AUTOMATED TESTING FRAMEWORK

A Practice School Report submitted to
Manipal Academy of Higher Education
in partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

Submitted by

Mansi Sharma

180905294

Under the guidance of

Hrituparn Alreja
Analyst
BlackRock

&

Dr Geetha Maiya
Professor, CSE
MIT,Manipal

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

<July> 2022

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

INSPIRED BY LIFE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

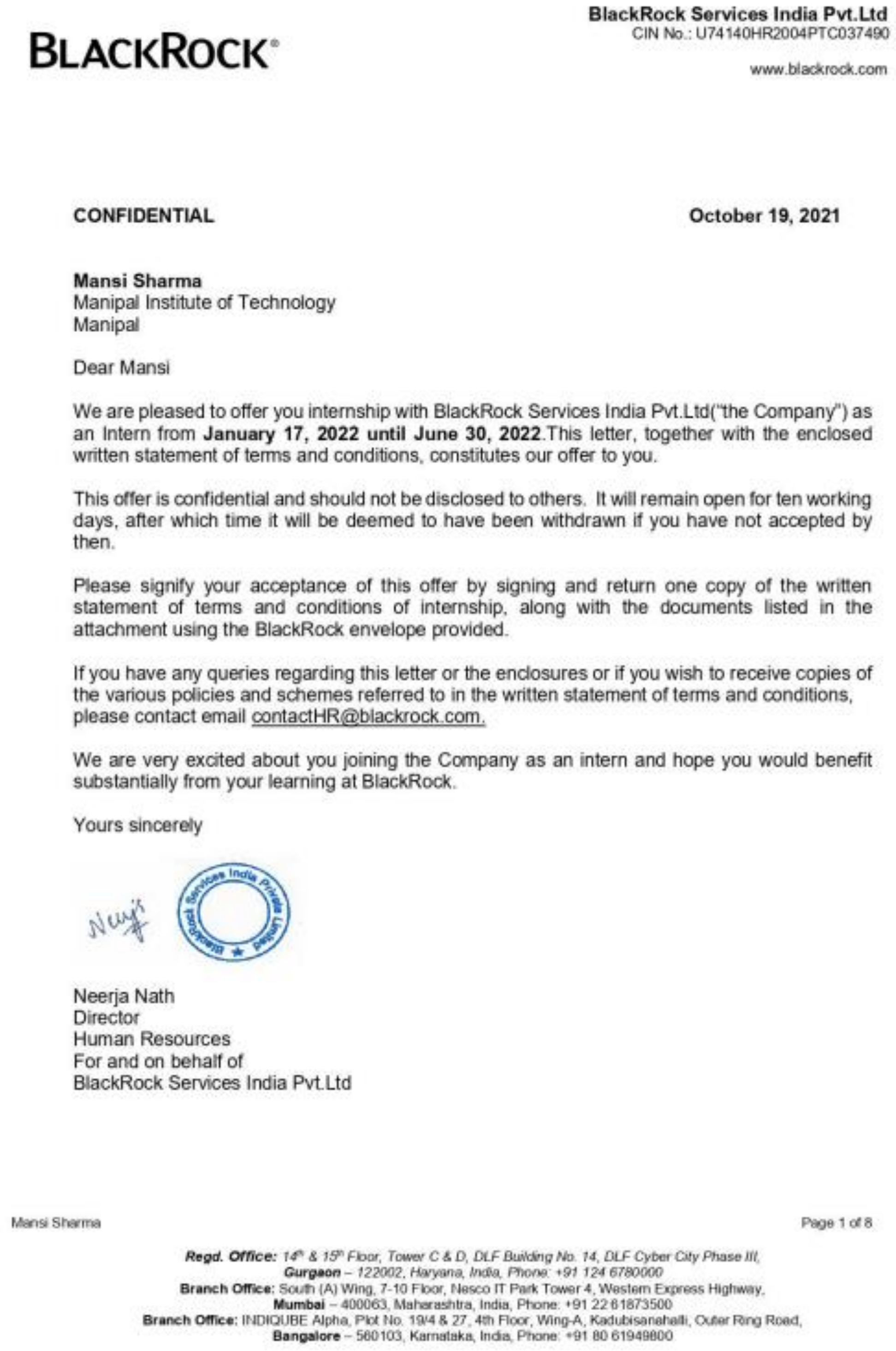
Manipal
07-07-2022

CERTIFICATE

This is to certify that the project titled **Automated Testing Framework** is a record of the bonafide work done by **Mansi Sharma** (*180905294*) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology (B.Tech.) in **COMPUTER SCIENCE & ENGINEERING** of Manipal Institute of Technology, Manipal, Karnataka, (A Constituent Institute of Manipal Academy of Higher Education), during the academic year 2021-2022.

Dr. Geetha Maiya
Professor, CSE
M.I.T, MANIPAL

Prof. Dr. Ashalatha Nayak
HOD, CSE Dept.
M.I.T, MANIPAL



BLACKROCK®

BlackRock Services India Pvt.Ltd
CIN No.: U74140HR2004PTC037490

www.blackrock.com

CONFIDENTIAL

October 19, 2021

Mansi Sharma
Manipal Institute of Technology
Manipal

Dear Mansi

We are pleased to offer you internship with BlackRock Services India Pvt.Ltd("the Company") as an Intern from **January 17, 2022 until June 30, 2022**.This letter, together with the enclosed written statement of terms and conditions, constitutes our offer to you.

This offer is confidential and should not be disclosed to others. It will remain open for ten working days, after which time it will be deemed to have been withdrawn if you have not accepted by then.

Please signify your acceptance of this offer by signing and return one copy of the written statement of terms and conditions of internship, along with the documents listed in the attachment using the BlackRock envelope provided.

If you have any queries regarding this letter or the enclosures or if you wish to receive copies of the various policies and schemes referred to in the written statement of terms and conditions, please contact email contactHR@blackrock.com.

We are very excited about you joining the Company as an intern and hope you would benefit substantially from your learning at BlackRock.

Yours sincerely

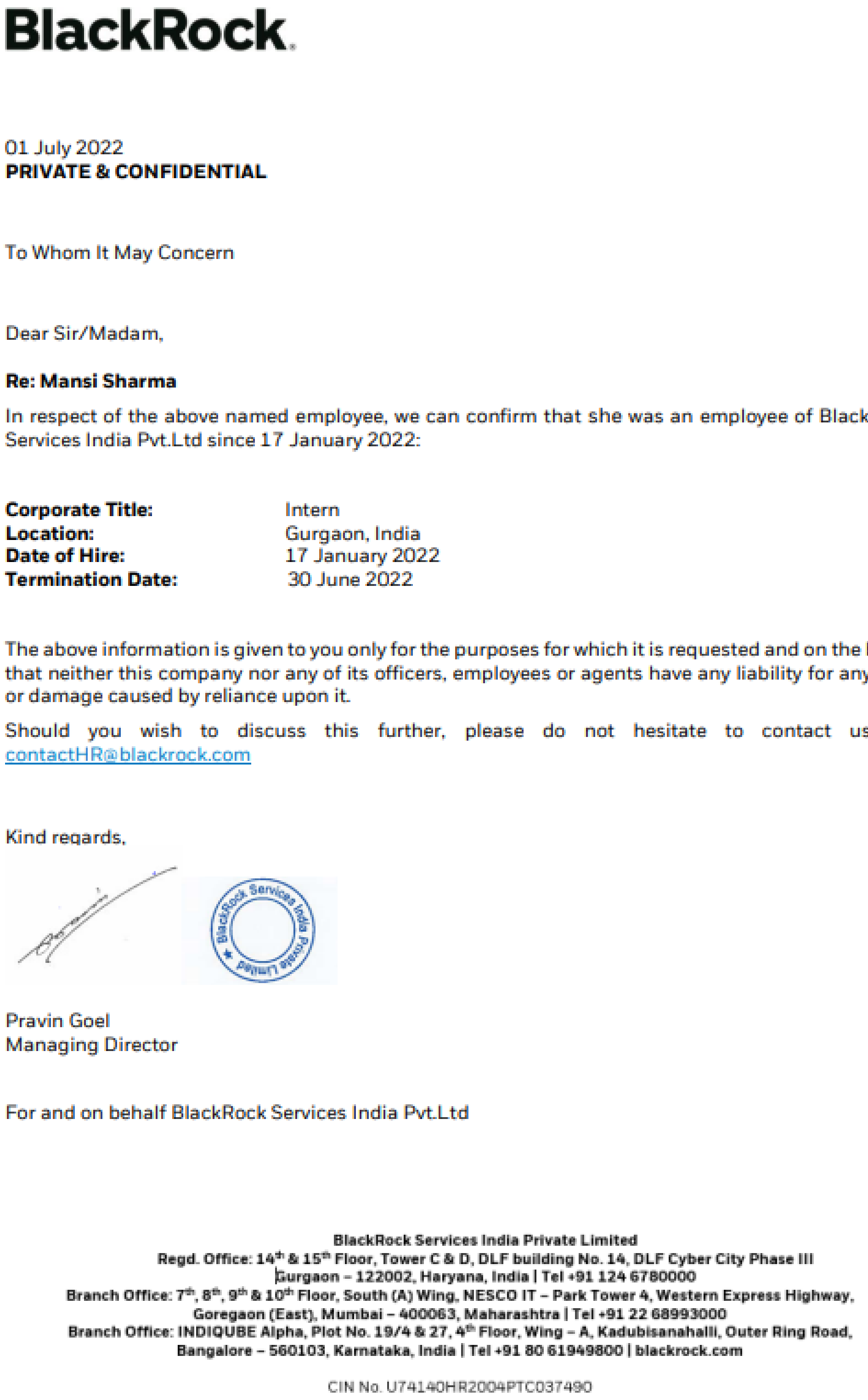
BlackRock Services India Private Limited ★

Neerja Nath
Director
Human Resources
For and on behalf of
BlackRock Services India Pvt.Ltd

Mansi Sharma

Page 1 of 8

Regd. Office: *14th & 15th Floor, Tower C & D, DLF Building No. 14, DLF Cyber City Phase III,*
Gurgaon *– 122002, Haryana, India, Phone: +91 124 6780000*
Branch Office: South (A) Wing, 7-10 Floor, Nesco IT Park Tower 4, Western Express Highway,
Mumbai – 400063, Maharashtra, India, Phone: +91 22 61873500
Branch Office: INDIQUBE Alpha, Plot No. 19/4 & 27, 4th Floor, Wing-A, Kadubisanahalli, Outer Ring Road,
Bangalore – 560103, Karnataka, India, Phone: +91 80 61949800



BlackRock.

01 July 2022
PRIVATE & CONFIDENTIAL

To Whom It May Concern

Dear Sir/Madam,

Re: Mansi Sharma

In respect of the above named employee, we can confirm that she was an employee of BlackRock Services India Pvt.Ltd since 17 January 2022:

Corporate Title:	Intern
Location:	Gurgaon, India
Date of Hire:	17 January 2022
Termination Date:	30 June 2022

The above information is given to you only for the purposes for which it is requested and on the basis that neither this company nor any of its officers, employees or agents have any liability for any loss or damage caused by reliance upon it.

Should you wish to discuss this further, please do not hesitate to contact us on contactHR@blackrock.com

Kind regards,

BlackRock Services India Private Limited

Pravin Goel
Managing Director

For and on behalf BlackRock Services India Pvt.Ltd

BlackRock Services India Private Limited
Regd. Office: 14th & 15th Floor, Tower C & D, DLF building No. 14, DLF Cyber City Phase III
Gurgaon – 122002, Haryana, India | Tel +91 124 6780000
Branch Office: 7th, 8th, 9th & 10th Floor, South (A) Wing, NESCO IT – Park Tower 4, Western Express Highway,
Goregaon (East), Mumbai – 400063, Maharashtra | Tel +91 22 68993000
Branch Office: INDIQUBE Alpha, Plot No. 19/4 & 27, 4th Floor, Wing – A, Kadubisanahalli, Outer Ring Road,
Bangalore – 560103, Karnataka, India | Tel +91 80 61949800 | blackrock.com

CIN No. U74140HR2004PTC037490

ACKNOWLEDGMENTS

I would like to thank the Director, Dept HOD Prof Dr. Ashalatha Nayak, Dr. Geetha Maiya, Mr. Hrituparn Alreja(Analyst at BlackRock), and Mrs. Sonam Midha (Vice President at BlackRock) for their constant help and support without whom the project wouldn’t have been completed successfully and on time. I would also like to thank my family and friends for their constant moral support.

ABSTRACT

Main goal of this application is to find a way to make testing the team’s software easier and more scalable. The application starts with the extraction of data from the Aladdin Dashboard and the file is stored in the local system in the form of a CSV File. This file consists of payloads(message) which are then sent to the GPException server for testing purposes. This server then sends a payload back in response, which is updated and checked with the user input. This newly updated payload is again sent till the testing is complete.

Table of Contents

			Page No
Acknowledgement			5
Abstract			6
List Of Figures			9
Chapter 1		**INTRODUCTION**	
	1.1-	Work Overview	10
	1.2	Team Overview	11
	1.3	Area Of work	11
	1.4	Motivation	11
	1.5	Objective	11
Chapter 2		**BACKGROUND THEORY**	
	2.1-	Developer’s Trainings	13
	2.4	Related Tools	13
	2.5	Green Package Process	20
	2.6	Data	20
	2.7	GP Production Environment	21
	2.8	WDI Support	22
Chapter 3		**METHODOLOGY**	
	3.1	Introduction	23
	3.2	Application Development Cycle	23
	3.3	DevOps Deployment	25
	3.4	Testing Environment	26
	3.5	Testing Framework	26
	3.6	Documentation	27
	3.7	Some Essential Technologies	27
	3.8	Implementation Details	29
	3.9	Product Management	30

Chapter 4	**RESULT ANALYSIS**	
4.1	**Observations and Findings**	**31**
4.2	**Challenges Faced**	**31**
Chapter 5	**CONCLUSION AND FUTURE SCOPE**	
5.1	Summary of the work	33
5.2	Significance of the results	33
5.3	Future Scope of work	33
REFERENCES		**34**
PROJECT DETAILS		**35**
PLAGIARISM REPORT		

LIST OF FIGURES

Figure No	Figure Title	Page No
1	Scope of Maven	18
2	Green Package Process	20
3	GP Production Cycle	21
4	Application Development Lifecycle	24
5	Azure DevOps in Parallel with Development	26

CHAPTER -1
INTRODUCTION

BlackRock is a world leader in providing institutional, intermediate, and private customers with asset management, risk management, and consulting services. Across equity, fixed income, cash management, alternative investments, real estate, advising services, and the market-beating iShares exchange-traded funds, BlackRock manages a total of US$6.5 trillion in assets. It provides risk management, strategic advising, and enterprise investment system services through BlackRock Solutions to a wide range of clients with assets totaling around US$13 trillion.

BlackRock is a genuinely global company that combines the advantages of global reach with local ties and services. They look after customers' assets across Europe, Asia, Australia, the Middle East, and Africa as well as North and South America. BlackRock, which has its headquarters in New York, has operations in 27 different countries and more than 14,900 skilled experts work there.

Corporate, public, union, and industry pension plans, governments, insurance firms, third-party mutual funds, endowments, foundations, charities, corporations, official institutions, sovereign wealth funds, banks, financial experts, and individuals from all over the world are among Blackrock's clients.

BlackRock has a significant presence in both institutional and retail channels and offers a broad range of fixed income, cash management, equities, and alternative investment products. By providing a variety of products, the company's clients are able to make use of the combined experience of all of its resources, including its investment and risk management experts and its highly advanced, integrated, proprietary analytical tools.

In order to help customers and portfolio managers navigate the global economy, markets, geopolitics, and long-term asset allocation, the BlackRock Investment Institute (BII) makes use of the firm's knowledge and produces unique research.

BlackRock's flagship product, Aladdin (Asset, Liability, Debt and Derivative Investment Network), is an operating system for money managers that aims to link the data, people, and technology required to manage money in real-time. To support educated decision-making, efficient risk management, successful trading, and operational scaling, the Aladdin platform integrates complete portfolio management, trading, and operations features on a single platform.

1.1 Work Overview

I had the opportunity to work with elite experts at the core of a prominent global investment management organization as a BlackRock intern. I was able to build a solid professional foundation in financial services with the aid of a protracted and rewarding training program, frequent speaker events, mentorship and feedback, and multiple growth possibilities. We underwent a challenging learning curve throughout the month-long training process that preceded the internship. The firm's ethics, compliance training, fundamentals of finance, corporate finance, accounting principles, fundamental and advanced excel, presentation skills, and an overview of the numerous databases the company uses were all taught to us. We may join our respective teams after the training session was through. I was given "Data Ops Engineering - Data Workflow" to work on.

1.2 Team Overview

To link our supply chain and operators with partner systems and tools covering financial modeling, accounting, and reporting, the Data Production Engineering team under DataOps Engineering is responsible for orchestrating data flow and delivery. The Data Production Engineering is in charge of scalable delivery, dependency and configuration management, automation, and optimization approaches for Aladdin data processing. To support production monitoring and the pursuit of continuous process and improvement, the team develops and owns its own instrumentation, predictive analytics, and reporting systems.

1.3Area of Work

The project will be in the domain of Data Workflow Engineering. We will use Java as well as BlackRock's internal libraries to establish database connections, execute SQL queries, logging, and exception handling.

Hardware and Software Requirements

- Java
- Apache Ignite
- No SQL DB
- Central Dashboard
- BlackRock's internal servers

1.4 Motivation

Blackrock receives data from its clients and vendors to perform the required analysis and generate the Green package report. The data is sent in files. The files are sent to BlackRock using Blackrock's FTP File transfer admin. The files and then transferred to the vendor directory.

This tool would run end-to-end testing for enhancements in GPExceptionServer, GPWorkflowServer, and GPNotes2Server. This would make the testing process a whole lot easier and loading scalable. Parallelism would be employed thus making it an efficient application and an improvement on the current use cases.

1.5Objective

- ✓ To develop a new Java application to run end-to-end testing for enhancements in GPExceptionServer, GPWorkflowServer, and GPNotes2Server
- ✓ To send a large number of signals in parallel to servers.
- ✓ To query actual results from servers cache
- ✓ To simulate different scenarios testing based on the use case of each server

- ✓ To achieve performance and load testing

CHAPTER – 2
BACKGROUND THEORY

Developer Training

Before taking up any project involving the development of software applications at BlackRock, one must go through the basic developer training in JAVA. The training is done in 3 stages, which are:

- **Aladdin Java Developer Certification (AJDC) Bronze**: This is a beginner-level assignment in which an overview is given, on how projects are developed at BlackRock and what is the software development process in the firm. At the end of this course, we were supposed to complete an assignment in which we had to create classes that will act as DAO (Data Access Objects) and write functions, according to a given problem statement.

- **AJDC Silver**: This is an intermediate-level assignment in which we had to build further upon the work done in AJDC Bronze. This assignment involved using Spring and Hibernate to create a JAVA-based server that will query a database table and serve the results to a client. We also had to perform testing of the server using the JUnit Testing module.

- **AJDC Gold**: This is an advanced level assignment in which we had to build further upon the work done in AJDC Silver, this assignment involved using the result given in the previous stage and displaying it on a local host for the user. We also had to perform testing of the front-end application.

For the above certification, we used Eclipse and also got a flavor of how to develop and deploy a project at BlackRock. The next step after the completion of the training was to get a good commands over server models and programming structures. For this, I went over BlackRock-specific WIKIs, which gave an in-depth description of how-to setup environments to create a project in the firm’s development environment, and tutorials that provided the right knowledge to create an application server using JAVA.

2.1 Microsoft Azure

Microsoft's public cloud computing platform is called Microsoft Azure, formerly known as Windows Azure. Numerous cloud services are offered by it, such as computation, analytics, storage, and networking. These services are available for users to select from while creating and scaling new apps or using the public cloud to operate already existing applications.

The Azure platform intends to assist organizations in managing difficulties and achieving their objectives. It is interoperable with open source technology and provides solutions that assist many industries, including e-commerce, banking, and a number of Fortune 500 organizations. This gives people the freedom to use the tools and technology of their choice. Azure provides four main types of cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS)

2.1.1 What is Microsoft Azure used for?

Microsoft Azure's use cases are quite varied since it comprises different service offerings. One of the most widely used applications of Microsoft Azure is the cloud-based running of virtual machines or containers. These computing resources may be used to host third-party applications, Windows Server services like Internet Information Services (IIS), DNS servers, or other infrastructure elements. Microsoft encourages the usage of alternative operating systems like Linux.

Azure is frequently utilized as a platform for cloud-based database hosting. Microsoft provides both relational and non-relational serverless databases, such as Azure SQL and NoSQL.

The platform is also commonly used for disaster recovery and backup. Many businesses utilize Azure storage as an archive to address their long-term data needs.

2.2 Java

Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible.

2.2.1 Java Terminology

Before learning Java, one must be familiar with these common terms of Java.

1. Java Virtual Machine(JVM): JVM is the common name for this. A program's execution is divided into three stages. The program is written, compiled, and executed. A Java programmer, such as you and I, write programs. The JAVAC compiler, a key Java compiler included in the Java Development Kit, does the compilation (JDK). It produces bytecode as output and accepts a Java program as input. A program's JVM runs the bytecode produced by the compiler during the Running phase. We now know that the Java Virtual Machine's job is to run the bytecode generated by the compiler. Although each Operating System uses a distinct JVM to execute bytecode, the results are the same for all of them.

2. Bytecode in the Development process: As said, the Java source code is converted into bytecode by the JDK's Javac compiler before being executed by the JVM. The compiler saves it as an a.class file. A disassembler like javap can be used to inspect the bytecode.

3. Java Development Kit(JDK): When we first learned about bytecode and the JVM, we used the name JDK. Therefore, as implied by the name, it is a full Java development kit that comes with everything such as a compiler, Java Runtime Environment (JRE), java debuggers, java manuals, etc. We need to install JDK on our computer in order to write, build, and run the java program for it to work.

4. Java Runtime Environment (JRE): JRE is part of JDK. The java application can execute on our PCs thanks to the JRE installation, but we are unable to build it. A browser, JVM, applet support, and plugins are all included with JRE. JRE is necessary for a computer to run the Java software.

5. Garbage Collector: Java doesn't allow object deletion by programmers. JVM provides a software named Garbage Collector that may be used to erase or recover that memory. The trash collectors have the ability to recollect unreferenced things. Java, therefore, simplifies programming by taking care of memory management. However, when employing things that have been around for a while, programmers should use caution. the memory of the objects being referenced cannot be recovered by garbage.

6. ClassPath: The file path where the Java runtime and Java compiler seek for.class files to load is known as the classpath. JDK comes with a lot of libraries by default. External libraries should be added to the classpath if you want to include them.

2.2.2 Primary/Main Features of Java

1. Platform Independent: The JVM then executes the bytecode that the compiler produced after converting the source code to bytes. This means that if we develop a program on Windows, we can run it on Linux, and vice versa. This bytecode can run on any platform, including Windows, Linux, and macOS. Although each operating system uses a separate JVM, after bytecode has been executed, all operating systems yield the same results. Java is a platform-independent language for this reason.

2. Object-Oriented Programming Language: Organizing the program in the terms of collection of objects is a way of object-oriented programming, each of which represents an instance of the class.

The four main concepts of Object-Oriented programming are

Abstraction

Encapsulation

Inheritance

Polymorphism

3. Simple: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, and Explicit memory allocation.

4. Robust: Java is a strong language, which implies it is dependable. As a result of its design, which prioritizes error detection as early as possible, the Java compiler is able to identify mistakes that are difficult for other programming languages to see. Garbage collection, exception handling, and memory allocation are the three primary components of Java that provide it robustness.

5. Secure: In java, we don’t have pointers, so we cannot access out-of-bound arrays i.e it shows ArrayIndexOutOfBound Exception if we try to do so. That’s why several security flaws like stack corruption or buffer overflow are impossible to exploit in Java.

6. Distributed: The Java programming language allows us to build distributed applications. Java distributed applications are made using Remote Method Invocation and Enterprise Java Beans. Java applications are simple to spread among one or more computers that are linked to one another over the internet.

7. Multithreading: Java has multithreading capability. It is a Java feature that enables the simultaneous execution of two or more software components for optimal CPU efficiency.

8. Portable: As is common knowledge, Java code created on one system may execute on another. Java is portable because of its platform-independent characteristic, which allows its platform-independent bytecode to be run on any platform.

9. High Performance: Java's architecture is designed to minimize runtime overhead, and at times, it employs a Just In Time (JIT) compiler to generate code on-demand, generating just the methods that are actually used, allowing programs to run more quickly.

10. Dynamic flexibility: Due to the fact that Java is entirely object-oriented, we have the freedom to add classes, new methods to existing classes, and even brand-new classes through sub-classes. Java even allows native methods, which are functions created in languages other than Java, such as C and C++.

11. Sandbox Execution: With the aid of a bytecode validator, Java programs run in a distinct environment that enables users to run their apps without impacting the underlying system. The

bytecode validator, whose function it is to examine the code for access violations, also adds an extra layer of protection.

12. Write Once Run Anywhere: As was already said, a Java application creates a ".class" file that is identical to our programs but includes code in binary form. As bytecode is independent of any machine architecture, it offers ease that is architecture-neutral. Java's widespread adoption in the innovative IT sector is mostly due to this.

13. Power of compilation and interpretation: The majority of languages, whether they are compiled or interpreted, are created with a specific goal in mind. But because the Java compiler converts source code to bytecode and the JVM runs this bytecode into machine OS-dependent executable code, Java incorporates arising immense power.

2.3 Maven

The Apache Group created the well-known open-source build tool Maven to build, publish, and deploy several projects simultaneously for improved project management. Developers may create and describe the lifecycle framework using the provided tool.

2.3.1 Architecture

This diagram, provided by Apache itself, gives the best scope of Maven’s architecture.

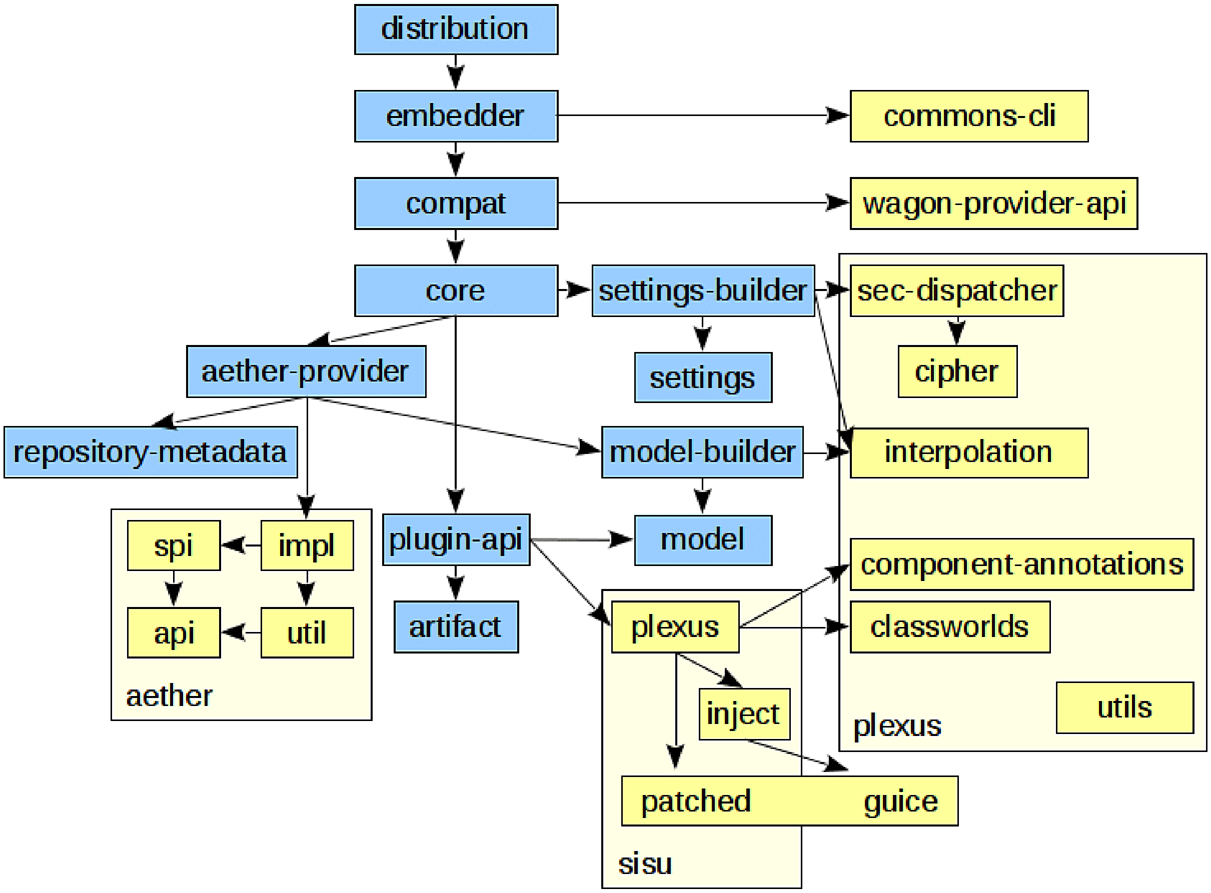


FIGURE 1 : SCOPE OF MAVEN

2.3.2 Build Tools

The software or tools used to build an executable application from source code is known as build tools. It is necessary for creating or scripting a wide range of jobs, as the name implies.

2.3.3 Project Object Model (POM)

The Project Object Model (POM), an XML file that contains all the data pertaining to project and configuration parameters, is what makes Maven so helpful. The project's description, versioning information, and configuration management information are all contained in the POM.

The project home directory is where you may find the XML file. Maven looks for the POM in the current directory when you run a task.

2*.3.4 Steps/Process Involved in Building a Project*

Here are the steps to follow when building a Maven project:

Coding for application creation can be added or written, then processed into the source code repository.

Edit any configuration, pom.XML, or plugin information that is required.

Create the program itself.

Save the results of your build process to a local server or another place as a WAR or EAR file.

Deploy the file to the client or production site by gaining access to it locally or from a server.

If required, change the date and updated the application version number in the application document.

Create and produce a report in accordance with the application's or the requirement's specifications.

2.4 Spring Framework

A lightweight framework is spring. Because it supports several frameworks, like Struts, Hibernate, Tapestry, EJB, JSF, etc., it may be viewed as a framework of frameworks. In a larger sense, the framework may be thought of as a platform where we can organize our efforts to solve various technological issues.

A number of modules, including IOC, AOP, DAO, Context, ORM, WEB MVC, etc., are included in the Spring framework.

2.4.1 Advantages of Spring Framework

There are many advantages of the Spring Framework. They are as follows:

1) Predefined Templates

2) Loose Coupling

3) Easy to test

4) Lightweight

5) Fast Development

6) Powerful abstraction

7) Declarative support

2.5 Green Package Process

BlackRock Solutions produces and quality controls risk reports on behalf of clients:

- Receives trades or positions from clients or their agents (e.g., custodians, accountants)
- Calculate full suite of risk analytics at the security and portfolio level
- Perform extensive quality control and data integrity validation
- Provide reports to clients through customized, password-protected website

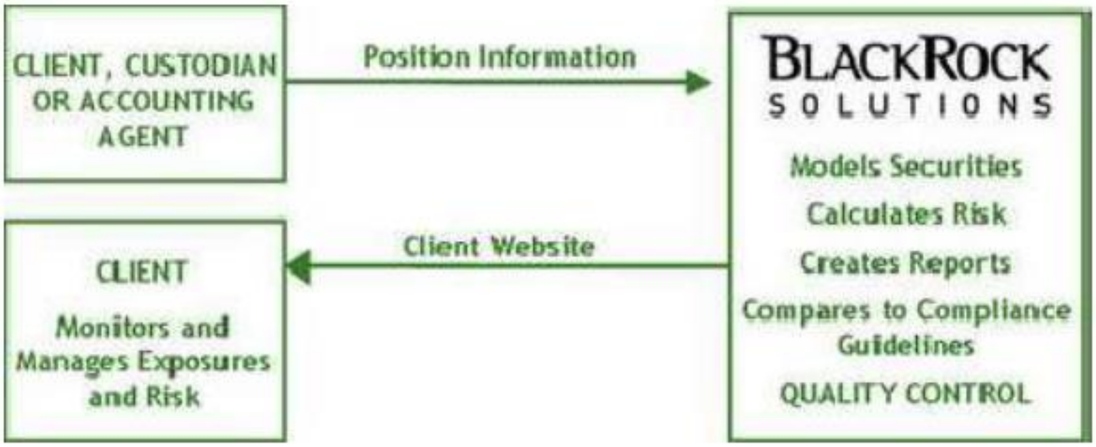


Figure 2: Green Package Process

2.6 Data

2.6.1 Financial Data

Pieces or groups of information pertaining to a business's financial situation make up financial data. Internal management examines the data to assess company performance and decide whether to adjust tactics and strategies.

As long as financial data has been accessible, vendors of financial data have existed. Starting in the 1870s, ticker tape was the first technology that allowed data suppliers to spread information. Pre-trade information, like the bid/ask data required to price a financial instrument, and post-trade information, like the most recent trade price and other transaction data, are both included in financial data.

2.6.2 Economic Data

Economic data, often known as economic statistics, are numerical measurements of actual economies, both historical and current. These are often seen in cross-sectional data spanning a single time period or in time-series form, that is, encompassing more than one time period (for example, the monthly unemployment rate for the previous five years) (say for consumption and income levels for sample households). Additionally, data may be gathered via surveys of, say, persons and businesses, or they may be compiled into sectors and industries of a particular economy or for the global economy. A data set is a compilation of such data in table form.

2.6.3 Risk Data

Risk can take many different forms, but it can be roughly defined as the possibility that an investment's actual return will differ from its projected return. The danger of losing some or all of the initial investment is included in the risk. The standard deviation of past returns or the average returns of a particular investment is typically used to calculate various types of risk.

2.6.4 Funds

A fund is a collection of funds set aside for a certain objective. A fund may be created for any objective, including the construction of a new civic center by the municipal government, the awarding of scholarships by a college, or the payment of customer claims by an insurance company.

2.7 GP Production Environment

Green Package is a risk and investment management tool, a comprehensive set of web-based reports of holdings and risk. It’s an integral process - A centralized and quality-controlled feeder of portfolio and holdings data to other Aladdin processes, including compliance, etc. Raw data is accumulated from vendors and it goes through multiple quality checks to generate reports. They include high customization portfolio structures, benchmarks, suites, and packages.

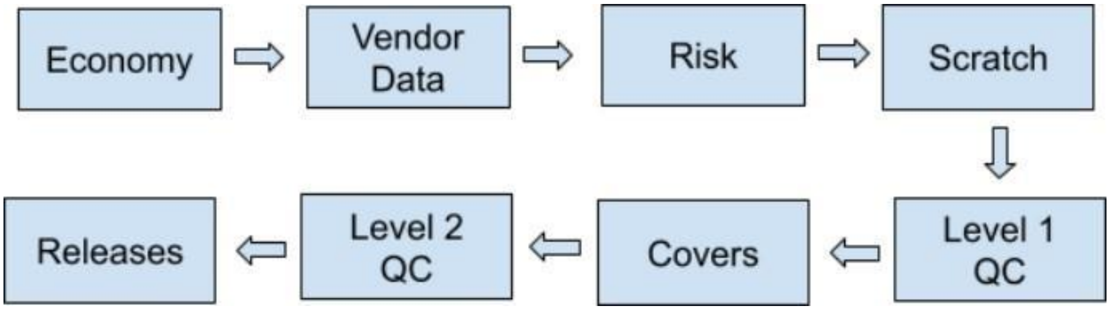


Figure 3: GP Production Cycle

- Once the trading day completes, all positions from the economy are processed. Pricing jobs kick off in no time. When the vendor data arrives, the job scratches the exceptions. Level 1 Quality check solves these exceptions and covers are kicked off. The data is used to write reports. After this, Level 2 checks start, and then completed reports are released to clients.
- Script - pms_batch.pl and chk_pms.pl are run to scratch exceptions from vendor data. Exceptions are mostly in missing data, outbound dates, wrong manager names, etc.
- GPException Monitor shows security-level exceptions populated by chk_pms and security information that can assist the investigation. Exceptions are categorized into different levels. Levels 1 to 2 are considered critical, including missing price/risk and excessive price/risk changes. Level 3 and above can be followed up.
- GPA Menu is a Unix-based application that exposes common commands to analysts for addressing exceptions.
- GPWorkflow Monitor shows the portfolio-level status on portfolios and benchmarks of a package. Options to rerun PMS on a portfolio or to proceed to cover the process.
- Level 2 QC checks are the final line of defense. Reports are being examined by analysts in a top-down fashion. Follow-ups from Level 1 QC are being examined. Changes in portfolio and benchmark risk exposures must be explained.

2.8 WDI Support

WDI (former GPA) is a consultant service provided by the DWE team to different teams in Blackrock. Issues related to Green Package Automation products are addressed during WDI Support work. These products include Central Dashboard, Taskboard, GPNotes, GPWorkflow, GPException, GPSettings, and ExConfig. Users can submit a question or report a problem/bug by raising a JIRA ticket. The request is reviewed and approved by senior members of the team. Issues are categorized as low and high-priority issues. Once the JIRA is done queuing, a developer picks the issues and investigates and resolves non-prod and production issues.

Different levels of WDI:

- Level 1: India DEV team works in EST-friendly hours to have an overlap of at least 5 hours with the US DEV team. Level 1 issues occur at a frequency of 4-5 per week and take up the DEV time for investigation. Even on Sundays the Level 1 often gets called as most of the server-related changes go out on Fridays and are caught on Sunday.
- Level 2: More property issues are handled by Level 2 consultants. These issues include, server flush and bounce, cache reload, and wrong server status display.

GPA Portal Debugging and commands GUI will help consultants quickly investigate the issue and take many of the corrective actions from a single portal, hence cutting precious time.

CHAPTER – 3

DESIGN AND METHODOLOGY

3.1 INTRODUCTION

In this chapter we will deep dive into the implementation level details of the project. First, we will take a look at the application development lifecycle, then we will move on to describe the application architecture. After that, we will take a look at the functional diagrams showing the data flow between the server and client. Throughout this chapter, we also get a detailed overview of the module specification and the tools used to create the project.

3.2 Application Development Lifecycle

The application server is being developed following the Agile software development model Figure describes the application development lifecycle followed to develop the application server.

- The first step involves recognizing the bug to fix or the feature to develop by making use of the developer specification or the user feedback.

- The second step is creating a Jira ticket which helps keep track of the status of the issue/feature. The different states a Jira ticket can take are

– New: this state is the default state of every new ticket

– Development: this state is observed when development activities, i.e., designing and coding, of the bug/feature is taking place

– Testing: this state is observed when the bug/feature is released in the TST environment and is being tested by the testing team.

– Approved: This state tells us that the changes made using the ticket have been verified by the testing team and the code reviewers and are ready to be released

– Resolved: this state is taken when the changes made using the ticket have been released in the production environment

– Closed: this state is observed when the released changes in the production environment look good and the issue/feature is working according to user expectation

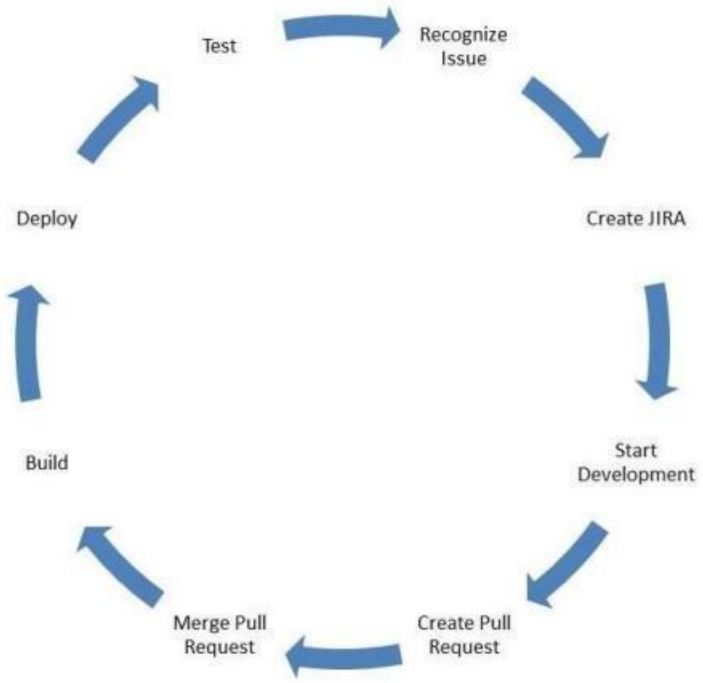


Figure 4: Application Development Lifecycle

The third step is to get your hands dirty and begin with the development activities of the feature/bug. This includes designing (in the case of UI) and writing or modification of code.

- After developer-level testing of code, we create a pull request with proper commit messages following the standards laid by GIT, and using the GIT version control framework.

- The pull request is reviewed by the expert of that particular technology in the author’s team, after which it is merged with the development branch, ready to be released in the development environment.

- After merging the changes, the project is built, for the portal, we have used 4D and Jenkins to build the project.

- After this, it is deployed to the required environment based on the status of the Jira ticket.

  After deployment, it is tested and then the whole cycle continues.

The project's development cycle is iterative, with specifications and solutions evolving by coordination between self-organizing cross-functional teams. Teams typically use agile practices to facilitate disciplined project management processes that encourage regular inspection and adaptation. Teams commonly use agile practices that facilitate disciplined project management methods that enable regular inspection and adaptation, as well as a leadership style that promotes coordination, self-organization, and responsibility. Best practices of Agile SCRUM are implemented using Azure Dev Ops, a software platform to easily adopt agile among teams.

a) Product Vision: The product's long-term objectives are outlined in its product vision. In order to achieve the ultimate objective and, ultimately, the vision, all teams must coordinate their efforts.
b) Product Roadmap - A roadmap outlines how the product will develop in order to realize the aforementioned vision. A roadmap that details the strategy with objectives and specifies the steps that will get the business closer to its goal will be created by the stakeholders.
c) Release Planning - The features that will be provided in the designated timetable are listed in the release plan.

d) Sprint Planning - Teams pledge to deliver a number of product backlogs during Iteration Planning. In Blackrock, 10-day sprints are preferred for teams.

e) Daily Stand-up - Every day, teams collaborate to update managers on the launch's success, which may include procurement specifics.

3.3 DevOps Deployment

A method of developing software that involves cooperation between information technology operations (Ops) and software development (Dev) across the service lifecycle, from architecture through production support. To stabilize the environment, speed up the development process, Push-button deployments, continuous builds, and requirement and code-repository management. Azure DevOps is the platform used by our team to facilitate Continuous Integration (CI) and Continuous Deployment Of applications and servers.

a) Quality Pipeline - This is set up for continuous integration of new checkouts from the repository. Quality is graded, ‘A’ represents best coding practices. Another filter is code coverage in testing, 80% is the minimum code coverage acceptance.
b) Build Pipeline - This is set up for continuous deployment of new checkouts from the repository. Building applications and checking its runtime are major checks in this process.

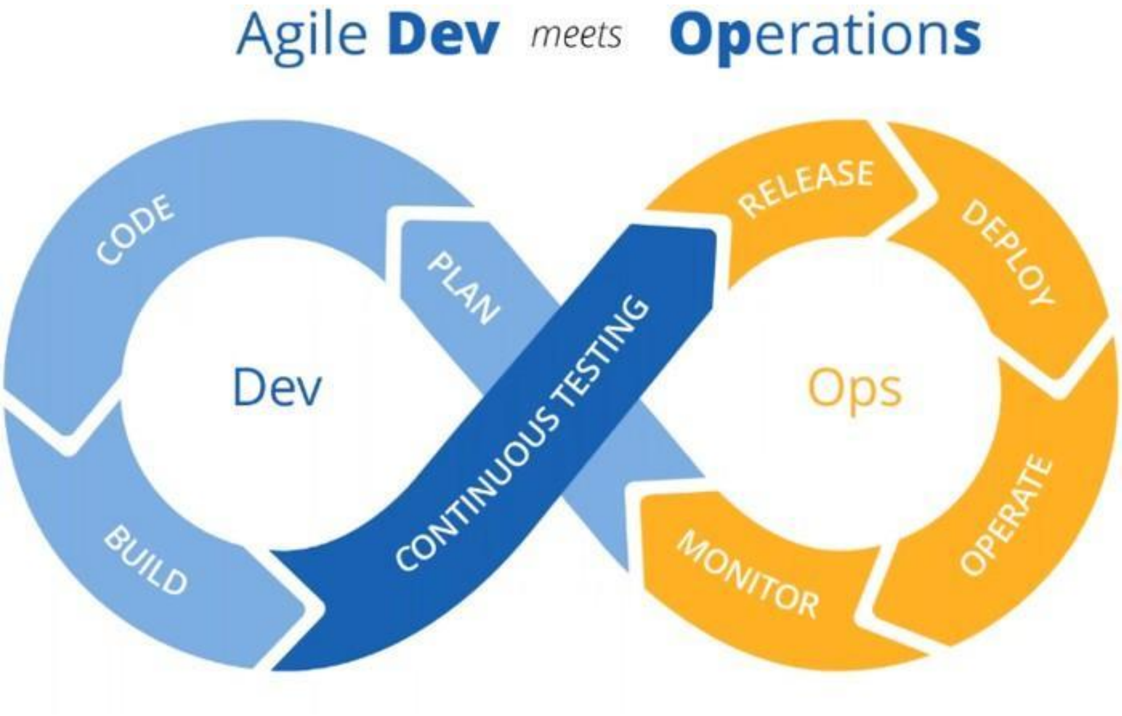


Figure 5: Azure DevOps in Parallel with Development

3.4 Testing Environment

A software and hardware arrangement called a continuous testing environment allows teams to run test scenarios. In other words, it enables test execution with the specified network, software, and hardware. The setting has been made to accommodate the needs of the application being tested. In certain instances, a test bed consists of both the test data and the test condition. By creating the ideal test environment, software testing success is secured. Any errors in this procedure might result in additional expenses and delays.

a) Unit Testing - It's a testing of isolated units in the application, usually tested by developer only. Individual units of source code may sometimes be provided with mock data to test.

b) End to End Testing - It's a testing technique where a complete application is tested from end to end (beginning to end). It ensures all integrated modules are working as expected.

c) User Acceptance Testing - It's a test conducted to ensure acceptance from users. It ensures that the application is in parallel to vision and requirements.

d) Code Review - This practice helps to keep a check on programming practices used by developers in projects.

3.5 Testing Framework

Initially the approach was to understand all the applications and their significance since Blackrock uses only in-house programs, thus requiring a multi-disciplinary methodology. Files were received in from format from clients/vendors which needed to be compatible with Blackrock required format for further processing.

The sources need to be credited based on a certain hierarchy as directed by the client. The data sources need to be closely monitored to check for any discrepancies which might further hinder the analytic calculation process later on.

Creating a centralized environment for holistic processing of all kinds of data for analyzing delays and then producing reports after Quality Control.

Once we read the files from the Aladdin Dashboard and extract them to our local system in CSV format, we need to convert them to the specified payload format (BlackRock Map). This payload is then sent as an Exception to the GPException server along with Reload message. The response received from the server is then compared with the user input. And testing is done on the received file.

This tool incorporates testing in an automated form by ensuring all the functions of the application are working appropriately and can be safely deployed in the production environment.

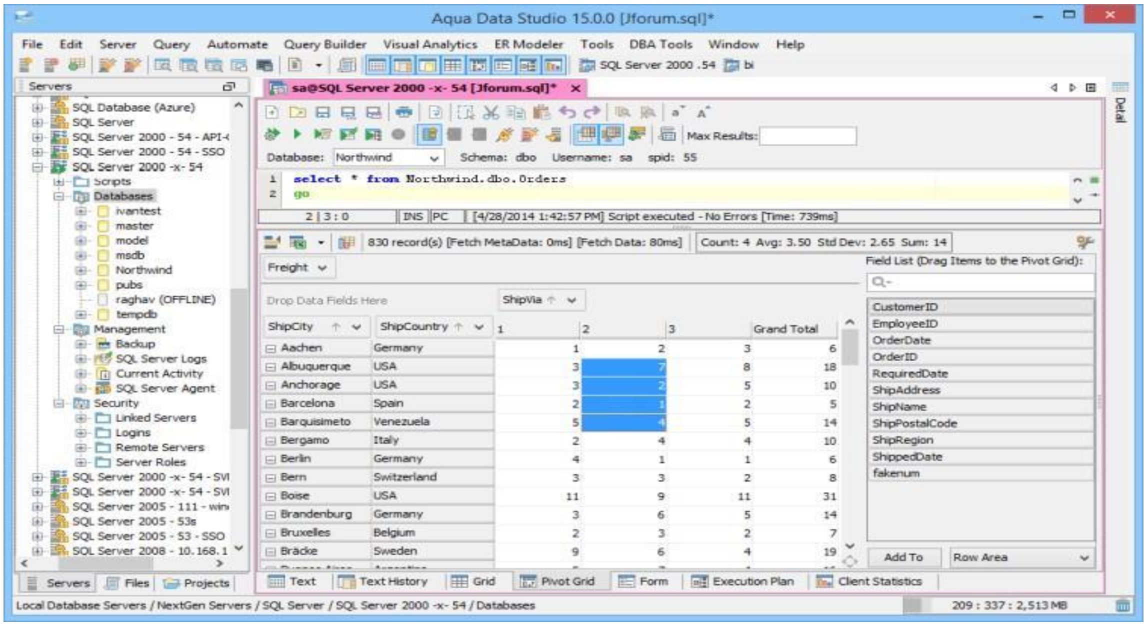
3.6 Documentation

The process of development involved not only writing code but also coming up with proper documentation for it. The documentation process included the following:

- Create JIRAs for tracking the project.
- Provide proper comments for every function and the global variable used in the code.
- Create user guides for each feature of the application server which will help the user understand the process and workflow.
- Maintain wiki pages for all the features with their statuses.

3.7 Some Essential Technologies

i. **Aqua Data Studio**: Aqua Data Studio is an IDE for databases. SQL is integrated with version control systems. Aqua data studio is the database hub of every client in Blackrock. It is an easy-to-use GUI-based IDE that can support multiple hosting servers from all around the globe making it a central database for organizations.



Database: Northwind Schema: dbo Username: sa spid: 55

```
select * from Northwind.dbo.Orders
go
```

2 | 3 : 0 INS PC [4/28/2014 1:42:57 PM] Script executed - No Errors [Time: 739ms]

830 record(s) [Fetch MetaData: 0ms] [Fetch Data: 80ms] Count: 4 Avg: 3.50 Std Dev: 2.65 Sum: 14

ShipCity	ShipCountry	1	2	3	Grand Total
Aachen	Germany	1	2	3	6
Albuquerque	USA	3	7	8	18
Anchorage	USA	3	2	5	10
Barcelona	Spain	2	1	2	5
Barquisimeto	Venezuela	5	4	5	14
Bergamo	Italy	2	4	4	10
Berlin	Germany	4	1	1	6
Bern	Switzerland	3	3	2	8
Boise	USA	11	9	11	31
Brandenburg	Germany	3	6	5	14
Bruxelles	Belgium	2	3	2	7
Bräcke	Sweden	9	6	4	19

Local Database Servers / NextGen Servers / SQL Server / SQL Server 2000 -x- 54 / Databases 209 : 337 : 2,513 MB

ii. **BlackRock Messaging Service**: Blackrock Messing service is an HTTP-based intranet service to let BlackRock servers communicate with each other. Due to security reasons, Blackrock doesn't rely on http-based open-source connection services for most of the application servers that included the business logic of financial securities and portfolios. A new BMS service can be constructed by making an app server. The receiver of the message will be the client. Messages can be sent in different formats including plain text, BlackRock Map (specialized Map only to be used in Blackrock servers), integers, Booleans, etc.

iii. **JAVA**: Java is an object-oriented, class-based general-purpose programming language. The design of the programming language allows programmers to write code anywhere and execute it everywhere without having to worry about the underlying computer architecture. Also known as "write once, run anywhere" (WORA). This implies that Java code only has to be built once in order to execute on all systems that support the Java Runtime Environment.

Primitive types and reference types are the two subcategories of types in the Java programming language. The Boolean type and the numeric types are the primitive types (4.2). The integral types byte, short, int, long, and char make up the numeric types, and the floating-point types are made up of float and double.

iv. **Junit Testing Framework** - For Java programmers, it is an open-source testing framework. Java programmers have the ability to write test cases and test their own programs.

It is one of the frameworks for unit testing. The most recent version is 4.

We must develop test cases in order to do unit testing. The unit test case is a piece of code that confirms the program logic operates as anticipated.

Many interfaces and classes for JUnit testing, including Assert, Test, Before, After, etc., are included in the org.junit package.

v.**Eclipse Ide**: For creating applications utilizing the Java programming language as well as other programming languages like C/C++, Python, PERL, Ruby, etc., Eclipse is an integrated development environment (IDE).

The Eclipse platform, which serves as the basis for the Eclipse IDE, is made up of plug-ins and may be expanded by adding new plug-ins. The Eclipse platform was created using Java, and it may be used to create integrated development environments, rich client applications, and other tools. Any programming language for which a plug-in is available can be used as an IDE with Eclipse.

3.8 Implementation Details

A script is run at the start of the day which reads the vendor paths and the config file. It also pipelines other functions in other files and creates a delivery data frame. It also maintains a separate list of extra files that weren't expected but have come nevertheless.

These exceptions are stored in the cache by servers like GPExceptionServer, GPWorkflowServer, and GPNotes2Server.

Earlier the task of checking the exceptions and errors used to be done manually, which was cumbersome and time-consuming. The application automatically takes the files and stores them in Map format to send the payload. To do that we have created a file reading and CSV to map conversion functionality.

The application then sends the signals to the server. To do that we have made use of a message sender class of the BlackRock repository.

Then once the message is sent, the response is received and stored in BlackRock element (object) format. This message would then be compared to user input. This task would be automated.

Currently, our prototype is making use of a single scenario but after being deployed, the application would be able to use many different scenarios testing based on the use case of each server.

The final step is to achieve performance and load testing in an efficient manner.

3.9 Product Management

At every step of the product life cycle, product management is a function of an internal organization that deals with new product development, corporate reasoning, planning, verification, forecasting, pricing, product introduction, and marketing.

a) Assessment and Analysis - Understanding the requirements and opportunities lying within an object is the first step in product management. Decisions finalized in this step lays out the product development cycle and efficiency of the project.

b) Delivery - It’s a step-by-step procedure used to develop and release a product. Many stakeholders like developers, quality analysts, clients, operations, etc play their part in the step.

c) Execution - Analytics and feedback gathered are put into action with never-ending releases and updates.

CHAPTER 4

RESULT ANALYSIS

4.1 OBSERVATIONS AND FINDINGS

The internship at BlackRock was my first experience in the corporate world and was an amazing experience. I experienced what it is like working in a multinational company and delivering work before deadlines. I realized the importance of studying and doing proper research about the objective properly before starting coding. I added many skills to my knowledge. I had never done finance before going to the company.

I understood the importance of financial markets and how vast this market is. I was able to grasp the 'why' of the architecture of my daily tasks. Every decision I made had to be iterated with my mentor. I realized the importance of writing rules based on clients' wishes. Through code review, I learnt what it is like to code efficiently, the naming conventions, and analyzing the network and memory usage effects of my code.

I observed that the Agile mindset is a great part of the organization. In BlackRock, a lot of new agile practices are being introduced to the teams. The scrum boards are an essential part of the daily stand-up meetings. I also completed courses in BlackRock Library on Agile mindset and Why scrum is so important. It gave me a great insight into its importance in organizations nowadays.

Not just the work but my mentors were super encouraging about the approach towards tackling software issues just like real-life problems. The team outings and fun-related events were also good.

4.2 CHALLENGES FACED

Being completely new to the corporate culture and the finance industry, I faced many challenges and I am happy that I was able to overcome them and find the solutions by connecting with various people within and outside the company. Following were the main challenges I faced:

- No prior knowledge about financial markets.
- Had never worked in a professional development environment.
- Working on a technology that is specific to some of the fintech companies only.

- Large volume of client restrictions.
- Testing product on the development, preproduction, and production servers.
- Grasping the complex portions of already made rules and understanding them.
- Architectural issues arising while development.
- Learning the mechanism of releasing scripts in Blackrock
- Decoding bugs and fixing them after releasing them

CHAPTER – 5
CONCLUSION AND FUTURE SCOPE

Overall, the team feels very optimistic about the work done, the production stage of the project is successful and in parallel to the vision. The report has acquainted the development that is being completed and future plans for the project. Overall, I got to learn new technologies in the software development process, new Information Technology standards, and methodology that is practiced to improve the efficiency of team and development.

To **conclude** my internship experience, I learned how corporate works and how we as software engineers can benefit our organization. This experience taught me a lot of patience and the ability to speak for ourselves in front of everyone and to stand out and present my point of view.

The **future scope** of this project is quite huge as the vision for this testing framework is large with its final release in production. The users and developers will be highly benefited in terms of the support work they perform weekly. The features that have been recognized have been added to the developer specification document of this application. The application is released in DEV and is being thoroughly tested in the DEV environment and there are enhancements to be done to it.

CHAPTER 6

REFERENCES

[1] “About Us,” BlackRock. [Online]. Available: https://www.blackrock.com/corporate. [Accessed: 23- May-2022]

[2] “Green package - bobsguide,” Bobsguide.com, 12-May-2021. [Online]. Available: https:// www.bobsguide.com/products/green-package/. [Accessed: 23-May-2022]

[3] K. Rungta, “Agile Methodology: What is agile software development model,” Guru99.com, 01-Jan-2020. [Online]. Available: https://www.guru99.com/agile-scrum-extreme-testing.html. [Accessed: 23-May-2022].

[4] F. M. Fowler, “What is scrum?”, in Navigating Hybrid Scrum Environments, Berkeley, CA: Apress, 2019, pp. 3–8.

[5] D. R. Prasanna, Dependency Injection: Design Pattern using Spring and Guice. 2009.

PROJECT DETAILS

Student Details			
Student Name	**Mansi Sharma**		
Register Number	180905294	Section / Roll No	C/ 37
Email Address	mansisharma.8.k@gmail.com	Phone No (M)	8607200363
Project Details			
Project Title	**Automated Testing Framework**		
Project Duration	6 months	Date of reporting	17th Jan 2022
Organization Details			
Organization Name	**BlackRock**		
Full postal address with pin code	Building No. 14, DLF Cyber City, DLF Phase 2, Sector 24, Gurugram, Haryana 122002		
Website address	https://careers.blackrock.com/locations/gurgaon/		
External Guide Details			
Name of the Guide	**Hrituparn Alreja**		
Designation	Analyst		
Full contact address with pin code	Building No. 14, DLF Cyber City, DLF Phase 2, Sector 24, Gurugram, Haryana 122002		
Email address	halreja@blackrock.com	Phone No (M)	9968938924
Internal Guide Details			
Faculty Name	**Dr Geetha Maiya**		
Full contact address with pin code	Dept of Computer Science & Engg, Manipal Institute of Technology, Manipal – 576 104 (Karnataka State), INDIA		
Email address	geetha.maiya@manipal.edu		

CO and PO Mapping

NBA:

Table A1.1 Course Articulation Matrix

CO		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CSE 4299.1	Apply mathematics, science and engineering skills to identify, formulate, synthesize and solve the problems from various areas of computer science engineering.	3	2	3	3	1	2	3	2	2	2	2	3	3	2	3	3
CSE 4299.2	Have knowledge of new trends in engineering/technology by developing programmable coding in various computer programming languages.	3	3	3	2	2	1	2	2	3	2	1	3	3	2	3	3
CSE 4299.3	Use the industry standard tools to analyze, design, develop and test software engineering based applications.	3	2	2	2	3	2	1	3	2	3	3	2	2	2	2	2
CSE 4299.4	Apply theoretical knowledge to real-world engineering problems and manage complex engineering projects.	2	2	1	3	3	2	1	3	3	1	2	2	1	1	1	2
CSE 4299.5	Acquire skills of collaboration and independent learning.	3	2	2	2	3	3	3	1	1	3	3	3	2	1	2	2
CSE 4299 (Avg. correlation level)		**2.8**	**2.2**	**2.2**	**2.4**	**2.4**	**2**	**2**	**2.2**	**2.2**	**2.2**	**2.2**	**2.6**	**2.2**	**1.6**	**2.2**	**2.4**

PROGRAM OUTCOMES (PO)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO)

1. Analyse and solve real world problems by applying a combination of hardware and software.
2. Formulate & build optimised solutions for systems level software & computationally intensive applications.
3. Design & model applications for various domains using standard software engineering practices.

4. Design & develop solutions for distributed processing & communication.

Table A1.2 Program Articulation Matrix

COURSE Code	Course Title	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PS02	PS
CSE 4299	Project Work	**Avg: 2.8**	**Avg: 2.2**	**Avg: 2.2**	**Avg: 2.4**	**Avg: 2.4**	**3**	**2**	**1**	**3**	**3**	**2**	**2**	**2**	**3**	**3**

IET (AHEP Mapping):

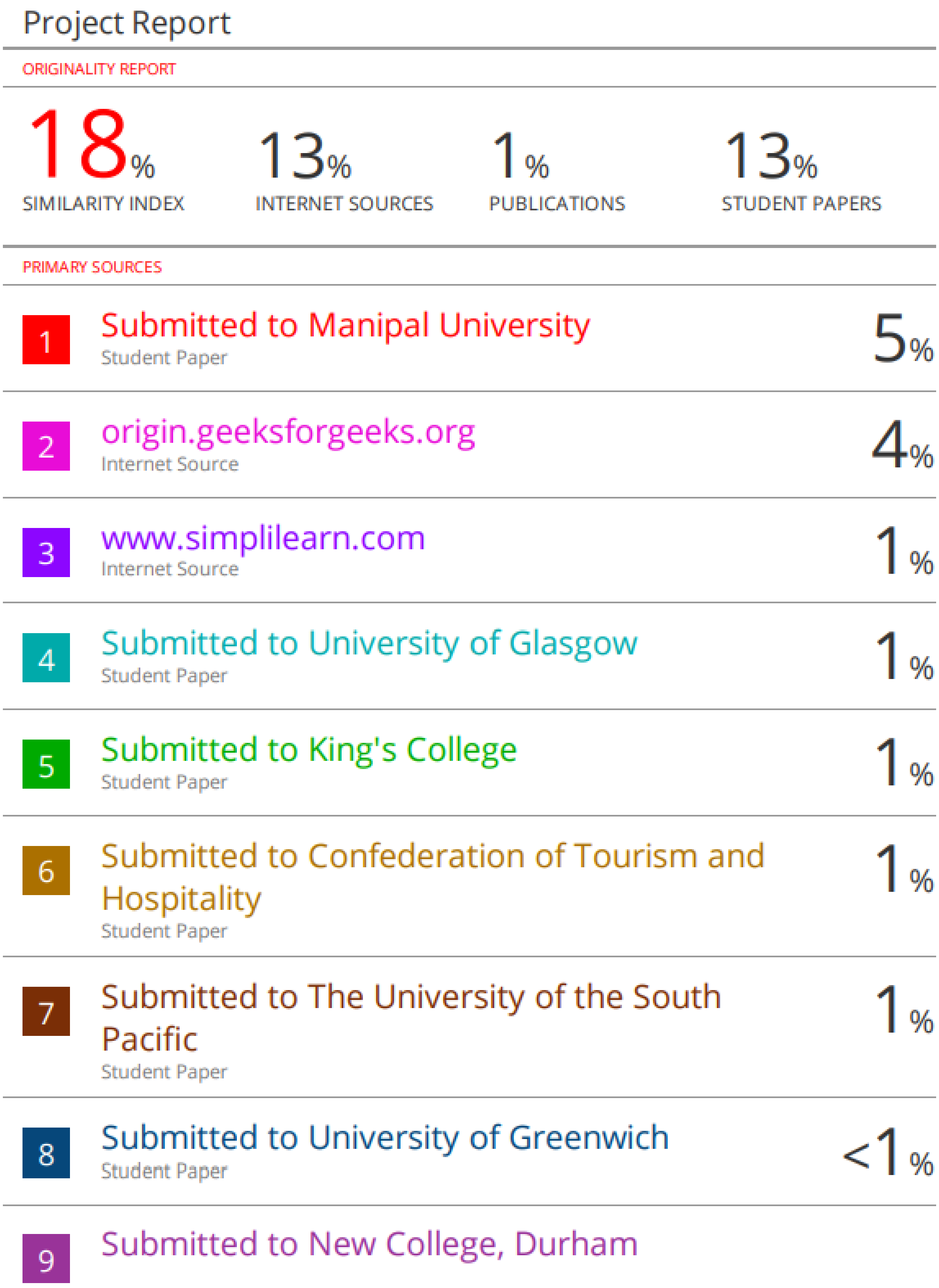
Table A1.3 CLO-AHEPLO Mapping

		AHEP LOs						
CLOs	**Statements**	**C8**	**C9**	**C10**	**C12**	**C15**	**C16**	**C17**
1	Apply mathematics, science and engineering skills to identify, formulate, synthesize and solve the problems from various areas of computer science engineering.	3	2	2	1	3	3	2
2	Have knowledge of new trends in engineering/technology by developing programmable coding in various computer programming languages.	3	2	1	3	1	2	3
3	Use the industry standard tools to analyze, design, develop and test software engineering based applications.	2	3	3	2	2	3	2
4	Apply theoretical knowledge to real-world engineering problems and manage complex engineering projects.	2	3	2	3	1	2	2
5	Acquire skills of collaboration and independent learning.	3	1	2	2	1	2	3

AHEPS

C8	Identify and analyse ethical concerns and make reasoned ethical choices informed by professional codes of conduct
C9	Use a risk management process to identify, evaluate and mitigate risks (the effects of uncertainty) associated with a particular project or activity
C10	Adopt a holistic and proportionate approach to the mitigation of security risks
C12	Use practical laboratory and workshop skills to investigate complex problems

C15	Apply knowledge of engineering management principles, commercial context, project and change management, and relevant legal matters including intellectual property rights
C16	Function effectively as an individual, and as a member or leader of a team
C17	Communicate effectively on complex engineering matters with technical and non-technical audiences



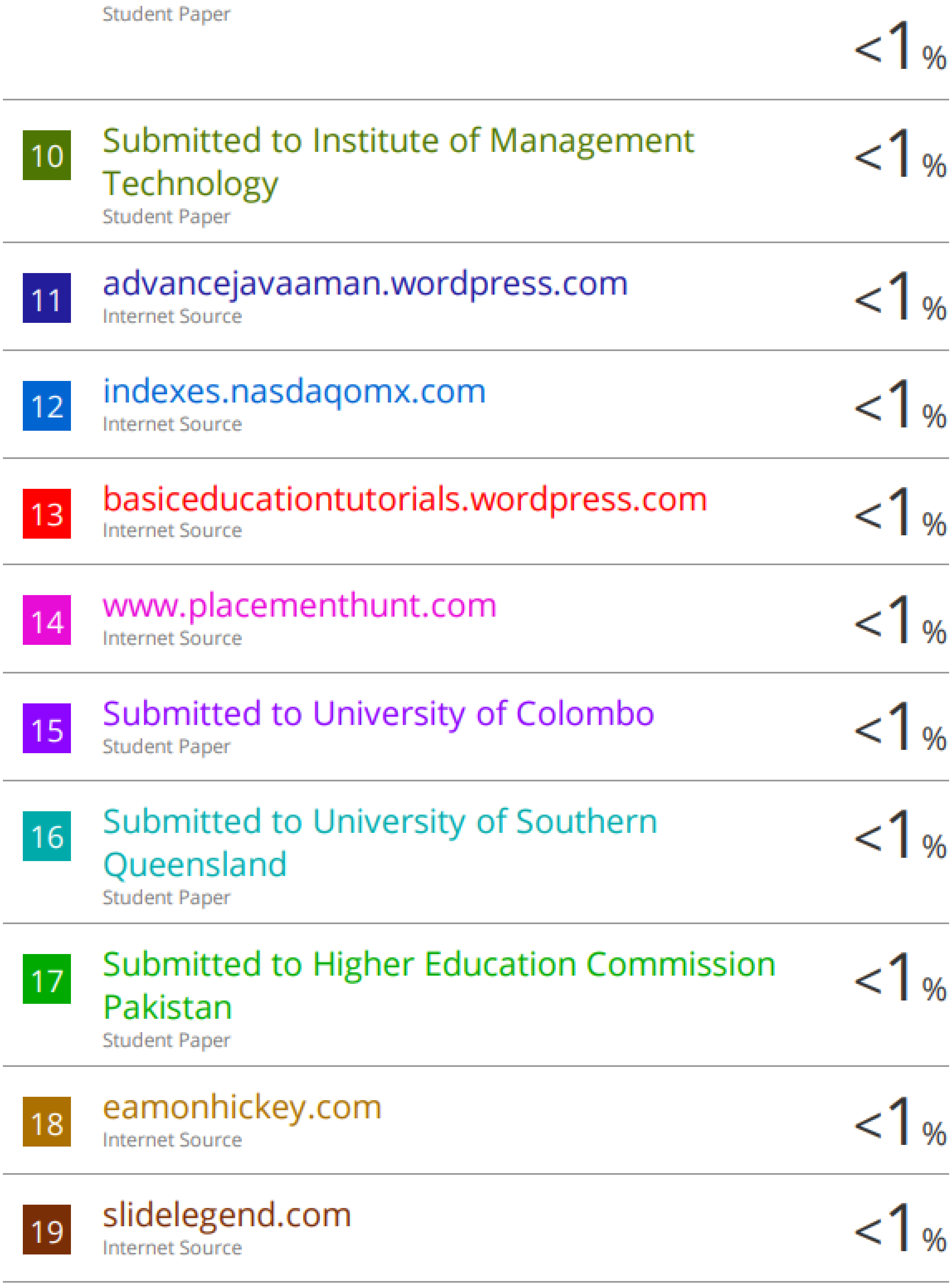
Project Report

ORIGINALITY REPORT

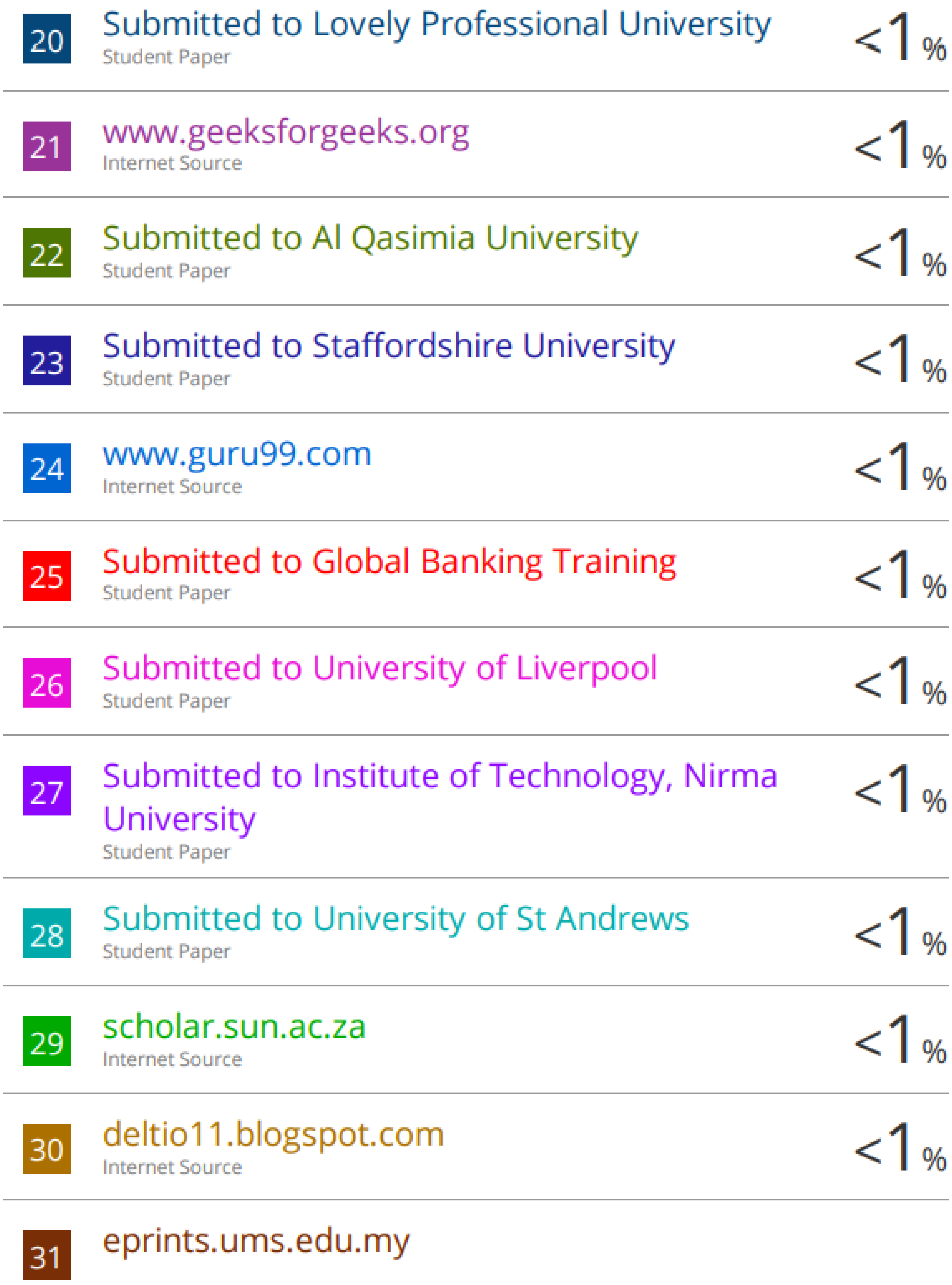
18%	13%	1%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

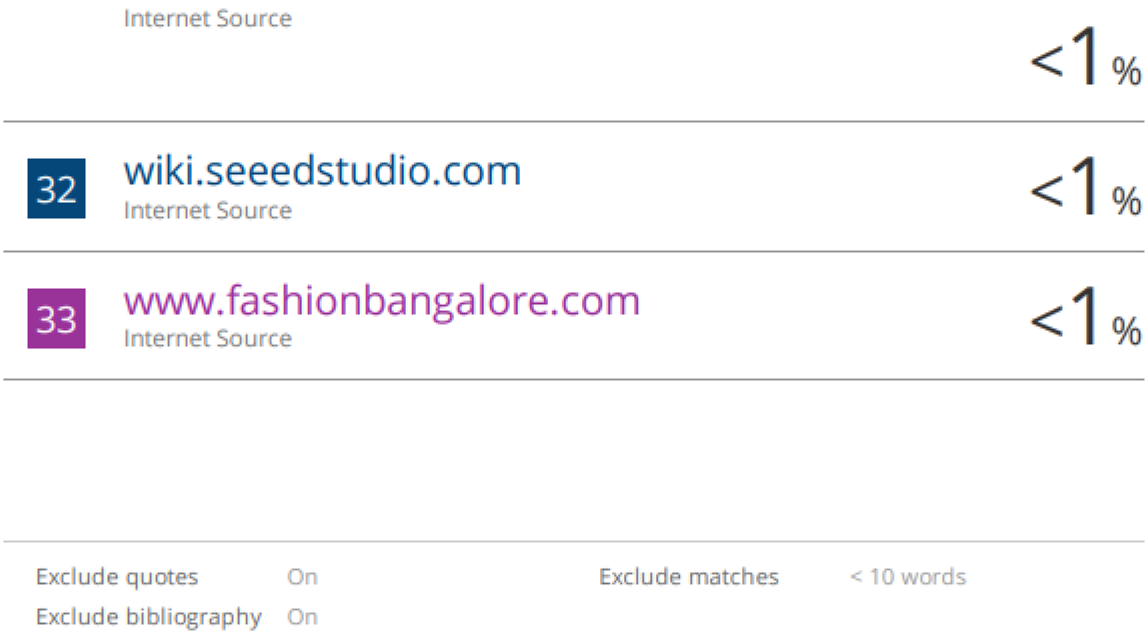
1	Submitted to Manipal University Student Paper	5%
2	origin.geeksforgeeks.org Internet Source	4%
3	www.simplilearn.com Internet Source	1%
4	Submitted to University of Glasgow Student Paper	1%
5	Submitted to King's College Student Paper	1%
6	Submitted to Confederation of Tourism and Hospitality Student Paper	1%
7	Submitted to The University of the South Pacific Student Paper	1%
8	Submitted to University of Greenwich Student Paper	<1%
9	Submitted to New College, Durham	



	Student Paper	<1 %
10	Submitted to Institute of Management Technology Student Paper	<1 %
11	advancejavaaman.wordpress.com Internet Source	<1 %
12	indexes.nasdaqomx.com Internet Source	<1 %
13	basiceducationtutorials.wordpress.com Internet Source	<1 %
14	www.placementhunt.com Internet Source	<1 %
15	Submitted to University of Colombo Student Paper	<1 %
16	Submitted to University of Southern Queensland Student Paper	<1 %
17	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
18	eamonhickey.com Internet Source	<1 %
19	slidelegend.com Internet Source	<1 %



20	Submitted to Lovely Professional University Student Paper	<1 %
21	www.geeksforgeeks.org Internet Source	<1 %
22	Submitted to Al Qasimia University Student Paper	<1 %
23	Submitted to Staffordshire University Student Paper	<1 %
24	www.guru99.com Internet Source	<1 %
25	Submitted to Global Banking Training Student Paper	<1 %
26	Submitted to University of Liverpool Student Paper	<1 %
27	Submitted to Institute of Technology, Nirma University Student Paper	<1 %
28	Submitted to University of St Andrews Student Paper	<1 %
29	scholar.sun.ac.za Internet Source	<1 %
30	deltio11.blogspot.com Internet Source	<1 %
31	eprints.ums.edu.my	



	Internet Source	<1 %
32	wiki.seeedstudio.com Internet Source	<1 %
33	www.fashionbangalore.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		