Intuit– San Diego, CA
June 2021 to Present
Java Developer
Intuit is a global technology platform, powering prosperity with TurboTax, QuickBooks, Mint and Credit Karm. Intuit helps the customers and communities it serves overcome their most important financial challenges.
Project: Electronic Filing Engine

Responsibilities:
- Designed Core Java classes, interfaces, enums with **extends, implements** relationships
- Wrote **Unit Tests** with **PowerMock** and **jUnit** to increase code coverage from below **30%** to greater than **85%**
- Used **functional interface** and **polymorphism** to handle similar functional class
- Implemented **custom exceptions** and **log4j** to debug and handle error
- Configured **Apache Maven dependencies** to migrate test tool from **TestNg** to **jUnit**
- Wrote **DockerFile** to configure project deployment and dockerization
- Provided different kinds of **scripts** to automatically **deploy, clean, start, stop docker containers**
- Configured **Maven** pom **plugin** for **automatic dockerization** when release project
- Used **Jenkins** and **CDD** to **CI/CD git branches** into remote server
- Wrote **Spring WebServiceConfig** to migrate HttpServlet from **Jboss** to **Spring Boot**
- Developed different **Rest endpoints** to simulate different performance of government agencies
- Wrote **component test** to send **http post, put, get, delete** requests to simulator endpoints
- Handled different payload formats, such as **Json, Xml** inside simulators
- Realized different functionality use **JavaScript** to simulate government websites
- Used **Oracle database** to test filing transmission status
- Wrote **SQL** statement to check the data in database
- Used **Postman** to call the endpoints and update/debug code base on the response
- Participated in implementing and maintaining **Microservice** based on AWS EC2 servers
- Participated in several phases of Software Development Life Cycle including analysis, designing, coding, testing and deployment of the system in **Agile Scrum** SDLC
- Used **JIRA** to track stories and specifying the solution

Wawa, Inc Oct 2019 – Present
Media, PA
Frontend Developer
Digital transformation program Order Management Portal(OMP)
As part of the digital transformation program to enable online catering orders for Wawa customers, OMP is a web page application for internal employees at Wawa to manage those catering-lite orders. The website is built using a central Next Js application, consuming several React Js component libraries. As an in-store associate or general manager, the employee can login to OMP and see online catering orders placed by customer within particular date range. They can also search for an order, click into the order to view order detail, decrease line-item quantity, remove line item, apply discount/promotion/refund to the order, add instructions, print build instructions, or cancel the order. By enabling these features, it will help Wawa employees to better plan their future workload and enable then to mark down requests from customers on the order.

Responsibilities:
- Participated in every phase of Software Development Life Cycle: Requirements Analysis, Design, Implementation, testing and Maintenance in **Agile** methodology environment. Engaged in Team activities such as PI planning, story grooming, iteration planning and retrospective.
- Created the applications from scratch in **Nodejs** environment and used **JavaScript ES6** to handle logics**.** Used **Yarn** to install and run the application.
- Built front-end application for OMP, using **Next JS** to perform **server-side rending** and generate static webpages.
- Built several shared **React Js** component libraries and implement different features as small React components. Import the components to the Next JS app to perform the functionalities.

- Implemented ReactJs component as ES6 classes in **JSX** syntax extension.
- Followed the wireframes from design team, designed and maintained front-end page in **Rebass**, **Styled-components** and **SCSS** for the front-end display.
- Managed states using **REDUX store, reducers, actions,** and **selectors.** Implemented different **Redux-slices** across the component libraries so that the states from separate libraries could be managed in the same store configured in OMP app.
- Used **Redux-saga** to call service functions when the actions are triggered. The service functions will then handle Backend APIs calls using **Axios.**
- Set up **Storybook** to visually test any component during development.
- Performed unit testing on the separate applications and libraries using **Jest** and **Enzyme**.
- Tested deployment on local using **Docker** & **Kubernetes,** with **Istio** and **helm** setups**.**
- Triggered **Codefresh** validation pipeline when creating PR and **Github**, and triggered build pipeline to merging and deploy to dev and upper Integration platforms.
- Helped with new team member onboarding, knowledge transferring and problem solving for any Frontend developer within or outside the team.

Environment
NodeJs, JavaScript, ReactJs, NextJs, Redux, SCSS, Styled-components, Rebass, Yarn, Docker, Kubernetes, Github, Jira

---

Client: Goldman Sachs & Co, Jersey City, NJ **Aug 2015 - Present**
Role: Java J2EE Developer
Project: Exchange Limit Monitoring, Position Limit Monitoring
Exchange Limit Monitoring (ELM) is a redesign of Position Limit Monitoring (PLM) system in order to use newer technologies such as Java 8 and RAMP Framework (GS Framework) and fix some calculation logic. The purpose of PLM is to prevent "sudden or unreasonable fluctuations or unwarranted changes in the price of commodity". In ELM, the system will generate T+1 client position reports on a daily basis and email the reports to front office to review. Operation users can upload reference data through ELM Reference Data Uploader UI. ELM also provides a RealTime tool for front office to check position data at anytime.

Responsibilities:
- Engaged in several phases of **Software Development Life Cycle** including analysis, coding, testing, deployment and go live process of the system.
- Collaborated in developing Exchange Limit Monitoring (ELM) report system by using **Java7/8, Mithra** and **RAMP** framework (GS Framework).
- Implemented several **RESTful** Web Service endpoints in ELM RealTime system and provided data for other modules.
- Created Java based **REST client** to test ELM RealTime endpoints and interacted with front office users to ensure the APIs match their methods.
- Enriched Reference Data Uploader UI to upload data by using **JSP**, **AJAX**, **JavaScript**, **JQuery**, **HTML5, AngularJS** and **Dash-UI** (GS Framework).
- Enhanced Reference data validator for different file types to validate the data that users upload.
- Deployed code to DEV **Linux** machine for user to testing
- Customized PDF and Excel templates for all reports by using **Jasper** and **JXL**.
- Published report specific exceptions to Items for Attention (IFA) framework and collaborated with IFA team to display exceptions on **RAMP** UI.
- Created Data Source Alias (**DSA**) for ELM RAMP reports to fetch data from **COBRA's** canonical tables.
- Built **FitNesse** test cases of different modules for UAT testing.
- Setup **H2** local database to store COBRA metadata and used **Mockito** to mock up position data for local in memory testing.
- Integrated **Etch** with ELM to automatically upload building results to remote repository for future reference.

- Analyzed report calculation results by comparing retrieved data from different databases and user data.
- Loaded **COBRA** raw position data to ELM database and generated position reconciliation file between PLM reports and ELM reports to help Operation users identify system gaps.
- Established a daily futures reconciliation tool for OneChicago Exchange project to compare daily exchange data with data sourced from database.
- Developed **RESTful** web services as an alternative approach to existing DF-based Position Limit Monitoring (PLM) RealTime APIs.
- Worked with manager to release ELM into **production** and performed **technical supports**.

Environment:
JDK 1.7/1.8, Maven, JSP/Servelt, XML, CSS3, JavaScript, jQuery, HTML5 , AJAX, JSON, JUnit, FitNesse Test, Mockito, Linux, Eclipse , Jenkins, Sonar, Apache Tomcat, AquaData Studio, SQL, Oracle 11g, Sybase IQ, H2 Database, Git, Log4j

---

Client: Mercedes Benz USA LLC, Montvale, NJ **June 2014 – July 2015**
Role: Java Mobile/J2EE Developer
Project: Meeting Registration (MBEvents)
MBEvents is a meeting registration system that provides Mercedes Benz Events information and registration forms to employees, dealers and guests. The system includes the IdScanner app, admin portal and general user portal. Admins can create events, edit event information, manage user profile and do other maintenance work. General users are able to view events details and register for living event. When people coming to the event, the reception will use the scanner app to scan people's badge and all attendee information will be retrieved.

Responsibilities:
- Participated in several phases of **Software Development Life Cycle** including analysis, designing, coding, testing and deployment of the system.
- Involved in both **front-end** and **back-end design** and **development** of the web application.
- Developed a **QR code** and **barcode scanner**, the **IdScanner**, based on **ZXing** project by using **HTML5**, **CSS**, **jQuery Mobile** and **JavaScript**.
- Created both **Android** and **iOS native app** of **IdScanner** by using **Apache Cordova**, **XCode**, and **Eclipse** and deployed them on different devices.
- Designed the project based on **Spring MVC** framework and connected to **Oracle 11g** database by using **JDBC** in developing environment.
- Used **Spring security** and **JSTL** to address **authentication** and **authorization** to handle secure login/logout, check user roles and display more functional items for admin users, like edit events and profile maintenance etc.
- Designed front end pages by using **HTML5**, **CSS3**, **BootStrap** and **JSP**, and used **JavaScript**, **jQuery** and **Angular.js** to implement page functions and routing pages.
- Applied **responsive web design** to the project by using **media queries**, and made all pages compatible in major mobile explorers.
- Used **Ajax** call to handle **asynchronous request** and retrieved employee data and dealer information in **JSON** format from **MBUSA EAI REST web service**.
- Customized **jQuery validator** plug-in to do instant form validation like checking empty field, input data length and giving customized error messages.
- Used **Spring form tag** to binding values from/to backend controller and handle form submission tasks.
- Integrated **JavaMail** with Spring framework to send confirmation emails and reminding emails when user completes registration, before registration due date or in some other situations.
- Configured **JNDI connection pool** to connect database in **QA** and **production** environment.

- Generated **war file** by using **Maven build** and deliver the war file to deployment team to deploy it on **IBM WebSphere** server.
- Wrote **stored procedures** and **SQL queries** to **insert**, **update**, **delete**, **retrieve** data and manipulate tables and used **triggers** to keep audit tables up to date.
- Loaded project dependencies by using **Maven** and used **Git** to connect to remote **Stash** repository.
- Monitored debug logs by configuring **Log4j** in the project.
- Configured **security headers** in **Spring Security** and **JSTL** for **JSP** output fields to prevent **cross-site scripting attacks**.
- Create **JUnit** test cases throughout the project to test Java functions

Environment:
JDK 1.7, Spring 3, Spring MVC, Spring Security, JSP/Servelt, XML, CSS3, JavaScript, jQuery, HTML5, BootStrap, Angular.js, JSTL, AJAX, JSON, JUnit, XCode, Eclipse, JNDI, Apache Tomcat, IBM WebSphere, SQL, PL/SQL, Oracle 11g, Git, Maven, Log4j

---

Walmart Inc – Bentonville, AR
Java Full Stack developer
November 2019 – Present
Cost-Negotiation Web Application provides a platform displaying sales data in comparison of each company in Walmart stores and online in chart and graph, and also indicating the advantage and disadvantage between Walmart merchandisers and suppliers. It allows the merchandisers to adjust negotiation strategies by understanding visually and directly how important a suppliers' company to Walmart. We are using Microservices architecture for back-end with Spring Boot, and Micro front-end UI with React.

Responsibilities:
- Designed and implemented restful API based on **Optional** and **Stream API** with Spring Boot and Spring MVC
- Optimized code format and structure with **Spring Expression Language** (SpEL)
- Implemented meta cache for commonly used data
- Extensively used **Core Java** like Collections, Annotations, Serialization and Exception handling to handle back-end logic functions
- Implemented custom validators for Java Bean validation
- Designed and implemented the persistence layer with **JdbcTemplate** and implemented dynamic query for SQL parameters based on SpEL
- Debugged and maintained queries for Clickhouse with DBeaver
- Extracted a big component from single UI project to sub app with **single-spa**
- Used React router to render different components based on the URL and turn application into single page application
- Built a new UI component in JSX based on UX design in Zeplin with HTML, CSS
- Implemented corresponding action, saga, reducer to communicate with the backend API
- Implemented responsive web design with Material UI
- Participated in the requirements analysis, model and function design, development, and unit and regression testing of applications using **Agile Scrum Methodology**
- Applied **Test Driven Development** (TDD) to design and implement and test functions
- Used **Enzyme** + **Jest** to perform unit tests for React, and reached over 90% code coverage
- Experienced checking logs and located prod issue for remote server with **Kubernetes**
- Improved the code coverage from 30% to 98% for several backend services using **Mockito + jUnit**
- Designed and implemented integration tests for backend service using **Cucumber**
- Designed and implemented regression tests for entire system with **Cucumber** + **Selenium** to notify the developer if the new feature works well or not

- Ran **load test** for important API periodically to adjust remote server configuration to make sure the application working fine in the appropriate usage of CPU and memory
- Applied **Kanban** approach in **JIRA** to track the development process
- Participated in PR review for code reliability and readability
- Used Looper and WCNP as CI/CD to deploy our application to remote server
- Utilized **GIT** to do version control

Environment: Java 8, Spring Boot 2.1.4, Spring 5.1.6, SQL Server 2016 v13, Clickhouse v21, React 16.8, JavaScript ES6, JSX, Material UI, HTML5, CSS3, DBeaver, Kubernetes, TDD, Agile, Scrum, Kanban, jUnit, Mockito, Jest, Enzyme, Cucumber, Selenium, Continuous Integration, Continuous Deployment, Looper, WCNP

Cost & Negotiation Tools
Client: Walmart Inc – Bentonville, AR
Role: Full stack developer
Duration: August 2019 – Present
Description: Cost & Negotiation Tool is an application that designed for Walmart merchandisers to help merchandisers capture available profit between competitors and suppliers, vendors choose where they will invest their dollars for growth and then make optimal decision to minimize internal operating costs. The final goal is to consistently reset the merchandising cost structure to allow Walmart to deliver on its mission of enabling customers to save money and liver better through profitable growth with great everyday low prices, which is the core idea of Walmart. My task was to implement two main modules for merchandisers, including backend API design and implementation, database management and frontend UI implementation.

Responsibilities:

- Follow Agile development method strictly including daily scrum meeting and two-week sprint, using version control tool Git to ensure all the processes and JIRA Kanban board for project tracking.
- Designed required table columns and relationships between different tables in SQL Server database.
- Set proper table constraints like primary key, foreign key, sequence and trigger on each table and created Entity Relationship Diagram for review.
- Used Spring boot framework for backend development and Maven for project dependencies management.
- Used React framework for frontend development and NPM to manage basic React packages.
- Designed basic structure of RESTful API for data collection and calculation.
- Constructed required Spring Beans and configured their mapping relationships with the tables in relational database.
- Created Data Access Object (DAO) and used Java Persistence API to access data in relational database in a consistent way.
- Developed Services for Create, Read, Update and Delete methods using Spring IOC annotations like @Service, @Autowired.
- Added @Transactional to ensure data integrity and consistency during CRUD operations.
- Designed Spring-Controllers and configured accurate GET, POST, PUT, DELETE mapping to handle http requests from front-end.
- Used JUnit/Mockito in Java and Enzyme in React for Unit Testing.
- Used Selenium to perform automation testing for API and UI
- Provide support for testing at QA, integration and Production.
- Deployed database, front-end and back-end from local environment to cloud environment using WCNP services.

Techniques: Core Java, Spring Boot, Spring Security, Maven, Hibernate, Tomcat server, Angular 4 framework, React, JavaScript, Typescript, Bootstrap, Git, JIRA, Eclipse IDE, Intellij, WebStorm, Postman, JSON, HTML5, CSS3.

Walmart Inc, Bentonville, AR
Full Stack Web Developer
Oct 2019 - Present
Private Brands PLUM Product Cycle Management System

Description:
In Walmart, I build a product cycle management system using Java Spring Boot, React.js and Azure serverless database. The

goal of this product is for both internal users and client suppliers to search, update and submit product, quotes, samples etc.

Responsibilities:
Backend:

- Utilized **Spring Boot** building **Spring RESTful API** to process requests from the front-end that posted by React.js axios Library
- Used **Apache Spark** package to sync data among databases, data processing pipeline includes loading, cleaning and feature engineering
- Utilized **RestTemplate Library** to consume other web services and send HTTP requests, including getting pictures and refreshing cookies
- Had a strong understanding of **JpaRepository**, connected several databases within one project, hence one project can retrieve data from multiple databases
- Utilized **Lowagie** Library to print nice pdf files for users. In pdf, I include tables, images and watermarks
- Using **Metrics** library and to optimize performance and find bugs. Successfully improve the response time from 5 seconds to 2 seconds
- Used **Jackson ObjectMapper** Library to convert java objects and json strings
- Had a solid understanding of **CICD**, used **Ansible** for automation configuration
- Wrote **Mokitco** unit test cases to cover all functionalities and used **Cucumber Library** to write integration tests
- Took **on-call** service for one week every three months, updating and cleaning data, fixing tech debts and production bugs

Frontend:

- Developed front end Single Page Application using **React.js and Material UI**
- Utilized the **Redux** and **redux-promise**, used reducers and actions to create, read, update and delete data in the **Redux store**, passed data through **props** to provide single source of truth and to protect the state read-only
- Expertise in creating custom reusable **React Stateless Components**.
- Had a strong understanding of **State Hook** and **Effect Hook** to work with stateless components
- Used **Axios** library to post AJAX requests to the back-end RESTful API
- Utilized **React Lifecycle** to dynamically rendered content
- Used **Material components** such as Grid, Dropdown, Stepper, List, Box, Paper, Button etc.
- Wrote **Enzyme unit test** cases to cover all functionalities, used **Shallow** to mock props
- Wrote **Selenium** integration tests to cover end to end use cases

Environment:
Sprint Boot, Spring MVC, Spring Data JPA, Spring Security, Docker, Oracle, Microsoft Azure database, ReactJS, HTML5, CSS3, Bootstrap4, Material UI, JavaScript, JSX, jQuery, Ubuntu 16.04, WCNP, Sonar monitoring, Medusa monitoring, Splunk logging, Dynatrace metrics

---

Sam's Club
Bentonville, AR
Software Developer
Rewards2Go, Sam's Cash 09/2020 – present

Sam's Cash Reward is a program that allows member to accrue cash rewards on qualifying purchases. Rewards2Go team is responsible for designing and developing services for partner offers, which includes RESTful APIs that will be consumed by external services, and listeners that will consume transaction messages and member events.

Responsibilities:

- Follow **Agile** development principles, use **Jira** for **Scrum** management
- Design and implement a common **Azure CosmosDB** module, write **SQL** queries for CRUD operations
- Develop listener apps that consume messages from **Azure Event Hub** and **Azure Service Bus** Topic
- Design and develop web services with **RESTful APIs** using **Spring WebFlux** framework
- Implement Router Functions for functional programming, develop handlers and services using **Java 8**
- Integrate **Azure Application Insight** for request, dependency, error and trace logging
- Create custom recover annotation using **AOP** and **AspectJ** for load-time weaving, publish message to **Azure Service Bus** Queue
- Add retry and circuit breaker support using **Resiliency4J**

- Manage API inbound and outbound rules on **Azure API Management**
- Implement unit tests using **Junit** framework and use **SonarQube** for code quality monitor and management
- Use **Maven** for **Spring** project management and comprehension
- Adopt **DevOps**, work with **Looper** and **Concord** for **CI/CD** pipeline.
- Integrate helm chart for deploying **Docker** image on **Azure Kubernetes Service**

Environment:
Java 8, Spring, Spring Boot, Spring WebFlux, SQL, Azure CosmosDB, Azure Application Insight, Azure API Management, Azure Kubernetes Service, Azure Event Hub, Azure Service Bus, AOP, AspectJ, REST, JSON, Jackson, Git, JUnit, SonarQube, Maven, Agile, Scrum, Jira, Docker, DevOps, CI/CD, Looper, Concord

Core API, Membership On Cloud 07/2019 – 09/2020
As part of Membership Redesign / Mainframe elimination project, we are designing new service development framework, referencing new libraries, functional programming style. Our team was responsible for designing and developing RESTful APIs for CRUD operations on profiles and memberships.

Responsibilities:
- Follow **Agile** development principles, use **Jira** for **Scrum** management
- Design and develop **RESTful APIs** with **Spring WebFlux** framework for reactive programming
- Implement Router Functions for functional programming, develop handlers and services using **Java 8**
- Write **SQL** queries for CRUD operations on **Azure CosmosDB**
- Integrate **Azure Application Insight** for request, dependency, error and trace logging
- Create custom annotation using **AOP** and **AspectJ** for load-time weaving
- Manage API inbound and outbound rules on **Azure API Management**
- Implement unit tests using **Junit** framework and functional tests using **JBehave** framework
- Use WebClient to make external API call, and use **Jackson** to parse **Json** data to create request payload
- Use **SonarQube** for code quality monitor and management
- Use **Maven** for **Spring** project management and comprehension
- Adopt **DevOps**, work with **Looper** and **Concord** for **CI/CD** pipeline.
- Deploy **Docker** image on **Azure App Service**

Environment:
Java 8, Spring, Spring Boot, Spring WebFlux, SQL, Azure CosmosDB, Azure Application Insight, Azure API Management, Azure App Service, AOP, AspectJ, REST, JSON, Jackson, Git, JUnit, JBehave, SonarQube, Maven, Agile, Scrum, Jira, Docker, DevOps, CI/CD, Looper, Concord

---

Client: Sam's Club, Bentonville, AR Sep. 2020 - Now
Role: Java Developer
Project Description: Sam's Club is an American chain of membership-only retail warehouse clubs owned and operated by Walmart Inc. The vision for Sam's Rewards project is to enable a world-class loyalty program that clearly communicates the value of membership to Sam's Club members and allows them flexibility in accessing and managing rewards. Our team is responsible for Accounting and Supervisor systems. The former is used to aggregate payout amount at the end of the day. The latter monitors the other systems and sends alerts when there is an anomaly.

Responsibilities:
- Designed the architecture for supervisor system to monitor events from other teams
- Constructed an efficient way to read event flows from YAML and improved configuration readability

- Used Azure Event Hubs and Service Bus for producing and consuming messages; explored performance differences among Azure SDK, Spring JMS Listener and Spring Cloud Stream
- Set message properties and applied Subscription Filter to filter redundant messages
- Performed CRUD operations with Cosmos DB, SQL Database and Table Storage; designed partition key of Table Storage for faster query
- Created a starter for Table Storage; enabled operations on tables in different regions
- Applied reactive programming and Java Stream to process query results
- Integrated Resilience for retry logic and Application Insights for alert logging
- Implemented a Restful API for accounting metadata with Spring IoC, Spring Data JPA, and Spring MVC
- Improved performance by use Guava Cache for accounting metadata
- Performed unit test with JUnit and Mockito
- Coordinated with other teams and performed functional test with Cucumber
- Followed Agile methodologies including scrum meetings and bi-weekly sprint
- Used Git for version control, JIRA for story tracking, and Looper for CICD

Environment:
Java 8, Spring Boot, Spring IoC, Spring MVC, Spring Data JPA, Spring Data R2DBC, Spring JMS, JUnit, Cucumber, Git, Jenkins, Postman, Insomnia, IntelliJ IDEA, Azure (Service Bus, Event Hubs, SQL Database, CosmosDB, Table Storage, Application Insights, Key Vault)

Company: Google ─ New York, NY **April 2017 - Present**
Role: Software Developer
Project: CEDI API/Cloud, Sphinx
Worked on multiple projects and applications in CEDI team including AutoAPI, Carafe, Titanoboa, AppMaker, Sphinx, API Crawler, HR API, Facilities API and Teams. Involved in numerous tasks such as pipeline, data dumper, alert/prober, cronjob, webApp, productionization, POC (Proof Of Concept), service migration and Tin test.

Responsibilities:

AutoAPI – a set of tools used to generate Apps Framework APIs running on Boq (Java, Microservices).
- Implemented features to support specific API field for list request with constraint, fix reported issues and refactor code to avoid linter/conformance warnings.
- Created multiple Tin and golden tests for crudl features to ensure the generated code works as expected.

AppMaker – a cloud-based platform for building data-driven webapps with little programming (Java).
- Refactored the code structure of appMaker codebase and modularized the project to provide more flexibility for code migration and maintenance.
- Assisted 30+ clients to build and deploy their apps on GAE appMaker runtime.

API Crawler – a tool to extract API metadata for generating API documentation exposed on PLX (Java, Python).
- Worked on the API crawler to fetch, transform and dump all CorpEng APIs' metadata into datahub. Expose API metadata to PLX dashboard as API documentation accessible to internal users. Created cronjobs running in multiple environments and worked on Productionization for continuous build, release, and deployment.
- Created alerts to monitor the job status in multiple environments. Worked on production support to add features, fix issues and improve the project health.

Sphinx – a tool for managing access to systems and resources within Google (Java, Python, Bash).
- Created status-fetcher to dump access anomalies of user membership and registered it to be a command-line tool for support-team to analyze the root cause efficiently, instead of doing it manually.
- Worked on membership reconciler and create cronjobs to correct membership status of users and report inconsistencies.

- Created service probers monitoring the health of RPC services in project and sending email to the team if service is down.
- Created a pipeline job to locate and backup dangling references generated in access management workflows to analyze how invalid data is generated in the system.
    - Optimized the pipeline to automatically process ~1 billion records efficiently in less than 3 hours for daily run.
    - Created customized metrics to collect data of each table for statistical analysis and sprint report. Created alerts to monitor metrics and notify team of any anomalies.

Carafe/Titanoboa – appHosting projects to provide a container solution for migration of internal Google App Engine (GAE) apps (Java, Python).

- Fork legacy libraries and worked on python modernization to support 5k+ internal existing and new GAE apps for migration from GAE python 2.7 to 3.x runtime.
- Worked on configurations and customized library to provide a template for client apps to build, push and deploy the app image on GAE for Java and Python. Assisted 20+ clients for their app migration in java.

Facilities API CloudPoC – POC to verify availability of Facilities API on cloud (Java, C++, Angularjs, SQL).

- Created client services of Facilities API with Stubby/gRPC/eStubs to verify and demonstrate Facilities API, created a client webApp running on GAE to demonstrate the cloud-based Facilities API service via UI.
- Worked on Protoserver service to enable async/streaming with gRPC/eStubs for Facilities batch-API and expose data in Dremel accessed through GoogleSQL.

Client: **Apple – Sunnyvale, CA**
Position: **Front-end Lead Developer**
Duration: **Apr 2020 – Current**
Projects: This project is to create a platform for Applied Machine Learning team and other business groups such as Apple Online Store, Marcom and Support to provide an easier and faster flow to configure 'search result' being returned based on input queries.

Responsibilities:

- Worked with all stakeholders including UI/UX team, AML team and future users to identify the functionalities, user experience and the design of the application
- Used **LEAN** principles and **Design Thinking** to identify strength/weakness of a flow for next iteration improvements
- Built Prove of Concept and revision using various technologies:
    - **Vue.js** combined with Vuetify.js to quickly implement concepts of workflows for review.
    - **Mobx.js** for application state management.
    - **Nest.js** as mocking Back-end server to mimic API calls.
    - Revision with **React.js** and **Sass** with minimal packages to align with Apple design with best efficiency.
    - Utilized React **Hooks** such as useState, useRef and userContext with functional components.
    - Used **React-Redux** for Store management as a single source of truth.
    - Implemented routing using **React Router** and test React components using **Jest**.
    - Worked with authentication team to implement authentication and authorization of the application.
    - Integrate Colombo (internal) as the analytic tool to track user activities.
- Worked closely with DevOp team for Continuous Integration and Delivery (**CI/CD**)
- Developed using **Git** for versioning and **Quip** for documentation, info sharing and as a Kanban board.

Technologies:
Vue.js, Vuetify.js, Mobx.js, React.js, Redux, Nest.js, Jest, JavaScript, CSS3, HTML5, Node.js, JSON, GIT, Quip, Radar

---

Client: Morgan Stanley – New York, NY

Position: Full Stack Developer

(10/2018 – Present)

Project: Financial Advisor Compensation is a hybrid platform with legacy mainframe system and modern distributed system, which processes and calculates each FA's compensation on daily basis based on rule engine.

- Design, develop, and deploy the REST app for FA Compensation from front end to back end; provide production support for releases and on call support.
- Design/Implement system features with business users from FCG and other teams, provide draft sample system for demo in early stages when concrete requirement is often not present.
- Work with offshore team in India to implement the system in each sprint/release cycle and deploy to servers on webfarm, support testing with QA and business team before /after release.
- Constructed the system with Java 8, Spring, Apache Camel and Angular 4. It can take in user inputs in different volume in UI side, enrich data by calling third party APIs, validate forms and saves to database through back end REST server, and run daily batches to process all the transactions involved every night.
- Developed back-end REST server using Core Java features, e.g, Collections Framework, I/O, Generics, and Exception Handling; used Multi-Threaded design to run concurrent tasks to improve the performance of the system.
- Implement the authentication/authorization filters with Morgan Stanley entitlement system to validate user identity, and grant layered roles based on entitlement available.
- Reduced the UI app loading time for bulk upload by 70% by redesigning the DOM structure and API; solved exponentially growing RAM usage issue by redesigning the result set returned from SP.
- Implemented the reprocess system that scans the database, filter out ineligible records, and generate multiple files that contains target data on shared NAS drive, to process again with updated business rules for downstream systems. It can handle more than 2 million records within 15 mins with 4GB max JVM heap size setting.
- Schedule, update and maintain AutoSys jobs to run batch jobs every day; communicate with mainframe/informatica team to cascade the daily batch seamlessly. Implement different file writers that writes on NAS drive to communicate with mainframe and incorporated them with a series of cascading Autosys jobs to ensure their executing sequence.
- Monitoring monthly release build process and fix prod issues on demand.

04/2019–Present

Java Back-end Software Engineer

Apple Inc. Sunnyvale, CA

Project: Entitlement/Cloud Config/ Audit Services

Cloud Config Service is a central configuration manager that supports dynamic configurations. Backed by a MongoDB database, supports property versioning, uploads and downloads as files. Audit Service is a solution that designed to ingest and store an application's auditing data. It is designed as a multi-tenant solution. Entitlement Service is a generic, light-weight, scalable and performant solution that helps application architects disengage the authorization model from the business data model, keeping the data model flexible, assisting the move to SOA from a monolithic architecture, and help horizontal scaling.

Responsibilities:

- Developed Non-Blocking APIs to obtain the frontend response and error reports using Reactive Programming which boosted the Scalability and Efficiency by 30%
- Designed and developed MVC style **Restful** API to handle requests and make the response through **Spring Web-Flux** and configured JAX-RS style Non-blocking APIs using **ServiceTalk** to increase Scalability.
- Participated in implementing and maintaining three internal microservice based java servers by utilizing **Spring Cloud Gateway**, **Eureka Server** to improve the Security, Scalability and Flexibility.

- Used **Spring Security, Spring JWT** to configure user-service to build the authentication work-flow for api-gateway process
- Used the **FeignClient** and **RibbonClient**(Rest Template) to communicate with microservices and implement Load-Balancing for the entire backend services
- Built embedded **MongoDB**, and **Cassandra** for frameworks to bulk the data analysis and searching quickly
- Involved in installing, configuring and monitoring the **Datastax**, **DevCenter** and **OpsCenter**
- Built the bridge from the middleware service to business service and wrote **Rest API docs** to improve the project's readability.
- Developed high coverage of unit testing based on **Test Driven Development** and conducted **Integration Test**
- Followed **Agile development methodology** and attended **scrum meeting** on a daily/weekly basis to ensure predicable delivery and focus on business value
- Used **Git** as version control tool and for code review and **Rdar** to manage the task and progress

Environment: Java 8, Java 11, Datastax 5.0.1, Datastax 5.1.0(upgraded), Intellij 2019.1.1, MongoDB v4.0.3, Spring-Web-Flux 5.2.6, Spring-Cloud-Sleuth-Core 2.2.3, Spring-Security-Web 5.3.2, Spring-Cloud-Starter-Gateway 2.2.3

Client: Morgan Stanley – New York, NY **11. 2015 - Present**
Role: Senior Java/J2EE Developer
Project 1: Entitlement Central (eCentral)
Entitlement Central (eCentral) platform at Morgan Stanley is to ensure that every application within the Firm properly utilizes standardized/centralized mechanisms for system and user authorization and audit controls, and provides the firm with tools and process to better manage the risk associated with inappropriate system access including penetration of firm systems, reliance on manual processes, fraudulent transactions etc. The system functionalities including Application On-boarding, Reference data management, Access Management Administration, Entitlements Lifecycle Management, Run Time Access Control, Entitlement Data Aggregation and Reporting.

Responsibilities:
- Involved in all the phases of **SDLC** including Requirements Collection, Design & Analysis of the System Specifications, Development and Customization of the Application.
- Involved in the integration of business process using **Core Java Collections, Generics, Exception handling**.
- Analyzed the functional specifications of firm-wide applications and their entitlement models for access management integration with eCentral entitlement system for a centralized access management control.
- Worked on High-level specifications documents to drive for the future development phases, utilize **Microsoft Visio Studio** for creating complex system diagrams and integration workflow charts.
- Developing the backend business process based on **Java** to support application on-boarding and the Entitlement Lifecycle Management using **IntelliJ** IDEA IDE.
- Developed and built a data load process based on **Java** for loading entitlement reference data through various firm entitlement sources such as A2, Datagate, JDBC, E3, **LDAP**.
- Designed and developed Database staging Tables, **Stored Procedures**, **Views**, and complex **SQL** queries using the IDERA **DBArtisan** tool on IBM **DB2** database to collaborate with the Java program on various functionalities, such as staging data propagation, sync rule mapping return, source and target data comparison and data reconciliation.
- Analyze performance issues for the reference data source which contains a large number of records by using **Log4J** for logging essential debugging and error information. Collaborated with DBA to identify the SQL queries/Stored Procedures with high cost, generate **Query Execution Plan** using IDERA **DBArtisan** and optimize the query/Stored Procedure/Table Index for the best performance. Enhanced the reconciliation process with **Java Multithreading**.
- Creation of **REST** Web service operations based on design contract to facilitate communication between client and servers.
- Designed and implemented end-to-end approval workflow functionalities under Access Management Java module to facilitate business users to request and approve entitlements.
- Developed entitlement reporting functionality to assist weekly business entitlement review to highlight potential

entitlement risks across the firm applications that are using eCentral.

- Followed **Test Driven Development (TDD)** and implemented Java unit test cases for the on-going developing functionalities using **Junit** and **Mockito** framework
- Use **Git** as version control and perform code review with team members to make sure the code is implemented properly based on the requirement and follows the Java best practice.
- Followed the **Agile** methodology and participate in daily **Scrum** meeting, sprint planning meeting and periodical meetings with correlated teams.
- Followed the firm **SDLC** 2.0 and **Technology Change Management** (TCM) policies to prepare release documents and promote release changes onto the Production environment.

Environment:

JDK 1.8, IntelliJ, Spring, JDBC, HTML5, AngularJS, Ajax, JavaScript, CSS3, DB2, Git, Tomcat, Putty, SQL, Gradle, Junit, Mockito, SoapUI

Project 2: Access Enforcement Tools (AET)

AET project is used by application development and infrastructure teams to define, administer, manage, audit and report on access management policies of applications and services that act on firm data and hence need to be secured. It facilitates access-control by processing relevant firm data sources for entitlement purpose into firm authorized entitlement stores. Its products include Datagate, Xcalibur, RDE, E3.

Responsibilities:

- Involved in **Agile** development team for gathering the requirements, analyzing the user specifications, developing and testing.
- Documents requirements from Project Manager and design high level UML diagrams to depict the flow of the system.
- Used **Spring MVC** integrated **Hibernate** in the back-end side taking advantage of **IoC**, **AOP** and **Hibernate Dao Support**.
- Applied Core Java concepts such as **Collections**, **Exception Handling**, **Generics**, etc that designed for the business logic in the back-end side.
- Performance tuning on the varies service operations and improved system reliability and stability.
- Applied **SOAP Web Service** and **JavaMail API** to generate and deliver alert email and processing report.
- Tested Java functions using **Unit Testing** & **Regression Testing.**
- Used **Git** for version control and **JIRA** for bug tracking and project management.
- Followed the **Agile** methodology and involved in daily Sprint meeting.
- Investigate into tickets that users raised to us through the **ServiceNow** system and communicate back to users with specific instructions or detailed issue root cause analysis.
- On-call for debugging production failures, collaborate with off-shore team and provide full support for the development perspective.
- Review changes tickets, builds and follow-up with web-ops, DBAs for deployments.

Environment:

JDK 1.7, Unix/Linux, Spring 3.1, Hibernate, HTML, JavaScript/jQuery, AJAX, DOM, JAXB, XML, JSON, CSS, Mainframe DB2, Sybase, Eclipse IDE, JavaMail, JUnit, Git, JIRA.

Walmart Headquarters, **Bentonville, Arkansas** **Nov.2019 – Present**

Software Engineer, Cost and Negotiation Team

PAL Negotiation Platform @ Walmart

- Design and develop core back-end features for project PAL Negotiation Platform
- Responsible for unit test, load test and integration test implementation
- Coordinate with PM, define/clarify business requirements and convert to concrete technical tasks
- Responsible for full stack development, including both front-end UI design and back-end core business logic
- Responsible for database migration from SQL Server to ClickHouse
- Maintained and optimized database query to enhance database performance

- Deployed the back-end services using CICD tool WCNP
- Created API gateway using Netflix Zuul to load balance requests to different services

---

Samsung SDS America – Ridgefield Park, NJ
Java Developer
Jan. 2020 – Mar. 2020

Description:
I was a part of the Samsung.com support team to improve back-end logic and refactored the core CRUD logic to reduce execution time.

Responsibilities:

- Extensively involved in an **Agile development team** for gathering the business team requirements. There were daily **Scrums** meetings for updating the status of the project and resolving any issues.
- Integrated Spring injections with dependencies to achieve **Inversion of Control(IoC)**
- Refactored the core CRUD logic to reduce execution time in **Oracle** database
- Created **DTOs** (Data Transfer Objects) with **Project Lombok**
- Implemented service layer using the **Java interface** and implementation classes.
- Involved in building backend logic with Core Java knowledge, such as **Collection**, **Generics**, **Iterator**, **Stream API**, **Reflection** and **Serializable** to increase performance of service application
- Used **Apache POI** to generate Microsoft Excel
- Wrote Unit test cases with **Junit** for backend functionalities
- Used **Swagger** API description format for API documentation
- Worked on **Jenkins** for Continuous Integration
- Used **GitLab** as a version control tool and used **Postman** to test API development

Environment:
Java 8, Spring boot, Spring MVC, Project Lombok, RESTful Web Service, Oracle DB, Oracle DB developer, Mybatis, Maven, Junit, Jenkins, Swagger, IntelliJ

Samsung SDS America – Ridgefield Park, NJ
Java Full Stack Developer
Oct. 2019 – Dec. 2019

Description:
Samsung SDS America is a global software solutions and IT services company. I worked on the COE (Contact Center) project with the goal of combining Samsung Birty ICC (Samsung AI engine), Samsung STT (speech to text service), Samsung ASP (agent service portal), the Google search API, and the GENESYS softphone together into one web application to increase the working efficiency of the contact center's agents workflow. This system is built with Angular 7, Java 8, Spring Boot, Spring REST API, Mybatis, and MySQL. My responsibilities were to create DTOs by using Project Lombok, entities, services, and controllers to collect ASP and Google search data to be returned to the client. I also configured a spring boot which integrates the MyBatis multiple data sources together, created a batch service, built Angular components, and created services to assist the front-end developer team. In addition, I worked with the server infrastructure team to configure load balancer (L4 switch) to handle the project built using Jenkins.

Responsibilities:

- Involved in analysis, design, development, testing, deployment and maintenance stages, using **Agile** scrum methodology
- Built Single Page Application with **Angular 7**, **Angular Bootstrap**
- Implemented Spring beans and bound them with Spring Container with **Dependency Injection** (DI), and **inversion of Control** (IoC)
- Created **DTOs** (Data Transfer Objects) with **Project Lombok** by using annotations like @Data
- Built RESTful Web Service, and controller with **Project Lombok** annotations like @Slf4j, @RequiredArgsConstructor to handle different Http Requests
- Used **Spring WebFlux** to handle asynchronous/non-blocking HTTP request to communicate with Samsung ASP server and to collect data from the Google search API
- Utilized **Spring Security**, **JWT** to implement user authentication and authorization
- Involved in building backend logic with Core Java knowledge, such as **Collection**, **Generics**, **Iterator**, **Stream API** and **Serializable** to increase performance of service application
- Experienced in frontend technologies like **JavaScript**, **HTML5**, **CSS3**, and **TypeScript**
- Created highly customizable **module**, **component**, **pipe**, **service**, **authentication guard**, **router**, and **router-outlet**
- Implemented **MyBatis** multiple data sources and multiple mappers in Spring Boot
- Inserted data into **MySQL** tables using **Mybatis** and store data from the ASP site and select data to interact with Samsung Birty ICC
- Used **Oracle VirtualBox** to create Linux environment to build front-end angular project and deploy the **Node module** to the development server
- Worked with diverse teams including: the UI/UX team, the front-end team, and the server infrastructure team
- Used **Swagger** API description format for API documentation
- Worked on **Jenkins** for Continuous Integration
- Wrote Unit test cases with **Junit** for backend functionalities
- Used **GitLab** as a version control tool and used **Postman** to test API development

Environment:
Java 8, Spring 2.15, Project Lombok, Angular 7, Typescript, ES6, RESTful Web Service, MySQL, MySQL workbench, Mybatis, Maven, Oracle VirtualBox, Junit, Jenkins, Swagger, WebStorm, IntelliJ

Client: University of Phoenix **Feb 2020 – Aug 2020**
Role: Java J2EE/UI Developer
Address: Phoenix, AZ

Project: **Faculty Center/Program Workbench System**

The faculty center/program workbench systems are two modules used by University of Phoenix to manage students' academic/personal information, faculty performance and programs/ courses management. These two modules provide with the functionality to display/update students' information in multi campuses of UOP. The business purpose of these two modules is to help administrators to better manage students in multi programs/campuses more effectively.

Responsibilities:

- Responsible for **feature enhancements/code hardening & migration/defects solving/QA Testing/Daily Deployment** in both **Backend** and **Frontend**
- Involved in design of new UI pages for faculty reviews using **React 16.8**, implemented **Hook** to handle state/props changes for pages. Used 3rd party library such as **Material UI, React Router, Axios** for functional implementation. Configured page layout using **CSS**.
- Participated in functionalities enhancement of **React**, implements **Javascript Callback**/**Asynchronized** functions for multi-functional purposes to update/fetch data from backend Services
- Designed and Developed faculty review **microservices/REST** services to display/update student's faculty performances/reviews, using **Spring Framework/Spring Boot** with **MVC** pattern. Implemented **SQL** with technologies such as **JDBCTemplate**, **JPARepository**.
- Involved in Old Code Hardening/new feature enhancements in backend of Program Workbench using **Java6/ Java8,** implemented **Java 8 new features** such has **Lambda Expressions/Stream**, reduce time complexity of logic using **HashMap**. Implemented logic between several layers such as **Controller/Service/Dao**. Implemented Dao layer using **MyBatis**.
- Involved in code migration, migrated services in Program Workbench from **Struts** to **Spring** Framework
- Involved in old front code feature enhancements such as adding new columns for data model using **JSP/Ext JS**
- Designed and created data module for student's review in database using **Microsoft SQL**
- Used **Git**/**Bitbucket/Bamboo** for **CI/CD deployment**
- Configured **properties**/**yml** file for database connection both in local and cloud side, configured **docker** file for development, used **Azure** for cloud deployment
- Participated in **QA** testing, involved in **unit testing, integration testing**
- Worked in **Agile/Devops** environment with 2 weeks/sprint

Environment: Java8/Java6/Java11, Maven, Spring, Spring Boot, MVC, Struct, Hibernate, Microsoft SQL,MyBatis, React, CSS, JavaScript, jQuery, Extjs, AJAX, JSON, JUnit, JIRA, Git, SVN, Bamboo, Bitbucket

---

ViacomCBS - New York, NY **Oct 2019 – Aug 2020**

Software Engineer, DevOps

ViacomCBS is a multinational media, film and television company, parent organization of MTV Networks, Paramount, Nickelodeon etc. I was dedicated to the Broadcasting Software Development team under MTV group and responsible for all DevOps tasks include building, testing, deployment and monitoring.

Responsibilities:

- Monitored **TeamCity** build server and managed all build, test and deployment jobs.
- Provided supports for system and server incidents and troubleshooting as needed.
- Maintained and improved efficiency of **CICD (Continuous Integration and Continuous Delivery)** pipelines with the help of commercial or internal-use plugins and scripts.
- Migrated Pull Request build process from **TeamCity** to **Jenkins**, analyzed dependencies map, implemented **Jenkinsfile** that used to define **Jenkins Pipeline** and source control checkout using **Groovy**.

- Developed freestyle and scripted pipeline projects on **Jenkins**, including the whole process of building dependencies, installing packages, building components, running unit and integrated test cases, deployment and monitoring.
- Designed **Parallel build pipeline**, integrated it with Pull Request build, successfully improved the build time from 20 mins to 10 mins per build, with 10+ builds one day, save more than 100 mins build time one day for the whole team.
- Set up and configured 3 new build agents, which doubles the running nodes and enable distributed jobs running simultaneously.
- Brought **SonarQube** static code analysis tool into Pull Request build, let developers get code analysis report in PR build, before merging into master.
- Implemented internal use plugins worked with **Bitbucket** server, improved reliability of master branch and product robustness.
- Developed **PowerShell** scripts that improved automation build performance like 'Scheduled backups and clean old artifacts', 'Cancel old builds in same branch'…
- Worked on team monitoring dashboard using **New Relic One**.
- Cooperated with IT Systems and Infrastructure team to ensure hassle-free software development cycle.
- Created and maintained technical documentation on **Confluence** page.
- Participated in each phase of the **Agile** software development lifecycle i.e. daily standup, sprint closing/planning, retrospective, weekly 1-1, demo meetings.
- Used **JIRA** as ticketing system, Confluence for team documentation, **Bitbucket** for source control.

Environment:
Groovy, PowerShell, TeamCity, Jenkins, Bitbucket, SonarQube, JIRA, Confluence, New Relic One.

AT&T – Richardson, TX 2019 Dec. - Present
Java Full Stack Developer
Telecommunication platform

Description:
I worked in AT&T Halo.E project, Halo is an identity and access management consumer-facing web platform which company users can register/ eset password, update security profile, and apply for roles. Admin users can batch manipulate direct reporters, modify role access list, and create automatic jobs. I developed RESTFul services and microservices by using Spring MVC, Spring Boot. For frontend, we utilized AngularJS to construct the portal user interface.

Responsibility:
- Implemented RESTFul APIs/microservices such as Bulk password reset/ batch profile update using thread pool, stream API.
- Developed Jobs to query LDAP Process that can query the LDAP for a configured attribute to get the value, encrypt that value, and write the encrypted value to a new attribute in LDAP and Oracle.
- Implemented remove user security PIN and security answers job for credential compromised users.
- Wrote unit tests using Junit, Mockito, Power Mockito in TDD environment with minimum coverage of 90%.
- Fixed bugs and troubleshooted the defects for legacy Code.
- Created merge queries to insert or update records in database.
- Used veracode scanner to review and validate the content and code quality in security platforms.
- Discussed user stories' details and requirements with product owner, peers, and architect.
- Developed error message stack functionality for user viewing security profile page in portal with Angular JS.
- Used Log4j to monitor error logs and execute result as well as the function running time.

- Troubleshooted the application in Prod by checking logs using Splunk.

Environment:
Java8, Oracle, LDAP, Tomcat, REST API, Angular/Angular JS, Java Script, Veracode Greenlight Scanner, GIT, Putty, Splunk, Kubernetes

---

AT&T, Dallas, TX
January 2020 – August 2020
Java Developer
HALO E Order Management System
HALO(High Automation, Low Overhead) is a Service Delivery transformation initiative that allows orders to smoothly flow through the systems with minimum intervention and delays. The overall goal is to deliver service quickly and efficiently and improve the end-to-end Customer Experience. The project focuses on improving the order capture and order management activities. Halo addresses the need for a highly automated simplified experience for customers who want to purchase our mass market products and want them fast.

Responsibilities:
Backend:
- Used **Spring Boot** with build-in annotations to create applications
- Used **Spring MVC** to dispatch HTTP requests to controllers
- Used Spring Bean **Dependency Injection** via **Spring IoC**
- Generated **Microservices** architecture with **ECO** Pipeline, and set up configuration via **K8S (Kubernetes)**
- Used **DME2 to** transport JMS data using HTTP protocols
- Used **RESTful API** to fetch data from **Oracle** database
- Used **LDAP(Lightweight Directory Access Protocol)** to store and modify to store information
- Used **Object Pool** design pattern to create the **LDAP** connection, to avoid redundant resource usage
- Used **Singleton** design pattern for paged-search function for LDAP to guarantee that it executes in the same connection.
- Used **Factory** design pattern for different departments to make it easy to extend
- Used **multi-threading** and Spring **Async/EnableAsync** annotation to complete the massive modification of order information
- Used **Jxplorer** to browsing LDAP servers and LDAP Data Interchange Format files
- Used **Java 11** to develop backend, including new utility methods in **String** class (such as isBlank, strip, repeat, lines), **local variable** for **lambda expression**, and the **access control** for **nested class**.
- Used **XML, YML** and **property** files to do configuration
- Used **Maven** for dependency controls
- Applied unit test **Junit** with **Mockito**

Others:
- Worked in **Agile** Software Development Life Cycle, and used **JIRA** for project management. Participated in daily **Scrum** meetings following the guideline of Agile methodology.
- Used **Git** with **CodeCloud** for version controls

Environment:
Java 11, Java EE, Spring Boot, Spring Security, Spring IoC, Spring JPA, Spring AOP, Hibernate, Maven, Oracle, Microservices, RESTful APIs, Kubernetes, ECO, DME2, LDAP, Jxplorer, Junit, Postman, JIRA, Git, GitLab, Agile

---

Verizon Media
Baltimore, MD **Feb 2020 - July 2020**
Java Developer

Verizon Media focuses on the advertisement bidding system by following the rules of OpenRTB, providing a platform for the client to bid the placement on the Yahoo website. The features I have been involved in includes the mediation & exchange, which gather information from the user, location preference etc. . I also participated in the migration of jdk8 to jdk11.

Responsibilities:

- Developing new features using **Java 8** functional interface, lambda function, stream API
- Replaced Collection, List, Set with Java 8 Stream API. Once created, the instance will not affect its source, allowing multiple instances with one source
- Checked the intermediate operations which reduce the size of the **stream** in the right order
- Implemented **Factory Pattern** in Native class, and made sure superclass was abstract class or interface, removes the instantiation of implementation from client side
- Replaced parallelization by implementing **parallelStream**() method, once the source changed to the source of stream, Stream API automatically use the ForkJoin framework to execute
- Updated **OpenRTB Response** contract with **Native** fields, with **Spring Boot** annotation configuration
- Used **Jenkins** to build AMI to run the integration test before submitting to production
- Worked on migrating from jdk8 to jdk11 by upgrading the version of maven, Spring framework
- Supported the native type of the advertisement in mediation by following the spec of **DSP & SSP** from the rules of **OpenRTB**
- Updated **Flume dispatcher configuration** to support encryption by changing the properties file with ssl implementation
- Updated **mediation Flume client** to support encryption, making sure ssl enable to secure the component
- Wrote both unit test and component test using **jUnit.** Checked the test via Maven clean build before running in **Jenkins**
- Committed the changes on my local repository and pushed those modifications to the **github** repo, and pulled the newest version from the repo.
- Worked in **Agile/Scrum** development

Koch Industry-Wichita, KS
Role: Front End Developer (Vue.js)
Duration: Sep 2019 – Present
Description: This project is a rewrite based on the legacy system which used to show the energy consumption and generate the reports or dashboard in different time range, different format for users. This application had been used for many years by company like Nissan and Ford. User can log in to account with password and see the electricity usage reports or dashboards in different chart formats. Admin can see all the reports, dashboards and do management.

Responsibilities:

- Followed the legacy system to build the **Single Page Application** with **Vue.js, VUEX**, **Nuxt**.**js**, **highcharts** and the **BootstrapVue**.
- Helped to make the webpage responsive with **media query** and **BootstrapVue**.
- Made the application accessibility with **ARIA** attributes.
- Built the status widget component for dashboard view page, use **props** and **event-bus** to get data from a different component. The data show in status widget can add threshold value with different color, the widget in dashboard will show the color depends on the threshold value.

- Built the back-button component with vue props, which make the button works in different components and can return back different url because of **Nuxt.js routing**.
- Built different page for search function in reports and dashboards. If the search result existed, show the "no result" page, otherwise, show the results page.
- Helped to maintain the display pattern for reports and dashboards function with **Highcharts** attributes.
- Provided French, Spanish, Japanese language dropdown selection for all the texts in application with **internationalization**.
- Followed **Agile** methodology for development.
- Used **Azure devops** for CI/CD platform and for version control.

Environment:
Vue.js, VUEX, nuxt.js, HTML5, CSS3, BootstrapVUE, JavaScript, TypeScript, ASP.net, SQL server, Azure devops, Git

---

Client: UBS AG, New York City, NY **October 2019 – Present**
Role: Front End Developer
Project: OCR Search Application + ETL Workflow Editor
Developed single page web applications that consisted of a smart search bar that could search and autocomplete keywords, tags, and date ranges. The application displayed original PDF of documents and OCR translated text and allowed users to select sections of text to tag and persist in database and highlight. The workflow editor was created where nodes represented transformers, loaders, and models. Users could drag and drop these nodes to create edges to construct a workflow. These nodes had inputs for parameters and users could load files and inputs for the nodes. Users could save certain workflows and execute these workflows from the browser that communicated with our Flask server.

Responsibilities:
- Implemented stateful and stateless React components using React Hooks and local state management
- Used CSS3 and LESS to provide animations and UI design
- Used MaterialUI and Ant Design libraries for specialized components
- Helped design and implemented **RESTful** Web Service endpoints
- Used Axios to make API calls with appropriate headers
- Used PDF.JS to render PDF streams from backend server
- Used **React Redux** to store critical information in application and created actions and reducers
- Created workflow canvas using Ant Design and GG-Editor
- Participate in **Agile** development and used Jira for project management

Environment:
React, JavaScript, CSS3, LESS, HTML5, Ant Design, MaterialUI, Axios, GG-Editor, Mozilla PDF.JS

Pearson Vue – Bloomington, MN
Sept. 2019 – Present
Role: Full Stack Developer
Project: PTE Enhanced Candidate Journey

Description:
This project is to add new features to the exam scheduler application. Candidates can use it to schedule their

exams and many education companies use this application to provide their exams. As a full stack developer, my main responsibility was to implement a side navigation bar to the existing application.

Responsibilities:

- Found a proper JQuery library that can match the sidebar needs
- Edited the original **JSF** file logic. Generated two files for the new sidebar and the original top bar. Both files reused an old file which has a list of links on it. Passed parameters to the reused file so it can display different links for the new sidebar and the original top bar.
- Created a decision **java bean**, so frontend can decide to use sidebar or the original top bar by the output of the decide() method in this java bean file.
- Injected decision bean into **xhtml** template using **<c:choose>, <c:when>, <c:if>** elements
- Created a right sidebar for right to left language such as Arabic.
- Implemented accessibility by adding 'Enter' keyboard listener on the sidebar trigger button by **JQuery**.
- Implemented the animation of the sidebar by **JQuery** and **CSS**
- Used **CSS media queries** to switch between desktop view and mobile view
- Used Font-awesome to add icons for each link
- Stored sidebar's expanded-collapsed status on cookie by local storage.
- Developed back-end API for the SSO to return to client URL.
- Developed back-end API for schedule flow and submit order by concurrent programming, such as building **threads** and **locks**.
- Developed back-end API to fetch data and build an order summary page.
- Committed my changes on my local, pushed those changes to the repository, and pulled the newest version to my local by Git tool.
- Retrieved current assignments on VersionOne with estimated points
- Reported my progress on the stand-up meetings every morning

Environment:
VersionOne, Java 8, JSF, CSS3, JQuery, Git

Konica Minolta, Ramsey, NJ **May 2019 - Present**

Role: Full Stack (Java/Angular) Developer

Project: Konica Minolta MyKMBS Portal – A portal for customer to search, monitor, and analyze their devices and to order supplies for their devices. This is a migration project from legacy project.

Responsibilities:

- Developed Order Supplies Module and Order History Search Module.
- Built Restful web service with Spring MVC framework, Restlet for server side and Angular 7, NGRX for client side.
- Developed web pages with Angular 7, NGRX, Bootstrap 4 and Kendo UI, including components, services, models, stores, actions, reducers, effects.
- Built reusable components with ngIf, ngClass and ngStyle.
- Built breadcrumb to implement website navigation and traction stepper.
- Built nested forms with Kendo Grid, Angular FormArray, FormGroup, @Input, @Output and EventEmitter to

implement complicated data format editing and submission.

- Used Spring MVC framework and Restlet to handle web requests.
- Applied Java 8 new features to implement main functionalities, like functional programming, concurrent package etc.
- Applied Spring AOP to test performance of service and log necessary data process steps.
- Used Gson to do data transformation and data clean (adapter).
- Used Maven and Jenkins to do project management and Git to do version control.
- Used Log4j to do logging.
- Applied Postman to do server functionalities test.
- Unit test with JUnit, Morkito for Java and Jasmine, Karma for Angular.

Environment: HTML5, CSS3/SASS, Typescript, Javascript(ES6), Restful, AJAX, Angular 7, Java 1.8, Java Mail, Spring 4.2.1, Spring AOP, PL/SQL, Git, Junit, Mockito, Karma, Jasmine, Maven, Log4j, Postman, MySQL

Client: **JPMorgan Chase - Columbus, OH**
Position: **Senior Web Developer / Tech Lead**
Duration: **May 2016 – August 2019**
Projects: **Appkit-Utilities, BlueJS Framework, Chase 3.0**

JPMorgan Chase is a multinational banking and financial services holding company headquartered in NYC. This project is to extend and maintain the existing complex web applications developed by JPMC. My duty is to participate in a scrum team and develop generic modules and optimal solutions to be used across different applications and teams.

Responsibilities:

- Experienced in various phases of **Software Development Life Cycle**, such as conducting Requirements Analysis, Application Design, Application Development and Maintenance.
- Participated in software development using both **Agile and Kanban Methodology** and met bi-weekly sprint requirements by joining daily scrum stand-up meetings and achieving daily goals.
- Managed day-to-day operations of Appkit-Utilities project. Main duties as a Tech Lead include:
    - Participated in weekly prioritization calls, troubleshooting calls, collect business requirements, communicate and collaborate with teams in NY, SF and offshore teams in India.
    - Evaluated, redesign existing modules, or create new utilities based on customer requests. Works are done in generic fashion so they can be used across multiple applications.
    - Maintained a reference application to show descriptions and demo usages.
    - Wrote release updates notes and make periodic product releases.
    - Expanded documentation on individual modules and customer support.
- Proficient in **modular programming** to design and create small modules using **Node.js** for easy distribution, version control and dependency management.
- Proficient in **functional programming** using reactive libraries such as **Keflr.js** and **Bacon.js**.

- Handled **Cross Browser Compatibility** across browsers, including chrome, firefox, IE and safari.
- Handled **Cross Platform Compatibility**, including Windows, Mac, iOS and androids.
- Provided solutions and worked with **Hybrid application** between native apps and web apps.
- Utilized **JSP Session Scope Object** to maintain client status using **Session, Cache** and **Cookies**.
- Regular application and framework **validations** and **debugging** to ensure application qualities.
- Ensured code quality by conducting **Unit Testing** using **Chai** and **Sinon** with **Karma**, bringing up the test coverage of the project to as high as **94%** and generated reports using **Blanket.js.**
- Utilized **Git, Stash** and **Bitbucket** to achieve **Version Control** and used **JIRA** for bug tracking.

Technologies:

HTML5, CSS3, Bootstrap, JavaScript, jQuery, AJAX, JSON, Node.js, Lodash.js, Unix, VIM, Jasmine, Chai.js, Karma, Blanket.js, Sinon.js, Meld.js, Ractive.js, Keflr.js, NPM, Gulp, GIT, JIRA

Company: Apple, Sunnyvale, CA
Position: Java Developer
Apr.2019-Present
Project: Apple Support Applications JDK11 Migration

Apple is maintaining many support applications to provide the customers online after-sale services. Currently most of the apps using Spring as their backend framework are going to migrate from JDK8 to JDK11. Some apps which are relatively low weighted were decided to upgrade at first. I was working on 3 of these projects and they were all released successfully and are running with JDK11 in production.

Responsibilities:

- Followed **Agile Project Development** and used relative Apple software to work with the team for the projects. Attended daily standup meeting the synchronize with the team members.
- Worked on migrating from **JDK 8** to **JDK 11** for the backend upgrade.
- Used **Spring Framework 4.3.22** to develop backend services and used **Maven** to manage and import new dependencies to make them work with JDK11.
- Worked on **Junit** and **Mockito** for backend unit test to make the services always maintain the same functions as before migration.
- Used **Postman** to send requests to test apps and **Jmeter** to test the Java application performance.
- Used **MongoDB 3.6** as the Database to keep the data. Worked on **Mongo Shell** to manually load user data to the Database regularly.
- Used **slf4j** to log the backend running information and **Splunk** to check the logs on servers and query specific logs for quick debugging.
- Built the apps with **Jenkins** and deployed them on **Ansible** for CI/CD.
- Worked with **Git** to manage the code and control the version and used Apple internal software **Radar** to report and track bugs.

ADP-Roseland, NJ

Role: Front End Angular Developer

Duration: March 2019 – August 2019

Description: This project was a migration form an old project to a new one written in Angular. The name of this project is Managed File Transfer(MFT). Which is used for the associate in ADP to get and download the files related to the client. Also

it's a platform for client to upload their files to ADP with the information they need. This project belong to a bigger system named adminUI. For the adminUI. After finished the MFT, they need to migrate the adminUI with Angular.

Responsibilities:

- Followed the old project layout and built the Single Page Application by **Angular5+**, **TypeScript, JavaScript** and the ADP synerG component library.
- Built the dashboard page for the MFT and shows the information like username, last login time, user settings and user upload files details, which get the data form the backend.
- Built the layout of the dashboard so it will be the same as the old project.
- Created the component and the service to show the error page to some specific users when they log in the MFT system.
- Helped to fixed the defect for user settings auto refresh.
- Built the transmission search page for the adminUI which have several input boxes, the dropdown list, the date picker and the search, reset, advanced search and close search button.
- Wrote the button function for transmission search so it can do search, reset, advanced search and close the search box.
- Used the **router**-**link** to help access in different part of the SPA.
- Used the **tableplus** to get access to the **Cassandra database**.
- Followed **Agile** methodology for development.
- Used **RALLY** for tracking project and record the bug.
- Used **Git** and **Bitbucket** for version control.

Environment:
Angular5/6, SynerG component library, HTML5, CSS3, JavaScript, TypeScript, Bitbucket, Cassandra database, Git, Rally

---

Client: KeyBank – Cleveland, OH
Position: AI Chatbot Engineer/JavaScript Solution Lead
Duration: Oct 2019 – Feb 2020
Projects: KeyBank is a regional bank headquartered in Cleveland, OH. This project leverages Google's Dialogflow, a natural language processing (NLP) platform, and other services on Google Cloud Platform to create AI virtual assistants to automate some core features and services provided by the internal help desk support center.

Responsibilities:

- Worked directly with other product owners to define features and groomed technical stories for sprint planning
- Provided expertise and training on various JavaScript topics and best practices to team members
- Participated in **System Architecture Diagram** design and **Flowcharts** to demonstrate workflows and concepts
- Identified and patched potential security vulnerabilities within the agents, applications, and interconnected systems
- Built good and effective working relationship with DevOp team and Cloud Native team to meet project milestones
- Collaborated with other chatbot engineers and companies to automate help desk support with virtual assistants:

    - Defined core features such as definition lookup, password reset/unlock, and incidents/requests lookup
    - Analyzed requirements and designed **Intents** with **Training Phrases**, **Action**, **Parameters**, and **Responses**
    - Generated list of **Entities** and **Slot** to increase the understanding ability of the virtual agents
    - Created follow-up/trigger **Events** and managed **Contexts**/**Session Storage** to enhance customer experience
    - Set up Dialogflow Agents with **Webhook** calls for back-ends **Fulfillments**/services
    - Redesigned, modularized, and rewrote the AppEngine virtual agent code bases to enhance overall security and performance of the front-end and back-end applications
    - Enabled **Basic Access Authentication** and followed principle of least privilege to enforce **Access Controls**
- Hands on experience with other cloud services provided by Google Cloud Platform:
    - Evaluated and compared the performance of Cloud Functions, App Engine, and **Kubernetes** Engine
    - Managed **IAM** & **Admin** for access control and created **Roles**, **Groups**, and **Service Accounts**
    - Monitored application usage and user access by creating **Metrics**, **Alerts**, and **Policies**
- Created custom connecters to bridge different IT systems into one integrated system
- Set up **Unit Testing** environment to safeguard code quality using **Mocha** and **Sinon**, bringing up the test coverage of the projects to 80**%** as the requirement of first-day production release**.**
- Regular applications **validations** and **debugging** to enhance overall products quality and user experience
- Participated in discussions for **Risk management**, Disaster Recovery Plan and Contingency Plan
- Developed using **Git** and **Bitbucket** for **Version Control**, **JIRA** for bug tracking, and **Confluence** for documentation

Technologies:

Dialogflow, Google Cloud Platform, Amazon Lex, AWS Amplify, JavaScript, Node.js, JSON, Jasmine, Chai.js, Karma, Sinon.js, GIT, JIRA, ServiceNow

---

Project: Express Membership 1.0+
Company: Walmart Sam's Club, Bentonville, AR
Role: Full Stack Angular Developer
Duration: July 2019 – Present
Description: I worked on development and test of both front-end and back-end for Sam's Club membership system.

Responsibilities:

- Front End:
    - Developed Responsive HTML web pages using **HTML**, **CSS, SASS, TypeScript.**
    - Used **Angular**, and **Angular Material** to build rich, interactive user interfaces.
    - Used **Jest** and **Jasmine** to perform unit testing of services, components and pipes. Mocked the API response by **Jasmine**
    - Developed form by **Angular Material** form component to manage the form value and implemented validation for user input.
    - Used **ng2-charts** to visualize data from the server to build the histogram of report and **lodash** to simplify the complexity processing of data.
    - Implemented the library **numeral** formatting and manipulating numbers and library **moment** for parsing, and formatting dates.
    - Consumed **RESTFUL API** with **Fetch API** (ES6+) request from client side to server with JSON format

data and used **Apollo-Client** to create **GraphQL** request to fetch multiple resources.
- Customised upload component of ant-design to implement upload(post) files/images to the AWS S3 bucket from the front-end by AWS-SDK.
- Used **zone.js** to wrap asynchronous task for tracking, errors log and error tracing in development.
- Developed services such as API services, error handler and utils services, that can be utilized across applications.
- Assigned stories to my track and get progress report from the team using **JIRA** and used GIT for version control
- Used **Jenkins** to build, test and deploy the application automatically for continuous integration.

- Back End:
  - Created Spring based applications using **Spring Boot** to implement auto configuration and used embedded web server **Tomcat**.
  - Wrote unit and integration tests to test all the desired functionality using **jUnit** (TDD) and mock response with **Mockito**.
  - Implemented **Swagger** to visualize and interact with the API's resources.
  - Established Continuous Integration and Continuous Deployment(CI/CD) practices and standards in creating APIs with JIRA, Jenkins
  - Designed and developed a MS **SQL** database based external Restful API for exposing the functionality of existing platform.
  - Applied **SDLC** including investigation, analysis, design, implementation and maintenance.
  - Developed a RESTful based service and API for data retrieval to make it independent of the current application.

Environment:
Angular 5.1.2, HTML5, CSS3/SASS, Angular Material 5.0.2, TypeScript 2.4.2, Jest 22.0.1, Jasmine 2.8.6, Apollo Client 2.6.4, lodash 4.17.4, Java 8, Spring 4.0, Spring Boot, Spring Security, Spring MVC, MS SQL Database, Junit 5, Mockito 2.3

Project: SMART OPS Board System & Minerva **Oct 2014 – July 2016**
Client: Sanitation Department of New York, Manhattan, NY

Description: SMART OPS Board System is a system used by DSNY employees to dispatch people and equipment, make task arrangements. This system changes the traditional way of department management into high efficient, user friendly way. It helps tacking people and equipment, making plan in advance, and displaying boards based on district and date for employees. Minerva breaks up the monolithic into Microservices by using Cutting edge technologies.

Release One: SMART OPS Board System
Role: Full Stack Developer
Responsibilities:
- Participated in most phases of the Software Development Life Cycle (SDLC). Involved in design and development of system and developed specifications writing Use Cases, Class Diagrams, and Activity Diagrams.
- Extensive use of core Java collections, generics, exception handling, I/O, in back end development and logic implementation.

- Used SAX, DOM for XML data generation and presentation.
- Implemented customized authentication in Spring Security to verify user-role and secure the user pages, such as login, admin/user operations, etc.
- Extensively used Spring Dao Support to access the database efficiently.
- Used multi ways in Hibernate to update data, such as HQL queries, load or get, SQL, criteria.
- Applied multithreading to accomplish synchronization hierarchy in both class and object level synchronizations to support multi users operations.
- Customized RESTful Web Service using Spring RESTful API, sent JSON format data packets between front-end and the middle-tier controller, robust massaging by using RabbitMQ.
- Confirmed SSL certificate and configured server.xml under Tomcat 7.0, applied HTTPS certificate encrypt data transfer secure the web access.
- Applied GoF Design patterns for the back-end design: Command, Factory and Proxy.
- Created JUnit test cases for Unit Testing on Java Class, and participated in Integration Testing.
- Responsible to check in and check out the developed code using SourceTree and Git.
- Attending daily and weekly SCRUM meetings to update the work status, issues clarifications etc.
- Used JIRA for project bug tracking, issue tracking and management.

Release Two: Minerva
Role: J2EE/Java developer
Responsibilities:

- Participated in most phases of the Software Development Life Cycle (SDLC). Involved in design and development of system and developed specifications writing Use Cases, Class Diagrams, and Activity Diagrams.
- Extensive use of core Java collections, generics, exception handling, I/O, in back end development and logic implementation.
- Extensive use of Java RX and simplify code by using Java 8 API.
- Customized Spring Boot to simplify configuration steps and files.
- Utilized Docker to pack multi sub projects dependencies into standardized units, based on Microservices architecture.
- Used Spring Cloud to build common project, discovery service, configuration service, etc.
- Customized RESTful Web Service using Spring RESTful API, sent JSON format data packets between front-end and the middle-tier controller, robust massaging by using RabbitMQ.
- Used MongoDB to store data, and insert and export data by using MongoDB Aggregate Query.
- Tracked Unit Test Coverage and ensure code quality by using Sonarqube.
- Created JUnit test cases for Unit Testing on Java Class, and participated in Integration Testing. Applied Mockito framework.
- Responsible for check in and check out of the developed code using SourceTree and Git.
- Attending daily and weekly SCRUM meetings to update the work status, issues clarifications etc.
- Used JIRA for project bug tracking, issue tracking and management.

Techniques: *SDLC, Core Java, Java RX, Agile Methodology, Design Pattern, Spring MVC, Spring Boot, Spring Security, Spring Cloud, MicroServices, Event Sourcing, Docker, VM VirtualBox, RabbitMQ, Web Socket, SQL, CQRS, Sonarqube, Junit Test, Gradle, Git, Oracle, MongoDB.*

Client: **State Street Corporation, Boston, MA** **May 2016 – Oct 2016**
Position: **Sr. Web UI Developer**

Project: Data Management System
State Street Corporation, known as State Street, is an American worldwide financial services holding company. The Data Management System is used to add, save and retrieve data and is based on an internal database.

Responsibilities:
- Extensively involved in design, implementation, testing and maintenance phases of Software Development Life Cycle (SDLC), using Agile methodology.
- Used React.js to combine HTML and Javascript as JSX, Redux as predictable state container, Webpack as module bundler to build UI.
- Developed user interface and implemented logic by using React.js and made data transportation by state and props.
- Initialized the state by using Constructor, state object and propTypes object.
- Get reference to the actual DOM node element by using ref. Changed state by using setState function and fired render by using render function.
- Built the entire lifecycle by using built in functions like componentWillMount, componentDidMount, componentWillReceiveProps, componentWillUpdate, componentDidUpdate, componentWillUnmount.
- Complemented filters, drop lists, menus, zooming, selector panel, sorting, table, refresh etc. by React.js.
- Implemented user validations by customized function with HTML5 Form Validation.
- Adopted Bootstrap layout/grid system to implement Responsive Web Page.
- Used React-Bootstrap to build Buttons, Navigation, Overlays, Page Layout, Tabs, etc, and implemented user experience with DatePicker in Bootstrap.
- Fetched/Saved data from customized ESP database by AJAX call and composed data as specific format for rendering.
- Used npm based on Node.js to manage modules.
- Involved in Eclipse, Linux System for design and development.
- Used Git for version control, Squirrel SQL Client for database administration.
- Experience working with offshore team. Hosted daily teleconference with database team in China and London. I was responsible for assigning task and tracking progress.

Environment:
React.js, Redux, Webpack, JSON, AJAX, Node.js, Git, Shell Scripting, Linux, Squirrel SQL Client, ESP data management tool, JDK1.8, Apache Tomcat 8.0

Client: JPMorgan Chase&Co., 4 New York Plaza NY **July 2014 – Jun 2016**
Role: Full stack Developer
Project 1: Private Bank Account Opening

Project Description:
In the wealth management department, Account opening system is used for registering account for institutions, agent and other group who use JPMC services. This web application provides multiple of rules, documents and reference. Clients can use these information finish the account opening.

Roles and Responsibilities:
- Implement soap and rest **WebServices**. Apache CXF.
- Familiar with **AngularJS** and **Customized Directive** making reusable components that fill the request, and also worked on **java script and bootstrap**.
- Worked closely with **UI** Design team and used **HTML5**, **CSS3**, **JavaScript**.

- Implemented **Spring MVC** framework, project Build by using **Jenkins** and deployment by using **ARM**.
- Implement database work by using **SQL developer, TOAD**
- Designed the business object with **Hibernate** and using **Spring DAO**. Responsible for writing mapper files.
- Used **Git** for version control.
- Working on **DEI(Development Efficiency Index)** and **Sonar Analysis and Selenium**, on testing, bug fixing, and code coverage improvement
- Tracking the issues in the **JIRA** and specifying the solution

Client: JPMorgan Chase&Co., 4 New York Plaza NY
Role: Full stack Java/Angular Developer
Project 2: Reporting System

Project Description:
The Reporting System was built for JPMC Global Investment Management department. Reports contain different kinds of statistical information as requested by the user. The entire webserver was built using micro services design model, it provides many different services such as generation of the Mutual Fund Fact sheet, Performance service and Characteristic service. This system can also auto generate Clients report for US Equity and Fixed Income.

Roles and Responsibilities:
- Expertise in **One/Two way Data Binding, MVC Pattern** to organize our web application properly, and used **Dependency injection** to test our code.
- Used **AngularJS** modules such as **Controllers**, **Services**, **Filters**, and **Directives**, to build well structured web pages, and made **Ajax** call to communicate with **Restful WebServices**.
- Implemented **Customized Directive** making reusable components that fill the request, and also worked on **Angular form Validation**.
- Worked closely with **UI** Design team and used **HTML5**, **CSS3**, **JavaScript**.
- Implemented **Spring MVC** framework, project Build by using **Jenkins** and deployment by using **ARM**.
- Edited Technical Document (Run Book) for CPM Report Builder.
- Implement database work by using **SQL developer, TOAD**
- Designed the business object with **JDBC** and using **Spring DAO**. Responsible for writing mapper files.
- Used **SVN** for version control.
- Working on **DEI(Development Efficiency Index)** and **Sonar Analysis and Selenium**, on testing, bug fixing, and code coverage improvement.
- Developed test plans, heavy coding on **unit test**, by using **JUnit** and **Mockito**. Used **Log4J** to log the debug, error and informational messages at various levels.
- Worked in **Unix** and used **shell script**.
- Tracking the issues in the **JIRA** and specifying the solution
- Implemented complex business logic as PL/SQL stored **Procedure, Function, Package and Triggers** by using Toad.
- Involved in all phases of the **SDLC (Software Development Life Cycle)** including analysis, code development, testing, implementing and maintenance and worked in fast paced **AGILE** environment.

Environment: JDK 1.7 J2EE, Mockito, Spring, Hibernate , Rest, Birt, XML, Design Patterns, Oracle, PL/SQL, JavaScript, JUnit, JavaScript, Eclipse, Unix.

Client: JPMorgan Chase&Co., 4 New York Plaza NY
Role: UI/Java Developer
Project 3: Trade Real Location

Project Description:

Trade Real Location is one of the components of the new stock trading system for JPMC. This portal records various trading information. Customers can do requests to transfer asset to different location by correspond trading engine.

Roles and Responsibilities:

- Implemented Graphical User Interface (GUI) of the single page application using **Ext JS**, **JQuery**, **AJAX**, **JSON** and **XML.**
- Implement **ExtJs** web application logic in controller, and involved in Design View, manage Store.
- Working on **ExtJs** and components like Grids, ToolBard, Panels, Combo-box and button etc.
- Implemented the UI logic by creating controller and correspond service.
- Used **Selenium** IDE for Open source web testing
- Designed the business object with **Hibernate ORM** mapping and interfaced with the business managers using **Spring DAO**.
- Worked with offshore team and doing **regression** test and **black & white box** test.
- Involved in all phases of the **SDLC (Software Development Life Cycle)** including analysis, code development, testing, implementing and maintenance and worked in fast paced **AGILE** environment.
- Used **SVN** for version control.
- Implement database work by using **SQL developer, TOAD**

Environment: JDK1.6, Eclipse, J2EE, Spring 3.0, Hibernate 3, Oracle, Apache Tomcat, JavaMail, RESTfulWevService, XML, JUnit, CSS3, JavaScript, JQuery, JSON, AJAX, Bootstrap. SVN, Unix shell script, Toad.

Client: MarketAxess – New York, NY (06/2018 – 05/2019)
Position: Java/Full Stack Developer

Project: BondLink is a platform built on Java (JDK 1.8) that allow institutional dealers and buy side clients to inquire, negotiate, buy, sell and post bonds through a set of API messages under FIX protocol. It has a GUI workstation for clients who do not trade through API, but my team was focusing in its API trading aspect.

- Implemented RTS/Trading API/Post Trade Messaging aspects with FIX protocol 4.2/4.4/5/0SP2 for 4 new bond products from Emerging markets, with ForeignExchange negotiation part of the trading process. Applied Proxy, Factory, Decorator design patterns in the message enrichment process, then sending back responses to all parties involved, meanwhile persisting data to Oracle 12g database.
- Analyze, design and implement enhancement features with business analysts and architects to enhance server memory performance, reduce respond lag, optimize system efficiency in Presentation Server layer by applying a better event queue design, replacing instrument objects with reference IDs and use ConcurrentHashMap to cancel out duplicated events that was sent multiple times.
- Manage FIX message APIs in trading platform and update features including batch processing inventory instruments, post trade reporting, local market settlement FX protocol, etc.
- Manage XML mapping in FIX Gateway for the data communication between API layer and Application Server to handle requests to translate FIX messages to Java objects.
- Maintain/Update the data model for persisting instrument/inquiry data in servers to Oracle databases according to design requirements from BA team.
- Developed technical enhancements using Core Java features, such as Collections Framework, I/O, Generics, and Exception Handling; used Multi-Threaded design to run concurrent tasks to improve the performance of the system.

- Perform sanity tests and organize QA environments builds for servers in Ashburn, VA

Project: MA Portal is a browser based platform that manages the post trade side of the business, clients can extract trading history, update and amend existing trades, explore market information, and also monitor post trade settlement process.

- Implemented different Bond Data views for InstrumentAdmin, pulling data from Oracle database and enrich raw data with different workflows, then display them with AngularJS.
- Clean up defects in backlog, fixing issues throughout different aspects in the project from database persistence layer to front end browser issues. Maintain and fix issues from third party frameworks.
- Utilize asynchronous messaging service to send and receive the object messages between different sub systems of the application, communicate between different servers with SonicMQ JMS service.
- Monitoring quarter release build process and fix issues on demand.

Client: Paychex – Rochester, NY (11/2014 – 04/2018)
Position: Full Stack Developer

Project: Paychex Flex Portal - Migration Phase: Employee Journey is an HR management tool used to view and edit employee's payment record, direct deposits and personal information. This project utilizes AngularJS 1.6, HTML5, Java 8, Spring and Web Components/Services built in the previous Framework phase to carry out the designs from UI and business logic analysis team.

Front End
- Developed multiple sub projects using customized, reusable Angular Directives. Implemented services for different components to share data and communicate.
- Implemented ngRouter to navigate between views on the Flex Portal SPA.
- Retrieve and wrap composite data into ngModel for a variety of forms across different applications.
- Optimized app performance on both desktop and mobile platforms.
- Wrote unit tests in Karma/Jasmine for new features every sprint; wrote automated end-to-end tests with Protractor; wrote automated testing with internal testing tool.
- Built RESTful API for transferring data in JSON format between the system and remote server.
- Performed code review and uploaded to BitBucket repository to ensure style consistency and code quality.
- Built and tested code with pre-built ANT/Grunt tasks before submitting to repo; fixed unit tests and ensured continuous integration on Jenkins build successfully; document defect groups and provide logistics to Splunk tickets while on rotation duty.
- Saved the project $5,000 per year by designing and implementing Paychex Walkthrough Framework, instead of using third-party software license.

Back End
- Utilized multi-thread calculation to boost tax calculation speed for clients.
- Applied BlockingQueue to achieve synchronized accessing issue in multi-threaded environment.
- Configured the Java application environment and Spring IoC by utilizing Spring Boot to handle Java Beans.
- Implemented the Java middle tier using Spring and Hibernate, used controllers and services to handle request from Spring MVC and respond to calls from front end.
- Applied Spring AOP framework for log management, used Spring Data JPA for connecting with Oracle 12c and designed DAO layer.
- Implemented business logic for web services, using Spring Annotations which enables dependency injection.

- Utilized Java 8 features like Lambda expression and forEach loop to make code concise.
- Used SourceTree as version control system for history tracking and branch management in project codes.

Others
- Implemented, tested and delivered Paychex Flex portal with an Agile team, reviewed JIRA board in daily scrum meetings.
- Deployed applications on the cloud with AWS EC2, S3; deployed web apps on AWS Elastic Beanstalk.

Project: Paychex Flex Portal - Framework Phase: This project was to analyze, design, and implement the landing framework, a reusable framework, and provide internal services that can be consumed across the app from different teams and projects, and provide support for other teams' updated requirements and future needs.

Front End
- Managed the lifecycle of various sub-applications written in HTML5, Flex, and legacy HTML.
- Created a new Paychex internal framework focusing on performance, reusability and a design to enable developers from other teams to achieve their goals in their projects.
- Implemented form-framework (a UI framework for forms that wrap components from Google MD) for other teams to consume in their individual sub applications.
- Used cascaded Promises in application to retrieve data from back end with RESTful API.
- Used Splunk as log management system to fix and monitor defect. Documented and distributed defect tickets for different teams to fix.
- Refactored code with JsHint/ EsLint to promote coding style/code quality, delivered the code with test cases that have more than 85% coverage rate.

Back End
- Configured the Java application environment in Spring Boot container to manage Java Beans.
- Applied Core Java features to business logic in different scenarios, including collections framework, reflection, exception handling, generics and annotation.
- Extensively used Java Collections (ArrayList, ConcurrentHashMap, etc) to make implementation efficient and versatile.
- Used Hibernate to invoke Stored Procedures and database connectivity
- Created tables in database based on Entity-relationship model that is provided by the architect.
- Implemented Unit testing, Integration testing and Regression testing.
- Used Jenkins for continuous build and integration build.

Client: Welch Allyn – Syracuse, NY (05/2014 – 11/2014)
Position: Java/J2EE Developer
Project: Online Product Configurator is a web-based configuration tool to alleviate the time-consuming process of configuring parameters for clients in the Healthcare industry. It can specify over 200 complex parameters through a set of relatively simple, quick, smart and interactive web-page based questionnaires, then store the data in cloud server, ready for engineering team to configure products before shipment.

- Used Bootstrap, JSP, HTML5, JavaScript, jQuery and CSS3 for UI design & development. Tuned graphic using CSS3 through jQuery; managed front-end page integration using Struts 2 framework with Tiles to assort and stitch different parts of pages.
- Utilized Bootstrap slider, progress bar on webpage to implement features and minimize development time.
- Built dynamic HTML tables using jQuery, append/delete rows of data on both UI and server side, as well as

manage lifecycle of the components on initialize, reload, and destroy manually.
- Wrote custom Struts 2 validator in a Java file, then applied on the XML based Struts validation framework to fulfill special server-side validation demands.
- Developed Ajax validation calls for client side when the user attempts to submit a form and used jQuery to dynamically validate each field.
- Implemented Struts2 MVC architecture to determine 3 different work flows for 3 distinctive groups of clients.
- Implemented the business logic on generated zipped core XML file; walked through each field with engineering team about the complex logic relationships between a large number of parameters.
- Tested & fixed bugs using Log4j, documented reported issues and requirement updates on JIRA.
- Built caches in MongoDB for fields with prefixed-value sets.

Client: Morgan Stanley, New York City, NY **Sep. 2016 - May 2017**
Position: Sr. Java/J2EE Developer
Project: Electronic Signature Platform
eSign platform is a fully digital alternative to previously paper-based method for obtaining new account documentation from clients. Packages including all documents can be prepared through eSign portal and send to clients through DocuSign by email. Clients will also receive PIN by text or call back to access the package. After clients signed documents, eSign platform will receive notification from DocuSign and forward to manager for approval.

- Generated new notification event in database and showed on UI after package was signed through DocuSign.
- Implemented prefill fields on AACA-Business forms for (SOL, PAR, LLC) account types in different flow.
- Designed and created ClientAuth tables' schema in DB2 database.
- Designed and implemented ClientAuth persistent layer using **DAO** pattern.
- Done object relational mapping using **Hibernate** annotation and **JPA**.
- Implemented **Spring** service to parse ClientAuth request and persist to database.
- Used Spring **transaction** annotation to manage database transaction.
- Used **postman** to send request to test ClientAuth functionality.
- Done analysis, design and implement Callback PIN delivery for both frontend and backend.
- Implemented Callback PIN delivery front-end part by using **AngularJS**, **JavaScript**, **HTML** and **CSS**.
- Added IRA and VSLOA form when create package by using JavaScript, validate user input using **AngularJS** and **JavaScript**.
- Designed and implemented admin screen front-end for DB operation by using **JavaScript**, **AngularJS**, **AJAX**, **JSP**, **HTML** and **CSS.**
- Designed and implemented admin screen for DB operation server side controller and services by using **Spring REST**, use Java **Reflection** to implement extensible framework.
- Wrote unit test and integration test cases on various modules by using **JUnit** and **Mockito**, and pass **Sonar** coverage requirement**.**
- Used **Splunk** to analysis log, **JIRA** for bug tracking, **Git** and **Bitbucket** for version control**.**
- Resolved defects managed by JIRA and ALM based on business requirement.
- Involved in all phases of software development life cycle (**SDLC**) in agile environment.
- Done continuous build, integration build using **Teamcity.**

Environments:
IntelliJ, Eclipse, Tomcat, JDK1.6-1.8, J2EE, Spring 4, Hibernate, JDBC, HTML5, CSS3, Bootstrap, jQuery, AngularJS, JSP, JUnit, Mockito, Git, JIRA, Ivy, Gradle, Teamcity, DB2, Splunk.

Client: Goldman Sachs, Jersey City, NJ **Nov 2015 - July 2016**
Position: Sr. Java/J2EE Developer
Project: Security Trade Processing System
Hydra is a middle office post execution trade processing platform. It is a large high-throughput, distributed message driven system, it receives millions of orders and executions from front office through JMS. Different services such as allocation capture, trade explosion, trade aggregation, customer confirmation generation, booking transactions coordinate with each other to generate different business objects according to their lifecycle.
Titan is an event driven message delivery system, used to process input transaction data and generate documents and deliver to client via different channel. And it also includes web portal providing account confirmation and backend RESTful service.

- Involved in Jordan river capacity test, implement timestamp generation class to generate timestamp with configuration file.
- Added timestamp to each EMS message of CDL file and schedule to replay them according timestamp.
- Wrote Zebra script to configure capacity test and run automatically.
- Done capacity test performance analysis and tuning by using **Shell script**, **JProfiler**, **JConsole**, **Jmap**, etc.
- Wrote **awk** script to statistic Block generation performance.
- Enhanced booking trigger logic to speed up Block generation.
- Fixed issues of Titan paperless e-delivery migration process and refractor implementation.
- Implemented RRDPivotServer **RESTful** web service in **Spring** framework to provide e-delivery accounts to GS360 portal.
- Wrote integration test to generate documents used for migration process by using **JUnit**, **Mockito**.
- Done FourTenB report elimination and openFax elimination for US Titan instances.
- Fixed Titan RCC fax delivery issue, responsible for technical support.
- Involved in all phases of **SDLC** in agile environment and daily **Scrum** meeting.
- Used **JIRA** for bug tracking, used **Jenkins** for continuous integration, and used **SVN** for version control.

Environments:
Eclipse, JDK1.6-1.8, J2EE, Spring 4, Hibernate 3.5, JDBC, DB2, Sybase, Maven, SVN, JIRA, Linux.

Apple – Sunnyvale, CA | Oct. 2017 – Current
Full Stack Developer
IOS System and Product Engineer System
Worked on several Apple projects as a Full Stack software developer. Mainly focus on the one which enables Apple managers and carrier partners to sign documents, grant permissions and manage pre-launched devices online. Those work were done manually until this application released. Implementation include scheduled notification system, integration of E-Signature documents, internal/external software/hardware release access control and tracking system and etc.

- Responsible for requirements gathering, DB design, UI design, code implementation, code build, feature deployment and application maintain.
- Worked in **Agile** Scrum team as **scrum master** with daily standup and tri-weekly sprint meeting, manage the tickets use Apple **Agile** tool.
- Developed a Java/J2EE based Web Application with **Spring framework** and **Spring modules**.
- Implemented **Hibernate** as ORM and integrated to Spring.

- **Oracle 11g** is used as the persistence layer, configured and optimally indexed for searches.
- Developed the User Interface Screens for presentation using **React, Angular JS**, **HTML5** and **CSS3**.
- Implemented web based applications using the Model-View-Controller (**MVC** framework) with **Spring REST**.
- Used **TestNG** framework for unit testing of application and **Log4j** to capture the log that includes run-time exceptions, we have **Splunk** UI to tracking logs for UAT and Prod environments.
- Coordinated with various teams and raise support ticket for all issues (we need to coordinate with our **APS** and **SRE** team for data, **IDMS** team for user credential check, **E-Sign** team for e-signature maintenance), analyzed root cause and assisted in efficient resolution of all pre-production processes.
- Used **APC**(Apple Private Cloud) to deploy code change to make the code cloud accessible.
- Used **DS Client** to execute **RPC** (remote procedure call) between different APIs, which provided an easy way to get access to the services from another API.
- **RabbitMQ** is used for message Queue for **asynchronized** services, like email service and **e-signature** services.
- **Cassandra DB** is implemented as part of the new features as non-relational DB is famous nowadays.
- Java code is implemented with **Java 8**, using **Lambda expression** and **Stream API** for composing high performance code.
- **Spring IOC** is used for lightweight dependency injection(**DI**), which ensures automatic dependencies creation.
- Created **TestNG** testing case for backend Unit Testing, used **Mockito** for data mocking.
- Worked on **Data Encryption and Decryption** part for sensitive user data stored in DB within the application, which is part of the Apple InfoSec requirements, user sensitive data (PII) will be stored in DB with encrypted data, the encrypted property key is maintained by **Devops** team and will not share any where, also key rotation strategy is implemented here to ensure security.
- Updated the **Schedule Email Notification** for new software/hardware release, user accept or confirm packages, user perform **E-signature** and so on.
- Implemented more user-friendly Web UI with new components implemented using **React JS**, **HTML5** and **CSS3**, accessibility is considered for entire system.
- **Jasmine** and **Karma** is used for Front-end Unit Testing.
- **Bug-Reporter** and **Radar** are used as bug tracking and ticket management.
- Used **GIT5** as Version Control tool to perform codes Checking-in and Checking-out from the repository.

Techniques:

Java 7/8, Spring REST, Hibernate, Angular JS, React JS, RPC, Eclipse, Oracle DB 11g, Cassandra DB, git5, Junit, TestNG, Mockito, HTML5, CSS3, RabbitMQ, Jasmine, Karma, RESTful, APC, LESS, Agile, Radar, PostMan.

Google – Sunnyvale, CA | July. 2017 – Sept. 2017
Full Stack Developer
Compensation API and gThanks Application

Compensation API's main objectives is to provide a robust and secure authoritative source for planning, reading, and communicating Googler's compensation. It provides the access through programmatic APIs, enabling multiple different applications to avoid reimplementing the same business logic or security checks. gThanks is an innovative new Compensation Team Product which strives to foster a culture of recognition at Google by introducing a centralized and scalable platform for awards.

- Worked in **Agile** Scrum team with daily standup and bi-weekly sprint meeting.
- Focused on Compensation API **performance improvement**. Refactored structure of the code organization, designed the database schema to avoid duplicated defined fields and functionalities.
- Used **Stubby** to execute **RPC** (remote procedure call) between different APIs, which provided an easy

way to get access to the services from another API.

- Used **Protocol Buffer** to define message formats in a proto file, compiled them into serialized java class.
- **Spanner** is used for persisting data, defining the Spanner schema using Schema Definition Language(**SDL**) Reference, which can add **PROTO** (protocol buffer) and **Enum** directly in table structure.
- Java code is implemented with **Java 8**, using Lambda expression and Stream API for composing high performance code.
- **Guice** is used for lightweight dependency injection(**DI**), which ensures automatic dependencies creation.
- Created **JUnit** testing case for backend Unit Testing, used **Mockito** for data mocking.
- Worked on **Audit Logging** part for **gThanks** application, which is part of the **DPCI** (data protection core infrastructure), as the final step of requesting processing so supplemental information is available for the audit record (e.g., request success or failure).
- Built the application locally use Local Deploy with **blaze**, Web UI would be available through localhost.
- Updated the **mail** notification for peer bonus and spot bonus, fixed all the related broken unit testing and integration testing.
- Modified the Web UI with new components using JavaScript, HTML5 and CSS3, removed outdated mandatory fields and got all affected test passing.
- **Jasmine** and **Karma** is used for Front-end Unit Testing.
- **Buganizer** is used as bug tracking and ticket management.
- Used **GIT5** and **Piper** as Version Control tool to perform codes Checking-in and Checking-out from the repository.
- Composed changelists(**CLs**) in **citc** (client in the cloud) to make the code cloud accessible.

Techniques:

Java 8, Guice, Protocol buffer, Stubby RPC, Blaze, IntelliJ, Cider, Eclipse, Spanner, Unix, JavaScript, Angular, Angular Material, Soy, Piper, git5, Junit, Mockito, HTML5, CSS3, Hibernate, code search, Jasmine, Karma, RESTful, App Engine, Less.

Client: Jet.com **Aug. 2016 – October 2017**
Location: Hoboken, NJ
Project: Spectrum
Role: Frontend Developer

Description:

The Spectrum is an internal assortment view system, which provides views of all kinds of resources, also applying modifications to the resources. The agents are able to investigate and update any issues with SKUs. Involved in all phases of application design from coding through testing, integration and deployment. Applying the design across the application.

Responsibilities:

- Participated in the **Software Development** in **Agile** methodology and attended daily **Scrum** meeting.
- Used **Bootstrap** to create responsive web pages and optimized the website look by applying **SASS** as **CSS Pre-Processor.**
- Created **Angular2 Components, Directives, Services, Injectables** across the whole system, such as build the customized filter for different categories and every part of main assortment dashboard.
- Utilized **Angular2 Datatable package** to display large sets of the sku information data with functions such as selectable columns, pagination, sorting, reordering.
- Created custom and reusable **Modal** using **Angular2** features to display the sku details and editing functionalities.
- Heavily used **Angular2 Router** to navigate and passing data through the different status and sub status and multiple modals.
- Used **Angular2 Form** to do validation and implement authentication feature.

- Created customized **Angular2 Pipes** to display sku information and transform data in required format.
- Created customized reusable components such as **Loaders, Search, Image Editors,** etc.
- Extensively used **Rxjs Observables** and **Operators** to handle data from **HTTP** requests, using operators like **map**, **filter**, **and merge** to deal with the **Stream**.
- Created unit test cases of **Angular2** components, services using **Karma-test-runner** and **Jasmine,** end to end testing in **Angular** using **Protractor**.
- Implemented **Normalized.css** to enhance user experience **Cross Browser Compatibility**.
- Used **tsc** as compiler to compile typescript code into javascript, and also used **Typing**s to manage Typescript definition.
- Worked on **Webpack** as module bundler to build workflow for Web Application and **webpack-dev-server** as server.
- Used **Spring IoC** for **Dependency Injection** and handling request from Front End.
- Implemented **Spring Rest Controller** module for handling client request, **Hibernate ORM** for interacting with **MySQL Database** for fetching and saving data.
- Used **Github** for Version Control, **JIRA** for defect tracking, built the application on **Node.js** and used **NPM** to manage open source libraries.
- Worked extensively on **Jenkins** for **Continuous Integration(CI)** and for **End to End Automation** for all build and deployments.

Environment:

Angular2, Typescript, Ngrx, Redux, Rxjs, HTML/HTML5, CSS/CSS3, Bootstrap, SASS, Webpack, Karma, Jasmine, Protractor, Spring, Hibernate, MySQL, Github, WebStrom, JIRA, Jenkins, Scrum

---

Java/J2EE Developer

Coface North America – East Windsor, NJ Apr 2017 to Present

Project: CrediTrack (CTrack)

CrediTrack is an online insurance applications and policies management site for Sales, Brokers, Underwriters and Financial Analysts in the North America region. It is used for whole lifecycle of an insurance policy from initiating NBI draft, negotiating coverage liability and premium between agent/broker and underwriter, to submit/track payments for premium, to analyze gross loss ratio and etc.

- Major responsibilities are maintaining CTrack application by refactoring existing structure and code, discussing requirements and technical design, identifying or fixing issues in an **Agile** development setting. Weekly meeting guaranteed and small meeting on demand.
- Always attempted to do more than the basic requirements and deliver beyond expectation results to end users. As well, kept the coded function quality, robust, decoupled.
- Experienced in working with HQ team in Paris, France on infrastructure level networking and database.
- Developed and improved web pages using **JQuery, Ajax, JSTL**; Made them interactive, intuitive, and enjoyable to use. Utilized registerHelper, compile, precompile etc. functions in **Mustache** based **Handlebars.js** to dynamically generate HTML page and improve the structure of the application.
- Wrote interactive **HTML/CSS** assets for easy requirement communications between developers and business analysts; Made requirements defining phase quicker.
- Involved in the development of project back-end layer by most using most of the **Core Java** features such as **Collection**, **Interfaces**, **Exception Handling**.
- Implemented Oracle **Functions**, **Store Procedures**, **SQLs** mainly for requirements involved with Database access and changes. Kept performance in mind to ensure they are usable, fast and smooth.
- Utilized **Synchronized JavaMail** SMTP function with **Cron** schedule to send emails to agents with attachments when invoices are generated.
- Organized and contributed in **code review** sessions when implementations are ready to be promoted.
- Experienced in **Jenkins** continuous integration tool to build and develop automatically from **GIT**.

Environment: *Java, J2EE, JavaScript, JQuery, Handlebar.js, HTML, CSS, AJAX, Native JSP/Servlet, JSTL, Oracle DB,*

Store Procedure, Oracle Functions, JDBC, Apache POI, REST, NetBeans, Fiddler2, Agile, Servlet, Ant, JavaMail, Tomcat, Git, DiffMerge, WinSCP, Sublime

---

Java/J2EE Developer
Samsung SDS America, Inc – Ridgefield Park, NJ Aug 2014 to Mar 2017
I was a Java/J2EE Developer at Samsung SDSA from Aug 2014 to Mar 2017. I was part of the Samsung.com group from Aug 2014 to Feb 2015, then Resource Management Support group from Feb 2015 onwards.

Resource Management Support Group
Under the Resource Management Support Group, I worked on 3 different projects simultaneously:

Project 1. Resource Management System (RMS)
RMS is an online budget requesting and tracking system with thousands of active users in USA, Canada and Mexico. Samsung employees use RMS to request for reimbursement for travel, event, tuition and office expenses. The system is well-maintained so that it is extended to cover not just expense-related, but also non-expense request such as asking for promotion, or even organization hierarchical changes

- Utilized workflow based **Zasati** engine to create new request template, and modified or merged existing template with integration of Samsung **mySingle approval module (SOAP)** and Samsung **Knox Portal**
- Monitored and modified mySingle status updating **Spring Batch** that sync up approval progress for on-going requests
- Imported employees' company card transactions and travel expense data from **SAP** to RMS database daily, monitored batch server progress by checking server logs and oracle database
- Supported request tickets directly from end users through IT service desk system. Analyzed and solved issues on timely manner, actively reported ongoing issues to management
- Implemented and monitored scheduled **Java mail** with **Velocity VM** template in different contexts, including System Admin received exception email, General Admin received organization change notification, manager received over budget warning notification, employee received course opening reminder email etc.
- Implemented PDF download functions using **iText** with **JSP** templates
- Implemented user friendly list using **jQuery Grid**, provided sorting, search and pagination functions
- Received system exception emails and fixed the issue seamlessly without affecting user experience
- Implemented and modified path resolvers in Java, that dynamically generate approval path based on factors such as user's organization, request type, request amount and pre or post approval request and etc. The approval path follows a predefined template from **Oracle DB**
- Implemented hierarchical organization tree with autocompleted users, departments and cost center searches for administrator using **Ajax** and **JQuery Simple Tree**
- Regularly deployed the changes into production server using Remote Desktop and ODEN (deploy tool)

Project 2. TEMS(Telecom Expense Management System)
TEMS is a mobile devices managing system for Samsung employees to request for devices, and for admin to keep track of which device is distributed to whom, and the device's monthly usage and expenses.

- Extensive experience in rendering user data in complex pie, line bar charts using **Highcharts** and **AJAX**
- Implemented web based Excel upload function for administrators to read new device info up to 10,000 lines, into **Oracle DB** and Excel download function that generates customized user usage data (by month, by carrier type, by device type) using **Apache POI Workbook**

- Implemented batch job that back up device assets lines at 1st day every month using **Store Procedure** and **Scheduler** using SQL Developer

Project 3. SMS (Sample Management System)
SMS, similar to TEMS, keeps track of new Samsung mobile devices that are sent to testing facilities. I worked extensively on an enhancement project that re-faced (fixed existing bugs, add new functions, beautify webpage) over 20 pages.

- Extensively used **tablesorter** to render sample data, provided excel-like header search, sorting function with the use of **JQuery** and **AJAX**
- Provided pagination that allows user to customize number of lines to display on page and navigate through all the data, using **SQL** in **iBatis** XML
- Implemented multiple edit function, allowing users to edit multiple selected sample lines at the same time using **jQuery** and **AJAX**
- Created and monitored Oracle **materialized view** and **synonym** to retrieve global HR data into application's own database

Samsung.com Group
Project: Samsung.com Support 5.0
Samsung.com and its Content Management System (CMS) is designed to help Samsung marketing team manage Samsung.com website seamlessly by allowing instant, real-time updates. Users can better allocate resources and utilize sales and marketing dollars. CMS provides simple, interactive tools that help business user manage products, categories and pictures more efficiently. CMS enables the marketing team to spend less time managing information and more time on consumer shopping needs.

- Mainly worked on the support landing page (**www.samsung.com/support**), contact page (**www.samsung.com/us/support/contact/**) and Q&A page (**http://www.samsung.com/us/support/customer-shopping-faqs/**)
- Fixed open issues on Shop pages(**https://shop.samsung.com/in/**)
- Converted user defined wireframe documents to detailed technical requirement and planned schedule for development
- Experience in **SQL tuning** and **performance optimization**, using **Index**, **Partition, Materialized View**
- Solid understanding of database structure for multi-layer, categories based products
- Implemented and modified oracle **Store Procedure** for loading new products designed in Korea HQ
- Extensive use of **Java Collections (List, Set and Map)** to store temporary retrieved data from databases
- Attended daily **scrum** meeting with development group and weekly/bi-weekly progress meeting with clients from Samsung Electronics America

Web Security Trainings:
Attended quarterly company-wide web security sessions and passed exams on web application security. Topics include **Cross-site Scripting**, **JavaScript Injection**, **SQL Injection, Spring Security** and etc.

Environment: *Java, J2EE, JavaScript, JQuery UI, HTML5, CSS3, AJAX, Bootstrap, Spring, Oracle DB, Store Procedure, Scheduler, iBatis, Velocity,, SQL, JDBC, JNDI, Apache POI, REST, Eclipse, Fiddler4, JUnit, Agile, JSP, JSTL, Servlet, Maven, JavaMail, Tomcat, WebLogic, WinSCP, BeyondCompare*

Client: Morgan Stanley (New York) **Oct 2018– Present**

Role: **Full Stack Java Developer**
Projects: The project deals with upgrading the existing application and modifying defects of Wealth Management system which helps Morgan Stanley financial advisors to simplify the process of wire fraud report and wire post-payment exception track.

Responsibilities:

- Followed **Scrum Agile** methodology including iterative application development, Sprint every two weeks, daily meeting.
- Used Observable, Subscribe, Subject and operators like map, filter from **RxJS** in HTTP calls to manage asynchronous data and preprocess data.
- Upgraded the HttpModule to HttpClientModule in Angular service to allow middleware logic to be inserted into the pipeline.
- Building application using the **NgRx** library including state, actions, reducer, selectors to enforce unidirectional data flow.
- Maintained Cross Browser compatibility and implemented Responsive Design using **Bootstrap** grid system, custom media queries**, Angular Material**, **HTML 5** and **CSS3**.
- Combined **Reactive form** default validation and implemented onkey function to implement input filed real-time monitoring.
- Strengthened front-end unit test using **Jasmine**.
- Performed task runner **Karma** for integration test and **Protractor** for end-to-end test.
- Implement the service layer based on Spring container and **used Spring IOC** features for bean management.
- Used Java **Stream** API and **Lambda expression** to operate the java collections.
- Enabled session factory level cache by **Eh-cache** in Hibernate to accelerate CRUD operations.
- Created **RESTful** API for generating wire fraud report
- Used **Spring AOP** with @After annotation to add Logger for Logging.
- Added unit test cases using **Junit**.
- **Log4j** and **JIRA** are used for logging and defect tracking.
- Performed **TeamCity, Git** for continuous integration (CI), build tools, and version control.

Environment:
Scrum, Agile, HTML 5, CSS3, Bootstrap 4.0, Angular material, Angular 5, JDK 1.8, Spring, TeamCity, Jasmine, Karma, Protractor, Junit, Git, Jira, Log4j

---

Gordon Food Service, Inc -- Wyoming, Michigan **September 2016 – October 2017**
Java Developer

Project 2: Event Tech
Event tech project supports food shows and related events to finish the data integration and complete certification process with data loading. Built a MPV project to support PMA team to upload the promotion into database during post-event.

- Created batch file extract for iManagement, populate date from database by using **PL/SQL** query.
- Created the MVP for eventTechPma from scratch, both front- and back-ends. Implemented server side with Java and Spring. Applied Spring Security for user authentication and API authorization.
- Involved in the **requirement analysis** and **designed** the system to incorporate those requirements and coming up with the design to utilize existing services.
- Used **ui-router** to make the project a **single page application(SPA)** with nested views

- Established business logic and encapsulated it into **Angular** Service layer, injected **$http** and **$q** to do
- request and return a promise, packaged data into view using AngularJS data binding.
- Applied **Angular form validations** to entries like username, password, email etc.
- Implemented **Data-filter** in **AngularJS** to support data searchable and sortable.
- Developed Controller for request, response paradigm by **Spring Controllers** using **Spring-MVC**.
- Implemented the Project structure based on **Spring MVC** pattern using **spring boot.**
- Involved in the front-end design and decoration by using **HTML5**, **CSS3** and **Bootstrap.**
- Responsible for creating and maintaining architecture for **Restful API** using Spring Boot.
- Used **Bitbucket** for version control management.
- Developing **Junit** test cases using **Mockito** framework.
- Part of **SCRUM** meeting to discuss daily tasks.
- Added **CSRF** protection to application that uses **Spring Security**.
- Used **karma-test-runner** and **Jasmine** for JavaScript Unit testing.

Environment:
Java 1.7, J2EE, JSP, Servlets, Spring MVC, SQL/PLSQL, Web Services, WebLogic 12.c, Oracle 11g, HTML, CSS, JavaScript, JQuery, Jasmine, Karma-test-runner, Gulp, Git, Windows 7, PL/ SQL, SQL, Windows.

Project 1: Evergreen
Evergreen is a project that helps the team to migrate the EDI Ordering project, Invoice-EDISubscriber and CreditEDISubsciber to Weblogic 12. Converted projects to standard maven structure.

- Worked in the **SDLC**(Software Development Life Cycle) for EDI applications, development with **Agile** methodology and daily standup meeting.
- Used **Tibco EMS** to create pending queue for credit server and bridge to the topic, finished **JMS/EMS** queue configuration.
- Completed the integration enhancements on batch job and used **WCC** workload automation tools for scheduling, monitoring.
- Updated **Wily** dashboard for Invoice & Credit application on **Weblogic 12c** and implement work managers in **Weblogic** configuration.
- Applied **SOAP** Service for interacting with Java components to retrieve data from database or third part services.
- Refactored Invoice & Credit application into standard **Maven** project structure in order to deploy to **Weblogic** 12c.
- Used **Antics** and **Maven** for deploy and release work in production environment and support the production issues.
- Used **Spring AOP** to add retry definition for back-end service and configured beans using **Spring IoC** for dependency injection.
- Upgraded **Spring** and **Hibernate** to the latest version and resolved conflicts reported by **WLS-CAT**, encrypted property values by using **Spring-Util** and **Maven**.
- Developed scripts for build, deployment by using **Jenkins** and fixed blocker issues and javadoc errors reported by **SonarQube**.
- Coordinating and tracking all projects for seamless releases using Project Management System **JIRA**, Source Code Management System **SVN**.
- Implemented **Log4j** for logging and appended to a log file for debugging.

Environment:
Java 1.6, J2EE, JavaMail, Spring 4, Spring boot, Spring AOP, Log4j, Hibernate4, Oracle, JUnit, SOAP Web Service,

Weblogic 12c, TIBCO GEMs.

J.P Morgan Chase&Co| Front End Developer**,** Big Data Engineer New York, NY
Industry: Financial Service Sept.2017 - Present
Descriptions: J.P Morgan Global Technology Infrastructure Big Data Team provides a managed Hadoop Framework for dynamically scalable and cost-effective solution to process vast amounts of data. The fully managed service enables a broad range of scenarios such as Data Analytics, Machine Learning, ETL, Data Warehousing and more. In this project, I am responsible for providing Big Data solutions and support for internal business lines on shared and designated clusters. I also participated in designing and implementing user interface of internal Big Data applications.

Responsibilities:

- Implemented Hadoop services (HDFS Directory structure, Hive database, Hbase namespace, Yarn and Impala queue) based on finalized requirements
- Attended and coordinated review meetings with tenants to gather requirements and provide cloud-based solutions from a platform security and operations perspective
- Validate configuration results and handover to tenants. Provide support on further big data related issues
- Designed and created UI wireframes using Balsamiq and draw.io for initial onboarding automation application
- Implemented Hadoop components configuration user interface which takes requirements data input using Angular4, TypeScript, HTML and CSS
- Created responsive web design using Moneta Bootstrap and optimized CSS using SASS
- Consumed RESTful web services based on Angular Http module and Observables to resolve data transaction between front-end and back-end and perform CRUD operations on the clusters
- Applied Angular Template-driven Forms to build user log in page. Used form validation to validate input user ID and password
- Used multiple Angular Directives such as ngIf, ngFor to control and display data output on Hadoop components configuration page
- Implemented Routing service for navigation between UI pages
- Used Jasmine for front-end unit testing, Git for version control and Webpack as bundler to manage modules and libraries
- Used Jira to track project progress and issues in Agile software development methodology
- Implemented OS version editing feature for internal cluster monitoring app using Angular.JS, JavaScript, HTML, CSS, Bootstrap
- Integrated REST API with new feature to update and retrieve cluster data

Environments: Angular4/5, TypeScript, Angular.js, REST API, CSS3, SASS, HTML5, Webpack, Jasmine, Single Page Application, VS Code, Hadoop, Hive, Hbase, Yarn, Impala, Spark, JSON, ES6, Balsamiq

---

Client: Publicis Media - New York, NY **November 2018 - Present**
Project: Taxonomy Configuration Management
Role: Frontend Web Developer

Description: Publicis Media is one of the solutions hubs of Publicis Groupe and aims at creating value for clients through global media agency brands and scaled capabilities across investment, strategy, performance

marketing and content. This project is to build a one-stop centralized information portal to manage clients' taxonomy information from multiple departments and facilities, and generate reports based on needs of clients and agencies.

Responsibilities:

- Frontend:
  - Developed front-end web presentation using **HTML, CSS 3** and **React-Bootstrap** in accordance with the design given by a UX designer in the team.
  - Implemented a single-page web application using **ReactJS** and **React-Router** backed by .Net Core
  - Extensively used **Axios** to actively communicate with backend and handle data flow using actions and reducers from **Redux.**
  - Coordinated with back-end developers within the team and utilized **Postman** and **SQLSever** to conduct end-to-end testing for features.
  - Customized elements and components from **Carbon-Component** to support the functionality and usability of the application.
  - Displayed data received in stream from backend to table format with **React-table** and customized in-grid filtering and out-of-grid searching functionalities.
  - Construct accessible status flow using **React-Beautiful-DnD** allowing user to sort lists of content vertically and horizontally among multiple containers.
- Other:
  - Illustrated technical feasibility of features during discussion with Product Owner, stakeholders and UI/UX Designers about business requirements for the application.
  - Customized **Microservices** architecture with **REST APIs** to support data fetching for downstream application.
  - Worked in the **Agile SDLC** (Software Development Life Cycle) and participated in meetings following the guideline of Agile methodology.

Environment: ReactJS, Redux, HTML 5, CSS 3/SCSS, React-Bootstrap, JavaScript ES6, Microservices, REST APIs, Carbon-Component, React-Router, React-Table, React-Beautiful-DnD, Axios, Postman, SQLSever.

Client: Oppenheimer Funds – New York City, NY **May 2018 – October 2018**
Project: Portfolio Management Tool
Role: Full Stack Web Developer

Description: Oppenheimer Funds is an investment and asset manager. This project is to build an application for asset management of various securities (shares, bonds, equity, etc.) to better assist portfolio managers and traders to put together management strategy choices, asset allocation, investing strategies and management of risk with complete visualized trading data. It is built with a tech stack of frontend in React and backend in Spring Boot.

Responsibilities:

- Frontend:
  - Worked extensively on **ReactJS**, **HTML**, **CSS/SCSS** and **React-Bootstrap** to develop rich styling web pages.
  - Built the single page application and managed the overall states with **Redux reducers, actions** and **React lifecycle hooks**.
  - Developed forms and input boxes with **redux-form** to control the form value and validate user input.

- Consumed **RESTful API** with **AJAX** requests and **query string** by sending the promise-based http request from client side to server side with **JSON** format data.
- Displayed data which is received in stream from backend to table format with **Ag-Grid** library and customized sorting and filtering functionalities.
- Realized functionality to export data to excel using **react-excel-workbook NPM package**.
- Applied **Unit Test** with **Jasmine** and **Karma, Feature Test** with **Cypress.**

- Backend:
  - Configured and externalized application's configuration properties for **Java** and **Java EE**,
  - Implemented REST Microservices using with **Spring Boot** and **Java.**
  - Constructed new endpoint for manual **cache** resetting, involved in refactoring caching strategy of the application.
  - Refactored **service** including database structure to retrieve and store users' entitlement and preference data, created scripts for seeding data to **mySQL** database.
  - Adopted **Oauth 2.0 authentication protocol** together with **Spring Security** for user authorization and authentication.
  - Utilized **JavaMail** to enable functionality of sending email to application support team directly from the webpage.
  - Integrated with **JMS** topic and message queue in order to receive notification about data update from upstream and other data source.
  - Applied **Unit Test** using **Junit** with **Mockito** and end-to-end **Function Test** with **Selenium.**
- Other:
  - Followed **Pair Programming** and **TDD** to manage the development of the project. Participated in reviewing business requirements and technical requirement documents with Product Manager on regular basis.
  - Used **Git** as version control system and **Pivotal Tracker** as project management tool for tracking changes and progress in the project.
  - Utilized **Jenkins** and **Docker** for continuous integration and deployment for Microservices.

Environment: ReactJS, HTML 5, CSS 3/SCSS, React-Bootstrap, JavaScript ES6, Ag-Grid, Java 8, Java EE 7, Spring 4.0, Spring Boot, Spring Security, Microservices, mySQL, OAuth 2.0, Karma 1.2, Jasmine 2.5, Junit 5, Mockito 2.3, Cypress, ixp cache, JavaMail, Selenium

---

Client: US Foods - Rosemont, Illinois
Role: Full-Stack Developer
Project: Enterprise Systems Dashboard (ESD) **2018.4 – 2019.2**
This project is to build a website from scratch to highlight the critical incidents so that it helps US Foods's Help Desk team to track Critical Services Incidents(CSI) reported which are captured on HP Service Manager(HPSM) software.

Responsibilities:

- Participated in the **Software Development Life Cycle** (**SDLC**) including planning, design, building, testing and deployment in **Agile environment**.
- Worked on the **data modeling** in designing the ER Diagram and wrote the DDL scripts for the DB.
- Adopted **Redux**(ngrx) to store and manipulate the information data of locations/services in the front end thus reducing the frequency of API calls.
- Built **customized tables** to satisfy the requirements of sorting and filtering for different criteria, wrote customized **pipe** to convert different data formats.

- Applied **reactive forms** to create an event/incident/comments, edit details of events and used **regular expression** in form validations to get the expected inputs from the users.
- Customize the **primeNG** component of calendar to overcome the limit of the third-party library.
- Applied **Google Map** to visualize the most updated information for services impacted in the whole united states.
- Used **ng2-idle-keepalive** to implement the active session in the front end, which expires in 5 minutes without any user active interaction and makes API call to refresh the token every 1 hour. Wrote the Nodejs API in backend to call the security API to get the token.
- Adopted **squel.js** as the query string builder and **express.js** to implement the middleware part of the project.
- **Optimize query** for searching events from 4 seconds to 0.5 seconds using group concatenating.
- Designed the Nodejs API, wrote **Swagger files** for API and shared with the teammate who is working on the UI side for data mapping.
- Involved in designing the whole architecture on the Nodejs side, each API call from the frontend is neatly stored in corresponding directory and can be called easily.
- Wrote unit test cases for both Angular and Node with **Mocha** and **Karma**, ensuring the code coverage more than 70% using **Istanbul** and **Angular cli**.
- Worked with DevOps team on the chef scripts on **Bamboo** for CI/CD and solved the deployment issues for both Angular in **S3** and API in **EC2**, worked with DB team to deploy DB in **RDS**.
- Managed the project using **Gitlab**, involved in designing the source structure of Gitlab, merging codes from teammates, doing code review. Used **husky** to prevent bad commit and push.
- Cooperate with the business team to get the requirements, UX team to tackle the styling issues and discuss the user interface design, participated in the daily scrum meeting with client and offshore teams, made minutes for the meetings and updated the team.
- Gave demo to product manager at the end of each sprint, collected feedbacks from her and participated in the retrospective meeting to discuss the status of whole project.
- Worked with QA to discuss and solve the bugs and defects.

Environment: HTML5, CSS3, primeNG/PrimeFaces, Typescript, Angular 4, Redux, Nodejs, Karma, GitLab, Bamboo, JIRA, AWS S3, AWS EC2, AWS RDS, MySQL

Client: Verizon Location Service, Denver, CO *April 2018—April 2019*
Position: Full Stack (Angular/Node.js) Developer
Project: mapquest.com website

Mapquest.com is a free online web application owned by Verizon Media. I am in the Web Core team as a full stack developer. My main role is to work with the UI team to improve the UI of the website, work with the marketing team to track the proficient generator features, maintain the code base including updating to the lasted version of Angular and the dependencies and refactor functions, and do some code refactors.

Responsibilities:

- Involved in the single search feature development and release (which is the biggest change in the last couple of years). Built the new version application with **AngularJs**, **RequireJs, Leaflet** and **CSS/LESS**.
- Worked with the marketing team to track user actions and page reviews with **Google Analytics**. Set feature toggles with new feature to allow A/B testing.
- Followed instructions of the official Angular Doc and updated the application from **AngularJS 1.7** to **Angular 7**.

- Rewrote the modules of directives into **components** and reorganized the application with **Angular7**.
- Applied the **DragDropModule** into our direction page to allow user to move the item to reorder the routes.
- Wrote custom **Pipes** for our new UI to allow users to filter the search results and used **reactive form** to do the form validation and form data processing.
- Built libraries with new cli command **ng g library** and made them as **npm** packages.
- Worked on different versions of UI for **mobile app**. Implemented mobile-specific features including mobile drawer with **hammer.js.**
- Implemented **Fontcustom** to create fonts for icons for the application. Wrote custom Bootstrap styles. Applied **UI bootstrap** to a few components.
- Produced histogram using **D3.js** to display popular times of the restaurants and stores.
- Updated the privacy rules based on the GDPR. Worked with **MySQL** to remove user information in Europe.
- Used **Nginx** reverse proxies to prevent CORS errors and allow the app run smoothly locally.
- Configured the MapQuest admin tool and managed the assets according to the contracts with the clients.
- Worked with modules for database persistence using **Node.js** to interact with **MySQL.**
- Used **Express.js** for development of RESTful web services and middleware configurations.
- Implemented unit test and End 2 End test with **Jasmine**, **Karma** and **Protractor**.
- Participated in the **AWS** building work with **EC2** and **S3**. Created automated build with **Jenkins**.
- Applied **GitFlow** to collaborate better among a team with more than 6 developers work on the same application.

Environment:
AngularJs 1.7, JavaScript ES6, HTML5, Angular 7, TypeScript, RequireJS, Leaflet, CSS3/LESS, WebJars, Play Framework, JSLint, Nginx, Google Analytics, Gitflow, MySQL, Node.js, Amazon Web Services

---

Client: Goldman Sachs, NJ **Nov 2015 - Present**
Role: Sr. Java/J2EE Developer
Project: Middle Office Trade Processing Systems

Middle Office has the main responsibilities of trade/confirm generation, trade processing, trade figuration, and trade booking. The project involves multiple business products which includes but not limited to Equities, Fixed Income and Futures. I worked on major application group in Security Middle Office including Hydra, Kirin, Titan and NewT. We also have internal tools for test automation message rendering and delivery to different down streams. The testing environment can generate regression test comparison results. The project will support business globally across 24/7 trading environment with high reliability, real time and scalability processing.

Responsibilities:

- Worked in a fast paced **Agile** development environment and made contributions to bi-weekly releases following **SDLC**
- Worked with core application Hydra maintenance for **trade booking**, **confirmation** and **capture** by using **Core Java, SQL, Stored Procedure, Shell** and internal testing frameworks
- Setup **JMS** configuration to receive **Binary** daily trades and send acknowledgement back to front office
- Enhanced system to support 20 additional input files and performed **Capacity Test to ensure services load will not run out of memory**
- Designed application to process large files based on **iterator pattern** and **filtered** the results to reduce

volume to downstream systems
- Implemented function to support trade **aggregation** which involved **Java Reflection, Predicate** and **Singleton**
- Implemented **Drool** rules to enrich trade specifications and also add function to **Drool's UI** to support front office by using **Spring REST**
- Used **SQL** for data analysis, retrieval data and compare report result in **Aqua Data Studio 7**
- Wrote **SQL** statement to **aggregate** some unique transactions to one big transaction
- Wrote **integration testing** to keep specific client **Account Balance Type** as **HIC** by using internal testing framework **Alpaca**
- **Refactored Java classes** to prevent difficulty of feature enhancement
- Applied **HTTP** Service and **REST WEB Service** to connect the front-end **AngularJS** to Back-end **Spring** framework with **JSON** format
- Applied **Data-Filter** in **AngularJS** to support data searchable and sortable
- Develop GS testing environment page by using **HTML5, CSS3**, use Dash UI to implement grid layout, responsive web page
- Wrote **regression test** cases for new feature and trading flow to make sure there is no impact on released versions
- Used **Mockito** test cases to mock different scenarios trades for each service
- Designed and debugged **Autosys** jobs to trigger some process such as automatically sending paper, fax or email confirmation over midnight
- Monitor production data processing against QA environment and responsible for exception handling during **RTF** running under **Unix**
- Global **UAT** middle office support for North, America, South America, Europe and Asia regions
- Setting up QA environments to test different client requirement and **connectivity tests** for major projects trading flow
- Cooperated with **front office** and **back office** teams to satisfy their requirements or resolve urgent production issues such as, missing commission or enriching trade from Hydra
- Develop **automation regression testing tools** for **NewT** and **Hydra** by using internal testing framework **Zebra** which is written by **Java** and **Shell** script
- Generate daily trade report to analyze potential risks for **production migration**
- Used **JIRA** for bug tracking and project management, **SVN** for version control and **WinSCP** for **FTP** file management

Environment: JDK 1.7, JDK 1.8, J2EE, JMS, Mockito, Junit, Spring 3.2, Hibernate 4.3.5.Final, JNDI, DAO, MVC, Unix, Maven 3, Sybase IQ, DB2, SVN, AngularJS, Jenkins, JIRA, WinSCP, Autosys

---

Client: **BBVA Compass** Feb 2017 – March 2018
Role: Senior Web Developer
Project: Automik
This project is a portal through car dealer webpage to bank loan system. My responsibility is to design and apply entire OOP functionally component for the system in order to reused the application for different languages and countries. By using the bbva-cellsjs and polymer.js to custom the BBVA own component plugin.

Responsibilities:
- Participated in various phases of **Software Development Life Cycle** including Design and Implementation and Maintenance in **Agile** mythology.
- Designed and maintained front-end page layout using **HTML/HTML5**, **CSS/CSS3**, **Bootstrap, Polymer1.0**.

Polymer-cli, cells.js, cells-cli for the project based framework.

- Used **Cells.js** work with **Polymer1.0** framework to implement code **OOP.**
- Applied **Polymer TPL** such as **iron** and **paper**, used **iron-validator-behavior** for Form Validation.
- Provide Polymer **data flow concept** to implement the communication between components. Used pass method such as **one-way binding**, **two-way binding**.
- Created **Responsive Web Designs** using **Bootstrap4.0** grid system and **CSS3 Media Queries**.
- Created Custom element to extend the function about standard **DOM element**. **Configured** attribute and properties, manage isolate DOM by using **shadow DOM.**
- Supported **Single page Application** by using **Polymer and Cells** .
- Utilized **PolymerJS** to achieve the DOM manipulation like **dom-if, dom-repeat, dom-template.**
- Used **Polymer Iron-ajax Call** via **JSON** format.
- Created **Polymer SVG** for animation and achieve component reusable, also utilize the **piper** to make the function filter.
- Used **normalize.css** to let the web pages have compatibility of different browsers like chrome, Firefox and Internet Explorer.
- Provided interaction support with **Restful** service.
- Supported to inject and mock Angular service into unit tests by using **cells testing tool**.
- Utilized **bower** to do the dependency.
- Developed server-side APIs under **Node.js** runtime environment with **JavaScript** as back-end server.
- Involved in various testing methods such as **Unit testing, Integration testing and regression testing**.
- Used **Git** to fulfill the revision control, **Jira** to do issue tracking and project management function, **Jenkins** as the continuous integration platform.

Environment:

HTML/HTML5, CSS/CSS3, Bootstrap4, JavaScript, PolymerJS, Polymer-cli, Cells.js, bbva-cells-components, Node.js, Jasmine, Karma, Grunt, Git, Shell Scripting, Linux, Jenkins, VSC

Client: Goldman Sachs, Jersey City, NJ **Aug 2015 – June 2016**

Role: Senior Java/J2EE Developer

Project: GSEC-GSCO Entity Migration

This project focuses on merging the GSEC-US clearing business into the legal entity GSCO. The main work is to migrate some of the accounts and entities information from GSEC platform to GSCO platform. And also do the clearance and settlements for GSEC and GSCO. The main work involves creating new environments, data sourcing, report logic changing and UI changing. Also need to help fix bugs of existing reports and help with regression tests.

Responsibilities:

- Involved in working with **Sybase IQ and DB2** database using **SQL** to retrieval data, checking program process and comparing report results in **Aqua Data Studio**. Used **JNDI** for naming and Directory Services.
- Involved in the development of various **Java beans** with Builder Design Pattern and reference data module using **Core Java collections**.
- Designed and debugged **Autosys** jobs to run some process such as daily data sourcing or monthly report regression automatically.
- Setup and maintained report running environment and data sourcing environment by using company self-designed framework **Ramp** and **Cobra** and involved in solving **multithreading** problems.
- Involved in the development of Adjustment framework by using **AngularJS** to make it possible to adjust exceptions in front-end.

- Tracked report running process to retrieval unusual data using **Java**, **Hibernate Framework** and meet with **offshore** team to fix bugs together.
- Wrote **shell scripts** to batch trigger reports, run Java program, and transfer data in database.
- Patched project, tracked breaks, checked deployment with **Linux** Commands.
- Wrote **Junit, fitnesse** tests to test Java functions and wrote Java program to implement the **regression test** to do the **reconciliation** of reports result and do **exception handling**.
- Refactored Java classes in report modules to fix **Sonar** violations and maintain code quality.
- Used **Java Mail** technology to generate and send alarm or results emails.
- Involved in the **software development life cycle (SDLC)** of the project from user requirement gathering, business logic analysis and design to integration testing.
- Followed **Scrum** approach for the development process.
- Code review by using **CodeStream** to increase code quantity and track changes in main modules.
- Conducted integration testing for the application reproduced issues and reported with **JIRA**, trigger continuous build and integration build with **Jenkins**.
- Used **SVN** for version control and **WinSCP** for FTP file management.

Environment:
JDK 1.7, JDK 1.8, Eclipse IDE, Hibernate, JNDI, Java Mail, SAX, Sybase IQ, DB2, SVN, Linux, Jenkins, Tomcat, WinSCP, Junit, fitnesse test, JIRA, Sonar, Autosys.

Ultimate Software – Atlanta, GA **Jul 2017 – Jul 2018**
Software Engineer (Full Stack/MEAN Stack developer)
Ultimate Software produces software related to Human Resource Management and Payroll Services. This project is to implement a Data Migration Service (DMS) to migrate data from the old recruiting system to the new one. This service will free 75 mins per customer from manual input. This is a consumer facing role and my main responsibilities were to maintain and implement new feature for DMS.

- Implemented Single Page Application with **AngularJS** and **Jade**, and explored MVC features by developing custom directives, controller, and service
- Applied Bootstrap components include navigation bar, pagination and grid system to create responsive pages
- Created robust RESTful APIs using **Node.js** and **Express** based on Microservices architecture
- Improved the performance of large scale data extraction by optimizing queries in both relational database (**PostgreSQL**) and non-relational database (**MongoDB**)
- Implemented Data Access Layer to connect to MongoDB and defined data model using Mongoose
- Integrated both frontend and backend 3rd party libraries such as lodash, agenda, fs(file-system), path, ssh2, p7zip, async-retry, etc
- Built CI/CD pipelines by extensively using software delivery tools and platforms like **Docker**, **Pivotal Cloud Foundry**, and **TeamCity**
- Created unit tests based on **Mocha** framework and use **Sinon** to stub and spy on functions and services.
- Maintained a high standard of code quality by achieving 100% of unit test code coverage with Mocha, Karma and Grunt, and performing code reviews in BitBucket
- Followed Agile methodology, utilized Agile tools and participate in daily standup and weekly sitdown meetings.
- Worked closely with end clients to demo in UAT sessions, troubleshoot high priority defects and

applied hotfix to ensure smooth customer experience
- **Git** was used to fulfill version control and **JIRA** was used as ticket tracking system

Technologies:
Jade, CSS/CSS3, Bootstrap, JavaScript, AngularJS, Typescript, Ajax, MongoDB, Mongoose, JSON, Node.js, ExpressJS, Mocha, Sinon, Grunt, Karma, Docker, Pivotal Cloud Foundry, TeamCity

---

- **Company: Educational Testing Service – Ewing, NJ**
  May 2015 – Feb 2017
- **Role: Java Full Stack Developer**
- **Project: eReg**

The eReg is a test registration and management web application. It allows test-takers to schedule/makeup tests like GRE and HiSET, purchase test preparation materials and manage their own test accounts. It also used by administrators with different permissions to manage the test information, test-takers' profile, accommodations, and other services. The specific administrators can also manage test-center data.

Responsibilities:

- Worked on scheduling/makeup web-flow to provide steps for test-takers to complete their test registration.
- Followed Axure style guide and discussion with Business Analyst (BA) to create and update web UI for each step.
- Created a Single-page application (SPA) module with Angularjs for test-taker profile service.
- Implemented Spring-MVC based services in Controller and Service layers. Wrote SQL queries and created DAO layer services with ORM, Ehcache to access Oracle database.
- Worked with BA to update design/spec doc for each service module as required.
- Fixed reported issues, created unit/integration test and automation test scenarios for eReg web-app with Sahi.
- Worked on load testing with SoapUI and optimized the performance of each step.
- Worked on web service API and notification module to notify test-takers through email and SMS.
- Worked with remote team on DTO model design for data transfer through web service. Implemented RESTful web service sync/async APIs for accessing third-party web services and fetching test data from remote data servers.
- Created email service library, SMS service library with Clickatell API and various templates for different notifications
- Worked on cronjob to load test data from remote data servers and update database at specified time.
- Created retry service to process notification failure.
- Worked on test-loading application module for admin to manually update test-center data in database from a excel sheet.
- Created a SPA module with kendoUI for file uploading and used Spring Security for authentication.
- Created a data processing library with Apache POI for data loading/cleaning/validation and updating the data in database under customized rules.
- Tracked and analyzed issues via Splunk logging and fixed issues reported by client in Production environment.

Environment:

Java 7, Spring 3x, Hibernate 4x/JDBC, Spring Framework, Spring Security, Spring Batch, JavaScript, HTML5, CSS3, Angularjs, jQuery, Kendo UI/mvvm, JSP, JSTL, JAX-RS/resteasy, Eclipse, SVN, jrebel/xrebel, Oracle 11g, Apache Tomcat 7.0, Maven, Sonar, JIRA/Silk Central, joda, Clickatell, Apache POI, MelissaData, SL4J, Splunk, FreeMarker, SqlDeveloper, SoapUI

McGraw-Hill Education – New York, NY **January 2018 – Feb 2019**
UI Developer

Project Description:

McGraw Hill is a learning science company and a major educational publisher which provides educational content and software. The first project I worked on is called Common Login Module (CLM), which provides the entrance for all McGraw Hill users. CLM, which achieves web accessibility, uses Angular 5 (TypeScript) as the front end and node.js (ES6) as the back end. It allows user to create an account, login, forgot password or username. The entry point is my.mheducation.com. The second project I worked on is called Internal Admin Tool, which uses Angular 5 and Java 8. Admin Tool is a tool like Postman or Swagger which helps internal users to perform CRUD operations.

Responsibilities:

- Followed **Agile** development methodology and worked closely with the team's technical lead and business analyst to achieve exceptional user experience.
- Developed front end page using **HTML5**, **SASS**, and **Angular 5**.
- Implemented all web pages achieving **Web Accessibility** using **Aria** attribute and created **Responsive Web Design** using **Bootstrap**.
- Used **Angular built-in form validation** and **customized form validation** to verify information before user submits the form.
- Designed and developed **Dynamic Form** for project form simplification.
- Used **Reactive Form** and **Template Driven Form** as well to support data binding and form submission.
- Used **Angular Router**'s **CanActivate**, **CanActivateChild**, **CanDeactivate**, **Resolver** to preprocess and filter request.
- Applied **ngx-pagination** on search result to improve performance.
- Applied **JWT** for user **authentication** and **API authorization**.
- Implemented **Nodemailer** to send user email when they forgot their password.
- Designed and developed **RESTful** services using Http get, post, put, delete connecting front end and back end.
- Maintained pages written by **JSP**.
- Written test cases for every component and service using **Jasmine** and **Karma**, **Mocha** and **Chai**. Configured **Istanbul** for the test coverage.
- Configured **SonarQube** to analyst unit test coverage, check the code to ensure a good code quality.
- Configured **Apache** to solve **cross origin issue**.
- Added **TSLint** in the project to formalize the code.
- Participated in code quality reviews, and helped ensure an excellent standard of the team's code quality.
- Used **Jenkins** and **CircleCI** for continuous integration.
- Used **Morgan** for http log and **Winston** for async logging.
- Used **Swagger** to verify the back-end API and queried data from **MySql** database.
- Used **JIRA** to track tickets assigned to me and used **Trello** to do the team retrospective and communicate with other developers, QAs, and manager and used **GIT** for version control.
- Supported releases and inspected issues via **Sumologic** and **New Relic**.

Java Developer Jul. 2014 – Jan. 2017
United Parcel Service, Inc(UPS) – Paramus, NJ
Projects: Web Tracking(WT), My Choice Delivery Instruction(MCDI), Tracking Component(TC), Cache Manager Client(CMC)

Web Tracking: Web Tracking provides UPS customers the complete and newest info about their packages

via multiple ways. The services UPS customer could find including package tracking, package service updating, package delivery option updating and etc.
MCDI: MCDI is a project which provides customers multiple choices of how, when and where to deliver their packages, customers have to sign up for premium MyChoice users in order to qualify for these additional deliver options.
Tracking Component: TC is a project which resides under Web Tracking and several other projects to handle raw data coming from backend.
CMC: Cache Manager Client is a cache project used to store temporary data retrieved by user from database, in order user to retrieve quicker next time.

- Responsible for Design, coding, unit testing of Java based application changes to new and existing **JavaScript** UI pages as well as **JAVA** components that link the UI to various back end systems. Participate in integration, system, performance testing and product test defect resolution.
- Build a **waterfall**-like development environment for the team to achieve a high work efficiency which includes a weekly project meeting and a weekly staff meeting, in between is emergency meeting in case there's a critical issue needed to be resolved the next day.
- Involved in the UI, front-end logic design and implementation of UPS.com by using technologies such as **HTML4/5**, **CSS3**, **Javascript**, **JQuery**, **Ajax** and **JSP**.
- Applied **AJAX** combined with **JSON** to incorporate the input validation of users' log in/register request; Using **AJAX** calls to call different projects with stored session data.
- Ups.com uses a self-designed framework wrapper of **Struts 1** called PGF/Cclamp to build its own back-end framework structure for UPS projects.
- Responsible for the back-end business logic of the website by using **TDD** to implement and test with **JUnit**, using Core Java concepts and technologies such as **Collections** with **Generics**, **POJO**, **Exception Handling**, **Annotation** and Http/Https technologies such as **cookies**, application **session** and EJB **session**.
- Involved in building reading/writing/deleting data retrieved by Cache Manager Client project with **Restful** web service published by **Spring**.
- Responsible for **Performance test** of Web Tracking and MCDI using **JMeter2.1.2** deployed on local desktop side or remote Linux side. Responsible for creating performance test steps documents and jmx, csv files.
- Responsible for the update and modification of multiple language version control with **String Bundle** files and **Country Bundle** files.
- Participated in the update of Java1.6 to Java1.7, including java environment updates, projects library updates, **Weblogic** and Linux configuration updates such as heap and build path modifications.
- Participated in projects **Output encoding** and **Regression testing**; Participated in QA testing on behalf of WebTracking team by using UPS designated Quality Center.
- Responsible for Log configuration for several projects on remote side using **Log4j**.
- Participated in the entire life cycle of several projects, from **Integration test**, **System test**, **Product test** to **Staging test** and finally Production release. Responsible for submitting migrations to Product test and Staging test with **RPM** Package Manager to deploy on **EJB** based distribution system.
- Using **Ant** for building and updating Java war files, Jar files, Javascript files and Configuration files. Using **CVS** and **TFS** for projects version control and source code management.

Environment: *JDK 1.6/1.7, Javascript/JQuery, JSON/XML, HTML4/5, CSS3, Ajax, Struts1, Spring3.0, JDBC, RESTful, EJB, Javamail, JUnit, Oracle 10g, Eclipse, Weblogic/Jboss, Ant, CVS/TFS, JMeter, Log4j.*

Pearson VUE – Bloomington, MN
Java Full Stack Developer **Nov. 2018 – Apr. 2020**
Web Team Developer
Worked as a consultant for Pearson VUE, taking on any work that was needed at the time. Mostly worked with the web team in their ongoing efforts to maintain and upgrade their product. Worked mostly on the backend, working with upgrading Spring to Spring boot.

- Involved in upgrading services from Java 7 to Java 8
- Worked on moving services from a **JBOSS** module based structure to a **microservice** architecture using **Tomcat**
- Improve deployment by making project more independent and highly available
- Worked on solving issues using **HTML**, **CSS**, and **JavaScript** in the front end sites.
- Work to update websites to make the follow accessibily guidelines and create a mobile-friendly experience.
- Experience working on **Angular** applications in for fixing problems or upgrading applications.
- Use mainly **Gradle** as a build tool for applications and services, with some exposure to **Ant** and **Maven**
- Worked with **node**, **nodist**, and **npm** to manage application dependencies for front end
- Created a **RESTful api** using Spring Framework for the backend.
- Developing web application using **Spring Framework** in a team
- Used **Spring IoC** for dependency injection
- Used **Spring Boot** for auto-configuration of the Spring Framework
- Used **Spring Data JPA** to handle interaction with the SQL database
- Used **Git** and **Bitbucket** for version control
- Worked in a Scrum-based **Agile** development environment
- Unit testing using **Junit** and **Karma**

Environment:
Spring Framework, HTML, CSS, Oracle, SQL, Node.js, npm, Git, Mocha, Jest, Mongo DB, Bootstrap, JIRA, Agile, ActiveMQ, JMS, AWS, React js, Material ui, Bootstrap, Axios js, AJAX,

---

AT&T **2019 December – Present**
Richardson, TX
Java Backend Developer
Project: Identity Life Cycle Management System
For some governmental department like police station, hospital and fire station. They choose to cooperate with the AT&T company to establish a single, interoperable network for internal communications. My team mainly focus on identity-services part. Such as add, update, deactivate the user, do the notification by e-mail or SMS.

Responsibilities:

- Used **Agile** as a development methodology and joined Daily **Scrum** meeting
- Used **JIRA** as a tool for check stories, create tasks and fix defects.
- Build the basic backend on **J2EE** and **Spring Boot** framework
- Used **Postman** and **Talend API Tester** for HTTP request test
- Created functionality and technical specification details on **core java** concepts
- Stored SQL statements in **Oracle** database

- Deployed application on **Apache Tomcat** as a web server,
- Utilized **JSON** format to exchange data and Configured dependency on **Maven**
- Utilized **Git** for version control on different branch
- Tested the project on **Mockito** for unit test
- Recorded the log documents by **log4j**

Environment:
Spring Boot, Spring MVC, Spring IOC, JSON, Mockito, JIRA, Agile, Git, GitHub, Maven, SQL, JSON, Oracle, Tomcat, Postman, Talend API Tester, log4j

---

Walmart Labs, Sunnyvale, CA **Sep 2019 – Feb 2020**
Role: Quantitative Engineer
Supply Chain Automation
Walmart Labs is a subsidiary of Walmart Inc, playing the role of IT department in company. Supply Chain Automation is implementing testing automation for all Walmart projects. With the in-memory automation and E2E automation, developers will be able to trigger their tests, monitor the test workflow, generate and analysis logs for different environments. This process makes testing and deployment much easier and faster than the traditional way.

Responsibilities:

- Write **Dockerfile** and **docker-compose** file to implement **dockerization** and **containerization** for each application or service. Deploy required public services, database and Kafka into docker container with target application to build a real environment.
- Follow **Maven lifecycle** to manage compilation, packaging and testing. Use **Maven plugins** to control each Maven phase.
- Build **Report Portal** platform and implement it with TestNG, Mocha and MongoDB in a VM so that all team members can generate test logs through Report Portal framework for any projects. Maintain Report Portal platform and team accounts. Set up report portal configurations for each project.
- Implement **functional tests** for projects. Discuss with developers about the project workflow and logics, then design test cases for it. Use professional test tools to write tests, such as **TestNG, RestAssured, Arquillian, Arquillian Cube, Arquillian Drone, etc**.
- Bring up **Oracle, SQL Server, MongoDB, Cosmos DB, Cassandra, etc** in docker container and build connection with other containers. Write sql file to pre-load metadata into database. Only use **JDBC** to connect to DB in test case.
- Write script file to implement automation. Design different condition and approach to implement automation for different environment.
- Use **WireMock** to mock 3rd party services in test case. Create a RESTFUL service with **Kafka** so that we can produce and consume message in browser with corresponding URL.
- Do operation on **Json** object via **Jackson**. Convert Json string into **avro** format, encode avro string with **Base64** and generate ramdom UUID for encryption purpose.
- Integrate **Scala/Play/sbt** project with Java/maven project. Build target Scala project via sbt, and write test cases for it in Java. Use Maven to control the entire automation approach.
- Implement E2E automation with **Node JS** and **Mocha**. Set up **Cron Job** for E2E tests.

Environment: JDK 1.8, Junit, TestNG, RestAssured, Arquillian, Report Portal, Postman, Json, Git, Wire Mock, Maven, Docker, Tomcat, Node JS, NPM, NVM, Mocha, Homebrew, sbt, iTerm2, Oracle, SQL Server,

Cosmos DB, MongoDB, Cassandra, Kafka, Zookeeper, Avro, Base64, React, Selenium WebDriver, Jest, Cron job, Looper

---

Google, San Jose, CA **Nov 2018 – Aug 2019**
Role: Java Backend Developer
Company Master
Google is a multinational technology company that specializes in Internet-related services and products. It is considered to be the biggest and most famous technology company in this world. The Company Master is a system to extract and transform master data coming from 3rd party companies and provides Java APIs to export processed data to other teams inside of Google.

Responsibilities:

- Update and improve existing Java mapper classes. Use new entity and attribute structure instead of **GSON**. Migrate traditional code to fit JDK 1.8 code style.
- Implement new mapper class for BillingContact attribute from **proto** file by using Google libraries including **ImmutableList, Predictions, Optional**, etc.
- Build and redesign unit tests for mappers. Implement **flags** setup and read test data method for unit tests. Create **proto, proto.txt, json** format files to be test data.
- Introduce new **flags** "scrambler" and "key_path" to control obfuscating of business data for all type of values via **HmacSha256**. Implement hash function by using **Guava hashing library**. Implement key file reading from **BNS** by using **GoogleFile** and **Google Channel**. Use **Supplier** to cache key in memory for different dev environments by 1 hour.
- Remove deprecated fields and add new required fields into **proto** file. Adjust corresponding mappers and tests. Redesign **proto** messages structure to fit new data format changes in mdm platform **Reltio.**
- Modify **Borg** file to enable push collections to prepare for **Streamz** legacy pull protocol turndown.
- Analyze the impact of **Clarinet** migrating **BCID** policies to L3+E. Update **BCID** L3+E compliance for project **MPM** packages.
- Setup **monitoring and alert system** via **Monarch**. Define custom **streamz matrics** in related actions in order to monitor server actions. Use **mash** to query data. Build alert system by using **Panopticon** and migrate it to **Automon**.
- Create **REST** endpoint for search method of Company Master API. Use **Apps Framework** to create **HTTP** actions and install required modules. Use **Producer** and **PromiseGraph** to implement **async** return. Use **Envelope Server Framework** to respond to both **RPC** requests and **HTTP** requests.
- Create individual module to implement **authentication, authorization and audit logging** for both RPC and HTTP actions. Use **IamAuth** and **IamAuthHelper** to build Policies. Try out various solutions: **GaiaUser auth, LOAS, EUC, SID/SSID, OSID, OAuth2, UpTick, UberProxy, CorpSSO** for authentication. Install **GinModule** to enable audit logging.
- Create unit test for HTTP actions by using **HTTPServiceTester**.

Environment: JDK 1.8, Junit, Mockito, Truth, gLinux, MacOS, Postman, GSON, Json, Google3, Piper, Blaze, Protocol Buffer, Stubby3, MPM, Apps Framework, Guice, Guava, TAP, Borg, Borgcron, Boq, Sigma, BUILD, Ganpati, BCID, Rapid, Blueprint, Clarinet, Streamz, Monarch, Panopticon, Automon, Mash, Borgmon, Buganizer, Critique, Pantheon ETL, Reltio, BNS.

ETS

Princeton, NJ
November 2018 – May 2019
Role: UI/UX Software Developer
Project: Four Skills Assessment App
Description:
The Four Skills Assessment App is used to test university students on their English skills and help place them in the right English course. Currently the project is in its research phase and using lean startup methodology. For this project, I developed different versions of testing items used to survey university students and professors for their opinions.

Responsibilities:

- Converted wireframe from UI developer and used **React** to create MVP testing items into webpages.
- Used **React Hooks** in function components to create testing items.
- Used **Draft.js** to create text editor for users to input and edit text.
- Added audio recorder to testing item to record test takers voice to listen and repeat what the instructor is saying or read aloud the words display on the screen.
- Added **JavaScript, HTML, CSS** to **Qualtrics** to create items used for surveys.
- Used **SASS / CSS** & **Bootstrap** to design webpages.
- Added JSON to **Amazon DynamoDB** to display the text and recording time for different components.
- Created token to uniquely identify test appointment records in **Amazon DynamoDB**.
- Deployed web application to **AWS S3**.
- Participated in daily scrum meeting and weekly Sprint retrospective and planning meeting.

Environment:
React, JavaScript, ES6, JSX, SASS, CSS, Bootstrap, AWS

ETS
Princeton, NJ
November 2018 – Present
Role: UI/UX Software Developer
Project: Winsight
Description:
Winsight is a project that creates different testlets for K-12. For this project, I worked on modifying the design for the testlets.

Responsibilities:

- Used **SASS / CSS** to design different testlets based on wireframe and component library.
- Added CSS animation to testlets based on wireframe and Jira tickets requirements.
- Edited **JSON** to fixed HTML contents.

Environment:
SASS, CSS, JQuery, JavaScript

BNSF Railway Oct 2018 – May 2019
Fort Worth, TX

Java Developer
BNSF Railway is going through modernization initiative for its system from legacy mainframe-based technologies (Natural/COBOL/IMS/DB2) to robust and secure modern systems (Java/J2EE, Spring micro service, Spring Data etc.). As part of the Waybill 2.0, HCL is responsible to transform the legacy system into new system with latest tools and technologies and enhanced features solving existing system problem to better serve customers. Waybill 2.0 application is partnering with the customer and revenue programs to plan, solution and deliver together. Based on complicated business and system requirements, team is analyzing, defining, designing and implementing the business capabilities as software in a robust, reliable and highly scalable system to enable BNSF to deliver and manage the shipment in effective way.

Responsibilities:

- Implemented **J2EE** with **Spring Boot** with **DB2 RDBMS** as back end.
- Implemented **Spring Boot** controllers as **REST** endpoints.
- Extensively used **Spring MVC**.
- Extensively used **Hibernate** with **annotations** to map relationships and to create model classes using Hibernate annotations.
- Implemented **DAO** modules in Java using JPARepository.
- Developed **RESTful** web services by using existing REST API's.
- Used **SQL** statements and procedures to fetch the data from the database.
- Processed **JSON** Request according to different types modules, actions and data request.
- Using **JaCoCo** to ensure unit test line coverage and branch coverage is above 90%.
- Migrated data from old database to new database.
- Implemented custom exception and Validation library to catch exception and error.
- Used REST client / **Postman** to check the response of various API's, and **Swagger** UI as a visual documentation for back-end implementation as well as client-side consumption. Handled various exceptions based on the response.
- Used GIT repository for software configuration management and version control.
- Used **Agile** methodologies to implement the application and participated in daily **Scrum** and biweekly **SPRINT**.
- Utilized **Jenkins** to build and deploy the application.
- Used Gradle as dependency management/build tool, exercised **LMA logger** utility to log error, info and debug messages and using **Junit** and **Integration** Testing to do testing.

Environment
Java1.8, Spring boot, Spring JPA, Hibernate, Gradle, DB2, Junit, TestNG, JaCoCo, Agile, Git, Intellij, Postman, SQL Developer.

Interactive Reporting
Client: ADP – Parsippany, NJ
Role: Software Developer
Duration: Jan 2017 to Aug 2017

Description: Interactive reporting is a part of ADP WFN project. It allows user to choose report from our project and generate report according to the information stored in WFN project. Users can also save their settings for one particular report and reuse it in the future. The report template is changed from Crystal version to Birt version. Birt designer is used to design and generate report using data from DB.

Responsibilities:

- Followed **Agile** development method strictly, having daily scrum, 3-week **sprint**, PI planning, used version control tool **Rally** to ensure all process running correctly.
- Designed and developed front-end page using **HTML/HTML5, CSS/CSS3, AngularJS** and **Bootstrap**.
- Utilized **AngularJS** as the main framework for data-binding and logic handling.
- Developed **Single Page Application** using Angular **UI-Routing, Dependency Injection, Service** and **Factory**.
- Designed different kind of **Filters** to select and style the displayed data.
- Created **Directive** using Angular for repetitive use and customized style.
- Used Angular Form validation and **$http** AJAX Call to check the user input correctness.
- Created Modal to pop up dialogue window, **Pagination** to list options, image carousels to make slide effects using Bootstrap.
- Created front-end unit test case using **Jasmine**.
- Used task runner **Karma** for Integration test and Protractor for End-to-End test.
- Developed using **Java 8** with new features **Lamda** Expression and **Stream**.
- Work as the **Birt** designer to design and develop report template using **Javascript** and Birt's own palette.
- Generate prototype function for all reports using **Javascript** native function combined with Birt's API.
- Create connection to Oracle database using **JDBC** in Birt IDE and use stored procedure to fetch data from DB.
- Write **Restful** web services using **JAVA** to pass parameter chosen by user to Birt server to create the specific report.
- Using **Google Guice** as the tool for **dependency Injection**.
- Adopt **Jersey** as the restful framework to return **Json** format data.
- **SVN** and **BitBucket** are the version control tools we used to do daily commit, merge and revert operations.
- Use **Maven** and **Nexus** for library management.
- Build and deploy project automatically using **Jenkins**

Environment: Agile Methodology, Angular1.5, Java 8, HTML5, JavaScript, dojo, Ajax, Maven, Junit, Oracle, Jenkins, Rally

---

Policy Engine

Client: IPC Systems Inc – New Providence, NJ

Role: Full Stack developer

Duration: June 2016 to Jan 2017

Description: Policy Engine (PE) is a platform designed to help customers understand the impact of rules and regulations in their environment with IPC systems' main product – Turret. PE will import attributes stored in Unigy, which is the system of Turret and let users to make decision to choose what kind of call can be permitted or denied. PE contains three part: PAP, PEP and PDP. PAP (Policy Administration Point) is the part for users to make policy. PEP (Policy Enforcement Point) is an interface, which is responsible for the communication between Unigy and PDP. PDP (Policy Decision Point) aims to

make decision according to policy that users make.

Responsibilities:

- Practiced **Agile** principles including **scrum**, **1-week iterations**, **Continuous Integration**, using **Git**, **Maven**, **RTC** and **Jenkins**.
- Implemented **HTML/HTML5, CSS/CSS3, AngularJS** and **Bootstrap** for web pages.
- Used HTML5 new features such as **web worker** to improve performance.
- Achieved **Single Page Application** and nested views by **AngularJS UI Router** with **$stateProvider** and **$state**.
- Used **Filters** to format displayed data.
- Validated the client inputs by using **AngularJS Form Validation**.
- Used **underscore.js** to simplify operation.
- Used **Bootstrap** to implemented **Responsive design** for different devices, Multi-column based layout and build **Navigation bar, Drop-down menu, Modal, Form, Table, Tabs** etc
- Implemented **ngAnimate** to achieve animated effect such as transition, transform, etc.
- Analyze and represented the data as financial situation pie chart by using **D3.js**.
- Implemented pure annotation based **Spring** Configuration without using xml configuration file.
- Designed and developed **RESTful** service interface using **Spring MVC** to the underlying customer event API.
- Applied **Spring Data JPA** for persistence layer implementation.
- Used **Spring Security** for login authentication and authorization and SHA5 for password encode
- Injected dataSource using **Java Naming/Directory Interface (JNDI).**
- Implemented **Java mail** to send confirmation email and **freemarker** for HTML email build
- Deployed **Atomikos** as **JTA** transaction manager for tomcat to combine multiple transaction mangers into one.

Environment: Agile Methodology, Java 8, Spring, HTML5, CSS3, JavaScript, AngularJS 1.4, Ajax, Bootstrap, Java Mail, JNDI, Maven, Junit, Mockito, MySQL, RTC

Verizon – Piscataway, NJ *Nov 2019 – Present*
Front-end Developer (ACSS Team)

Description:

ACSS (Auto Customer Service Support) team is working for migration code from the old customer service system into a new system using React, Java Script. Our application contains 300 functional components and need a high-performance UI loading support, admin UI can manage all the components, and also it supports worldwide Verizon customer service open, close and manager multiple calls at the same page and connect web socket when the call is activated.

Responsibilities:

- Worked with agile team to migrate Verizon Automatic Customer Support System into responsive, interactive feature by using **JavaScript**, **React, Redux**, **HTML** and **CSS**.
- Made **Saga** calls to handle asyn api calls and store the data into redux store, also use **Redux Ducks** to manage code structure.
- Created dynamic components, like dynamic tables, modals and dropdown box, and programmed interactive features: **styled component,** animations, and navigation bars.

- Made one landing API call to load whole view components and increase the loading time by 40%.
- Wrote **HTML**, **CSS** for development of new highly responsive, web-based user interface.
- Implemented **Recharts**, **ag-grid** and other 3rd party libraries to create dynamic data visualizations for the Dashboard component.
- Used **react tools**, chrome debugger and other browser developer tools to fix the bugs quickly.
- Used **JIRA** to assign tasks, check the acceptance criteria for every task and update progress every day.
- Wrote unit test case using **Enzyme** and **Snapshot** testing with **Jest**.
- Used **Git** commands to raise PR, merge code, code review and commit code.

Environment:
JavaScript, React, Redux, Saga, CSS, GraphQL, Visual Studio, Oracle, JIRA, Git, Enzyme, Agile, Jabber

---

Publicis Media - New York, NY *Feb 2019 – Oct 2019*
UI Developer (Data Science Team)

Description:
Publicis Media is a global media group that creates value for clients through global media agency brands and scaled capabilities across investment, strategy, insights and analytics, data and technology, commerce, performance marketing, and content. For the data science application, my responsibility is to build applications for global agencies and clients which are used to operationalize and automate global media buying across Publicis Media. I implement UI components for a single application by using React.js and designed layouts for web pages by using css. I also help manage the UI team to ensure the high quality application from the team with a focus on continuous improvement. Our platform enforces clean data entry and generates fully formatted taxonomy workbooks for trafficking within local buying and billing systems.

Responsibilities:
- Implemented two applications by using **React.js, Redux, Flux** and some other web libraries.
- Used **Axios** to create different API calls to retrieving or processing data with backend and database.
- Implemented **Utils**, **Actions**, **and Reducers** to manage **redux store** and achieve the interaction with different components.
- Imported **React-table**, **Carbon component**, **react-bootstrap**, **react-select** to reuse those libraries in different pages.
- Applied **react-model** to implement validations and filter data by using pure java script function.
- Allocated tasks for UI teams to make every developer work on different components to decrease our team code conflict.
- Cooperated with DBA, QA and backend team, and in charge of merging and pushing code to different stage of environments.
- Cooperated with design team for UI design, and wrote **HTML**, **SCSS** and imported **SVG** images to implement the layout for each web page.
- Used **GIT** commands to merge code and review code for other UI developers to maintaining the integrity of the code.
- Involved in unit testing and integration testing for each story to make sure all the functionalities are working together.
- Involved in various **Software Development Life Cycle (SDLC)** phases of the project by using **Agile** methodology.

Environment:
JavaScript, React, Redux, Flux, Carbon, React-table, SVG, SCSS, Axios, Visual Studio, Oracle, JIRA, Git, GitHub, SDLC, Agile, Slack