**ABHILASH CHAVA**

**Java Full Stack Developer**

**(931)-532-0607**

**abheejava@gmail.com**

**Professional Summary:**

* **Sr. Full Stack Java Developer**with over **8+ years** of experience in **Analysis**, **Design**, **Development**, **Testing** and **Implementation of** **Enterprise** **Client/Server** and **Distributed** **Applications** across the entire **Software Development Life Cycle** using **Java** **and** **J2EE** **Technologies.**
* Expertise on advanced software development methodologies such as **Agile/Scrum**, **Waterfall** and **Test-Driven Development** (TDD) model processes.
* Experience in **Use Cases**, **Sequence Diagrams** and **Class Diagrams** using **UML** and **J2EE Design Patterns** like **Model-View-Controller(MVC)**, **Singleton**, **Abstract Factory**, **Session Façade**, **DAO**, **DTO, Service Locator** and **Business Delegate** in developing multi-tier distributed web applications.
* **Strong web development** **skills** using **HTML/HTML5**, **CSS/CSS3**, **JavaScript (ES5 & ES6)**, **jQuery**, **Bootstrap**, **React JS**, **XML**, **AJAX**, **JSON** which meets **W3C Web Standards**.
* Solid Experience with Core Java technologies including: **Collections**, **Interfaces**, **Synchronization**, **Generics, Serialization**, **IO**, **Exception Handling**, **Multi-Threading**.
* Expertise in **J2EE Architecture** using **Java**, **JDBC**, **EJB 3.2**, **Servlets 3.x/4.x**, **JMS, JSP**, **Hibernate 3.2/4.x**, **Struts**, **Maven**,

**Spring**, **Web services (Axis2.0**, **JAX WS 2.0**, **JAX RS)** and **Rule Engines** **(Drools 4.1, Easy Rules, Rulebook)**.

* Expertise in **Spring framework**, extensively used **Spring IOC**, **Spring MVC**, **Spring DI**, **Spring AOP**, **Spring Eureka**, **Spring Ribbon, Spring Netflix, Spring Zuul Proxy, Spring JDBC** and **Spring Boot Microservices**.
* Experience in **Spring Security Framework** for User Access Management and involved **LDAP** configuration for authentication, authorization & Implementing **Spring Cloud**-REST call using **Netflix Feign Client.**
* Experience in **Java** **8** features like **Lambda expressions**, **Stream API** & **Predicate functions**.
* Experience in **Hibernate, Hibernate Query Language (HQL), Transactions, Exceptions, Java Message Service (JMS).**
* Expertise with Messaging Service like **Kafka,** **RabbitMQ** and **Zookeeper** apart from **JMS**.
* Experienced in **MongoDB, Express JS, AngularJS 2.0 & 4.0**, **Node JS**, **(MEAN Stack)** development.
* Working knowledge on **Micro services**that helps in creating independent services which can be deployed easily.
* Proficiency in server-side Java development with **RESTful web services, AJAX** powered web applications.
* Experience in **Web servers** like **Tomcat** 8, **WebSphere** 8.x/9.x, **Jetty**, **JBoss**7.1.0 and **WebLogic** 10.0.
* Experienced on **NoSQL**databases - **Cassandra** & **MongoDB**, database performance tuning & data modeling.
* Experience in **RDBMS** technologies like **Oracle**, **MySQL**, and **Postgres** using Functions, Triggers, and Stored Procedures

and involved in the activity of complex queries, by using **MySQL** 5.7.19, **DB2**, **Oracle** 9i/9g/10g/11g.

* Experienced in development, testing and deployment of enterprise applications on **Windows** & **UNIX** platforms using

IDE's such as **Eclipse**, **Spring Tool Suite**(STS), **NetBeans**, and **IntelliJ**.

* Good Experience in working with open source tools **Selenium**(IDE, RC, and Web Driver), Performed **Unit testing**,

**Integration Testing** & generating of test cases for web applications using **Junit**, **TestNG**.

* Developed **CI/CD** system with **Jenkins** on **Kubernetes** container environment utilizing **Kubernetes.**
* Hands on experience with **build tools** like **Ant**, **Maven**, **Gradle** and **logging tools** like **Log4J** and **Splunk**.
* Deployed applications (**EAR**, **WAR**) on **Web Sphere** Application servers and clusters.
* Hands on Experience of utilizing cloud technologies including **AWS** (Amazon Web Services), **Spring** **Cloud**, **Microsoft**

**Azure** and **PCF** (Pivotal Cloud Foundry).

* Experienced in configuration and deployment on **Application Servers** such as **IBM WebSphere**, **Oracle WebLogic**,

**Red Hat JBoss** and **Apache Tomcat**.

* Hands on experience in **Amazon Web Services** (AWS) provisioning and good knowledge of AWS services like **EC2**, **S3**, **Elastic Beanstalk**, **Elastic Load Balancing**, **VPC**, **Route 53**, **Cloud Watch, Cloud Formation**, **SQS**, **IAM**, **SNS** & **Elastic Cache**.
* Quick to learn modern technologies and ability to work in tight schedules on different applications concurrently.

**Technical Skills:**

|  |  |
| --- | --- |
| **Java/J2EE Technologies** | Servlets, JSP, JSTL, JDBC, JMS, JNDI, RMI, EJB, JFC/Swing, AWT, Applets, Multi-threading, Java Networking. |
| **Programming Languages** | Java JDK 1.4/1.5/1.6/1.7/1.8, C++, C, SQL, PL/SQL, Python , Golang |
| **Application/Web Servers:** | Oracle/BEA, WebLogic 8.1/9.1/10.3, IBM WebSphere 5.1/6.0/6.1/7.0, JBoss, Tomcat 5.0.28/6.0, Bug tracking, Jenkins, Cucumber. |
| **IDEs** | Eclipse, Spring STS, IntelliJ, WebStorm, NetBeans. |
| **Web technologies** | HTML5, CSS3, XHTML, JavaScript(+ES6), TypeScript, Ajax, jQuery, AngularJS, Angular 2, Angular 4, ReactJS, NodeJS, Socket.io, ExpressJS, JSON, Swagger UI, Bootstrap. |
| **AWS Skills /cloud** | EC2, S3, RDS, EBS, IAM, AMI, ELB, CLI, SNS, RDS, VPN, NAT, GIT, DNS, Route53, DBA, Dynamo DB, PCF. |
| **Web Services** | JAX-WS, JAX-RPC, JAX-RS, SOAP, REST, SOAP UI, Microservices. |
| **Methodologies** | Agile, Scrum, RUP, TDD, OOAD, SDLC, Waterfall model. |
| **Modeling Tools** | UML, Rational Rose, Visio |
| **Message Brokers** | Kafka, RabbitMQ, ActiveMQ, TIBCO, IBMMQ. |
| **Testing technologies/tools** | JUnit, JMeter, Protractor, Mockito. |
| **Database** | Oracle 8i/9i/10g/11G, DB2, SQL Server 2005/2008, MySQL, PostgreSQL, Sybase, MongoDB, Cassandra. |
| **Big Data Tools** | Hadoop, Spark, MapReduce, HDFS, HBase, Zookeeper, Hive, Sqoop, Cassandra, Flume, Couchbase, MongoDB, Neo4J, Redis, MLlib. |
| **Version Control** | CVS, SVN, Git, Stash, Bitbucket. |
| **Build Tools / CI/CD** | ANT, Maven, Gradle, JENKINS. |
| **Spring Tools** | Spring MVC, IOC, AOP, JDBC, JTA, IO, Spring Boot, Spring Micro services, Spring REST, Spring Eureka, Spring Hate os, Spring Cloud, Swagger UI, Sleuth, and Spring Zuul, Feign Client. |
| **Containerization Tools** | Docker, Docker Swarm, Kubernetes. |
| **Platforms** | Windows 10/7/XP, LINUX, UNIX, OS X. |

**Client: Dish Network - Nashville, TN JUN 2016 - Present**

**Designation: Sr. Full Mean Stack Developer**

**Description:** AXIOM is an order entry tool developed for Dish Network's OE/Telco Partner and self-user customer. This application is used for creating new customer account and updating of existing customer account. It was later updated to be used by retailers and by direct Dish agents. It replaced the legacy application Reconnect which was using in past by retailers. Also, now Dish agents are using Axiom to create new Orders, in past they user another application named Dish Promo. Using this application partners, retailers and dish agents can select different programming (Core, Premium, Ala Carte, and International) and products based on their zip code location. It also has some facilities to add/modify the services for an existing customer.

**Responsibilities:**

* Involved in the Design and Development of project modules, gather requirements, analyze, and implemented business logic using **Agile Scrum** Software Development methodology.
* Defined and developed the application’s presentation layer in UI development with technologies like **HTML 5**, **CSS3**, **JavaScript** and **Angular JS**.
* Worked in all the modules of the application which involved **front-end presentation logic** developed using **JSP**, **JSTL**, **JavaScript** & **jQuery**.



* Used Angular JS and **Node JS** to structure JavaScript code in an MVC model (**Model-View-Controller**). Used Angular JS, D3JS to build a **single-page application** and used **Angular JS 4.0**, we created custom directives for re-usable components (multi-field form elements, background file uploads, etc.)
* Used **Java 8** features like **lambda** **expressions** for communicating between business layer and database, **stream** API for Bulk Data Operations on Collections using both streams and parallel streams.
* Implementing or exposing the Microservices architecture with **Spring Boot** based Microservices interacting through a combination of **RESTful** and **Apache Kafka** message brokers.
* Worked on **Camel-based integration** middle-ware solution for Provisioning Services by design and the Implementation of business logic and data processing routes using **Apache Camel**.
* Configured **Swagger-UI** registered **Micro service** with **Eureka server** to monitor the service health check from **Spring Boot**, **Spring JPA 2.1 administration console** and **Hystrix dashboard**.
* Implemented fail safe and **circuit breaker** patterns in **Spring Boot** **Microservice** applications using **Hystrix**, configured **Swagger-UI** to registered Microservice with **Eureka server** and monitored service health check from **Spring Boot** admin console and **Spring Ribbon** for load balancing.
* Initially we have used **NodeJS**/**Express.JS** to mock **REST** services and built the UI app using that mocked backend. Later we integrated with the real backend system and coordinated with server-side development team.
* Implemented pre-authentication and data base security using **Spring Security** and securing application access and **Spring Batch** for scheduling and maintenance of batch jobs.
* Used, Configured Apache **Kafka** cluster, replications , partitions using stream API and run arbitrations.
* Implemented **Spring** **ORM** with **Hibernate** taking advantage of features like **Annotation** metadata, **Auto wiring**, and **Collections** to implement **DAO** layer with Hibernate **Session Factory**, **HQL**, and **SQL**.
* Extensively used **Hibernate** concepts such as **Lazy loading**, **Dirty checking**, **Locking**, and **Transactions**.
* Created/managed **Mongo DB**/**NoSQL** collections. Developed Server-side automation using **Node.js** scripting and connecting different types of SQL and NoSQL databases.
* Experience with building **Docker Images** and running them on **Docker container**.
* Used **Drools engine** for validating business rules before executing in runtime.
* Used Micro service architecture with Spring Boot, Spring JPA 2.1 based services interacting through the powerful messaging tools such as **REST**ful, **Apache Kafka** message brokers.
* Used **Jenkins** as an AWS code deploys plug-in to deploy to AWS. Experience with **DevOps** methodologies, implementing **CI/CD Pipelines** using tool sets like **GIT**, Jenkins, **Chef** and **AWS**.
* Implemented micro services using Spring Boot and **Pivotal Cloud Foundry**.
* Deployed Spring boot application using **Pivotal Cloud Foundry.**
* Worked with several testing frameworks such as **JUnit**, **Easy** **Mock**, **Mockito** and **Power** **Mock**.
* Created **Test automation** framework using **Cucumber** and **Selenium** web driver.
* Used **Git** for version control and **JIRA** for bug tracking, project management and **Maven** as build tool.
* Worked closely with QA team and fixed QA bugs as well as production issues with a quick turnaround time.

**Environment: Java 8, Spring Boot, MVC, Spring Eureka, Hibernate, JSP 2.0, JPA2.1MongoDB, NodeJS, AJAX, XML, Golang, Oracle 11g, Docker, Nginix, jQuery, Angular 2/Angular 4, JavaScript, IBM RAD, Express.JS,** **Kubernetes, Multi-threading, Agile, Design Patterns, Git, Apache Maven 2.5.1, Oracle WebLogic, HTML 5, XSLT, RESTful Services.**

**Client: Change Healthcare, Lombard, IL JAN 2015 – JUN 2016**

**Designation: Sr. Full Stack Developer**

**Description:** This is a web application based on J2EE architecture. It provides web-based access to inpatient and outpatient data from the Data Repository. Users can review as well as update or enter new data through data repository. Many clinicians use Data Repository daily for accessing patient data, which aids in making important clinical decisions. Data Repository is mainly composed of various modules running inside of the application. The project aims to provide the company with data to develop evidence-based health policy decisions as well. The client programs are used for issuing or retrieving data from the database depending on the query for finding health policy premium.

**Responsibilities:**

* Actively involved in **Analysis**, **Design**, **Development**, **System** **Testing** and **User Acceptance Testing** and successfully followed agile methodology in the Scrum model.
* Developed Interactive web pages using **Angular JS 2.0**, **HTML5**, **CSS3** and **Bootstrap**.
* Created the **React.JS Express.JS Server** combined with **Socket.io** to build **MVC framework** from front-end side Angular JS to back-end **MongoDB**, in order to provide broadcast service as well as chatting service.
* Built **RESTful web service** by **AngularJS Server** in the back-end side to handle the particular requests which are sent from the front-end side jQuery Ajax calls.
* Extensively used Core Java concepts **collections**, **generics**, **Concurrency** to improve the performance of the applications and to schedule tasks using **Multithreading**.
* Used **Spring Annotations** to simplify the business logic for creating **Spring MVC RESTful** web services and **Spring Boot Actuator** to externalize configuration properties for different environments.
* Deployed **Docker** contained **Spring Boot** Microservices in to **AWS EC2** container service using AWS admin console and used **lambda** to run applications in **AWS** console.
* Used **Spring Transaction** to provide a consistent programming model across different transaction APIs such as Hibernate and to Support declarative transaction management in application.
* Developed **Hibernate** in **Data Access Layer** and Implemented Hibernate Template to great extent making use of HQL, Named Parameters, Named Queries and interfaces such as Expression, Query and Criteria.
* Build **Docker containers** and used for synchronous/asynchronous connection between two applications running in different servers. Tested REST services using google chrome's **POSTMAN** plugin.
* Used micro-framework to implement with REST API and **Mongo DB** (NO SQL) as backend database.
* Implemented **Apache Camel** routes using Camel-Spring XML and Camel-Spring processor beans.
* Implemented **Gradle Script**s to create JAR, WAR, EAR and dependency JARS and deployed the entire project on the **JBoss** Application Server. Involved in developing and executing **automation scripts** using **Selenium** web driver with Eclipse IDE.
* Setup the continuous Integration (CI) process and continuous deployment (CD) jobs in **Jenkins** brew master instance, which reduces the time for the deployment of **micro services**.
* Prepared **Junit** and **Test NG** test cases and executed as part of auto build process from **Jenkins** Jobs.
* Integrated **JMeter** and **Load Runner Test** cases into Gradle and Monitored error logs using Log4J and fixed the problems.

**Environment: Java 7, J2EE, CI/CD, Spring Boot, Webservices, Microservices, Swagger, Spring Zuul, Netflix Eureka, Restful, Feign Client, Apache Camel, Kafka, JUnit, JBPM, Drools, Spring Cloud, AWS, EC2, S3 ,Express.JS, MongoDB, HTML5, CSS3, Python, JavaScript/ES6, Bootstrap, jQuery, Angular 2, Gradle, AJAX, jQuery, XML,XSLT, Java Beans, UML, Log4j, Agile, JBoss Server, Docker, Spark, Kubernetes.**

**Client: Comdata, Brentwood, TN JUN 2013 – JAN 2015**

**Role: Sr. Java Full Stack Developer**

**Description:**  Corporate Payment is contained framework with multiple applications. These are the applications which hosts different kinds of tests in different areas which test user’s skills and abilities in that particular area. The stakeholders of the applications are mainly test takers and administrators who host these tests from different parts of the world. Test takers use the scores of these tests as credibility for multiple educational opportunities.

**Responsibilities:**

* Involved in various **SDLC** phases like Requirement gathering, **Design**, **Analysis** and Code development with both iterative (**Agile**, **Waterfall**).
* Implemented **AngularJS** Controllers to maintain each View data and Service calls using Angular Factory.
* Developed A dashboard based on **Ajax** calls with Business Customers Count, Flags, Real-time graphs for Analytical Reporting using **NodeJS**. Tested **REST** services using google chrome’s **POSTMAN** plugin.
* Used **ReactJS** to create views to hook up models to the DOM and synchronize data with server as a SPA.
* Implemented **Java EE** components using **Spring** **IOC**, **Spring Transactions** and **Spring Security** modules.
* Consumed messages from **JMS** driven beans and **Active MQ**. Integrated Active MQ services with Beans.
* **Hibernate DAO** classes were developed to perform CRUD operations using Spring manager classes.
* Developed views and templates with **Python** and **Django's** view controller and templating language to created user-friendly website interface.
* Developed real time tracking of class schedules using NodeJS (socket.io, Express JS framework).
* Cloud environment for Microservices was provided using **Spring Netflix** (**Eureka server**, **Hystrix** and **Ribbon**).
* Expertise with employment of **Spring Kafka** and **Zookeeper** apart from **JMS** as messaging service.
* Experience in Creating, Dropping Indexes, Tables, Stored Procs and Views on **Oracle DB**.
* Worked with modules like **mongoose** for database persistence using **NodeJS** to interact with **MongoDB**.
* **Apache Mesos** and **Kubernetes** were used as the best ecosystem to deploy and manage Microservices.
* Deployed the Application into **Docker** container and made easily accessible at runtime using **Pivotal** **Cloud Foundry** and other cloud services like **Netflix Eureka**, **Spring Hystrix** as fallback method, **Spring Ribbon** for load balancing and a broker service to maintain status using **Spring** **Kafka** and **Zookeeper**.
* Developed applications using **Rule** **Engines**, **Drools 4.x**, and **ILOG** to validate the business User Roles.
* Analyze and resolve conflicts related to merging of source code for **Git**.
* Deployed applications on multiple WebLogic Server instances and maintained Load balancing, High availability and Fail over for the server. **Log4J** is used for logging purposes and debug levels are defined for controlling what we log.
* Writing Unit test cases using **JUnit** testing framework and performed unit and system testing.
* Expertise on **Selenium** Data Driven framework and integrating it with **Maven** (Build Management Tool), **TestNG** (Unit Testing tool) and **Jenkins** (Continuous Integration Tool).

**Environment: AWS, Java 6, J2EE, Pivotal Cloud Foundry, Java Beans, NodeJS, Spring, Spring Kafka, Zookeeper, AWS, CI/CD, jQuery, EJB, Hibernate 2.0, Eclipse, ReactJS, DevOps, Angular, Spark, Jenkins, Oracle 11g, MongoDB, PL/SQL, Selenium, Cucumber, Log4j, XML, HTML5, CSS3, REST, JMS, Active MQ, Web Services, Web Logic server, Jasmine, Maven, Git.**

**Client: GMAC Insurance and Financial Services, Salem, NC MAY 2012 – JUN 2013**

**Designation: Sr. J2EE/Java Developer**

**Description:** The Service Automation Program is to develop a technology platform which will facilitate effective automation of back office business processes in order to consistently achieve high level of process efficiency as well as make the back-office process more extensible, adaptive to future changes. This project is implemented for GMAC servicing Organizations (GMAC Auto Insurance, GMAC Finance) in US and Canada.

**Responsibilities:**

* Experienced with **Software Development Life Cycle(SDLC)** with object oriented of business process and worked on developing Service oriented architecture applications in agile development environment.
* Developed UI using **AngularJS**, **HTML5**, **CSS3**, **jQuery, Bootstrap**, **JavaScript**, **Ajax**.
* Well versed with core Java concepts like **Collections**, **Swing**, **Serialization** and **Servlets** & **JSP.**
* Developed and implemented Business Requirements using **Spring MVC** framework. Implemented cross-cutting concerns such as logging, authentication and system performance using **Spring AOP**.
* Implemented web services in **Service Oriented Architecture (SOA)** to retrieve data from client side **NodeJS** and parsed the data to client and handled the security and Authorization using **Spring Security** and **OAuth** authentication protocol respectively.
* Developed reusable and interoperable services, based on **SOAP**, **WSDL**, **JAXWS**, **JAXRPC** Web services.
* Used **Jackson** parser for **Marshaling** and **Un-Marshaling**.
* Used **Jackson** Custom Mappers for Custom manipulation for date, currency conversations on **JSON** Data.
* Developed **POJO** objects and used **Hibernate** as the **Object-Relational Mapping (ORM)** tool to access the persistent **data from SQL Server.**
* Used Hibernate Detached Criteria, **HQL** and written some SQL queries for data retrieving.
* Hands-on writing **Stored Procedures**, **functions**, **triggers**, **DDL**, **DML SQL** **queries** and **PL/SQL**.
* Worked on **JPA** as persistence layer, involving EJB annotations to synchronize with DB through **XML**.
* **JBoss** Application server implementation, support, workload management and performance tuning including Web server setup configuration with JBOSS App Server.
* Developed **PL/SQL** store procedures to convert the data from **Oracle** to **MongoDB**.
* Developed Test classes using **JUnit** and **Easy Mock** and test coverage using **EMMA** plug-in.
* Analyze and resolve conflicts related to merging of source code for **Git** and **Maven** as build tool.

**Environment: Java, J2EE, Java Beans, JMS, NodeJS, Spring, jQuery, EJB, Hibernate, IntelliJ, Angular, Oracle 11g, MongoDB, PL/SQL, Cucumber, Log4j, XML, HTML5, JBoss Server, CSS3, JavaScript, Bootstrap, SOAP, Maven, Git.**

**Client: American National Bank Cleveland, OH NOV 2010 – MAY 2012**

**Role: Java Developer**

**Description:** American National Bank is one of the secured financial institutions that serves huge domain of customers. Bank offers various financial and banking services to its customers. The current application is a part of online banking that allows a customer to pay bills securely as well as enroll in monthly automatic recurring bill payment. The functionalities involved in e-bill payment are Add a Payee, make a Payment, set up Automatic Payments, Receive Bills Electronically, Request E-mail Notifications and Review Payments. The application also supports intra and interbank transfer of funds for a given customer.

**Responsibilities:**

* Involved in Full Cycle of **SDLC** through Design, Development, Integration and testing phases.
* Designed and developed **Spring MVC** and **SOAP** based **Web services** to integrate multiple applications.
* Integrated **SOAP** based web services into the web application to enable payments and claims through mobile apps.
* Involved in the upgrade process of **Spring** 2.0to **Spring** 3.0and **Hibernate** 2.0to **Hibernate** 3.0 to take advantage of new Spring and **Hibernate** features.
* **Web** **Services** used **Java** 5, **JSP**, **Servlets**, **EJB** 3.0, and deployed on **WebLogic** and **WebSphere**.
* **Jenkins** to Build integration and deployments to the various environments and aligned the release implementations.
* Configured **faces-config.xml** and application-context.xml for **JSF** based.
* Dynamic web pages are developed using **JSF** 2.0, **JavaBeans**, **html**, **JavaScript** and **Custom** **Tags**.
* Implemented state-specific business logic in Java code using **Spring configurations**.
* Reviewed code and developed **Junit** test cases for all complex business scenarios and ensured 100% code coverage.
* Implemented different types of spring controllers as per application requirements, **DAO** and service layer components using **Spring**/**Hibernate** **API’s** and annotations. Extensively used **Hibernate** (HQL).
* Extensively used **Spring IOC**, configured Application Context files and performed database object mapping using **Hibernate** annotations, used **Jasper** Reports as a reporting tool.
* Developed and deployed Session Beans and **JPA** Entities for database updates.
* Developed client request validations and processing using **JavaScript** and **HTML**.
* Used **ANT** for compilation and building **JAR** files, Used **SVN** version control tool.
* Deployed applications on **Web Logic** Application Server and performed required configuration changes.

**Environment: Java 1.5, JavaScript, AJAX, JSP, Servlet, JMS, Spring MVC 3.0, Hibernate 3.0, JSF, SOAP Web Services, Axis2, JPA, WebLogic Application Server, Jenkins, ANT, HTML, SVN, XML, Cloud Foundry, Unix Shell Scripting, SQL, Windows and Oracle 9i, Tomcat, Jasper reports and Maven.**

**Client: United Telecom Limited, Bangalore, India JUN 2009 – SEP 2010**

**Role: Jr. Java J2EE Developer**

**Description:** United Telecom Limited is an Indian company which specialize in networks of telecommunication which is based in Bangalore India.

**Responsibilities:**

* **Analysis**, **design** and **development** of application based on J2EE and design patterns.
* Involved in all phases of SDLC (Software Development Life Cycle) using **Waterfall** **model**.
* Developed user interface using **JSP**, **HTML**, **CSS** and **JavaScript**.
* Involved in developing functional model, object model and dynamic model using **UML**.
* Development of the Java classes to be used in JSP and **Servlets**.
* Implemented asynchronous functionalities like e-mail notification using **JMS**.
* Implemented **Multithreading** to achieve consistent concurrency in the application.
* Implemented **Struts MVC** design pattern and front controller pattern to this project.
* Used Test Driven Development approach and wrote many unit and integration test cases.
* Used **Eclipse** as IDE tool to develop the application and **JIRA** for bug and issue tracking.
* Wrote SQL queries for accessing data from database and used **JDBC API** for DB Communication.
* Used **ANT** tool for building and packaging the application.
* For configuration management of the project used **SVN**

**Environment: J2EE, Java 1.5, Core Java, Struts, Servlets, HTML, CSS, JSP, XML, JavaScript, Eclipse, MySQL, JDBC, JUnit, ANT, SVN, Apache Tomcat Server, JIRA.**