NTT Data**

**Software Developer**

As a software development service provider, Amplify Systems strives to provide their customers with quality application development services and web development services that helps them to remain in step with their competitors. This application organizes client data, tracks due dates, manages staff and workflow, automates billing, and manages documents and Reports.

**Responsibilities:**

* Analyzed business requirements, participated in technical design, development, and testing of various modules assigned following Agile Methodology
* Designed and developed J2EE application using Servlets, JSP, JavaBeans and EJB
* Worked on developing web service calls using RESTful API
* Designed and Implemented User Interface in Model-View-Controller Architecture, which accomplishes tight and neat coordination of Structs, JSP, Servlets
* Developed user interface using HTML, CSS, and AJAX and performed client-side validation using

JavaScript and jQuery

* Interacted with backend MySQL and wrote stored procedures, functions, and cursors for deleting and inserting
* Used GIT version control tool for managing different versions of the application, Maven as a build tool, and Log4j for logging purposes

**Environment:** Java/JDK 1.7, Spring3, Spring DAO, SOAP, Eclipse, Maven, MySql5, HTML4, CSS, JavaScript, AJAX JSON, Bootstrap3, Servlets, XML, JUnit, Tomcat7, Postman, Log4j, GIT, AWS, EC2, Agile.

**Education:**

Bachelor of Engineering (B.E.): Computer science and Engineering - Anna University, Coimbatore, Tamil Nadu - 2017