J.P. Morgan & Co.

Foreign Future Variation Margin Reconciliation

Automating a Manual Reconciliation Process

Kevin Kell, Jiahui Li, Daniel Matos
12/16/2014

Contents

Introduction

Executive Summary

Technology is an important aspect in any industry nowadays. It has helped improve the living and working qualities of many individuals in society by enhancing the development of products and performance of duties in any environment. Computers for instance are an indispensable tool in our daily lives. They have helped us by organizing large amounts of information, automating daily processes and reducing time consumption of necessary tasks. Like most aspects of daily life, the financial has been greatly affected by changes in technology. Tasks that used to be done manually now can be completed automatically by the use of different technologies.

JP Morgan Chase & Co. is one of the many financial institutions that are fully invested in the use of technology in order to improve their business. Often, technological improvements can save the firm large amounts of money. In addition to saving money, technology can help a firm comply with new regulations.

One task that has yet to be automated within JP Morgan is the reconciliation process of foreign futures. The accurate comparison of foreign future information between the firm and its counter parties ensures the integrity of trades made by JP Morgan. However, the current process requires several manual steps that increase the chance of human error and it requires a considerable amount of time. JP Morgan has decided that is time to enhance this process by automating several of the manual steps.

Our team analyzed the current process and determined which steps of the process could be automated. We took the manual steps and translated them into features that could be added to an existing application. The next challenge was to develop those features and test them with the users. After the implementation the new process requires a fraction of the time and presents lower risk for error. In addition to this, the application displays the data in more simplified manner allowing an easier analysis.

Abstract

One task of the operations team at J.P. Morgan is to reconcile the balances of their foreign future holdings on a daily basis. This involves comparing data on excel sheets generated by J.P. Morgan and by the brokers from whom J.P. Morgan has purchased its foreign futures. This task is called the "reconciliation process". For many years the reconciliation process has been completed manually. However, the process is inconvenient for the operations team because it requires a large amount of time to be completed, uses several excel files, and is prone to human error.

The MQP team was assigned the task of enhancing the current process. Through analysis of the process, the team came up with several features that could be built into an existing internal application (Variation Margin Application) in order to reduce the time the Operations team needs to complete the reconciliation process.

Several features were implemented into the Variation margin application. These include a "View Unified Report" button, which allows the user to upload excel files into the database. After uploading the files, they will be able to view a new unified report that allows them to quickly see any problems. The resulting product reduced the time needed to complete the reconciliation process from about two hours to five minutes, ensuring a reliable and simple method of reconciling J.P. Morgan's foreign future balances.

Acknowledgement

Purpose

The current process of reconciling foreign future accounts requires several manual steps. The purpose of this project is to outline the requirements involved with reconciling foreign future accounts and specify the technical measures that can be taken to minimize the manual steps necessary to complete this process. Ideally this will reduce the time needed to complete each step of this process and significantly reduce any possibilities of human error. The applications involved, sources of data necessary, and the resulting workflow will be described in detail.

Background

Company

This project was completed in a seven week period while the team worked at J.P. Morgan Chase & Co. J.P. Morgan is among the most respected banks in the world. It is the largest in terms of total assets, and

prides itself on its reputation as a reliable financial institution. It operates in more than 60 countries and has approximately 260,000 employees (J.P. Morgan Chase & Co., 2014).

History

J.P. Morgan Chase & Co. had its origin in the year 1799 when the earliest predecessor firm, The Manhattan Company, was formed. The first firm to bear the name J.P. Morgan was The New York firm, which changed its name in 1895 when J. Pierpont Morgan took control (J.P. Morgan Chase & Co., 2014). Over the years, many mergers have grown and shaped the firm. Some of the most significant mergers include the merger of Chase National Bank and The Bank of the Manhattan Company in 1955 and the merger of J.P. Morgan with The Chase Manhattan Corporation in 2000 (J.P. Morgan Chase & Co., 2014).

Structure

J.P. Morgan Chase & Co. (JPM) has businesses in many lines of business that are quite varied. The one that is relevant to this project is Asset Management. Led by Mary Erdoes, Asset Management is responsible for approximately 1.7 trillion dollars coming from a combination of corporations and high net worth individuals (J.P. Morgan Chase & Co., 2014). There is a technology branch of Asset Management which develops software to support the business side of Asset Management. The users of the final product of this project are on the business side of Asset Management.

Specifically, the users are in the operations division. The operations division is what is commonly known as the Middle Office and is responsible for settling trades, monitoring and tracking financial information, and making sure regulations are followed. Most of the trade settling is done electronically these days and is almost entirely automatic. For some more complex assets, such as foreign futures, the process still involves a human copy and pasting data into excel and manually searching for discrepancies. Reconciling foreign futures variation margins is one of the responsibilities of the operations team, and enhancing that reconciliation process is the focus of this project.

Foreign Futures

Foreign futures are a complex asset type. They are futures that have been purchased using a foreign currency, so the value of the foreign future in US dollars depends on two things. The first is the value of the future, and the second is the exchange rate between the base currency (the currency used to purchase the foreign future) and US dollars.

Futures

To fully understand foreign futures, one must first understand futures. A simplified description of a future contract is an agreement between a buyer and a seller to exchange a specified quantity of a specified asset at a specified time for a specified price (Heakal). Initially, futures were usually purchased for commodities such as wheat. The future would protect both parties from radical changes in wheat price. Today, the asset can also be something other than a commodity, such as a stock price. Futures such as these are known as financial futures.

Modern future contracts do not require the asset to actually be exchanged. Instead, the parties will exchange money in a way so that the agreed quantity of the asset could be bought or sold at the agreed price, despite any changes in the price since the agreement. For example, let us say that a future

contract is made between a buyer and seller to exchange 100 bales of hay for $5 each in one year. After a year when the contract is up, the price of hay has gone up to $7 per bale. If the buyer wanted to buy 100 bales of hay at that price he would have to pay $700. Fortunately for the buyer, he made the future contract to buy 100 bales at only $5 each. The total cost at that price would be $500. So, instead of the seller actually giving the buyer 100 bales of hay, he simply gives him $200, or the difference between the agreed price and the current market price. Now, if the buyer chooses, he can get 100 bales of hay by only paying $500 of his own money and $200 of the counterparty. Keep in mind, that this way of managing futures does not require the buyer to actually buy hay. If he chooses, he can just keep the $200. In the same way, if the price of hay had gone down, the buyer would owe the seller money to make up the difference, but the seller would not actually need to sell hay.

Variation Margin

Futures contracts as described above have a big risk to them. There is a risk that at the end of the contract, one of the parties will not be able to pay the amount that they owe. To continue the above example, it is possible that in between the signing of the contract and the end of the contract, the seller went bankrupt. At the end of the contract, they will owe the buyer $200 but will not be able to pay. In this situation, the buyer would lose $200 that was rightfully his. To combat this, futures contracts are settled on a daily basis. This means that each day, depending on the price change of the asset, the buyer and seller will exchange the difference. For example, if the price of hay goes up during the first day of the contract, the seller will pay the difference as if it was the end of the contract. If the next day the price goes back down, the buyer will pay the seller the difference. This way, if one party goes bankrupt, the other party only stands to lose the money that would have been owed to them that day.
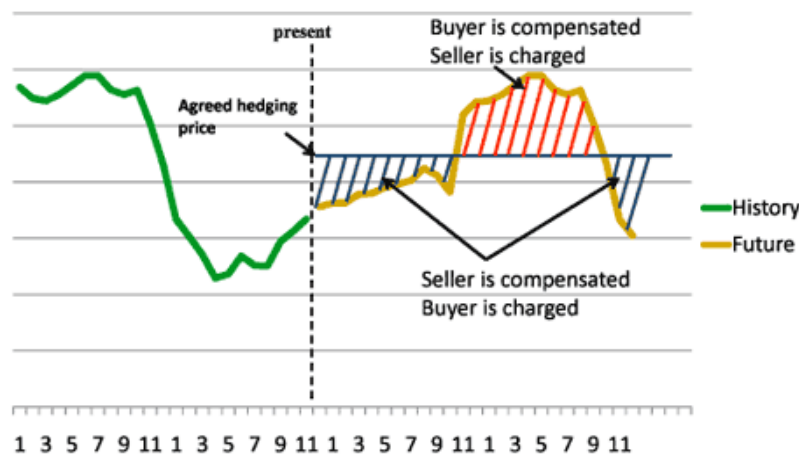
Figure 1: Illustration of the flow of money in a futures contract

The difference that is paid daily as described above is what is known as a variation margin. In the context of the Asset Management Operations Team, this amount must be agreed upon and double checked to ensure that we are not paying too much or too little and that no mistakes are made. Each day a manual reconciliation process must be completed to ensure the integrity of the future.

Foreign Futures

As mentioned above, a foreign future is a future that has been purchased using a foreign currency. An example would be a future contract for wool purchased in British pounds (GBP). The account will be in GBP, and the variation margin payments will be in GBP. The value of the account to JPM does not rely solely on the price of wool. Since any earnings or losses have to be translated into US dollars (USD) the value of the account relies on the conversion rate between USD and GBP in addition to the price of the future. In the event that one of JPM's foreign futures is doing well and is earning money day after day, they do not want to immediately withdraw the money. It is more cost effective to wait until the balance builds up and transfer it from GBP to USD all at the same time. One of the things that the operations team needs to do on a daily basis is to watch the balance and wait for the best time to convert back to USD.

Variation Margin Application

The operations team currently uses an application called Variation Margin to assist them with the foreign futures reconciliation process. The Variation Margin application has several functions. It assists the operations team with reconciling issues with both futures and cleared OTC and provides features that allow users to view data, upload data, and correct errors.

As can be seen in the screenshot below, the application displays data in a spreadsheet like way, and the user can navigate to other views by using the dropdown menus. The screenshot has had account numbers blacked out to protect sensitive information.

Figure 2: Screenshot of the Foreign Futures Dashboard, part of the Variation Margin application

Within the Variation Margin application, is the Foreign Futures Dashboard. This dashboard displays all the data relevant to foreign futures. The screenshot above is showing the Foreign Futures Dashboard. When the reconciliation process is begun, the operations team will download this data as a CSV file and use the data to run the process using several excel documents.

Methodology

Introduction

Four phases of development were used to produce the final product. The first phase was to analyze the current process workflow to identify the problems of the current system and the areas for improvement. Second was a design phase that determined what the final product should look like based on the requirements and expectations of the users. Then the features were developed and finally it was tested. The following sections describe our methods in completing these phases.

Analysis

Data Sources

The operations team gets data from three data sources, which are labeled emailed Eagle File, Variation Excel File and Broker File in the diagram below. The flow chart describes the flow of data to the operations team.

```
Eagle Database  ──(1) Emailed nightly by batch process──▶  Emailed Eagle Excel File

Eagle Database  (1) Updated nightly by batch process  ─▶  Variation Margin Application

Variation Margin Application  (3) Manually Downloaded as CSV ─▶ Variation Margin Excel File ─▶ Operations

Broker ──(1) Emailed nightly from broker──▶ Broker Excel File

Broker Excel File (2) Manually uploaded in Mumbai ─▶ Variation Margin Application

Emailed Eagle Excel File ─▶ Operations

Broker Excel File ─▶ Operations
```

Figure 3: Flow of data to the operations team. The result is the three excel files containing data highlighted in red.

The process begins at the eagle database, which stores all the accounts, cash, and trades information. At night, there are two batch processes that run to deliver the data to the operations team. The first process emails the eagle file from the eagle database and the second process updates data onto the Variation Margin application. At the same time, the broker emails the broker file, which is manually uploaded onto the Variation Margin application. Then the operations team will manually download data from the variation margin application as a CSV file, which is called Variation Margin Excel File. After this

process, the operations team has received three data sources, which are labeled in the above diagram as Emailed Eagle Excel File, Variation Margin Excel File, and Broker Excel File. Then the operations team will copy and paste the data from these three excel files onto excel sheets to start the reconciliation analysis.

Current Reconciliation Process

The reconciliation process is comprised of four steps. The operations team calls them Recon, No Activity, ForCash and Account Tie Out. Each step compares two items to determine if there is a situation that needs attention from the operations team. Recon and ForCash compare balances from different sources to ensure that there are no disagreements. No Activity compares today's balance with yesterdays to see if the account is active. Account Tie Out compares the accounts that are included in one data source with the accounts included in another to ensure no accounts are left out. The following sections will explain these four steps in detail and the process that the operations team currently goes through to complete each step.

Recon	No Activity	ForCash	Account Tie Out
• Application Balance • Broker Balance	• Previous Day Balance • Current Day Balance	• Application Balance • Eagle Balance	• Accounts in Broker Report • Accounts in Eagle Report
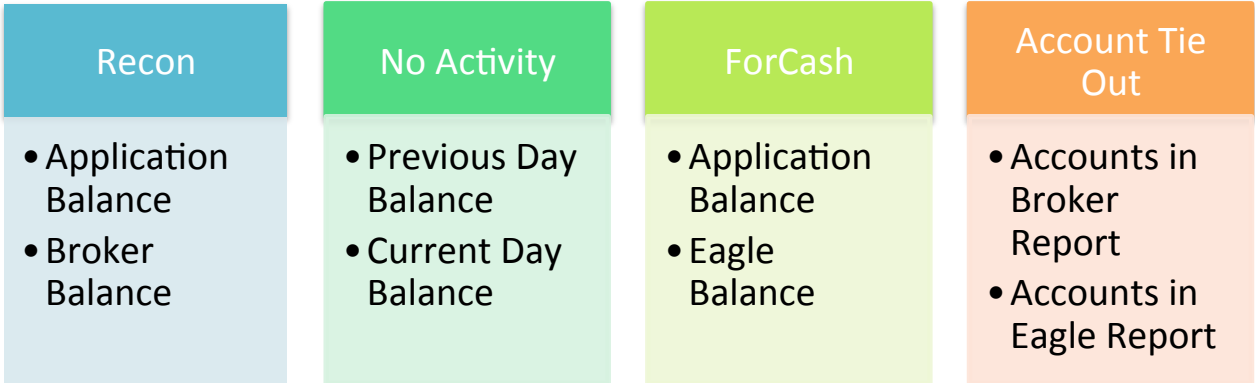
Figure 4: Summary of reconciliation process steps and what each step compares.

Recon

The first step is Recon, which compares the application balance with broker balance. If there is any difference, the analyst will raise and reconcile the issue with the counterparty or broker. The following flow chart details the steps:
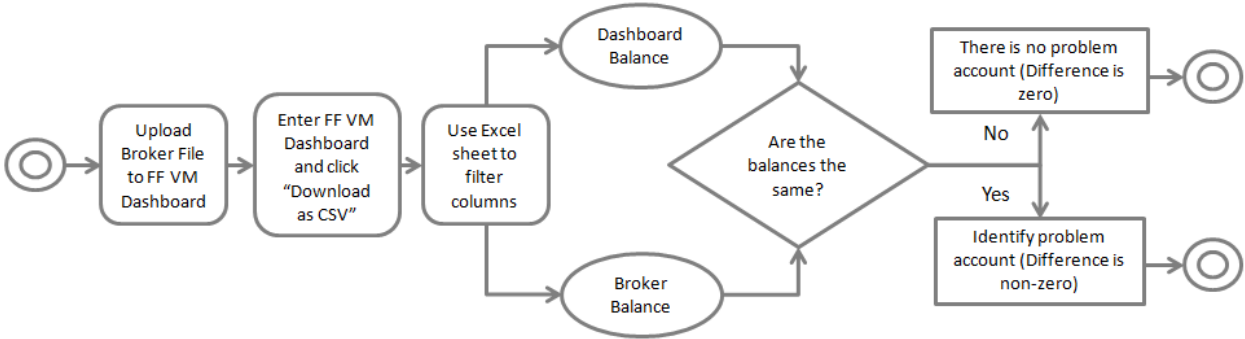
Figure 5: Summary of the "Recon" step of the reconciliation process

1. User accesses Foreign Futures Dashboard and
2. User downloads all data from Foreign Futures Dashboard data as a CSV file
3. User copies CSV file into the **Error! Reference source not found.**
4. User isolates the columns in spreadsheet
5. User compares columns to identify discrepancies between the broker balance and the Foreign Futures Dashboard balance
6. User refers to the "Difference" column on sheet for indication of any discrepancies

No Activity

The second step is No Activity, which compares the previous day balance with the current day balance. If the difference is zero for multiple days, it means the account is possibly inactive. The account balance will be withdrawn if an account is found to be inactive. The following flowchart details the steps:
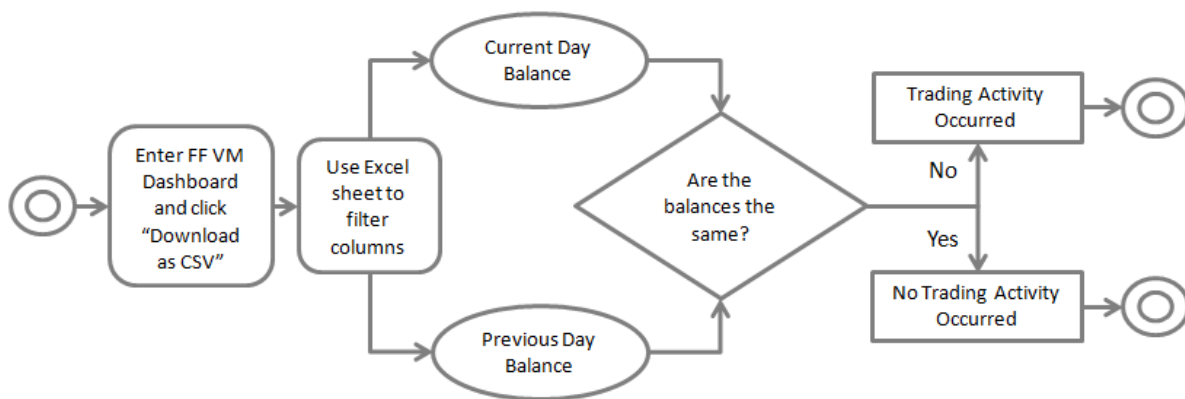
Figure 6: Summary of the "No Activity" step of the reconciliation process

1. User downloads Foreign Futures Dashboard data as a CSV file
2. User copies data into the **Error! Reference source not found.**
3. User isolates the spreadsheet columns
4. Foreign Rec File checks to see if the previous day's balance is the same as the current day's balance
5. Foreign Rec File fills in the column "No Activity" with the result
6. User determines which accounts have no trading activity by examining "No Activity" column

ForCash

The third step is ForCash, which compares the application balance with the eagle balance. This difference could be caused by the commission fee that is stored by one side but not the other side. The following flowchart details the steps:
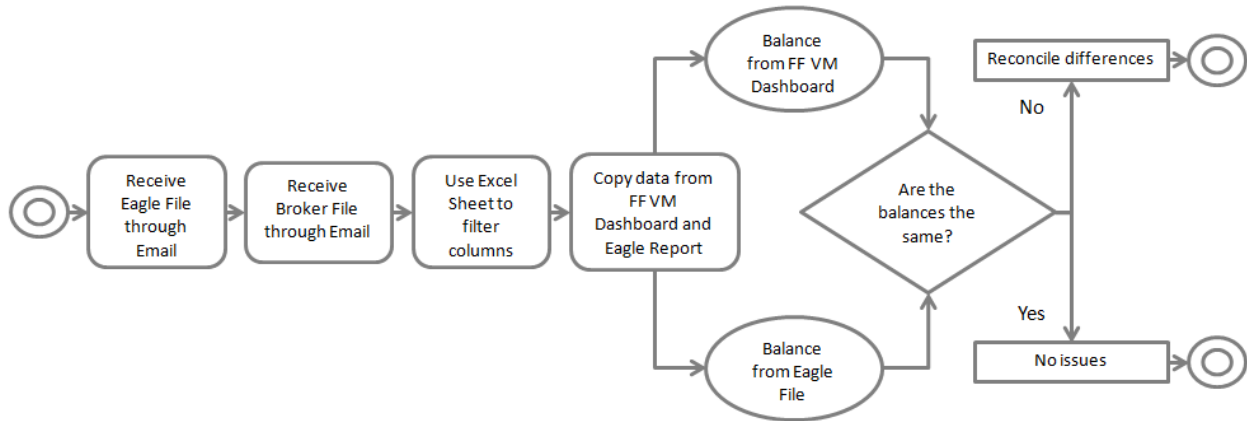
Figure 7: Summary of the "ForCash" step of the reconciliation process

1. User gets all of the data from the Foreign Futures Dashboard
2. User gets all data from email received from Eagle
3. User gets all data received from Broker
4. User copies all data from the three sources into the **Error! Reference source not found.** .
5. The User uses the newly created ForCash Spreadsheet file to compares the balances from each data from all sources for discrepancies
6. User looks primarily for balance differences between the Foreign Futures Dashboard data and the emailed report from Eagle

Account Tie Out

The last step is Account Tie Out, which compares the number of accounts in broker report and the number of accounts in eagle report. If there is any missing account, the analyst will figure out the missing account and reconcile the issue with the counterparty or broker. The following flowchart details the steps:
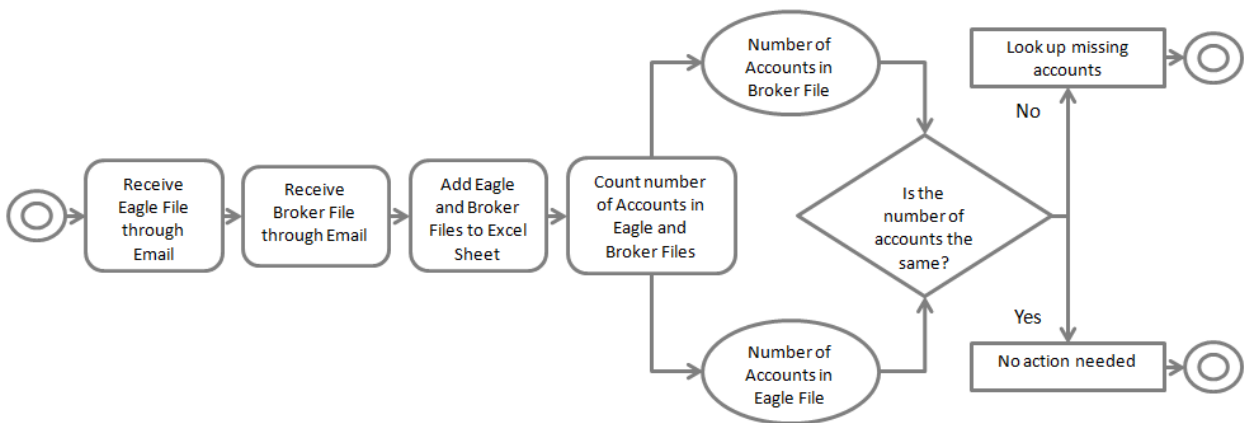
Figure 8: Summary of the "Account Tie Out" step of the reconciliation process

1. User receives Eagle report via email
2. User receives Broker report via email
3. User combines data from both sources into the **Error! Reference source not found.**

4. Account Tie Out File counts the number of accounts contained in the broker files and the number of accounts in the Eagle report and summarizes the result
5. User looks to identify any missing accounts in either the broker files or the Eagle report

Analysis Results

Since the procedures involved with of each step of the processes described are somewhat are similar, we summarized them into the following workflow. For each step of the reconciliation process, the following manual steps must be completed:
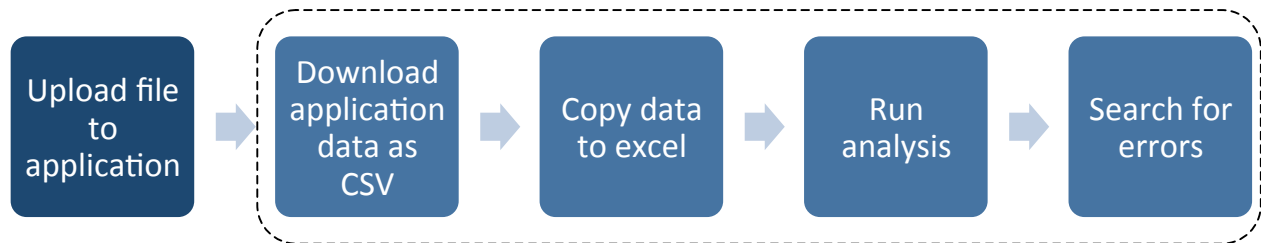
Figure 9: Manual steps involved with each step of the reconciliation process. The steps in the dotted square will be eliminated by the new features that will be added to the Variation Margin Application.

After the data was uploaded onto the Variation Margin Application, the operations team will download the application data as a CSV file. Then the operation analyst will copy and paste the data onto excel sheets, conduct reconciliation analysis and identify the error accounts. However, there are some areas need to be improved. Each step requires many manual procedures that demand a large amount of attention and time, and it involves data in several locations. It is also hard to manually search for the error accounts and examine the results. In order to solve these issues, the team determined that it was possible to eliminate the manual steps, which are enclosed in the box, by adding features into the current application.

Design

Once the areas of improvement for the current application were determined, the new features were designed by outlining the requirements and brainstorming ideas to fulfill the users' requirements and expectations.

Future Process Options

The following flowchart is the new feature options that were considered:
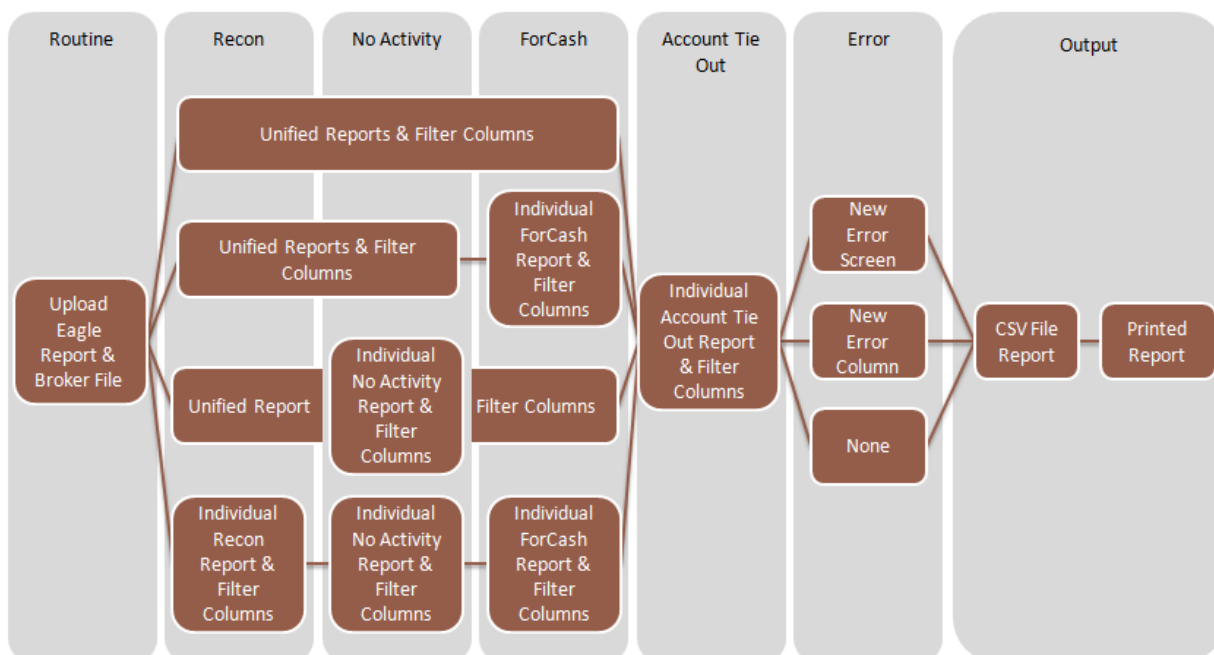
Figure 10: Feature option illustration. The different branches in the diagram represent potential workflows based on the options that are chosen.

Routine

Uploading broker file and eagle file are the daily steps that are required for the operations team to get access to data sources. The files are parsed by the application and their data is extracted and stored in the database for use when the reconciliation process is run.

Recon, No Activity, & ForCash

The Recon, No Activity and ForCash processes are currently being computed, stored, and displayed in separate reports (e.g. Excel sheets). Our goal is to provide options to simplify the process of generating and viewing these reports. To simplify the use of the reports that are generated, the results can be combined so that they can be viewed all at once. These options would include **(1)** a combination of all reports, **(2)** a unified Recon and No Activity report and individual ForCash report, **(3)** a unified Recon and ForCash report and Individual No Activity report, and **(4)** Individual reports for each of the processes (as it currently is). These four options are explained below in detail:

(1) **Unify All Reports**
Instead of continually downloading the entire data set from the Foreign Futures Dashboard, a user would be able to download a completed reconciliation report containing the results of the Recon, No Activity and ForCash processes in a single CSV file.

(2) **Unify Recon and No Activity Report – Individual ForCash Report**
This option would display the results of the Recon and No Activity processes as a unified report. The ForCash data set would be displayed in a separate individual report.

(3) Unify Recon and For Cash Report – Individual No Activity Report
This option would display the results of the Recon and ForCash processes as a unified report.
The No Activity data set would be displayed in a separate individual report.

(4) Individual Recon, No Activity, and ForCash Reports
This option would separate the data set for each of the current processes (Recon, No Activity and ForCash) into individual reports. The users are currently using this method.

The extent to which the Recon, No Activity, and ForCash reports can be unified relies upon whether or not the data will make sense when put together.

Column Filtering

In order to only get the records that are relevant to a given analysis, a feature would be added to the Foreign Futures Dashboard to allow the user to filter out unnecessary columns for the analysis of the each process. This feature would include a download button that only downloads the selected columns as a CSV file. This option could be used to generate any combination of columns, in order to provide more flexible and simpler reports in the future.

Account Tie Out

The Account Tie Out test should be displayed separately since it requires the application to find and list accounts that appear on one data source but not the others.

Error Display

New Error Screen

A new screen could be added to the Variation Margin Application that would collect discrepancies among the data sources. A user could easily see what problems are occurring and which accounts are causing them. If necessary, the screen could contain a report of the accounts in questions and enable the user to export the report as a CSV file

New Error Column

Alternatively, a new column could be added to the current Variation Margin Application screen that would contain a message indicating the records causing errors. The records that cause errors would be highlighted in a different color. This will allow the problem causing records to be quickly identified without the complexity of creating a new screen in the Variation Margin application.

Output

Export to CSV

A CSV file report can be obtain directly from the Variation Margin application. This file is downloaded by the user, copied and pasted into an excel sheet and then formatted into the reports that are to be analyzed.

Print from Dashboard

To avoid the process of downloading of CSV files, copying and pasting data, and analyzing the data in an excel sheet, the Variation Margin application could have the option to print a reconciliation report directly from the web page.

Fix Eagle Discrepancies

One of the major checks that are made in the current process is to ensure that the data from the emailed Eagle report and the Foreign Futures Dashboard are the same. Ideally there should not be any difference since both of these sources get their data from the same place (the Eagle database).

If this issue could be isolated and resolved then the ForCash step of the current process could be removed and only the Recon, No Activity, and Account Tie Out will be needed. In this scenario, only the broker files would have to be uploaded to the Variation Margin application and all error checking could be done there. It is unclear whether this is an issue that can be resolved, and if it can be resolved it is unknown what is necessary.

Selected Future Process

Several features will be added to the Variation Margin Application to simplify and further automate this process.
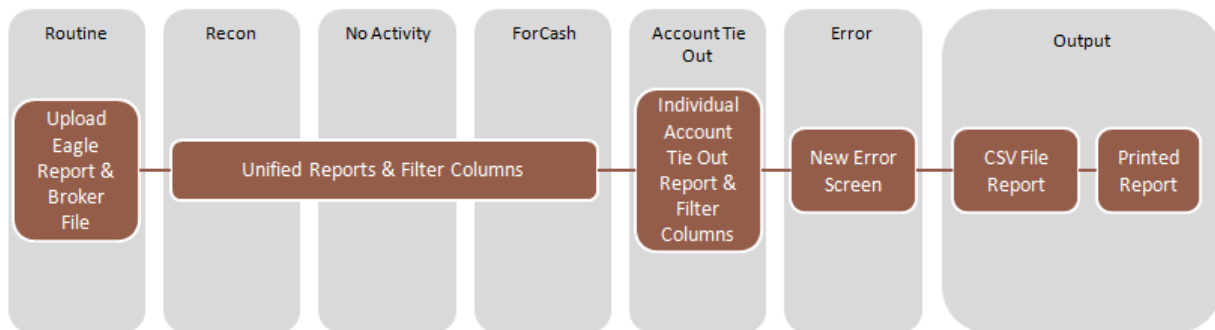
Figure 11: Chosen features. This is the new workflow based on the features that were chosen to add. This diagram is derived from figure 10 and has had the workflows that were not chosen eliminated.

A feature allowing for Eagle Files to be uploaded and stored will be added. This way all the data used in the reconciliation process (the broker file and the eagle file) will be available to the Variation Margin application. The eagle file will be uploaded in a very similar way to how the broker files are currently uploaded. The user will select the date and the file and click upload. The file will be parsed and its data will be stored in the database for use later.

Since all the necessary data will be in the application, the tests can be run in the application without using excel. After careful thought, it has been determined that it makes the most sense to display the ForCash, Recon, and No Activity reports together. Since they only require that a difference be found between various balances, the data that needs to be displayed is similar for each of them and it makes sense to put them together. A mockup of how the data from Recon, No Activity, and ForCash can be combined into one view can be seen below:

COMMON COLUMNS				RECON			NO ACTIVITY			ForCash		
JPM Account	Broker Account	Broker	Currency	Current JPM Balance	Broker Balance	Difference	Current JPM Balance	Previous Balance	Difference	Current JPM Balance	Eagle Balance	Difference

Figure 12: Unified report mockup

The Account Tie Out test should be displayed separately since it requires the application to find and list accounts that appear on one data source but not the others. A mockup of how the Account Tie Out data is displayed in below:

Common Columns				Account Tie Out		
JPM Account	Broker Account	Broker	Currency	Dashboard	Broker File	Eagle File
12345	12341	GSCO	AUD	Yes	Yes	No
56146	23452	GSCO	HKD	No	No	Yes
34361	61433	GSCO	SEK	No	Yes	No

Figure 13: Account Tie Out report mockup

Development

The Variation Margin application is built like most web apps. It has a front end which runs on the client's machine, a middle tier which runs on a server, and a back end database which runs on a dedicated database machine. One challenge that was faced while developing the features to be added was the size of the application. By J.P. Morgan standards, the application was not particularly large, but it still presented a challenge to understand the existing application before new features could be added. The first part of the development phase was spent sifting through the approximately 15,000 lines of JavaScript on the client side and 43,000 lines of Java on the server side.

Front End

The front end, written in JavaScript, takes advantage of the Ext JS framework to create a model view controller application. The client side actually has one more entity, called a store. The function of each entity is the following:

View: Controls what the user sees. It sends all user interaction to the controller to be processed.
Controller: Works as an intermediary between the view and the model. If the view needs data to show the user, the controller fetches it from the user. It handles all user input and mediates the flow of data to the view.
Model: Holds the data needed for the application. In Ext JS, it is a file that defines the structure of the data. A file maps name value pairs so data can be easily accessed.
Store: The store is responsible for determining when to fetch data from the server to refresh what is in the models. It keeps track of the URL from which data should be requested, and when fresh data needs to be retrieved.

Server

The middle tier, or server, is responsible for handling network requests from the client. It properly stores any data sent from the client in the database and ensures the correct data is returned when the client requests data. It is written in Java and uses both the Spring framework and Apache Hibernate.

Spring is used to handle network requests. The server uses Spring controllers to handle requests sent to specific URLs. Spring makes it easy to get parameters and other information about the request to pass on to internal methods. Spring is also used to reduce the number of dependencies within the application. Dependencies are mapped in context xml files and spring takes care of them automatically when the application is run.

Apache Hibernate is used to simplify access to the database. The Variation Margin application uses it in the following way. Objects in the project are mapped to tables in the database using an xml configuration file. The fields in the object relate to the columns in the table. A data access object is created to provide methods for retrieving and storing data to the database. When those methods are called, Hibernate uses the configuration files to automatically generate a query to send to the database. For example, if the application needs to retrieve some data from the database, the application would call a method in the data access object of the table containing the data. The method would automatically generate a query to send to the database. It will automatically create the Java object, which has been mapped to that table, and populate its fields with the data from the response. The method will then return the populated object and the application can do whatever it needs to with the data.

Database

The database for the Variation Margin application is a combination of Sybase databases and Oracle databases. This project altered only the Oracle database. Oracle databases are high-grade relational databases and are a standard for enterprise level applications. Relational databases have tables with a number of columns in each table. The tables hold the data and the data are organized into the various columns. The data can be queried using SQL, or structured query language.

To add the chosen features to the Variation Margin application, each of these entities needed to be altered. Below is a brief overview of the changes that were made.
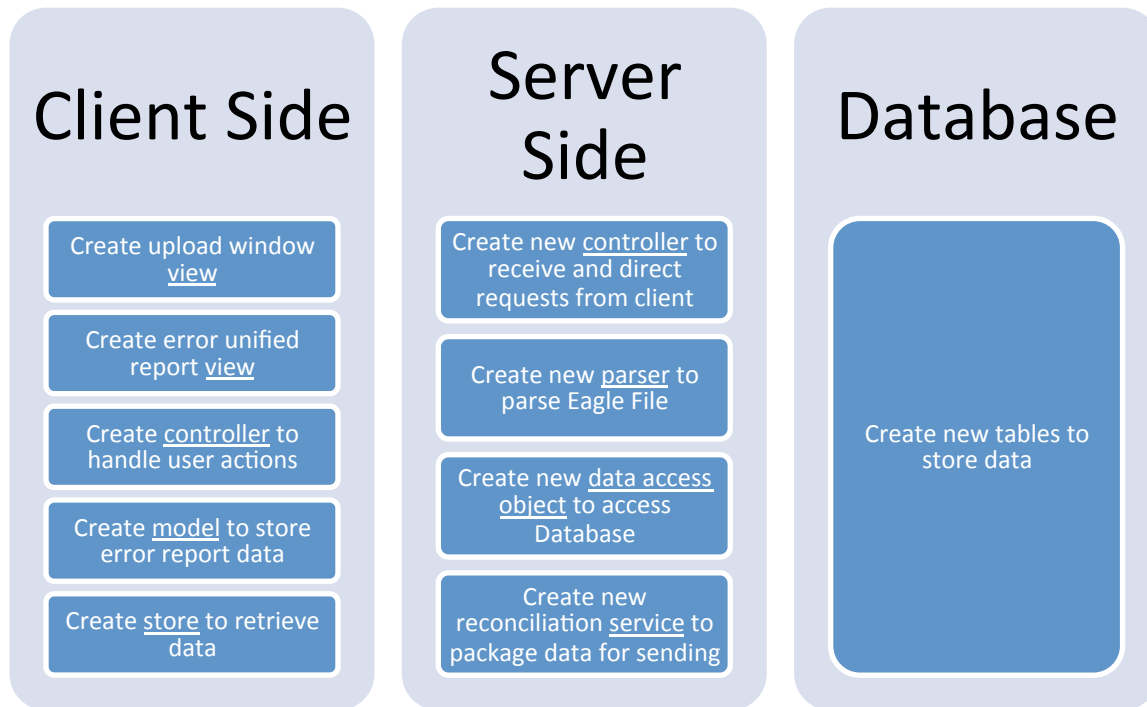
Figure 14: Application architecture overview

There were two major functions that needed to be added to the system. The first is the ability to upload an eagle file and store the data it contained in the database. The second is to retrieve all the data needed for the reconciliation tests, run the tests, and display the results to the user.

Eagle File Storage

To store the eagle file, the client needed a new view to allow the user to select a date and upload a file. The controller for that view sends the file to the server in a network request once the upload is complete. The client then waits for a success response from the server. Meanwhile, on the server side, the file is received by a Spring controller and sent to a parser built to parse eagle files. The parser reads through the excel file and populates a java object with the data it extracts. When it is done parsing, it returns the populated object. The populated object is sent to a Spring service specifically designed to store and retrieve eagle file data. The service sends the populated object to the data access object, which generates an SQL query and sends it to the database.

The database stores the data in a table and sends a success response. The server gets the success response and sends a success response to the client. Finally, the client gets the success response back and can notify the user that the file was safely stored. Below is a diagram detailing this process.
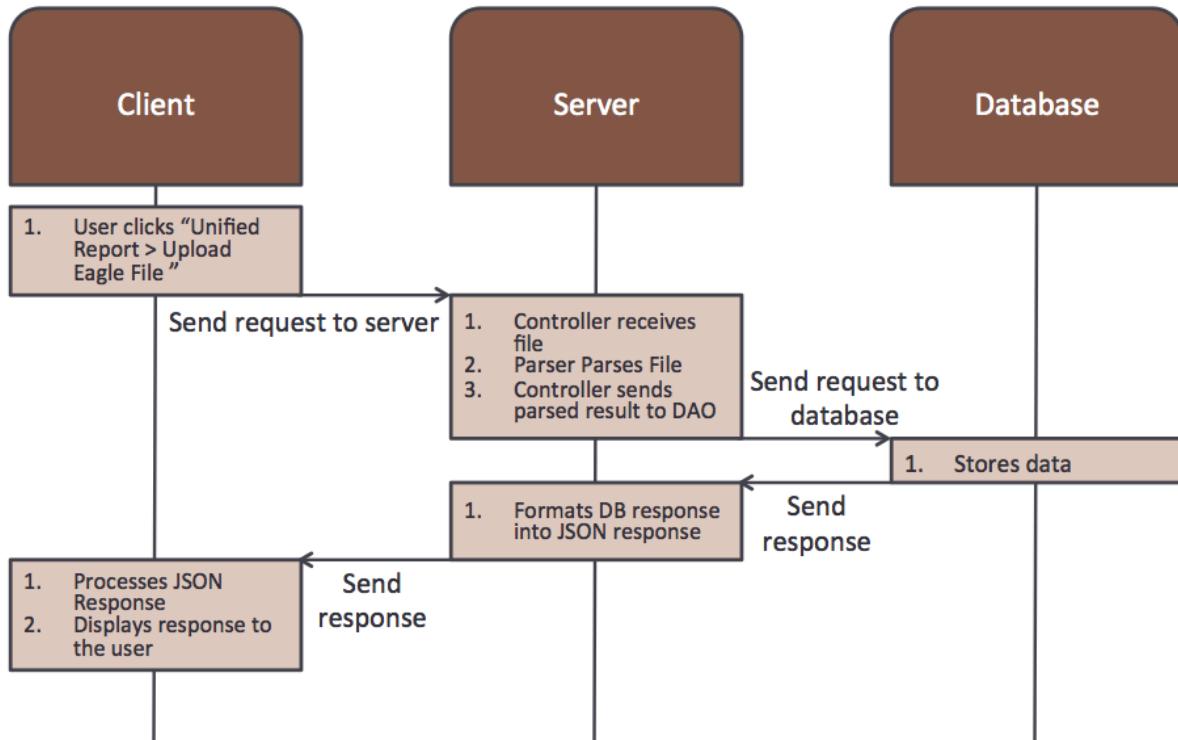
Figure 15: Request to store eagle file illustration

Report Generation

The second feature that needed to be added was the ability to aggregate all the data needed to produce the Recon, No Activity, and ForCash reports, execute those reports, and then display the results to the users. The process begins when the user requests the report by clicking on a button. The client will then attempt to show the error report view, but will then realize that the data is needed to populate the view. Accordingly, the store will make a request to the server asking for the required data. On the server side, the request is received. The data necessary for the report is located in three different tables in the database, so the server will utilize three different data access objects to make three different requests to the database for data.

The database simply executes those requests and sends the data back. Once the data arrives back at the server, it is sent to a reconciliation service which takes the data from the three different tables and combines them with the other pieces of data that match. This is done by matching on the JPM account code and the base currency of the foreign future. This is enough to uniquely identify records. Once the data is matched up, it is used to populate a response object to be sent back to the client. It is at this time that the calculations are done to complete the reconciliation process. With all this done, the server turns the response object into JSON form and sends it to the client. The client receives this data and uses it to populate the error report view. Now the user can immediately see the results of the error report. Below is a diagram showing a summary of this process.
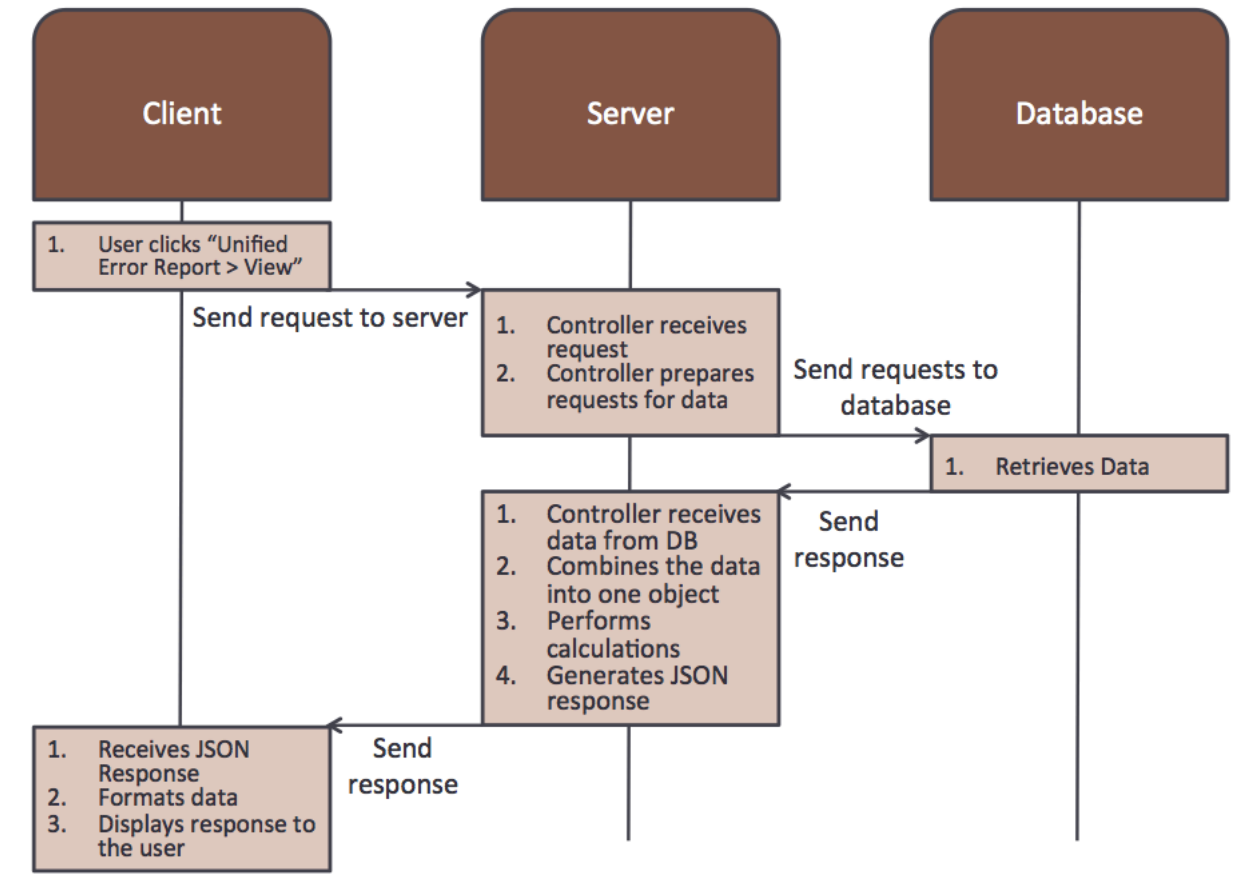
Figure 16: Request for error report illustration

Coding Style

When developing these features, much care was taken to respect the existing coding style of the project. The controller pattern was used to receive incoming network requests, the service pattern was used to accomplish tasks that could be done in multiple ways to allow for interchangeability, and data access objects were used to interface with the database. Respecting the existing style and techniques will make the project more maintainable and understandable for any future JPM employees.

Testing

JPM has a well established path that an application follows from development to production. There are three environments that it will pass through along the way. The first environment is called DEV, which is short for development. This is an environment that is safe to experiment in. Developers can alter the application and database without worry that it might destroy real data that is important to the operations of the company. All the databases are only used by developers so if something goes wrong there will be no damage.

Once the features are added in the DEV environment, it is moved to a testing environment called UAT. This is where the application is tested. Similar to the DEV environment, this environment is protected so that any errors or issues caused there will not mess up any real data. In the UAT environment, quality

engineers test the application rigorously and users are allowed to test the application. Any errors or issues are noted and fixed.

Once all issues are fixed, the application will be allowed to pass to the production environment known as PROD. PROD uses real data and is the environment used by the users on a daily basis. Ideally, by the time an application makes it to PROD, it should work perfectly and will not cause issues.

All of the development of the features for this project was done in the DEV environment. In the last week of the project, it was deemed good enough to pass into the UAT environment. There, the testing process has begun. Users are experimenting and commenting on the new features, and over the course of the next few weeks a team of quality assurance engineers will test it. After a month of testing and fixing, the features built in this project will be able to pass into the production environment.

Results

It was decided to implement two distinctive features into the Variation Margin application. These features include (1) a Unified report button and (2) a Unified report screen. These would also allow the current process to become simpler to analyze, less time consuming and more reliable.
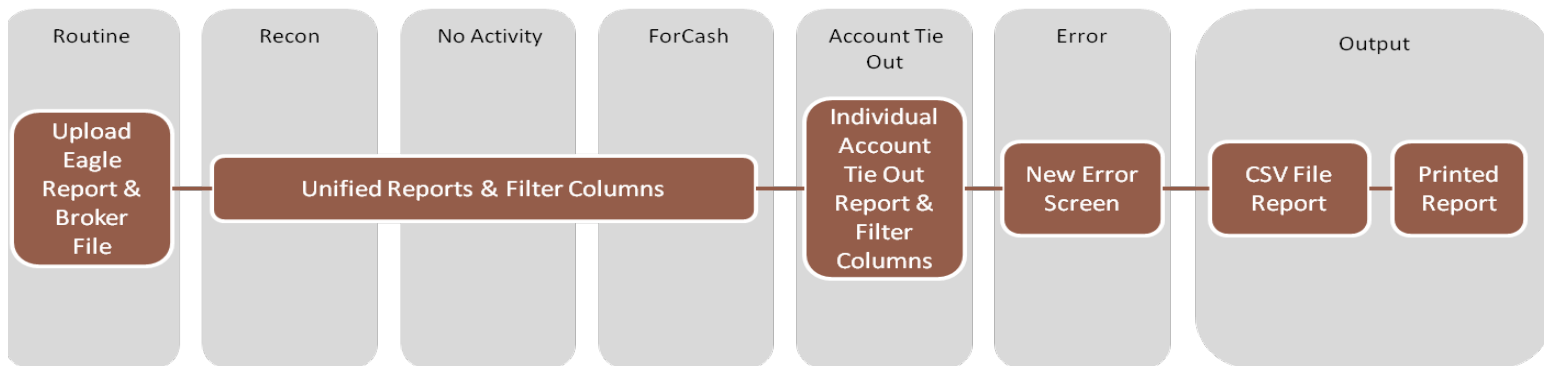

Figure 17: Chosen features. This is the same as figure 11. It has been inserted here for convenience.

The diagram shows the new process that the Operations team would follow. Notice that the Unified report feature includes the Recon, No Activity and ForCash results. The Account tie out process was left out as a separate report due to the nature of its data not being compatible with that of the other reports. This process will be displayed in a separate dedicated screen.

1. Unified report button

The Unified report button is included under the Foreign Variation Margin drop down menu. It possesses two options "View" and "Upload Eagle File". The "View" option allows the user to display the new Unified report in a new screen generated by the application. The "Upload Eagle File" option allows the user to upload the Eagle file by displaying an upload box. The Upload box contains a file date and a file

picker to pick which file to upload. The application will extract data from the Eagle file for the analysis of the Unified report processes.
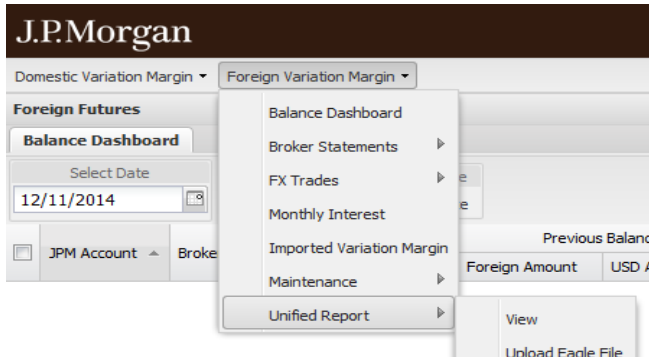

Figure 18: New eagle file upload window
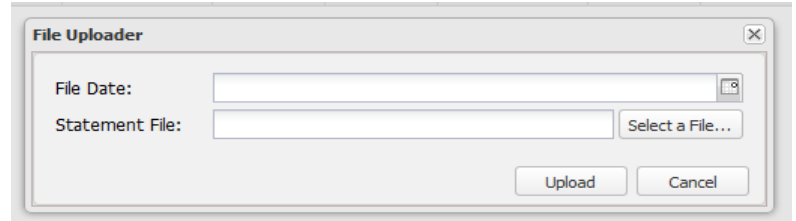

Figure 19: New menu options

2. Unified Report Screen

The Unified Report Screen displays the Recon, No Activity and ForCash reports in a single screen. This feature allows the Operations team to analyze the three processes simultaneously. This new screen has sections for account information, the No Activity report, the Recon report, and the ForCash report. Under each section are columns with data relating to each. This feature offers error detection by highlighting the erroneous row in red.

Common Columns				No Activity			Recon			ForCash		
JPM Account	Broker Account	Broker	Currency	JPM Balance	JPM Prev Balance	Difference	JPM Balance	Broker Balance	Difference	JPM Balance	Eagle Balance	Difference
104305	001-63729	GSCO	SEK	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
104305	001-63729	GSCO	HKD	-559424.00	-559424.00	0.00	-559424.00	-559424.00	0.00	-559424.00	286994.00	-846418.00
104305	001-63729	GSCO	AUD	8263.40	71630.22	-63366.82	8263.40	35743.80	-27480.40	8263.40	-17644.11	25907.51
104305	001-63729	GSCO	EUR	132044.00	125544.00	6500.00	132044.00	132044.00	0.00	132044.00	95450.00	36594.00
104305	001-63729	GSCO	CAD	-64744.75	-93864.75	29120.00	-64744.75	-64744.75	0.00	-64744.75	-2360.25	-62384.50
104305	001-63729	GSCO	GBP	211306.73	205506.73	5800.00	211306.73	211306.73	0.00	211306.73	171571.78	39734.95
104305	001-63729	GSCO	JPY	36720000.00	35760000.00	960000.00	36720000.00	36720000.00	0.00	36720000.00	13440000.00	23280000.00

Figure 20: Screenshot of unified error report view

The new process automated the majority of the steps necessary to complete the current reconciliation process. The addition of the new features to the system significantly reduced the time necessary for the completion of the analysis. The new process only requires the user to upload two files and go to a website to display the new Unified Report window.

Conclusion

Future Work and Recommendations

Due to time constrains a few of the features could not be finalized. The following are some of the features that could be added in the future to further improve the application:

1. Account tie out screen

The account tie out screen serves the purpose of displaying the account tie out report. This report cross checks the accounts in the Eagle file and the accounts in the Broker file to make sure that all accounts are present in both. As mentioned earlier, it does not make sense to put this result in the same place as the results from the other steps in the reconciliation process. Accordingly, there should be a separate screen to display the account tie out.

2. Store Eagle file by Creation date

The current "Upload Eagle file" option allows the user to upload the eagle file and specify the date the

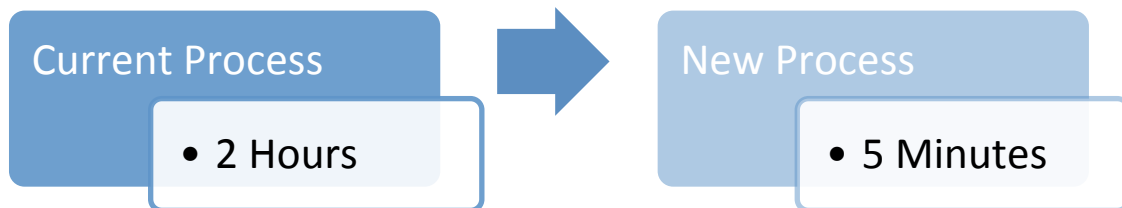Current Process
• 2 Hours

New Process
• 5 Minutes

Figure 21: Time improvement

eagle file is being uploaded. However, for the sake of retrieving previous uploaded Eagle files and accurate data retrieval it is desirable that Eagle files could be stored by creation date.

3. Delete and Update stored Eagle files

Eagle files that are uploaded to the database cannot be removed directly from the user interface currently. A feature to allow the Operations team to remove unwanted files from sight would be quite valuable. Future versions of the application would have the option to display such files, replace them with updated version and delete older versions files.

4. Days of No Activity tracking

Besides the comparison of previous day balance and current day balance, part of the No Activity process is to determine how many days has it been since any change happened within an account. The balance in an account should be withdrawn if no activity has happened for an extended period of time.

The following chart summarizes the above recommendations and how long they may take to complete.

Recommendations	Reasons	Time Estimate
Account Tie Out Screen	• Summarize accounts data automatically • Identify missing account more easily	2 weeks
Store Eagle File by Creation Date	• Get more accurate file information	Half week
Add Ability to Delete and Update Eagle Files	• Provide flexibility for editing files	Half week
Keep Track of the no activity days	• Determine account status more accurately	Half week

The following improvements we see necessary for a better performance of the entire system:

1. Improve loading speed

After analyzing the application it became apparent that certain screens required a significant amount of time to load. This loading time extends the analysis process. The exact cause of this delay is unclear and would require further investigation to pinpoint. Perhaps more efficient logic could be added to the system in order to display the screens faster. Resolving this issue would increase the work efficiency and reduce the non-value added time to the processes.

2. Eagle file and Application data discrepancies

Since the Eagle file and the Variation Margin Application both get their data from the same location, it is not clear why sometimes the does not match. A speculated reason is that the Eagle file is created at a one point in time and the Variation Margin Application updates at another time and the underlying data in the Eagle database changes between these times. It would be appropriate to resolve this discrepancy because it would reduce reconciliation process, essentially eliminating the need for the ForCash test.

Below is a chart summarizing our recommendations for the system as a whole and an estimated time it would take to complete.

Recommendations	Reasons	Time Estimate
Improve Loading Speed	• Reduce non-value added time • Increase work efficiency	2 weeks
Resolve Eagle file and application data discrepancies	• Reduce the work for reconciliation analysis (ForCash) • Save time	N/A

Time Distribution

The project duration was seven weeks. The diagram below shows the distribution of the different tasks in order to deliver a prototype.
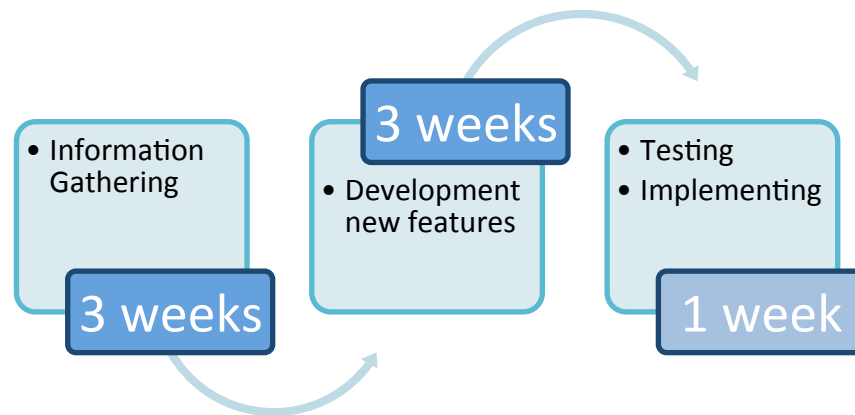

Figure 22: Time utilization to complete the project.

Reflections

The team collaborated with a coworker located in Glasgow headquarters. The collaboration allowed exposure to managing schedules considering time zone differences. The team learned new programming skills (Java, JavaScript, and SQL) and the use of several programming tools (Eclipse, Oracle SQL, and Variation Margin Application). The team also did extensive research on Foreign Futures and learned how the firm settles its trades.

Each team member contributed his or her own expertise to this project. Jiahui, an industrial engineer, contributed mainly to the business analysis portion of the project. She used her classroom experience with optimizing assembly lines and applied it to the optimization of this business workflow. She contributed ideas such as the use of decision trees to help in the planning process. Daniel is an electrical engineer and contributed organizational expertise as well as programming expertise. His experience planning and writing labs was valuable when trying to organize ideas and laying out plans. His knowledge of computer systems was also useful when turning the plans that were made into reality. Kevin is a computer science major and did the bulk of the programming for this project. He turned the plan that was created in the business analysis phase into a working prototype, and ensured that the other partners were informed enough about the inner workings to be able to discuss the new features in an intelligent way. Overall the varied expertise of the team members worked as an advantage so that there was a person with expertise in each phase of the project from planning through development.

Appendix

Business
Requirements Docum

Works Cited

Heakal, R. (n.d.). *Futures Fundamentals: How the Market Works*. Retrieved 2014 йил 18-12 from investopedia.com: http://www.investopedia.com/university/futures/futures2.asp

J.P. Morgan Chase & Co. (2014). *About Us*. Retrieved 2014 йил 18-12 from jpmorganchase.com: http://www.jpmorganchase.com/corporate/About-JPMC/about-us.htm

J.P. Morgan Chase & Co. (2014). *Asset Management*. Retrieved 2014 йил 18-12 from jpmorgan.com: https://www.jpmorgan.com/pages/jpmorgan/am

J.P. Morgan Chase & Co. (2014). *Company History*. Retrieved 2014 йил 18-12 from jpmorgan.com: https://www.jpmorgan.com/pages/company-history

Price Risk Management. (n.d.). Retrieved 12 2014, from foex.fi: http://www.foex.fi/index.php?page=alias-3