INTRODUCTION

**---------Morgan Stanley background--------**

As my project started I was introduced to various technologies, software and tools used by the team. The team uses a tool to store necessary calculations and do reporting on the received data. The tool stores data from different soft wares do some calculations on its own and acts as User Interface to show the obtained data in a user friendly way to the client who accesses this data from different locations and different environment. There are different calculations that are done by the main software.

The calculations are mainly Profit and Loss calculations which are shown as a change in value of the trade over a time period. This calculation is done on the basis that the trade value that was on the last working day minus the trade that is on the present day. The calculations are evaluated in the ripple and stored in the database.

For instance, Profit and Loss = Calculations (T) – Calculations (T-1)

Where T is the Time where calculations are done for today and the last working day. The last working day excludes all the business holidays.

Here, calculations are of many types like the capital that is being made, the capital that is assumed to be made in future, the necessary fees and so on. These calculations together are used to calculate the Profit and Loss that is further stored and analyzed by the monitoring tem.

The calculations also include earned revenues and incurred expenses that are summed to get the final total which is calculated as the final sum to be used as the estimation to see what all calculations are going wrong. These calculations are analyzed with calculations that actually occur to see the change or the change in the happenings of the trades.

There are other calculations that are done on the main software which requires the knowledge of the financial term Options. Options are a financial derivative sold by an option writer to an option buyer. The contract offers the buyer the right, but not the obligation, to buy (call option) or sell (put option) the underlying asset at an agreed-upon price during a certain period of time or on a specific date.

One of them is the Kappa effect which tells investors how much an option's price will change for a given change in implied volatility, even if the actual price of the underlying stays the same.

Then there is one calculation called theta which is a measure of the rate of decline in the value of an option due to the passage of time. It can also be referred to as an option’s time decay. If everything is held constant, the option loses value as time moves closer to the maturity of the option. Maturity date is the date on which the principal amount of a note, draft, acceptance bond or another debt instrument becomes due and is repaid to the investor and interest payments stop. It is also the termination or due date on which an installment loan must be paid in full.

There is another calculation called Gamma which is e rate of change in an option's delta per 1-point move in the underlying asset's price. Gamma is an important measure of the convexity of a derivative's value, in relation to the underlying.

Apart from this, there are other calculations that take place in the main software that are stored in the database. There are some internal calculations that take place here and require some parameters from the other soft wares to make calculations. So, the main software takes Profit and Loss calculations as well as some parameters from other soft wares and some risk and Profit and Loss data from other soft wares to store and display the data according to different regions to the monitoring teams according to their needs.

This is the main software my team works on. Basically, it takes input that is Profit and Loss calculations and risk analysis data from different soft wares and does some calculations on its own, as described earlier which is then put into the database and the necessary monitoring takes place according to the requirements of the team. The other software performs all sorts of necessary calculations and analysis depending on business logic. The main software is basically old school software made of C# GUI, Java server and a database of Morgan Stanley which essentially is a RDBMS.

--Diagram of GUI, Soap SERvice (Simple)

It also contains a cache with a filesystem used for necessary calculations which will be described later. The request from the main software is sent to the Server which is a Java server from the GUI using the SOAP interface. All the data that is obtained from other software/systems is stored in the cache. The technologies mentioned above are describes in detail below:

C# GUI, Java server and a database – describe - RDBMS

The SOAP interface, GUI

As mentioned earlier, there are many soft wares that perform necessary calculations and pass on data to the main software. Here is the description of all the soft wares:

1. In this software, the data from the last working data till the present working day is analyzed. The calculations are mainly Profit and Loss calculations and the risk calculations. SWAPS, BONDS, FUTURES.

There is this file storage to store all the files required for the calculations or the files that contain calculated data. Also the calculations take place on a server which is based on a programming language ------description------- .

All necessary tasks are submitted through the API for an intermediate between here and the main software because this API acts as an interface between A+ programming language and the Java/ C# systems. It uses SOAP interface to communicate.

File Storage

DB Server

Calc Server

Client

The API helps getting information of the data using some data entries’. Besides information, it also helps to access user’s data. The collected data is sent to the main software using this API. This software is based on Unix/Linux. The data that is obtained overnight is stored in the cache. This is all handled with the help of a broker which acts as a manager to manage resources and helps in distribution. Also, an engine is used which is helpful in the parallel execution of the numbers. All this information gathered from this software is sent to the main software which uses it for the further analysis.

1. Other software that sends its data to the main software is sending the risk calculations to the main software. This is a kind of library that is used for pricing purposes. It runs as a grid that calculates and manages many engines. This software is written entirely in C++ programming language which is ----description of C++----------

This software runs on Linux and uses optimized algorithms to make it better. It is also used via other soft wares by just referencing to the name of this software.

This software is mainly uses a grid distribution which consists of a grid server for the client to connect to, a virtual grid to perform specific function, some individual and virtual machine grid nodes. A node runs specific number of tasks per time and performs a proper connection on both sides to do some work. There are some tasks done individually through the nodes. This is the way; this software is working and doing some calculations to provide data to the main software. Also, as it is a kind of library, it is used by other soft wares to get some parameters or data from it.

The data is put into the main software from this internal software using middleware that is used to deliver asynchronous messages between many platforms and languages. The messages are read by one process that reads messages from one side and sends it to the main software. It acts as an intermediate to read and deliver messages. This process is written in A+ programming language. It takes object of this software and converts into the table which is sent to the main software. The risk related data is sent by this software which is then used by the main software.

--------------- diagram----------------

This software sends the data that is completed or is recovered to the main software. Now, the main software listens to the incoming data. The data is again stored in the cache for reporting in the main software. This data is then used by the team for monitoring or analyzing.

1. There is software that puts data into the main software. This software sends risk data to the main software. This software is built on top of the calculations infrastructure that provides development and run time environment for this software which makes it easier for the systems to share code and resources. This infrastructure improves the performance of many systems as it helps to become a model for many applications, acts a framework to manage many models, represents code level for some models and acts as an IDE. The infrastructure diagram is shown below.

------diagram of optimus-------------

The software is built on this infrastructure does all its calculations related to risk on this infrastructure and when these calculations are complete, the data is sent to the main software using the SOAP request. The data is stored in a CSV format for other systems to be used. This is handled by a process which can be done with either file based or SOAP interface. Basically, the process is such that the data file is written and when ready, a file to notify is made. This file will start the loading of files which will start the processing by renaming the files. The successful files are sent to a folder and failed files are kept aside.

------diagram of CSV importer----------

For the main software, the incoming data is listened with Java and the data is translated according to the needs or the way main software wants it to be and then stored again in the main software for further processes.

------------- diagram of RICE----------

These are some of the soft wares that send data. Also, there are other risk calculations that are put into the main software apart from what is discussed above.

Now the main software that accumulates the data from different soft wares and some calculations on its own follows some process to get data into the database and use it further. The software is such that to make it highly available and scalable to make its performance better. The internal software is managed using a load balancer that helps in further improving the data management.

There are different components of the main software which includes the soap service, the submission of risk and Profit and Loss tasks.

1. Soap Service:

The Graphical user interface of the main software requests data from the main software using a SOAP service. This SOAP service has an end point which receives the data and helps in the binding of the SOAP service with the Java method.

There are many servers used by the company which is managed by an application which is used to remotely control and monitor applications. There is a real time monitoring that happens with the help of this application. It allows users to control, inspect and manage different applications and different hosts.

So, as the request comes, the user interface randomly decides to go one SOAP server to process the request. As request reaches the request can be sent to either processes or to submit tasks like Profit and Loss calculations or Risk calculations or to get the status of the tasks.

------diagram of Soap Service------

As mentioned earlier, there is a load balancer which helps in distributing the requests evenly among different SOAP servers.

1. Task submission:

The task submits that are requested from the SOAP service are sent to one of the classes for the further process to take place. Theses tasks can be Risk calculations or Profit and Loss calculations or any other calculations which are then sent to the broker. Other calculations are sent to another system which manages such kind of calculations and is used as a submission and monitoring tool for such requests.

The Risk and Profit and Loss calculations are sent to the broker who is managed by the main software which further sends these requests to an engine. This engine is different for Risk and Profit and Loss calculations which will be described later. This engine will ask other soft wares for the calculations and will also use the calculations made by the main software itself for the processing

-------diagram of this -----------

1. Profit and Loss submission:

There is a class that handles the requests that are made particularly to this category. The request is sent to the one of the engines by the broker who picks one engine to perform the calculations. The engine sends the request to the task manager which sends it further to the calculations handler which are further sent to the Calculations factory where the calculations take place and are sent back to handler which stores these calculations in the database and are used when the request is made.

-------diagram-------------

1. Risk submission:

In this submission, the requests that are sent to the broker which picks one engine to perform tasks. The engine sends the request to the task manager which sends it further to the calculations handler which further sends it to servers to obtain data for this category after which some files generated and are send to the handler. After all this, handler stores these calculations in the database.

-----------diagram-------------

1. Other submissions:

These requests are sent to another software that is describes earlier and this software gets input from all other internal soft wares that do the calculations and give the result back to the handler and again stored in the database.

------------------screeenshots of ripple with explanations------------------

**Tool:**

I started working on a tool which is a kind of health checkup tool which is required to check the main software, the databases, networks and other functionalities to check if everything is working fine. This tool is run in the early hours of the day to check if the particular application is up and is ready to perform operations and can be used to perform business logics.

This particular tool uses one internal library for all processes to take place. It is a kind of API which is used during the development of the tool to check the main software. It is used as a building block such that a series of checks are performed in some order. These tests are performed on various parts of the system to ensure the right of the status of the main software. It will check if the test is passed, failed or have a warning and sends a report to the user every morning.

This library works on a concept of tree like hierarchy of different tests which are accomplished on the application and the ecosystem of this application. All of these different checks are together agglomerated to get the result globally.

One check is a task that is repeated to return a status that indicates weather the outcome is Pass, Fail and Warn. These checks are indicated in the result in the form in form an email which is sent to the team on the basis of results that are obtained and are colored on the basis such that if Pass than the color is indicated as green, if Fail, it is marked and Red and otherwise if warning, it is indicated with Amber color.

There are some examples of these tests.

1. Ping a computer:

This test is executed to check the connection of some network which includes pinging a particular host and a TCP connection to check a particular host an d a particular port on that host.

1. Test a database connection:

This tests as indicated in the name, is used to check the database connection and see if the database is reacting to some query.

1. Test a Message Queue.

There is this test to check if a particular message queue is performing its actions. This depends on the soft wares that are using this tool.

1. Execute some business logic

There are also tests to check some business logic behind some soft wares by sending some check from this tool or a connection to some functionality of the main software that returns a particular value which is checked on this tool and gives back the required result.

There are some post processes that take place after the results have been calculated. The postprocessor takes the end result and does some processes with it by running a tree, as described earlier. These post processes are run in a chain such that it picks up the result of the previous one and gives the result in the tree like structure.

Some examples of the post processes are:

1. Generate a report:

A report is generated out of the processes which is send to the team.

1. Send an email

The report that is generated is sent to the team via an email which is used by the team to easily monitor/ check what all is going on in the main software.

1. Save results to the tool’s database

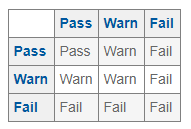
The results that are obtained are also stored in the database.

1. Publishes the results

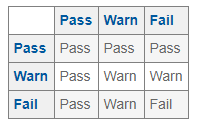
The results are stored and published when required.

----------------diagram of above----------

The library executes different checks and does them in groups. There are logics applied to the results. It checks AND-like logics such that it returns PASS if every test is passed, returns WARN if at least one test is warn and no failure and returns FAIL if at least one test is failed.



Similarly, It checks OR-like logics such that it returns PASS if at least one test is passed, returns WARN if at least one test is warn and no test is passed and returns FAIL if all tests failed.



The main class of this library is started such that there is a context file written in spring where different instances are defined which also must have automatic start property set to true in order to start. Also, a parameter for the Checks to be executed should be defined. There are also other ways to start the main class then using the automatic start option.

So, to use this library in one project, it should contain configuration files written in spring containing spring beans. For instance, the tool that is used by my team uses the same concept of using different context files as many needed. For example, if database check is needed to be performed it contains one context file that contains all the databases that needs to be checked. A particular check could have child checks or zero child checks that are executed weather the parent check returns pass, fail or warn.

And after the checks are being executed and return some results, the post processing is started and the results are processed in the form an email using the library class. There are other things that are managed using the template file that is used to make email look better by using some properties for processing an email.

To start with the project description, I would like to introduce Spring.

-----------spring desiption-----------

For the tool that our team works on is structured in a way that the spring configuration file that contains other child context files that are called with the help of a class that uses the main library to perform some checks. Further, each context file has its own functionalities. For instance, the context file for the database check is made such that it checks each database that is required by the team and sends the query to the database to check if it is working. This database is connected to the class in the library that will perform checks and gives necessary results according to the results obtained after querying the database. It runs a program to check the database.

There is another check that is performed by the team. This check is for checking of the functionalities of the main software. This includes checking the tasks performed by the main software. The main software performs some sort of calculations as mentioned earlier. It takes some tasks and submits it using some functions in the main software. These tasks help in calculations of the Profit and Loss and Risk which are further used by the monitoring team. Now, we require making a system to check if these tasks have been submitted and the calculations are being done so that it is for the team to evaluate the main software. This will allow my team to check the systems if working fine by letting them understand the system through an email rather than going into the whole thing and seeing what is going wrong. So, to reduce the overhead of going and checking tasks for each submission and checking each database, this check library is used to implement the tool to get better insight of the work. This reduces the work for the team. So, in this case there are checks performed which are particular to one region and environment. For example, every time a check is performed, it particularly is for one region and environment and tasks in that region and environment. Also, the databases in that region and environment are checked.

Basically, a configuration file is run which runs a list of different checks. These checks have their own spring configuration files that send the check request to the library. In case of the tasks that are checked inside the main software. The check request is send to a class that will try to submit the tasks first which functionality is send to the main software to check the tasks if submitted or not. These checks have children which are the Profit and Loss and Risk calculations which are required to check the tasks that have been submitted to the Profit and Loss and the Risk. These checks returns result as Pass, Fail or Amber on the basis of which these are sent to the report. These act in the AND-like structure where they are marked Red if at least one test fails and marked green if all test pass.

Now, the results are sending as report which is send to the post processor for the email to be sent to the team every morning to see the status.

--------- diagram of structure of how is working------------

------diagram of initial result of RFB and explanation------------

**Problem statement:**

The first thing that I was required to learn was about the platform and architecture that my team works on. There was an introductory session given to me on the main software that my team works on. After all this I was asked to work on all these soft wares on my won and learn more about the architecture and get a deep learning on what is going on the system. This included learning everything related to the functionalities of the main software and getting acquainted with the existing code and learning about the system.

The next problem was to get all the permissions required for the software to run on the system that was provided to me. This included getting into the team by getting permission to get into the different team groups and getting permissions to download many soft wares that are required for the team to be worked on. To take these permissions various approvals were required and I was asked to talk to people to get approvals because different people or teams are responsible for different approval requests.

After this, I was asked to get proper knowledge of the code and try to run and debug it, to get better insight of what is working inside the main software and what are the different functionalities. I was asked to go through the code which is written in Java programming language (described earlier) and run on IDE IntelliJ. The GUI is written in .Net (described earlier) for which I was asked to install visual Studio which is also an IDE. I was asked to run these codes and get more empirical knowledge of the main software.

The next issue was to learn about the tool that is used to check the main software to see if the functionalities are working fine. Apart from this, I was also asked to run the code and get the results to better get knowledge how the results are being obtained and debug and get to know, how the software is working. This tool is written in Scala which was also a task to learn a new programming language and code in it.

After this I was asked to enhance the functionalities of the tool which started with implementing small things first and then go on to moving to change structure of the tool. Some of the steps include:

1. First of all, to get acquainted with the code I was asked to implement a small network check into the code. This check included Ping check that gets that connection from the host see if it is connected or not and TCP check that gets a connection from a host and a port to see if the connection is made or not.
2. Then I was asked to check if a particular functionality in some server is working. It was such that I had to check if the functionality of the software is returning some data. The data that was returned was required to have some value, which would return FAIL if there was not data returned in that functionality and would return PASS otherwise.
3. I was asked to get a better understanding of the entire code and figure out the better how the things can be done. As in, I was asked to integrate the structure of the tool in a better way such that there are fewer hierarchies in the result and that the code has to go through lesser number of hierarchies to give back the result. This included sending the data as an email for all the regions and environments rather than for one specific region or environment. There was combination of many regions along with many different environments that were supposed to be sending via mail.
4. After this, I was asked to improve the template/ design of the tool such that the email that is sent every morning has a better view and gives out only some information rather than delivering everything that is not even required.
5. Last thing in this was to get the results as PASS as much as possible to see everything is working fine.

DOCUMENTATION:

After improving the working of the tool, I was asked to write a document for the same that included all the details of the working of the tool. As it is said, “With a great tool there comes great power; however, to fully unleash this power it takes commitment and a drive for excellence. Real success using any tool starts by understanding why its use is so important”. So, it is really important to have a documentation for a tool so that we can go through it whenever needed to get to know what is going on in the project if it would ever need improvement in future by some other developer.

I was asked to write a document so that the procedure of the tool is easily understandable. Other developers who in future might have to use or maintain this tool could understand the toll in the most efficient way. Even if I need any help with the tool ever, I can get back to this document and understand what is going on in the tool and what the process is. This would also reduce the training time for the developers who are willing to work in future in this team and are required to work on this tool. Apart from this, it is always professional to write a document. As Morgan Stanley is a professional organization that requires employees to be professional at all times which also reflects in the way that the employees write the code or write the documents required for the tools and soft wares.

Ripple GUI:

The next task that was supposed to be done was to improve the performance and efficiency of the graphical user interface of the main software which is a C# based user interface that sends requests to the Java based back end through a SOAP service.

There are different task submissions going on in the software at one time and that it requires some time to process all the submissions which results in decrease of performance and it is difficult for the users to cope up with this. So, to increase the performance and to make it better for the users to access, I was asked to improve the quality of the user interface such that by adding another window that would help in reducing the overhead.

It was basically getting all the regions and environments in one window and looking at it in a most efficient way to better get an idea of what is going on and where all the calculations are and creating columns in such a way that there are not many windows for different items opened at one time which would increase the performance of the main software.

Script:

There was another task that was assigned to me which was related to the run configurations that are used to run some specific programs in the software, which requires for software to run, has its own functionalities.

A program requires its own parameters for it to run. These parameters are defined in the run configurations which different xml files are containing all the useful data and some parameters that might be specific to software or may be common to many configurations. It is with the help of these configurations that we are able to run a program.

Now that the configurations are hundreds in number, it is difficult to manage all the configurations in case there is a change that is required to be made in one file. So, to reduce the effort of going into each script and making the change, it would be feasible to make change in one file and that every file that requires automatically gets the parameters changed.

So, I was required to create a script such that I can make a change in one part of the file and all the xml files get the change and overhead of going into each configuration and making the change is reduced.

WORK DONE

As I started my internship, my first task was to learn about the team, the main software that is being used by the team that required for me to learn all the architectures of my software and other soft wares that provide data to my software. This included learning about the databases and also the code that is used for the main software in my team. I started by understanding all the necessary databases, learning about the architecture, learning about the firm and getting into the code of the main software.

Along with all this, I started downloading all the applications that were required for working in the team. This included all the necessary soft wares that were required for my program to run. For instance, I downloaded the IDE IntelliJ. Also, it required for me take some permissions while downloading some stuff because of the security in the firm. This took me some time, contacting people who were responsible for the approvals. After all this, I successfully configured all the required tools on my machine with the help of teammates and could now get to the main software that my team works on.

-----image of all downloaded apps from myapps-----------

This helped me to get into the next step of getting to learn the architecture of the main software that my team works on. I went through the documentation of the main software that is used and structure of the software. I tried learning as much as possible of the software that my team uses and business logic is implemented into an automated form. This gave me more explanation on how my software is working and how the technologies are used in this software. More detail of the learnings will be given in the Literature Review to cover detailed perspective of what is learnt in the main software. Learning about the technologies used in the main software and the architecture of the main software itself helped me a lot while working for the team.

I had to go through couple of documents to get everything configured on my computer and get a better learning of how to get the main software on my computer. So, with the help of all these documentations and my teammates, I successfully configured all the required tools on my machine. With this, I started looking into the main software relating architecture and the code that is written in Java. I started understanding the software in a better way as I started looking more into it. I tried learning as much as possible of the software that my team uses and business logic is implemented into an automated form.

With all this, I had to make sure that I get a good understanding of all the technologies, databases and so on that is used by the team. The architecture of the software was explained to me by my manager and other team mates. This was necessary for me as a beginner to get acquainted with it as it is very old written program that needs a good understanding to make myself better relatable to it. I did my best to understand all the necessary architecture and code related to it to be ready to solve problems related to it in future.

After this, in my team we use JIRA to assign tasks to the team, so I was asked to learn more about JIRA and how it functions to get a better understanding at assigning tasks. JIRA is a tracking tool for the issues. It provides various functionalities for the tracking of bugs, issues, project management and so on. So, it helps the team get a track of the issues going on in the team. One team member assigns the task. It may or may not be to some specific person. There is a procedure to follow to let these tasks taking place appropriately.

Since it is required to follow specific way of storing and maintaining the code, there should be a procedure to store the code and keep a track of it. Being a professional company, it is required that it follows some standards in this case. So, in Morgan Stanley, GIT is used to keep the track of the code and keep it going. So, to get a better understanding of the tool, I was asked to learn different commands of the tool. So, I started learning about the tool and how everything works so that I can work on it in a better way. I started with learning basics of the tool and then moved towards learning in detail about the commands used and other features of the tool. This is what led me towards pulling my project into my system, I managed to clone the project into my machine and make fork on the GIT software. I checked out a branch to make changes in the code. I started working on the software and understanding the code.

I was introduced to a new this tool which is a monitoring tool that I was required to make enhancements into. This is used to manage the main software and acts as a health checkup tool for the main software. It is used to check if databases are connected or not and the tasks that are required to be submitted by the main software are successfully completed or not. This tool helps in checking the status of the main software and sends an email in the morning everyday giving a report to the team.

I started by learning Scala programming language as my project was based on it and I was required to start coding in Scala. It is a programming language that combines object oriented and functional programming and is a high level programming language. So, I started learning about Scala and some great features of the same. Later, I started getting into the code and tried running it and understanding it. As I started learning about the code and how the tool works, I tried making small changes to check if it is working fine. For instance, I changed the email address from the entire team’s email address to my email address so that I can test it. I ran the tool and got the following result for first time:

-------image of RFB-----

I was asked to add functionalities to this monitoring tool so that we can add more checks into it and get better insight into existing software’s. I started by writing a simple Hello World program in Scala to get to understand Scala better. I tried some functions in the Sample Hello World program to get some hands on experience in it before getting into the main part.

I started by adding some network checks into the software that included Ping check and a TCP check that had some host and a post to be connected. This I started by writing a configuration file which would take different hosts and ports and would give a result in the form of Email such that it checks the host and ports. It will give result in GREEN if the network is connected to a particular host and FAIL if the connection is failed. The initial result was as follows:

---image if RFB----

After this I was asked to check some functionality of a server that would return a valid value. The check was to make sure that the function returns a non-empty set. I started by using a function in the main software to check this server and its functionality. After this, I started implementing this function to use the library so that I can get a PASS or FAIL after the functionality is run. I did in a way such that when there was output, I could check the result if the returning data was non empty. For that, I checked the functionality and could get the result green if passed and red otherwise. I did it in the same of writing a configuration file and making a list of checks. These checks included the server functionality check. The result that I got was as follows:

------image of RFB--------

The next task was to change the structure of the monitoring tool such that the entire structure of the code must be changed to make it more automated. Till now, the structure of the code was such that the result was coming from one region and environment. This would not help the team to monitor all the regions and environments. So, to get a better look into the tool and operate it in better way, I was required to add this feature. I had to add more value into the tool to make it look better and give results for all the regions and environments at once.

I purposed to do something like this, based on my understanding:

-----diagram of what I thought-------

So, I made a discussion with my teammate and we agreed on it and I started working on the same. I had to make the structure such that it is readable and looks fine to the team so that if there is some problem, it is easily predictable.

I started by changing the entire inside of the code by changing different classes and using different methods to solve the problem. It was little tricky task as it needed to change configuration dynamically. I started to debug everything and started asking questions from my teammates and my manager that helped me a lot. As I started, it was challenge for me to change the entire code and make it into something which is not specific to one region and environment. I changed the configuration files entirely and managed to get them into one file, so that I can reduce hierarchies and accommodate the entire thing autonomously rather than making hundreds of configuration files which are specified to many regions and environments. After all this, I changed the configuration file and tried some other configuration file for a trial and got some satisfactory results which gave me rays of hope to go on further with the project. Now, I started my implementation towards the actual thing that took me 1 week to completely come out with the result that I wished. There were some checks that were already implemented in the original software that gave me hard time. There were some checks to see if databases are working fine which did not work after I changed the whole structure of the project. So, I tried something out of the boxes and was able to get some successful results.

As I started and made some changes, I got one result as follows:

----------diagram of RFB-----

As I moved forward, I was not able to get results for the server as I was getting into one region and environment. Basically, I had to separate some checks in a way that some are general and others are based on location specific. This was one of the major challenges for me and I asked my teammate for the help. So, with his help and some of my efforts, I was able to complete that task as well.

As shown below:

----------diagram of RFB magic check RED-----

So, I started by solving this problem, with the help of a teammate. So, I had to change the entire thing into something which is two parts. One part was for those checks which were common and others were for those which were specific to one region and environment. So, the next step was to make a configuration file set parameters for the configuration file that was specific to one region and environment. After some efforts, I was able to get some results and the structure looked like in below:

----diagram of RFB---

Now, after implementing this, I was asked to change the template such that the way we obtain result. It should be such that I could get the result in a manner that the useless things not related to the project. I was asked to remove all the stuff that is inclined towards the library that is being used for the project. I started by changing the template in the form that I can get a better view of it. It took some time to get a better view of the template. The main challenges were not use library’s template and use the template that is used by the team. So, to get to the result I had to go through various issues. I tried making a script which later I thought was not a good idea. So, I started by using a template into my tool and avoiding the one used by the library. I started making it look better as I proceeded. I changed the paths on the configuration files and tried to make it happen. Below is the way it looked after this:

----diagram of RFB-----

After this, I had to improve the structure in a better way such that there were some places that it was giving result that was wrong. For instance, if all the checks one check in a part is RED, it should give a RED but it was giving GREEN. So, I spent next couple of days working on this and improving it. And after some time, I was able to get some results as below:

---diagram RFB----

Next, there were some problems with the code that was already there in tool which was creating a problem. So, I took help from my teammate in improving that part. After all this, I got something like this:

---diagram----

As I was completing my code, I was asked to commit my changes into the GIT so that everyone can review my code and comment on it if something is going wrong. I made a request to review the code that I made to my team. There were some comments about the coding conventions, code changes, questions about the project and so on. I had to answer the questions and improve the standardization of my code.

I have learnt how to assign a task to me in the software used by the company. This helped to get know better how the process works in the organization. Apart from that, there used to be SCRUM meetings to discuss everything happening in the team. This helped me understand what others are working in the team, at that point of time.

My team follows Agile process to complete the tasks to make software look better each day. More explanation about the Agile process will be given in Literature Review Section.

At the end for RFB : draw the diagram of working (like a class diagram) and explanation of changes

I was also asked to make the User Interface part of the main software that was introduced to me for which I cloned the project of the User Interface part of the main software. Visual studio was downloaded and the code was cloned into it. I did some debugging into the code and was required to improve the performance of it.

Although, due to lack of time, I was not able to start doing this task a sit required a lot of efforts and time to start and implementing it as I was required to start by making mock ups for the project.

Also, I got an idea of how this works as I was marked as reviewer by my teammates to get to know what the procedure in the company is.

After I completed the tool, I was asked to write a python script which would result in different run configurations. I started by writing a simple Python script that would separate xml files.

-----code of this (screenshot)-----

The aim was to reduce the overhead of changing each configuration one by one when it has same parameters. So, I had make it such that I can reduce it all into one xml file with common parameters in one tag and when the python script is run, it gets into its own xml with a name, according to the name of the configuration.

As I started by writing a simple script in python to separate xml into many, I started using common parameters into one place and when required, we can easily make it look better. As I am completing my internship, I would like make it look better such that there are lesser changes that can be made to make one change in all configurations.

By completing all these tasks in my internship I got a better understanding of the norms if the company and of the software that is used by my team. I learnt a lot and looking forward to learn more during rest of my internship. Completing all tasks in the project helped me to learn more about the project in general and about the working of the software. Technically, I got a more firm grip over the project now and understand the project. I have learnt a lot about the team and company and attended many meetings to get to know about the product. I have got a great depth into the source code while working for the tool.

Challenges Faced:

As I was working for the company, I faced a couple of challenges in the tool and in the company that I would like explain in detail below:

* The first challenge that I faced was getting permissions for working for different tools and different soft wares used by the team as well the company. This took me some time as I had to contact different people in the organization. I had to take many entitlements to be able to start working in the firm.
* Next challenge came while I was working on the tool. I had to run my tool to test it again and again after adding some functionality into it. So, I ran my software many times, but continuously got an error. I got stuck on it for one day and was unable to resolve for one day. Late, when I asked my teammate, I was informed that this error was due to the fact that some other software that sends data to the main software was creating a problem. That is how, I learnt that I could have talked to my teammate when I was getting the unknown error rather than wasting a day.
* Another challenge that I faced was when there were two kinds of checks, one was specific to region and environment and others were common. So, as I was implementing my tool, I did not take this into consideration and stuck after some time because I continuously worked without thinking about this issue. Later, as I was working, I took help of my teammate and resolved the issue by putting the checks into two categories.
* Next challenge that I faced was when I had to put RED when there were one RED in the section. This was little challenge for me because I tried a lot while implementing the code. I had to think hard to make it come up in a better way and get satisfactory results out of it. So, after thinking and trying hard, I was able to complete this by just writing one line of code.
* While I was implementing my tool, I got this bug that I got results for first two regions and environments but could not get results for the rest of them because of some issue. I try resolving that issue by starting to debug every part of the code. Debugging helped me a lot to understand the code and better visualize what is happening in the tool. I could get deeper into the code and was able to get results because of it.

So, facing these challenges helped me lead a better path as I was able to learn more as I faced problems. I could think hard and solve the problem. It brought best of me as I was able to get into the code deeply. It gave me positive energy as I was able to complete one challenge. I was ready to face the next challenge and had more confidence that I would be able to clear it.

There were different lessons that I learnt while I was going through the issue. I could learn more through my mistakes and could get more insight into the software that I was working because of the challenge. I got to embrace change, as I was asked to change the software such that the structure would look differently. This helped me learn more and more and liked me to make change such that my tool looks better as I improve it.

Literature reviewed:

I learnt a lot while I was working for this firm. Starting from the norms of the company, learning about the policies, values and culture of the company, I learnt about the various tools and soft wares that are used by my team. I learnt about the main software and got deep into the code of the same. Apart from this, I learnt about the tool that I worked and code of the same. I could get to learn many tools that are used by the company for example GIT and JIRA. For instance, I learnt how to make a request to get to review my code by my teammates and my manager. Some of the technologies that I have learnt are given in detail below:

Literatures Reviewed:

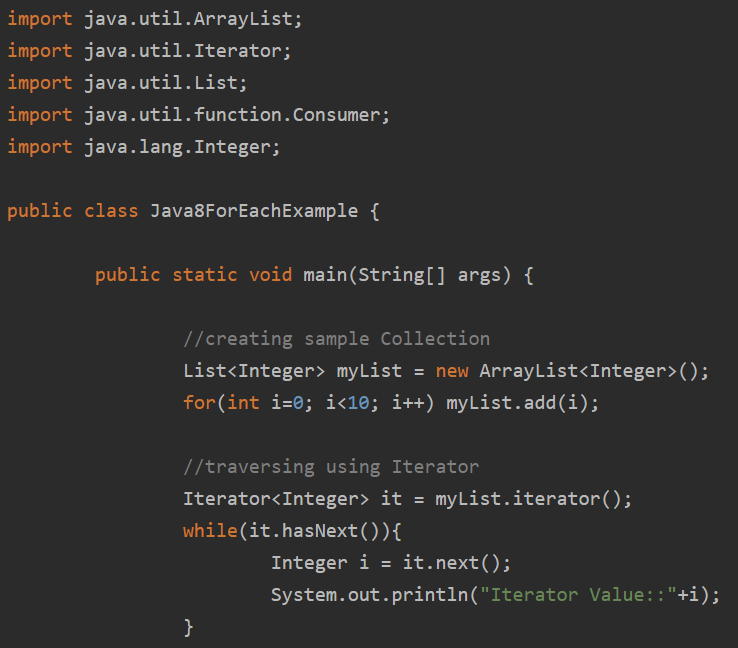
**C# GUI:** As I was going through my main software, I had also get into the Graphical User Interface of the main software. For that I needed to know C#, so I started learning about it. I learnt that it has features of any powerful, modern language. In C#, the most rapid and convenient way to create your user interface is to do so visually.

So, I started learning about it and got a little into this language which helped in a lot in going through the code. This assisted me in understanding the code of main software better as I did not know C# before this. I learnt a lot while working over here.

<http://csharp.net-informations.com/gui/gui_tutorial.htm>

**Java server:** I was asked to go deep into the understanding of the main software which was written in Java. Although, I had a great understanding of this programming language but to get more into it and understand the code, I had to study some features of Java 8 and some design patterns.

I went through the features of the Java 8 like Iterable that uses foreach() functionality using the library java.lang.Iterable. As shown in figure:



Next feature that I learnt was the @FunctionalInterface annotation.  An interface with exactly one abstract method becomes Functional Interface. We don’t need to use @FunctionalInterface annotation to mark an interface as Functional Interface.

So, other than these, there are many other features that I learnt about Java 8. That built my understanding of the programming language more deeply. I could not get into the advanv=ced version of Java.

[**https://www.journaldev.com/2389/java-8-features-with-examples**](https://www.journaldev.com/2389/java-8-features-with-examples)

There were some design patterns that I learnt about. There was this Singleton pattern that helped me implement the things in a better way. This pattern says ensures that a class has only one instance and provide a global point of access to it. <https://sourcemaking.com/design_patterns>

Similarly I studied other patterns.

RDBMS:

There was the database that we used. That database was Relational Database that is a set of formal tables from which data can be fetched. It has many ways without recognizing the database tables. The user and application programming interface of a relational database is the Structured Query Language (SQL). SQL statements are used for interactive queries and for gathering data for further collection.

<https://searchdatamanagement.techtarget.com/definition/relational-database>

This helped in understanding the databases in a better way as it is always important to brush up your skills before starting a new thing.

**SOAP Interface:**

In our main software, which is a Java Server connected to GUI of the main software which uses SOAP interface to connect with each other. This motivated to learn little bit about the Simple Object Access Protocol which is a protocol to send and receive messages especially for exchange of information structurally while implementing web services in computer networking. The features of this protocol are extensibility, neutrality and independence. XML Information Set is used essentially for the message format and is used on application layer protocols, most often Hypertext Transfer Protocol (HTTP) or Simple Mail Transfer Protocol (SMTP), for message transmission.

<https://en.wikipedia.org/wiki/SOAP>

This helped me get to understand the things in a better way.

**GIT:**

As mentioned earlier, GIT commands are used to maintain the project and code in our company. I was required to learn the commands of GIT to better understand the maintenance of the code.

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.

<https://git-scm.com/>

Some commands that I learnt were:

* git add: Adds files in the to the staging area for Git.
* git commit: Record the changes made to the files to a local repository.
* git branch: To determine what branch the local repository is on, add a new branch, or delete a branch.
* git merge: Integrate branches together.
* git clone: To create a local working copy of an existing remote repository, use git clone to copy and download the repository to a computer.

<http://guides.beanstalkapp.com/version-control/common-git-commands.html>

**JIRA:**

The tasks to the team are assigned with the help of this tool. To get better understanding of this software, I learnt about this software. It is a proprietary issue tracking product. It provides bug tracking, issue tracking, and project management functions.

<https://en.wikipedia.org/wiki/Jira_(software)>

Under Issues, there are other useful features like Issue Types that are all types of items that can be created and tracked, Workflow’s are a set of statuses and transitions that an issue goes through during its lifecycle, Screens, Fields and Issue Attributes.

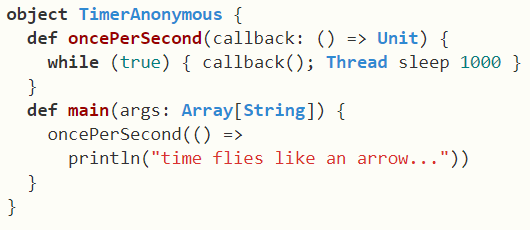
Scala:

For the monitoring tool that I was asked to implement, was implemented in Scala programming language, which I never used in my entire career. So, this was a challenge for me. So, I was required to learn Scala language and some basics of the same so that it is not a big deal for me while implementing the software. So, to start with I started learning about basics of Scala. Scala is both functional and object-oriented programming language, which serves as a good programming language. Since I knew Java which is an object-oriented programming language. I started learning from a tutorial which served this purpose.

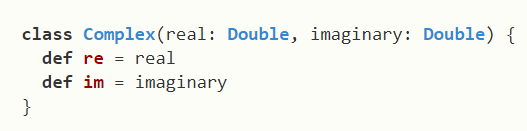
I started by writing a simple Hello World program like in the figure:



After that learning some features of the Scala. Like learning about the fact that functions are the object here. The interesting part of it being a functional language was its ability to manipulate functions as values. For example in the following figure:



Other interesting fact was that methods are without arguments. Methods without arguments differ from methods with zero arguments in that they don’t have parenthesis after their name, neither in their definition nor in their use. Example is in the following figure.



<https://docs.scala-lang.org/tutorials/scala-for-java-programmers.html>

SCRUM:

There are SCRUM meetings that happen every day with the teammates and my manager. In Scrum, on each day of a sprint, the team holds a daily scrum meeting called the "daily scrum.” Meetings are typically held in the same location and at the same time each day. Ideally, a daily scrum meeting is held in the morning, as it helps set the context for the coming day's work. These scrum meetings are strictly time-boxed to 15 minutes. This keeps the discussion brisk but relevant.

<https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum>

IntelliJ:

There is an IDE that is used to implement the main part of my software which is written in Java language. So, for that I needed to have some knowledge about this tool. So, I learnt about this tool and got a better understanding. IntelliJ IDEA is a Java integrated development environment (IDE) for developing computer software.

<https://en.wikipedia.org/wiki/IntelliJ_IDEA>

Visual Studio:

There was another IDE that I used for the GUI of my software is implemented on this tool. So, for that I gained some knowledge of the tool and its features.

<https://en.wikipedia.org/wiki/Microsoft_Visual_Studio>

SDLC:

The project that I worked on followed a framework called SDLC which allows us to define and perform tasks at every point of time that is each step in the software development cycle. It is followed by my development team within the software organization. The life cycle defines a methodology for improving the quality of software and the overall development process. It consisted of following steps:

Planning:

Implementation:

Testing:

Documentation:

Deployment and maintenance:

Maintaining:

<https://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc>

Agile:

JAXB:

Internal Wiki: