A
Project Report
On
Stock Portfolio Project

Prepared by
Aditya Lalwani (18IT052)

Under the guidance of

Prof. Dr. Parth Shah

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology

in Information Technology

(8th Semester Software Project Major-IT447)

Submitted at

SMT. KUNDANBEN DINSHA PATEL DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology

At: Changa, Dist.: Anand – 388421

April 2022

CANDIDATE'S DECLARATION

I hereby declare that the project entitled "**Stock Portfolio Project**" is my/our own work conducted under the guidance of **Prof. Dr. Parth Shah** and **Mr. Sudhir Panchware**.

I further declare that to the best of my knowledge, the project for B. Tech does not contain any part of the work, which has been submitted for the award of any degree either in this University or in other University without proper citation.

Aditya Lalwani
(18IT052)

Prof. Dr. Parth Shah
Head & Professor,
Smt. Kundanben Dinsha Patel Department of Information Technology,
Chandubhai S Patel Institute of Technology,
Faculty of Technology & Engineering,

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled "**Stock Portfolio Project**" is a bonafied work carried out by **Mr. Aditya Sanjay Lalwani (18IT052)** under the guidance and supervision of **Prof. Dr. Parth Shah** & **Mr. Sudhir Panchware** for the subject **Software Project Major (IT447)** of 8th Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Dr. Parth Shah
Head & Professor
Smt. Kundanben Dinsha Patel Department of Information Technology

Mr. Sudhir Panchware
Functional Consultant
Enterprise Application Service
Tata Consultancy Service.

ABSTRACT

Stocks, even known as equities, represent fractional ownership in a corporation, and the stock market is a platform for investors to buy and sell such investible assets. And a portfolio is a collection of most of an investor's stock market investments in various asset classes. This project is a Portfolio Management System that is used to keep track of stock trading information. This application contains a variety of stock records for managing significant transactions such as stock purchases and sales. There are many investing portfolio applications are available on the market. These applications take effort, time, and money to set up and maintain the Portfolios. So, we have created a solution that eliminates the need to separately invest each resource in the investing portfolio that is user could able to add different resources such as Mutual funds, Debts etc. We have additionally add Income feature where user will be able to see, income generated through selling assets. With a portfolio like this, we can simply track and understand the performance of each of our investments. The major objectives of this initiative were to research potential investment possibilities, balance their risks and advantages logically, and make informed investment decisions.

ACKNOWLEDGEMENT

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many hands. I would like to extend my sincere thanks to all of them.

I am highly indebted to **Prof. Dr. Parth Shah** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

I am grateful to my external guide **Mr. Sudhir Panchware** in TCS for giving me the support and encouragement that was necessary for the completion of this project.

I would like to express my gratitude to **H.O.D. Dr. Parth Shah** and I am also grateful to all my **faculty members** of **Chandubhai S. Patel Institute of Technology** for their kind cooperation and encouragement which help me in completion of this project and preparing the report.

Last but not the least; I would also like to thank my **colleagues**, who have co – operated during the preparation of my report and without them this project has not been possible. Their ideas helped me a lot to improve my project report.

"We may not achieve everything we dream, but we cannot achieve anything unless we dream." –

Aditya Lalwani (18IT052)

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 PROJECT OVERVIEW

This project is a Stock & Portfolio Management System that is used to keep track of stock trading information. This application contains a variety of stock records for managing significant transactions such as stock purchases and sales. The major objectives of this initiative were to research potential investment possibilities, balance their risks and advantages logically, and make informed investment decisions.

Extensive study and analysis were used to gain a clear grasp of the risks and chances for significant returns presented in each investment opportunity. This knowledge will aid in the selection and implementation of smart investments in the future. This information aided in the identification and prediction of market trends, influencing future investment decisions.

1.2 OBJECTIVE

The objective of this project is for the user to gain a general understanding of securities and investments. To understand how investing in various stocks reduces risk while increasing returns. To gain an understanding of the various elements that influence investors' investment decisions. To understand how different companies manage their portfolios, such as when and where they invest. To get an understanding of the role (and functions) of portfolio management and to gain an understanding of investing, decision-making and asset allocation.

1.3 SCOPE

In Scope:

In this application we have created these functionalities' that will help user to create their portfolio simple and easy:

- Add new portfolio
- Add new stocks, its quantity, purchase price
- View Summary of all the shares in his portfolio.
- View all the different shares, total shares, current price, profit/loss and net profit/loss of the portfolio.
- Add different assets like Mutual Funds, Debts etc.

Out of Scope:

- Features for actual acquiring and selling of securities. That is, actual stock purchases and sales are made outside of Portfolio.
- Tax computations for profits/losses.
- Any market related assumption.

1.4 TOOLS & TECHNOLOGY USED

Salesforce [6]

Salesforce technology is based on cloud computing. This Salesforce CRM platform helps us to use it as a web CRM. This is customer success platform making use of all the available technology to help build streamline requirements and satisfy customer's needs.

In this platform we have use Lightning Platform, which underlies that this app has a rich feature set for configuring, creating, and customizing. We have used this technology to store, manipulate, and secure customers data, we have used trigger to automate applications requirement and have manage users and their access to this application. Through this lightning platform we have create responsive user interface.

Workbench [12]

Workbench is a powerful application that helps developers interacts with their Salesforce data. We have used this technology to Process records asynchronously via Bulk API for insert, update, upsert, delete, export data, and much more. We have used workbench to load different stocks in the Assets object.

Salesforce Apex Trigger [1]

Apex is a programming language similar to Java, used to develop and enhance the Salesforce CRM application. We have used this feature to access data from the CRM's database and manipulate it as well by using an API (application user interface). We have also used it to customize and personalize the application as per the requisites of the requirements. We have used trigger to execute flow and transaction control statements on the Lightning platform server in conjunction with calls to the Lightning PlatformAPI.

We have performed custom actions before or after events to records in Salesforce, such as insertions, updates, or deletions. Just like database systems support triggers, Apex provides trigger support for managing records.

2. PROJECT MANAGEMENT

2.1 PROJECT PLANNING

2.1.1 Project Development Approach and Justification

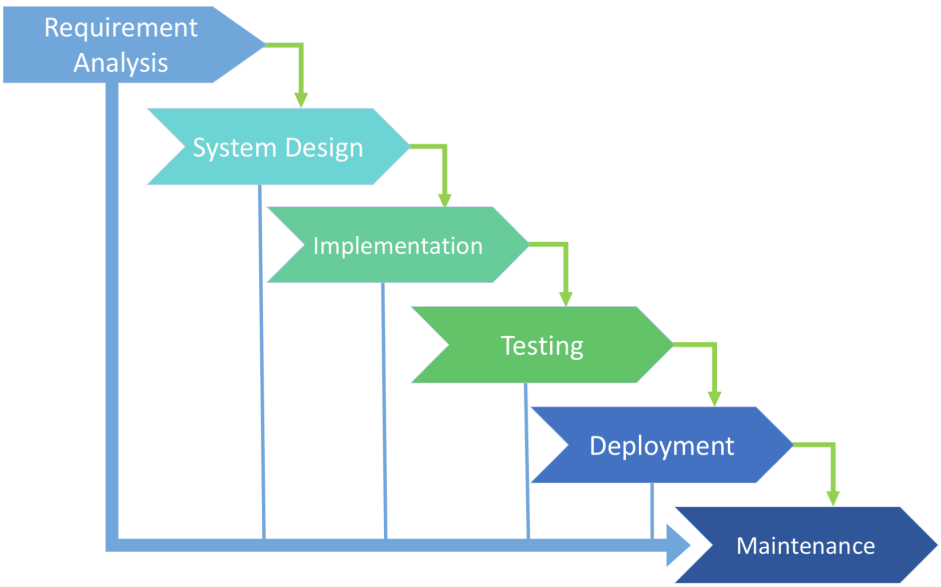

Fig. 2.1 Waterfall SDLC Model Approach Is Been Used In This Project [8]

We have used Waterfall SDLC model (Fig. 2.1), in which development process looks like the flow, moving step by step through the phases of analysis, projecting, realization, testing, implementation, and support. Our requirements for this project were clear and fixed that may not change. So there are no ambiguous requirements (no confusion) and it is good to use this model when the technology is well understood. This Project is of mid-sized and Risk is zero or minimum.

Due to this model management is simple as every phase has a defined result and process review. It is easy to determine the key points in the development cycle and to classify and prioritize tasks.

2.2 PROJECT WORK SCHEDULING

Fig. 2.2 describes that how week we have connect with mentors to set up assignments, tasks, and subtasks for the project and completed each phase. Researching, Designing, implementing, and testing are the phases used while developing the project.

ID	Task name	Start	Finish	Duration	Complete
1	Research + Discovery	2022-02-28	2022-03-14	11.0 d.	100.0%
2	Define Project Scope	2022-02-28	2022-03-03	4.0 d.	100.0%
3	Research Review/User Research	2022-03-03	2022-03-07	3.0 d.	100.0%
4	Requirements Gathering	2022-03-08	2022-03-14	5.0 d.	100.0%
5	Design Phase	2022-03-14	2022-03-25	10.0 d.	80.0%
6	Design Revisions (if needed)	2022-03-14	2022-03-18	5.0 d.	60.0%
7	Project Structure Implementation	2022-03-21	2022-03-25	5.0 d.	100.0%
8	Development Phase	2022-03-28	2022-04-14	14.0 d.	93.3%
9	Connecting Different Objects	2022-03-28	2022-04-01	5.0 d.	100.0%
10	Apex Triggers	2022-04-04	2022-04-11	6.0 d.	100.0%
11	Validations	2022-04-11	2022-04-14	4.0 d.	75.0%
12	Testing + Revision Phase	2022-04-13	2022-04-15	3.0 d.	100.0%

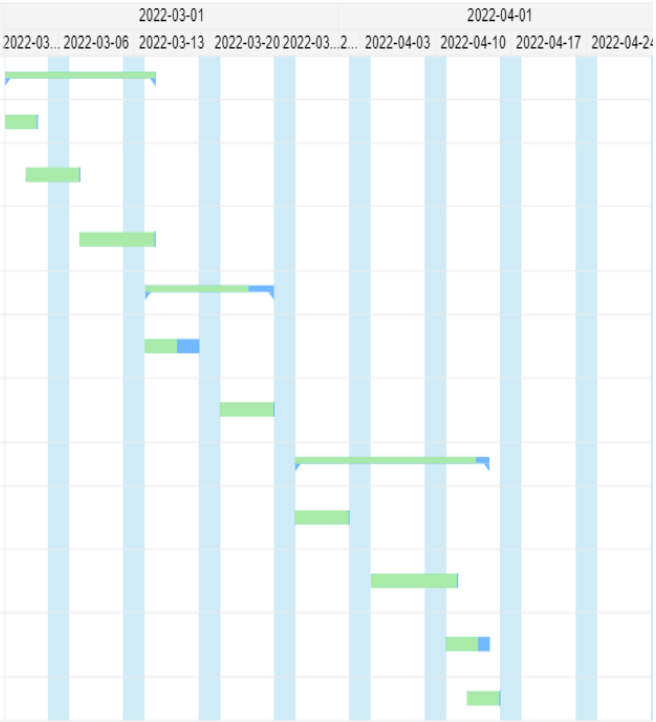

Fig. 2.2 Gantt Chart Showing Work Done Per Week.

3. REQUIREMENT SUMMARY

This investment Portfolio Helps us to understand which investments work best under which market situation and how to distribute resources into different asset classes.

3.1 TECHNICAL REQUIREMENTS [10]

- At least 5 GB of RAM, with 2 GB available for Salesforce browser tabs.
- An Octane 2.0 score of 20,000 or greater.
- Network latency of 200 ms or less.
- Download speed of 1 Mbps or greater.

3.2 APPLICATION REQUIREMENT

- User should be able view Summary of all the shared in his portfolio. It should show all the different shares, total shares, current price, profit/loss and net profit/loss of the portfolio.
- User should be able to add new shares, its quantity, and purchase and/or sell price to the portfolio.
- User should be able to Buy and Sell Shares.
- He/She can able to see Profit/Loss Gained by Selling each share
- Additionally, user should also be able to see other assets like Mutual funds, Debts, etc.

4. REQUIREMENT DETAILS

4.1 ADD NEW PORTFOLIO

User can Create or Update many portfolio according to his need. Adding a new portfolio means that user can assign different members of whom he is managing the portfolio.

This Portfolio shows many details such as it belongs to which user, total investment done by the user, current value of all the purchased shares, Net gained profits/losses from these shares and total profit/loss gained by selling the Shares.

4.2 ADD/UPDATE STOCKS

Stocks are updated on daily basics via workbench. User could able to see different Stock lists with their type such as equity, book entry, government securities and many more. Besides that user can see the current price of the Shares and High, Low, Open, Close and Last Trading Price of all the Shares of the Day.

As these Stocks varies the profit gained in the Portfolio also changes via triggers in Salesforce.

4.3 BUY/SELL STOCKS TO ASSET TRANSACTION

User can Buy new shares and add details like Name, number of shares, purchase/ Sold, transaction date, transaction price. To buy any Stock user has to Select a Stock from the Asset and also select a portfolio in which he/she want to add the asset. User also has to mention the Quantity he/she wants to buy.

After Buying the stock new asset transaction will be created in which the purchased amount, total Expanse and average transaction price will be calculated and new record in Owned Asset will automatically create for successful transaction.

To Sell any Stock User has to confirm that the Stock must be in Owned Asset object and its quantity must me less than or equal to owned assets quantity.

4.4 OWNED ASSETS

Purchased Share will be shown here with Net Profit of each share. When any Purchase Transaction is pushed in asset transaction object that purchased stock will automatically be entered in Owned Asset object. This object will now fetch current price of the asset and then calculate the profit according to the purchased price.

4.5 SOLD ASSETS

Sold Shares will be shown here with total income gained of each share. When any Selling Transaction is pushed in asset transaction object that Sold stock will automatically be entered in Sold Asset object.

4.6 USER REQUIREMENTS

This Fig. 4.1 identifies the interactions between the system and its actors. and what requirement need to be fulfilled. This defines scope of the project, the use cases and actors in use-case diagrams describe what the system does and how the actors use it.

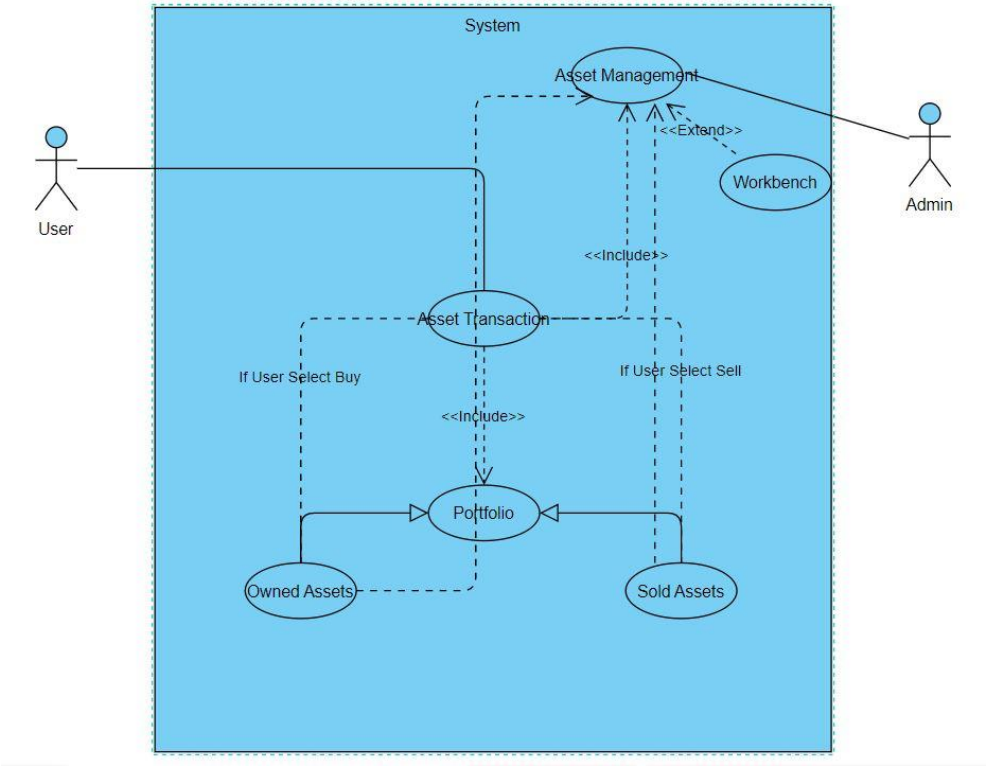

Fig. 4.1 Use Case Diagram [11]

5. SYSTEM ANALYSIS

5.1 STUDY OF EXISTING SOLUTION

There are many investing portfolio applications are available on the market. These applications take effort, time, and money to set up and maintain the Portfolios. So, having a solution that eliminates the need to separately invest in each resource in the investment portfolio while providing the same features is something that anyone would want. Portfolio currently doesn't offer Mutual funds, Debts etc. alongside in the same Portfolio. And after Selling the Stocks, users cannot able to calculate Net Profit gained from that. And to accomplish this, we have created a Salesforce web application that provides the same functionality while providing a better user experience. We have make use of Workbench to Process records asynchronously via Bulk API for insert, update, upsert, delete, export data, and much more. It can process data up to 5 million records.

While there are many software available on the market, but most of them require hardware/Software requirements. With the help of Salesforce, we can create simple browser based application. We can View and validate the generated output using the Schema Builder.

5.2 FUNCTIONAL REQUIREMENTS

For Customers:

- User has to add which Stocks he/she has purchased on daily basis.
- User also has to add which Stock he has Sold.

For Admin:

- Needs to Update Asset object that will be the price of each Share on daily basis.

5.3 NON FUNCTIONAL REQUIREMENTS

Availability:

- Application should be available 24 hours a day.

Security Requirements:

- Users' accessibility is censured in all ways.
- Salesforce App security.

5.4 CORRELATION OF ENTITY

Below Fig. 5.1 illustrate the relationships and source code dependencies among different objects/entities in the Unified Modeling Language. This also shows various relationships with other classes.

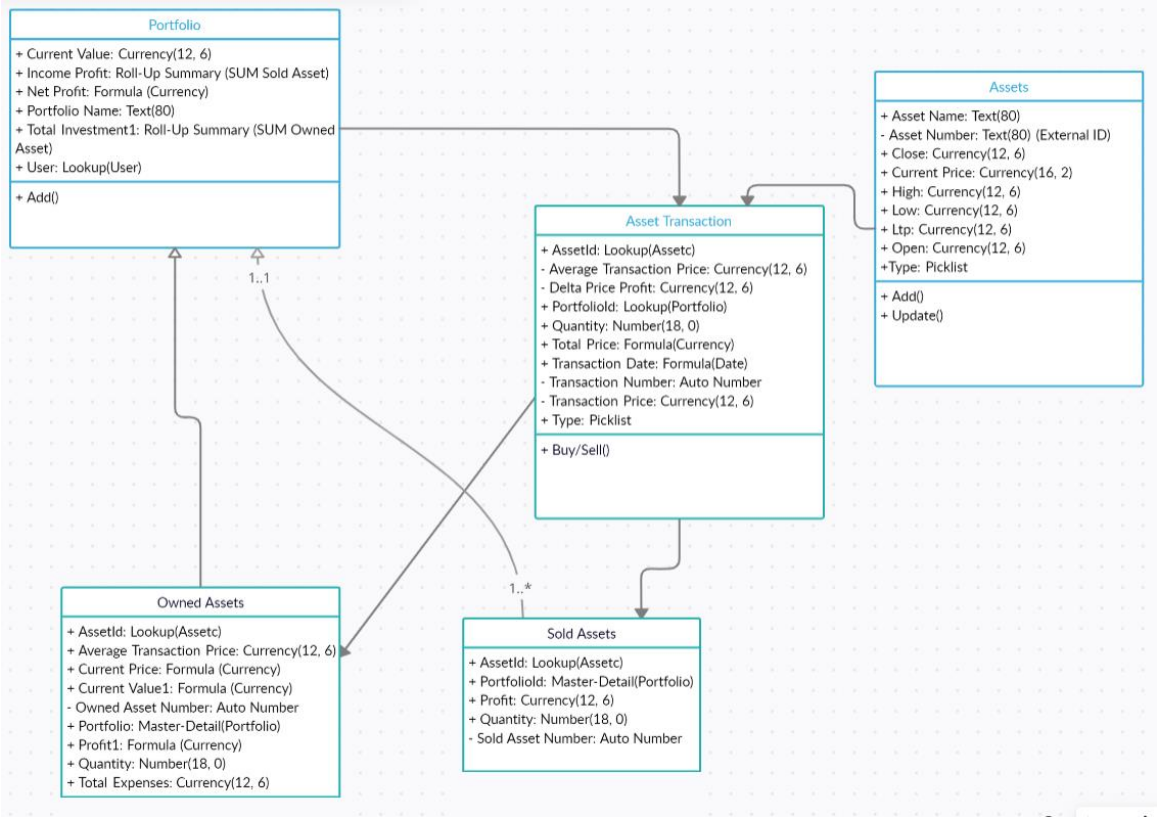

Fig. 5.1 Class Diagram [11]

5.5 FLOW OF THE PROJECT

This Fig. 5.2 represents series of actions or flow of control in the project that how user will interact with the application, that is how user will buy/sell stocks in the portfolio to calculate his investment and profit.

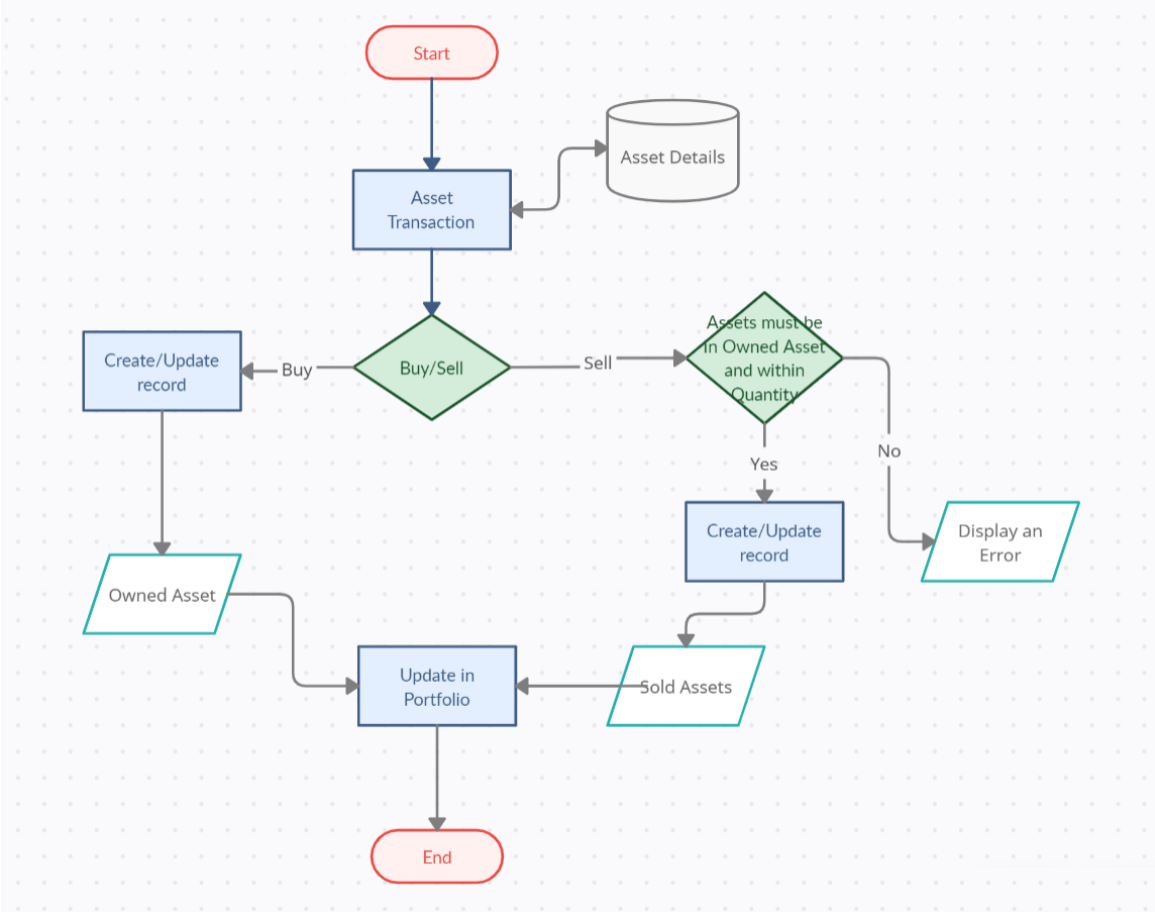

Fig. 5.2 Activity Diagram [11]

5.6 USER INTERACTION WITH OBJECT

This fig. 5.3 shows the interaction between the user and the system and how system responds. In the figure each object has its lifeline and the messages they exchange overtime, for example if user will select any assets he need to find the asset in asset object.

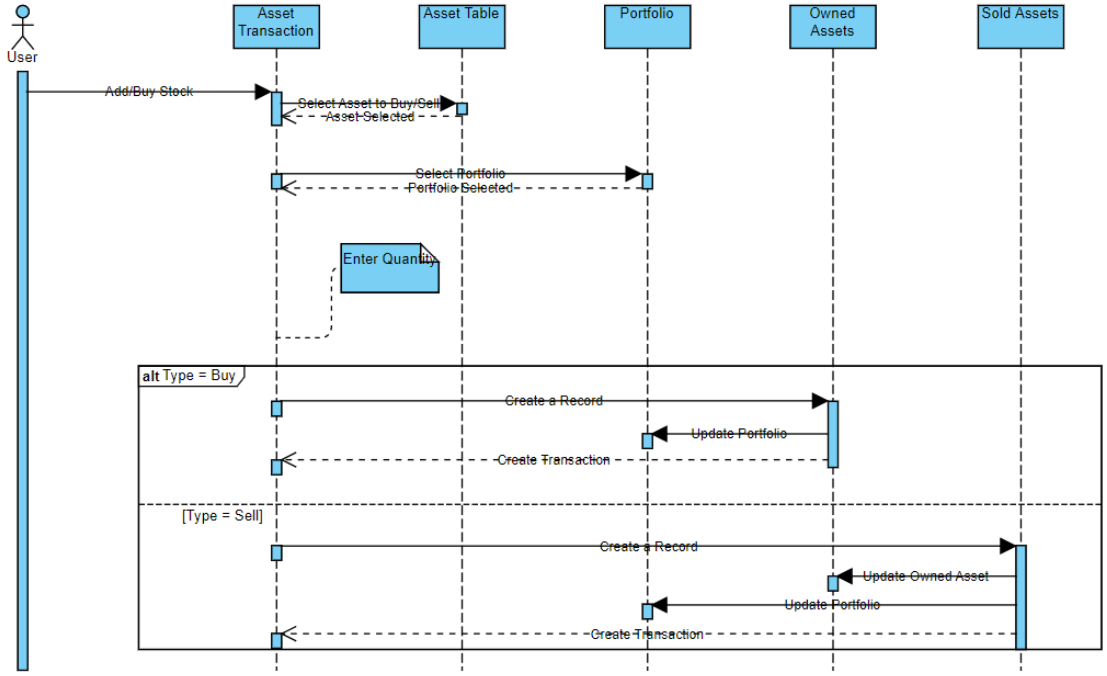

Fig. 5.3 Sequence Diagram [11]

6. DATA MODEL

6.1 PSEUDO CODE

Before Starting Implementation our project in Salesforce we have tried many possibilities that occurs while maintain an investment portfolio, through trial and error we have acquire which object would be needed and what are the fields required for maintaining the objects objective.

From the Fig. 6.1 below every week we got doubts and errors and to solve them we create real time scenario that is what happens when we change something which field should get triggered and which records should get created. From this we have also created formula to in use inside our project to calculate profit/loss.

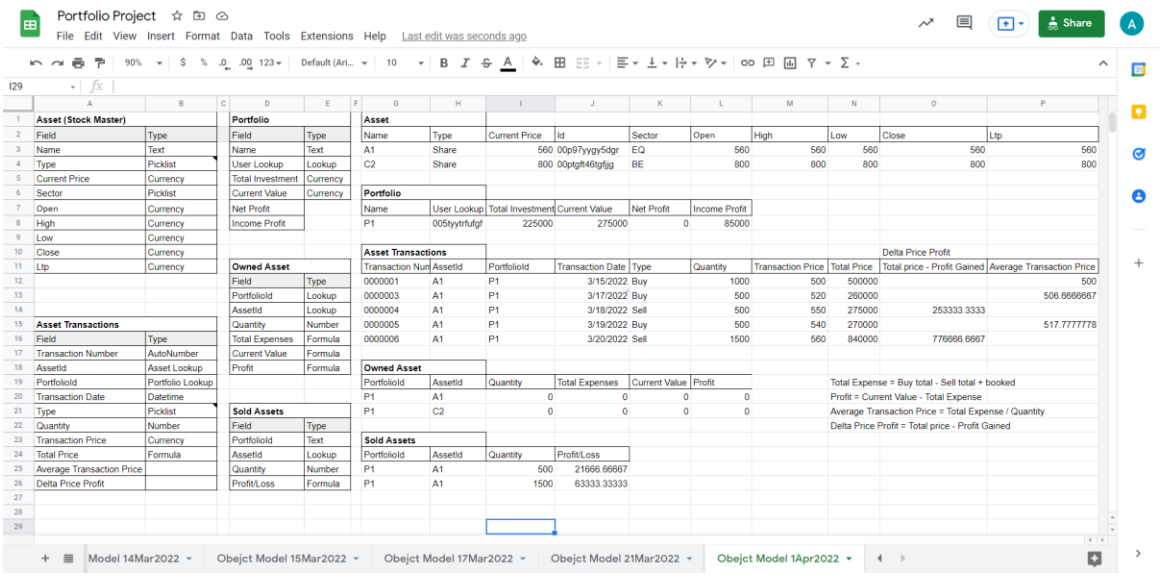

Fig. 6.1 Excel Sheet Showing Pseudo Code [4]

6.2 LIST OF OBJECTS AND ITS PURPOSE

Portfolio:

This Objects purpose to show Overview of the portfolio that is the Total Expanse done by the user on this portfolio, Profit gained from this Expanse, sum of current value of all the shares and the Income gained by selling these shares.

Asset:

This Object acts as a database that stores all the values of the Share. This Object displays each stocks type or it belongs to which series, Current Price, Highest Trading price of the day, Lowest Trading Price of the day, at which Price it has open, at which price Stock Market closes and the last trading price of the day.

Asset Transaction:

This object is the considered to be the Center of the Project because through this object User will be able to Buy and Sell assets. This object contains various fields such as which assets user needs to buys, in which portfolio, how much quantity and total price user has paid for this stock same goes while selling any stock.

Owned Assets:

This Object purpose is to show all the stocks that have been bought by the user and under which portfolio he has bought. This object shows the profits/losses of individual stock and their current value.

Sold Assets:

This Object shows the net profit/loss gained from each asset while selling them.

Below Fig. 6.2 shows the relationships between different objects that how they are dependent on each other and shows all the fields of each Object their name, type, and private fields.

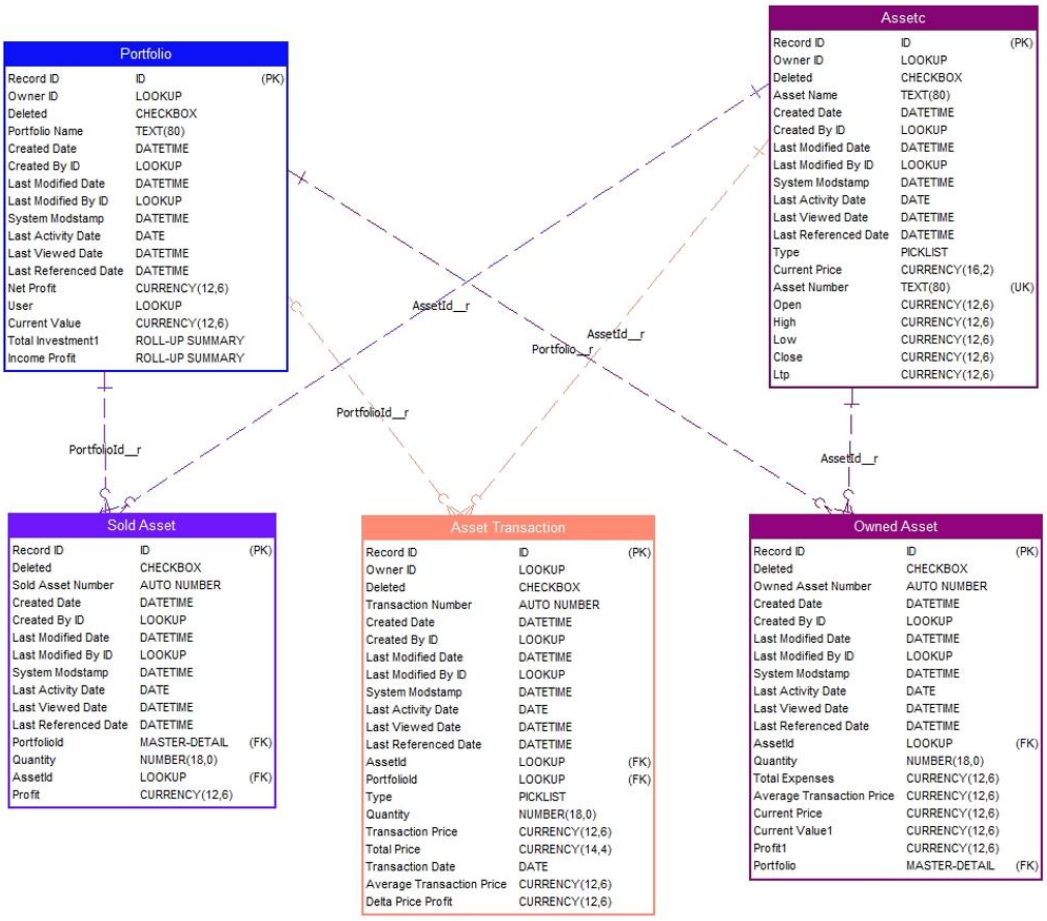

Fig. 6.2 E-R Diagram [11]

7. IMPLEMENTATION DETAILS

7.1 SALESFORCE FEATURES [6]

Flows:

A flow in Salesforce is used to automate difficult and complex project requirements. To put it another way, it takes data from the user and then triggers something with it. In this project we have used flows to set the Current Price value from the Assets object to the transaction price. To create a flow we have used the functionality named Flow Builder.

Trigger:

A trigger is an Apex script that executes before or after data manipulation language (DML) events occur. Apex triggers allow us to conduct custom actions before or after Salesforce events like insertions, updates, and removals. Similarly in this project we have created triggers on many objects some of them would be Asset transaction, Assets. In Asset Transaction object trigger is used to insert or Update the records which are affected by creating the transaction. Similarly to update the value of Current Price in every object one Trigger is created on the Assets object.

Objects:

Custom objects are objects that we have construct to store data that is unique in our project. These Objects are not only containers for our data, but they also provide unique capabilities such as when you create a custom object, the platform produces things like the page layout for the user interface automatically. Similarly various different object were created for this project that are Asset (Stock Master), Portfolio, Asset Transactions, Owned Asset and Sold Assets.

Tabs:

Custom tabs in Salesforce allow us to display custom object data, add new records and their fields. In Salesforce when we add a custom tab to an app in Lightning Experience, it displays in the app's navigation bar and the App Launcher as a menu item, where we have added these tabs. We have created tabs for all custom objects and arrange them in a single Application in Salesforce.

Workbench:

Workbench is a web-based set of tools for administrators and developers to interface with Salesforce.com enterprises using the Force.com APIs. It gives admins access to define, query, alter, and migrate both the data and the metadata in the project. We have used workbench to Process records asynchronously via Bulk API for insert, update, upsert, delete, and export data.

App:

An app is a collection of objects that work together to accomplish a certain task. In this app we don’t have to waste time and mental energy remembering to refresh our screen as you create tasks, log calls, and take notes, because the app fetches data directly from the Lightning Platform database. We have created an App named Stock Portfolio Manager where we have added the useful objects in the project.

7.2 CODING STANDARDS [2]

Coding standards used in this project for the specifications are as follows:

- Name your files logically according to the job that they perform.
- Clean code is a self-commenting (using the right variable and function names).
- Use comments and only to explain complex functions.
- Reusing code in several places, should be replaced with a single method and reduce the clutter in the code base.
- Do not put DML in loops, DML means any insert, update, delete, undelete, merge, convertLead transaction.
- Only use one trigger per object, having multiple triggers across the same object greatly increases the complexity of your Salesforce org.
- Keep logic outside of triggers, Triggers should only contain a handful of lines of code, used for calling methods.

7.3 TESTING METHODS

Manual:

We basically first have tested the application manually thus we have not used any other type of test in the application. We became the end user and tried to test the application in different ways and possibilities to enter through. We also tried parts to check out if any bugs are present and what result occurs if present. For testing all the services used in the trigger we used a functionality called apex testing.

Apex Testing:

Later we have developed the Apex test code to execute tests for our Apex triggers on the project. This ensures high quality for our Apex code and let us meets requirement for deploying Apex. There should be at least 75% of Apex code must be covered by tests, and all those tests must pass. In addition, each trigger must have some coverage.

8. SAMPLE DATA AND SCREENS

In Fig. 8.1 we can see the portfolio created by the user with total investment, current value, net profit and income profit of the portfolio.

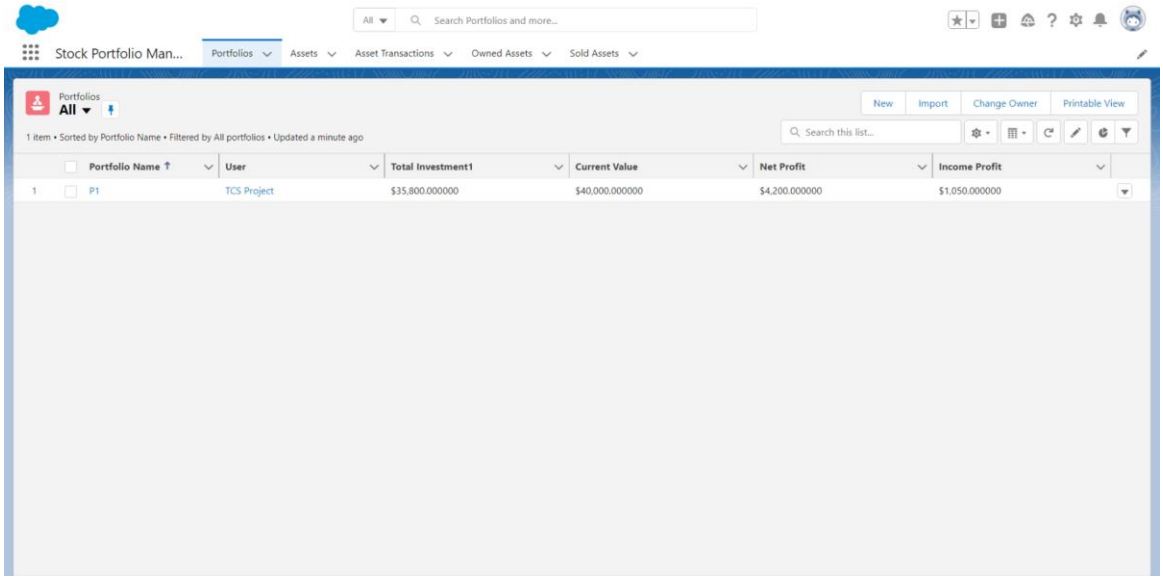

Fig. 8.1 Different Portfolios in Portfolio Object

Below Fig 8.2 show the Asset object with its field values that is name, series, current price, High, Low, Open, Close and Last Trading Price.

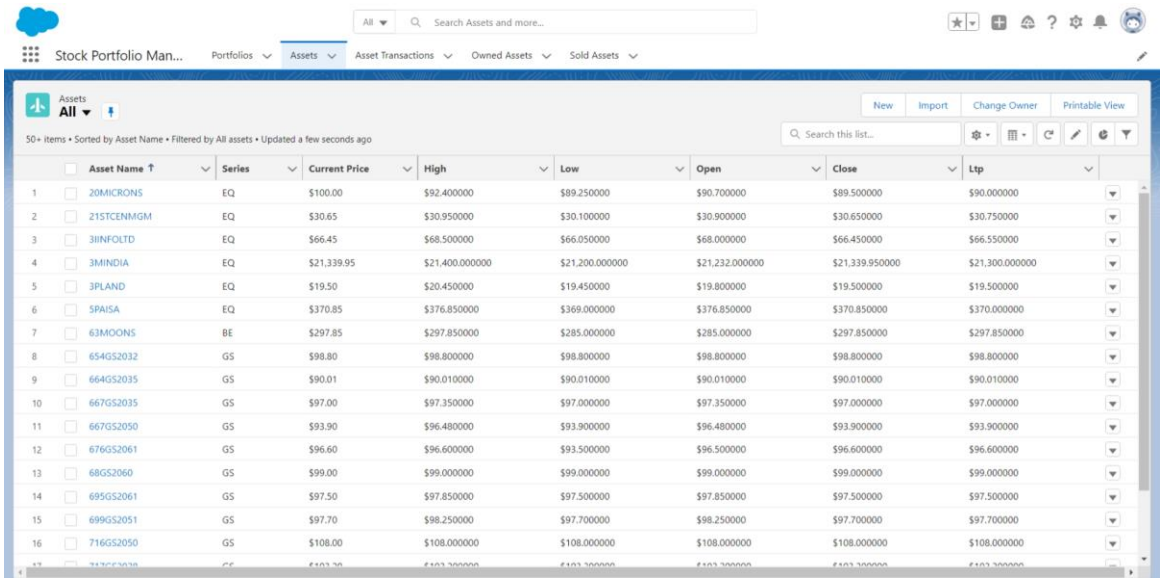

Fig. 8.2 All Assets with Name, Current Price, Series, High, Low, Open, Close and Last Trading Price

Below Fig 8.3 show the Asset Transactions object in which every transaction done by the user i.e. buy or sell will get displayed here.

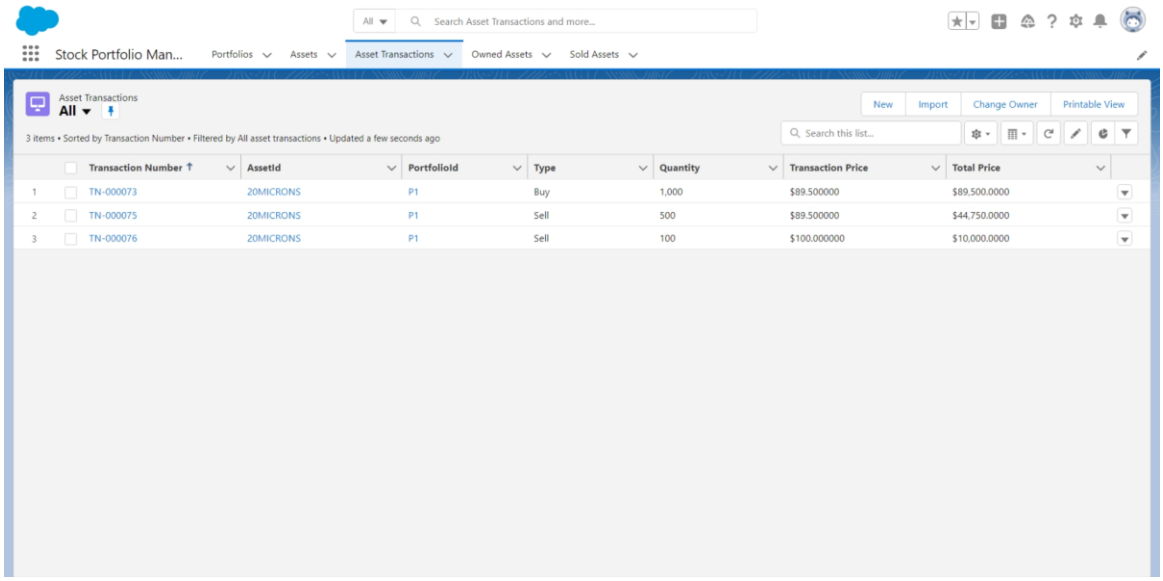

Fig. 8.3 Showing Transactions Made To Buy/Sell Shares

Below Fig 8.4 show the Owned Asset object which displays assets owned by the portfolio/user. It displays the asset, quantity, total expanse, current value and the profit/loss gained.

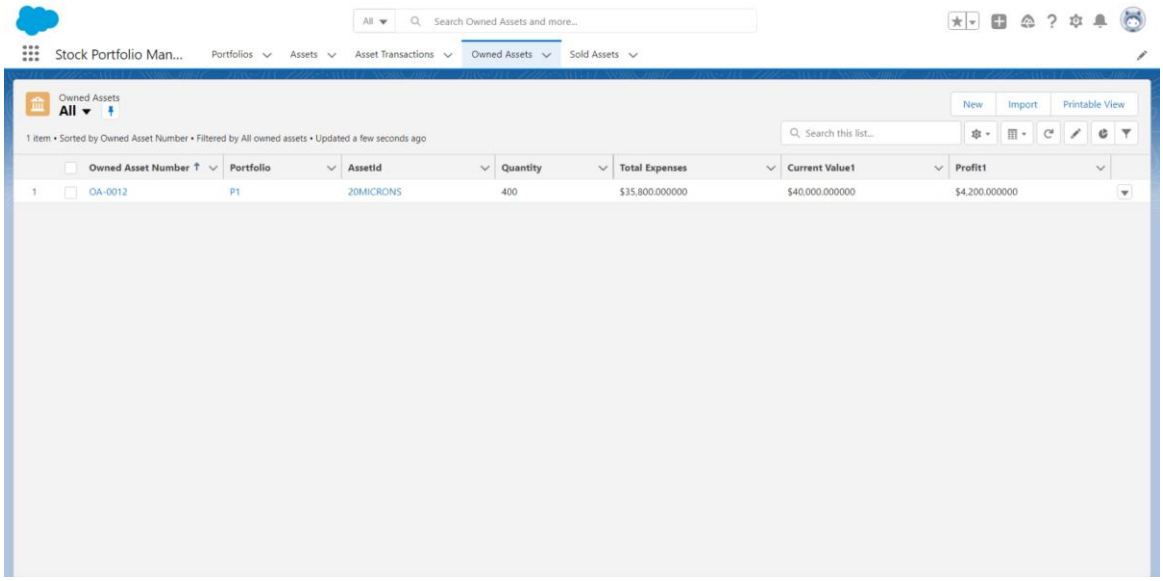

Fig. 8.4 All Assets Owned By the User

Below Fig 8.5 show the Sold Asset object which displays assets sold by the portfolio/user. It displays the asset, quantity, and the profit/loss gained from selling it.

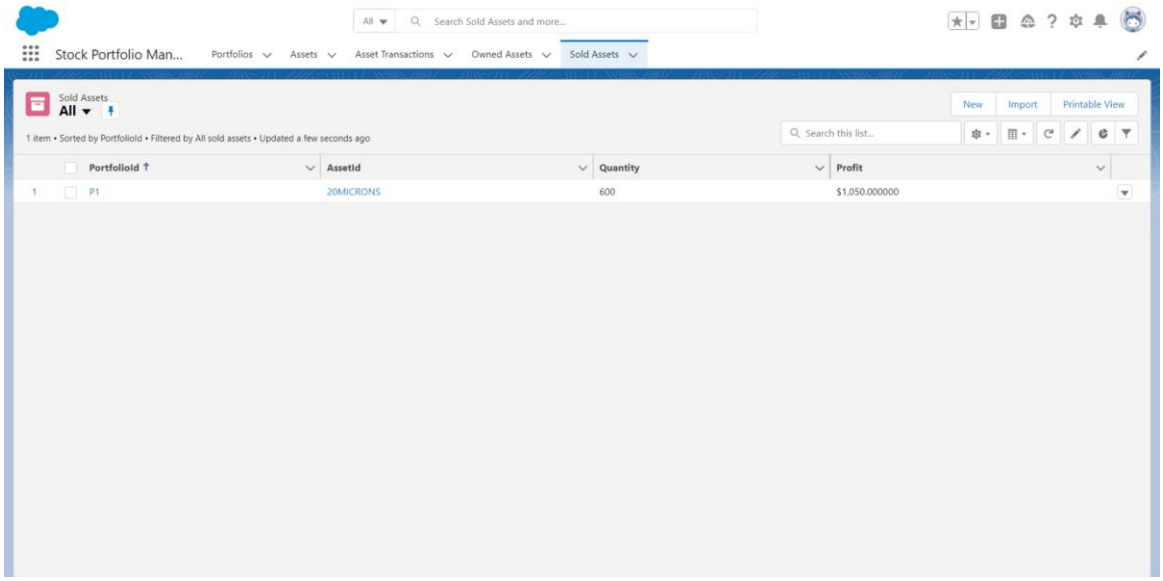

Fig. 8.5 Sold Assets with Profits

Below Fig 8.6 show the form used to create a new portfolio.

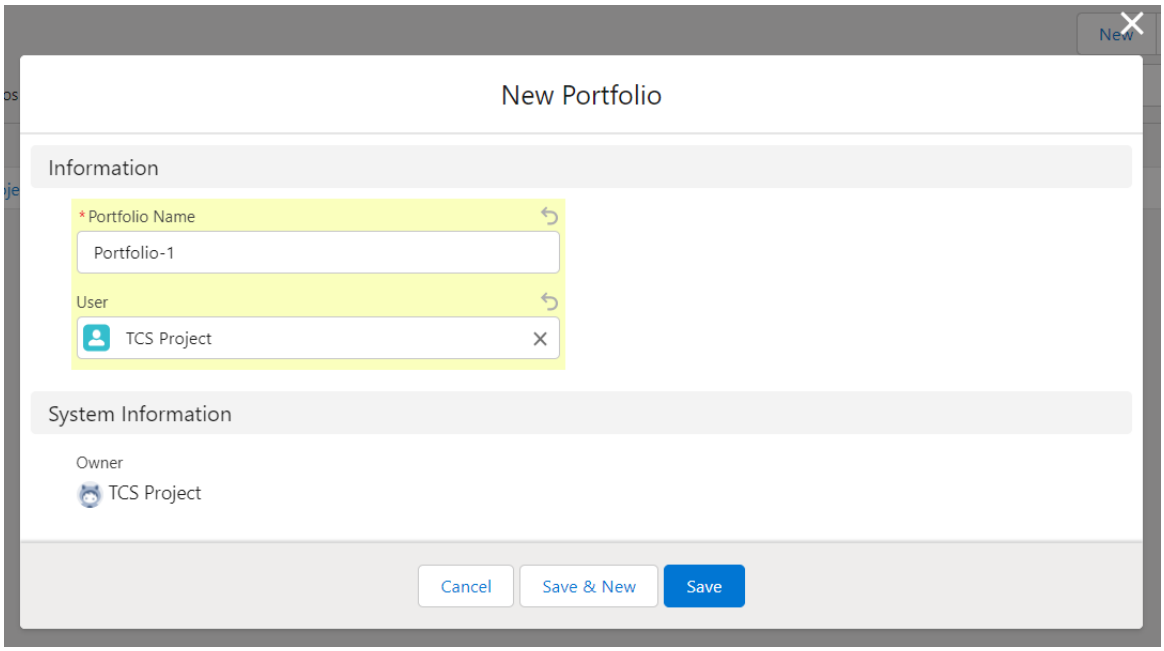

Fig. 8.6 Form to Add a New Portfolio in the Portfolios Object

Below Fig 8.7 show the form used to create a new Asset.

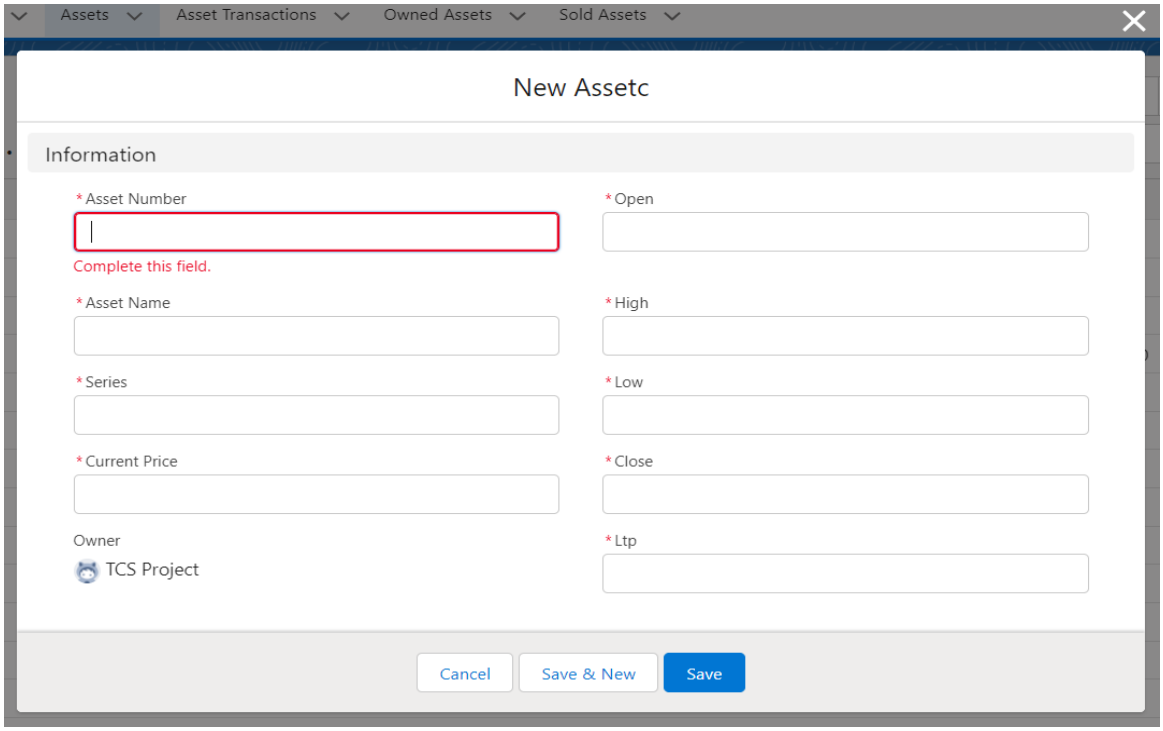

Fig. 8.7 Form to Add a New Asset in the Assets Object

Below Fig 8.8 show the form used to create a new asset transaction with type buy/sell.

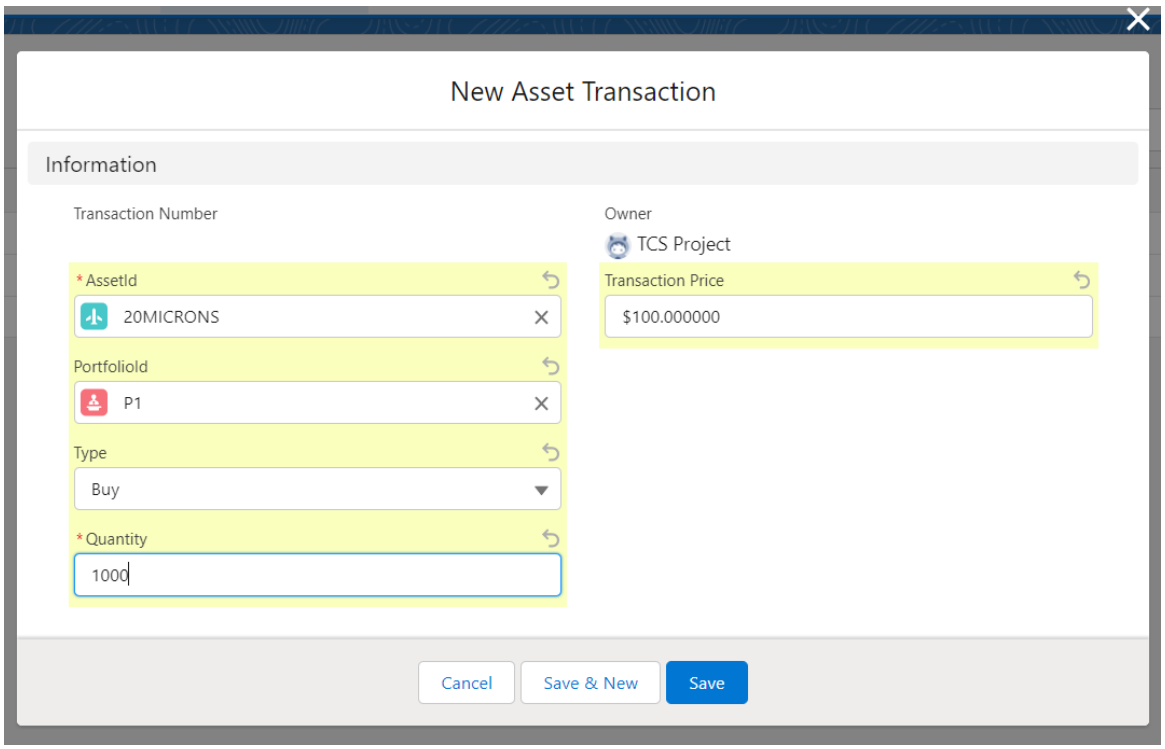

Fig. 8.8 Form to Buy/Sell Any Asset from the Asset Transactions Object

Below Fig 8.9 show the validation error, that user hasn't bought this asset.

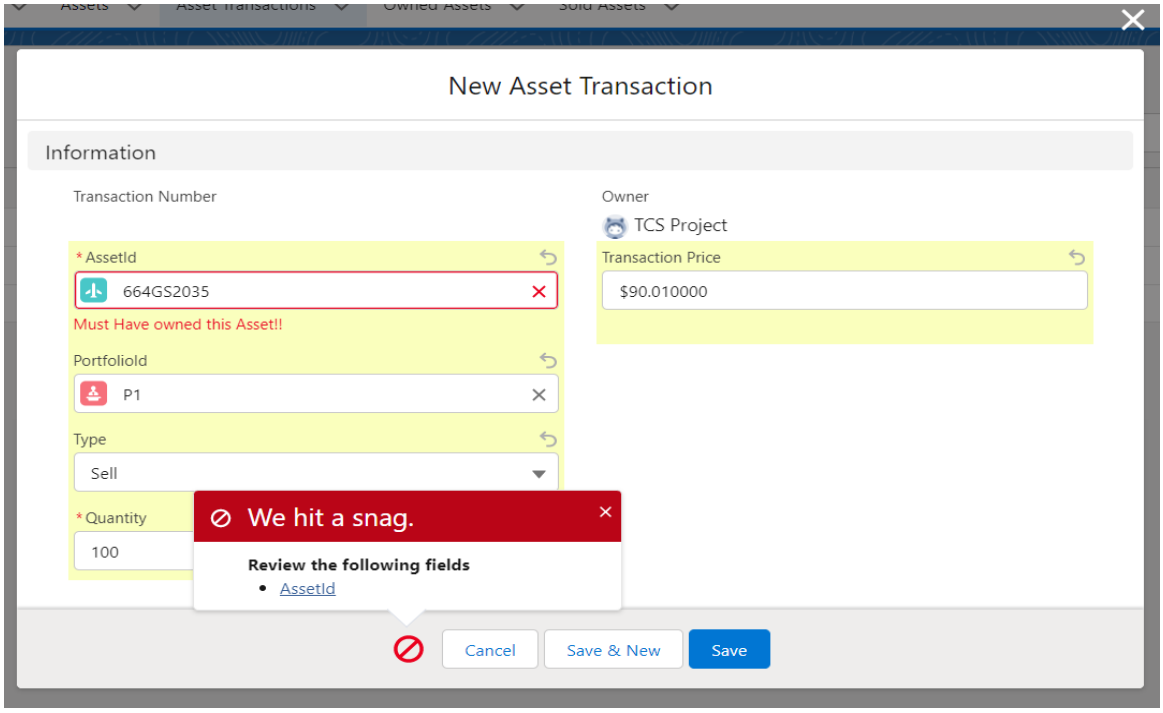

Fig. 8.9 Form Validation When Selling an Asset Which Is Not Owned By the User

Below Fig 8.10 show the validation error, that quantity owned is less than 1000 so user cannot able to sell it.

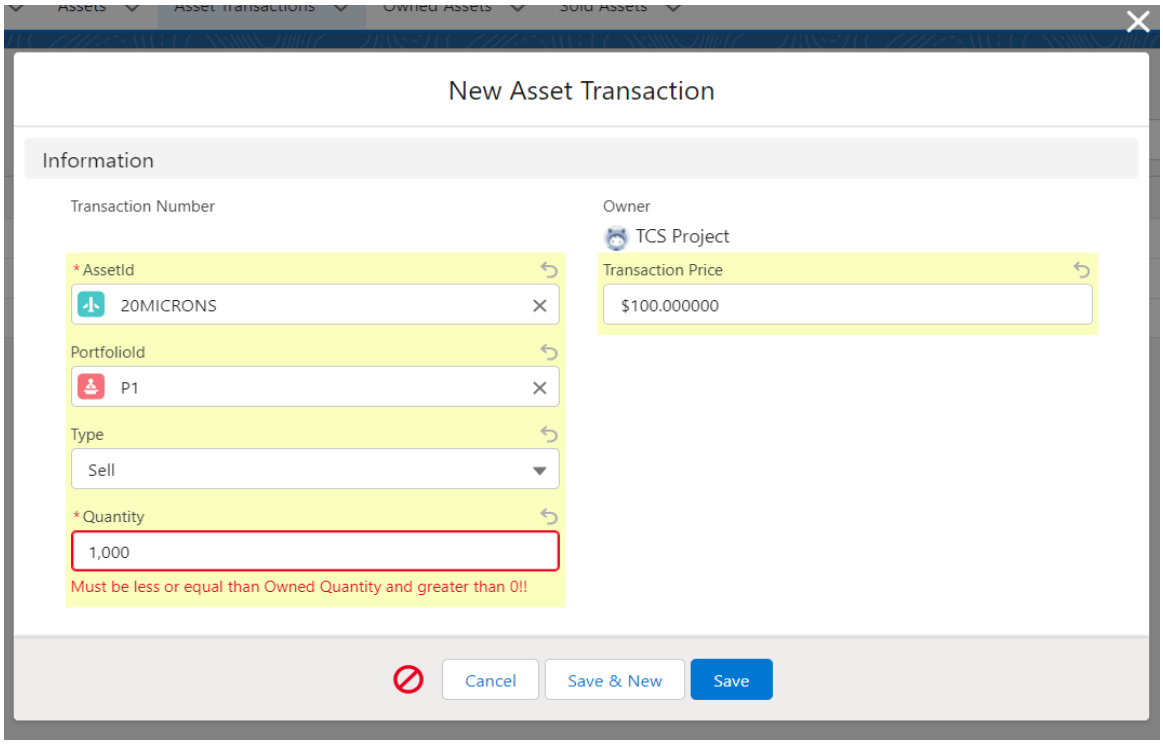

Fig. 8.10 Form Validation When Selling an Asset Where Quantity Is Greater Than Owned

Below Fig 8.11 show the related fields that are affected by this Asset.

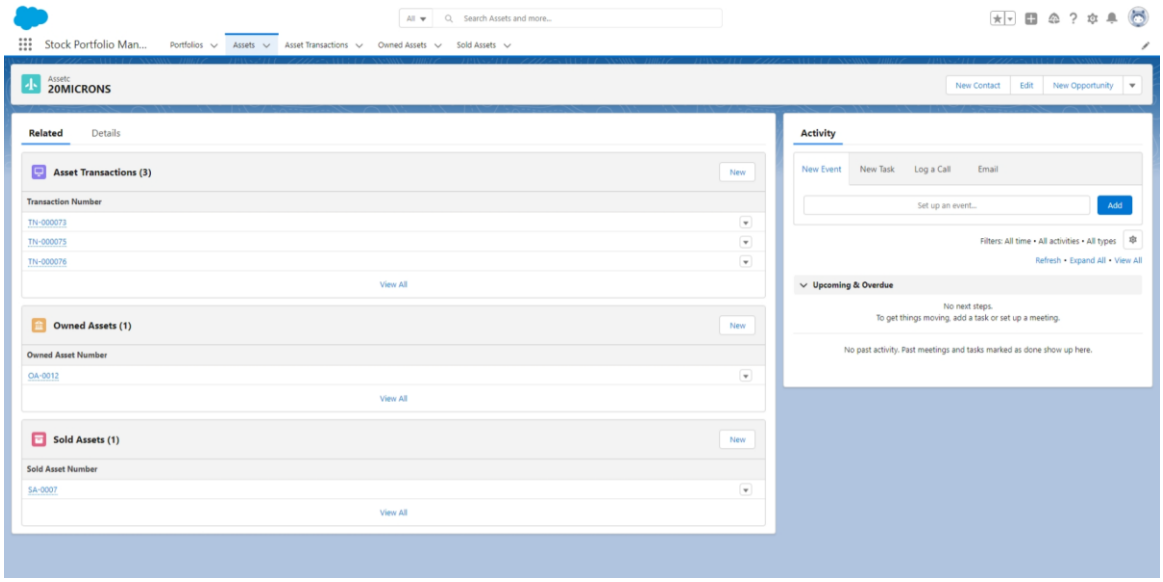

Fig. 8.11 Showing Related Records of Asset That Is Where This Asset Is Been Used

Below Fig 8.12 show the details column of this Asset with Name, Current Price, Series, High, Low, Open, Close and Last Trading Price.

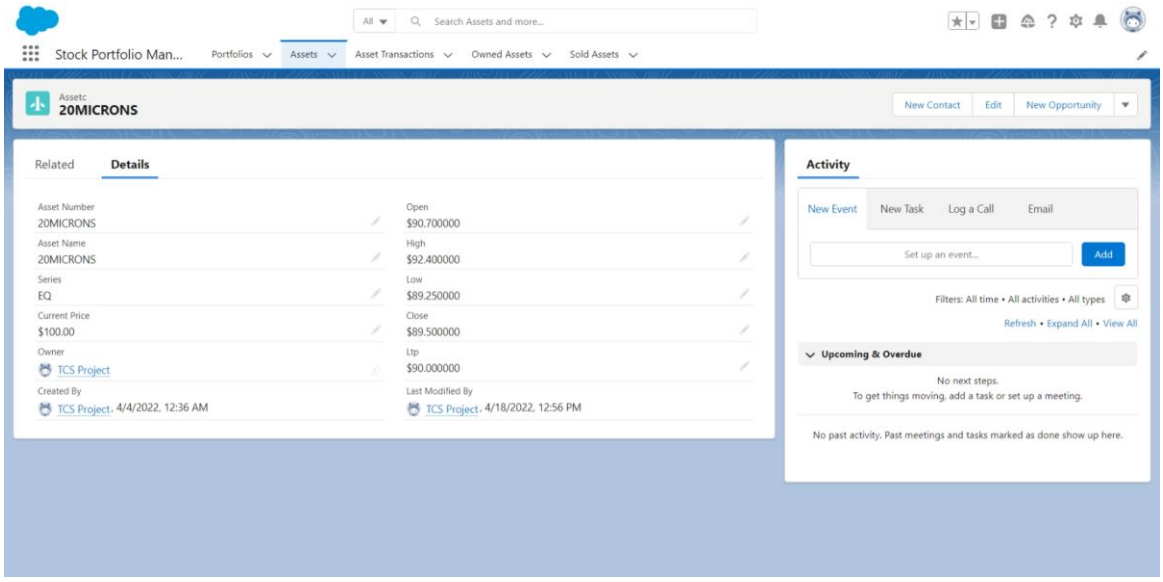

Fig. 8.12 Detail View of Single Asset

Below Fig 8.13 show the related fields that are affected by this portfolio ‘P1’.

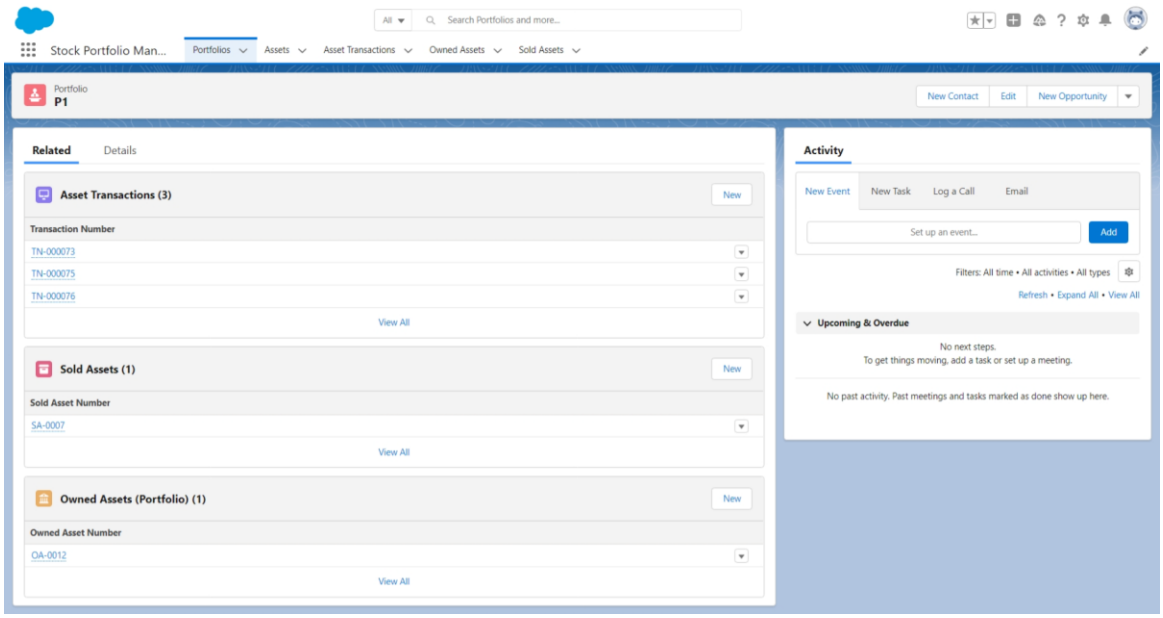

Fig. 8.13 Showing Related Records of Portfolio That Is Where This Portfolio Is Been Used

Below Fig 8.14 show the details column of this portfolio object with Name, total investment, current price, net profit and income gained.

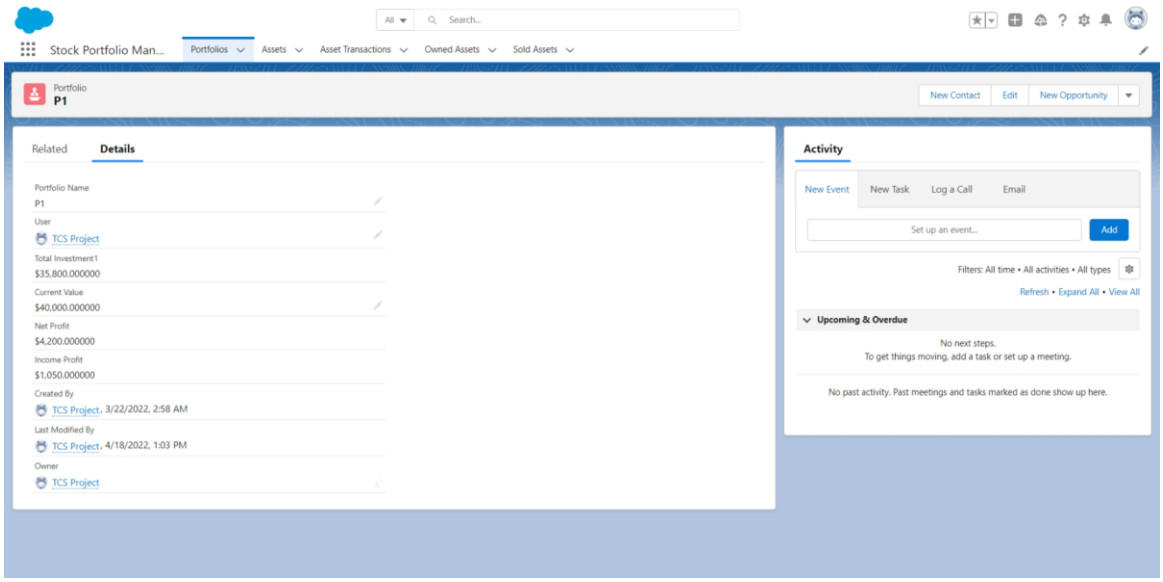

Fig. 8.14 Detail View of Single Portfolio

Below Fig 8.15 show the details view of Asset Transaction of Type Sell with portfolio, asset, type, quantity, date, transaction price, average transaction price and delta price profit.

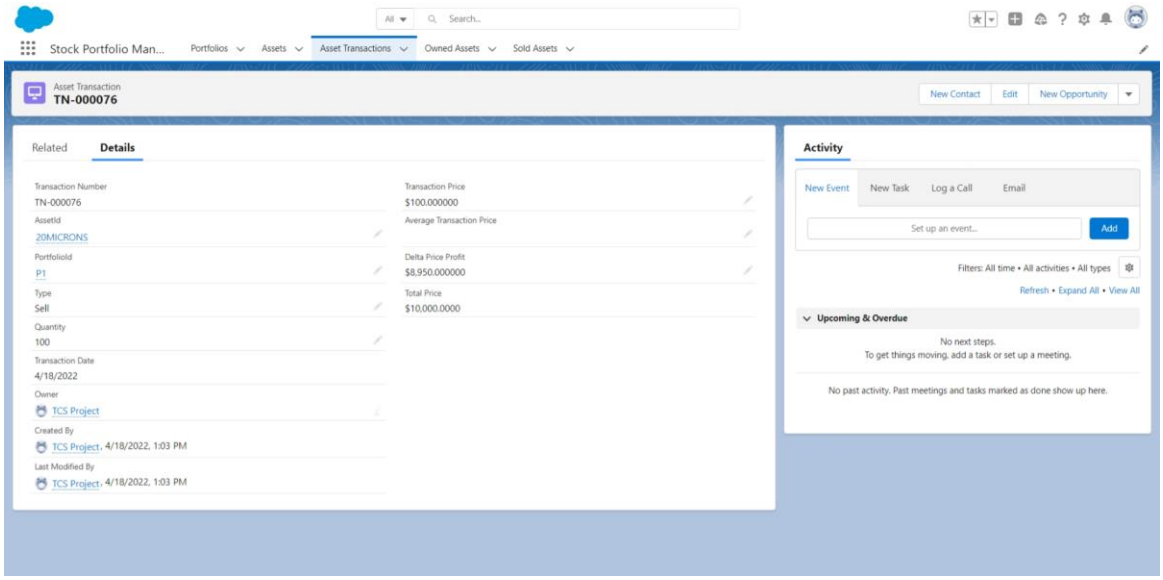

Fig. 8.15 Detail View of Single Asset Transaction of Type Sell

Below Fig 8.16 show the details view of owned Asset with asset name, portfolio, quantity, total expanse, current value and the profit/loss gained.

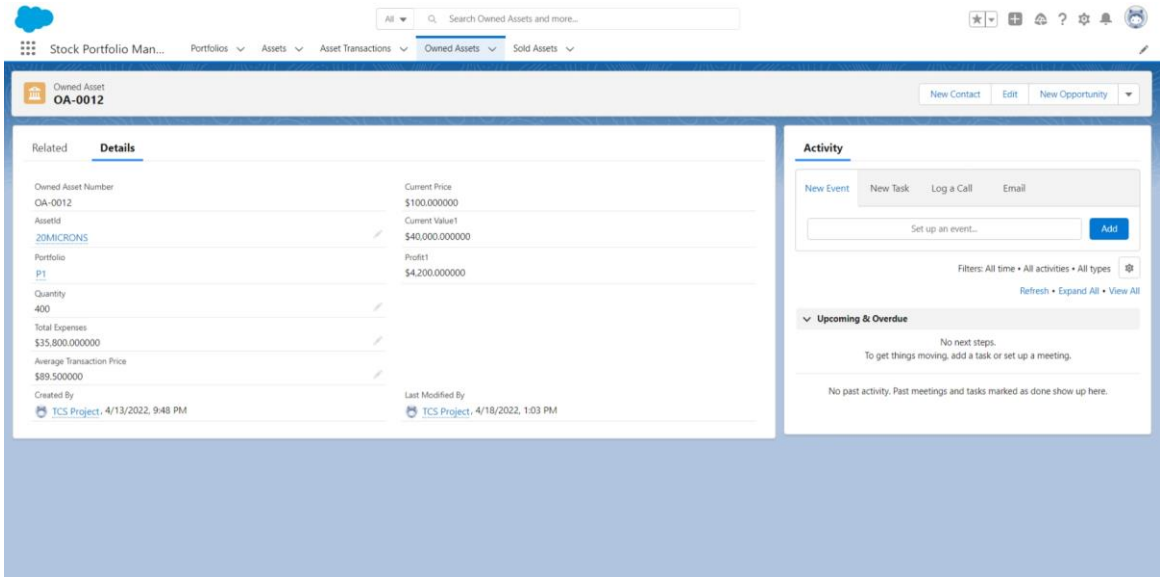

Fig. 8.16 Detail View of Single Asset Owned By User

Below Fig 8.17 show the details view of sold Asset with asset name, portfolio, quantity, and profit gained by selling the asset.

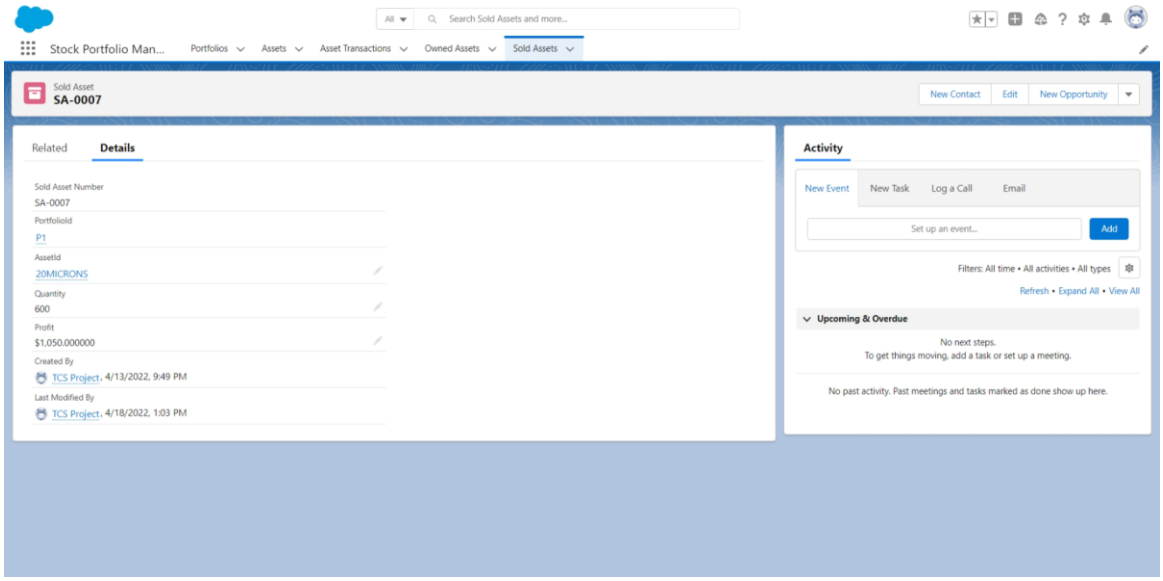

Fig. 8.17 Detail View of Single Asset Sold By User

9. CONCLUSION

9.1 SELF-ANALYSIS OF PROJECT VIABILITIES

We believe that the project design and concept are very implementable and feasible; in addition to being of great importance, we believe that the suggested product fulfills almost every expectation and suggest the best and unique investment policy for them with minimum risks involved.

9.2 PROBLEM ENCOUNTERED AND THEIR SOLUTIONS

- First Problem that we encountered is that how this whole Investment Portfolio works, what formulas are used while calculating profit and how to implement it in Salesforce. Through research and guidelines of the mentor we were able to solve most of the problem but were facing error in average price later through this formula ((price*number of shares)/total number of shares) we were able to solve this error.
- How to Update more than 2000 records asynchronously within seconds and changes done in every other object to calculate profit/loss if the value of current price increases/decreases. We were able to solve this problem through workbench where it can Process records asynchronously via Bulk API for insert, update, upsert, delete, export data, and much more. If any asset is change then the profit/loss fields will also changes through triggers.
- There were many problems occurred while implementing the Coding section as Apex Language is new to us. So, after many trial and error methods we were able to accomplish our Objective.

9.3 SUMMARY OF PROJECT WORK

In this Stock Portfolio Project, We were able to solve most of the Requirements that are needed for the user to build an investment portfolio. Now user could me able to view Summary of all the shared in his portfolio. It shows all the different shares, total shares, current price, profit/loss, net profit/loss and income gained of the portfolio in the Portfolio object. User can able to add new shares, its quantity, purchase and/or sell price to the portfolio in the Asset object. User can create many different portfolios according to his/her own needs. User can able to known how many share he has bought under which portfolio in Owned Asset object and also will able to see income when he/she sell any share at greater margin in Sold Asset object. This portfolio assists in the development of effective plans and the rebalancing of asset composition in light of current market conditions, allowing investors to make the most of their existing investments. This helps in determining which assets operate best in particular market conditions and how to allocate resources among various asset types.

My internship at Tcs was overall informative, and we learned a lot about different technologies such as Salesforce, CRM, Workbench, Triggers, Apex language and many more. Working under such a corporation was very educational, how team is formed, how to collaborate with coworker, how to use scrum methodology, how to move forward in the project with everyone contribution and much more. Overall, the experience was both educational and enjoyable. We enjoyed the work environment and team experience, and we hope to connect with them again in the future.

10. FUTURE ENHANCEMENT

- This project is still in developing phase. It has multiple customizations based on the Customer requirements and feedbacks.
- API to change current Price on seconds basis.
- Adding Mutual Funds, Debts in the Same Portfolio.
- It needs work on Analytics' and user interaction.
- This project is still not complete as its design, and some of the concepts are yet to be implemented for its complete execution.

11. REFERENCES

[1]. Apex Trigger – https://www.sfdcpoint.com/salesforce/apex-trigger-in-salesforce/
[2]. Coding Standards – https://gist.github.com/KorbenC/24f04b4d0f4bcf65ce5a
[3]. Domain Knowledge – https://www.indiainfoline.com/knowledge-center/share-market/what-is-share-market
[4]. Excel Sheet – https://docs.google.com/spreadsheets/d/17yNbjwIy_8Ngp2HgKKc57njrEMhujfaC-jjx1Xih_cI/edit?usp=sharing
[5]. Price File – https://www1.nseindia.com/products/content/equities/equities/archieve_eq.htm
[6]. Salesforce Documentation – https://developer.salesforce.com/docs
[7]. Salesforce Help – https://help.salesforce.com/s/
[8]. SDLC Models – https://existek.com/blog/sdlc-models
[9]. Trailhead Modules – https://trailhead.salesforce.com/content/learn/modules
[10]. Technical Requirements – https://help.salesforce.com/s/articleView?id=sf.technical_requirements.htm
[11]. UML Diagrams – https://www.visual-paradigm.com/ https://creately.com/ https://soft-builder.com/erbuilder-for-salesforce/
[12]. Workbench – https://workbench.developerforce.com

Final Report

ORIGINALITY REPORT

22%	16%	2%	19%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Charotar University of Science And Technology Student Paper	13%
2	www.coursehero.com Internet Source	2%
3	Submitted to Open University of Mauritius Student Paper	1%
4	Submitted to American Public University System Student Paper	1%
5	www.geeksforgeeks.org Internet Source	1%
6	www.followlist.com Internet Source	1%
7	www.educba.com Internet Source	<1%
8	2k67.com Internet Source	<1%
9	Submitted to HELP UNIVERSITY	

Sudhir Panchware

Today, 10:15 AM

Aditya Lalwani; PREETI AASI

Reply all

PRJ-2022-IT-120_18IT05...
2 MB

Download

Action Items

Hello Aditya,

Please consider this email as confirmation of Internship project "Stock Portfolio Management".

I have reviewed the project implementation and the report report (attached) you have submitted. Both are considered as completed.

Thank