

Cardinal

Project Report

by

Akanksha Gupta (151311)

Under the guidance of

Mr Provash kumar Dowari



DEPARTMENT OF COMPUTER SCIENCE.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT, INDIA.

MAY 2019

(I) TABLE OF CONTENTS

Chapter No.	Topics	Page No.
	Student Declaration	III
	Acknowledgement	
	Abstract	
	List of Figures	VI
	List of Tables	VIII
	List of Symbols and Acronyms	IX
Chapter-1	Introduction	X-XII
	1.1 About the Organization	
	1.2 About the product	
	1.3 About my role	
	1.4 Technology Used	
Chapter-2	Products and Technologies	XIII-XXIV
	2.1 Falcon Assurance Navigator	
	2.2 Tableau	
	2.3 Chart.js	
Chapter-3	Software Engineering Paradigm	XXV-XXVII
Chapter-4	Analysis Document	XXVIII-XXXI
	4.1 Software Requirement Specification	
	4.1.1 Purpose	
	4.1.2 Project Scope	
	4.1.3 Overall Description	

4.1.4 Functional Requirements

Chapter-5	Testing	XXXII-XXXVII
Chapter-6	Conclusion	XXXVIII
References		XXXIX

(II)
STUDENT DECLARATION

I hereby declare that this submission is my own work carried out at **Fair and Isaac Corporation** from **4.02.2019** to **30.06.2019** and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or another institute of higher learning, except where due acknowledgment has been made in the text.

Place: Bangalore

Signature:

Date: 30.06.2019

Name: Akanksha Gupta

Enrollment No: 151311

(III)

ACKNOWLEDGEMENT

I would like to thank **Fair and Isaac Corporation** for providing me with an opportunity to pursue my industrial training, as it is an important part of my B.Tech course and it is the one that exposes you to the industry standards and makes you adapt yourself to the latest trends and technologies. At the same time, it gives an experience of working on a project. I feel proud and privileged in expressing my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavor if I do not place my acknowledgement.

I express my sincere gratitude to **Lokesh Pant** for his inspiration, constructive suggestion, mastermind analysis and affectionate guidance in my work, without which this project work completion would have been impossible for me.

I would like to express my deep sense of gratitude to my supervisor **Mr Provash Kumar Dowari** who has been a source of inspiration for me all through the project and he not only provided me the technical guidance but also his moral support and encouragement.

Name of Student: Akanksha Gupta

Enrollment Number: 151311

Date: 30.06.2019

(IV)

ABSTRACT

FICO deals with many products , FAN is one of the product whose client is Stanford University. FAN is a product in FICO's Fraud, Waste, and Abuse portfolio. FAN is currently focused on expenses incurred by institutions that are using grant money received from the Government of the country as well as private foundations. This product is able to demonstrate the ability to apply analytics against historical transactions to detect instances of fraud and abuse. FAN has applicability for private foundations as well, as they care that the grantees are good financial stewards of their funds, and often impose constraints of their own. The workflow of the project is explained in the report. Development , Testing and DevOps are the important parts in the project which are done with proper planning and care. Various types of transactions are posted by the students of the university for which validity check is performed by this product to notify the university whether there is any fraud in the or not and accordingly cases are generated.

Name: Akanksha Gupta

Date: 30.06.2019

(V)
LIST OF FIGURES

- 1: Login Page
- 2: Home Page
- 3: Create New Queue
- 4: Search Cases
- 5: Search Transactions
- 6: Throughput Report for Past day
- 7: Create New group
- 8: Create New Role
- 9: Tableau Connectors
- 10: Tableau Product suite
- 11: Web Data Connector
- 12: Data in Tableau
- 13: Report 1
- 14: Report 2
- 15: Reporting UI

16: Stages involved in SDLC

17: High Level architecture of the application

18: Defect Life Cycle

19: JIRA Issue Type

20: JIRA Issue Type Scheme

21: JIRA Issue Type Scheme II

(VI)
LIST OF TABLES

1: Operating Environment

2: Hardware Specifications

(VIII)
LIST OF SYMBOLS AND ACRONYMS

- 1: FAN – Falcon Assurance Navigator
- 2: FICO – Fair and Isaac Company
- 3: QA – Quality Assurance
- 4: UI – User Interface
- 5: ODM – Object Document Mapper
- 6: SDLC – Software Development Life Cycle
- 7: DDS - Design Document Specification
- 8: SRS – Software Requirements Specification
- 9: API – Application Programming Interface
- 10: UAT- User acceptance testing
- 11: OCR – Optical character recognition
- 12: SDK- Software Development Kit

CHAPTER 1 – INTRODUCTION

1.1 About the Organization

Designer Bill Fair and mathematician Earl Isaac established FICO in 1956 — with an underlying venture of \$400 each — on the rule that data, utilized appearing, will improve business determinations' presented investigative arrangements like credit reviewing that have made credit extra wide possible, in not just the us anyway round the world .FICO has spearheaded the occasion and utilization of essential advances behind call the board.

These typify prophetic investigation, business rules the executives and optimisation. We utilize these advancements to help organizations improve the exactness, consistency and agility of their muddled, high-volume choices.

FICO conveys predominant prophetic examination arrangements that drive more brilliant decisions. The organization's pivotal utilization of arithmetic to foresee customer ehavior has changed whole ventures and reformed the manner in which chance is overseen and items are marketed. FICO's creative arrangements exemplify the FICO Score — the quality live of bank line chance inside the United States along the edge of industry-driving answers for overseeing credit records, trademark and limiting the effect of misrepresentation, and tweaking customer offers with pinpoint accuracy. Most of the world's high banks, moreover as driving safety net providers, retailers, pharmaceutical organizations and government offices have confidence in FICO answers for quicken development, the executives chance, help benefits and fulfill administrative and focused needs. FICO could be a leading analytics software package company, serving to businesses in 90+ countries build higher choices that drive higher levels of growth, gain and client satisfaction. The company's ground-breaking use of massive knowledge and mathematical algorithms to predict shopper behavior has remodeled entire industries.

FICO provides analytics package and tools used across multiple industries to manage risk, fight fraud, build additional profitable client relationships, optimize operations and meet strict

government laws. Many of our product reach industry-wide adoption — like the FICO® Score, the quality live of line risk within the U.S.

FICO solutions leverage ASCII text file standards and cloud computing to maximise flexibility, speed preparation and cut back prices. The company additionally helps several folks manage their personal credit health.

1.2 About the product – Falcon Assurance Navigator (FAN)

FAN is a product in FICO's Fraud, Waste, and Abuse portfolio. FAN is currently focused on expenses incurred by institutions that are using grant money received from the Government of the country as well as private foundations. This product is able to demonstrate the ability to apply analytics against historical transactions to detect instances of fraud and abuse. FAN has applicability for private foundations as well, as they care that the grantees are good financial stewards of their funds, and often impose constraints of their own.

The foremost objective of “Falcon Assurance Navigator” is that system provides the investigative capability for freeform data visualization across historical transactions. It is injecting automated decisioning logic into existing multiday, low volume manual processes. Inserting seconds of processing delay into a 24 hour procurement or expense process will obviously have a negligible impact. Additionally, transaction volumes are typically in the thousands per day range. As such there are no anticipated performance challenges such as those found with other high volume, low latency fraud transaction systems. However, some of the original systems' processes may be batch oriented, causing bursts of transactions at particular times of the day. This may require throttling of the transaction bursts, preferably at the adaptor level.

1.3 About my role as a Software Engineer Intern

I worked on tableau which is a software to generate UI reports. Tableau was introduced for the first time in the team and had the sole responsibility to learn and generate required reports for the product. I also worked with Angular 6 and used chart.js library to generate same set of reports I generated using Tableau and integrated those reports with the UI of the product called the Case Manager Client. I was also assigned the task of data masking i.e. changing confidential field of the data and maintaining the referential integrity in the database.

1.4 Technology Used

Various Technology Used are listed below

Java: Java innovation is each a programming language and a stage. With most programming dialects, you either accumulate or decipher a program so you'll run it on your pc. The Java programming language is unprecedented in this a program is each ordered and comprehended. With the compiler, first you make an interpretation of a program into partner middle of the road language known as Java bytecodes — the stage autonomous codes comprehended by the translator on the Java stage. The translator parses and runs each Java bytecode guidance on the pc. Gathering happens just once; elucidation happens at whatever point the program is dead.

Morphia: It is a Java-based Object Document Mapper (ODM) Framework for MongoDB. Falcon Assurance Navigator uses Mongo database which is a NoSQL database. To query the database using Java Morphia framework is used.

Tableau: It is a data visualization software . It has various types Tableau Public which is free , Tableau Desktop , Tableau Server , tableau online etc. I have worked with Tableau Public and Tableau Desktop. The Software is populated with data and then required reports can be generated. It has various options to import data . I used Web data Connector to import data into Tableau. Web data connector allows to make API calls using some JavaScript.

Angular 6: The UI of the project is developed using Angular 6. My part was to gearate the same reports generated using tableau and integrate those with the UI. Chart.js library was used to generate those reports. This library has code implementation in different web languages . I used it with Angular 6 . The library has options to create different types of graphs , one can generate the report according to their usage.

CHAPTER 2 - PRODUCT AND TECHNOLOGIES

2.1 Falcon Assurance Navigator (FAN)

Falcon Assurance Navigator consists of six microservices

Receiver Module: The receiver implements the Transaction API. It receives transactions, parses them, stores them in the database and returns a result to the caller. If it returns success, then the transaction is accepted by FAN and will either run to completion, or notify the source system of failure.

Attachment Processor Module: If the transaction contains pointers to attachments, the Attachment Processor either retrieves the attachment(s) from the local file system (S3), or makes a call to the Document Server to retrieve the attachments, calls the OCR Service, and injects the resulting text into the transaction.

Event Handler Module: This component subscribes to the events that other components publish, and executes RESTful calls to the source system to provide all kinds of information: transaction evaluation results, cases created, cases dispositioned, and all kinds of errors.

Case Management Module: The Case Manager component is the collection of services that provide an interface with the end user to handle suspicious transactions.

Context Processor Module: The Context Creator Service prepares a Context Object to be passed to the Decision Services and the Analytics Engine. This is accomplished by enriching the transaction with the necessary associated transactions, according to the needs of those services.

Management Module: This component handles the notifications of the various types of transactions. The hold functionality enables the notifications to be held in the Notification collection and not consumed further.

The Client of FAN is Stanford University. The software is used to detect any kind of fraud if there is any kind of fraud done by students of the university while applying for the reimbursement process.

The implementation of the above microservices is done in Java except of the case management module which is implemented in Angular 6 .

The UI home page is as below . This UI is for the Stanford through which they can analyze different types of transactions done , amount involved in that and other confidential things they wish to see.

Case Management Module Overview

The Login Page

To log on, open your browser and enter the URL provided by your system administrator. The Log on page appears. Falcon Assurance Navigator will have User name and Password to enter. After entering the details user should able to Log on to access Home page.



Fig 1. Login Page

The Home

After you Log on, Home page will appear. It will allow the user to access the cases generated in the Transaction type Queue, allow to view the throughput reports, create the queues, create the views for the transaction types, allow permissions to be assigned to various roles, assign the roles to various groups.

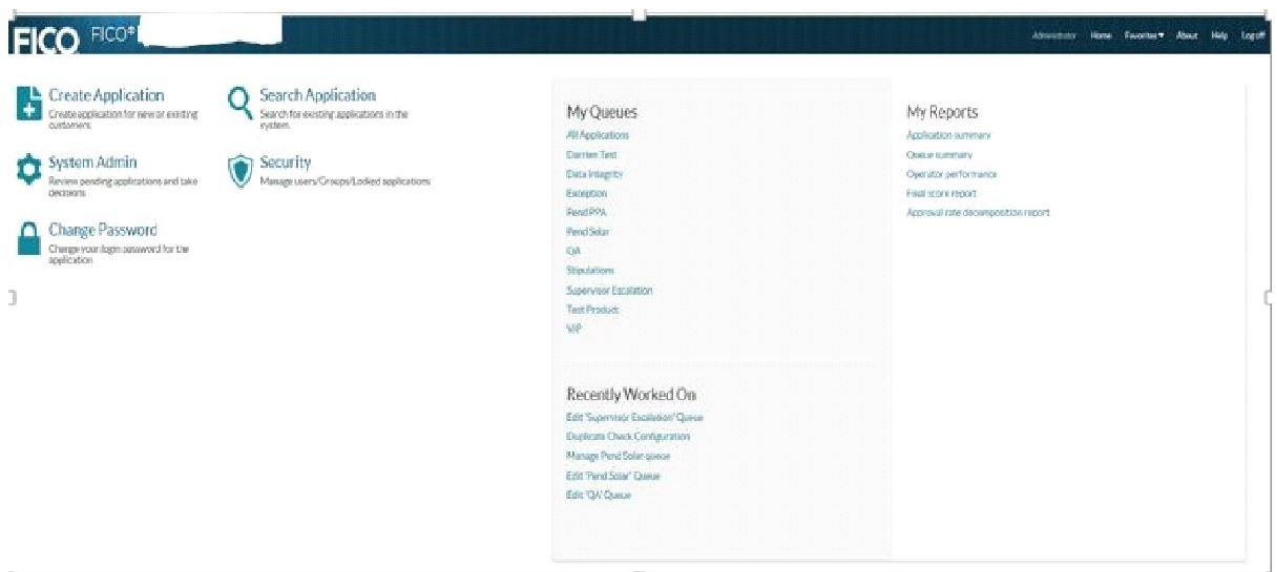


Fig 2. Home Page

Create Queue/Views:

The Create View/Queue option under the System tab on the Home page allow the users to perform various operations on the queues/views. The user can create,delete,edit the queues/views.

Create New Queue

DEFINED QUEUE

Queue name:

Transaction type: Expense report

Allow skip cases:

Description (Optional):

DEFINE FILTER CRITERIA (OPTIONAL)

Field	Operator	Value	Case sensitive
No filters defined Add one			

ADD ROW

DEFINE SORT CONDITIONS (OPTIONAL)

Execution order	Field	Sort order
No sort conditions are defined Add one		

ADD ROW

ASSIGN QUEUE TO GROUPS (OPTIONAL)

Type to search groups...

SAVE QUEUE Cancel cases Cancel

Fig 3. Create New Queue

Cases:

The Cases option on the Home page lets you search the cases in the transaction type queues and view the recently searched cases.

FICO® Falcon® Assurance Navigator

HOME CASES DISCOVERY REPORTS SYSTEM RULE CONFIGURATIONS

Search cases

Transaction type: Expense report

New search Saved searches Recent searches

Select a queue: ExpenseReport All Queue

Queue filter definition: Select Any

ADDITIONAL FILTER CRITERIA

Field	Operator	Value
No filters defined Add one		

ADD FILTER CRITERION

EXECUTE SEARCH Clear Save this search

Fig 4. Search Cases

Discovery:

The Discovery option on the home page lets you search the transaction in the Transaction type views and view the recent transactions.

Fig 5: Search Transactions

Reports:

The Reports option on the Home page provides a way to view the rejection rate and throughput reports for the selected time duration (past day, past week, past month, past year, or user specific dates).

Transaction type	ALL (Computed - Discovery + Failed)	Discovery (transactions)	Case Search (cases)	% Flagged	Failed	% Failed	New	Pending	Closed
Expense report	0	0	0	0.00%	0	0.00%	0	0	0
Payment request	0	0	0	0.00%	0	0.00%	0	0	0
PCard	0	0	0	0.00%	0	0.00%	0	0	0
Purchase requisition	0	0	0	0.00%	0	0.00%	0	0	0
Purchase order	0	0	0	0.00%	0	0.00%	0	0	0
Invoice payment	0	0	0	0.00%	0	0.00%	0	0	0

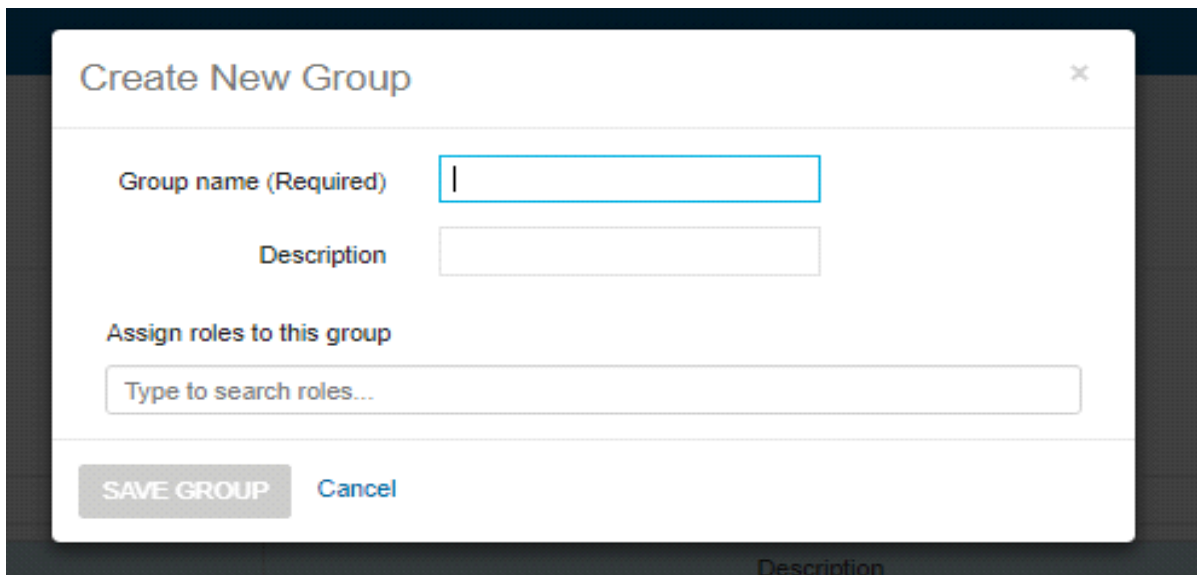
Fig 6. Throughput Report for Past day

Rule Configurations:

The Rule configuration option on the Home page displays the various Merchant Category Codes(MCC),the expenditure types(federal/non-federal),award types(federal/non-federal) and provides option to adjust the various rule configurations.

Create Group:

The Groups option under the System pane on the Home page helps to create the new group to which the various roles can be assigned which further can be assigned various permissions to perform various action on the Case Manager UI.

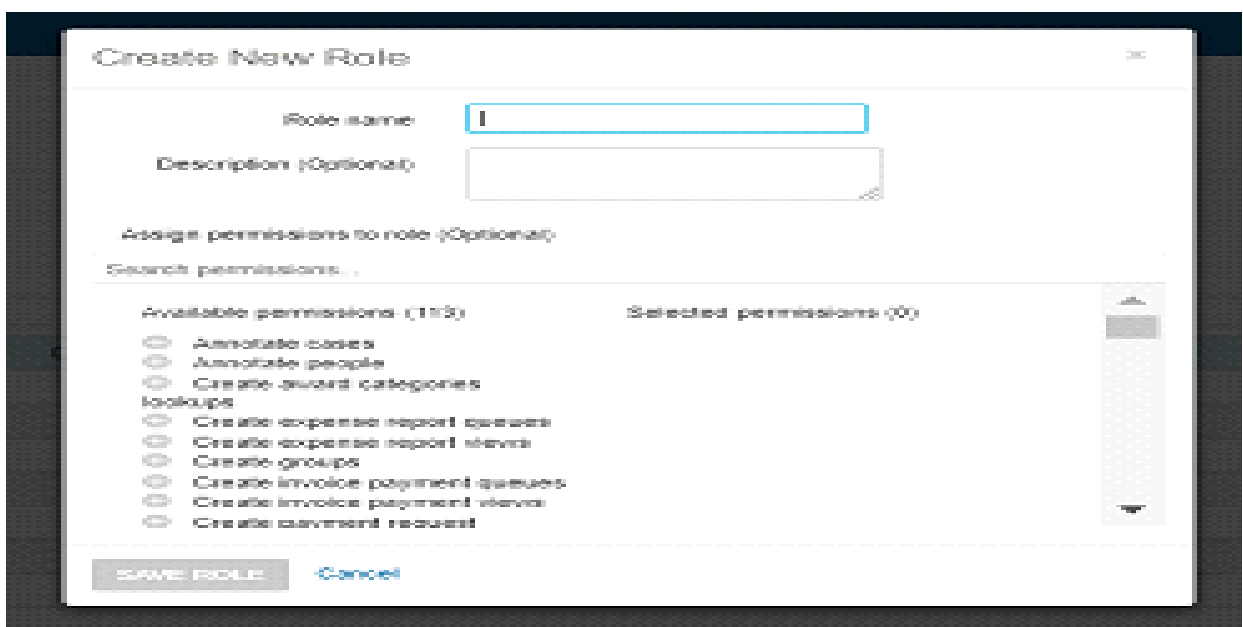


The screenshot shows a 'Create New Group' dialog box. It has a title bar with a close button (X). Below the title bar, there is a 'Group name (Required)' label followed by a text input field. Underneath that is a 'Description' label followed by a text area. A section titled 'Assign roles to this group' contains a search input field with the placeholder text 'Type to search roles...'. At the bottom of the dialog, there are two buttons: 'SAVE GROUP' and 'Cancel'.

Fig 7. Create New group

Create Roles:

The Roles option under the System tab in the home page helps you create the role and assign the role various permissions to create/view the transaction type views. It also displays the groups assigned to the roles.



The screenshot shows a 'Create New Role' dialog box. It has a title bar with a close button (X). Below the title bar, there is a 'Role name' label followed by a text input field. Underneath that is a 'Description (Optional)' label followed by a text area. A section titled 'Assign permissions to role (Optional)' contains a search input field with the placeholder text 'Search permissions...'. Below the search field, there are two columns: 'Available permissions (113)' and 'Selected permissions (0)'. The 'Available permissions' column contains a list of permissions with radio buttons next to them: 'Annotate codes', 'Annotate people', 'Create award categories lookups', 'Create expense report queries', 'Create expense report views', 'Create groups', 'Create invoice payment queries', 'Create invoice payment views', and 'Create payment request'. The 'Selected permissions' column is currently empty. At the bottom of the dialog, there are two buttons: 'SAVE ROLE' and 'Cancel'.

Fig 8. Create New Role

Logging Off

There is only one correct way to exit the system. It is very important to log off at the end of each session before closing the window or proceeding to a different URL so that the old session is cleaned up. If you encounter a problem while logging off or are unsure whether you have logged off from the system, close your browser or browser tab to end the session.

4.2 Receiver Module Overview

The Receiver is responsible for ingesting the transaction, parsing its payload, persisting the transaction, and returning a result to the source system. The following results can be identified:

- Some serious system error occurs, such as database unavailable, or resource starvation. The Receiver returns a fatal error: HTTP 5xx.
- Parsing failed; the receiver detects an error in the payload that it cannot recover, and returns a fatal error: HTTP 4xx.
- The transaction is successfully parsed, the JSON transformed into a valid Java Object Graph, and subsequently persisted in the MongoDB database. The Receiver returns a successful code: HTTP 200.

My part was to integrate tableau generated or the chart.js generated report with this UI . So I used following technologies were used to accomplish the same.

2.2 Tableau

Tableau could be a powerful and quickest growing information visualisation tool utilized in the Business Intelligence trade. It helps in simplifying raw data into the very easily understandable format that is it connects and extracts the data stored in various places and import the data into it and allows us to generate different types of reports , story and dashboard . The data can be imported from various kind of databases i.e Relational or NoSQL both or can be imported using API calls. To connect to a particular database a specified type of connector is provided by which

we can connect to the database of our interest.

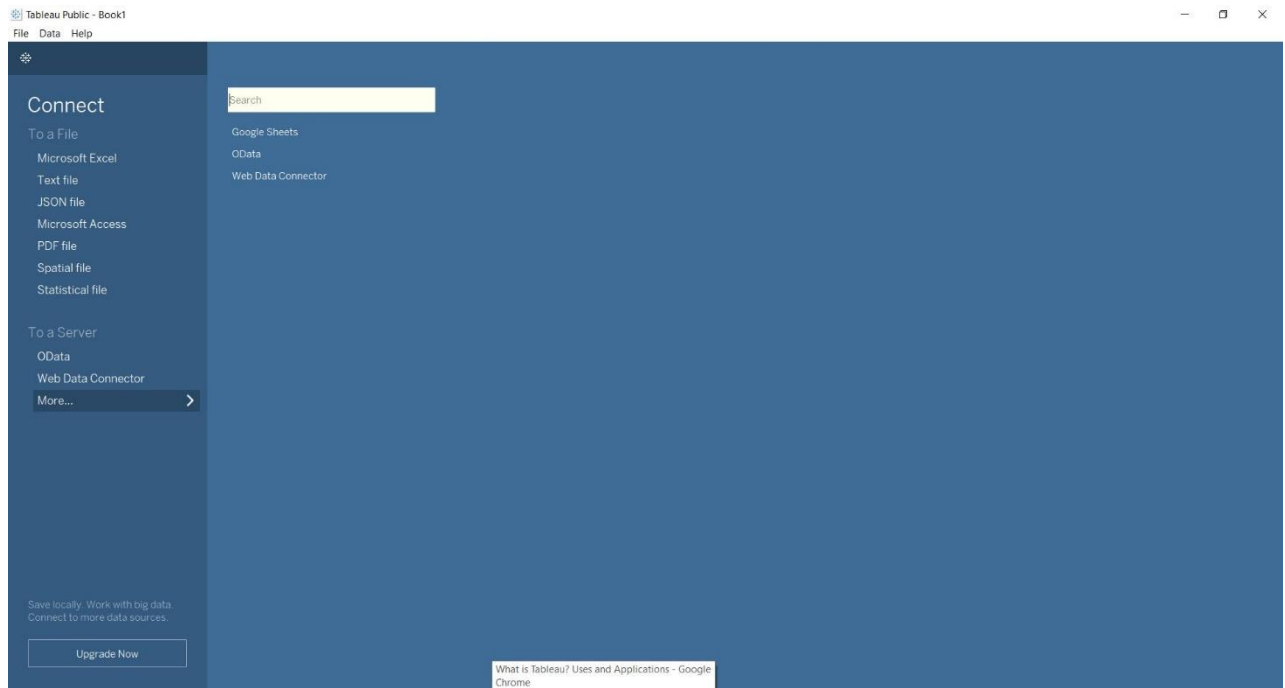


Fig 9. Tableau Connectors

Tableau product suite consists of

- 1) Tableau Public
- 2) Tableau Desktop
- 3) Tableau Server
- 4) Tableau Online
- 5) Tableau Reader

Developer Tools : Scene Public and Tableau Desktop are the designer devices for example these apparatuses are utilized to create various sorts of charts, reports and so forth utilizing the information brought into it. Scene Desktop contains a made list of capabilities and grants you to code and tweak reports. In Tableau open, exercise manuals made can't be spared locally, thus, it should be spared to the Tableau's open cloud which might be seen and gotten to by anybody.

Sharing Tools: Tableau Server, Tableau Online and Tableau Reader are used to share the reports generated with the general public or the authenticated users depending upon the type of tool you are using.

Tableau server is specifically utilized to share the workbooks, visualizations that area unit created within the Tableau Desktop application across the organization

Tableau on-line has all the similar functionalities of the Tableau Server, however the info is keep on servers hosted within the cloud that area unit maintained by the Tableau cluster. Tableau Reader may be a free tool that permits you to look at the workbooks and visualizations created victimization Tableau Desktop or Tableau Public.

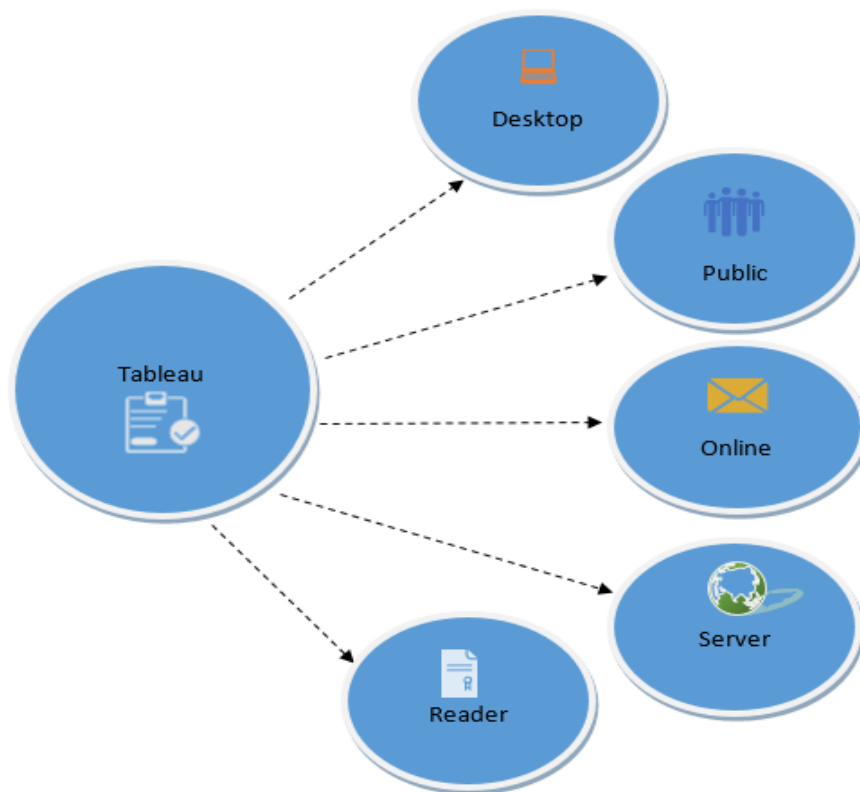


Fig 10. Tableau Product Suite

To import data into tableau I used web data connector which is used to make API calls. FAN has created its own APIs which I used to import data into tableau.

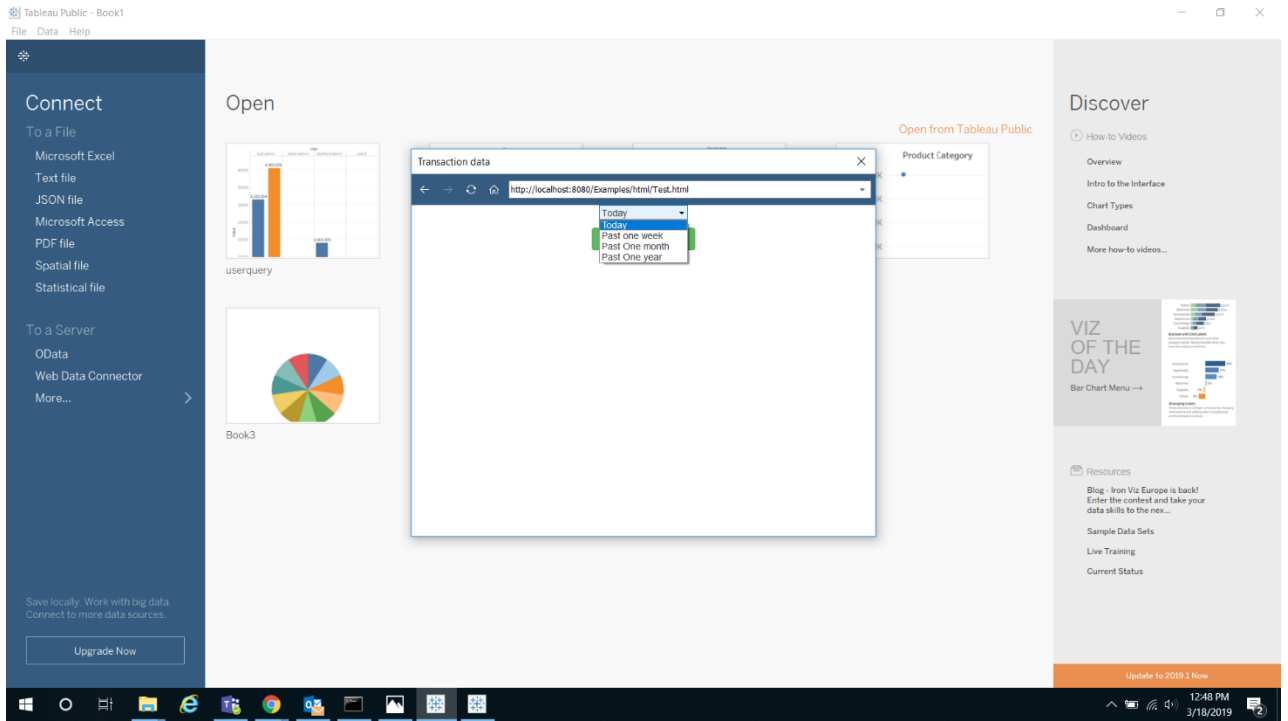


Fig 11. Web Data Connector

After importing data

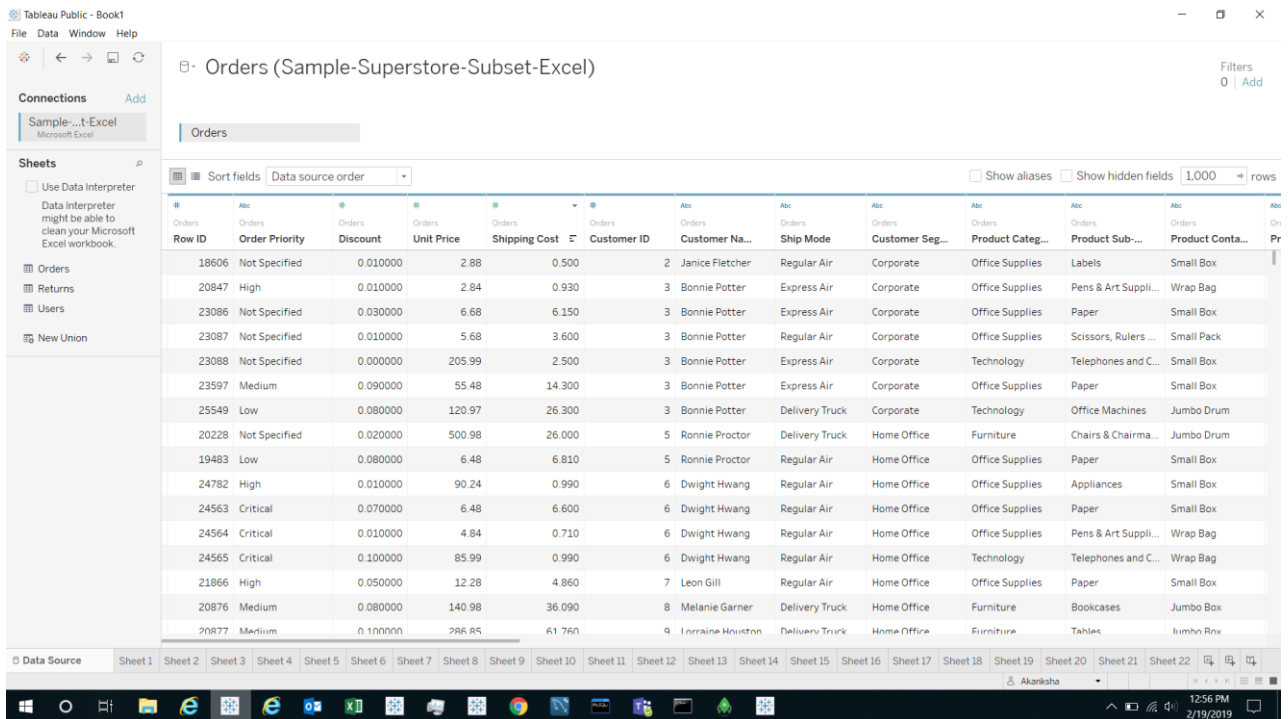


Fig 12. Data in Tableau

(Can not show the original data, as it would be against the company's policy)

Some reports that I generated are

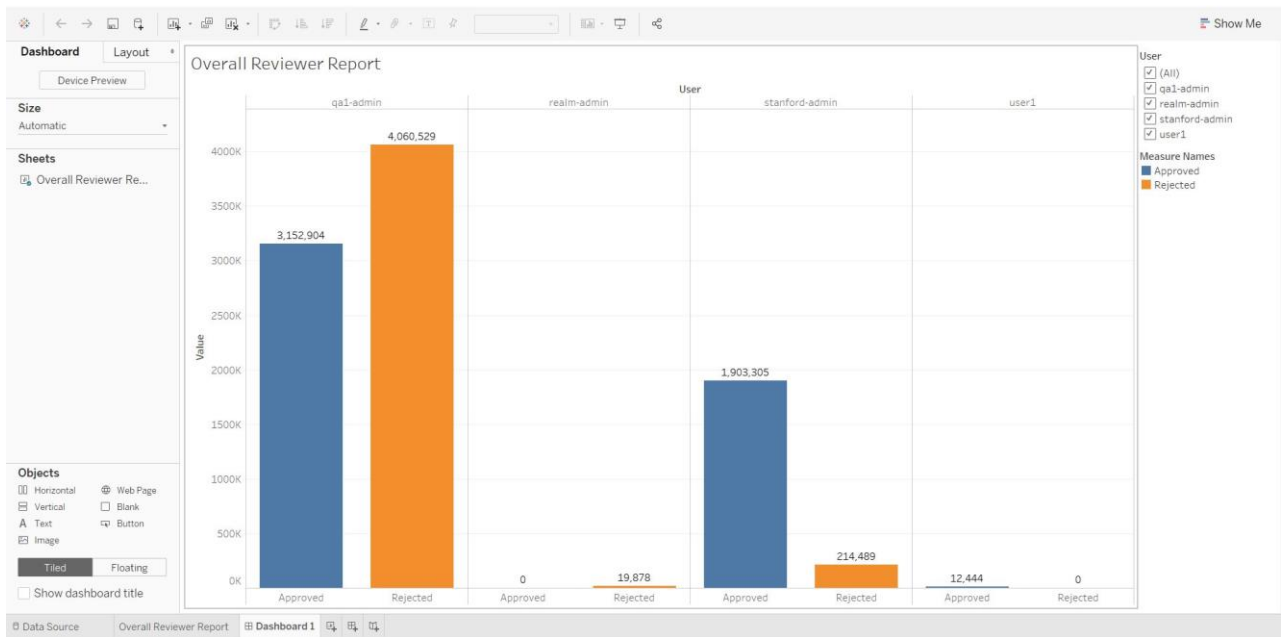


Fig 13. Report 1

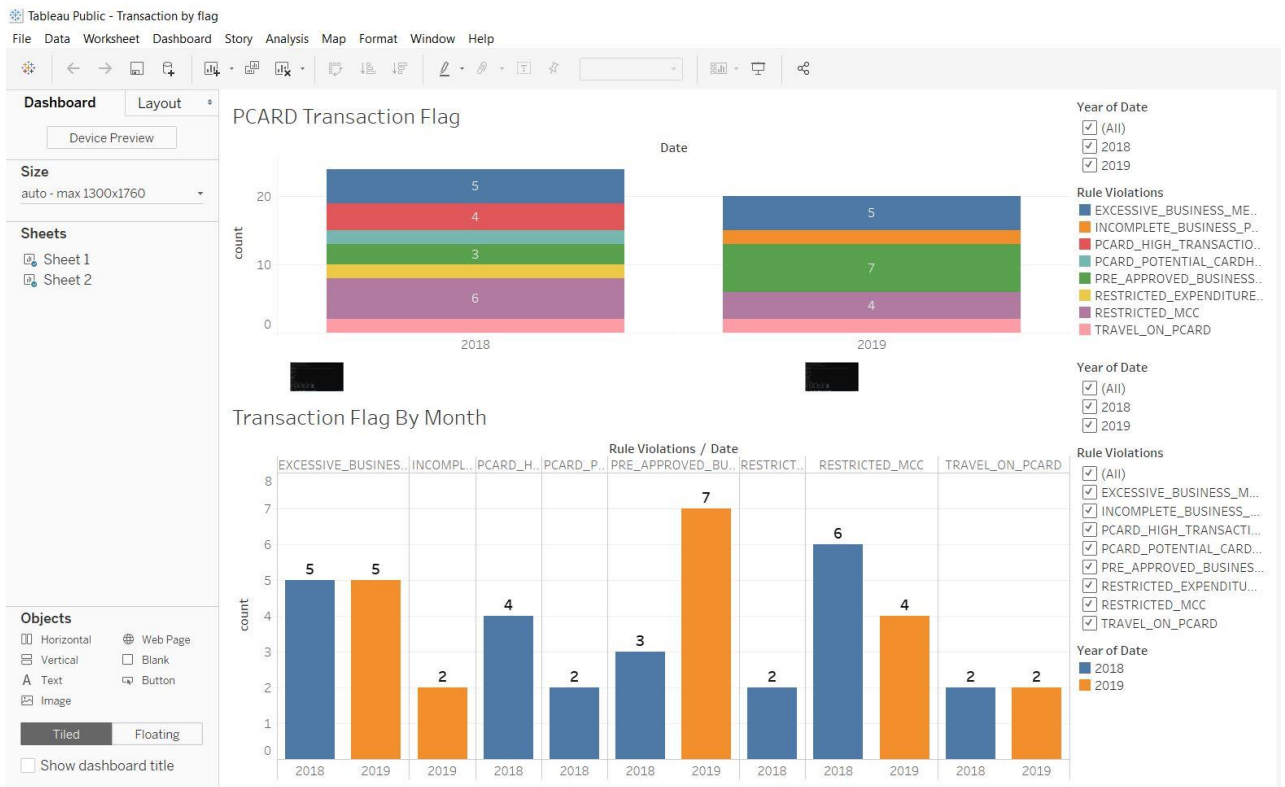


Fig 14. Report 2

Such reports has to be created using Angular 6 to integrate with the existing FAN UI. Following library was used to create the report.

3.3 Chart.js

Chart.js could be a network kept up ASCII content document library that causes you essentially envision information abuse JavaScript. It bolsters eight entirely unexpected outline sorts (counting bars, lines, and pies), and they're all responsive. As it were, you set up your graph once, and Chart.js will do the hard work for you and ensure that it's constantly clear (for instance by evacuating some uncritical subtleties if the outline gets littler).

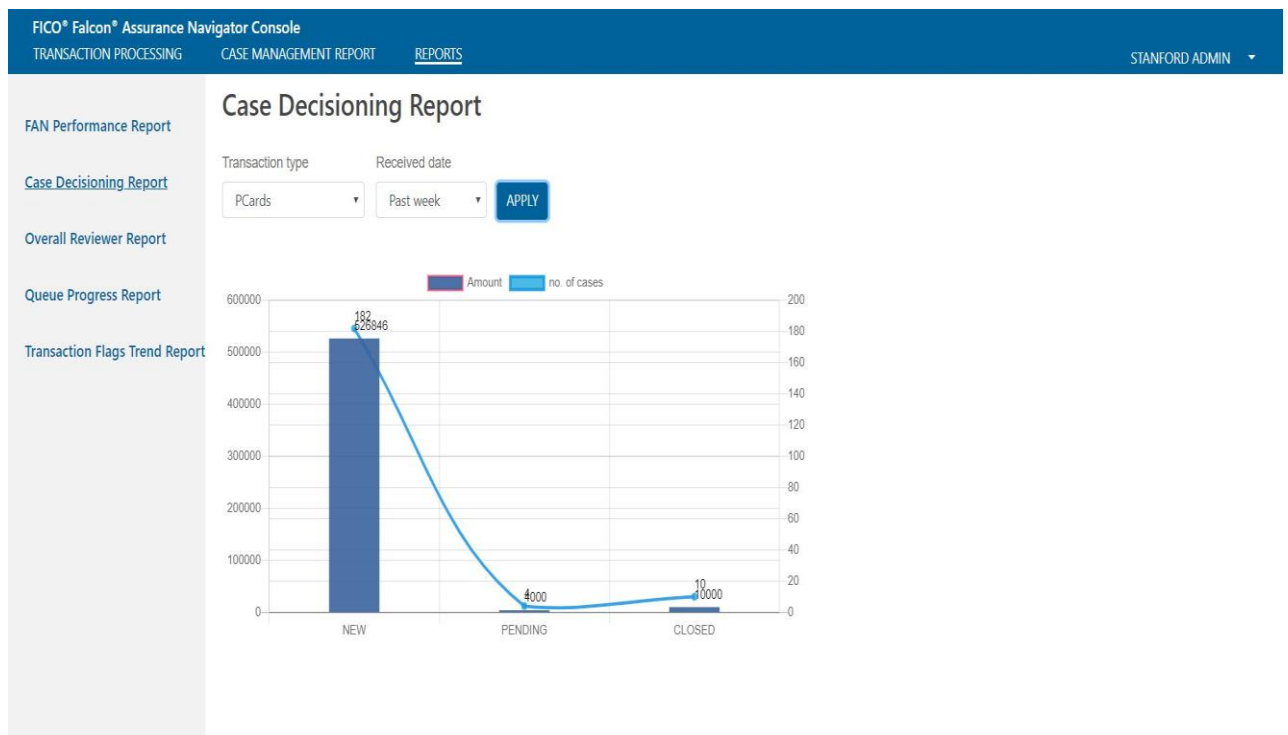


Fig 15. Reporting UI

CHAPTER 3 – Software Engineering Paradigm

Introduction

A product bundle improvement lifecycle is really a progression of steps, or stages, that offer a model for the occasion related lifecycle the executives of an application or bit of programming bundle. The strategy among the SDLC technique will shift crosswise over ventures and associations, anyway guidelines like ISO/IEC 12207 speak to forms that build up a lifecycle for programming bundle, and supply a mode for the occasion, obtaining and arrangement of programming bundle frameworks. The plan of partner SDLC technique is to help make an item that is productive, powerful and of top quality. When partner application is framed, the SDLC maps the correct preparing and decommissioning of the product bundle once it turns into a gift. The SDLC approach commonly contains the resulting stages: Analysis (necessities and plan), development, testing, release and upkeep (reaction).

Model Used

Agile Model is used.

Lite SDLC model is a blend of iterative and steady procedure models with concentrate on procedure versatility and consumer loyalty by quick conveyance of working programming item. Spry Methods break the item into little gradual forms. These constructs are given in emphasess. Every iteration involves cross functional teams working simultaneously on various areas like –

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.

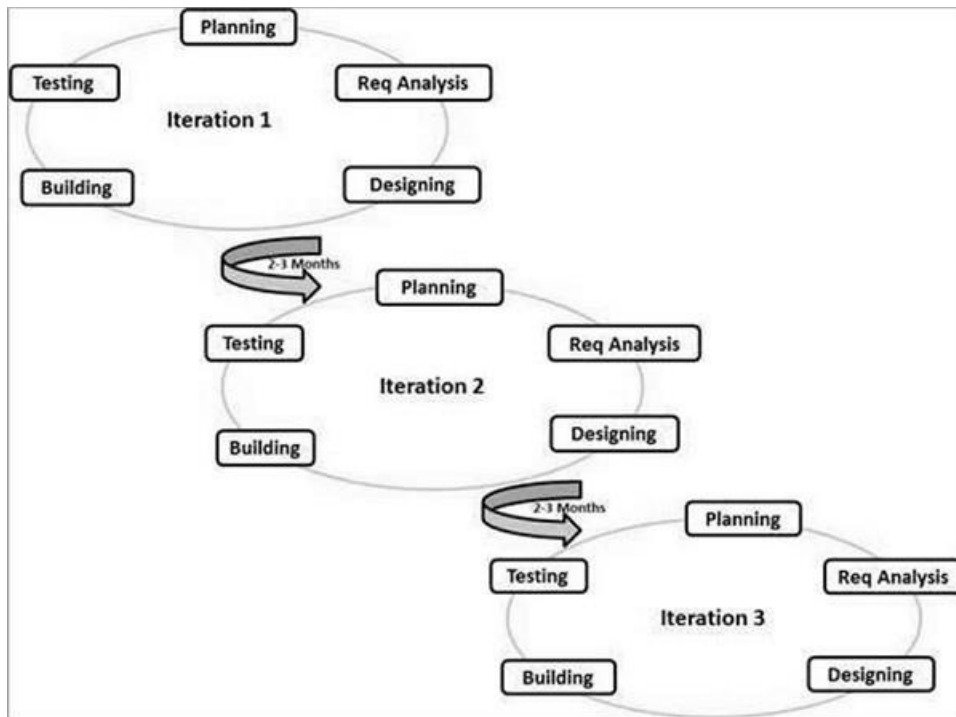


Fig 16. Stages Involved in SDLC

Stage 1: Planning and Requirement Analysis

Planning and requirement analysis is the paramount step in whole software development life cycle process. In this all the requirements are gathered from the client and accordingly a planning is done how to proceed further with these requirements. Each requirement given by the client is checked means it's feasibility is analyzed whether it would be possible to meet those requirements or not. This step must be handled with great care as if requirements are not clear and there is no proper planning on how to proceed, the end results could be very disastrous.

Stage 2: Defining Requirements

After the planning and requirement gathering phase there comes the defining requirement phase where each requirement given by the client are documented and the document is called SRS (Software Requirement Specification).

Stage 3: Designing the Product Architecture

Design of the product is based on the SRS document. SRS works as a reference document for the architects to design the product. Different architecture for the product are proposed based on the SRS and is documented in DDS – Design Document Specification. This document is

reviewed by the important stakeholders and they analyze various risk involved and other parameters like time constraints , budget etc and the best design is accepted.

Stage 4: Building or Developing the Product

Coding part comes under this stage of Software Development Life Cycle. Whatever design is adopted should have detailed description in the DDS so that coding becomes easy. One can refer to the DDS and start defining what kind of data structure to be used and other coding standards. The kind of programming language to be used depends on the type of project and what your company usually use. Different HLL used these days are Java,php,C,C++ etc.

Stage 5: Testing the Product

This stage involves testing of features developed by the developer. Various types of testing like White box testing , Black Box testing are performed. The testing is also done separately in various environments like dev environment , production environment etc to check If the product or the feature works fine after the deployment.

Stage 6: Deployment in the Market and Maintenance

After the testing of the product it ready to be deployed in the market. Sometimes the deployment of the product happens in a way defined by the company. Before releasing of the product it may have to first go through the testing in UAT environment (User Acceptance Testing). Based on the feedback the product is released or if any enhancement is to be done, that are done and released.

CHAPTER 4 – Analysis Document

4.1 Software Requirement Specification

4.1.1 Purpose

The purpose of this document is to gift a close description of the Falcon Assurance Navigator. it'll justify the aim and options of the system, the interfaces of the system, what the system can do, the constraints below that it should operate and the way the system can react to external stimuli.

4.1.2 Project Scope

This application is designed to detect frauds during reimbursement by providing an interface which will assist in receiving the transactions, scanning the document attached, calculating scores and based on all this making the decision whether to approve or reject the transaction.

4.1.3 Overall Description

Product Perspective

This application simplifies the cumbersome task of viewing all the documents referenced for reimbursement of expenses and based on that taking the decision whether or not to approve the transactions processed. The various microservices involved helps in scanning attached documents, calculating scores, triggering the rules and generating cases which makes it easier for the designated individual to approve or reject the transaction.

Internal Interfaces

The various internal interfaces include-

1. Case Manager API's:

Returns case information like status, transaction details and case disposition information, get all queues, create queues, get details of the queue, update given queue, delete the queue, return transaction information etc.

2. User Manager APIs:

Gets groups/roles/permissions, return details of groups, roles or permissions assigned to roles, return the wrapped object with id, remove the group or roles.

3. Stream Engine API's

4. Decision Service API's

External Interfaces

The system exclusively exposes a RESTful API as its external interface. The RESTful API is exposed by the Interface Layer in order to provide end points for data ingestion, file upload and management, and the Case Management and User Management service for their user interfaces.

Operating Environment

The software requirements are as under-

	Module	Version
Operating System	Microsoft Windows	Windows-10
Database	MongoDB	3.4.14 Enterprise
Java	SDK/JRE	1.7or Higher
Source Repository	Bit bucket	2.15.1.2-64-bit
Selenium	Selenium Web Driver	3.9.1
IDE	IntelliJ	3.2
Web Browser	Google chrome	8
Framework	TestNG	6.9.8

Table 1 Operating Environment

Since the software platform is based on Java, the hardware platform can be anything that supports a recent JVM.

HARDWARE SPECIFICATIONS	
CPU	Intel I-3 (or compatible)
Clock Speed	1.2 GHz
RAM	4 GB
Hard Disk	500GB

Table 2 Hardware Specifications

4.1.4 Functional Requirements

o Apply Validation Check

Transaction ID should only be alphanumeric. It should not contain any special characters(\$%^& etc).The special characters allowed are “_”,“.”,“:” In case it includes invalid transaction ID,the application must throw an error message that “Transaction ID can only be the combination of alphanumeric characters,”_”,“.”,“:”.

o The receiver module then post the transaction JSON with the following details-

o Transaction ID

o Transaction Number

o Vendor Specific Details

o Attachments, if any

o Attachments must specify “Internal document ID”

o If the transaction specific collection already has the transaction with the posted transaction ID,it does not displays error message, instead it replaces it with the new transaction posted.

o The attachment module then runs various OCR scanners to scan the attachment if present in the transaction.If any attachment failure occurs and if it is nonrecoverable, the transaction stops getting processed further.

o If the attachment failure occurs and if it is recoverable,the attachment recovery cycle runs upto its maximum threshold value and recovers the transaction for further processing.

o The context processor then calculates the score and triggers the rules integrated

with Blaze

o The Case Manager then generates the case and allows the concerned individuals to either accept or reject the transaction.

The High Level Architecture of the application is as under-

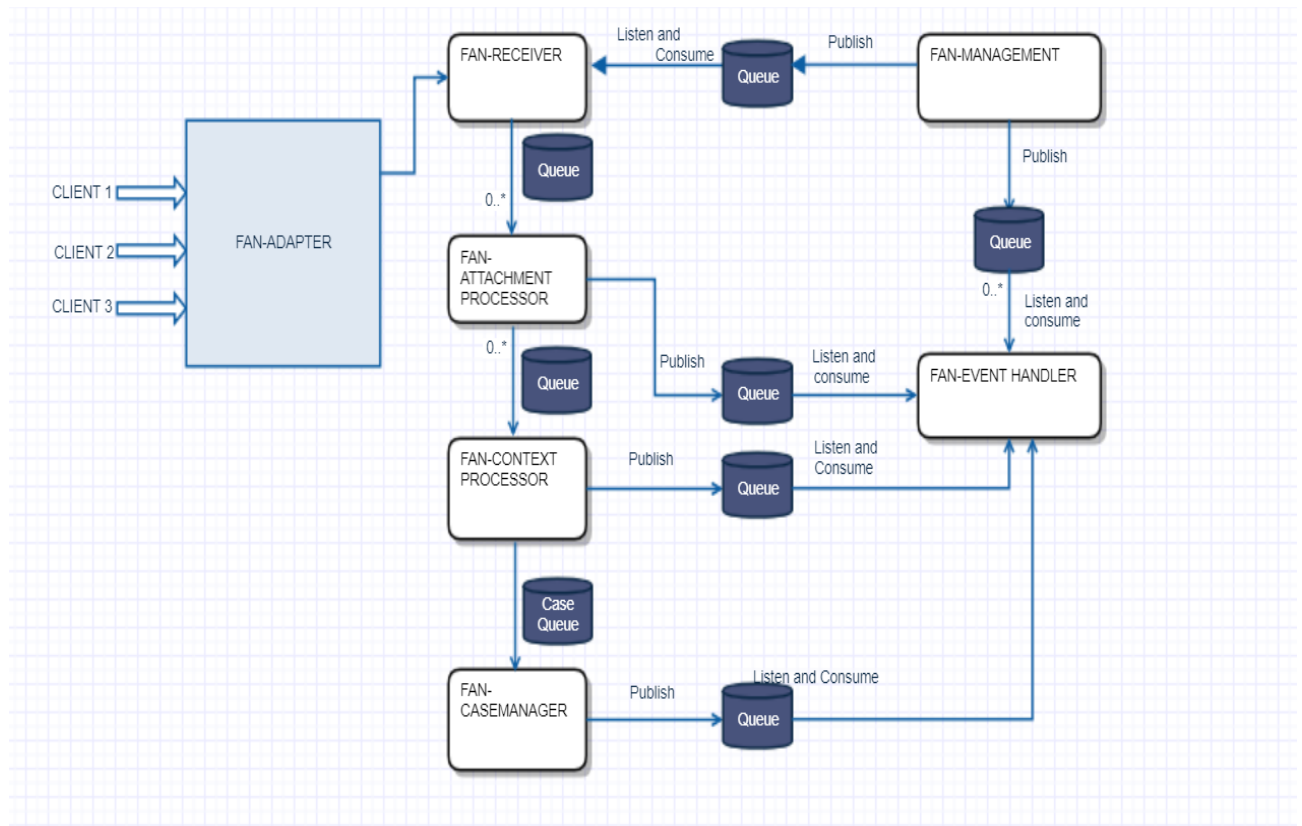


Figure 17 :High Level architecture of the application

CHAPTER 5 – Testing

Test Execution

Before beginning the take a look at Execution section the take a look at setting setup ought to be prepared. In take a look at Execution section the take a look at cases area unit dead within the testing setting. While execution of the test cases the QA team may find bugs which will be reported against that test case. This bug is fixed by the developer and is retested by the QA.

Defect Life Cycle:

Defect Life Cycle or Bug Life Cycle is that the specific set of states that a Bug goes through from discovery to defect fixation. the quantity of states that a defect goes through varies from project to project. Below lifecycle diagram, covers all doable states.

New: When a new defect is logged and posted for the first time. It is assigned a status NEW.

Assigned: Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team

Open: The developer starts analyzing and works on the defect fix

Fixed: When developer makes necessary code change and verifies the change, he or she can make bug status as "Fixed."

Pending retest: Once a defect is fixed, the code is given for retesting by the developers to the testers and status assigned to the defect is "Pending"

Retest: When tester starts retesting of the defect the status is moved to "retest".

Verified: If after retesting no bug is found then the status of the product is moved to “verified” by the tester.

Reopen: After fixing the defect by the developer and even after retesting any bug is found then the status is moved to “Reopen”.

Closed: When no bugs exist any longer then status is moved to “Closed”.

Duplicate: When the same defect or bug is found twice based on the same concept and status is moved to “Duplicate”.

Rejected: When developer sees that defect found is not of any importance and need not to be fixed , then the status is moved to “Rejected”.

Deferred: When the defect to be fixed is not of prime priority but has to be fixed in upcoming release. The developer mark such defects as “Deferred”.

Not a bug: If any defect does not affect the functionality of the application , it is referred to as “Not a bug”.

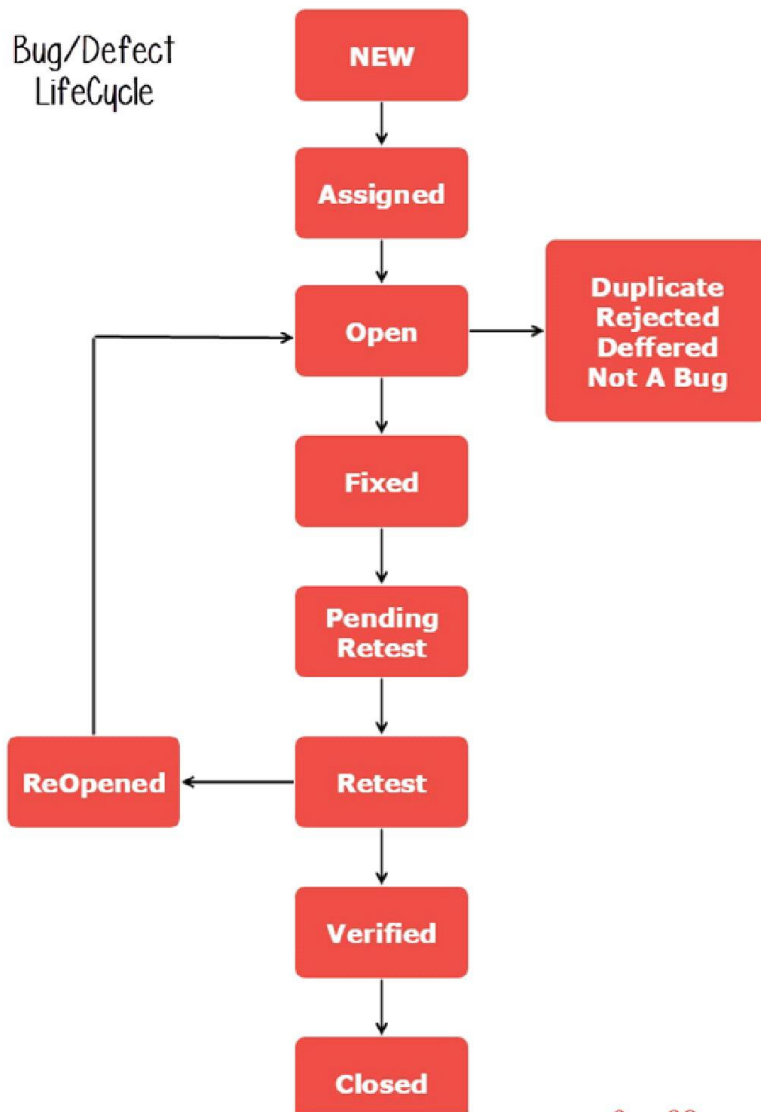


Figure 18: Defect Life Cycle

Jira:

JIRA is an instrument utilized for bug following, issue following, and venture the executives. The essential utilization of this device is to follow issues, and bugs identified with your product and Mobile applications. It is likewise utilized for task the board. The JIRA dashboard comprises of numerous valuable capacities and alternatives that make treatment of issues basic.

JIRA Issue:

JIRA issue would follow bug or issue that underlies the venture. When you have imported venture then you can make issues.

Under Issues, you will discover other helpful highlights like Issue

Types

Workflow's

Screens

Fields

Issue Attributes

Issue Types:

Issue Type shows a wide range of things that can be made and followed by means of JIRA. JIRA Issues are grouped under different structures like new element, sub-task, bug, and so on as appeared in the screen shot.

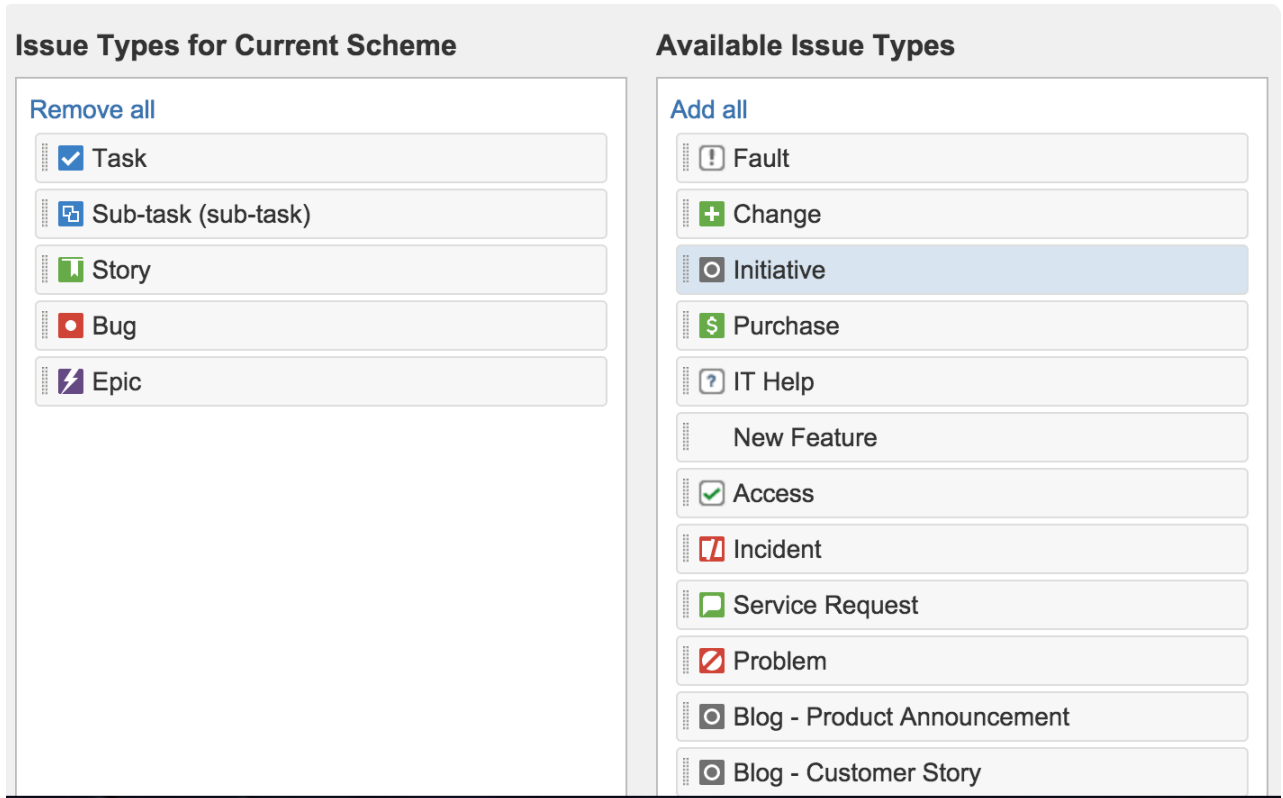


Fig 19: Jira Issue Types

There are two types of Issue types schemes in JIRA, one is

Default Issue Type Scheme: In default issue type scheme all newly created issues will be added automatically to this scheme

Agile Scrum Issue Type Scheme: Issues and project associated with Agile Scrum will use this scheme

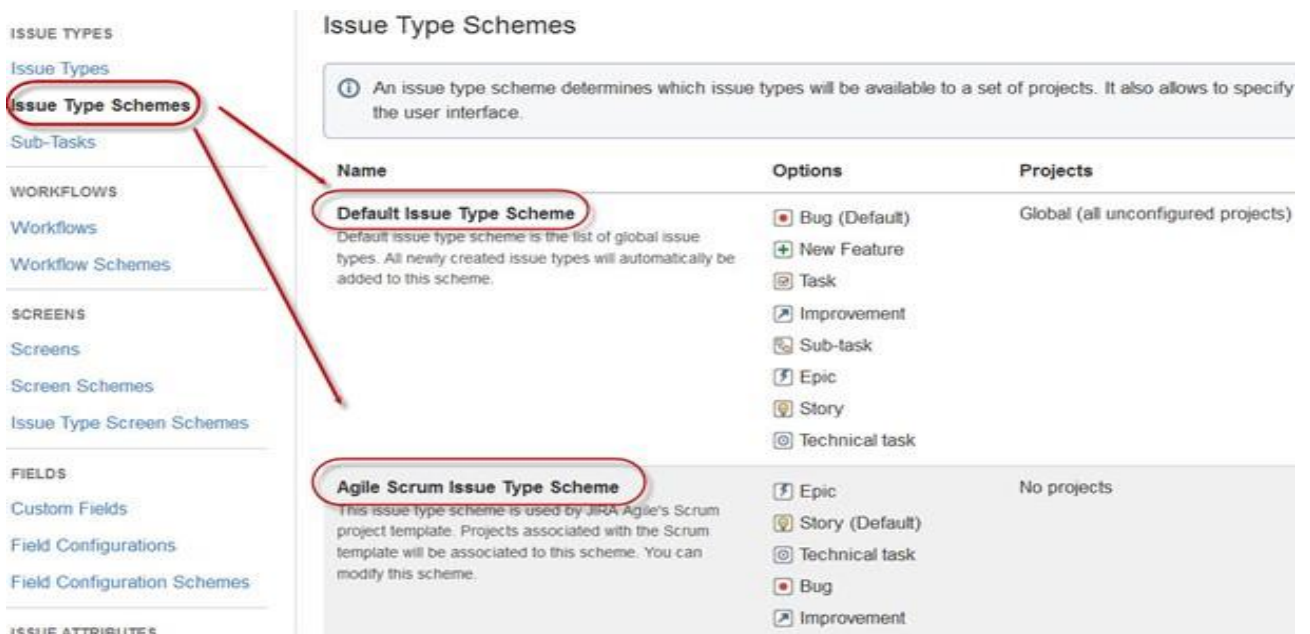


Fig 20: Jira Issue Type Scheme

Apart from these two issue type schemes, you can also add schemes manually as per requirement, for example we have created **IT & Support** scheme, for these we will **drag and drop** the issue types from the **Available Issue type** to **Issue type for current scheme** as shown in the screen shot below

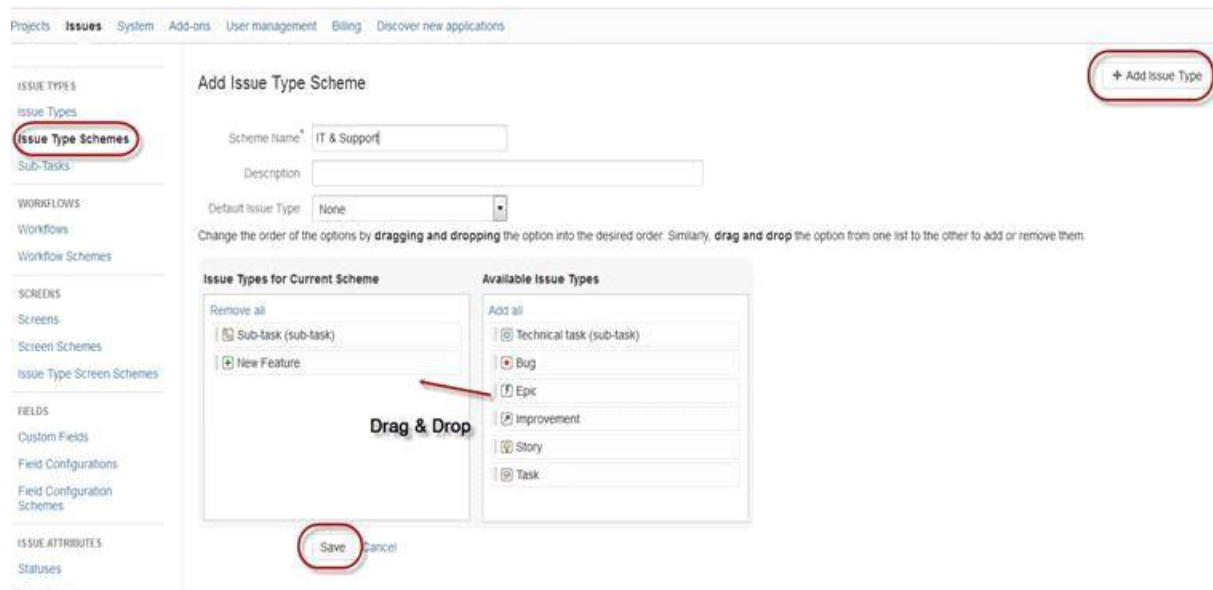


Fig 21. Jira Issue Type Scheme

CHAPTER 6 – CONCLUSION

FICO has many products out of which Falcon Assurance Navigator i.e. FAN is one of the product. One team of 17 people is dedicated for this product. Developers , Testers and DevOps people work in a process to deliver the features. Whatever new things to be added to the product or defects or any bugs found are listed in JIRA and tasks are assigned to the developer or tester or the DevOps person accordingly. The sole client of the product right now is Stanford University which uses this to detect any fraud in the reimbursement claimed by the students of the University on travel , food etc. The complete project is divided into six microservices . These six microservices follow a workflow as described in one of the chapters. I worked as a software engineer intern where my task was to create various reports using FAN's data integrate them with the UI. I also created one API using Java for my own use to generate the report. Development team works with Java 8 , Angular 6 and other java frameworks. The complete object model is written in java8. This project is not meant for the large enterprises solutions. It is just for the small enterprises solutions. It is specific to the university employees. For the future, we can use this project can be taken as the template and can be customized based upon the client's requirement. This application can be extended to accomplish the requirements of different university employees as well.

REFERENCES

<https://intellipaat.com/blog/what-is-tableau/>

<https://www.chartjs.org/>

<https://www.guru99.com/software-development-life-cycle-tutorial.html>

<http://www.guru99.com/software-testing.html>