MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

(An institute of national importance)

Department of Computer Science and Engineering

Major Project report on

Retail as a Service

Submitted by

Shreyash Gajbhiye

161112247

Submitted To

DR. MANISH PANDEY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

(An institute of national importance)

DECLARATION

I hereby declare that this project work for the Major Project has been carried out by me under the supervision of Dr. Manish Pandey, Department of Computer Science and Engineering, MANIT BHOPAL , Mr. Nitish Agarwal, Mr Shubham Gupta and peers at Flipkart. No part of this project has been submitted for the award degree or diploma to any other Institute.

Shreyash

SHREYASH GAJBHIYE

161112247

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY, BHOPAL

(An institute of national importance)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that Yash Dhingra student of 4th year, B.Tech, Computer Science and Engineering have successfully completed his 6 Month Internship Project titled "**Retail as a Service**" at Flipkart as a Major Project in fulfillment of their B.Tech degree in Computer Science and Engineering.

Nitish Agarwal

DR. MANISH PANDEY
(PROJECT COORDINATOR)

MR. NITISH AGARWAL
(MANAGER, FLIPKART)

ACKNOWLEDGEMENT

With due respect, I express my deep sense of gratitude to my respected major project coordinator Dr. Manish Pandey for his valuable help, support and guidance.

I express a deep gratitude to my Project Manager Mr. Nitish Agarwal and my mentor Mr. Shubham Gupta, who have always helped me in every aspect in contributing to the project. I am thankful for the encouragement that they have given me in completing the tasks successfully. Their rigorous evaluation was of great assistance.

I am also grateful to my respected Director Dr. N.S. Raghuwanshi and H.O.D Dr. Nilay Khare, who allowed me to go for the 6 month internship. I would like to express my deep appreciation towards my family members, batch mates and fellow interns and peers at Flipkart for providing the much-needed support and encouragement.

TABLE OF CONTENTS

1. ABSTRACT

BUSINESS-TO-BUSINESS (B2B) COMMERCE, the exchange of products, services,or information between businesses—instead of businesses and consumers—is expanding by leaps and bounds. This growth is pushing more companies to rethink their fulfillment practices, particularly in the areas of inventory management, order velocity, and the ability to meet order commitment dates. To effectively run scalable and sustainable business retail B2B suits are needed. One such project is RaaS (Retail as a Service) in flipkart.

2. INTRODUCTION

To effectively run scalable and sustainable business, procure to receive systems are needed. Having well defined system for procure to receive process helps in managing these basic Core dimensions of supply chain management :

- **Reliability & Predictability** - This core dimension helps in optimising overall process so as to reduce cost and improve efficiency in structural and planned manner. Core measure for this dimension.
- **Responsiveness & Agility -** This core dimension helps in reacting to ever changing physical world and how the same process, people and systems are adjusting to ensure a high plan is adhered or losses are minimised.
- **Transparency & Visibility -** Not having right visibility and alignment leads to too much unstructured communication between buyer, ops and warehouse. For instance in Fk ecosystem before scheduling launch, for getting slots for a single PO there are at least 4 contacts being performed (On average it takes about 2-3 days for getting slots from vendors due to to-fro communication gap). Following way this can be measured.
- **Compliance -** Every country rules and regulation keeps on changing, changing rules heavily impacts the way one org operates. Having a generic and extensible system allows org to be resilient in ever changing domain at least cost. Aim of developing procurement to receive multi-tenancy is to make sure that there should be no or very minimal impact on day to day business operations.

Therefore, considering the problems Retail as a Service (RaaS) can solve not only for the flipkart ecosystem but can also be externalized as a service to other retailers for B2B interactions, it can bring enormous changes in the commerce industry.

3. LITERATURE SURVEY

There has been a thorough study of different core product suits to compare what capabilities are available in industry which customers are expecting. There are basic softwares available for order life cycle management, basic configurable workflow, order acknowledgement and spend management. For schedule line life cycle, various softwares are available by SAP, Oracle, GEP. Similarly, for automating basic purchase orders, SAP, GEP, Oracle and Jagger provide basic softwares. No softwares is available for capacity management, supplier and buyer expectation management. Ivalau provides software for Supplier Inventory Management & Collaboration. For Operational Policy Adherence, SAP and Oracle provide different softwares. Coupa, Baswara and GEP softwares support E-invoicing. Therefore, no B2B suit is available that can support the entire retail business operations. RaaS will be able to provide all functionalities/operations required in the commerce industry.

4. METHODOLOGY AND WORK DESCRIPTION

As RaaS is a big and long term project on which multiple teams of the Retail Department(in Flipkart) are working together to make it a success, I have contributed to different micro-services in RaaS. Also, I have contributed to the current working infrastructure that is internal to flipkart as new bugs are discovered from time to time or new product requirements are always needed.
Following are the tasks performed by me that contributed to Retail as a Service project directly:

4.1. Create Invoice Receipt Note API for Invoice Receipt Note Service

IRN or Invoice Receipt Note is a kind of document that is created while inwarding of a consignment for a PO. This document notes all the items that are received in the warehouse against a PO when a consignment of that PO arrives at the warehouse. This document is later used by the Accounts team to verify what was the quantity that was received at the warehouse as this is important information needed while computing payments. Although this document is generated at the very end and not a part of the retails team, all the information required for this is collected while inwarding itself. This API that I wrote had the responsibility to create an IRN in the IRN entity. The API would return a status code 201 on successful creation of an IRN.

4.2. Store Consignment Note Document API for Consignment Management Service(CMS)

CMS is a micro-service in RaaS project which provides different REST APIs related to consignment entities. Consignment forms an integral part of the b2b fulfillment system as it helps in creating a clear translation of vendor expectations to retailers or vice versa. By creating consignment, the vendor/retailer sets clear expectations on retailer/vendors by telling when(time) and what(units) stocks will be delivered. In the consignment entity a document called Consignment Note Document is created which consists of the items that are to be returned back to the owner because of some fault or some error. This document is generated in other services but the document id of that generated document is stored in the CMS as well as it is a document owned by CMS. Although this API developed was later dumped as an alternative design was made which did not require CMS to keep this document id with it.

4.3. Get Consignment Note Document API for Consignment Management Service(CMS)

This API is also a part of the CMS entity and is a GET API for the above POST API to fetch the document id which will then be used to fetch the actual Consignment Note Document from the document service. Although this API was also dumped as the new approach adapted does not require this functionality.

4.4. Document Solutioning for Generation, Storing and Retrieval of Documents

This task was different from other tasks as this was not a coding or implementation type. The nature of this task was to design a solution for creation and generation of documents for the RaaS project. The result of this document solutioning was then going to be used by all the services in the Retail universe that either needed to generate or store a document or to retrieve a document. The task took a lot of effort in understanding various documents services available in flipkart that could be used. Each of these services were thought of and in depth analysis was done as a part of the task. As a result of this task, A combination of two document services was proposed by me to fulfill all the needs. This solution was accepted and has now been implemented as well.

4.5. Registering Document Commands in Multitenancy Commons Library

As a result of the previous task, after finalization of the design and services to be used it was necessary that the functionalities to generate, store or retrieve a doc are provided to any service with a minimum understanding of how this is being taken care of (abstraction). Thus new commands were added to the library as a part of this task so that anybody who wants to generate, store or retrieve a document could do it easily by just using the commands from the library without actually having to understand how it is being taken care of. Two commands, one for generation of document on the basis of a template and the other to store a document were added to the library. There was no need for a specific command to retrieve a document in the library as it would be directly requested on the document service itself.

4.6. Developer On Call for all on call issues in production for a sprint (2 weeks)

Developer on call is a developer that is appointed to look after the issues that occur in the production code. For example if some service that in real world physical implementation is not working efficiently or incorrectly. This is a very high issue as the service is failing in some sense in the real world implementation. Hence 2 weeks a developer is appointed specifically to handle all these on call issues. I was also given this opportunity to act as an oncall developer and deal with problems occurring in the deployed production code that is causing some discomfort to some entity using the service. A part of this is to identify and fix the issues. I handled 2 issues as a part of this. Both of them required some updation in the condition statements as needs had changed according to time. As this code is needed to be deployed in production as soon as possible therefore a very strict procedure is followed for it’s deployment in a little time.

5. TOOLS AND TECHNOLOGIES USED

5.1. Git (Version Control System)

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

5.2. Build Automation Tools

5.2.1. Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Maven addresses two aspects of building software: how software is built, and its dependencies. Unlike earlier tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache. This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

Maven is built using a plugin-based architecture that allows it to make use of any application controllable through standard input. A plugin for the .NET framework exists and is maintained, and a C/C++ native plugin is maintained for Maven 2.

5.2.2. Gradle

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language instead of the XML form used by Maven for declaring the project configuration. Gradle uses a directed acyclic graph to determine the order in which tasks can be run.

Gradle was designed for multi-project builds, which can grow to be quite large. It supports incremental builds by intelligently determining which parts of the build tree are up to date; any task dependent only on those parts does not need to be re-executed.

5.3. Mysql

MySQL is the world's most popular open source database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

5.4. Dropwizard

Dropwizard is a Java framework for developing ops-friendly, high-performance, RESTful web services. Dropwizard pulls together stable, mature libraries from the Java ecosystem into a simple, light-weight package that lets you focus on getting things done. Dropwizard has out-of-the-box support for sophisticated configuration, application metrics, logging, operational tools, and much more, allowing you and your team to ship a production-quality web service in the shortest time possible. Dropwizard straddles the line between being a library and a framework. Its goal is to provide performant, reliable implementations of everything a production-ready web application needs. Because this functionality is extracted into a reusable library, your application remains lean and focused, reducing both time-to-market and maintenance burdens.

5.5. Guice

Guice is an open source, Java-based dependency injection framework. It is quite lightweight and is actively developed/managed by Google. Guice embraces Java's type safe nature, especially when it comes to features introduced in Java 5 such as generics and annotations. Ideally, the language itself would provide most of the same features, but until such a language comes along, we have Guice.

Guice helps you design better APIs, and the Guice API itself sets a good example. Guice is not a kitchen sink. Guice aims to make development and debugging easier and faster, not harder and slower. In that vein, Guice steers clear of surprises and magic. You should be able to understand code with or without tools, though tools can make things even easier. When errors do occur, Guice goes the extra mile to generate helpful messages.

5.6. Hystrix

Hystrix is a latency and fault tolerance library designed to isolate points of access to remote systems, services and 3rd party libraries, stop cascading failure and enable resilience in complex distributed systems where failure is inevitable. This is a very famous library provided by Netflix which is considered as one of the organisations with best technologies as they handle a very high traffic on their servers.

5.7. Java

Java is a general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low-level facilities than either of them.

As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

5.8. Ruby

Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write. Ruby is most used for building web applications. However, it is a general-purpose language similar to Python, so it has many other applications like data analysis, prototyping, and proof of concepts. Although nowadays Ruby is not being used enough and even in Flipkart we are moving from Ruby to Java itself, but some code base is still fully operating in Ruby.

5.9. Padrino framework

Padrino is a free and open-source web framework, written in Ruby and based on Sinatra. It is an alternative to other Ruby web frameworks such as Ruby on Rails, Merb, Nitro and Camping. It is dependent on the Rack web server interface. Padrino was created and open-sourced in 2010. The framework was created in order to extend Sinatra to more easily support rich web applications.

6. IMPLEMENTATION AND CODING

- **Create Invoice Receipt Note API for Invoice Receipt Note Service**

  In this task, the old implementation of this API was to be modified with the new implementation as some new fields in the input data were added. Both the API's were to be maintained though as the old one will be used currently in production while the new one in RaaS. The coding for this task was done in Ruby and a POST API was registered to be used for creation of Invoice Receipt Note.

- **Store Consignment Note Document API for Consignment Management Service(CMS)**

  In this task a POST API was created that would be called by the Orchestrator (the service that takes care of the order of implementation and workflows) after the Consignment Note Document is created in some document service. In this call the unique document ID would be stored in the database which can be later used for retrieval. This was coded in Java using the dropwizard framework.

- **Get Consignment Note Document API for Consignment Management Service(CMS)**

  In this task a GET API was implemented that would be called by the service that needs to fetch the consignment note document ID to fetch the actual document from the document service. This code was also implemented in Java using the dropwizard framework.

- **Document Solutioning for Generation, Storing and Retrieval of Documents**

  This task was a solutioning task that required a lot of research within the document services available in Flipkart and proper reasoning as the result of this task was something that was supposed to be general as it wasn't solutioning regarding one kind of document or something.

The final conclusion after in depth studying of all the document services available in flipkart was to use a combination of two document services.

A service named DSS will be used to generate documents based on a template.

Another service named RYUK will be used for storing and retrieval of documents.

The reasons being as follows :

1. DSS doesn't provide the facility to store documents that are not generated by it. It only stores the document generated by it.
2. In the DSS archival of documents is a manual procedure to be carried out.
3. RYUK is not capable of generating documents for a given template.
4. RYUK allows to have a configurable Archival policy for documents.
5. Also in RYUK documents can be fetched on the basis of metadata.

- **Registering Document Commands in Multitenancy Commons Library**

As a result of the above solutioning, two commands were to be registered in the Multitenancy Commons Library so that it is easy for others to use these functionalities of the document service.

1. Create Document in DSS Command
2. Store Document in RYUK Command

Both of these commands were added to the library. This was implemented in Java. This was one of a kind of a task as uploading and downloading of documents APIs of any kind were not implemented before in this library.

- **Developer On Call for all on call issues in production for a sprint**

Two issues were identified in production. They were

1. PO Warehouse Change API

In order to change warehouse for a purchase order, the validations had to be updated. Earlier change of warehouse was allowed if the PO had no IRN against it. But according to the new needs PO warehouse will be allowed for a PO even if it has an IRN against it given that the status of that IRN is "Rejected". The changes were made in the Ruby Code base.

2. Update Status of an IRN to “Rejected”

   In order to update the status of an IRN to “Rejected”, it should not have any irn items against it.

   But change of validation was required that status could still be updated to “Rejected” even if it has an IRN item, just that the inwarded quantity for it should be 0. The changes were made in the Ruby Code base.

7. RESULT ANALYSIS

Ultimately, with successful completion of RaaS, any retailer can onboard easily. The implementation tasks required complete end-to-end local testing and pre prod deployment along with Unit test coverage. The codes were reviewed by at least two senior developers from the team and only allowed to be merged in branches if they were satisfied and identified the code as the most optimal version and had enough code coverage of more than 90%.

8. CONCLUSION AND FUTURE SCOPE

In this report, I briefly explained the motivation of this project and showed some background materials. Then, I precisely illustrated the tasks along with the work description, including the learning task and the performance task.

The goal is to create a long term compliant multi-tenant retailing software suite for ensuring business and customer experience continuity and creating externalisation/monetization opportunities by making RaaS a success. A lot more development is yet to be done and more features and functionalities will be added in the future.

9. REFERENCES

[1] https://www.dropwizard.io/en/stable/getting-started.html

[2] https://github.com/google/guice/wiki

[3] https://www.mysql.com/

[4] Flipkart internal documentation

[5] https://github.com/Netflix/Hystrix/wiki

[6] https://gradle.org/

[7] https://maven.apache.org