4CCS1PRA Coursework 2: Report

Introduction

This report will concentrate on analysis of the application "Stock Market App Pro" and on how it could be extended to provide better stock analysis tool for its users, stock market brokers. The application should provide users with data they are looking for in a useful and easy way. The main purpose of the application is to allow the user make queries for stock market indexes and obtain historical stock information and then compare these information side-by-side, so that the user could accomplish their job and select several best performing stock indexes from all compared stock information and prepare an investment plan a their client.

To provide a constructive analysis of the application it is important to take into an account all possible user requirements as well as known scenarios in which the application could be used. To do so this report will consist of:

- **A domain model**, which will describe every entity of the application and how they relate to each other;
- **A hierarchical task analysis**, which will explain how each task that needs to be completed is progressively completed;
- **A virtual windows model**, which will illustrate the partitioning of individual components and actions to separate individual screens;
- **A global navigation structure**, which will provide description of separate sections and how they are linked between each other by action.

Domain Model

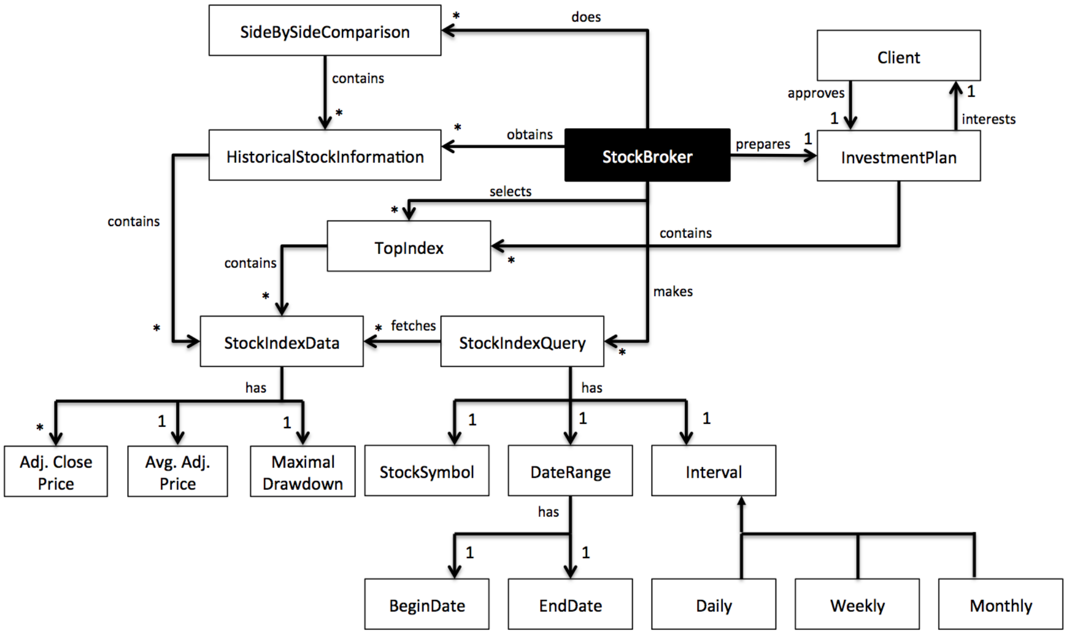

Figure 1. The domain model of the application.

The domain of this application is stock marker and its analysis. The main task of this application is to provide the user - market stock broker - with enough information so that they could make an education decision on which stocks they should suggest for their client to buy and which not. Additionally this information needs to be provided in convenient way and needs to be easy to understand and easy to use.

Based on the provided hypothetical interview with a potential user of the application we can say that the application will need to be especially designed to allow easy comparisons of different stock indexes.

Figure 1 shows the domain model for the application. The domain model provides a representation of the main concepts. The domain model starts with the user - stock broker, in the Figure 1 shown in the black rectangle. The domain model then illustrates what actions can the user do and in what states will these actions result. The domain model was created based on the provided hypothetical interview with a potential user. The rectangles indicate different parts of the application, the arrows then indicate how these parts relate to each other. The symbols “1” and “* ” also denote what kind of relations it is. If there is “1” at the end of an arrow, it means that the “parent” object has only one this property, if there is “*” at the end of a arrow, it means the relation is 1:N, meaning that 1 parent object can relate to many other if this specific type.

An example of using the application would be a user making stock index query by entering a stock symbol they are interested in, a date range and an interval. After receiving the results the user would query for more stock index information, compare these and select 3 best stocks and suggest these to their client.

This partition of the domain model into provided instances fully agrees with the provided interview, and also provide space for another extension, if that would be needed in the future.

Hierarchical Task Analysis

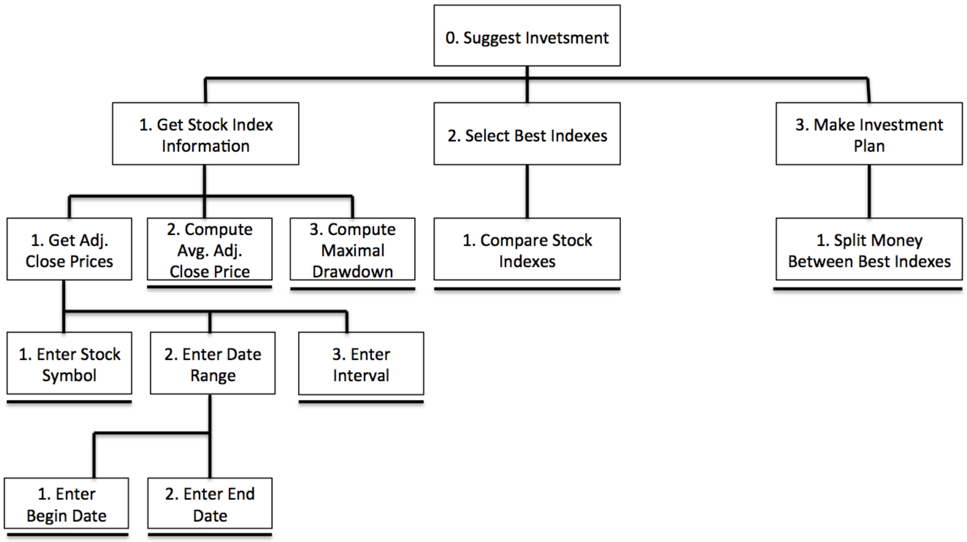

Figure 2. The hierarchical task analysis of the application.

A hierarchical task analysis represents individual tasks and their subtasks of the application. To construct a hierarchical task analysis it is important to determine which task is the main one and then into what subtasks is this main task divided. The same process is then applied to its subtasks until we reach a level at which it does not provide any addition information to divide subtasks to other subtasks. The Figure 2 shows the hierarchical task analysis of the application.

When creating the hierarchical task analysis I needed to identify the main task. I have used the provided interview and have decided that the main task is clearly providing a suggestion for an investment to a client. These are the plans following the given hierarchical task analysis:

- **Plan A:** Do 1 (incl. **Plan B1**) as often as needed, then do 2 (incl. **Plan B2**) and finally do 3 (incl. **Plan B3**).
- **Plan B1:** Do 1.1 (incl. **Plan C1**), 2.1 and then 2.3.
- **Plan B2:** Do 2.1.
- **Plan B3:** Do 3.1.
- **Plan C1:** Do 1.1.1, 1.1.2 (incl. **Plan D1**) and then do 1.1.3.
- **Plan D1:** Do 1.1.2.1 and do 1.1.2.2.

By following these tasks the user will be able to provide their client a suggested investment plan, because he will do everything what is needed to do to create an investment plan.

This provided hierarchical task analysis represents one of the variants in which the tasks could be diveded. For example selecting best indexes could be under making and investment plan, because to make an investment plan we need to find first into what we are going to invest. On the other hand we could want to select best indexes even without making an investment plan, so I have decided to put it as a separate subtask of the hierarchical task analysis.

The hierarchical task analysis shown in Figure 2 is in sound with the provided interview and satisfies potential user's needs.

Virtual Windows

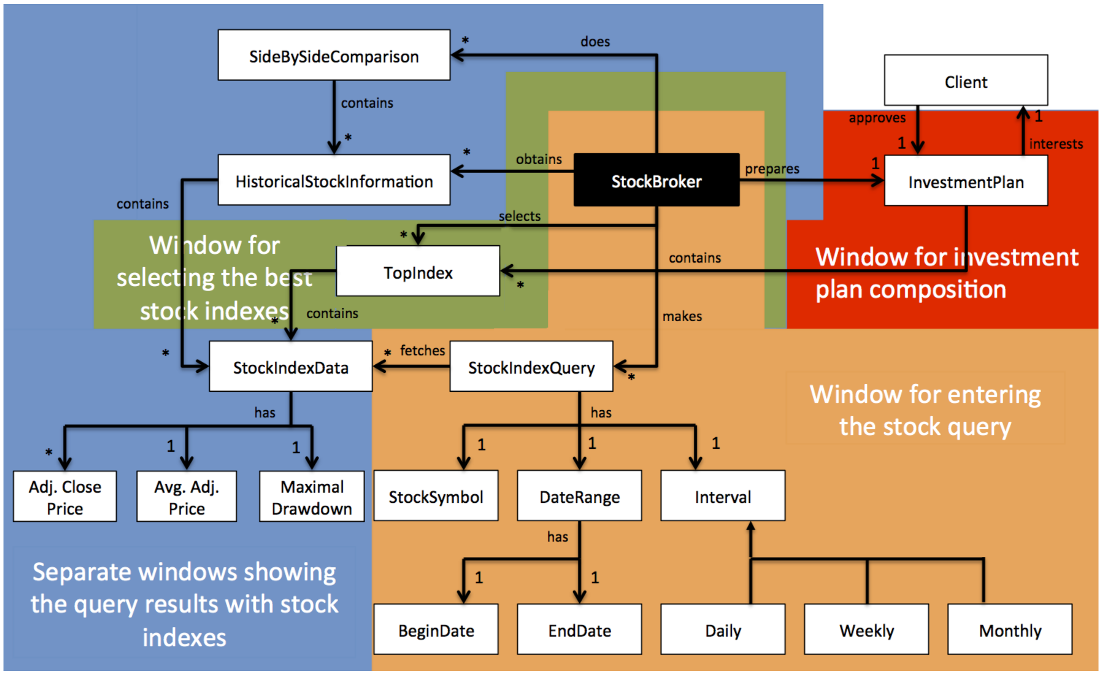

Figure 3. The virtual windows model of the application.

The goal of a virtual windows model is to divide all individual states from a domain model into separate windows, each dealing with one task or a set of similar tasks. A virtual windows model provides first concept of how will the user interface relate to the goal of the application. I have used the Domain Model from first Section of this report, and built the virtual windows model of this application (shown in Figure 3) upon it. I have further analyzed how each state of the domain and how they relate to each others.

The user - stock broker - will interact with the first window (shown yellow in Figure 3) by entering a query for stock information. In this window the user will have to provide all information describing what is the user interested in, that includes entering stock symbol, date range and interval. This window will also contain a button, which if pressed will display the second type of window.

The second window (shown blue in Figure 3) contains a table showing all information from the given range in the given interval about the given stock symbol. There can be opened many of these windows, which provides to the user the ability to place them next to each other and do the side-by-side comparison. This window will contain a button or a keyboard command to save given searches, so that the user could go back to them.

In the next window the user will load stock index information about several different stock symbols, which have been previously saved, and display all of them in one table, with only general information. If the user wants to see more details, clicking on the name of the stock symbol shows a window (blue in Figure 3) showing all detailed information about the stock symbol (from the last search). The user will also be able to select several stock symbols they think are best performing.

This selection is then transferred to last (in Figure 3 shown red) window, where the user will complete the task and will create a text document using template system. This document is then sent to a client.

These described virtual windows make it easy for the user to query for many different stock indexes and show them in a convenient way, so the user can compare the results very easily. This is in agreement with the provided interview, where there is a strong emphasis on the ability to easily compare different stock symbols and their information.

Global Navigation Structure

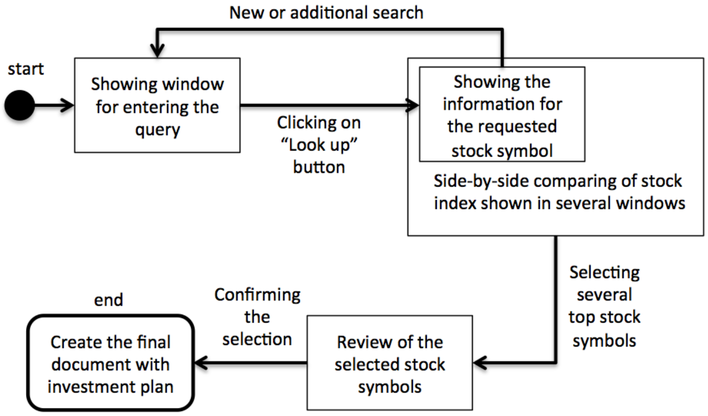
New or additional search
start
Showing window for entering the query
Clicking on "Look up" button
Showing the information for the requested stock symbol
Side-by-side comparing of stock index shown in several windows
Selecting several top stock symbols
Review of the selected stock symbols
Confirming the selection
end
Create the final document with investment plan

Figure 4. The global navigation structure of the application.

The global navigation structure illustrates how can user go from starting the application to different stages of the application. Each rectangle in the diagram represents a stage of the application and each arrow is an action the user can make to get from one stage to another.

To create the global navigation structure I have utilized the virtual windows model, and used it together with the interview with a potential user provided. I have a used each window as a reference point for a stage in which the application can be, and I have extended this ideas by having one of the rectangles (the one showing stock index information for asked symbol) in an another rectangle. The outer rectangle represents the whole stage of comparing several different windows with different results by placing them next to each other and the inner rectangle represents one window with results. There can be opened many of those windows or they can be close.

The global navigation structure has been designed so that it is easy to understand and use, but also it has been design to meet the requirements from the provided interview.

Summary and Conclusions for Code Implemented

This report provides a basic overview of the Stock Market App Pro application. It presents several modules which all look at different aspects of the application design.

I have learnt that it is important to talk to a potential user before any actual software development happens, and gather as much data, that can tell us what would users want, as possible. However, equally important is then to process this data and create a research based on it. This planning ensures that the final application will actually be useful to the user and also makes sure that if needed it will be possible and easier to extend it. It also helps to provide better estimate for how long will the application development take to finish.

Even thought this report describes many features, I have implemented just some of them. On the other hand I have designed the code in such a way, that if it will be needed, the application can be extended. I have implemented first two windows (shown in Figure 3 in yellow and blue colours). This provides fully functional application providing the functionality, which was described as the most important in the provided interview.