

Thompson 'Tom' Rajan | Full Stack Java Developer
(845) 475-8714 | tomjben7@gmail.com

SUMMARY

- A result-driven **Full Stack Java Developer** with **5+ years of experience** working in enterprise **end to end**, and **web applications** using various technologies including **Spring, Angular, React, RESTful Web Service**, and **AWS** in all stages of **SDLC** from **Requirement Gathering** through **Maintenance**.
- Expertise in Java concepts such as **OOP, Collections Framework, Generics, Multithreading, Stream, Concurrency APIs**, and **design patterns** such as **Factory, Adapter, Facade, Decorator**, and **IoC**.
- Experienced in building **Enterprise End to End** applications from the **Ground Up** with **Spring Boot, Angular**, and **React** from planning, design analysis through deployment over **AWS ECR** registries.
- Experienced in creating **RESTful** services using **AOP** with **Spring REST** and **Data JPA** with **OAuth2** and **JWT** authentication including **endpoint testing, Spring Actuators**, and documentation with **Swagger** and **Hal Browser** as well as building **microservices architectures** with **Spring Cloud** and **AWS**
- Expertise in performance **profiling**, ability to identify **performance improvements** and **memory optimizations**
- Experienced working with **RDBMS / NoSQL databases** including **Postgres, MySQL, Oracle, MongoDB, DynamoDB**.
- Created front end **optimization** using **client-side templating** using **Angular** and **React** frameworks for dynamic **Single Page Applications** with **Material UI** as well as **Bootstrap** to build customized themes and interactive elements.
- Experienced in creating **IAM** roles for using a specific set of **AWS services** and **monitoring resources** with **CloudWatch** as well as configuring **VPCs** and **Security Groups**.
- Experienced with **AWS SDK** and **CLI** to manage **AWS** services and performance optimizations with **ElastiCache** in addition to building web services with serverless architecture using **AWS Lambda** function to invoke services such as **SES** and **SNS** as well as using **API Gateways** and **S3** to trigger **Lambdas**.
- Experience in developing **middleware applications** using a variety of modern technologies: **Mongo, Redis, SQL Server, Pivotal Cloud Foundry, Rabbit MQ, Angular, Spring**
- Experienced in **CI/CD** to build **docker images** and creating **customized deployment pipelines** with **Jenkins** and **GitLab CI/CD** as well as collaborating and integrating workflows with various teams on **GitLab, Bitbucket**, & managed in **Jira**.
- Experienced working with **Epic integration** to incorporate applications with **patient records** and **Deep Packet Analysis** using **HL7 packets** over **wired and wireless network adapters**.
- Experienced in building **Distributable Desktop Applications** with **Electron** and **ReactJS** along with installers for **Windows, macOS, Debian**, and **RHEL Linux** systems as well as building **Mobile Apps** with **React Native** for **iOS** and **Android**.
- Built **storage servers** and **AdBlock servers** over embedded systems w/ **Raspberry Pi Model B** and **Zero W**.
- Experienced in building tests with **JUnit**, **Mockito**, and logging with **Log4J**.

TECHNICAL SKILLS

Programming Languages / Scripting / Security Tools	Java, Golang, Python, Kotlin, Scala, R, C, C++, Zsh, Bash, Batch, Nmap, Tshark, Wireshark
Backend / Microservices / Web Services	Spring, Spring Boot, Gorilla, Flask, Django, Scikit-Learn, Play Framework, Spring Cloud, REST / SOAP service, Node.js, JNI
Databases / ORMs / Unit Testing	Postgres, MySQL, Oracle, SQLite, MongoDB, DynamoDB, Spring Data JPA, Hibernate, GORM, SQLAlchemy, Mockito, JUnit, PL/SQL, PL/pgSQL, Redis
Frontend Technologies / Desktop / Mobile	Angular, ReactJS, Thymeleaf, Electron, React Native, Material UI, Bootstrap, JSP, JSX, TypeScript, jQuery, JavaScript, HTML5, CSS
Deployment Tools / AWS / Embedded Systems	Jenkins, Gitlab CI/CD, Docker, Docker Compose, Kubernetes, ECR, ECS, EC2, EBS, RDS, Lambda, S3, API Gateway, SNS, SES, VPC, Java AWS SDK, Cloudwatch, NSIS, Electron Forge Server, Raspberry Pi 4 Model B, Zero W
IDEs / Web Servers / Data Process	IntelliJ, Eclipse, Xcode, IBM WebSphere, GlassFish, Tomcat, Apache HTTP Server, Nginx, Bitnami, JMS, Kafka, RabbitMQ, Pivotal Cloud Foundry
Project Management / Version Control / Dependency Management	Jira, Trello, Zoho Projects, ClickUp, Bitbucket, Gitlab, GitHub, Git, Maven, Gradle, Anaconda, pip, NPM

EXPERIENCE

MediTechSafe, Cincinnati OH

May 2018 - Current

Role: Full Stack Java Developer/Angular/Python Django

Description: MediTechSafe, located in Cincinnati Ohio, is a **medical device** and **IoT cybersecurity** company. It has the unique experience of combining **healthcare, clinical engineering, cybersecurity, and software development** while offering the right mix of **Information Technology (IT)** and **Operations Technology (OT)** required for enabling transformation in healthcare via digital innovations. Being a start-up firm during my time of joining, I have been given diverse responsibilities to handle risk assessment of medical devices located in hospitals and clinics since medical patients' personal identity info can be compromised over the internet. As a result, I built the **Enterprise Risk Management Platform App** from the ground up in **Angular** based on a **Full Stack Java Application** backed by a **Spring Boot Backend**. I also built a **desktop application** using **ReactJS** and **Electron** with a **Python Django** based **backend Service**. Using a **Microservices Architecture**, I built **services** to monitor for device vulnerabilities to be used on the **web application** as well as the **desktop application**.

Responsibilities:

- Involved in all phases of **SDLC** from requirements gathering to maintenance and participated in sprint meetings to discuss daily tasks.
- Created an **Enterprise Risk Management Platform App** from the ground up selectively registering medical devices from the hospital inventory provided, on the basis of risk factors that might compromise patient data and to confirm the working status of devices and ensure personal data from patients is secure. The **ERMP App** is built in an **Angular** based **Full Stack App** backed by a **Spring Boot Backend** for users to manage **risk profiles of medical devices in hospitals or clinics**.
- Design and build the **Spring Boot** application including **REST controllers** with **entity DAOs** and **services** using **Data JPA** with added performance optimization along with **Spring Actuators** and **Swagger UI** for API documentation. Use **Spring MVC** to add user registration and login views. The user data are stored in **Postgres RDS** instances
- Secured application performance using **Spring Security** to enable API authentication with **OAuth2**
- A **Microservice Architecture** with **Spring Cloud** on the backend application is built using **Eureka Server** and **Feign** clients to develop a **service to monitor device vulnerabilities** along with test cases using **Mockito**. This service is used across the **web app** as well as the **desktop app**.
- Medical device risk assessment information with their **3D sprite image** sets is stored in **S3 buckets** as **BLOB objects** so that it can be used on the **frontend application** as well as in **mobile applications**.
- Build the frontend with **Angular** with the discussed modules and components with styles in combination with **Angular Material UI** and **Bootstrap**.
- Implement **Angular custom directives** to add custom behaviors to **UI elements** and **custom pipes** to simplify frontend data processing.
- Designed using **Angular routing** with **route guards**, to handle authorized pages and **services** to authenticate **JWT** tokens with **HTTP interceptors** to add **authorization headers** to **API requests**.
- Design and developed a **desktop application** that **scans** and **profiles** targeted **medical devices** for **vulnerabilities** and **reports** to the **ERMP app** using **Electron** and **ReactJS** as the frontend view and **Django** backend service.
- Perform **Deep Packet Analysis** using **HL7v2 standard** packets to profile medical devices based on **DICOM headers** and integrate patient records using **Epic APIs** on the **Django** backend service. Both the **Electron React** frontend and **Django** backend service is built and bundled with **NPM**.
- A **distributed install package** for **Windows** is built using **NSIS** with **MUI2**. **Electron Forge** is used to package for **macOS, Debian, and RPM Linux** to build **DMGs, RPM, and DEB** application installers respectively.
- Use Java **AWS SDK** to create **AWS Lambda** functions on the **Spring Boot** backend application to invoke **SES** and **SNS** services through an **S3** trigger whenever a new device is added and also with **API Gateway** triggers when a new vulnerability is identified for a registered medical device.
- Packaged the **Spring Boot** backend app to a **docker image** and deployed it to the **ECR** container registry through a **Jenkins pipeline**. Version source code with **GitLab** and coordinate with the development team on **Jira**.
- Initiated beta testing campaign to distribute the beta application and gather feedback to make changes on the application through **Jira**

Environment: Java 8, Python, C, Angular, Spring Boot, Electron, ReactJS, Django, Git, GitLab, AWS S3, ECR, EC2, RDS, SES, SNS, EBS, AWS Batch, Jenkins, Docker, Postgres, ReactNative, Nginx, Tomcat, Bootstrap, Material UI, TypeScript, jQuery, JSP, IntelliJ, PyCharm, Xcode, Jira, Mockito, Maven.

Description: Healthreveal is a privately held healthcare based firm that works closely with teaching hospitals, clinical research organizations, and physician-scientists to provide **IT solutions** and **AI services** in **medical imaging** and **radiology** to clinicians and patients. A **Patient Management Portal** is built using **Angular** frontend and **Spring Boot** backend. The **PMP** allows clinics to manage patients and run imaging services on patient X-rays and CT scans to help identify and diagnose diseases. The **AI modules** were written in **python**. In order to bind **python** objects and methods with the **Spring Boot** backend, the **Java Native Interfaces** are created so that it can call **python** objects and methods in **Java**. The **PMP** makes use of **AWS** services to store **PACS images** on **S3** and invoke them using **Lambda functions** as well as trigger **SNS** topics. The PMP also interacts with **patient records** using **Epic APIs** to update and manage **patient data** on the portal. Both the frontend and the backend were installed on separate **docker images** and are deployed to the **ECR** through a **single Gitlab CI/CD pipeline**.

Responsibilities:

- Developed the **Patient Management Portal** working in all phases of the **Software LifeCycle Development Process: gathering information, design, analysis, development, deployment, and maintenance**.
- Collaborated with the team and participated in daily **SCRUM sprint** meetings to meet deadlines using the **AGILE** methodology.
- Designed and built the **Spring Boot** application with the required **entities, services and controllers**, making use of **Data JPA** in addition to **second level caching** with **EHCache** along with **Spring Actuators** and document **REST endpoints** with **Swagger UI**.
- Utilized **Spring Security** to build **UserDetails** based on clinic and patient accounts with corresponding roles from the **RDS** to generate **JWT** tokens using **OAuth2** and develop clinic and patient registration modules with **Spring MVC** and **JSP**. The patient registration module validates patient data using the **Epic API**.
- Created a **Lambda function** to invoke the **S3** bucket and trigger an **SNS** whenever a new image is uploaded or removed and also a **Lambda function** that is invoked by an **API Gateway** to trigger an **SNS** to send out patient validation status using the **AWS SDK**.
- Defined **JNI** interface methods for binding **python objects and methods** into the **Spring Boot** application and created a **microservice architecture** with **Spring Cloud** that takes **PACS images** to provide diagnostic results through a **REST controller** to the **Angular** application.
- Built dynamically designed modules and components required for managing files and images on the **PMP**.
- Optimized performance using **Angular** by distributing the load from the back to the front creating **client-side rendering**.
- Created an **Angular** frontend using **Angular's** dynamic **Single Page Application** with **Angular Material** to create **custom themes** that work on the **web, mobile and tablet devices**.
- Created **LifeCycle Hooks** and **Injected dependencies** into **Angular components** using **_HTTP** services invoked from the **back end**.
- Designed for users to create events that are **bound two ways** so that **Angular components** can communicate with each other from front to back or from back to front.
- Deployed the application by building the **Angular** and **Spring Boot** application into separate **Docker images** and pushed them to **ECR** with a **single GitLab CI/CD pipeline**.
- Managed frontend and backend projects on **GitLab** for version control.
- Used **Jira** to collaborate with the development team.
- Implemented End to End testing code reviews and Generating the test reports at the **unit level** and **integration level**.

Environment: Java 8, Python, Angular, Spring Boot, JNI, Git, GitLab, AWS S3, ECR, EC2, RDS, SNS, API Gateway, Docker, GitLab CI/CD, Postgres, Nginx, Tomcat, BootStrap, Material UI, TypeScript, jQuery, IntelliJ, PyCharm, Jira, Mockito, Maven, NPM.

Description: Liberty General Insurance is a private general insurance company headquartered in Mumbai, India. It has partnered with over 30 district cooperative banks to help issue **vehicle** and **personal accident insurance**, particularly to farmers. The online **insurance management system** helps clients to get a quote, buy insurance, manage their accounts online, **make a claim**, and **get assistance** based on the type of insurance. During my tenure at LGI, I worked on **health insurance** and **individual personal accident insurance**. The **health insurance module** handles the **deductibles** and allows customers to view the **packages** they're signed up for, make a claim for **medical expenses**, and also **make payments** if the insurance type is independent. The **personal accident module** allows customers to view **packages**, make a **claim**, and **payments**. These also send out a personalized **email notification** on the status of the account based on the appropriate circumstances. The web application is developed with **Angular** and **Spring** and is deployed on the **AWS ECR** container registry through a **Jenkins pipeline**.

Responsibilities:

- Created the **Health Insurance Module** which handles the deductibles and allows customers to view packages as well as the **Individual Personal Accident Insurance Module** which allows customers to view packages and make payments
- Participated in requirement gathering, development, deployment, and maintenance phases of **SDLC** and was involved in daily sprint meetings to discuss tasks.
- Design and develop health insurance and personal accident modules using **SpringMVC framework** and **Dependency Injection** to handle business logic and presentation and use **Microservice architecture** to build on top of an existing application.
- Use **Hibernate ORM** to create the mapping for entity beans and the database schema and maintain second-level caching to optimize CRUD operations at scale.
- Used **Spring REST** to build DAOs and services and controllers along with **Spring Actuators** and **Swagger UI** for API documentation and maintain user data in **Oracle RDS** instances.
- Use **Spring Security** to implement authentication and authorization with **OAuth JWT** tokens and build **Single Sign-On** functionality across all modules
- Use **Spring Session** to trace user sessions on the application platform as well as **Spring Batch** to be able to process large amounts of data without causing performance overhead on the application.
- Build **Angular** frontend module views with **Bootstrap** and **jQuery** to the existing application with **Routes** and **Route Guards**.
- Create **Pipes** to process data on the component template and **Angular Directives** to add custom effects on the template as well as **Angular Services** to perform backend API requests
- Intercept frontend HTTP headers with user **web authorization token** to authorize API requests with the Spring backend
- Use **Java AWS SDK** to create **Lambda functions** to invoke an **SES** notification whenever a new claim is detected and also when a payment is made.
- Automate deployment with a **Jenkins pipeline** by packaging the Spring backend application to a **Docker image** and pushing it to the **ECR** registry.
- Version source code with **Bitbucket** and coordinate with the development team on **Jira**

Environment: Java 8, Angular, Spring, Git, Bitbucket, AWS ECR, RDS, EBS, S3, SES, Jenkins, Docker, Oracle, Nginx, Tomcat, Bootstrap, TypeScript, jQuery, JSP, Eclipse, Jira, Mockito, Maven, NPM.

EDUCATION

- **MS Computer Science** - Marist College, Poughkeepsie NY
- **BS Information Technology** - PSG College of Technology Coimbatore IN