

Decorator Design pattern Decorator Design pattern Decorator Design pattern - Scivantage (499 Washington Blvd A seasoned Core/Server Side Java Developer with extensive experience in Multi-Threading, Concurrency, Synchronization, Thread Pool, Collection Framework, Data Structure, Design Patterns, Java Memory Model, GC tuning, Unix/Linux etc. Proficient in designing, developing of high throughput, high availability, concurrent and fault tolerant software systems/n-Tier Distributed Enterprise Application in different phases of SDLC, including Requirement gathering, Analysis, Design/Architecture, Development/Integration, Build Management, Enhancements, Testing etc. Background in e-Trading, pre-trade capture, Security Lending/Stock loan, Order Management System, Fixed Income. Experienced in Low Latency Java Development, GC Tuning/Elimination, JVM tuning, Stack Trace Analysis, Collections, Sorting, Algorithms etc. Have knowledge of Hadoop ecosystem, HDFS, Big Data, RDBMS, HIVE, HDFS, Map-Reduce. Experienced on working with Big Data and Hadoop File System (HDFS). Have extensive working experience in multiple BFS domains like Fixed Income, Risk (CCAR), Regulatory Reporting, Retail banking, Security Lending, Portfolio Management, Order Management System, Mortgage etc. A self-motivated, enthusiast, quick learner, detail-oriented developer, experienced in delivering high-quality, scalable, predictable, and high-performance applications. Work Experience Decorator Design pattern Scivantage (499 Washington Blvd May 2018 to Present New Port) May 2018 - Present Project Description: The objective of Fixed Income project Enchiladas is to build a Debt Processing Engine which can generate OID accrual schedules for both Corporate and Municipal bonds that may/may not be OID eligible and to process Fixed Income transactions, determine the purchase condition and apply the appropriate cost basis adjustments. In addition, the engine will also need functionality to create accrual schedules for Complex Debt instruments such as Contingent Debt, Variable Rate Debt, Inflation indexed, Stepped Rate, Payment-in-kind, NQSI, Deferred, STRIPS, and Fixed Rate Capital securities. This project is divided into Day 1 and Day 2. Day 1 requirements include Less Complex Debt securities and calculations. Day 2 Requirements include More Complex Debt securities and calculations. RESPONSIBILITIES: Designed and developed the Enchiladas Fixed Income Engine. The core Engine calculates the Base OID

Schedule, Payment Schedule, Accrual schedule, depending upon if the Adjustment type is Bond Premium, Acquisition Premium or Market Discount. Developed the OID YTM (Yield to Maturity) calculator using Newton Rhapson method. Developed the Engine project as a library (a Maven project). Designed and developed the REST API using JAX-RS, Jersey. Developed a generic ETL Framework by using Java Blocking Queue, Executor Services, and I/O APIs. Used Java Generics for developing generic Parser and Mapper classes. Designed and developed user friendly User Defined Exception handling Framework. Used Java IO APIs, Decorator Design pattern for Writing/Reading to/from multiple types of file. Developed Unit test cases using Junit, Mockito/Power Mockito. Tested REST APIs from POSTMAN Application. Extensively used JAXB APIs for parsing XML/Text based data files. Created/modified XSD schema for input data type (POJO) validation. Used Class Loader to invoke the right Parser/Mapper dynamically. Implemented Security Layer by using Web Token, Authentication Filter etc. Used Java 8 features like Streams, Lambda, Functional Interface, Producer, Consumer etc. Used Hibernate Query, Session, Transaction, Entity, Typedef APIs for ORM mapping along with DAO for data access using Hibernate to fetch data from database. Configured Hibernate Connection Factory (C3P0) Attributes to handle situations like Timeout and Stale Connection etc. Designed and developed Batch jobs to generate Transactions and Error reports for Day 0 and Delta. Used Executor Framework, JAXB, ehcache, Apache DBCP, JDBC, Collections, and Apache Commons etc. Implemented caching mechanism for Instrument details using Ehcache. Created Maven Project using POM file to configure the project artifacts, dependency management, build life cycle, excluding configuration files from the generated war files. Performed trouble shooting including stack trace analysis and JVM tuning by changing to Heap sizes according to the requirement. Managed and monitored the JVM performance by Heap Size, Garbage collection, taking Thread dumps. Implemented the logging mechanism using Log4j framework. Changed Rollover policy from TimeBasedRotationpolicy to FixedWindowRotationPolicy in order to have a fixed number of archive files over a rolling log. Deployed the war file in a WebLogic portal hosted on AWS EC2 instance. Managed deployment, maintained lifecycle of the hosted portal instance. Created AWS stack using script. Handled Load

balancer, Cloud Formation, Security group etc. Created/modified Jenkins CICD pipeline for build automation. Implemented Validation frame work for validating Input Data file. Developed shell scripts to automate Batch/ETL jobs. Investigated into JIRA tickets and did analysis and provided solution for them. Converted a monolithic app to microservices architecture using Spring Boot using 12-factor app methodology, as a POC. Involved in the implementation of a Continuous Delivery pipeline with Docker, Jenkins and GitHub and AWS AMI's. Environment: JDK 1.8, GlassFish, AWS, JAX-RS, Jersey, Eclipse, Apache IO, JAXB, Apache Commons, maven, XML, Postgres, Git, JUnit, Jenkins, AWS, JIRA, Hibernate etc. JPMC (575 Washington Blvd February 2017 to May 2018 Project Description: Metadata is the key driving force for all activities in the Ecosystem. All the incoming data has to have technical registration information and lineage defined before it is stored in the data reservoir. UDS (Unified Data Service) strives to maintain information about the end-to-end flow of data in the Ecosystem consisting of service-oriented, technical, operational, security and workload specific information associated with the data and its processes. Usually Customer PII data (Personally Identifiable Information) is registered for a later consumption (Ingestion). This PII data can be shared across LOBs as well. UDS Registration portal allows business user register zones which helps to connect different data sources, register data domains and data sets. It allows to register unified data processors and data flows as well; these components will be used in ingestion application to ingest data into Hadoop environment and further data will be used for data analytics. Data ingestion service allows for data to be ingested into one of the three Hadoop zones within the Hadoop data reservoir. Data can come in the form of a file, table or directory. Hadoop Zones are as follows: Hadoop Achieve, Hadoop Conformed, Hadoop Semantic.

RESPONSIBILITIES: Singlehandedly created/maintained workflow using Core Java, Collection, Spring Restful, Spring MVC to implement CRUD functionality on functionalities at Tenant and DataSet Level. Designed application modules using Micro-services architecture. Used Core Java, Collection Framework, and DAO Layer to implement this. Used Spring Integration, Spring Boot, and Core Java to build and connect Intra applications components. Created/maintained In-memory/ Junit test cases to incorporate End to End testing. Also provided Junit test cases using

Mockito, PowerMockito mocking frameworks for portal project. Created/maintained batch jobs using Spring Batch and Spring Boot for EOD batch processing of Customer data related batch jobs.

Created and maintained ARM scripts for automatic checkout from bit-bucket and consequent build and deployment using Jenkins, maven etc. Created Database objects like Tables, sequence, index, Views, GRANTS, and Roles etc. Maintained the DB Scripts in Git/bit bucket.

Used/modified using Java 8 features like Stream, Lambda Expression in the existing code. Used EclEmma code coverage tools and was part of code review process. Created/maintained deployment processes of Registration/ Ingestion in DEV/QA environment. Executed queries using Hive and developed Map-Reduce jobs to analyze data. Developed Hive queries for the analysts. Utilized Apache Hadoop environment by Hortonworks. Involved in loading data from LINUX and UNIX file system to HDFS. Migrated existing projects from SVN to Git with commit history.

Created Jenkins built scripts for separate projects and pipelines. Created Log4j configuration corresponding to different environment. Environment: JDK 1.8, Tomcat, Eclipse Mars, Spring Batch, Spring Boot, Spring Integration, Git/Git extensions, Maven, WinScp, Putty, JAXB, XML, Oracle, SVN, XSD, Mockito, PowerMockito, JUnit, Jenkins, Linux, JIRA, Hibernate, Hadoop, Cloudera cluster. Barclays - New York, NY October 2015 to January 2017 Project Description: Comprehensive Capital Analysis and Review (CCAR), a regulatory framework was the Dodd-Frank implementation specified by FED (Federal Reserve) in order to assess, regulate and supervise large Bank Holding Companies. Project Octon is Barclay's response to Dodd-Frank S165. CCAR, along with the annual Dodd-Frank Act stress tests (DFAST), were designed as complementary initiatives to strengthen supervisory assessments of capital adequacy and processes through which large, complex BHCs initially assess their capital needs. Initially calculation are created as Hadoop jobs, invoked by Monikers (which represent the SQLs). Later on because of the advantage of the in memory processing we have replaced the Hadoop jobs by spark processing. RESPONSIBILITIES:

Implemented E2E (End To End) process consists of one or multiple Processes, which in turn consists of one or more Activities or Sub-Processes. Designed BPMN models of the E2E process using Eclipse BPMN2 modeler, configured Activity/Process details. Developed Spark programs

using Scala API's to compare the performance of Spark vs Hive and SQL. Used Spark API over Hortonworks Hadoop YARN to perform analytics on data in Hive. Implemented Spark using Scala and Spark SQL for faster testing and processing of data. Developed WorkItemHandlers and service implementations corresponding to BPMN Models using jBPM (Eclipse plugin) and Core Java. Developed different types of Activities e.g. Data publishing, Data Validation, ETL, Report Generation (Axiom). Fixed Form Reports related MetaData is SFTPed to Axiom Reporting team. Developed SFTP framework using JSch (Java Secure Channel) APIs. Tested the code in Kerberos enabled platform. Developed File Watcher Script for SFTP application. During PPNR calculation an in-house Model Execution Framework is used to read input monikers, validate monikers, perform MEF execution, capture Output, validate output and write to mercury. Used Core Java, Collections, JSON, jBPM, Hibernate to develop WorkItemHandlers, Service Implementations classes. Projective RWA (Risk Weighted Assets) Calculations/ Transform are of two types, Spot and Standardized. WorkItem Handlers, Service Implementations classes are done by using Core Java, Collection, JSON, jBPM, Hibernate etc. CCAR and DAST test requires simulated stressed market data. BHC bank needs to carry out the Stress Tests for total 5 scenarios (3 Fed given and 2 Internal). Each scenario consists of 28 RiskFactors and each of them consists of a number of shocks values. SMT (Scenario Management Tool) is used to generate the simulated Scenario data.

Used jBPM WorkFlows, BPMN Modeler 2 plugins, Core Java, JSON, Hibernate etc. Static Runbook has the details of the Top level processes, their child processes and the corresponding activities. Static Runbook gives a reference of the ideal time to run a complete CCAR Execution from start to end. Whereas, Dynamic Runbook calculation provides actual time taken by each of the processes/activities. Designed the Static and Dynamic Runbook and developed using Core Java, Data Structure, Collection, RestTemplate client etc. Used Multiple Predecessor logic to determine the dependency of processes. Developed Context Services, Moniker Services, DrillDown Services etc. which provides data used in Calculation and Fixed Form Reporting. Developed the services using Spring RestTemplate, JDBCTemplate and Core Java. Developed different DAO Implementation using Spring JDBC Template in order to pre-populate the Run Details,

notifications etc. at the start of the application. Developed a tool to generate Fudged Messages and deliver it to a Queue/Topic to during Unit testing of the Workflows. Worked on Tomcat DB Connection Pool configuration to achieve optimum number of DB connections using Apache Commons DBCP. Extensively used Git/ Git Extensions as the repository. Used Mockito and JUnit for developing Unit Test code. Developed and maintained Stored Procedure, Named Query to implement WorkFlow Functionalities. Used Goals and Profiles in during build to create war for specific environments using Maven. Environment: Core Java 1.7, Tomcat, Eclipse Plugins(BPMN modelers, JBoss Drools), Git/Git extensions, maven, WinScp, Putty, JAXB, XML, Oracle, Eclipse Kepler, SVN, XSD, Mocito, JUnit, Jenkins, Teamcity, Ant, Jsch, Windows 7, Linux, HP ALM, Hibernate. Goldman Sachs (30 Hudson, Jersey City) December 2014 to September 2015 Positions Reporting (RegOps) teams Large Options Positions Report (LOPR) is for compliance reporting to OCC. The project goal is to migrate the US LOPR function to Regulatory Operations strategic infrastructure and to build an enhanced operational support model. RESPONSIBILITIES: Worked with Operations to analyze the OCC Reference document to understand the specification and created gap analysis document between the intended system and the existing system. Generated Model classes from the Position Schema of OCC system using JAXB APIs. Designed and developed a batch application to download table data from PROD databases to flat files(by identifying business entity) using Core Java, Spring JDBC, Spring Batch, Log4j2, Multithreading, Executor Service, Executors, DAO Framework etc. Developed some "File Watcher" application, which loads data files to some secured location using SFTP, Core Java, JSCH (Java Secure Channel APIs). Designed and developed a reconciliation tool (POC), which takes metadata information about previous report runs and get the report results. It takes the report results as inputs and generates the delta difference. This delta difference is going to be the basis of the LOPR report. Used Java Collection, XML parsing, Java IO, and Generics API's to code classes. Designed and developed application to parse Snap Shot data from OCC in the FIXML format and parsed it according to the OCC Reference manual and uploaded using JDBC batch insert to the Sybase IQ database. Did coding to include issuers coded as SECTION_604 regulation under BHCA

monitoring report. Changed Global Report to Regulation, Global Issuer to Regulation and Global Issuer to Threshold mapping. Tested in Dev and UAT environment. Coded change in business functionality to map Global Issuer to Regulation mapping and Global Product to Regulation mapping under the same Regulation and comments. Did a POC using Spring Batch to download Production data files of size 100 GB by a single threaded application to a data file. Developed Mock JUnit Test cases to mock Business functionality testing. Did JUnit data validation with static In-memory data instead of validation against database. Environment: Core Java 1.5/1.6/1.7, JAXB, XML, Sybase IQ, Sybase ASE, DB2, Eclipse Kepler, SVN, XSD, Mocito, JUnit, Maven, Jenkins, Teamcity, Windows 7, Linux. Contract First Web Service Capital One 360 - Wilmington, DE July 2013 to October 2014 Wilmington, Delaware July 2013 - Oct 2014 Project Description: TranSite, SASSy and IVR the three Web Service Clients communicate with Wesp (Server side). Wesp accesses Profile and Oracle Database to get the customer data. Checkmate is used to upload and deposit check images using mobile or ATM banking. RESPONSIBILITIES: Devised the algorithm, Coded to implement it (a Contract First Web Service) using Spring-WS, Spring MVC, Spring IoC, XSD, JAXB, DAO Framework, Exception handling, Collection and Core Java. Used Tidal Enterprise Client to create/modify scheduled jobs in windows based agent. Developed Java Server Side code used to be invoked from Tidal Clients. Promoted the jobs developed in Dev Tidal environment to QA and Production environment. Coded the Tax Form Generating Module for 2013 Tax year to generate tax forms for 1098, 1099-R,1099-S,1099-A,1099-C,1099-DIV,1099-INT,1099-MISC using Core Java, Collection Framework, Java Exception, XML, XSLT, Velocity Template, XSD and JAXB. Used Maven/Git/Gerrit during development code check out and local build before the code check-in. Also used SVN version control system during source code management process for old style version management system. Coded Unit Test cases for every Java class using Junit/Mocito Testing Framework. In order to meet or exceed the high code coverage (90% minimum code coverage) used tools like JaCoco and EcEmma. Used tools like CPD, PMD, FindBugs to find subtle bugs in the application like duplication of code, unused code, improper indentation etc. Coded in Java

Servlets, used to play the audio files needed depending upon the workflow executed for IVR application. Configured both 32/64 bit Tomcat instance for 32 /64 bit applications. Did performance tuning to the Server configuration for optimization. Designed, developed Server Side caching mechanism using Core Java, ehCache API, Collections, Spring API, RESTful API. Tested, bug fixed, deployed the application. Designed, developed, Coded the "KeepAliveConnection" implementation using Core Java, ExecutorService, Multi-Threading, Collection, Exception, Generics etc. Coded and configured Server Cache implementation using ehCache to enhance the performance. Implemented File based cache overflow configuration. Coded customized Cache Manager Factory classes to handle configurable Disk Store Location attribute. Implemented distributed Logging mechanism by using LOGback, logger and appender for each of the separate interfaces and web application. Coded using LOGback API and Java to implement per user Logging facility. Implemented Time and Size based Roll over policy for logger. Coded Logging Context set/ reset mechanism to facilitate the log files with different contexts. Developed solution to change the log levels of independent modules dynamically using UI tool. Implemented tool where users can view/download log files (for debugging purpose) for an environment without server access using RESTful API and Spring MVC. Developed Resource (images) Externalization Framework using Spring MVC to keep resources out of the deployed war file using Spring MVC. Coded Mock client interface as a backup of the actual implementation (when service is down) using Spring AOP.

Developed data access implementation using DAO Framework, spring JDBC Template and SQL Named Query. Coded web service client implementation using Spring MVC, Restful Web Service API. Retrofitted SOAP based client code implementation. Used AntHillPro during DEV/QA/UAT built process to maintain automatic build life cycle management. Environment: Core Java 1.6, Spring (MVC, IoC), Spring-WS, log4j, Git/ Gerrit, SVN, SOAP, JMS/MQ series, XML, XSLT, XSD, AntHillPro, Eclipse, Mocito, JUnit, Maven, Callable Statement, Stored Procedure, Velocity Framework, Apache tomcat, Jenkins Slave, CPD, PMD, JaCoco, FindBugs, EclEmma. Morgan Stanley Smith Barney (NYC) - New York, NY November 2012 to June 2013 Nov2012 - June2013

Project Description: MSSB 3D is an advisor platform developed/integrated as a result of the joint

venture of Morgan Stanley (MS 360 application) and Smith Barney (legacy NextGen application). The main goal of the project (PMR) is to provide a set of Portfolio Reports through a separate application within MSSB 3D. This application enables Financial Advisors (FAs), Branch Operations and their clients to obtain the financial and portfolio information they are accustomed to. The Portfolio Reports application utilizes new and existing MSSB 3D and MS360 web services, interfaces and data sources.

Senior Developer Wells Fargo Home Mortgage - Frederick, MD April 2011 to October 2012 Project Description: The ASDS (Asset Sales Delivery System) is a system for packaging loans (home mortgages) for sale on the secondary market. A Trade Desk Analyst requests Loan data be extracted from ASDS into the ASBE (Asset Sales Best Execution) system. The Analyst's request results in a query being sent to ASDS and ASDS responds by sending the resulting information to a file on a shared drive for access by the ASBE system or Trade Analyst. The ASBE system optimizes a mix of the imported Loans into Pools, which in turn helps to Trade the loans easily.

Developer JPMC (NYC) - New York, NY April 2010 to December 2010 Project Description: Stock Loan Strategic Investment's Pre trading activities has two parts: "Availability" and "Overnight Locates". In "Availability" application the number of each security from each of the client is gathered and then the "Haircut" rules are applied. In "Overnight Locates" application the end-clients sends requests for Locates of the securities they trade. After receiving the requests the response files are sent, which will have the quantities of securities allocates. Framework consists of components like Client Gateway adapters, In-memory Cache database, Rule engine, and JMS component and controller agent. System processes feeds from different clients using adapters which supports incoming files via different protocols (email, ftp etc.). Rule engine provides different rules on messages which gets persist in Database.

Rackson Asset Management, NY August 2009 to April 2010 Description: Rackson developed their own low latency e-trading system for their algorithmic trading system. Their trading system arguably low volume but faster than most of the contemporary Trading system available. One aspect of the application was to monitor the Order Updates and accordingly place new order, cancel the order, resubmit the order etc. In essence it was their indigenous Order Management System (OMS) they were developing. Role: Financial

Solution Hub - Princeton, NJ December 2008 to August 2009 Description: FSH is a complete solution (STP), which would help financial clients by integrating Front office, Middle office and Back office solutions. FSH has the following modules Integration Engine, Business Rule Engine, Reconciliation Engine, Enterprise Risk Engine, Accounting Engine, Basel II Engine and a BPM solution.

Sungard Financial - Parsippany, NJ January 2008 to December 2008 Description: Getpaid is a treasury solution suite in Accounts Receivable (AR) and Collections Management area. It helps corporations to optimize and manage working capital, particularly the credit and collections processes of the AR department. It helps streamline the order-to-cash process by consolidating customer invoice information, automating collection tasks, driving discrepancy resolution and generating advanced cash forecasts. It provides an enterprise-wide view of cash and working capital across accounts receivable, accounts payable and treasury.

Infinity Business Process Automation software is a BPM tool. It leverages by easy visual modeling, easy integration of Java/J2EE application with the Business Process Management framework, scalability etc.

Cognizant Technology Solution November 2002 to December 2007 Description: While in Cognizant I got chance to work for clients like Merck, Metlife, Credit Suisse and JPMC. A Relationship Manager maintains safekeeping account by creating/execution/reviewing of orders. Handling Real Time Positions, Order Entry, Settlements, Non Trading Orders are some of the major activities. Global AD Core was a monitoring project to monitor reconciliation of the accounts, portfolio management and trading applications. Developed APIs for an Order Management Systems (OMS) for an E-Trading platform.

Cogentech January 2002 to November 2002 Description: The Web Application developed for NJDOL (New Jersey Department of Labor) was meant for the Judicial system. Developed UI and Server side components like Action class, Servlet, Controller, ActionForm etc. using Struts , JDBC, JSP. Wrote some utility functions used in this project.

ADA Software and Services April 1999 to December 2001 Description: Worked in projects on Antilock Braking System (ABS). The software was developed in an 8 bit microcontroller (Siemens C505), which with the help of another 16 bit microcontroller (of Texas Instrument) continuously monitor the wheel speed of the vehicle and calculate the acceleration as well. The speed control is required to avoid the "locked"

condition of the car, while the speed of the car suddenly changes. Education BE in Electronics and Telecommunication Engineering Jadavpur University - Kolkata, West Bengal Skills BEA, BEA WEBLOGIC, ECLIPSE, EJB, J2EE

Name: Lauren Howard

Email: blairmanuel@example.org

Phone: 921-238-4751x93790