

INDUSTRIAL TRAINING REPORT

**TRAINING ORGANIZATION : LONDON STOCK EXCHANGE
GROUP TECHNOLOGY**

PERIOD OF TRAINING : FROM 19/03/2019 TO 02/08/2019

FIELD OF SPECIALIZATION : COMPUTER ENGINEERING

D.S.M.M.B. DASSANAYAKA

E/14/065

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude for following individuals and organizations who supported and encouraged me to succeed in my internship period at LSEG Technology (MillenniumIT) Pvt. Ltd.

My special thanks go to Mr. Asitha Gunathilaka - Specialist Software Engineer, Eng. Chanaka Karunaratna - Software Engineer, Mr. Madava Viranjan - Tech Lead, for their coordination and supervision throughout the internship period. I would like to thank my line manager Eng. Chaminda Aluthgedara - Associate Software Architect for great supervision throughout the internship period. My gratitude goes to Eng. Thilina Illangasinghe, Specialist Software Engineer who gave me consistent support to work on “Fortress” framework. I would like to extend my gratitude to all the team members in the OTC team for the support.

I would like to express my gratitude for our training coordinator, Dr. Janaka Alawathugoda – Senior Lecturer, Dr. Dammika Elkaduwa - Former Head of Department and all the lecturers in the Department of Computer Engineering, the University of Peradeniya for preparing and giving advice for working on internship training.

And also I would like to acknowledge Mr. W.R.M.U. Wickramasinghe and all the staff members in ITCGU in University of Peradeniya and NAITA, for all the commitments they have done for our internship period.

CONTENTS

Acknowledgments		i
Contents		ii
List of Figures		iv
List of Tables		v
List of Abbreviations		vi
Chapter 1	INTRODUCTION	1
1.1	Training Session	1
1.2	Introduction to Training Organization	1
1.3	Summary of the work done in Training	5
Chapter 2	TECHNICAL BACKGROUND TO THE COMPANY	6
2.1	Overview	6
2.2	Fortress Framework	6
2.3	Sysguard	7
2.4	Serendib Message Protocol	10
2.5	Bamboo Automation	10
2.6	Yaan Front End	12
2.7	Behaviour Driven Development	13
2.8	Nexus Repository Manager	15
2.9	Common Technologies	16
Chapter 3	MY TRAINING EXPERIENCE	20
3.1	Overview	20
3.2	My Training Placement	20
3.3	Induction And Orientation Program	20
3.4	Opportunities And Experience	22
3.5	Technical Exposure	22
3.6	How useful was the knowledge gained from the Faculty	31

Chapter 4	NON-TECHNICAL EXPOSURE	33
4.1	Overview	33
4.2	Annual Trip	33
4.3	Micarnival 2019	33
4.4	Presentation Skill Session	33
	CONCLUSION	34

LIST OF FIGURES

Figure 1.1	LSEG Logo	2
Figure 1.2	LSEG Technology Logo	2
Figure 1.3	MIT Head Quarter Building at Malabe	2
Figure 1.4	Top Level Management	3
Figure 1.5	Organization Structure LSEG Technology	4
Figure 2.1	Fortress Overview	7
Figure 2.2	SysGuard Login Window	7
Figure 2.3	SysGuard Main Workspace	8
Figure 2.4	SysGuard Process View	9
Figure 2.5	Bamboo Plan Anatomy	11
Figure 2.6	Bamboo Dashboard	12
Figure 2.7	Yaan Front End	13
Figure 2.8	BDD Process in MIT	14
Figure 2.9	Nexus Repository Manager	14
Figure 2.10	LogViewer from Fortress Studio Plugin	16
Figure 2.11	JIRA Dashboard	17
Figure 2.12	Xshell and Millennium System Manager	18
Figure 3.1	My Learning Page	21
Figure 3.2	Layered Architecture in Millennium Exchange	23
Figure 3.3	Fortress Framework Differentiation for each product/solution	23
Figure 3.4	Overview of Native Server	24
Figure 3.5	miniATS Class Diagram	27
Figure 3.6	Changepom.py script	28
Figure 3.7	Bamboo plan configured to show test results on Bamboo Dashboard	29
Figure 3.8	Online Cron Expression generator	31

LIST OF TABLES

Table 3.1	Native Server Classes and Functionalities	25
-----------	---	----

LIST OF ABBREVIATIONS

3TS	Third Party Trading System
BDD	Behavior Driven Development
BSL	Business Service Limited
CEO	Chief Executive Officer
CI	Continuous Integration
CSE	Colombo Stock Exchange
CV	Curriculum Vitae
DSO	Domain Specific Object
IDE	Integrated Development Environment
ITCGU	Industrial Training and Career Guidance Unit
LOB	Line of Business
LSEG	London Stock Exchange Group
MIT	Millennium Information Technologies
NAITA	National Apprentice and Industrial Training Authority
OTC	Over The Counter
PM	Project Manager
POM	Project Object Model
QA	Quality Assurance
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UK	United Kingdom
USA	United States of America

Chapter 1

INTRODUCTION

1.1 TRAINING SESSION

I did my internship as a research intern at London Stock Exchange Group Technology, No.1, Millennium Drive, Malabe during the period of from 19th of May 2019 to 02th of August 2019. I sent my CV for an internship through the Department of Computer Engineering, University of Peradeniya and got selected after an interviewing process.

1.2 INTRODUCTION TO TRAINING ORGANIZATION

LSEG Technology or previously recognized as Millennium Information Technologies is a company that is involved in Capital Market software products. Diving into a little bit of history, MIT was founded in 1996 by Mr. Tony Weerasinghe. The contract to design and install an automated processing system for the Colombo Stock Exchange in 1997 lead to MIT's suite of capital market software products. Since then the capital market software delivered by MIT found customers all around the world including the USA, Singapore, South Africa, and the UK, etc. In 2002, MIT was moved into its headquarters in Malabe with the first Silicon Valley-style software campus in Sri Lanka in a 19.6 acre land with a swimming pool, gym, café, playground, daycare, accommodation, transportation, badminton court, Basketball court, Table tennis area, Video gaming area, Yoga, etc.

In 2009 London Stock Exchange Group acquired MIT with a contract for a new enhanced, fast trading platform for equities. This acquisition was a big milestone in MIT history because after that the company entered the international market with a considerable number of orders from all around the world. Now the company is rebranded as LSEG Technology, the technical support provider for the mother company LSEG. It delivers six different capital market technical products & services.

- Millennium Exchange
- Millennium Surveillance
- Millennium SOR
- Millennium Market Data
- Millennium PostTrade

- Millennium LiveOps



Figure 1.1: LSEG Logo



Figure 1.2: LSEG Technology Logo



Figure 1.3: MIT Head Quarter Building at Malabe

LSE being the 7th biggest stock market in the world provides a solid base for the company to stand as a leading tech-company in Sri Lanka. As of 2019, LSEG Technology houses over 500 employees. It has a huge organization hierarchy expanded over more than 3 continents.



Figure 1.4: Top Level Management

LSEG Technology is one of the leading stock exchange software vendors and it belongs to the London Stock Exchange Group. Because of this international exposure, LSEG Technology is renowned for friendly, open, flat and peaceful company culture.

All the employees know each other and get together at various events. There is a club named 'MiClub' formed by employees in LSEG Technology to improve relationships among employees and organize events. And also this is recommended for all newcomers because it will ease the adaptation process for the working environment and the projects. Although it is part of a very large organization, the company maintains a beautiful environment. LSEG Technology promotes to follow 4 main values for its employees.

- Innovation
- Excellence
- Integrity
- Partnership

All employees in the company are addressed by their first name irrespective of their position. They allow employees to stay comfortable. So, casual wear with shoes or sandals is permitted to wear for the usual work. Usually, the team organizes lunch-outs twice a month and they never charge from the interns. There are birthday celebrations within the organization they cut their birthday cakes and decorate their working section with balloons

stickers, posters and also there are a lot of large scale events organized such as New Year Ceremonies, Musical events, and Sports events and Annual Trip, etc.

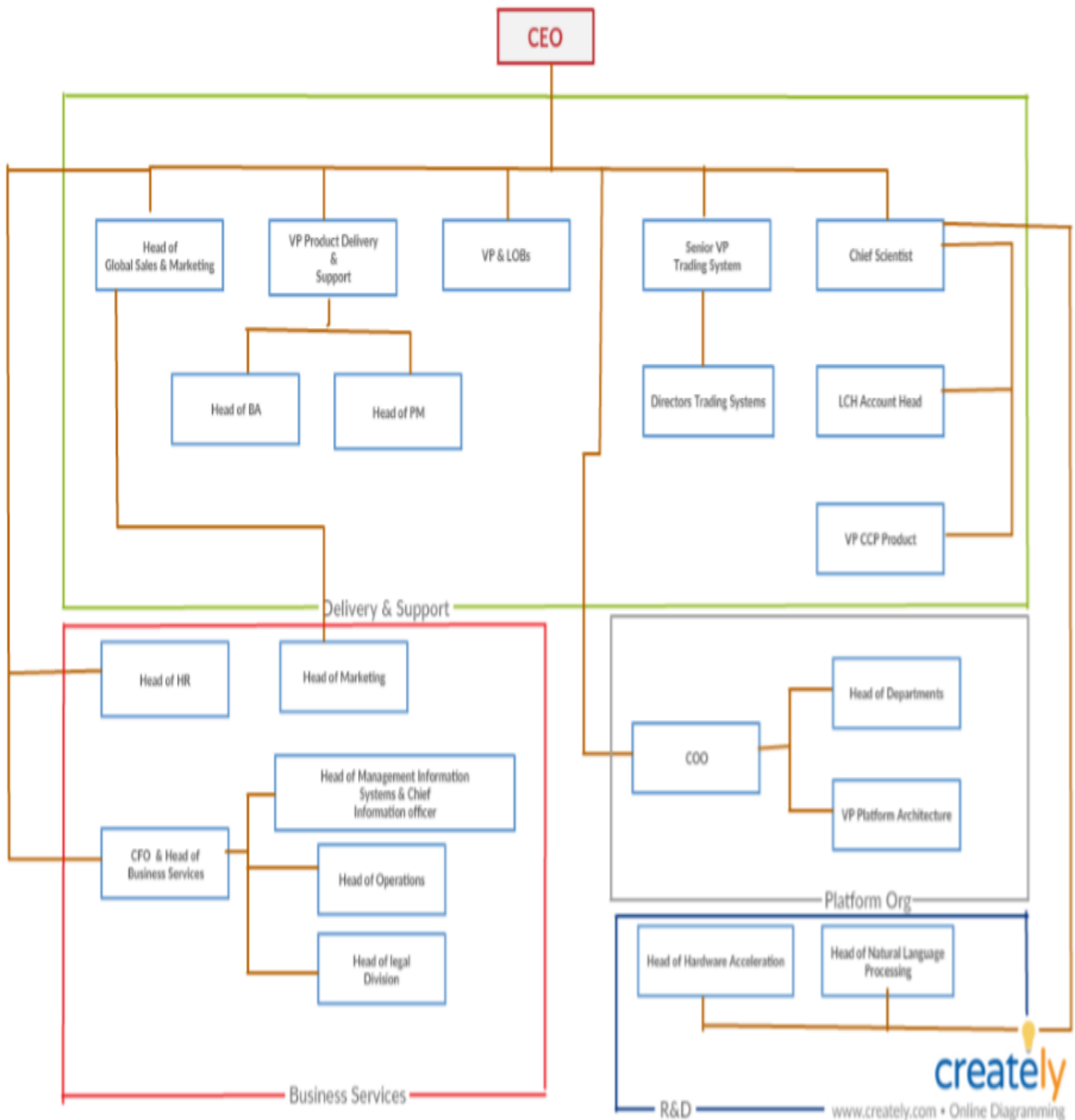


Figure 1.5: Organization Structure LSEG Technology

Company Vision - “MillenniumIT is the partner of choice for organizations needing real-time, high performance, agile & resilient capital market technology. We set the global standard for multiasset, pre-to post-trade STP solutions for financial markets.”

1.3 SUMMARY OF THE WORK DONE IN TRAINING

This training report is based on the overall training experience I had from LSEG Technology (MillenniumIT pvt.Ltd) within my 20 weeks of the internship period. This report consists of three chapters, the first chapter contains the information about the training establishment, main functions, organizational structure and hierarchical levels of the company. The second chapter includes a comprehensive description of the technical background of the company. It focuses on the general technology approach to the company. The third chapter describes my training experience including technical aspects how I face new problems and overcome them, new things I learned and how beneficial was the things learned from the Department, etc. The last chapter debriefs all non-technical exposure I had within LSEG Technology for the 20 weeks of training including social events, Annual Trip and Outings, etc.

Chapter 2

TECHNICAL BACKGROUND TO THE COMPANY

2.1 OVERVIEW

This chapter describes the technologies used in LSEG Technology. They have their own technology stack. Most of the tools are patented to the company. Only the used technologies are described here. Most of the underlying operations and functions of the tools are not described due to copyright policies in the company.

2.2 FORTRESS FRAMEWORK

Fortress is a test framework for MillenniumIT systems. It is a combination of three things. The Java language, the open-source unit test framework called TestNG/jUnit and a set of in-house built custom libraries and programs to augment the first two. The third element, custom libraries, and programs, consist of three layers.

1. Fortress Core
2. DSO Framework
3. Fortress Test Suite

The first one is the core layer that is common to all MillenniumIT products. The second layer is the framework layer that provides a set of domain-specific objects. The third is the test layer, which will host components or end-to-end tests. The framework and test layers will exist for both product and solution projects. TestNG and JUnit are open source testing framework. The other three layers are in-house-built tools. Fortress core is maintained by the Tools team and DSO Framework is maintained by Platform team. All the developers are writing test suites on the Fortress Test Suite layer using TestNG or jUnit.

The current implementation of Fortress requires each product or solution to keep its own framework to support the test projects. This separation is required to cater to the following differences.

- Message level differences
- Entity level differences

- Component level differences (Ex: New components)
- Maintain tests for new functionalities
- Customization of product functionalities

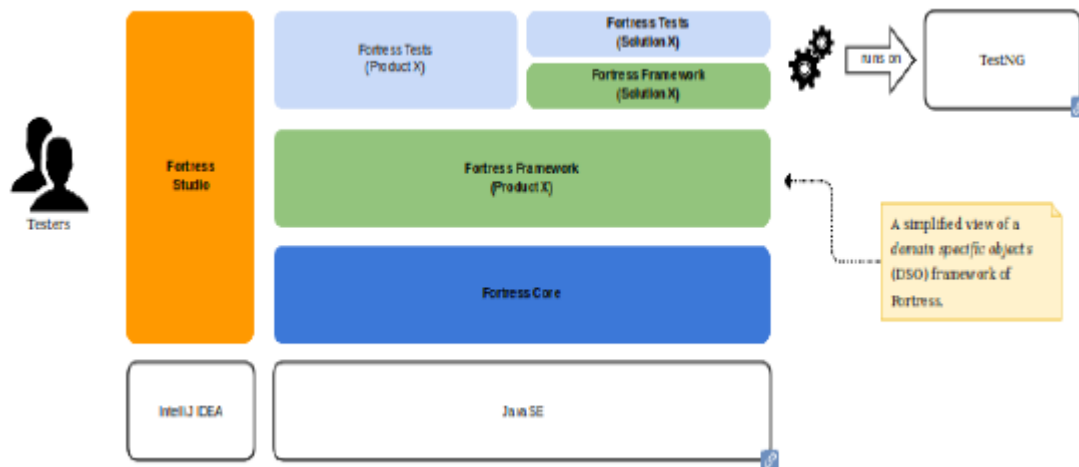


Figure 2.1: Fortress Overview

2.3 SYSGUARD

SysGuard is an in house built software that is used by LSEG Technology systems to do application operations and monitoring the systems via front end. SysGuard manual can be accessed through SysGuard front-end too. SysGuard is responsible for monitoring the health of machines and processes, providing fault tolerance capabilities, starting/stopping processes and scripts and maintaining system configuration, managing messages and statistics, etc. It is like a task manager used within LSEG Technology technology stack.

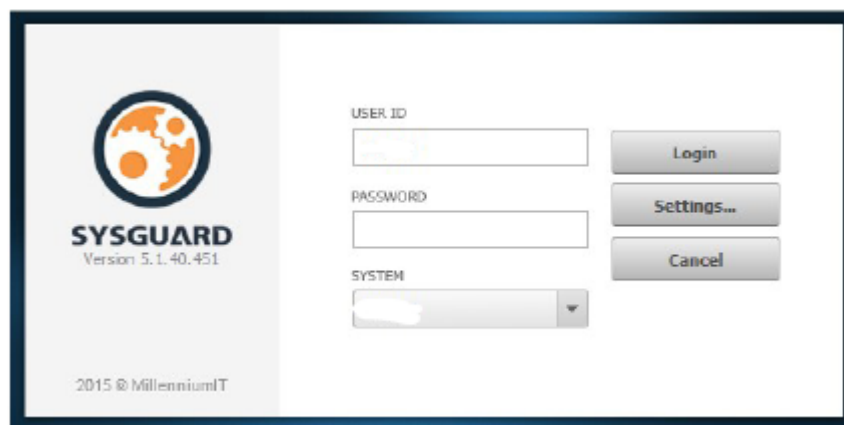


Figure 2.2: SysGuard Login Window

When SysGuard is started, the user has to enter user id which is created previously, usually this is a common one. Login will be verified with the user level and granted access to the system. Before login to SysGuard, an environment is needed to be configured, once configured the environment, user can log in to the front end.

When login to the SysGuard application, you are first directed to the main SysGuard workspace. The interface contains several elements such as the main menu, toolbar, SysGuard home, Message search pane and Status Bar.

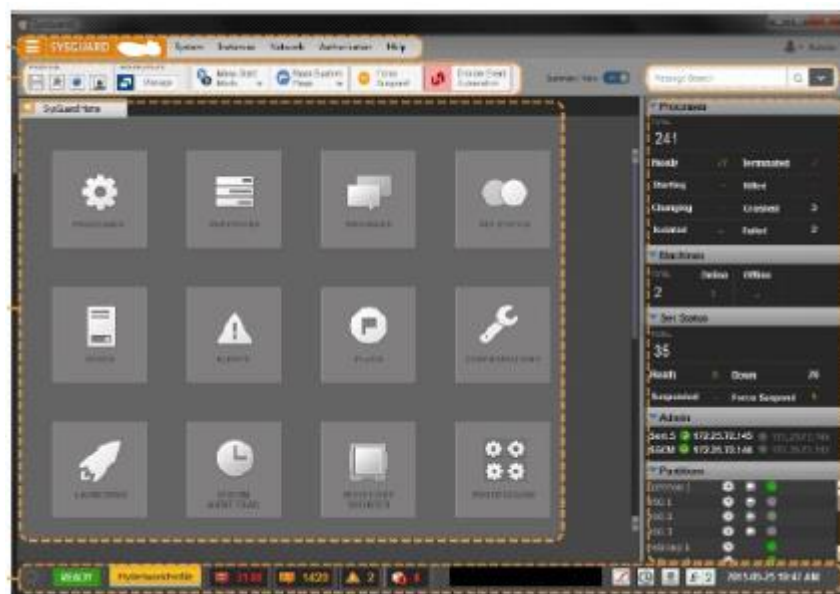


Figure 2.3: SysGuard Main Workspace

Processes tab lists the online processes running in the environment with their status like Ready, Terminated, Failed, Starting, Crashed, etc. Through Launchpad pre-setup startup sequences can be executed. SysGuard process view can be used to view the up and running processes in a more detailed way with memory usage, process name, rank and the machine where the process is running.

Name	Status	Rank	Set Rank	PID	Machine	Errors	Warns
Common:1:1:AuthServer:1	READY	1	1	5,273	Machine 1 (172.25.92.139)		1
Common:1:1:AuthServer:2	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:CheckPointer:1	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:CheckPointer:2	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:CppMockGateway:1	NOT RUNNING				Machine 12 (172.25.92.139)		
Common:1:1:CppMockGateway:2	NOT RUNNING				Machine 16 (172.25.92.139)		
Common:1:1:CppMockGatewayThree:1	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:CppMockGatewayTwo:1	NOT RUNNING				Machine 15 (172.25.92.139)		
Common:1:1:CppMockGatewayTwo:2	NOT RUNNING				Machine 20 (172.25.92.139)		
Common:1:1:JavaMockEventListener:1	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:JavaMockGateway:1	NOT RUNNING				Machine 10 (172.25.92.139)		
Common:1:1:JavaMockGateway:2	NOT RUNNING				Machine 15 (172.25.92.139)		
Common:1:1:LoaderAgent:1	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:LoaderTest:1	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:RefDataFED:1	READY	1	1	5,439	Machine 1 (172.25.92.139)		
Common:1:1:RefDataFED:2	NOT RUNNING				Machine 1 (172.25.92.139)		
Common:1:1:RefDataManager:1	READY	1	1	5,267	Machine 1 (172.25.92.139)	4	8
Common:1:1:RefDataManager:2	NOT RUNNING				Machine 1 (172.25.92.139)		

Figure 2.4: SysGuard Process View

SysGuard consists of several components as follows.

- SysGuard Serendib Life Support (SerLS)
- SysGuard Agent (SGA)
- SysGuard Configuration Manager (SGCM)
- Global Stat Daemon (GSD)
- SysGuard Interface (SGI)
- SysGuard Front-End

SerLS is the main component that is responsible for monitoring and managing LSEG Technology systems. SGA is the daemon that resides in each machine. It is responsible for starting, stopping and monitoring the statuses of processes and scripts on that machine. SGCM is the main process that is responsible for handling front-ends and database operations. GSD resides in each machine which is responsible for monitoring the machine CPU memory and disk usage. SGI is a library component attached to each of the processes which are managed and monitored by SysGuard. The SysGuard front

end is the component that is used by operators to issue commands and monitor LSEG Technology's systems.

2.4 SERENDIB MESSAGE PROTOCOL

Serendib is a messaging middleware that can be used to develop distributed applications in a heterogeneous environment. Serendib provides services by providing seamless communication between process components. Such services can be described as follows.

- Support for network transport.
- Data representation across multiple Operating Systems and Hardware Platforms.

It optimizes the network usage and application performance of the systems. It allows applications to discover servers and message formats at run-time making application and network changes painless. It is built on top of TCP/UDP and InfiniBand (IB).

Serendib is more popular among the LSEG Technology systems because Automated Trading Systems (ATS) are heavily distributed and so, each component is communicating with one or more other components in the system. Each component has a unique Service ID (SID) to identify each service and processes communicate by sending data. Serendib defines a message protocol that all senders and receivers can share.

There is a Serendib registry service which is a central place that keeps the information about which component is running on which machine, what are the ports used, IPs of the machines, what the communication protocol is (TCP/UDP/IB).

2.5 BAMBOO AUTOMATION

Bamboo is a continuous integration server that can be used to automate the release management for a software application, creating a continuous delivery pipeline. Continuous Integration is a software development methodology in which a build, unit tests, and integration tests are performed, or triggered, whenever code is committed to the repository, to ensure that new changes integrate well into the existing codebase. Integration builds provide early 'fail fast' feedback on the quality of new changes.

Release management describes the steps that are typically performed to release a software application, including building and functional testing, tagging releases, assigning versions, and deploying and activating the new version in production. This will be described later in this chapter.

The basic process for continuous integration with Bamboo is as follows.

Build > Test > Deploy

“Build” is the act of compiling a full build in one or more target platforms. “Test” is the stage of running unit tests. “Deploy” stage is the deployment of the system to development, QA environments.

As the compilers and test machines are separate, the build stage for MIT needs an additional step to transfer the binaries from the build machines to the test machine. Therefore, the process for MIT is

Build > Transfer > Test > Deploy

Basic building blocks of Bamboo are project, plan, stage, job, and task. The top-level container in Bamboo is the project. A project can contain one or more related build plans. A build plan consists of one or more stages that map to the chain described above. A stage contains one or more jobs which in turn consists of one or more tasks.

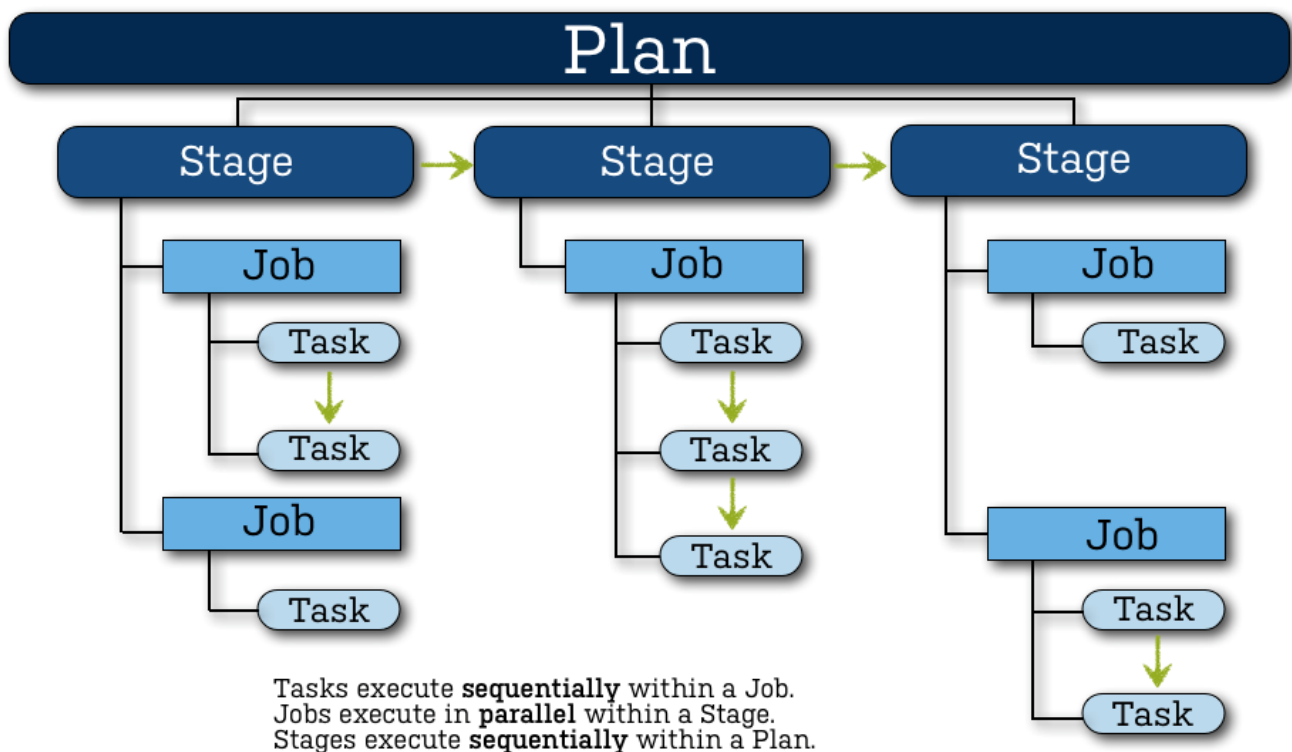


Figure 2.5: Bamboo Plan Anatomy

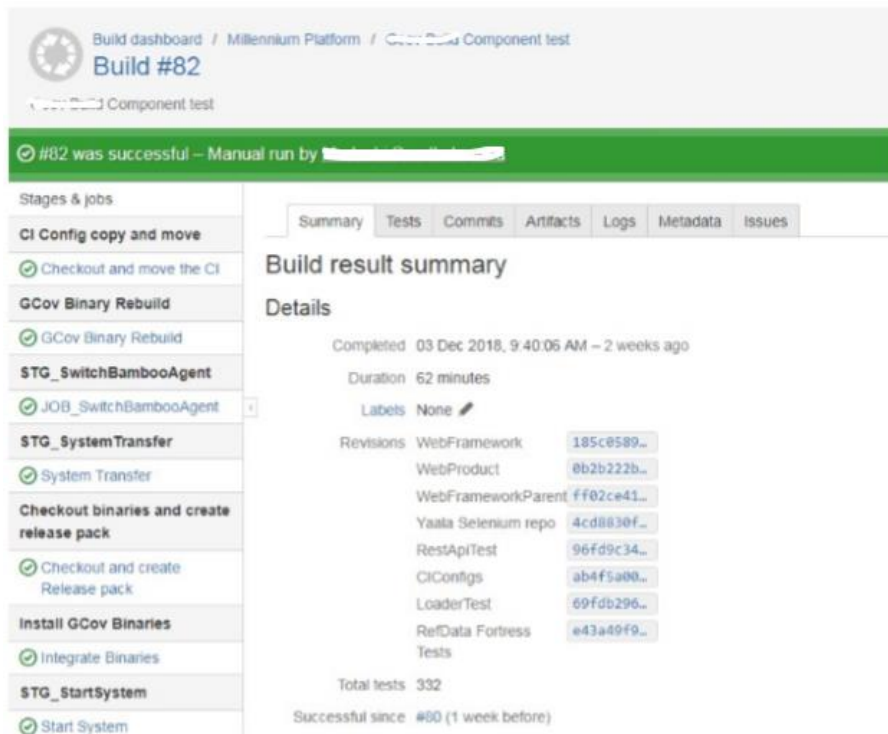


Figure 2.6: Bamboo Dashboard

On the technical aspects, Bamboo has its bamboo agents who can build plans. The capabilities of these bamboo agents differ from each other. Therefore the matching bamboo agent should be selected in order for the successful building of a plan. Generally in MIT Bamboo plans are run on two different environments. One environment is known as the “System Environment” and the “Build Environment”. The System Environment is where the Exchange system is running and the Build Environment is where the code is checkout from the repository and tests are executed.

2.6 YAAN FRONT END

Yaan is a web UI development framework that performs market activities through the front end. Through this front end trading day and also different markets can be started, user activities like logging in, logging out can be tracked and new reference data can be uploaded, etc. This process can be seen in SysGuard as the Yaan Process. For most of the test executions, this component is heavily used for monitoring as well as for functional requirements.

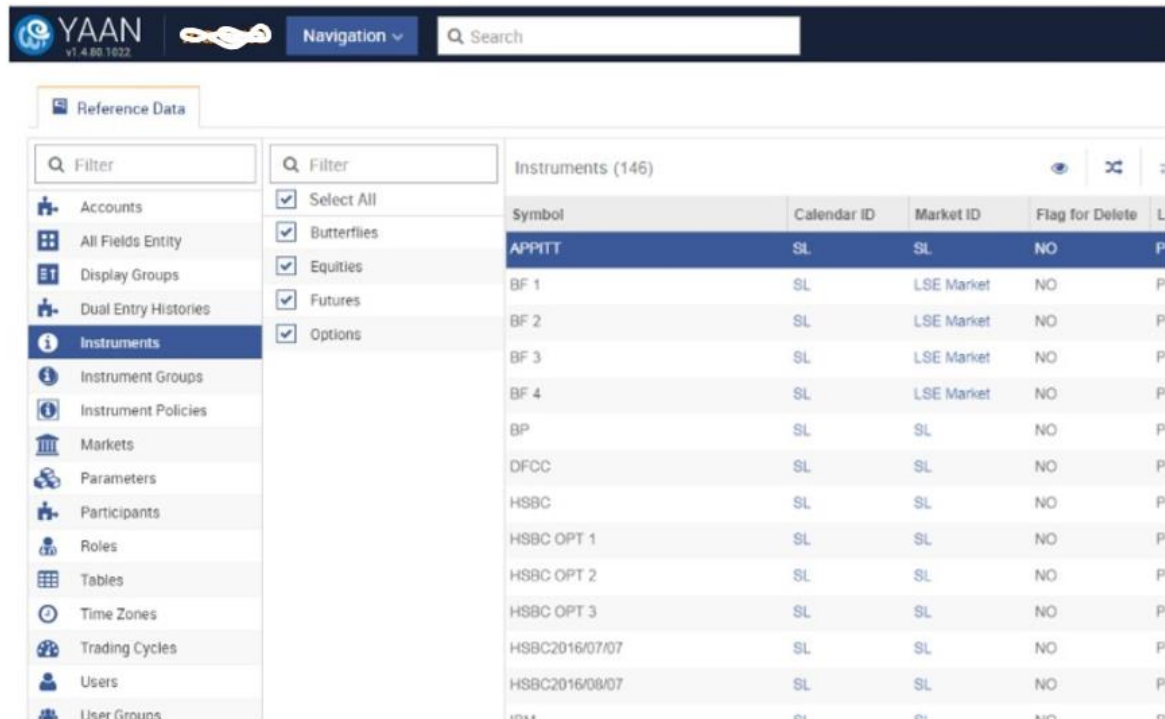


Figure 2.7: Yaan Front End

2.7 BEHAVIOUR DRIVEN DEVELOPMENT

BDD is a Software Development Methodology that focuses on the business behaviors of the product/solution. It encourages collaboration between business owners and the software development team and ensures that the delivered software contributes directly towards business value. In BDD first user acceptance tests are defined in collaboration with the business/product owners and then these tests are converted into automated test cases written in a preferred programming language like Java.

The main keywords associated with BDD used in MIT can be summarized as follows.

- Feature

All acceptance criteria related to a story will be constructed under a feature in Fortress.

- Background

Background contains specific actions that will be carried out prior to each scenario. Global Background, Epic Background, and Story Background are the three main types of background utilized.

- Scenario

A Scenario defines the acceptance criteria that will be converted into automated test cases by MIT using a language parser. (Gherkin)

- Dictionaries

For each Gateway interface (In MillenniumIT Exchange Product there are C++ components developed for different tasks like order matching, client handling, message parsing, etc. Each of these components is referred as a Gateway) there is a separate dictionary file defining message templates, field names and enumeration values within XML documents.

- Message Templates

Message templates store the default values for each message sent via gateway interfaces. When a scenario doesn't include a specifically required field fortress will fetch the default value from the message template and send it to the system.

- Validation Profiles

Validation Profiles will be used to test certain test fields by default.

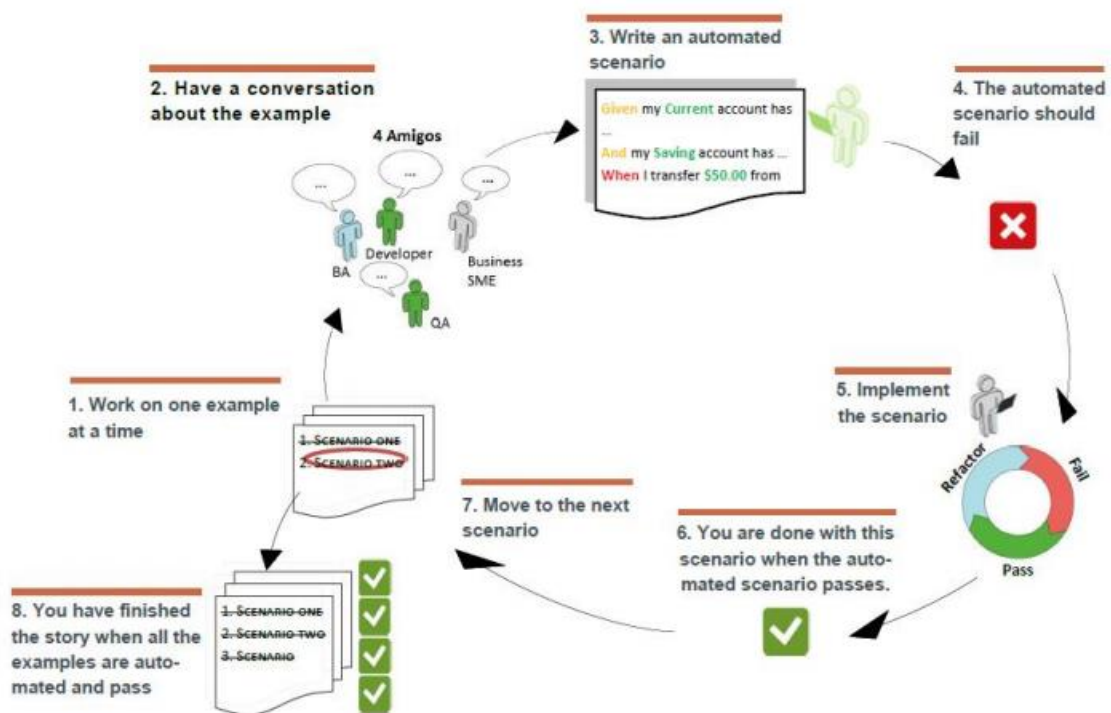


Figure 2.8: BDD Process in MIT

2.8 NEXUS REPOSITORY MANAGER

Nexus is a repository manager that allows managing project dependencies. When we build a project artifact to Nexus, then it becomes available for other developers. So it makes easy to distribute software. Nexus maintains two separate repositories for snapshots and releases. Snapshots are the versions that have not been released and they might get an update as well as the releases are the stable versions

The screenshot displays the Nexus Repository Manager web interface. The top header features the 'millennium' logo and the slogan 'INNOVATION IS POWER'. The left sidebar contains navigation links for 'Artifact Search', 'Views/Repositories', and 'Help'. The main content area is titled 'Repositories' and shows a table of 'User Managed Repositories'.

Repository	Type	Health Check	Format	Policy	Repository Status
Public Repositories					
3rd party	hosted	ANALYZE	maven2	Release	In Service
Apache Snapshots	proxy	ANALYZE	maven2	Snapshot	In Service
Atlassian 3rd Party	proxy	0 6	maven2	Release	In Service
Atlassian Public	proxy	250 336	maven2	Release	In Service
Central	proxy	626 1028	maven2	Release	In Service
Central M1 shadow	virtual	ANALYZE	maven1	Release	In Service
JBoss Enterprise Maven Repo...	proxy	2 9	maven2	Release	In Service
npmjs Registry	proxy	2 4	npm		In Service
Releases	hosted	ANALYZE	maven2	Release	In Service
Snapshots	hosted	ANALYZE	maven2	Snapshot	In Service
Thirdparty Raw	hosted	ANALYZE	site		In Service
Vaadin Addons	proxy	0 0	maven2	Release	In Service

Below the table, a 'Releases' section is expanded, showing a 'Browse Storage' view with a folder hierarchy: Releases > com > de > org > archetype-catalog.xml. A red box highlights this hierarchy.

Red annotations on the screenshot include:

- An arrow pointing from the 'Releases' and 'Snapshots' rows to the text: 'Separate repositories for releases and snapshots'.
- Text below the 'Releases' section: 'Artifact can be found in this folder hierarchy. e.g.: com->millenniumit->fortress-core>Version'.

Figure 2.9: Nexus Repository Manager

2.9 COMMON TECHNOLOGIES

2.9.1 IntelliJ IDEA Ultimate

IntelliJ IDEA Ultimate Edition, Java development IDE, is an extended version from its Community Edition. Since working on commercial and much in-built software at MIT every developer uses a licensed version. It is used with in-house built plugins like Fortress Studio as well as plugins like Cucumber. This IDE provides many unique features such as cross-technology refactoring, dependency analysis, code duplicate search, code coverage, integration with all popular version control systems, and more. Cucumber plugin is used to run BDD tests. Fortress Studio plugin offers specific functionalities like LogViewer to be used when test execution using Fortress Framework.

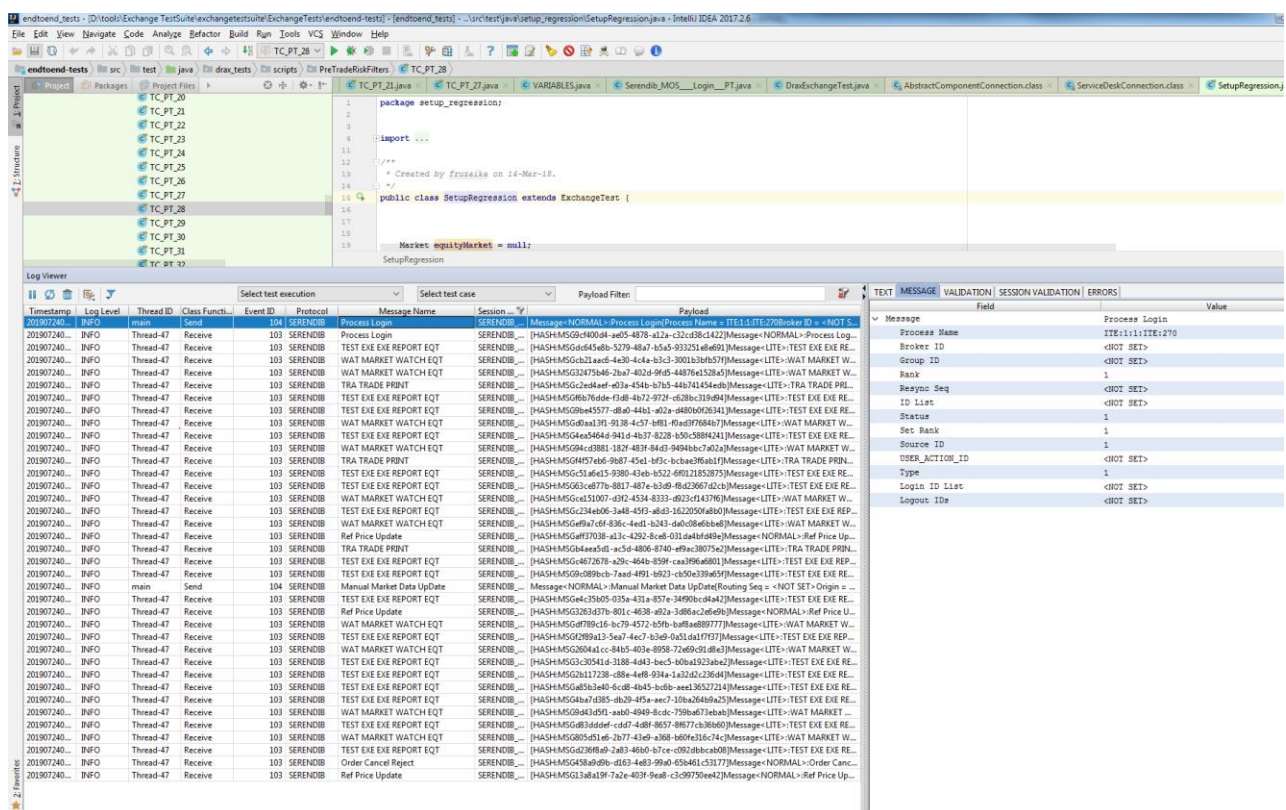


Figure 2.10: LogViewer from Fortress Studio Plugin

2.9.2 Apache Maven

Apache Maven is a software project management tool. Maven uses Convention over Configuration, which means developers are not required to create a build process themselves. Developers do not have to mention each and every configuration detail. Maven provides sensible default behavior for projects. When a Maven project is created, Maven creates a default project structure. The

developer is only required to place files accordingly and he/she need not define any configuration in pom.xml. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting, and documentation from a central piece of information. Using maven we can build and manage any Java-based project. Fortress projects are build using maven.

2.9.3 JIRA Ticketing System

Jira ticketing system is used for issue tracking while developing the project. Jira is an Atlassian product that can be used for bug tracking and agile project management. Jira helps to create stories and issues, plan sprints and distribute tasks across the software team as well as it helps to Prioritize and discuss the team's work in full context with complete visibility.

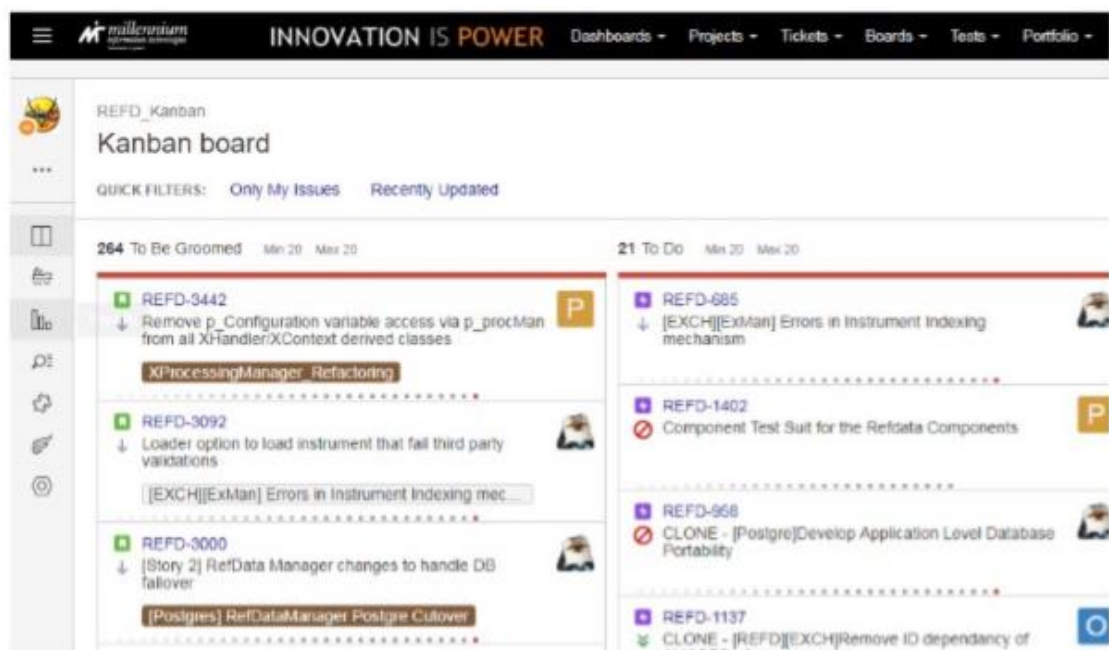


Figure 2.11: JIRA Dashboard

Every requirement of our projects was tracked via an Epic Jira ticket. A ticket consists of all the details about the issue. Generally PO and Scrum Master are the ones who are dealing with JIRA. In Scrum meetings, they assign each and everyone in the team the tasks for a project allocating a specific time and a unique ticket ID to track and record their progression throughout a sprint of 2 weeks.

2.9.4 Xshell & Millennium System Manager

XShell is used to connect to the environment. After set up and connected to the environment, the following details will display when entering the 'sy' short for support/sysman command which includes all the details about the system. When configuring bamboo plans Xshell is very useful to connect with multiple environments at once. Most of the time environment controlling is done through Xshell. Getting system transfer to an environment, starting system, stopping system, cleaning system are done through Millennium System Manager.

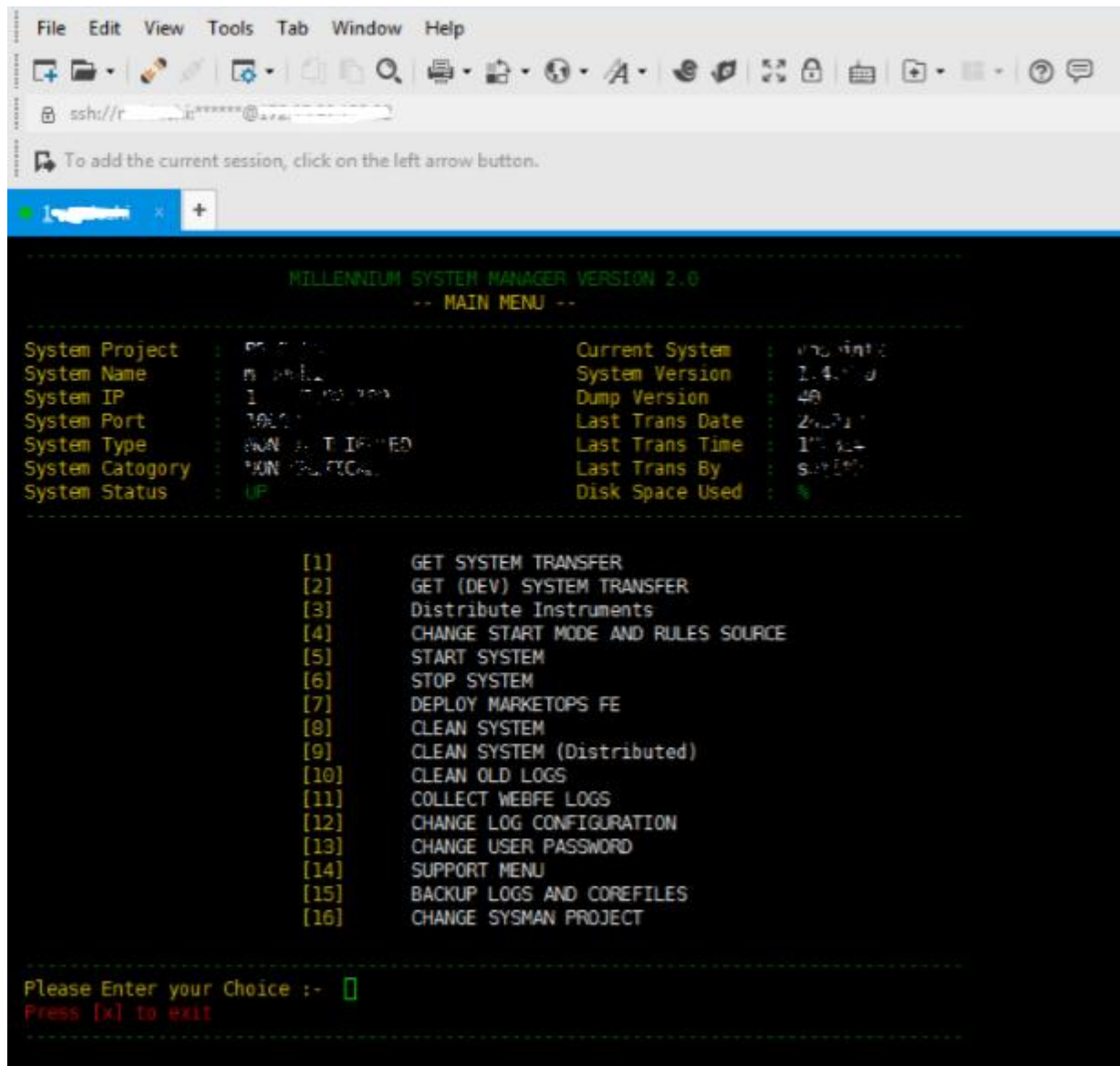


Figure 2.12: Xshell and Millennium System Manager

2.9.5 Git & BitBucket

Git is used for version controlling in Fortress. There are three main branches master branch, develop branch and release branch. All the changes in fortress platform repository were committed and pushed to the develop branch as well as the changes of fortress test suites were done to the master branch. When the development branch is ready to release, merge develop branch to release branch.

I followed a standard way to write git commits with the Jira Identifier while working on the project. Before merging our changes to the relevant branch, we need to create a pull request and need to request for the code review to our senior

members of the team. After approving the pull request, we can merge it to the relevant branch and otherwise we need to do code review changes before merging. The codebase is named as “stash”. Usually, Gitbash and Git plugin in IntelliJ is used.

Chapter 3

MY TRAINING EXPERIENCE

3.1 OVERVIEW

This chapter focuses on the work I did within my 20 weeks of internship at LSEG Technology. It includes the tasks I was assigned, how I approached each solution and problems faced along with how things learned in the Faculty is useful in those situations.

3.2 MY TRAINING PLACEMENT

The product suite of LSEG Technology has been well structured and layered to achieve the concept of LOB in the company as fulfill its product requirements. LOB is a concept that refers to a set of related products and services that serves a particular business transaction. This concept works like an assembly of software development within the organization. Different teams achieve different goals.

During my internship period, I was assigned to the OTC team. OTC team comes under the Third Party Trading System (3TS) which mainly focuses on third party products/solutions. Their main product outcome is the Millennium Exchange. My “line-manager” of my entire internship period was Eng. Chaminda Aluthgedara, Associate Software Architect. I had three supervisors for my internship period. Mr. Asitha Gunathilaka - Specialist Software Engineer, Eng. Chanaka Karunarathna - Software Engineer was my first two supervisors from my first scrum team “Team Scrumbledores” for my first 12 weeks of training. They were very helpful and they made my internship much more productive by sharing their knowledge and giving me tasks to practice. Then both of them were transferred to different teams and our team was renamed to “Team Gravity”. Since then my supervisor was Mr. Madava Viranjan - Tech Lead until my last day of internship.

3.3 INDUCTION AND ORIENTATION PROGRAM

The product suite of LSEG Technology has been well structured and layered to achieve the concept of LOB in the company as fulfill its product requirements. LOB is a concept that refers to a set of related products and services that serves particular business transactions. This concept works like an assembly of software development within the organization. Different teams achieve different goals.

During my internship period, I was assigned to the OTC team. OTC team comes under the Third Party Trading System (3TS) which mainly focuses on third party products/solutions. Their main product outcome is the Millennium Exchange. My “line-manager” for the entire internship period was Eng. Chaminda Aluthgedara, Associate Software Architect. I had three supervisors for my internship period. Mr. Asitha Gunathilaka – Specialist Software Engineer, Eng. Chanaka Karunarathna – Software Engineer was my first two supervisors from my first scrum team “Team Scrumbledores” for my first 12 weeks of training. They were very helpful and they made my internship much more productive by sharing their knowledge and giving me tasks to practice. Then both of them were transferred to other teams and our team was renamed to “Team Gravity”. Since then my supervisor was Mr. Madava Viranjan – Tech Lead until my last day of internship.

MY LEARNING PAGE

Welcome to your personal learning page. This page provides you with a summary of the courses in which you are enrolled or have completed as well as access reports detailing your learning activities. If you are a manager, you can also see learning reports for your team. The reports will populate as soon as you start completing self-learning activities and other courses.

Click on 'Manage my courses' to start or resume online courses, access course information or request cancellation.

MY COURSES

E-LEARNING COURSES
You have no courses in this category.

MY MANDATORY COURSES

- Information Security
- Conflicts of Interest
- Security Awareness for Privileged Users
- Preventing Bribery and Corruption
- Group Physical Security - 2018

[Manage my courses](#)

Completion status
Not started In progress Completed

MY ENROLLED COURSES
You have no courses in this category.

VIDEO COURSES
You have no courses in this category.

COURSES COMING UP
No activities.
[View all my courses](#)

MY MANDATORY COURSES
No activities.

Figure 3.1: My Learning Page

3.4 OPPORTUNITIES AND EXPERIENCE

During my entire internship period, I was able to gain a lot of industrial experiences. In the OTC team, all the team members are very friendly and they helped me a lot while working on projects.

Since I was working on a distributed system, I could learn a lot of high technical skills as well as I could learn how to design the best software by improving software quality. The main project I was engaged in had a unique domain and I got much knowledge of domain-related things also.

Engaging on a project in a company is more different than working on a project in the university because everything is well structured. While working on projects we have to participate in various types of meetings such as stand up meetings, scrum planning meetings, code reviews, retrospective meetings, etc. Everything is arranged via meetings and this was a new opportunity for me to enhance my professional skills and communication skills. Every communication is done via Emails. Sometimes I had to communicate with people who are in a higher position in the company via Emails. This was a good experience as well as a good opportunity to associate with people. There was a presentation session that was conducted for all the interns of the company. This was a good opportunity to improve our soft skills as well as communication skills.

Other than the official work, I participated in a lot of extra activities such as team dinner outs, annual trip, and some charity programs, etc. That was a good opportunity to gain a lot of experience in the training period.

I'm very confident about what I gained during my training period and these experiences and opportunities will be a great help for my future career in the software engineering industry.

3.5 TECHNICAL EXPOSURE

As an intern in the OTC team, I had the chance to engage in several tasks during my training period. From each of the tasks, I got the chance to learn new technologies. In the OTC team the interns are given tasks related to Fortress since 20 or 24 weeks of the training period is not sufficient to learn the Exchange System and the components and to deal with them. I was enrolled in the team for a java internship. I was the only one development intern for the team. My role was to provide side support for the tasks allocated to senior developers and perform testing tasks using Fortress. Listed below are the tasks that I was involved with during my training period.

- Native Server Creation and Testing in Fortress
- MiniATS (a mini version of the current exchange system) in Java.
- Writing support scripts.

- Creating, configuring and maintaining a few Bamboo plans on OTC pool.

As discussed in the previous chapter Fortress is a java test framework developed within MIT. Fortress requires each product or solution to keep its own framework to support the test projects. For Third-Party Trading Systems the Product is the Exchange. On top of that product, different Solutions are developed for requirements given by clients. Overall there is a layered architecture within the 3TS.

Layered Architecture

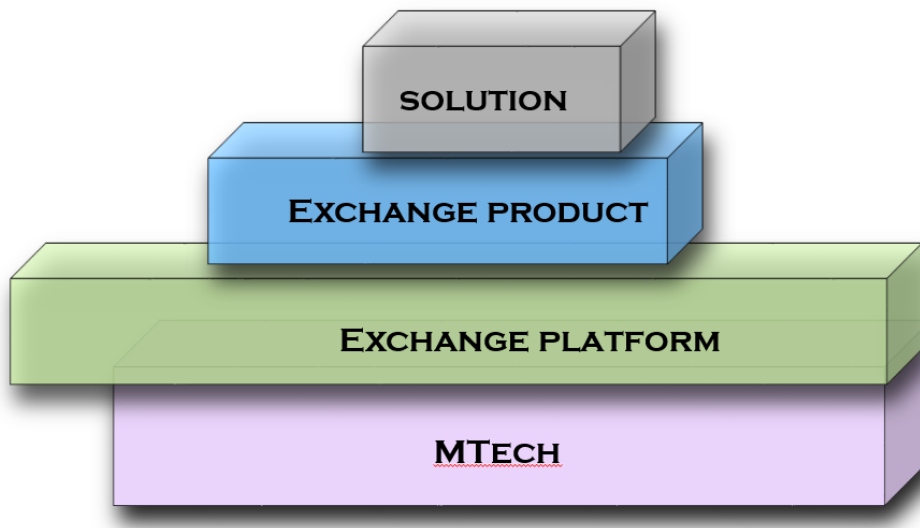


Figure 3.2: Layered Architecture in Millennium Exchange

Some solutions are built on top of the Exchange Platform and some others are on the Exchange Product. The products I worked with are ATOM and Aequitas. For these two products, two separate Fortress frameworks are maintained.

JSE QA Tests	JSE Dev Tests	ExPlus NSE QA Tests	ExPlus, NSE Dev Tests	NCDEX QA Tests	NCDEX Dev Tests	CVSA QA Tests	CVSA Dev Tests	BVL QA Tests	BVL Dev Tests
JSE Framework				NCDEX Framework		CVSA Framework		BVL Framework	
				ExPlus Framework					
Exchange Framework									
Fortress Core									

Figure 3.3: Fortress Framework Differentiation for each product/solution

3.5.1 Native Server Creation and Testing in Fortress

After my first week of Induction and Orientation, I was assigned a task to review the Native Server which was already completed about 85% and implement the missing functionalities required. Native Server was developed using Java and it was developed on top of the Fortress Exchange Framework. This Native Server was created for the testing process of the ATOM product and needed some implementations to match the requirements from that product. I had to spend 3 days reviewing the full code and comprehending the functionalities already implemented.

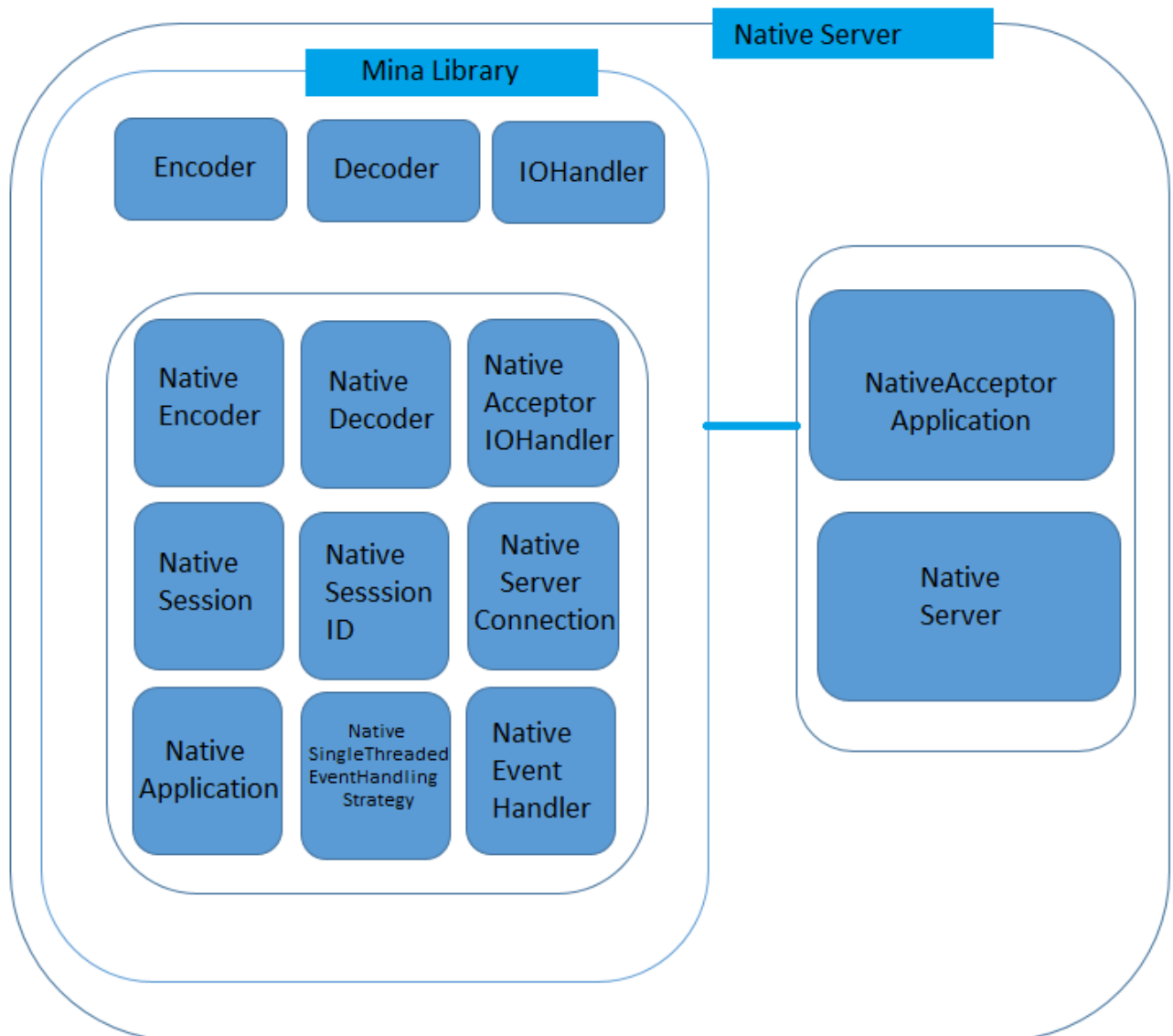


Figure 3.4: Overview of Native Server

Class Name	Functionality
NativeMessageEncoder	Convert Native Message to byte stream
NativeDecoder	Convert incoming messages from byte stream to Native protocol messages
NativeCodecFactory	Register Encode and Decode classes
NativeAcceptorIoHandler	Handle input messages
NativeSession	Session class which stores connections Send heartbeat messages
NativeServerConnection	Maintain received message queue Send native messages to native clients
NativeSessionID	Create unique ID for each session
NativeSingleThreadedEventHandlingStrategy	Create and handle threads for each accepted sessions
NativeAcceptorApplication	Handle admin and application messages
NativeServer	Create native servers

Table 3.1: Native Server Classes and Functionalities

- Implement sending Heartbeat messages for a live connection and to disconnect from Native Server-side if Heartbeats missed from connected gateways.

Heartbeat message is a message type defined in the Exchange System to signal the liveness of a specific connection. When a gateway connects successfully to a server or another gateway Heartbeat messages are sent. These messages are sent on a different channel than on the channel all other functional messages are sent. This happens in a defined time interval. In Native Server, this functionality was not implemented. First of all, I created a method that can send Heartbeat messages according to a predefined time given in the newtonproperties.xml file (Settings file for Fortress Framework). Then to create Heartbeat messages I created a separate Singleton class called OneTimeMessages and within that, I created the Heartbeat message according to Native Protocol Dictionary (It contains all message ids, names, fields, field lengths for the Native protocol). The reason for creating a Singleton class was that Heartbeat messages are unique light messages that contain no field data or payload. It is filtered from its message-id. Therefore there is no reason for creating heartbeat messages every time it has to be sent. Through Singleton Class message is created only once. Then the created message is sent according to a predefined time interval for successfully connected connections. On the other hand, functionality was

needed to disconnect a connection if the Heartbeat message is not received to Native Server. Since the heartbeat messages are received in a separate channel this task was easy. Received Heartbeat messages are checked for a time interval and if it happened 5 times continuously the connection is broke from the Native Server.

- Sending a Logon Reply message after gateways successfully connect with server

When a gateway/client (source component) connects to any component within the exchange system first of all the Logon message is sent. Then the receiving end must filter the Logon message and make the connection for the user included in the Logon message payload. After that, the receiving component must send a Logon Response message stating the connection id and other necessary data back to the source component.

In the implementation Logon Reply message was also created using the `OneTimeMessages` class according to Native protocol dictionary because this message is also needed only once. Then the message was sent using methods already implemented within `NativeAcceptorIoHandler` class.

- Writing unit tests to test the functions implemented.

After completing the implementation unit tests had to be written for the functions I implemented and also for some other functions previously implemented. Unit tests were written using `NativeGatewayClient` which is already implemented within Fortress Exchange framework. Unit tests were written to test sending and receiving different kinds of predefined messages like Logon, LogonResponse, Logout, NewOrder, NewQuote, etc.

3.5.2 MiniATS

This task was supervised by Eng. Chanaka Karunaratne. Its main objective was to get a brief idea of the current Exchange System functionalities. Every new joiner (permanent software engineers) are given a training task on developing a miniATS using the company's technology stack using C++, MFramework, Serendib, and MBase. But my task was to develop a simpler version using Java.

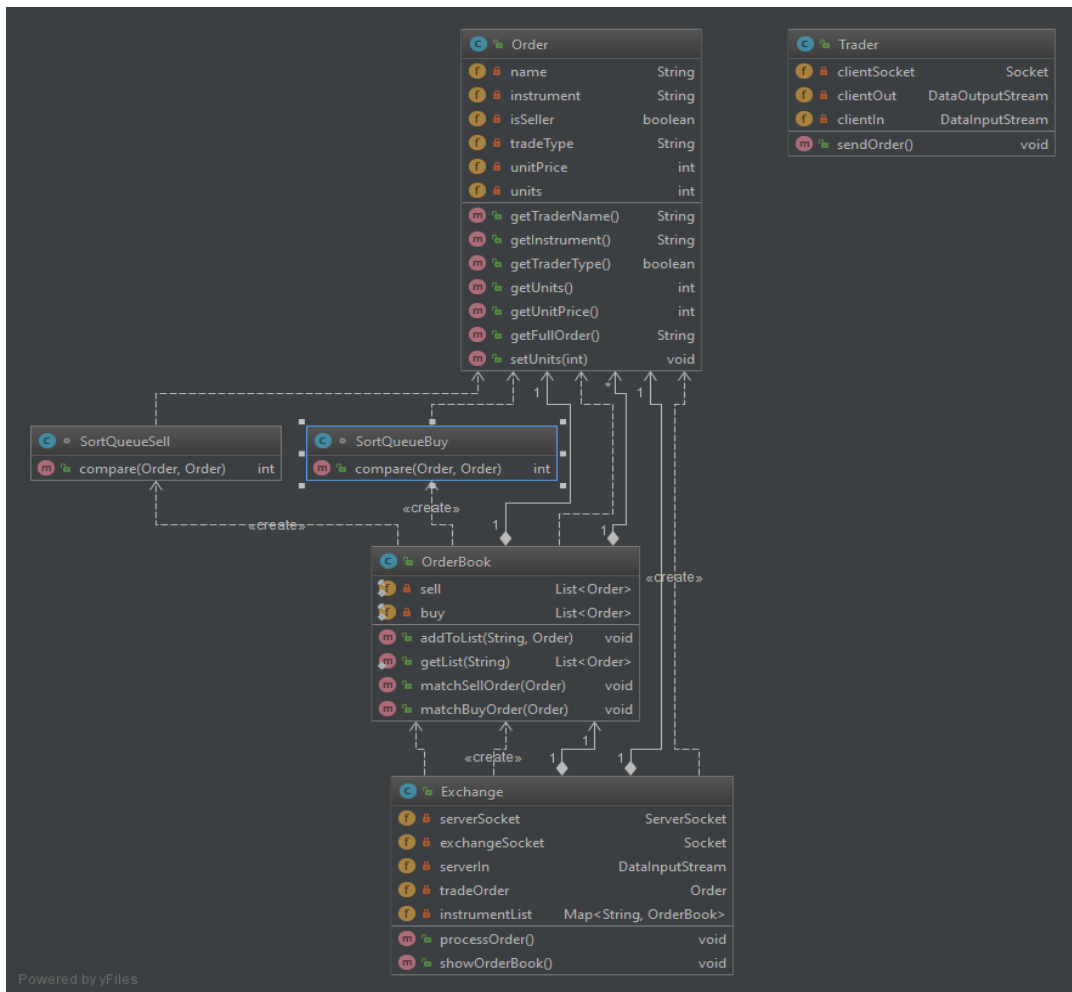


Figure 3.5: miniATS Class Diagram

Once a Trader place an order into the Exchange that order is processed considering the instrument type. For different instruments, there are different order books. Order books maintain instrument transactions in two main aspects - Buy and Sell. Once the order is directed to the specific order book after filtering through the instrument type/name then the order matching happens. In the current Exchange System, order matching is done differently on different order books. But for my implementation, every order is matched using the same logic and orders are sorted using price and time as the sorting criteria. The logic differs for buyers and sellers. For sell-side orders are sorted in a way that the order which has the lowest price offering is at the top. If two orders have the same price they are sorted according to the time they are placed. Order placed at the earliest takes precedence. For buy-side the order which has the highest buying price and earliest placement time takes the precedence. Finally to test the matching logic unit tests were written and tested.

3.5.3 Writing support scripts in python

I got to write two support scripts for the support team in the OTC team. One script was to update a pom file. It was used in a Bamboo plan to downgrade the Aequitas Fortress Exchange version and remove a dependency that causes to failing tests. For scripting, python is used with the ElementTree library.

```
import xml.etree.ElementTree as ET

namespace = {'ns': 'http://maven.apache.org/POM/4.0.0'}

# search parameters
matching_groupId = "org.apache.maven.plugins"
matching_artifactId = "maven-surefire-plugin"
corrected_version = "2.14"

groupId = "org.apache.maven.plugins"
artifactId = "maven-surefire-plugin"
version = "2.14"

pomFile = ET.parse("pom.xml")
root = pomFile.getroot()

dependencies = root.find('ns:dependencies', namespace)

for dependency in dependencies.findall('ns:dependency', namespace):
    groupId = dependency.find('ns:groupId', namespace)
    artifactId = dependency.find('ns:artifactId', namespace)
    currentVersion = dependency.find('ns:version', namespace)
    if( (groupId.text == matching_groupId) and (artifactId.text==matching_artifactId) ):
        print(currentVersion.text)
        currentVersion.text = corrected_version

build = root.find('ns:build', namespace)
plugins = build.find('ns:plugins', namespace)

for plugin in plugins.findall('ns:plugin', namespace):
    groupId = plugin.find('ns:groupId', namespace)
    artifactId = plugin.find('ns:artifactId', namespace)
    currentVersion = plugin.find('ns:version', namespace)
    configuration = plugin.find('ns:configuration', namespace)
    if ((groupId.text == groupId) and (artifactId.text == artifactId)):
        print(currentVersion.text)
        currentVersion.text = version
        plugin.remove(configuration)

pomFile.write("pom1.xml", default_namespace='http://maven.apache.org/POM/4.0.0')
```

Figure 3.6: Changepom.py script

The second script was written to compare some specific tables from the client's Golden Copy and our Integration. My task was to compare two CSV files generated out of two tables from Integration and Gold Copy and look for changed values for each field and additional entries on both sides. Each table had four columns. My approach was to compare two files interchangeably and write the difference

to a .xls (Excel) file including the compared line against difference in another file. Two .xls files were generated for changes in client-side and local. This was used in Aequitas Upgrade.

3.5.4 Creating, Configuring and Maintaining Bamboo Plans in OTC Pool

As mentioned in Chapter 2 Bamboo is used as the main CI/CD tool at MIT. I was assigned several tasks related to Bamboo plans in several sprints within my 20 weeks. All of the Bamboo plans were maintained under the OTC pool.

- Configuring Aequitas Bamboo plans to show test results on Bamboo Dashboard and scheduling all tests to run once in a week.

There was a problem with Aequitas bamboo plans not showing test execution results on Bamboo Dashboard. I was handed over the task to study Bamboo and resolve this problem. After going through some Bamboo documentation and comparing similar Bamboo Plan schemes that have already implemented this functionality for differences I could find that in order to show test results the generated test execution reports have to be passed to Bamboo agent home. I used a TestNG test result parser for that.

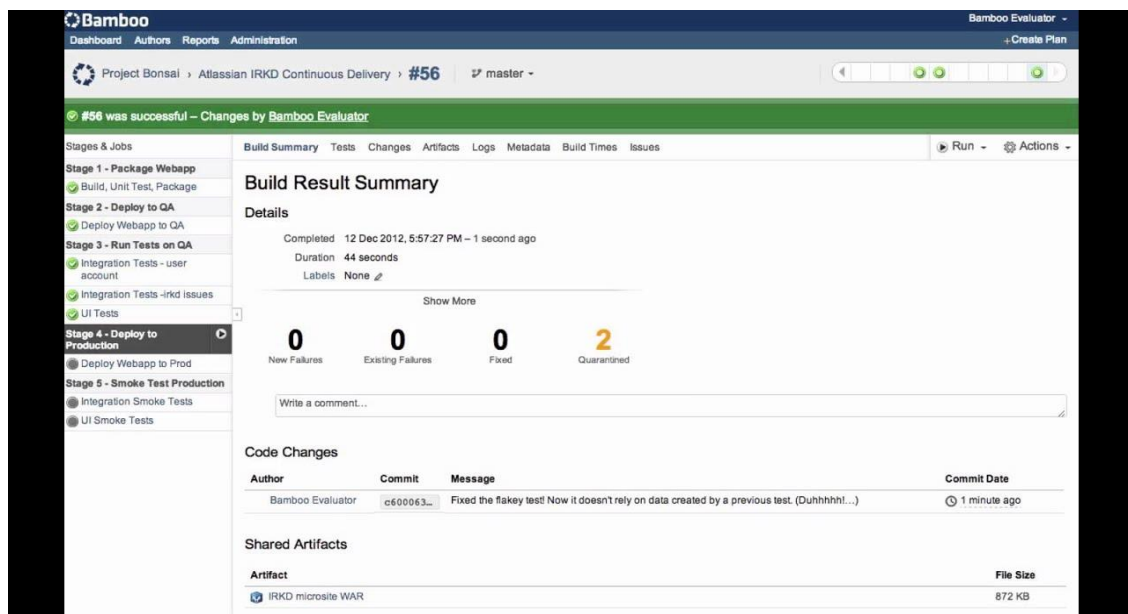


Figure 3.7: Bamboo plan configured to show test results on Bamboo Dashboard

In the Bamboo plans end to end tests (BDD tests are written using Java) were executed using a Java Runner class. All these Runner classes are stored in one directory. Each Runner class named FeatureExecutorXXX.java, where XXX is a number was executed by an SSH Task in bamboo by moving to Runner class location and passing Runner Class name which is defined in Bamboo variables to a python script. This python script is placed at Runner classes location using a previous Bamboo Task. Then each of these Runner Class executes the Global Background, Background, and Tests respectively. The method used before was entering FeatureExecutor class name in the Bamboo variables and building manually for each time. My task was to automate the process.

My approach was to pass all FeatureExecutor class names in the Bamboo variable as a string separated by commas. Then using shell scripting the string is split and the number of classes in the string was calculated. Finally using the following modular arithmetic equation Next Executed FeatureExecutor Index was calculated.

$$\text{Next_FeatureExecutor_Index} = \text{Bamboo_Build_Number} \% \text{FeatureExecutor_Class_Count}$$

Suppose:

Bamboo_Build_Number = 27

FeatureExecutor_Class_Count = 3

Then,

Next_FeatureExecutor_Index = 0

According to my approach, the first string from the Executor Class names variable is executed. The executed FeatureExecutor class name is attached to the email report sent to the supervising personal after each build. This email sending functionality was already implemented. I only add the executed FeatureExecutor class name to the e-mail heading.

Finally for scheduling each of these Bamboo plans I studied cron expressions, a triggering option available in Bamboo, and came with an expression to execute all of the plans given to me once in a week.

Cron expression generator - Quartz

Use this online user interface to generate cron expressions based on the Quartz engine

The screenshot shows the 'Cron expression generator - Quartz' interface. It has tabs for 'Seconds', 'Minutes', 'Hours', 'Day', 'Month', and 'Year'. The 'Seconds' tab is active. Below the tabs, there are three radio button options: 'Every second', 'Every 1 second(s) starting at second 00', and 'Specific second (choose one or many)'. The 'Specific second' option is selected. Below this, there is a grid of checkboxes for seconds 00 through 59. The '00' checkbox is checked. At the bottom, there is an option 'Every second between second 00 and second 00'.

Figure 3.8: Online Cron Expression generator

- Creating a Bamboo plan for Aequis Upgrade

In my 17th week of training, I was assigned to create a Bamboo plan to execute the DRAX tests (Java tests automatically generated from Verifix files. Verifix is a commercial test execution tool.) for the Aequis Upgrade for the upgraded version. There were Bamboo plans for the old version. After creating the plan there was a problem of the plan taking too much time (approximately 9 hrs) compared to other DRAX tests and majority tests were failing. These tests were passed when manually run on a local machine. After analyzing the problem it was found that the System environment's Yaan process is not started. Then the Day Start and Market Start operations were not performed and all tests were failed. It was caused by the wrong argument passing within the SysGuard Process Start sequence. After correcting the issue plan was run completely within 1.5 hrs.

3.6 HOW USEFUL WAS THE KNOWLEDGE GAINED FROM THE FACULTY

Since LSEG Technology has its own customized technology stack most of the things were new things. But when it comes to software practices the basic knowledge provided by the faculty was very important. Most of the things were not completely strange to me. Things learned in the CO328 Software Engineering, CO322 Data Structures and Algorithms, CO225 Software Construction course modules were very helpful. Using the most suitable data structure based on analyzing the characteristics of the task and different data structures, using git as version control, writing unit tests and using integration tools like Travis and design patterns in software engineering was directly used in my work in MIT. These helped me to reduce time in my learning curve and learn more new things like shell scripting, python

scripting, working in remote environments, professional practices like time management, organizing work, etc.

Chapter 4

NON-TECHNICAL EXPOSURE

4.1 OVERVIEW

LSEG Technology organizes a number of events and fun activities for employees' welfare. 'MiClub' is an association of employees of LSEG Technology, that organize these events to get together with each other and have fun together. During my internship period, I was able to participate in a number of those events

4.2 ANNUAL TRIP

The Annual trip is a wonderful event which was organized by the 3TS team. This year, the annual trip is organized in SOS (Scrum of Scrums) wise and we participated in it with about 60 members and their families. That was a very enjoyable weekend we went to Weligama Bay Marriott Hotel. There were a lot of fun activities they have organized and we stayed there for two days. We checked in to the hotel at about 6 pm on Friday that week and checked out from it on Sunday evening. During that time I was able to collect a lot of memories and able to get a lot of experiences with my team. We involved in fun activities like playing different card games, surfing in the sea, pool games. We enjoyed singing songs, eating different kinds of delicious food. It was a great experience in my life.

4.3 MICARNIVAL 2019

MiCarnival was an event with a lot of fun activities for the entire family of the employees at LSEG Technology which was held at the Millennium IT ground. There were so many food stalls, popcorns, Candy Floss, Bouncy Castle, and all are free. There was a football tournament also. There were eight teams from Turing and Von Neumann and we enjoyed here well.

4.4 PRESENTATION SKILL SESSION

There is a presentation session for all the interns at LSEG Technology which was conducted by Project Managers for interns every Wednesday. This is a 30-minute session and all the interns should participate in that. One of the interns should do a presentation and others are free to ask questions.

CONCLUSION

Over the four chapters of this report, I describe the information about the training establishment, technical tools used in the training organization and overall experience of my internship period including the non-technical exposure received, new things learned and how knowledge from my faculty helped me in my work.

I selected LSEG Technology for my industrial training as my first choice because it has been popular in the software industry for two decades. I am greatly satisfied with the knowledge and experience that I have gained during my training period within LSEG Technology. Those things will be a great help for my future career to graduate as a professional software engineer.

As the technical exposure, I was able to get industry level software development experience. I was a member of a scrum team and I could actively participate in several types of meetings such as scrum planning meetings, retrospective meetings, daily stand up meetings, etc. In a daily scrum meeting, all the scrum team members discuss the daily work and in a scrum planning meeting, tasks are distributed among the team members for a sprint. “Retrospective” meeting is a meeting that discusses the work done during the last sprint and how much each one's satisfaction with the work he/she did. In each time I could get a lot of industrial experience that I never had before from the university projects because in university projects, which are mainly focused on gaining marks over the learning things.

During my training period, the most important thing was working with people and communicating with each other. From the date, since I joined LSEG Technology, I worked with the OTC team and I had to communicate with them. OTC team consists of members who are in several types of positions and I had to deal with them. Professionalism is needed while working on projects as well as informality is important while talking with people face to face. Sometimes I had to communicate with people who are in a higher position in the company while doing on projects. That was a good experience for me to know a lot of people.

Not only the technical exposures, but I was also able to gain a lot of non-technical industrial exposures during my internship period. I could participate in a lot of extra events which are organized by LSEG Technology. Soft skills are also very important in the modern world. When I was working at LSEG Technology, I had a chance to improve my soft skills because there were several sessions for that. They gave me a good experience other than the technical knowledge

This was a good opportunity to apply theories that I learned during the past three years and this will be a great practice for the final year. LSEG Technology was a great place for me to improve my skills because there are a lot of technologies to learn. An undergraduate doesn't have more technical

experience and has more theoretical knowledge. So, this is a great place to train if the undergraduate is willing to learn. People at LSEG Technology are very open and friendly. When I was in trouble or has a question regarding the project, they always help me to resolve them and they always willing to help us in both technical and non-technical problems.