**Kanchha Gurung**

**javakg41@gmail.com**

**(510) 901-0835**‬

**SUMMARY**

* 11+ years of experience for creating and maintaining web applications using software development methodologies like **Water Fall**, **Agile**, **SCRUM, Behavior Driven Development (BDD) and Test Driven Development (TDD)**.
* Experience developing applications using Use Case Diagrams, Behavior Diagram (Sequence Diagram, Collaboration Diagram, State Chart Diagram, Activity Diagram), Packages and Classes using **Rational Rose**.
* Proficient in Design patterns and have extensively used them for developing efficient systems. Design patterns used during my project development includes **Singleton, Service Locator, Business Delegate, MVC, Session Facade, Front Controller, IOC, Composite View, DAO**, and **DTO**.
* Design and developed various multi-tiered client-server applications using **Java** and **J2EE** technologies for various domains like **Healthcare, Banking, Transportation, Investment and E-Commerce.**
* Have solid experience in Service Oriented Architectures **(SOA) like Web Services, WSDL and UDDI**
* Sold hands on experience in developing/building applications in distrusted environments using **Restful** Web Services using **SPRINGMVC** and **SOAP** **REST** Web Services.
* **Web Services** experience in **SOA** using **Apache AXIS, SOAP Framework, WSDL, WSDD, UDDI, SOAP, JAX-WS, REST**, **JAX-RPC.**
* Documented **RESTFUL Web Services** using **SWAGGER.**
* Experienced developing **User Interface** (UI/Frontend) using language like **HTML, XHTML, DHTML, Java Script, CSS, AJAX, Groovy Scripting, JQuery, Tiles, Tag Libraries, and Custom Tags**.
* Experience in **DevOps** tools and techniques like **Urban Code Deploy (uDeploy), Jenkins, and Nexus.**
* Experience in server side development using **Java/J2EE programming** and **JEE Servlet/JSP development, Struts, EJB, JMS, MQ, or Tibco-Queue, JNDI, Spring 2.5, Hibernate, JDBC, Mybatis, XML Schema/DTD**.
* Design and developed application using **Object Oriented Analysis and Design (OOAD) and Object Oriented Methodologies** like **UML**
* Experience building Application using Maven or ANT and Deploying enterprise applications on web/application servers such as **JBOSS, Tomcat, IBM Web Sphere, and WebLogic.**
* Worked with messaging format like **XML and JSON**
* Experienced in using Java IDE tools of IBM **Web Sphere Studio Application Developer (WSAD)**, **Rational Application Developer, Eclipse** and familiar with other IDE’s like **Net Beans**, **JBuilder and JDeveloper**.
* Experienced in writing the **ANT scripts**, Configuring the Connection Pool, **LOG4J** and installing the various application servers.
* Well versed in application development using back ends such as **Oracle, SQL Server, MYSQL** and **DB2** for RDBMS.
* Experience in Database Development using **SQL, PL/SQL, Stored Procedures, Functions, Triggers, TOAD and SQL Loader.**
* Experienced in version management systems with **Clear Case**, Microsoft **VSS**, **CVS** and **SVN**, **GIT** and **STASH**
* Have a good knowledge of **ESB (Enterprise Service Bus)**
* Experienced in troubleshooting in a production environment, System maintenance and monitoring processes
* Experience in testing frameworks like **Junit, Mockito, Cucumber, Selenium and SOAP UI.**
* Experience in triggering event using **QUARTZ** scheduler

**TECHNICAL**

|  |  |
| --- | --- |
| **Languages** | JAVA, XML, C, C++ LA |
| **Java/J2EE** | JMS, MQ, Tibco MQ, JDBC, Hibernate, MyBatis, JNDI, SOAP, REST, Web Services protocols (SOAP, WSDL, UDDI, BPEL), XML, JAXP, JIBX, DOM, SAX, Java Mail, java Swings, AWT, Multithreading, JTA, Custom Tags Servlet, JSP 2.1/2.0/1.2, JSF, JSTL 1.1,EJB 3.0/2.1/1.1/1.0  |
| **Web Technologies** | Html, DHTML, JavaScript, CSS, AJAX, SSL, Web Services (SOAP, REST, JAX RPC, WSDL) |
| **Databases** | Oracle, MYSQL, Sybase, SQL Server, DB2 |
| **OR Mapping** | Hibernate, JDBC, MyBatis |
| **Web/Application Servers** | Tomcat, Web logic, Web sphere, Jboss |
| **Frameworks** | Struts, Spring |
| **Operating Systems** | Unix, Linux, Windows, IOS |
| **IDEs**  | Eclipse, Net Beans, Spring One Source, IBM Integration Designer, IBM Urban Code Deploy |
| **Testing Tools** | Junit, Mockito, Selenium, Cucumber,  |
| **Methodologies** | Waterfall Model, Agile, OOAD, SOA, Agile Practices, TTD and BDD |

**EDUCATION**

* Bachelor in Science in **Computer Science** from **Texas A&M**
* Bachelor in Science in **Mathematics** from **Texas A&M**
* Certified **Java Secure Programmer**

**PROFESSIONAL EXPERIENCE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Client: AutoDesk**

**From : June 2018- Current**

**Role: Sr JavaDeveloper**

**Project : Rover**

Rover is an Autodesk application that handles delivery of applications like **AUTOCAD**, **MAYA, REVIT** to end users . It handles all aspects of delivery of apps like browser download , streaming , licensing, upgrade, bug fix and mapping . It stores all the mapping info like what products are free when you buy certain products and cost licensing and also bug fix and upgrade everything is handled using rover.

**Responsibilities:**

* Followed Agile Methodology for project development following 2 and 3 week sprint.
* Performed **Use case design, object modeling using UML, like transformation of the Use Cases into Class Diagrams, Sequence Diagrams**.
* Worked on **REST** and **SOAP** API using Spring controller, Spring Boot
* Handled exception using global **Exception** Handler with Spring
* Used **AWS S3** bucket to store any project related file
* Used **AWS dynamo DB** to retrieve info and also used **DAX** cluster for caching DB cal
* Used **AWS lambda** to create server less service
* Created **Token** based **Authentication** to validate user
* Worked with **DEVOP** tools like **Jenkins** for continuous Integration and Maven to Build
* Run script in **Linux** environment using **PUTTY**
* Worked with message format like **JSON, XML, HTML**
* Worked with tools like **SOAP UI** and **POSTman** to test REST API and SOAP
* Worked with Testing framework like  **JUNIT, MOCK** and **CUCUMBER**
* Encrypted critical info using **Spring 256** bit encryption
* Deployed application in Dev, QA and UAT environment using Jenkins and putty
* Used **Sonar** for code coverage
* Run script in **Linux** environment using **PUTTY**
* For Security purpose used two way **SSL** handshake to authenticate each other
* Used **git** for version management.
* worked on lot of script for data Migration and for data manipulation
* Implemented **Docker** Architecture to import dependency to different environment
* Converted old legacy application to small Micro service for productivity and speed and easy to deploy using SPRING BOOT

**Environment: Java, J2EE, JDBC, XML, XSD, JMS, SOAP, REST, Web Services, log4j, SOAP UI, POSTMAN, Spring MVC, Junit, Mockito, Maven, Jenkins, Super putty, LDAP, WinScp**

**Client: Innovis-Credit Bureau**

**From : July 2017- June 2018**

**Role: Lead Java Developer**

**Project : DvHost**

DvHost is internal Innovis API to provide credit information regarding individual to Fannie Mae. It provides info regarding individual when there is change in individual credit during the loan process, which typically will go from one to two month. This application will get info like borrower income, assets and employment from IRS and route it to Fannie Mae.

**Responsibilities:**

* Developed DvHost Application using **Agile** Methodology
* Performed **Use case design, object modeling using UML, like transformation of the Use Cases into Class Diagrams, Sequence Diagrams**.
* Created **REST** API using Spring controller
* Handled exception using global **Exception** Handler with Spring
* Created **Token** based **Authentication** to validate user
* Stored data in Oracle Database using **JDBC** template
* Worked with **DEVOP** tools like **Jenkins** for continuous Integration and Maven to Build
* Run script in **Linux** environment using **PUTTY**
* Worked with message format like **JSON, XML, HTML**
* Worked with tools like **SOAP UI** and **POSTman** to test REST API and SOAP
* Worked with Testing framework like  **JUNIT, MOCK** and **CUCUMBER**
* Encrypted critical info susing **Spring 256** bit encryption
* Deployed application in Dev, QA and UAT environment using Jenkins and putty
* Used **Sonar** for code coverage
* Run script in **Linux** environment using **PUTTY**
* For Security purpose used two way **SSL** handshake to authenticate each other
* Used **SVN** for version management.
* Implemented **Docker** Architecture to import dependency to different environment
* Converted old legacy application to small Micro service for productivity and speed and easy to deploy

**Environment: Java, J2EE, JDBC, XML, XSD, JMS, SOAP, REST, Web Services, log4j, SOAP UI, POSTMAN, Spring MVC, Junit, Mockito, Maven, Jenkins, Super putty, LDAP, WinScp**

**Client: AT&T**

**From: Jan 2016 – July 2017**

**Role: Lead Sr Java Developer**

**Project: OCE**

**Location: Richardson, TX**

 OCE stands for order capture engine which is a commercial and service order management system developed internally for AT&T. OCE is based on open source Jboss fuse which is highly scalable and customizable system that serve as the order engine for most of retail products. In OCE, every layer is exposed as API leveraging SOA and CAMEL. OCE supports multi-protocol like SOA, REST, and Messaging etc.

**Responsibilities:**

* Worked in **agile methodologies** which involves every day stand up, **Sprint, design and Story point**
* Create **Design Document** using Sequence Diagram
* Implemented **solution design** to fix bug using **design pattern** to have more efficient application
* Application release **design, analysis, Pre-Testing, Developing, and supporting Clients** after delivering the application.
* Responsible for helping **New Developers** to understand the business and **the environment**
* Managed source code using **SVN** but later changed to **GIT**
* Implemented **Restful** web services using **SPRING MVC and Spring Controller**
* Created **SOAP** based web service and tested using SOAP UI
* Developed **MicroServices** to eliminate long term commitment to same technology
* Implemented **Dependency Injection** using **Setter and Constructor** Method
* Design and implement Active Speaker server, a Multi-thread Java application with Web presentation, developed with AJAX, XML, DOM, CSS and Tomcat
* Worked with **DEVOP** tools like **Jenkins** for continuous Integration and Maven to Build
* Run script in **Linux** environment using **PUTTY**
* Created **Batch script** to create/populate table.
* For Security purpose used two way **SSL** handshake to authenticate each other
* Used GIT for version management.
* Responsible for developing the UI pages using **HTML5**, **CSS3**, **JavaScript**, **AngularJS**, **JSON**, **JQuery**, **and AJAX**.
* Created Controller using **Angular JS**
* Used JBoss fuse **ESB** to create mediation or route message between Services and also handling the message format
* Worked with **big data** using
* Worked on development of **Hibernate**, including mapping files, configuration file and classes to interact with the database.
* Worked on Behavior Driven Testing and Test driven Methodology.
* Used **SOAP UI, JUNIT, MOCK** and **CUCUMBER** for Testing
* Used **ECLEmma** to check whether code coverage is 80 percent or not

**Environment: J2EE, XML, XSD, JSON, SOAP, REST, SOAP UI, Spring MVC, JUNIT, JBOSS FUSE ESB, CUCUMBER, SWAGGER, MOCKITO, Maven, Ant, LDAP, Urban Deploy (uDeploy), JENKINS, HTML, Nexus, RIBBIT, RALLY, SVN, GIT, SQL**

**Client: Fidelity Investments**

**From: Sept 2014- Jan 2016**

**Role: Java Developer**

**Project: ALMaaS**

**Location: Morrisville, NC**

In Fidelity Investment we created application called ALMaaS from Scratch. ALMaaS can automate build process using Jenkins and run test case and coverage using sonar, create LDAP group using plugin, create access, add member to different group and deploy application in Tomcat server. What ALMaaS does is whenever any developer commits any code in any repo like SVN, GIT. ALMaaS app will trigger and run all the test case and also whether the test coverage is above 80 percent using Sonar if it is more then 80 percent then it will update to Nexus Staging repository from where Urban Deploy will retrieve the updated war /ear file from Nexus and deploy it to any application server.

**Responsibilities:**

* Developed application using **Agile Environment** which involved Setting up retrospectives, sprint reviews and removing obstacles and discussing about it in everyday stand up.
* Managed source code using **GIT**
* Consumed and created SOA services like **Soap** and **Restful** Web Services
* Developed Micro Service to make application loosely coupled
* Implemented **Restful** web services using **SPRING MVC and Spring Controller** and documented using **Swagger**
* Used **JSON and XML** message format for web service request and response
* Consumed **JMS** message via **EJB 3.0** simply saving to **POJO**
* Implemented **Dependency Injection** using **Setter and Constructor** Method
* Involved in **CMM** and **CMMI** method and practices as a developer analyst
* Design and implement Active Speaker server, a Multi-thread Java application with Web presentation, developed with AJAX, XML, DOM, CSS and Tomcat
* Used **Jenkins** for continuous Integration and Maven to Build
* Used **Sonar** for code coverage
* Run script in **Linux** environment using **PUTTY**
* For Security purpose used two way **SSL** handshake to authenticate each other
* Used GIT for version management.
* Automate the deployment process using **DEVOP** tools like **Urban Code Deploy** (**uDeploy**)
* Used **CHEF** to manage virtual server in CLOUD
* Created **REST** service using Spring Controller
* Generated **POJO** from **JSON** and **XML**
* Responsible for developing the UI pages using **HTML5**, **CSS3**, **JavaScript**, **AngularJS**, **JSON**, **JQuery**, **and AJAX**.
* Created Controller using **Angular JS**
* Used **ESB** to create mediation or route message between Services and also handling the message format
* Used **Mybatis** to connect with database and create automate mapping
* Worked on development of **Hibernate**, including mapping files, configuration file and classes to interact with the database.
* Worked on Behavior Driven Testing and Test driven Methodology.
* Used **SOAP UI, JUNIT, MOCK** and **CUCUMBER** for Testing
* Deploying application on **Tomcat** Apache Server
* Worked with Business Analyst and Architect to create solution design as per business requirement.
* Skilled in **DevOps** for maintaining continuous integration using **Jenkins** and deploying using **uDeploy** tools.
* Wrote build and deployment script in Unix environment to automate the build and deploy process
* Used **Groovy** to be efficient and productive.

**Environment: J2EE, XML, XSD, JSON, SOAP, REST, SOAPUI, Groovy**, Spring **MVC, JUNIT, TIBCO, CUCUMBER, SWAGGER, MOCKITO, Maven, Ant, LDAP, Urban Deploy (uDeploy), AJAX, JQuery, JSP, HTML, Nexus, RIBBIT, JIRA, GIT, SQL**

**Client: FedEx Ground**

**From: Oct 2013 – Sept 2014**

**Role: Sr. Java Developer**

**Project: Yard Management System (YMS)**

**Location: Pittsburgh, PA**

In FedEx my responsibility was to create application called YMS who keeps track of the entire trailer under FedEx using their coordinate send through GPS transponder. GPS transponder will send message to Tibco queue and YMS will consume JMS message from TIBCO queue where each message is routed to web services, which saved data in database. GPS transponder sends two messages containing coordinate info, which we send to Yard Smart app. We used C3 Yard Smart Web service and Telematics web Service to maintain the yard and keep track of the trailer.

**Responsibilities**:

* Worked in **agile methodologies** which involves every day stand up, **Sprint, design and Story point**
* Followed **Agile methodologies** to create a **solution design** for fix in production.
* Created and consumed **REST** and **SOAP** web services using Spring MVC
* Used **SOAPUI** to test Web Services.
* To interact with other systems asynchronously used **JMS** queue and topic.
* Consumed JMS xml message using EJB 3
* Used **LDAP** to connect with database and queue
* Building application **EAR** and deployed into **WebLogic** Application Server
* Created Design Document using Sequence Diagram (**DDS**)
* Did error handling for different kinds of error
* Developed Exception handling framework and used **log4J**for logging.
* For unit testing used **JUnit** and **Mock** the service using **Mockito**
* Worked with **Spring** Web Service template to send request to C3 Yard Smart Webserver
* Defined schema objects using **XSD** and used maven to generate Java POJO Classes
* Run script in **Linux** environment using **PUTTY**
* Build application using **MAVEN**
* For Security purpose, did two way **SSL** handshake to authenticate each other
* Used **Oracle Service Bus** for data transformation and protocol conversion.
* Redeploy and undeploy C3 application in JBoss Server
* Clean the code using **Refactoring** to make it more efficient and Effective.
* Experience configuring and deploying J2EE application using uDeploy, Jenkins, maven, ant etc.
* Experience in continuous integration and automation process

**Environment: Java, J2EE, JDBC, XML, XSD, JMS, Web Services, log4j, SOAP UI, Spring, Tibco, Junit, Mockito, Maven, Super putty, LDAP, WinScp, JBOSS, ESB**

**Client: Deloitte**

**From: Jan 2013 –Sept 2013**

**Role: Lead Java Developer**

**Project: Health Insurance Exchange for State of Connecticut (CTHIX)**

**Location: Camp hill, Pa**

At Deloitte we created interface for HIX app, which is part of Obama Affordable Care Act using Message Broker. We created **Message Broker** that can receive message from multiple destinations and determine the correct destination and rout the message to the correct channel. We created mediation for the entire health Insurance supplier using IBM Web Sphere ESB to communicate with Federal Hub, SSA, IRS, and DMV etc. to get personal information and to see whether person is eligible for health Insurance benefits. CTHIX System provides abilities for end-user to routes information or message between Health Insurance Company and Federal government.

**Responsibilities**:

* Worked in **Agile** Environment, and coordinated a standup as Scrum Master
* Create secure connection between Federal services using **SSL** handshake
* Involved in **Primary** and **Secondary production** issue **POC** and **resolve** the issues respectively.
* Create **Design Document** using Sequence Diagram
* Application release **design, analysis, Pre-Testing, Developing, and supporting Clients** after delivering the application.
* Responsible for helping **New Developers** to understand the business and **the environment**
* Minimizing server access by cohosting Web services and ESB in one application Server
* Connect to **DB2**database using **JDBC**
* Transformed XML messages to HTML using XSLT.
* Hands on experience in developing **SCA** components for **IBM-Web Sphere Enterprise Service Bus (WESB).**
* Used **Web Sphere Message broker** like **WESB** to create mediation or route message between Services and also handling the message format
* Used **JMS** to interact with other systems asynchronously.
* Used **Struts 2** to create elegant, extensible framework for enterprise-ready Java web applications
* Consume and created **SOAP** based Web Services and tested using **SOAP UI**
* Designed **XML Schema** using **XMLSPY** tool which is used as a data layout for Web Services between client/server and **JMS** messages
* Building application EAR and deployed into **Web Sphere Enterprise Service Bus (WESB) 7.5** Application server
* Build **ESB** application using **IBM Web sphere Integration Designer**
* Encrypting and Decrypting Data in **DB2**
* Used **WSRR** to centrally control services endpoint addresses.
* Integrate **REST APIs** in application and website to work with different kind of platforms
* **Refractor** the code to make application more efficient and clean by eliminating dead code and duplicate code

**Environment: Java, J2EE, JDBC, XML, XSLT, XSD, JMS, Web Services, Struts 2 log4j, SOAP, ESB 7.5, MQ, Web Sphere Integration developer 7.5, XMLSPY, SOAP UI, DB2, WSRR**

**Client: Matson**

**June 2010– Dec 2012**

**Role: Sr. Java Developer**

**Project: Oracle to JBOSS Migration**

**Location: Phoenix, AZ**

In Matson, my day-to-day responsibility was to deploy their Oracle based web application to **Jboss**. Test the application using **Selenium and Junit**. Create the **JMS** Bridge to send message from **Jboss** to **Web Logic**. Test the functional requirement of each project. Fix the Bug and create test case for testing. Add new functionalities to application depending upon the requirements.

**Responsibilities**:

* Data Operations are performed using **Spring ORM** wiring with **Hibernate**
* Involved in writing **HQL queries, Criteria queries and SQL queries** for the Data access layer
* Used **ant** and **maven** build tools to compile the enterprise application and create the WAR & EAR files
* Building application using **ANT** and **MAVEN**
* Developed Unit, Integration and Performance Test Cases using **Junit and Selenium**
* Developed Exception handling framework and used **log4J**for logging.
* Used **XPATH** to retrieve value from XML messages and **XSLT** to transform XML messages from one format to another
* Changed the oracle based data source to **JBoss** based data source.
* Developed the application on Spring One Source and deployed it on **JBOSS**.
* Test the accuracy of the data fetched from the DB using **Junit**
* Hands on Experience in **SOA and Web services** to get the data back and forth.
* Used **JMS** publish/subscribe model to interact with other systems asynchronously. The revenue data was pushed to multiple systems using the **JMS** model.
* Created mediation and routed message between Services using **ESB**
* Refractor the code to make application more efficient and clean by eliminating dead code and duplicate code

**Environment: Java, J2EE, JDBC, XML, XSLT, XSD, spring, JBOSS, ESB, Hibernate, JMS, Web Services, log4j, Jboss, Selenium, Unit, SOAP**