

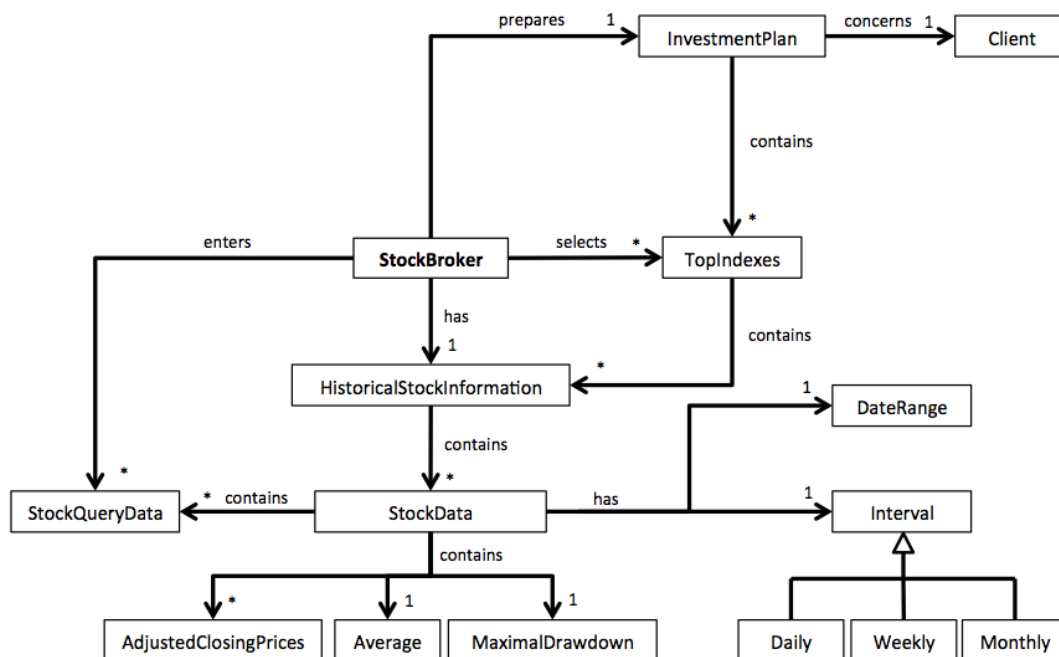
4CCS1PRA Coursework 2: Report

Introduction

This report will discuss how the 'Stock Market App Pro' application could be extended to cater for a more specific domain with additional requirements. The stock market is, again, the main domain of the extended application. However, this time the customer is assumed to be a stockbroker with interests in an application that will produce side-by-side comparisons of stock data when they submit the query information. This will allow them to provide their client(s) with a more effective and accurate investment plan on the top stock indexes. It is assumed that the last task of the stockbroker will be to return the completed investment plan to their client.

To construct an application compliant with the needs and requirements of the stockbroker, it is important and necessary to investigate and plan the exact requirements of the user. In order to do this, the domain needs to be efficiently modeled based on data gathered from the customer. So as to aid in producing an accurate analysis of the requirements, the report will incorporate: a domain model, a hierarchical task analysis, virtual windows model, a global navigation structure and an overall conclusion. Covering these models will ensure the requirements of the user are specific and fully understood so the application reflects the initial needs of the user.

Domain Model



The domain model for the application (shown above) has been constructed based on the terms of the domain mentioned within the discussion with a (hypothetical) stockbroker about her work.

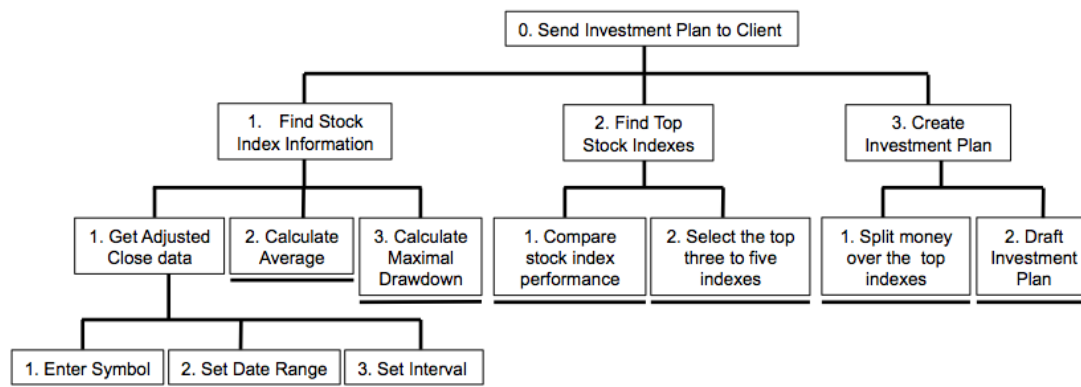
The problem that required a solution here was how and what the domain model structure should include. From noting down the key terms used within the domain of the stockbroker, and after applying careful consideration to the relationships between each term I was able to construct a model with an informative and meaningful structure.

In the design of the model I have been specific and detailed in the inclusion and structure of terms as it will provide many different domain terms that will come in useful when deciding the different types of data to be used during implementation. Moreover, I have provided arrows

with further information and a ‘*’ or a ‘1’ to indicate where there are one or more to further add to the clarity of the diagram and the relationships. For example, as is shown by the diagram, HistoricalStockData contains multiple (*) StockData, and each StockData will contain multiple (*) AdjustedClosingPrices, 1 Average and 1 MaximalDrawdown.

I have represented the domain model in this way as it mirrors the discussion with the stockbroker as the terms and how they relate to each other have been analysed and included; for example the StockBroker prepares an InvestmentPlan (as the stockbroker mentions). I also have designed it in this way to allow a potential new viewer of the domain model to still understand what is happening and how the domain terms are structured and relate. A clearer structure of the domain model allows the implementation of the application to be built more efficiently to the needs of the stockbroker.

Hierarchical Task Analysis

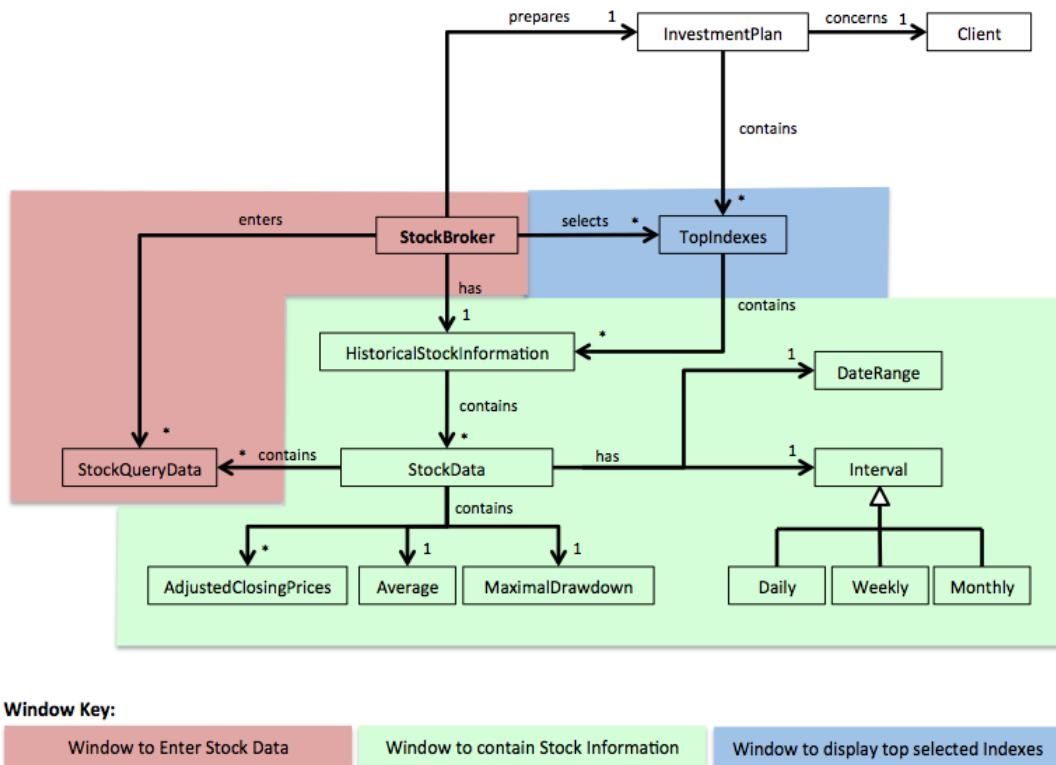


To build an effective application matching the needs of the stockbroker it is necessary to break down the tasks she wishes to carry out into individual steps. This has been done using a Hierarchical Task Analysis has been produced to represent the tasks and the subtasks that are undertaken by the stockbroker.

The problem that required a solution was how to effectively divide and display the process the stockbroker goes through to send off an investment plan to their client (it is assumed that sending off the investment plan to the client is the last task the stock broker will need to do, hence it is included in the Hierarchical Task Analysis). To do this I analysed the task process described by the stockbroker to isolate the individual tasks carried out, eventually narrowing down tasks down to their further subtasks until each task has been split into detailed subtasks. After establishing a collection of tasks and subtasks, I constructed a Hierarchical Task Analysis, displaying the tasks and their subtasks in a logical tree format. I have produced the Hierarchical Task Analysis in this way as it allowed for tasks to be easily identified and displayed in a clear and understandable format. It is furthermore easier to plan implementation when the steps to completing the tasks can be seen as task plans.

When planning the task analysis I found I could have included two subtasks to the original subtask; simply ‘Find Top Stock Indexes’ and ‘Create Investment Plan’ with ‘Find Stock Index Information’ as a subtask of ‘Find Top Stock Indexes’. However, when planning this solution, I found that it retracted from the overall clarity of the analysis as ‘Find Stock Index Information’ can be included as a subtask of the original by itself.

Virtual Windows



Using the domain model produced to show how the different potential data types interact, they can now be divided into further possible windows for the extended application (as shown above). Each different term within the domain has been placed within a potential virtual window that will allow it to become a possible data type in the implementation of that window.

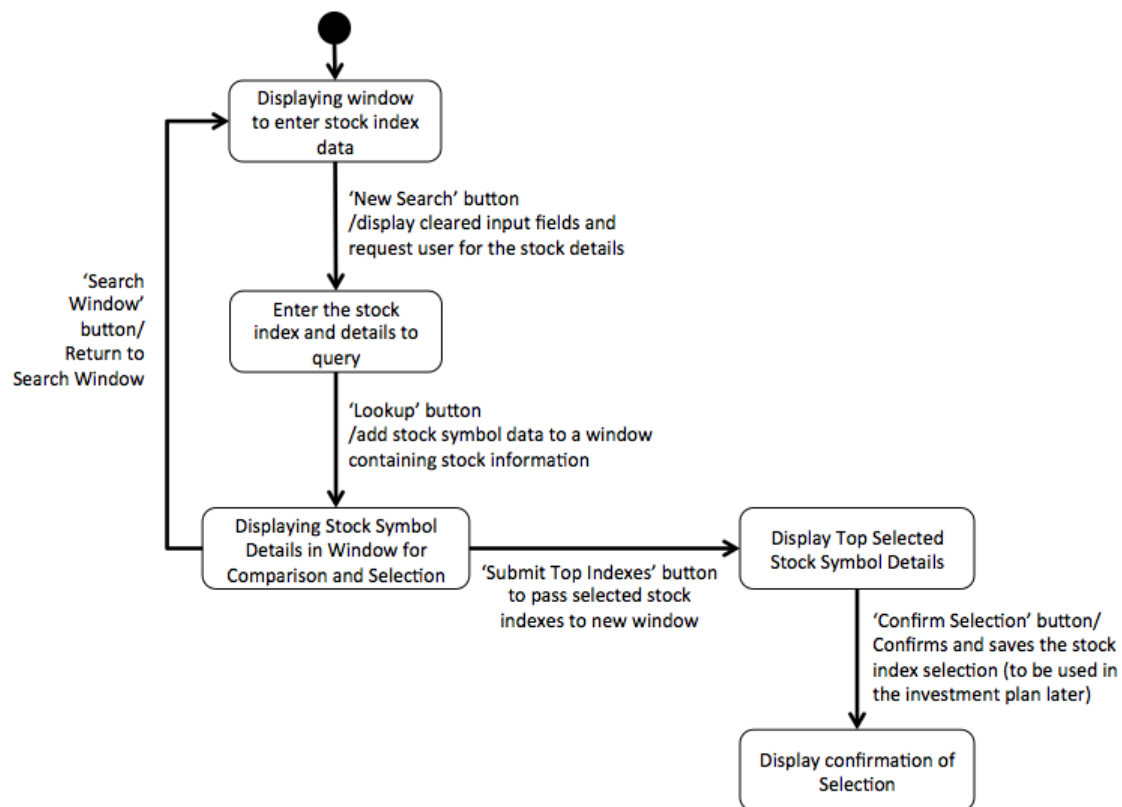
I went about building the model for the Virtual Windows by further analysing the tasks that the stockbroker must go through. She will need to enter a stock query and other details into the system to retrieve the related information; hence I have highlighted a possible virtual window to enter stock data (in the implementation they will have the option to use a 'Submit' button to submit it).

The next key requirement, which I picked from the stockbroker discussion, was her need to compare the various stock symbols queried with each other. I have provided a virtual window for this purpose; to contain all of the stock information (when queried) in a window to ensure that the comparison as simple and easy as possible (as all relevant data is there to be compared). As is shown, this virtual window contains all data to do with the stock query; previously I produced an alternative where only the StockData and its contents were included within this window. However, after further analysis I found that the dates are important to the stockbroker and important in the analysis of stock prices in general, so both the intervals and the dates have also been included within the comparison window.

The final virtual window is opened when the user has finished querying stock symbols, and has selected their chosen indexes from the comparison window. The window is designed to display the top selected indexes to allow the stockbroker to use the data in their investment plan.

The investment plan, although included in the domain model, is assumed to be completed and sent on a separate program such as Microsoft Word and Microsoft Outlook respectively. As such will not require a window within the extended application.

Global Navigation Structure



The global navigation structure shows the steps the stockbroker can take to get the final result screen of top stock indexes. Each box within the diagram represents a stage of the extended application and each arrow represents interaction from the user (e.g.: the click of a button).

To approach the structure of this model, I have used the discussion with the stockbroker in conjunction with the potential virtual windows to plan where and when users will interact with the system and on what windows they will interact with. For example, the user will begin by interacting with the main stock window that is first displayed; the user moves on to the next step interacting with the 'New Search' button and so on. Because the stockbroker has said she will need to query and compare multiple stock indexes, I have included a button back to the main window that is represented with an arrow. Furthermore, for purposes of clarity, on completion of the steps, a confirmation message will be displayed. Overall, the global navigation structure has been designed to provide a simple navigation between each virtual window and each step of the process through the use of buttons and confirmation messages. The number of steps has been kept low to ensure the user is

Summary and Conclusions for Code Implemented

In conclusion to the models produced and the research conducted, they can present a clear and informative insight into the structure any extended program must take to meet the needs of the stockbroker. From the domain model, I have learned it is important to include all terms to ensure all needs can be covered and have the right relationships to the rest of the terms. I have further established a greater appreciation of planning and modeling applications before implementing their solutions, initially the plan of implementation I had thought of was considerably different to the solution I was able to achieve with the help of the modeling.

Although this extended application plan includes many additional features to the Stock Market App Pro that I have implemented several of them in the current code. For example, many of the domain model terms have been implemented within the application and set up as data types within the application; the interval has been set up as a combo box containing the monthly, weekly and daily intervals as the stockbroker discussed.

Furthermore the virtual windows that have been implemented are similar to the ones planned; there is a main window that allows stock details to be input and queried and a window to display the stock results. In the extended application, however, the results window however would store all stock indexes side by side and allow selection of the top indexes to take the user to a new window.

Another similarity reflected in the code of my implementation is the task format of 'Find Stock Index Information'. The implementation I have provided for this task reflects the subtasks discussed in the Hierarchical Task Analysis of this task. However, the further tasks of finding the top stock indexes and creating an investment plan have only been listed to be included in the extension of the application.



The screenshot shows a window titled "Stock Market App Pro". It contains the following elements:

- A text input field labeled "Stock Symbol:".
- A "Begin:" section with three dropdown menus: the first shows "1", the second shows "Jan", and the third shows "2000".
- An "End:" section with three dropdown menus: the first shows "31", the second shows "Mar", and the third shows "2012".
- An "Interval:" section with a dropdown menu showing "monthly".
- A checkbox labeled "Chronological Order?" which is currently unchecked.
- A "Lookup" button at the bottom right.