University of Central Missouri – CIS

Anirudh Ambatipati

Advanced Systems Project

March 27, 2017

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Authors** | **Version** | **Change Reference** |
| 03/09/2017 | Sunitha Pedda | 0.1 | First Version - A Draft |
| 03/11/2017 | Sunitha Pedda | 0.2 | Second Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Participants**

**Authors**

|  |  |  |
| --- | --- | --- |
| **Name** | **Representing** | **Role** |
| Sunitha Pedda |  | Graduate Student |
|  |  |  |

**Reviewers**

|  |  |  |
| --- | --- | --- |
| **Name** | **Representing** | **Role** |
| Dr. Prasad Rudramuniyaiah |  | Professor |
|  |  |  |

Table of Contents

**Project Description**4

**Use case**5

Use Case Description5-14

Use Case Diagram14

**E-R Diagram**15

**Class diagram** 16

**Critical Process**16

**System Components and their Relationships**17

**Unit Testing**17-21

**Scenario Level Testing**21-25

**Project Description**

University of central Missouri Computer Information System program is primarily designed based on the fact that all organizations, large or small, for-profit, non-profit or charitable, multi-national rely on timely, accurate and useful information to enhance decision making. But, in the current market has a lack of people who are good at these decision making skills So, UCM has created a program that focuses on educating students how understanding the technology for an essential organizational success. UCM CIS engages students in all categories of IT industry like gathering user requirements, analyzing those requirements, designing a solution, developing software or configuring network components according to the design, testing the solution or supporting the solution.

In this project, I would like to come up with a UCM CIS website where different users of the CIS department like Director, Advisor, Student can login to the University CIS website to do some there role based functionalities and also can access the some common functionalities of the website where any user can access them if the user is registered with university. This would eradicate the use of multiple website by different users of CIS department. Instead all the users of department can access the university website by only one website. This role based mapping of website to users is handled through there login credentials provided by the University.

One of the Major Roles in the CIS department is played by Director; director has many administrative authorities over the website like creating a Student, creating an advisor, editing/adding a program to the CIS department etc. A director can create a student based on his GRE score and background criteria, once the student is added he can send a mail to the student to his/her personal mail id providing the login credentials to the student to login. All passwords are randomly generated and MD5CryptoServiceProvider encrypted. Director can add a new advisor to the system to a particular concentration.

Advisor can be added to the system by the director. The advisor can perform some functionality like fetching student information. Once he fetches the information he can check the student Pre-requisite score, GRE Score and other factors he can decide whether the student can take the course or not and he can update pre-requisite status to met or not met and enter some notes on the students status also.

Student can login to the system once got admission form the university. Director will send the student the related login credentials to the students personal mail id. Once the student is logged in he can change the password according to university password rules. Student will hav to submit

People who are new to the investing world can get financial advice online and in person. User can check how much money needs to be invested on different schemes based on risk type he/she chooses. User can schedule an appointment with Phoenix financial advisors, who can assist in investing better.

**Use Case**

This section includes use cases descriptions and use case diagram.

Use Case Description:

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-1 | | |
| **Use Case Name:** | User login in to the System | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 04/01/2017 | **Last Revision Date:** |  |
| **Actors:** | Student: submit pre enrollment quiz and Academic Code of Conduct and can shop on One Stop Shop  Director: Approve and do much of key functionality as per director role.  Advisor: Approve student enrollment and shop on the website | | |
| **Description:** | The website provides user(\students, Faculty) to login | | |
| **Trigger:** | Users login to system  Student: | | |
| **Pre Conditions:** | User is registered with the system | | |
| **Post Conditions:** | **Success:** The User is authenticated and the system displays a corresponding home page upon user logins.  **Failure:** User is unable to log in for one or more reasons. | | |
| **Normal Flow:** | 1. Enter Username and Password 2. Validate Username and Password. 3. View Home page | | |
| **Alternate Flow:** | 1. Invalid Username and Password 2. Re-enter the Username and Password 3. Normal flow continues | | |
| **Exceptions:** | * Power goes off while entering username and password * User Computer Crashes * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of Use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-2 | | |
| **Use Case Name:** | Director can add Student/Advisor as a User | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 04/01/2017 | **Last Revision Date:** |  |
| **Actors:** | New User | | |
| **Description:** | The Director can create a User can be an advisor/student to access the system. | | |
| **Trigger:** | Student: Can login from the details sent through mail.  Advisor: Can login from the details sent through mail. | | |
| **Pre Conditions:** | user is not registered with the system | | |
| **Post Conditions:** | **Success:** user enters successful information and is returned to the corresponding homepage as a user.  **Failure:** User is unable to log in for one or more reasons. | | |
| **Normal Flow:** | 1. Enter Username and Password 2. Validate Username and Password. 3. View Home page | | |
| **Alternate Flow:** | 1. Invalid Username and Password 2. Re-enter the Username and Password 3. Normal flow continues | | |
| **Exceptions:** | * Power goes off while entering username and password * User Computer Crashes * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of Use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-3 | | |
| **Use Case Name:** | Student can submit the ACC and Pre-enrollment form | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Director | | |
| **Description:** | The system provides director to add, delete and update the programs | | |
| **Trigger:** | If the Director wants he can update, delete, and new program to database. | | |
| **Pre Conditions:** | 1. Director should be able to login to his own webpage. 2. Webpage should allow him to update the program.(as Director role) | | |
| **Post Conditions:** | **Success:** Director can update any program he wants to. | | |
| **Normal Flow:** | 1. Director Logins 2. Director can navigate through his own webpage. 3. Looks through all the programs available already in the system 4. Chooses a program that he wants to update. 5. Chooses a program that he wants to delete. 6. Creates a program he wants to add to the system. 7. Once the delete, update, create is done database should be update 8. The updates done by the director are visible when he logins next 9. Navigate to home page 10. Logout | | |
| **Alternate Flow:** | 1. Go to Fetch student page 2. Go to create advisor page 3. Add a new student through add/edit student page 4. Delete any advisor/student. | | |
| **Exceptions:** | * Director cannot access the page due to access issues. * Director changes are not updated in the database due to Database connection issues * Power goes off creating/updating a program * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-4 | | |
| **Use Case Name:** | Director can create a student and sends an Email to student | | |
| **Created By:** | Anirdh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Director | | |
| **Description:** | Director can add a student to the system and send email to the new student with his login details. | | |
| **Trigger:** | User wants to sell mutual funds. | | |
| **Pre Conditions:** | 1. Director must be able to login to system. 2. Director must be able to access add/edit student page | | |
| **Post Conditions:** | **Success:** Student added to system, and has email with login details. | | |
| **Normal Flow:** | 1. Director logins 2. Go to his Homepage 3. looks for edit/delete student page 4. Enter the details of the student. 5. After he enters student details he should be able to submit. 6. Once he submits the details he should be redirected to send email page. 7. Email should be having pre texted message with login details of student. 8. Email should be sent to Student email   .   1. Submit the Student details and added to system 2. Logout | | |
| **Alternate Flow:** | 1. Go to Fetch student page 2. Go to create advisor page 3. Go to add/edit program 4. Delete any advisor/student. | | |
| **Exceptions:** | * Director cannot access the page due to access issues. * Director changes are not updated in the database due to Database connection issues * Power goes off creating/updating a program * Server down * Network Issues | | |
| **Includes:** | Email Id of the student should be available with director to send email | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-5 | | |
| **Use Case Name:** | Student Login and submits ACC | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Student | | |
| **Description:** | Student able to logins to the system with the director provided credentials and submit ACC | | |
| **Trigger:** | User student logins to system and submits Academic Code of Conduct | | |
| **Pre Conditions:** | 1. Student must receive an email with credentials to login with. 2. Student must have a ACC form to submit | | |
| **Post Conditions:** | **Success:** Student can be able to submit the ACC | | |
| **Normal Flow:** | 1. Student logins 2. Student can access his portal 3. Student can view ACC and pre-exam portal. 4. Fills the ACC form 5. After filling we can able to submit ACC 6. ACC form submission details are stored in the DB 7. Logout | | |
| **Alternate Flow:** | 1. Go to Home page 2. Go to pre-exam page | | |
| **Exceptions:** | * ACC is not available to submit * Power goes off while submitting * Server down * Network Issues | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User understands English. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-6 | | |
| **Use Case Name:** | Advisor can login to system | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Advisor | | |
| **Description:** | Advisor can login the system from preset credentials from director | | |
| **Trigger:** | Advisor login the system and review student profile | | |
| **Pre Conditions:** | 1. Advisor can login the system with login credentials | | |
| **Post Conditions:** | **Success:** Can review the submitted student details | | |
| **Normal Flow:** | 1. Advisor can login the system with login credentials provided by the Director. 2. Advisor can review the student details. 3. Advisor can update the pre requisite status of student 4. Advisor can update the details of student 5. Advisor can change the password 6. Advisor can logout | | |
| **Alternate Flow:** | 1. Can move on to password change page and notes page | | |
| **Exceptions:** | * There are no student assigned to the advisor * No students are admitted to the concentration | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Advisor credentials are provided earlier | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-7 | | |
| **Use Case Name:** | Advisor can write notes and fetch notes on student status | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | Advisor | | |
| **Description:** | Advisor can update the notes on student regarding student pre-requisite exam. | | |
| **Trigger:** | Advisor can be able to view student details | | |
| **Pre Conditions:** | 1. Student must have submitted the pre-requisite exam | | |
| **Post Conditions:** | **Success:** Advisor can submit a note on the student performance and edit notes also. | | |
| **Normal Flow:** | 1. Advisor can login the system with login credentials provided by the Director. 2. Advisor can review the student details. 3. Advisor can update the pre requisite status of student 4. Advisor can update the details of student 5. Advisor can change the password 6. Advisor can logout | | |
| **Alternate Flow:** | 1. Go to the folder where the downloaded data is stored and manually run the bulk insert script. | | |
| **Exceptions:** | * There are no student assigned to the advisor * No students are admitted to the concentration | | |
| **Includes:** |  | | |
| **Frequency of use:** | Daily | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Student has submitted the pre-requisite exam. | | |

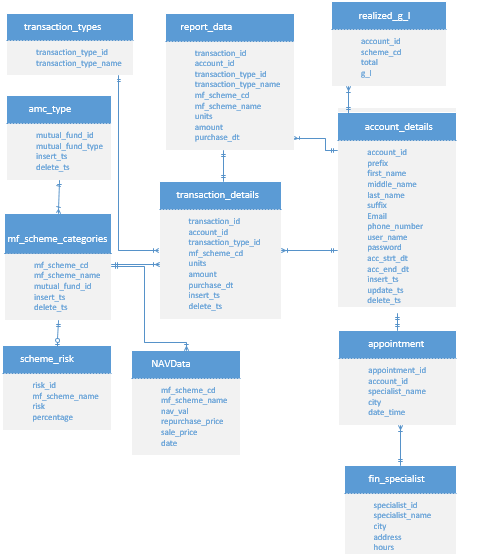
|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-8 | | |
| **Use Case Name:** | User (director, student, advisor) can login to UCM shop | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | User | | |
| **Description:** | User can login to the system to access the UCM shop website. | | |
| **Trigger:** | User can login to the webpage and view all the details on the page | | |
| **Pre Conditions:** | 1. User must be able to login to the system | | |
| **Post Conditions:** | **Success:** User can see the shop page | | |
| **Normal Flow:** | 1. User logins 2. Go to shop page 3. Check for items he needed to buy 4. Contact the seller 5. Select the item he wants to buy 6. Logout the system | | |
| **Alternate Flow:** |  | | |
| **Exceptions:** | * Not able to access the one shopping website * Network issues. | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | Some adds are already posted and login possible | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | UC-9 | | |
| **Use Case Name:** | User can login to UCM shop and post his add | | |
| **Created By:** | Anirudh Ambatipati | **Last Updated By:** | Anirudh |
| **Date Created:** | 03/07/2017 | **Last Revision Date:** |  |
| **Actors:** | User | | |
| **Description:** | User can login to the system and post a add in the UCM shop | | |
| **Trigger:** | User wants to post an advertisement in the UCM post | | |
| **Pre Conditions:** | 1. User can login to the system and access the post page | | |
| **Post Conditions:** | **Success:** User gets posts his add and adds contact details also. | | |
| **Normal Flow:** | 1. User logins 2. Go to post page 3. Add the item details 4. Submit the advertisement on the item. 5. Post details are added to the system database 6. Submit the advertisement | | |
| **Alternate Flow:** | 1. Enter shopping page | | |
| **Exceptions:** | * There are no advisors available. * Network issues. | | |
| **Includes:** |  | | |
| **Frequency of use:** | On demand | | |
| **Special Requirements:** | Performance and security | | |
| **Assumptions:** | User can login to the page | | |

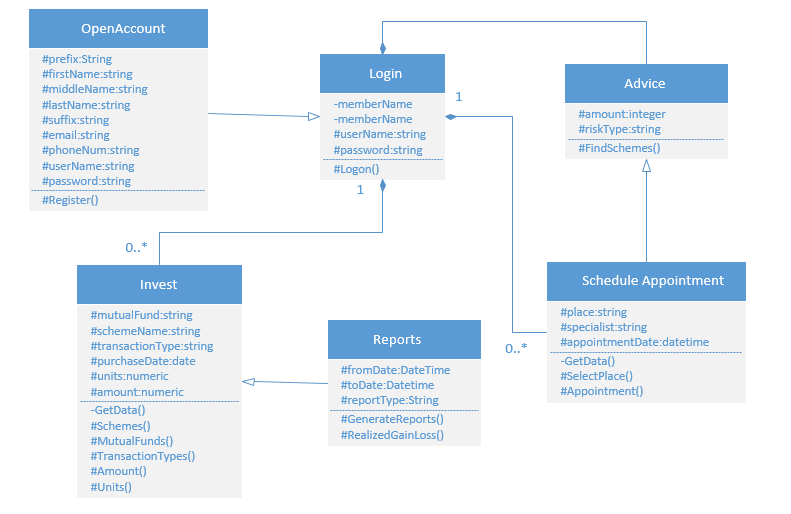
Use Case diagram

C:\Users\Anirudh raj\Downloads\Use_case.png

**Entity Relationship Diagram**

****

**Class Diagram**

****

**Critical Process**

This project has two critical processes.

1. Downloading and inserting NAV data into database: Every day NAV value gets calculated for each and every existing scheme. In order to run a website we need this data. Without this website cannot be run. As we need NAV data for investing, fetching reports and many more. Download data from Association of Mutual funds India every day without fail excluding weekends, as there will not be any business on weekends. After downloading, inserting the downloaded data into database properly is another major task. Every day, Phoenix needs to check whether the data is properly inserted or not before business starts.
2. Investing: Users interaction with the website is mostly done in this part. This is the place where user trades. So, proper functioning of this part is important before moving to any other parts like requesting reports or getting advice. Mutual funds, schemes falling under each mutual fund, transaction types, date, amount and unit textboxes must behave as expected. Calculation done for amount and units textboxes when transaction types sell and buy are chosen respectively. Once this process is done successfully, user can request for reports.

**List of System Components and their Relationship**

1. Windows task Scheduler: This is the only component in this project where user cannot directly interact with while using this website. This is used to schedule jobs for downloading data and inserting data into database.

**Unit Testing**

**Invest.aspx.cs :**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Phoenix Mutual Funds | Test Designed by: | Sunitha Pedda |
| Module Name | Invest page | Test Designed date | 03-11-2017 |
| Module Description | Test invest page and see if the record is inserted. | Test Executed by: | Sunitha Pedda |
| Version | 0.1 | Test Execution date: | 03-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Login | Username:Sunitha  Password:suni123 |  | Able to login | Successfully logged in and navigated to home page | P |
| 2. Navigate to invest page |  |  | Able to go to invest page | Successfully navigated to invest page | P |
| 3.On page load, only mutual funds data should be loaded | Mutual fund names are inserted into amc\_type table. (Axis Mutual fund, Baroda Pioneer Mutual Fund, Birla Sun Life Mutual fund) | mutualfunds procedure | User able to see all the mutual funds upon clicking the mutual fund dropdownlist | Successfully sees the mutual funds | P |
| 4.Upon selection of mutual fund, schemes dropdown list must be loaded only for that selected mutual fund | Selection of mutualfunds example, Axis Mutual Fund | schemes procedure | User able to see all the schemes falling under one mutual fund. | Successfully sees the schemes falling under one mutual fund | P |
| 5.Upon selection of scheme, transaction dropdown list must be loaded | Selection of scheme example, Axis Income saver- Growth Option | transactiontype procedure | Upon selection of scheme, user able to see transaction types | Successfully sees the transaction types | P |
|  |  |  |  |  |  |
| 6.Choose transaction date | Example: 03/09/2017 |  | User able to choose a date | Successfully chooses a date | P |
|  |  |  |  |  |  |
| 7.Upon choosing date, amount and units must be enabled to edit based on transaction type(for buy, amount textbox must be in editable mode and for sell units textbox must be in editable mode) | Example: Buy Amount – 1000  Sell: Units - 40 |  | Buy: Get the account id for inserting a transaction for user who logged in but not for others.  Get the NAV value for the selected scheme for the selected date(In financial world NAV value we consider is always previous day end of the value)  Calculate units based on amount (amount/nav) | Buy: Successfully calculated units  Sell: successfully calculated amount | P |
|  |  |  | Sell: Follow same step of getting account id and NAV value as did in buy  Calculate amount based on units (units \* nav) |  |  |
|  |  |  |  |  |  |
| 8.Upon submit button, record must be inserted in transaction details table |  |  | User is able to submit the transaction. | Successfully inserted a record into transaction\_details table | P |
|  |  |  |  |  |  |

**Advice.aspx.cs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Phoenix Mutual Funds | Test Designed by: | Sunitha Pedda |
| Module Name: | Advice page | Test Designed date | 03-11-2017 |
| Module Description: | Test Advice page and see if user is able to see the risk schemes based on amount entered. | Test Executed by: | Sunitha Pedda |
| Version: | 0.1 | Test Execution date: | 03-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Steps | Test Data | Scripts called | Expected Result | Actual Result | P/F |
| 1.Login | Username:Sunitha  Password:suni123 |  | Able to login | Successfully logged in and navigated to home page | P |
| 2.Navigate to advice page |  |  | Able to go to advice page | Successfully navigate to advice page | P |
| 3.Enter amount | Example: 10000 |  | User Should be able to provide an integer value | Successfully enters integer value | P |
| 4.select risk type | low |  | User should be able to select risk type | Successfully selects risk type | P |
| 5.Click find schemes button | Data entered in scheme\_risk with scheme names and percentage | Not scripts exactly:  1.Needs to get all the schemes and percentages for selected risk type  2. (amount \* percentage)/100 and display this amount along with scheme name. | Able to see all the schemes falling under chosen risk type showing how much user needs to invest on each | Successfully sees the schemes and how much needs to be invested on each scheme based on amount entered | P |

**ScheduleAppointment.aspx.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Phoenix Mutual Funds | Test Designed by: | Sunitha Pedda |
| Module Name | Schedule Appointment page | Test Designed date | 03-11-2017 |
| Module Description | Test schedule appointment page through advice page and see if appointment is booked. | Test Executed by: | Sunitha Pedda |
| Version | 0.1 | Test Execution date: | 03-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Steps** | **Test Data** | **Scripts called** | **Expected Result** | **Actual Result** | **P/F** |
| 1.Login | Username:Sunitha  Password:suni123 |  | Able to login | Successfully logged in and navigated to home page | P |
| 2.Navigate to advice page |  |  | Able to go to advice page | Successfully navigate to advice page | P |
| 3.Enter amount | Example: 10000 |  | User Should be able to provide an integer value | Successfully enters integer value | P |
| 4.select risk type | low |  | User should be able to select risk type | Successfully selects risk type | P |
| 5.Click find schemes button | Data entered in scheme\_risk with scheme names and percentage | Not scripts exactly:  1.Needs to get all the schemes and percentages for selected risk type  2. (amount \* percentage)/100 and display this amount along with scheme name. | Able to see all the schemes falling under chosen risk type showing how much user needs to invest on each | Successfully sees the schemes and how much needs to be invested on each scheme based on amount entered | P |
| 6.Click Schedule an appointment | 1.Provide cities for select a place dropdownlist  2.On page load, choose your specialist must be disabled as user needs to select the place before he wants to choose a financial advisor |  | Should be able to navigate to schedule an appointment page | Successfully navigates | P |
| 7.Select a place |  | Select\_specialist procedure | Should be able to select a place and able to select specialist as well | Successfully chooses a place and specialist for one-on-one advice | P |
| 8.select appointment date |  |  | Should be able to select a date | Successfully chooses a date | P |
| 9.click book appointment |  | Not script exactly:  SqlAdapter: to get account id and store it in variable  SqlCommand: to insert a record into appointment table | Should be able to book an appointment | Successfully book an appointment | P |

**Scenario Level Testing**

Includes user journey from login page, placing transactions and requesting for reports.

**Transactions**

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Phoenix Mutual Funds | Test Designed by: | Sunitha Pedda |
| Module Name | Overall Process of placing a transaction and request for reports | Test Designed date | 03-11-2017 |
| Version | 0.1 | Test Executed by: | Sunitha Pedda |
|  |  | Test Execution date: | 03-11-2017 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **(P/F)** |
| 1. | Login   |  |  | | --- | --- | | 1. | Navigate to login page | | 2. | Provide valid username | | 3. | Provide valid password | | 4. | Click logon button | | 2.User:Sunitha  3.Password: suni123 | 1. User should be able to navigate to login page.  2.User should be able to login | User is navigated to login page and successfully able to login. | P |
| 2. | Invest   |  |  | | --- | --- | | 1. | Navigate to Invest page | | 2. | Select Mutual fund | | 3. | Select Scheme | | 4. | Select transaction type   1. Buy 2. Sell | | 5. | Choose transaction date | | 6. | Amount | | 7. | units | | 8. | Submit | | |  |  | | --- | --- | | 2.Axis Mutual Fund | | | 3.Axis Income Saver - Growth option | | | 4.i Buy | 4.ii Sell | | 5.03/09/2017 | 5. 03/10/2017 | | 6. 1000 |  | |  | 7.20 | | 1. User should be able to navigate to invest page.  2. User should be able to see all the mutual funds available.  3. User should be able to see only schemes available for selected mutual fund.  4. User should be able to select transaction type  5. Able to choose a transaction date  6. For 20units, user should be able to see 336.70200 in amount textbox.  7. For 1000 buy transaction, user should be able to see 59.40 in units. | 1. User is navigated to invest page.  2. Sees all the available mutual funds.  3. Sees all the schemes fall under selected mutual funds.  4. selects the transaction type  5.chooses date successfully  6. Successfully sees the amount calculated based on provided units for sell.  7. Successfully sees the number of units calculated based on amount provided.  8. Successfully submit the transaction. | P |
| 3. | Weekly Reports   |  |  | | --- | --- | | 1 | Navigate to reports page | | 2 | Choose from date | | 3 | Choose to date | | 4 | Click submit for transaction reports | | 5 | Click realized gain loss button for realized gain loss report | | 2. 03/06/2017  3. 03/10/2017  4. Weekly | 1. User should be able to navigate to reports page.  2. User should be able to choose from date.  3. User should be able to choose a to date  4. User should be able to choose report type. | 1. User is successfully navigated to reports page.  2. User chooses from date  3. User chooses to date.  4. user chooses weekly report  5. On submit, user sees the transactions placed on 03/09 and 03/10 in step 2.  6. On clicking realized gain loss button, user sees the all the transactions with realized gain loss calculated. | P |
| 4. | Monthly reports   |  |  | | --- | --- | | 1 | Navigate to reports page | | 2 | Choose from date | | 3 | Choose to date | | 4 | Click submit for transaction reports | | 5 | Click realized gain loss button for realized gain loss report | | 2. 03/01/2017  3. 03/31/2017  4. Monthly | 1. User should be able to navigate to reports page.  2. User should be able to choose from date.  3. User should be able to choose a to date  4. User should be able to choose report type. | 1. User is successfully navigated to reports page.  2. User chooses from date  3. User chooses to date.  4. user chooses monthly report  5. On submit, user sees the transactions placed on 03/09 and 03/10 in step 2.  6. On clicking realized gain loss button, user sees the all the transactions with realized gain loss calculated. | P |
| 5. | Daily reports   |  |  | | --- | --- | | 1 | Navigate to reports page | | 2 | Choose from date | | 3 | Choose to date | | 4 | Click submit for transaction reports | | 5 | Click realized gain loss button for realized gain loss report | | 2. 03/09/2017  3. 03/09/2017  4. Daily | 1. User should be able to navigate to reports page.  2. User should be able to choose from date.  3. User should be able to choose a to date  4. User should be able to choose report type. | 1. User is successfully navigated to reports page.  2. User chooses from date  3. User chooses to date.  4. user chooses daily report  5. On submit, user sees the transactions placed on 03/09 in step 2.  6. On clicking realized gain loss button, user sees the all the transactions with realized gain loss calculated. | P |