### FUND ACCOUNTING SYSTEM

*Submitted in partial fulfillment of the requirements For the award of the degree of*

**Master of Computer Application (MCA)**

Guru Gobind Singh Indraprastha University, Delhi

Internal Guide: Mr. Manoj Purohit Submitted by: External Guide: Mr. Jatin Rijhwani Rishabh Porwal

Enroll no.03811804416



Centre for Development of Advanced Computing, Noida

2015-2018

### Certificate

##### I, Mr. Rishabh Porwal, Roll No. 03811804416 certify that the Project Report/Dissertation (MCA-302) entitled “ FUN D ACCOUNTIN G SYSTEM” is done by me and it is an authentic work carried out by me at “ThoughtFocus Information Technology” . The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student Date:

Certified that the Project Report/Dissertation (MCA-302) entitled “**FUND ACCOUNTING SYSTEM”** done by Mr. **Rishabh Porwal** , Roll No. **03811804416**, is completed under my guidance.

### 

**Certificate**

##### I, Mr. Rishabh Porwal, Roll No. 03811804416 certify that the Project Report/Dissertation (MCA-302) entitled “ FUN D ACCOUNTIN G SYSTEM” is done by me and it is an authentic work carried out by me at “ThoughtFocus Information Technology” . The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Signature of the Student Date:

Certified that the Project Report/Dissertation (MCA-302) entitled “**FUND ACCOUNTING SYSTEM”** done by Mr. **Rishabh Porwal** , Roll No. **03811804416**, is completed under my guidance.

Signature of the Guide

Date:

Name of guide: Mr.ManojPurohit

Designation:STO Address:B-601,Shivalik Homes,Surajpur Site-C,Zeta-1,Greater Noida 201306

### Acknowledgment

I wish and express my heartfelt gratitude to my external project guide Mr. Jatin Rijhwani for his guidance and suggestions throughout the project, he was there in each and every stage to assist and motivate, so that I could come up with good work and without his faith and trust this project would not have been completed successfully.

I wish to sincerely thank my internal guide Mr. Manoj Purohit , for his sincere guidance throughout the project.

I wish to sincerely thank my parents, all my friends, and faculties of CDAC for their valuable suggestions and ideas while carrying out this project.

Rishabh Porwal

03811804416

MCA

6th Semester

CDAC, GGSIPU

# Index

Certificate.........................................................................

Acknowledgement...........................................................

List of figures…………………………………………….

Abstract……………………………………………………

1. Brief Description of the Organization………………………….
   1. Blackstone
   2. BXI - Blackstone Innovations
   3. Thoughtfocus Information Technology
   4. Blackstone Centre of Excellence Thoughtfocus

2. Introduction……………………………………………

* 1. Testing
  2. Fund Accounting System
  3. Need for Fund Accounting System

3. Objective………………………………………………

* 1. Environments Involved
  2. Teams Involved
  3. My Objectives in the Testing Team

1. Testing Activities……………………………………..
   1. Unit Testing
   2. Integration Testing
   3. Smoke Testing
   4. System Testing
   5. Regression Testing
   6. Automation testing
2. Design of test suite file……………………………
3. Business workflow………………………………

7. System Design……………………………………

* 1. Use Case Diagram

ii iii Vi Vii

01

04

10

14

24

29

30

* 1. Data Flow Diagram
  2. Database Design
  3. Entity Relationship Diagram

1. Methodology and Tools………………………………… 37
   1. Agile Methodology for Analysis and Design
   2. Benefits to the customer
   3. Benefits to the project teams
   4. Phases
   5. Agile Development Methodology
   6. Agile practice developed in this project

9. Test Cases……………………………………………… 51

1. Bugs…………………………………………………… 62
2. Hardware and Software Requirements……………… 67
   1. Hardware Requirements
   2. Software Requirements

12. Softwares Used……………………………………… 68

1. Progress Report…………………………………… 69
   1. Gantt Chart
   2. Tasks Performed by me

14. Conclusion………………………………………… 71

15. References………………………………………… 72

### List of Figures

1. Fund Accounting System …………………………… 25
2. Business Workflow of Fund Accounting System…… 26

3. Use Case Diagram…………………………………… 28

4. Context Level Diagram……………………………… 29

5. Level-1 Diagram…………………………………… 30

6. ER Diagram…………………………….…………… 31

1. Agile Development Phases…………………………. 35
2. Agile Development Methodology………………… 77
3. Life Cycle Of a Bug……………………………… 62

10.Gantt Chart………………………………………. 69

### Abstract

Web Application Fund Accounting System for Blackstone (Client) is a tool that defines how cash flows that is, investments & gains from investments should be distributed amongst fund partners. It automates the process of calculating profit shares from investments. It performs calculations, creates reports & normalizes the effect of one transaction on transactions committed previously. It is an internal tool only used by one Client.

Fund Accounting System is an MVC web application that users use for fund accounting. The web application (aka the web tier) exposes a number of interfaces for the users to enter transactional parameters. When the user saves parameters/allocations into a transaction interface, the transaction parameters are persisted to the database using the application tier. When booking transactions in this Engine, some sort of capital or amount flows FROM one entity TO another.

For each such transaction, a quarterly account report is generated. The quarterly report transaction produces the Quarterly Account Statement. This is a report produced quarterly and provided to the L.P.'s to show them the value and performance of the Fund they invested in.

#### CHAPTER 1

##### Brief Description of the Organization

* 1. **Blackstone**

Blackstone is one of the world’s leading investment firms. They seek to create positive economic impact and long-term value for their investors, the companies they invest in, and the communities in which they work. They do this by using extraordinary people and flexible capital to help companies solve problems. Their asset management businesses with almost $336.4 billion in assets under management include investment vehicles focused on private equity, real estate, public debt and equity, non-investment grade credit, real assets and secondary funds, all on a global basis. Blackstone also provides various financial advisory services, including financial and strategic advisory, restructuring and reorganization advisory and fund placement services.

##### BXI - Blackstone Innovations

This is one of the business units of Blackstone. BXII team seeks to drive improvements across Blackstone’s businesses through technology. Technologists thrive on the sharing of ideas while designing and building

world class technology through an iterative process. Blackstone technologists play an integral part in helping to bring innovative technology solutions to a fast paced growing firm.

##### ThoughtFocus Information Technology

ThoughtFocus is a leading Product Engineering, Software Services and KPO Company founded in 2004. They are operating globally from 4 development centres in India and 3 locations in US. They are a team of 500+ associates, driven to help organizations across the globe realize their business-driven technology initiatives. Their solutions encompass innovative ways of aligning business with technology, to create an integrated, enterprise-wide IT platform that supports enterprise goals. With their proven methodology, domain expertise, cutting edge technology and research, they empower their customers to gain competitive edge in market place.

They are happy to engage with enterprises of all sizes and across industry segments. As an organization, they strive hard to build lasting relationships and drive their engagements with customers, associates and vendors with one principal goal 'Focused on Your Success.

* 1. **Blackstone Centre of Excellence Thoughtfocus:** Blackstone, in partnership with ThoughtFocus technologies has contrived to form the Centre of Excellence in Gurgaon. We as consultants to The Blackstone Innovations & Infrastructure Technology team seek to refine the use of technology across Blackstone’s business groups by continuous innovation of new & streamlining old business processes.

Mr. Jatin Rijhwani (Mentor at Blackstone COE Thoughtfocus)

#### CHAPTER 2

**Introduction**

##### 2.1 Testing

Software testing enables making objective assessments regarding the degree of conformance of the system to stated requirements and specifications. Testing verifies that the system meets the different requirements including, functional, performance, reliability, security, usability and so on. This verification is done to ensure that we are building the system right. In addition, testing validates that the system being developed is what the user needs. In essence, validation is performed to ensure that we are building the right system. Apart from helping make decisions, the information from software testing helps with risk management. Software testing contributes to improving the quality of the product.

##### Fund Accounting System

This tool defines how cash-flows, that is, investments & gains from investments should be distributed amongst fund’s partners. It automates the process of calculating profit shares from investments.

It is used to perform calculations, create reports & normalize the effect of one transaction on transactions committed previously. It is an internal tool only used by the organization’s fund accountants.

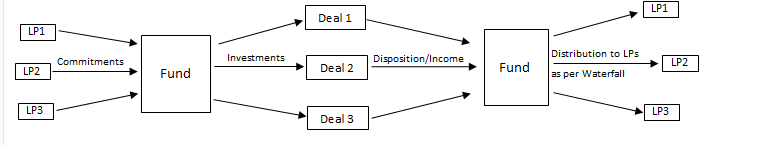


FIG. 1 FUND ACCOUNTING SYSTEM

##### Transaction Details:

* + - Partner Agreement: It tracks the Limited Partner Agreement and the terms and conditions of the fund. This includes the details of the distribution model i.e. the rates applied at different phases of distribution model, the fee basis and the associated rates and other significant details.
* Fund Phase:It tracks the various fund events like End of Commitment Period, Start of Investment Period, End of Investment Period, etc.
* Fund Closings:It tracks the amount promised by various investors at the beginning of the fund. There can be multiple such commitments from a single investor as well till the time a fund is in committing period.
* Capital Call: It tracks the amount which is drawn by various investors with respect to a particular deal and tranche on a particular date.
* Blockers:It tracks the amount kept as a blocker. It could be a general one or deal specific. The amount can be used to invest in the deal at a later stage of the fund.
* Investment: It tracks the amount which is invested by various investors with respect to a particular deal and tranche on a particular date.
* Reserve: It tracks the amount kept as a reserve. It could be general or deal specific. No amount is actually drawn from investors in this transaction.
* Earnings: It tracks the earnings generated by a deal.
* Disposal of an Asset: It tracks the amount generated by a deal when it is disposed. Disposition can be partial or full. It also tracks any loss recoupment or fee management which needs to be done.
* Distribution: It tracks the amount which the investor gets after the profits (or loss) generated by a deal are flown from the distribution model.
* Return of Excess Capital: It tracks the amount returned to investors which is not invested in any deal yet.
* Debt: It tracks the amount used as debt by investors.
* Management Fees: It calculates and tracks the fees for managing the fund for every quarter.
* Fees: It tracks other expenses incurred while managing a deal or fund.
* Valuation: It tracks and calculates the worth of a deal if it were disposed completely in this quarter.
* Quarterly report: It is one of the reports generated by the Fund Accounting System which describes how well the Fund is performing. It details the realized and unrealized profits and losses.
* Appreciation/Depreciation: It tracks the current value of the deals and tracks any appreciation or depreciation required.
* Commitment Transfer: It tracks the transfer of commitment from one investor to another.

##### Need of Fund Accounting System

* + 1. **Removes overhead of manual calculation of data by fund accountants:** Previously, the required reports like quarterly financial statements were presented through manual calculations by the Fund Accountants. The entire process was dependent on Excel Sheets for all the calculations but now the present system will overcome the demerits of the manual process and minimize the time required by the Fund Accountants to prepare the reports. This system will create reports which are easy to visualize by using diagrams, pie charts and tables. This also reduces the chances of any manual error that fund accountants may do since the work was being done manually only.
    2. **Calculation Accuracy:** All the data that is generated through a series of calculations are validated at the QA environment and then pushed onto the UAT server of the Web Portal. Only when the data is validated and all bugs are fixed, the data is pushed to production. Since this is financial data, it is important for this data to be correct.
    3. **Faster and more accurate generation of reports:** The Fund Accounting System is proposed to automate the manual process of calculating profit shares from investments based on a distribution model and generating reports for the investors. The main deliverable of the Fund Accounting System is to generate quarterly reports for the investors in a format that is directly distributable to them. The report has different sections pertaining to different data being represented in different formats. The data should be well represented and easy to visualize. The quarterly reports basically let the investors know how well the fund is performing and how much the capital that they invested in that fund is worth as of that quarter’s end.
    4. **Security:** Security Enabled, Password and Email Id checked before logging into the System.

CHAPTER 3

### Objective

The Fund Accounting System is to be tested thoroughly using various black- box testing techniques such as: Unit Testing, System Testing, Sanity Testing, Regression Testing and Smoke Testing. All these techniques are performed manually. All the functionalities are tested and approved by the testers. If there exists any bug or issue, the associate is to report and log the bug at once, and then it is the developer’s job to fix and move to awaiting validation state, where the testing associate checks whether the bug is fixed or not.

##### Environments Involved:

There are various environments involved in this process:

**3.1.1 QA Environment:** This environment is specifically for the Quality Assurance team. The enhancements and patches are added to this stage to test for individually.

* + 1. **Dev Environment:** The developers unit test the modules on this environment. After they have tested their modules they deploy the code to the QA environment.
    2. **UAT Environment:** This is a copy of the actual product that is provided to the users. Once the enhancements are tested and approved by the testers, it is included in this environment. User testing is done in this environment.

**3.1.4 Production Environment:** This is the main environment that is deployed to the user.

##### Teams Involved:

There are various teams involved in the project:

* + 1. **Development Team:** The team consists of efficient developers who code the product, patches and enhancements. They are responsible for the additions of the functionalities on different environments.
    2. **Testing Team:** The team consists of efficient testers who test the product, patches and enhancements released by the developers. They are responsible for the approval and reporting of a bug, if there exists, so that the developers can fix it. They are also responsible for the creation and review of test plans and test cases.
    3. **Business Analyst Team:** The team consists of efficient business analysts who gather requirements and analyse them to

understand its feasibility. They are responsible for the creation and review of design plans.

##### My objectives in the Testing Team as an Intern:

The organization works on a quarterly release process, that is, a final release of the enhancements are done quarterly. And since the project uses Agile methodology, every day a scrum is performed so as to understand your tasks of the day according to the requirements in hand. As an intern, I have been responsible for the following activities which are a part of the Testing process that is ongoing in the organization:

* + 1. **Regression Testing:** It is done manually or through automation for all the functionalities that have been affected by the new patch or enhancement released by the developers on the QA environment. It is performed whenever a new patch or an enhancement is deployed. Previous test cases are used to perform this testing. I am required to test and report the bugs on Jira, it is an online tool used as a Project Management tool. Once the code is ready to be deployed to prod the code is first moved to UAT environment to be tested before deployment.
    2. **Integration Testing:** It is done to check if the existing functionality is working perfectly fine with other modules of the project. It is done to check if the enhancement is working perfectly with all the other modules in the existing project. New test cases are formulated and tested. I am required to test and report the bugs on Jira, it is an online tool used as a Project Manager.
    3. **Smoke Testing:** A special kind of ‘Build Validation’ testing is performed once the functionality is added to the QA and QB environments. This is done after the release to check if the code is properly deployed on the QA and QB environments. This is performed once in a quarter. No tools are used since, this is performed manually.
    4. **Quarterly Report Testing:** The main aim of the project is to generate Quarterly Accounting Reports that includes all the details of the investments and profits of the partners. Every transaction/functionality is a part that in the end helps to achieve this report. Every time functionality is affected by the patch or enhancements, the Quarterly Accounting Reports are affected too. So, the reports are generated and tested using Microsoft Excel, if the calculations are performed accurately.

## CHAPTER 4

### Testing Activities

Unit testing is done on respective local environments manually. After completion of unit testing and defect fixing, the code will be checked in the configuration management system. The tester will carry out integration testing once the unit tested code is made available to them.

The testing team will then perform system testing and the defects uncovered will be logged. Defects will be assigned to associates. It is then the responsibility of the associates to fix the defect, check back in the code and update status of the defect. Any code fixing will be carried out in the development environment only. After the code is fixed it will follow the complete test cycle through various environments and for this unit, integration and system testing is done on various use cases.

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. In this project testing is done mainly to see whether all modules perform requested tasks in the manner that they should.

The main steps in the testing process are:

1. Plan Activity
2. Develop Tests
3. Prepare Testing Scope
4. Run Tests
5. Preview

##### Unit Testing

Unit testing is a software design and development method where the programmer gains confidence that individual units of source code are fit for use. Unit tests are typically written and run by software developers to ensure that code meets its requirements and behaves as intended. This is done specially in our Application since for any code to be deployed successfully; the unit test coverage should pass 75% for any newly created/updated class. The goal of unit testing is to check that each functional unit, Layer and respective file’s written are correct. Each function and file is tested in isolation with the respective method and tools.

1. The module interface is tested to ensure that information properly flows in and out of program.
2. Local data structure is examined to ensure the data stored temporarily maintain its integrity.
3. Boundary conditions are tested to ensure that modules operate properly at boundary limits of processing.
4. All independent paths are exercised to ensure all statements in a module have been executed at least once.
5. All error handling paths are tested.

##### Integration Testing

'Integration testing' is the activity of software testing in which individual software modules are combined and tested as a group. It comes between unit testing and system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies and delivers as its output the integrated system ready for system testing.

1. Entry Criteria
   * The entire Unit test Bugs are fixed.
   * All the new features are unit tested.
2. Exit Criteria
   * Code is peer reviewed.
   * Basic functionality is achieved.
   * All requirements been tested or verified.

##### Methodologies Used

Integration testing is carried out by the testers on a QA environment to test the functioning of various modules when integrated with each other.

##### Tools Used

Manual testing and automation through selenium is carried out on the QA environment.

##### Smoke Testing

Smoke Testing, also known as “Build Validation Testing”, is a type of software testing that comprises of a non-exhaustive set of tests that aim at ensuring that the most important functions work. The results of this testing is used to decide if a build is stable enough to proceed with further testing.

##### Methodologies Used

Smoke Testing is carried out by the testers on the QA as well as the QB environment to exhaustively check the working of all the functionalities of the system.

##### Tools Used

Manual testing is carried out on the QA as well as the QB environments.

##### System Testing

System testing of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limiting type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

1. Entry Criteria
   * All the screens are unit tested.
   * All test cases for System Tests are prepared.
2. Exit Criteria
   * All critical bugs are fixed.
   * If any bugs remain should not harm the basic functionality.
   * No data corruption errors.
   * No Incorrect data or data recorded in the wrong field, account or Database.

• No Significant performance errors.

* + No System crashes, instability or unpredictable errors.

##### Methodologies Used

System testing is carried out by the testing team on the QB environment formulated for them. System testing is carried out taking the use case description as the base.

##### Tools Used

Manual testing and automation testing through selenium is carried out by the testers on the QB environment.

##### Regression Testing

Regression testing is a type of software testing which verifies that software which was previously developed and tested still performs correctly after it was changed or interfaced with other software. Changes may include software enhancements, patches, configuration changes, etc. During regression testing, new software bugs or regressions may be uncovered. Sometimes a software change impact analysis is performed to determine what areas could be affected by the proposed changes. These areas may include functional and non-functional areas of the system.

The purpose of regression testing is to ensure that changes such as those mentioned above have not introduced new faults. One of the main reasons for regression testing is to determine whether a change in one part of the software affects other parts of the software.

##### Methodologies Used

Common methods of regression testing include re-running previously completed tests and checking whether program behavior has changed and whether previously fixed faults have re-emerged. Regression testing can be performed to test a system efficiently by systematically selecting the appropriate minimum set of tests needed to adequately cover a particular change.

##### Tools Used

Manual testing and automation testing through selenium is carried out by the testers on the QA environment.

##### 4.6 Automation Testing

Automation testing is a technique that uses an application to implement entire life cycle of the software in less time and provides efficiency and effectiveness to the testing software.

Automation testing is an Automatic technique where the tester writes scripts by own and uses suitable software to test the software. It is basically an automation process of a manual process. Like regression testing, Automation testing also used to test the application from load, performance and stress point of view.

In other word, Automation testing uses automation tools to write and execute test cases, no manual involvement is required while executing an automated test suite. Usually, testers write test scripts and test cases using the automation tool and then group into test suites.

Test cases to be automated can be selected using the following criterion to increase the automation ROI

* + High Risk - Business Critical test cases
  + Test cases that are executed repeatedly
  + Test Cases that are very tedious or difficult to perform manually
  + Test Cases which are time consuming

The following categories of test cases are not suitable for automation:

* + Test Cases that are newly designed and not executed manually at least once
  + Test Cases for which the requirements are changing frequently
  + Test cases which are executed on ad-hoc basis.

##### 4.6.1 Automation Framework

The purpose of the automation framework is to reduce the efforts of manual team while doing functional and regression testing. Many types of testing frameworks are used in the industry including Linear Scripting Framework, Modular Testing Framework, Data Driven Testing Framework, Keyword, Driven Testing Framework, Hybrid Testing Framework, and Behavior Driven Development Framework.

In Blackstone we use Hybrid Framework which is a combination of Keyword Driven and Data Driven approaches. The framework works in a way that it reads test cases from excel files, executes the actions defined in terms of keywords and generates a HTML report.

The idea behind the *Keyword Driven* approach is to separate the coding from the test case & test step. Each manual action has a corresponding keyword, like click on any object of webpage has action\_click keyword, etc. Using these different keywords in steps helps to automate a test case. It also supports data driven approach wherein you can run a single test case with different values by passing test data in an excel file. Internally, it is a JAVA framework which uses different open source tools like Selenium, EWS, JIRA Rest Client, etc. to automate tests and increase test coverage for web applications.

##### Working of the Testing Framework:

Any script which gets created using Hybrid Framework requires four files:

1. **Test Suite File:** This file contains a set of test-scripts which needs to be executed.
2. **Test Script File**: This file contains set of test cases along with the test steps.
3. **Object Repository**: A common location where locators are stored.
4. **Application URLs:** A common location where application URLs will be stored.

## CHAPTER 5

### Design of Test Suite File

Test Suite file: This file contains a set of test-scripts which needs to be executed. It has two sheets in it:

* 1. **Test Suite**: This sheet will have the name of test scripts which needs to be executed. It consists of three fields:

**TSID:** Name of the test script.

**Description:** Non-mandatory. A brief description about the test script.

**Run Mode**: A flag (Y/N) which decides whether to run that test script or not.

* 1. **Configuration**: This sheet contains all information which remains fixed in entire run. In short, it is a configuration file. Following parameters must be set in Config Sheet:

**browserType**: Browser name where script needs to be executed. Chrome/Firefox/IE

**implicitWait**: Default time a script will wait for an element. It is passed in seconds.

**JIRA Suite and Script Ids:** Test Suite and Test Scripts Ids are required for reporting.

##### 5.1 Design of Test Script Files

Test Script File: This file contains the set of test cases which needs to be automated along with their steps. A simple test script file will have following sheets:

***a. Test Cases:*** *This sheet contains the name of test cases which has been created. It consists of 5 fields:*

**TCID**: Name of the created test case like Login Test **Description**: A non-mandatory field where a brief description about the test case is written

**Jira Test Id**: A non-mandatory field used for reporting purpose. If you have a test case which you want to report after this case execution, you need to mention it in this field. This should be JIRA Test Case Id.

**Reportable Test Case**: The field when left blank qualifies itself in test case count. The significance of this column comes into

Picture when you are automating a test case which has a pre- requisite test case.

**Run Mode**: Flag(Y/N) which decides if a particular test case will execute or not.

1. **Test Steps**: This sheet contains the actual steps which are followed while automating the test case.

**TCID**: Name of test case. This should be same as that mentioned in TCID column of Test Cases Sheet.

**TSID**: Non mandatory but contains serial number

**Description**: A non-mandatory field which contains a description of keyword supposed to be used. This column helps to describe what the person wants to accomplish at this step.

**Keyword**: Contains a keyword name like action\_click .

**Object**: It contains the name of the locator which is present in the object repository (The name of the locator must be equal to the name used in test sheet). Locators are elements on your web page that you needs to interact with while running the automation scripts

**Data**: Contains test data like username, password which you want to pass.

**Proceed\_on\_Fail:** It’s a non-mandatory field. Its main significance is when you have added certain minor UI validations, and it gets failed; but you can still continue with your test case execution. Adding Y at this step will execute the next test step despite failure

**Depends\_on\_Test**: It’s also a non-mandatory field. This field is used if you have a test case which should run when a different test case has passed

1. **Test Case Data Sheet**: This sheet contains test data which is required in test steps .The name of the data sheet must be same as of test case name.

**Iteration Name**: Name of the iteration to be executed

**Iteration JIRA Id**: This is a non-mandatory field. This column is used for reporting execution status of test case on JIRA

**Run Mode**: A mandatory field which determines the execution mode of iteration.

**Result**: This column must be present. Its execution status gets updated by itself. It displays PASS for Passed Tests, FAIL for Failed Tests, and SKIP for non-executed tests, PRE REQUISITE- NOT-MET if the test case is dependent on another test case and it has failed.

***5.2 Design of Object Repository***

Object repository contains the alias name of the object and its actual path.

**Object Name**: it contains the name of the object which we use in the Test script in Object column .The name in OR should be the same as that in Test script file.

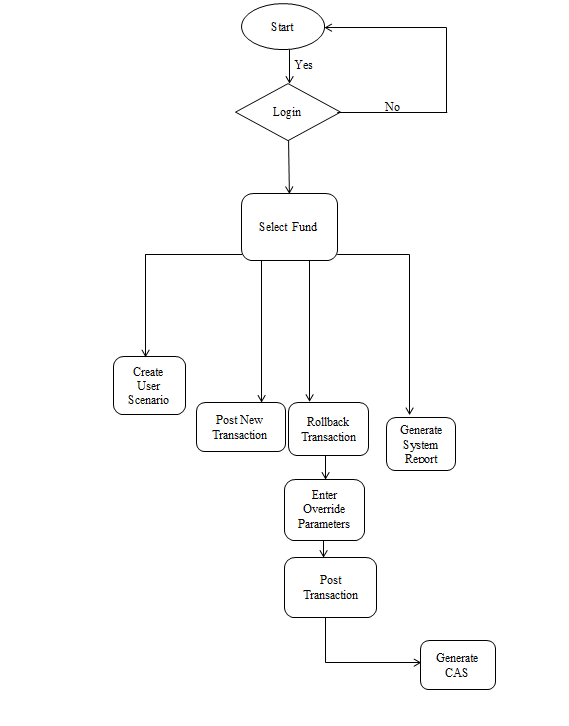
**Object Path**: It contains the locator of the object. There are 8 different types of locators - Id, Name, LinkText, Partial Linktext, Tag Name, Class Name, CSS and Xpath.

## CHAPTER 6

### Business Workflow

In this section, the workflow of the various important business transactions has been depicted to explain the process of the execution of the activities.

##### Fund Accounting System:



Generate

Quarterly Report

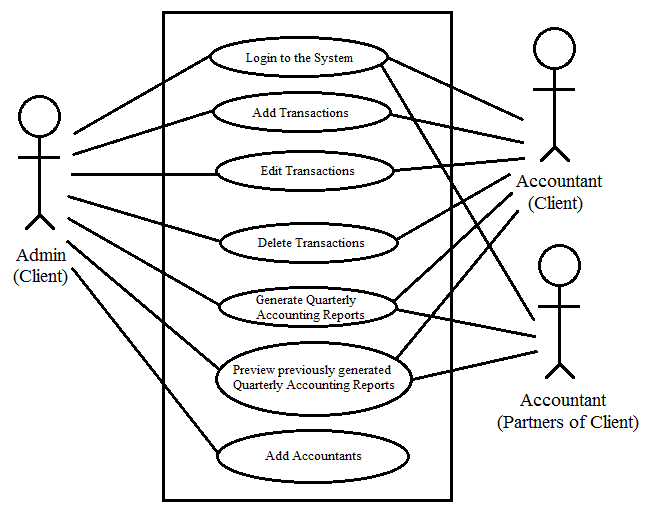
*Fig 2: Business Workflow of Fund Accounting System*

## CHAPTER 7

### System Design

##### Use Case Diagram

The use case diagram represents a set of actions (use cases) that the Fund Accounting System can perform in collaboration with one or more external users of the system (actors).



*Fig:3: Use Case Diagram*

##### Users:

* + - * **Admin of the Client:** It has all the rights over the system.
      * **Accountant of the Client:** This is the person who takes care of all the transactions that are to be posted in order to generate the Quarterly Accounting Reports.

##### Accountant of partners of the Client

* + 1. **Uses Cases:**

The following are the use cases identified in the system:

* + - * **Login to the System:** The users can login to the system using the registered email address and password.
      * **Add Transactions:** Once the user is logged into the system, the transactions can be added to the system that is required to generate the Quarterly Accounting reports.
      * **Edit Transactions:** The existing transactions can be edited and saved in case an erroneous transaction is accidently entered in the system.
      * **Delete Transactions:** The existing transactions can be deleted as well.
      * **Generate Quarterly Accounting Reports:** The Quarterly Accounting Reports can be generated by the

Users which gives you the details of the investments and profits incurred by the client and its partners.

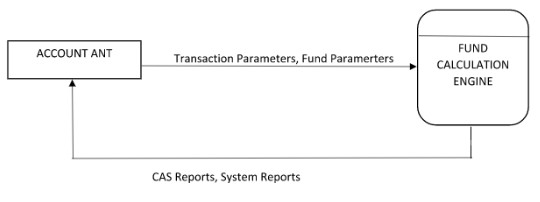
* + - * **Preview Previously Generated Quarterly Accounting Reports:** The users can view the previously generated reports from the repository maintained by the system.
      * **Add users:** The admin can add different kinds of users that have different rights over the system.

##### Data Flow Diagram

The data flow diagram represents the flow of information among the various processes of the system. The context level diagram defines the scope of the system in terms of various responsibilities bestowed to the system. Level 1 Diagram shows the major subsystem of the Fund Accounting System and the interaction among them.

##### Context Level Diagram

The overall scope of the system is represented through context level diagram. The system interacts with stakeholder for accomplishing the various requirements for execution of the system.

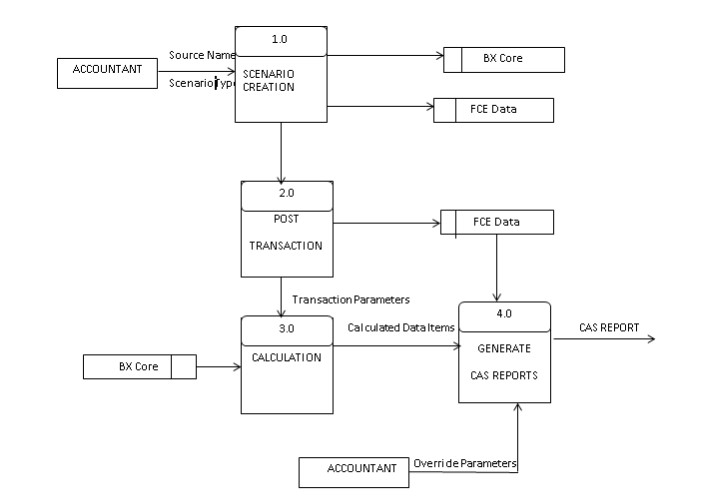


Account

*Fig.4: Context level Diagram*

##### Level – 1 Diagram

The major subsystems of the Fund Accounting System are the modules that generate the Quarterly Account Statement for the investors. The other modules are Calculation, Transaction posting.



*Fig.5: Level-1 Diagram*

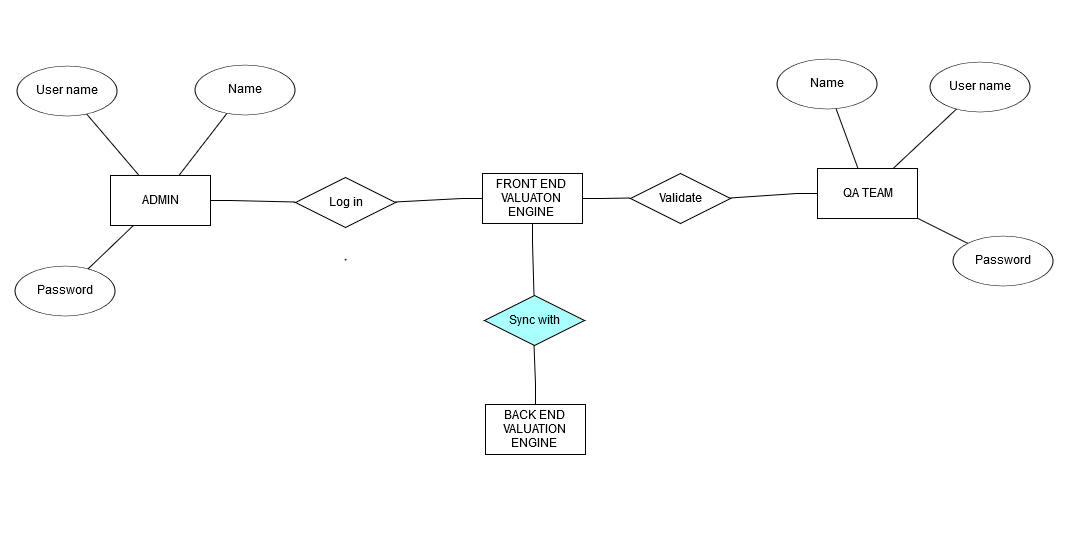
##### Database Design

Once the architecture of system is finalized all the data and process flow is cleared. All the processes are simple at interface level but consisted of many processes running at back to provide efficient functioning of the system.

The database design of the system is most important for the ease of data access and faster retrieval of information required .Due to the large amount of data and scale of the application, a strong and flexible and well-studied database design is needed. After analyzing all the Complex and needed functionalities of the system, the following database tables are obtained.

1. Fund\_table
2. Acc\_table
3. Report\_table
4. Data\_table
5. Trans\_table
6. TransAlloc\_table
7. DataValueOverride\_table
8. DataTransactionValueQB\_table
9. DataValueNumericQB\_table
10. Scenario\_table
11. ReportToDataGroup\_table
12. DataGroup\_table
13. DataGrouping\_table

##### Entity Relationship Diagram:



*Fig.6: ER Diagram*

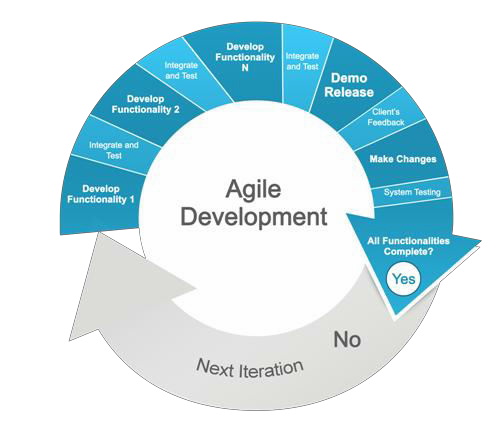
## CHAPTER 8

### Methodology and Tools

##### Agile Methodology for Analysis and Design

Agile software development is a group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross- functional teams. It promotes adaptive planning, evolutionary development and delivery, a time-boxed iterative approach, and encourages rapid and flexible response to change. It is a conceptual framework that promotes foreseen tight interactions throughout the development cycle.

The Agile movement proposes alternatives to traditional project management. Agile approaches are typically used in software development to help businesses respond to unpredictability.



*Fig.7: Agile Development Phases*

##### Benefits to the Customer

* + - Customer is more actively involved, and gets higher priority.
    - He gets to know regular and frequent status of the application
    - Requirements are accepted after each iteration
    - Since the methodology emphasizes rapid delivery, time-to-market is less. So the key functionalities can be available to use sooner.
    - Delivery is defined by fixed timescale. So customer is assured of receiving some functionality by a fixed time period.
    - More Testing is done, so better software quality is delivered

##### Benefits to the Project Teams

* + - Project teams are involved more actively in all the stages, have to ask right question. The teams collaboratively take the decisions and are more empowered.
    - Since the development is Incremental, teams can focus on the specific requirements at any given point of time.
    - The teams receive frequent feedback as the testing is integrated; so the rework is reduced.
    - Less time is spent in gathering requirements as all the requirements are not gathered upfront and are implemented as and when they arise.
    - So less time is required for planning.
    - Less cost of development as rework, management, documentation and other non-development work related cost is reduced.
    - Teams develop applications collaboratively and in cooperative environment.

##### Phases:

**8.4.1. Requirement Phase**

For the systematic development of any project, it is necessary to document the requirements and to sign-off the requirements. After completion of the requirements documentation, it is necessary to get the approval from the authorities to proceed further. This project started with the Requirement Engineering process which covered up the Requirement Elicitation, Requirement analysis and Requirement documentation. Requirements were gathered from all the senior managers in various meetings and sessions, writing down the objectives and sorting out the requirements which was

Followed by evaluation of each and every requirements which includes the covering up the “why you need the particular requirement?”

##### Design Phase

Design phase commences after the requirements are base lined and signed off. The design phase attempts to uncover various entities involved in the system and their associated behavior and also the interfaces that would be provided by the system. High level and Low Level Design Documents are prepared along with use cases.

##### Development Phase

At this stage, the development of project application begins. As the application is divided into 2 parts i.e. Website front end and backend process that maintains database for the application. The website pages for the front end implications are purely written in

.NET. HTML and CSS is also an integrated aspect for its development. Parallel to this, for backend SQL Server is used to maintain table’s details which are been generated through various Sql queries. After the completion of above, both the front and back end got integrated so the website could be enriched with the essential use of it.

##### Testing Phase

Individual developers on their respective local environments will do unit testing manually. After completion of unit testing and defect fixing, the code will be checked in the configuration

Management system. The tester will carry out integration testing once the unit tested code is made available to them.

The testing team will then perform system testing and the defects uncovered will be logged. Defects will be assigned to developers. It is then the responsibility of the developers to remove the defect, check back in the code and update status of the defect. Any code fixing will be carried out in the development environment only. After the code is fixed it will follow the complete test cycle through various environments and for this unit, integration and system testing is done on various use cases.

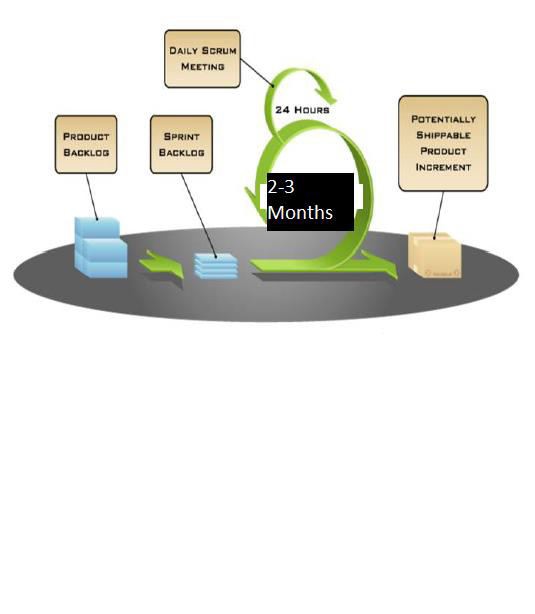
##### Post Implementation Maintenance

Once the product is developed, maintenance starts, which is never ending process. Once the problem is solved, quality records are maintained which includes code review records, test reports. Hosting will be done as soon as project completes, after that maintenance will be done for the same.

##### Agile Development Methodology:

The project undergoes the Agile methodology of software development life cycle. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software project. The proposed approach is to perform testing quicker as vast amount of data needs to be analyzed. The proposed approach uses a variety of techniques to perform the

testing, which includes, smoke testing, regression testing and other techniques with automation.



*Fig.8: Agile Development Methodology*

##### Agile practice adopted in this project

* + 1. **SCRUM:**
       - Scrum is an iterative and incremental agile software development framework for managing product development. It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" challenges assumptions of the "traditional, sequential approach" to product development, and enables teams to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as daily face-

To-face communication among all team members and disciplines involved.

* + - * A key principle of Scrum is its recognition that during product development, the customers can change their minds about what they want and need (often called requirements volatility), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an evidence- based empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team's ability to deliver quickly, to respond to emerging requirements and to adapt to evolving technologies and changes in market conditions.
      * Scrum is a management and control process that cuts through complexity to focus on building software that meets business needs. Management and teams are able to get their hands around the requirements and technologies, never let go, and deliver working software, incrementally and empirically.

##### The Scrum Events

Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined in Scrum. All events are time-boxed. Once a Sprint begins, its duration is

fixed and cannot be shortened or lengthened. The remaining events may end whenever the purpose of the event is achieved; ensuring an appropriate amount of time is spent without allowing waste in the process. The Scrum Events are:

* + - * Sprint
      * Sprint Planning
      * Daily Scrum
      * Sprint Review
      * Sprint Retrospective

##### Important SCRUM Terminologies:

1. Scrum Team:

It is a team of around 7 members. The team is a mixture of developers, tester, DBA etc. All these members work together in close collaboration for a recursive and definite interval, to develop and implement the said features.

1. Sprint:

Sprint is a predefined interval or the time frame in which the work has to be completed and make it ready for review or ready for production deployment. This time box usually lies between 2.5 months.

1. Product Backlog:

Product backlog is a kind of bucket or source where all the tasks are kept. This is maintained by Product owner. Product backlog can be imagined as a wish list of the product owner who prioritizes it as per business needs.

1. Sprint Backlog:

Based on the priority, user stories are taken from the Product Backlog one at a time. The Scrum team brainstorms on it, determines the feasibility and decides on the stories to work on a particular sprint. The collective list of all the user stories which the scrum team works on a particular sprint is called s Sprint backlog.

1. Story Points:

Story points are quantitative indication of the complexity of a test case. Based on the story point, estimation and efforts for a use case is determined. Story point is relative and is not absolute. To make sure that our estimate and efforts are correct, it’s important to check that the use cases are not big. Precise and smaller is the test case, accurate will be the estimation.

1. Burn down chart:

Burn down chart is a graph which shows the estimated v/s actual effort of the scrum tasks. It is a tracking mechanism by which for a particular sprint; day to day tasks are 12 tracked to check whether

the use cases are progressing towards the completion of the committed story points or not.

1. Definition of Done:

A use case is DONE in Scrum, only when it is development and QA complete and the feature is eligible to be shipped to production.

##### How agile Scrum model is incorporated in our team or work?

Our team is responsible for testing Fund Accounting Engine application which works as follows:

1. The Sprint is decided to follow 3 weeks cycle.
2. The Product owner has the prioritized list of test cases in the product backlog.
3. The team decides to meet on some day, before the start of next

Sprint cycle for the “Pre Planning” i.e. grooming meeting.

1. The product owner takes 1 story from the product backlog, describes it and leaves it to the team to brainstorm on it.
2. The entire team discusses and communicates directly to the product owner to have clear understanding of the test cases.
3. In a similar way various other test cases are taken. If possible team can go ahead and size the stories as well.
4. After all the discussion, Individual team member go back to their work stations and
   1. Identify their individual tasks for each case.
   2. Calculate the exact number of hours on which they will be working. How the member concludes these hours.
5. The entire Scrum team meets for the “Planning Meeting”.
   1. Final verdict of the test cases from the product backlog is done and the story is moved to the Sprint back log.
   2. For each story, each team member declares their identified tasks, if required can have a discussion on those tasks, can size or resize it.
   3. The Scrum master or the team enter their individual tasks along with their hours for each story in a tool.
   4. After all the stories are completed; Scrum master notes the initial Velocity and formally starts the Sprint.
6. Once the Sprint has started, based on the tasks assigned, each team member starts working on those tasks.
7. The team meets daily for 15 minutes and discusses 3 things:
   1. What did they do yesterday?
   2. What they plan to do today
   3. Any impediments (roadblocks)?
8. The scrum master tracks the progress on daily basis with the help of “Burn down chart”. In case of any impediments, the Scrum master follows up to resolve those.
9. On Last day of sprint, the team meets again for the review meeting. A `member demonstrates the implemented test cases to the product owner. Team meets again for the Retrospective, where they discuss
   1. What went well?
   2. What did not went well
   3. Action Items.
10. Team again meets for the pre-planning meeting for the next sprint and the cycle continues.

### Epic

An agile epic is a body of work that can be broken down into specific tasks (called “stories,” or “user stories”) based on the needs/requests of customers or end users.

Epics are a helpful way to organize our work and to create a hierarchy. The idea is to break work down into shippable pieces, so that large projects can actually get done and we can continue to ship value to your customers on a regular basis. Epics help teams break their work down, while continuing to work towards a bigger goal.

|  |  |
| --- | --- |
| ***[FAS-21000] Q1-2019 Automation Maintenance*** | |
| **Status:** | Closed |
| **Project:** | FAS |
| **Component/s:** | None |
| **Affects Version/s:** | None |
| **Fix Version/s:** | None |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Epic | **Priority:** | Medium |
|  |  |
| **Reporter:** | Shabib Mirza | **Assignee:** | Shabib Mirza |
| **Resolution:** | Done |  | |
| **Labels:** | AutomationEpic, AutomationMaintenance | | |
| **Remaining**  **Estimate:** | Not Specified | | |
| **Time Spent:** | Not Specified | | |
| **Original**  **Estimate:** | Not Specified | | |

|  |  |
| --- | --- |
| **Epic Name:** | Q1 2020 Automation Maintenance |
| **Epic Status:** | Closed |
| **Start:** | 10/Feb/2020 |
| **Risk Reduction:** | Small |
| **Project**  **Categorization:** | Optimizing |

**Task**

Creating a task is one of the parts of Test Planning. Tasks can be created for design discussion, test case creation, and test case execution. It is created to inform PM, testers and developers about some key issues of the testing process. This includes the testing objectives, method of testing, total time and resources required for the project and the testing environments.).

|  |  |
| --- | --- |
| ***[FAS-21882] Test execution: Report Wizard Data Item Calculation API*** | |
| **Status:** | In Development |
| **Project:** | FAS |
| **Component/s:** | None |
| **Affects Version/s:** | None |
| **Fix Version/s:** | None |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Task | **Priority:** | Medium |
| **Reporter:** | SHABIB MIRZA | **Assignee:** | SHABIB MIRZA |
| **Resolution:** | Unresolved |  | |
| **Labels:** | QATasks, TestExecution | | |
| **Remaining**  **Estimate:** | 0 minutes | | |
| **Time Spent:** | 3 days | | |
| **Original**  **Estimate:** | 3 days | | |

|  |  |
| --- | --- |
| **Sprint:** | Niagara Q2 Sprint 2020 |
| **Epic Link:** | Automation API Enhancements |

## CHAPTER 9

### TEST CASES:

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.Test Case acts as the starting point for the test execution, and after applying a set of input values; the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

|  |  |
| --- | --- |
| ***[FAS-11550] Verify that the Action shortcut keys are working properly instead of clicking on Action button on Screens.*** | |
| **Status:** | OPEN |
| **Project:** | FAS |
| **Component/s:** | UI |
| **Affects Version/s:** | None |
| **Fix Version/s:** | None |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Test | **Priority:** | Medium |
| **Reporter:** | Rishabh Porwal | **Assignee:** | Rishabh Porwal |
| **Resolution:** | Unresolved |  | |
| **Labels:** | None | | |
| **Remaining**  **Estimate:** | Not Specified | | |
| **Time Spent:** | Not Specified | | |
| **Original**  **Estimate:** | Not Specified | | |

|  |  |
| --- | --- |
| **Expected Outcome:** | Action Shortcut Keys(ALT+KEY) should be work accordingly as action button works |
|  |
| **Last Run on**  **QA:** | 06/Mar/2020 8:05 AM |
| **Automation**  **Status:** | Not Automated |
| **Test Execution**  **Status - QA:** | Pass |
| **Test Execution**  **Status - QB:** | Pass |
|  |  |

|  |  |
| --- | --- |
| ***[FAS-78965] Verify the accounts in prior Recall Adjustment transaction and new Recall Adjustment .transaction after posting CCT.*** | |
| **Status:** | OPEN |
| **Project:** | FAS |
| **Component/s:** | Recall |
| **Affects Version/s:** | None |
| **Fix Version/s:** | None |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Test | **Priority:** | Medium |
| **Reporter:** | SHABIB MIRZA | **Assignee:** | SHABIB MIRZA |
| **Resolution:** | Unresolved |  | |
| **Labels:** | RecallAdjustment | | |
| **Remaining**  **Estimate:** | Not Specified | | |
| **Time Spent:** | Not Specified | | |
| **Original**  **Estimate:** | Not Specified | | |

|  |  |
| --- | --- |
| **Expected Outcome:** | After posting CCT, New LP is showing in new Recall Adjustment Transaction. And CCT out LP is showing in Prior Recall Adjustment Transaction(before posting CCT) . |
|  |
| **Last Run on**  **QA:** | 11/Feb/2020 3:04 AM |
| **Automation**  **Status:** | Not Automated |
| **Test Execution**  **Status - QA:** | Pass |
| **Test Execution**  **Status - QB:** | Pass |
|  |  |

|  |  |
| --- | --- |
| ***[FAS-99556] Verify that in CAS after posting Recall Adjustment transaction, the UCC and recall numbers are correct.*** | |
| **Status:** | Open |
| **Project:** | FAS |
| **Component/s:** | Recall |
| **Affects Version/s:** | None |
| **Fix Version/s:** | None |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Test | **Priority:** | Medium |
| **Reporter:** | Rishabh Porwal | **Assignee:** | Rishabh Porwal |
| **Resolution:** | Unresolved |  | |
| **Labels:** | RecallAdjustment | | |
| **Remaining**  **Estimate:** | Not Specified | | |
|  |
| **Time Spent:** | Not Specified | | |
|  |
|  |
| **Original Estimate:** | Not Specified | | |
|  |

|  |  |
| --- | --- |
|  |  |
| **Expected**  **Outcome:** | UCC and recall numbers of RW should be correctly matched with CAS Report. |
| **Automation Status:** | Not Automated |
|  |  |

## CHAPTER 10

### BUGS

62

When a tester executes the test cases, they might come across the test result which is contradictory to expected result. This variation in the test result is referred as a **Software bug**. These defects or variation are referred by different names in a different organization like **issues, problem, defect or incidents**.

##### Life Cycle of A Bug:

OPEN

In Development

Awaiting Validation

Awaiting user validation

On QA

On Staging

Approve for

release

Awaiting Release

CLOSE

FIG.9 Life Cycle of a BUG

|  |  |
| --- | --- |
| ***[FAS-88665] Clicking on CAS /GAAP is giving error if there is no bx Reporting present in transaction stack*** | |
| **Status:** | Closed |
| **Project:** | FAS |
| **Component/s:** | HYPO |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Bug | **Priority:** | Critical |
| **Reporter:** | Rishabh Porwal | **Assignee:** | Rafael Michel Martins |
| **Remaining Estimate:** | Not Specified | | |
| **Time Spent:** | Not Specified | | |
| **Original**  **Estimate:** | Not Specified | | |

**Description**

Environment : QA

Fund tested for: Fund 2 Fund applicable for: Fund2

Summary: CAS/GAAP Hypo is throwing error in case there is no BX Reporting Hypo is posted in transaction stack. Working fine for staging.

Effect/Priority: Critical

Steps:

1. Login to QA
2. Navigate to the URL
3. Click on add new tab and then click on hypothetical liquidation then CAS/GAAP.
4. Try to post CAS/GAAP without posting BX Reporting HYPO.

Expected Outcome:

CAS/GAAP should post without posting BX reporting

Actual Outcome:

CAS/GAAP is not getting posted.

|  |  |
| --- | --- |
| ***[FAS-22643] Transaction Attribute Export report is not getting downloaded for all funds.*** | |
| **Status:** | Closed |
| **Project:** | FAS |
| **Component/s:** | System Reports |

|  |  |  |  |
| --- | --- | --- | --- |
| **Type:** | Bug | **Priority:** | High |
| **Reporter:** | Rishabh Porwal | **Assignee:** | Sam Asghari |
| **Remaining Estimate:** | Not specified | | |
| **Time Spent:** | Not specified | | |

**Description**

Environment: QA Fund tested for: Fund 3

Fund applicable for: All

Summary:

Transaction Attribute Export system report is not getting downloaded for any of the funds. It is throwing error on console. Attached are the screenshot of the console error and error log file.

Effect/Priority: High as we are unable to download the reports

Steps to reproduce:

1. Login to QA
2. Navigate to any fund's main scenario.
3. In the left menu panel, Click on System reports Tab.
4. Select the Transaction Attribute export report under General Category.
5. Click on Download Excel.

Expected Outcome:

Transaction Attribute Export report should get downloaded successfully without any error.

Actual Outcome:

It is not getting download and throwing 500 error on console.

CHAPTER 11

**Hardware and Software Requirements**

* 1. **Hardware requirements**
     + For the proper working of the functionality the basic requirement would be that the Operating system should have the version of at least windows 7 or above.
     + The System should be configured with the proper test set-up in order to make the workflows behave in an efficient manner depending upon the level of requirements.

##### Software Requirements

* Back end valuation Engine is required as the Fund Accounting Engine is dependent on the software for making a proper synchronization of the data.
* SQL server for maintaining the relational schema of the tables.
* JIRA is required for carrying out testing related works.

## CHAPTER 12

### Softwares used

* 1. **Visual Studio 2013 and Visual Studio 2015:** The framework is built using this technology.
  2. **SQL Server 2014:** The database is accessed using this technology.
  3. **SSRS- SQL Server Reporting Services:** For generating reports.
  4. **SSDTBI Package for Visual Studio.** This allows you to open SSRS project in reports designer.
  5. **Jira:** Jira is a proprietary issue tracking product, developed by Atlassian. It provides bug tracking, issue tracking, and project management functions. JIRA has add-ons that support the test management process.
  6. **Zephyr:** Zephyr for JIRA Server is an add-on application that augments JIRA provides cost-effective, highly sophisticated test management capabilities right inside JIRA. Together, Zephyr for JIRA and JIRA enable developers, testers and the entire project team to be better prepared at every stage of the software lifecycle to plan, build, test and launch great software. You can create tests/test suites/test cycles/bugs/reports and so on. You could have an additional add-on, ZAPI for automation integration.
  7. **GitLab:** GitLab is a web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of GIT as well as adding its own features. Unlike Git, which is strictly a command-line tool, GitLab provides a Web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

GitLab offers both plans for private repositories and free accounts, which are usually used to host open-source software projects.

## CHAPTER 13

**Progress Report**

##### Gantt Chart

*Fig 10: Gantt Chart*

##### Tasks performed by me as an Intern in the team

**13.2.1 Test execution:** Executing the test cases formulated for a particular enhancement or a patch. Various testing techniques are used, depending on the module.

* + 1. **Logging Bugs around the functionality:** While executing the test cases, if a bug is encountered, it is logged on Jira and is moved forward to the developers to fix them.
    2. **Retesting Bugs once they get fixed**: Once the bug is fixed, the functionality is tested again to test if the bug has been fixed or not. It is then accordingly approved or a bug is logged again.
    3. **Regression Testing around the functionality:** It is done manually for all the functionalities that have been affected by the new patch or enhancement released by the developers on the QA environment. It is performed whenever a new patch or an enhancement is released. No tools are used, since it is performed manually. Previous test cases are used to perform this testing. I am required to test and report the bugs on Jira, it is an online tool used as a Project Manager. Once this is approved, the functionality is moved forward to the QB environment for further testing.

# CHAPTER 14

**Conclusion**

##### 14.1 Conclusion:

Design discussion has already been done for the remaining epics planned for the current quarter release cycle. I am also working on a tool called ‘Selenium’ which is particularly used for automation. The project further delves into automating the manual cases and thus, I will be assisting the team in the very same.

##### Limitations:

Though every task is never said to be perfect in this development field, even more improvement may be possible in this system. These limitations will be overcome in the future by integrating this system with other projects.

##### Future scope and modification:

The current system is being integrated with other projects for covering a wider scope of functionalities. New transactions are also being added to the system to cater to newly introduced business requirements.

# References

[1] [www.blackstone.com](http://www.blackstone.com/)

[2] [www.w3schools.com/sql/](http://www.w3schools.com/sql/)

[3] https://prod.teamgantt.com/

[4] https://[www.atlassian.com/software/jira](http://www.atlassian.com/software/jira)

[5] <http://www.thoughtfocus.com/>