Investment Analysis System
5CCS2OSD
Object-oriented Specification and Design

Anton Pashov
1642171

26th November 2017

Contents

Introduction

This report is to accompany the Investment Analysis System. It consists of all the materials produced during the various stages of the project. For instance, it includes and analyses a variety of diagrams that facilitated the design stage. This project focuses on the investment topic and specifically on the purchase of bonds. Research on the bonds topic and the project brief gave a better understanding of what such a system needs. The system is a graphical application, which allows an admin to create different types of bonds. Those bonds then are available to investors to buy. From the investor's point of view, the software displays purchased bonds, along with all relevant data.

Background Study

"Investing is the act of committing money or capital to an endeavour (a business, project, real estate, etc.) with the expectation of obtaining an additional income or profit. Investing also can include the amount of time you put into the study of a prospective company, especially since time is money." (Investopedia, n.d.).

'Investing' is essentially a way for people to make a profit. It could be on a personal level where an individual invests for self-gain. On the other hand, a business might invest towards its growth on a much larger scale. Investments usually consist of giving money to an organisation (i.e. a bank) that re-invests it on a significantly larger scale. Depending on how much money individuals invest, they receive a portion of any turnaround the organisation makes.

There are various ways to invest. One is to loan money to those who need it and then add some interest. There are services available that make this process a lot easier by facilitating the connection between investors and managing the whole procedure. An example of such a service provider is "LendingClub". They provide investment opportunities for both institutions and individuals.

The current project focuses on a specific type of investment - bond. It is an investment where the investor loans money to a company. The organisation borrows the money for an agreed amount of time. During this time, it periodically pays the investor a variable or fixed interest rate. Finally, it returns the initial investment.

Bonds have properties defining their profitability. They typically last for a couple of years. The duration is known as 'term'. The coupon is the percentage (fixed or variable) of the original investment that the company pays to the investor at a given interval. When a term ends, the organisation pays the last coupon and returns the investment. When an investor examines purchasing a bond, he needs to consider the inflation rate and currency value.

Due to the gradual inflation that decreases the value of a pound, £100 today will not be worth the same in two years.

The Macaulay duration of a bond "measures the present value weighted average maturity for a bond. It describes how sensitive a bond's price is to changes in interest rates." (Investopedia, 2015)
A good Macaulay duration can solidify the decision to invest in a bond. The longer the term of a bond, the greater the risk the investor is taking due to the volatile value of money. Especially since all inflation rates are estimations, anything could happen.

Simply put, the Macaulay duration is the number of years after the purchase of a bond when the sum of the cash flows received would equal the amount paid for the bond. Therefore, a quickly maturing bond is a safer bet, as there will be a shorter time frame in which the value of money could possibly decrease.

The internal rate of return (IRR) is a percentage which states the rate at which the value of the bond equals the price paid, without taking factors such as discounted payments or inflation into account. An IRR can be calculated for most investments and is not necessarily restricted to bonds. All that is needed to calculate the IRR is the price paid and the cash flows. Example, if we pay £80 for a bond and get back the cash flows of: £10, £10, £10, £110; the IRR would be equal to 17.339%. This is as when using this rate:

$$10/(1 + 0.17339)1 = 8.52$$

$$10/(1 + 0.17339)2 = 7.26$$

$$10/(1 + 0.17339)3 = 6.19,$$

$$110/(1 + 0.17339)4 = 58.03$$

The sum of the results is £80 which is the amount we paid. If the IRR is being calculated manually, it can only be done through trial and error. Fortunately, software offering this functionality is available (e.g. Microsoft Excel). If the inflation rate in reality is higher than the calculated IRR, then the bond is not a good deal for investors, if it is lower than the IRR, it is a good deal for investors.

Requirements

From analysing the project brief, several key requirements have been identified.

Firstly, the system should allow new bonds to be defined with the attributes: *name, term, coupon, frequency of payments, purchase date, and price.*
The system should have the ability to add bonds to an investor's portfolio too.
In terms of computations, there are four calculations that the system is required to perform.

- Computing the sum of all payments from a bond.
  - If a bond pays out the following: £20, £20, £20, £120. The system should display the sum as £180. The payments would be calculated from the coupon, here the investor has invested £100 where the coupon is paid annually at a 20% rate.
- The system ought to be able to do this whilst taking an inflation rate into account too (as a separate feature).
  - For the same bond with a 2% inflation rate, the payments would be: £19.61, £19.22, £18.85, £110.86. The sum that would be expected from the system here is £168.54.
- Calculating the Macaulay duration of a bond.
  - The system should be able to calculate that the above bond has a Macaulay duration of 3.31 years.
    - £19.61 $*$ 1 =£19.61
    - £19.22 $*$ 2 =£38.44
    - £18.85 $*$ 3 =£56.55

- $£110.86 * 4 = £443.44$
- Total: £558.04
- $£558.04/£168.54 = 3.31$ years

- Calculating the internal rate of return from knowledge of the payments and amount paid for the bond.
  - Using the example bond's payments and assuming the bond price was £160, it should be possible to request that the internal rate of return is computed. The expected response in this case would be 3.619% since:
    - $£20/(1+0.03619)^1 = £19.30$
    - $£20/(1+0.03619)^2 = £18.63$
    - $£20/(1+0.03619)^3 = £17.98$
    - $£120/(1+0.03619)^4 = £104.09$
    - the sum of the results being £160 (the price).

Ambiguity

There are some ambiguities that are apparent with the assignment brief. The first ambiguity noticed was that the brief states: *'The system enables new bonds to be defined'*. However, it does not mention which type of users define them, nor does it state which type of users will be using the system. If the system is used by investors solely, then who defines the new bonds? Surely, if an investor had the opportunity to define bonds himself, he would define it to maximise his profits. This is not realistic as organisations selling bonds would essentially be giving out free money.

Another ambiguity is the amount that investors invest into each bond. The brief states that the investment is "100" yet doesn't say whether that is to mean £100, or means 100% of another value.

It is also unclear whether the name of bonds can be used to uniquely identify them as it has not been stated.Furthermore, the brief tells us that investors can purchase bonds, but it does not mention whether an investor may purchase the same bond more than once.

Although we are told the properties of the bonds (e.g. coupon, price, name), the data types of these attributes have not been expressed.

The number of investors using the system is ambiguous too, the brief does not give any indication that more than one investor might be using the system. This is as it has not mentioned that the system should have the functionality to create portfolios. If the system should be able to create additional portfolios, would some sort of authentication be required to prevent users from accessing another investor's portfolio?

Finally, the brief never mentions that there are different types of bonds in the real world. Therefore, is the system to stick to the type of bond described in the brief, or should it include other types such as "Gilts" (government bonds).

Assumptions

The following assumptions took place to handle the ambiguities stated above:

- An admin would have the authorisation to define new bonds, whereas an investor would only be able to view and purchase pre-defined bonds.

- The authentication system will be disjoint with the investment analysis system. Due to this, our system will make all operations available and assume that the current user has already been authorised (i.e. logged in).

- The amount that investors will invest into each bond is assumed to be £100, as it seems like this is what was meant when the brief said "investment = 100".

- An investor may purchase the same bond more than once. This is because there seems to be no harm in allowing an investor this opportunity. Especially since there was no mention of there being a limited quantity of each bond.

- The data types of the attributes will be what is deemed to be most appropriate. For instance, the name of a bond will be a String, and the term of a bond will be an integer.

- There will be only be a single type of bond, and it will be the way bonds have been described within the brief.

- The name of a bond isn't necessarily unique, thus bonds will now have an additional attribute known as "ID" which will allow bonds to be unique identified.

Specifications

Use Cases

For the current project, there will be two types of users - *Admin* and *Investor*. Even though the software will be investor-based, the admin also has an important role in the process.

Use Case Diagram

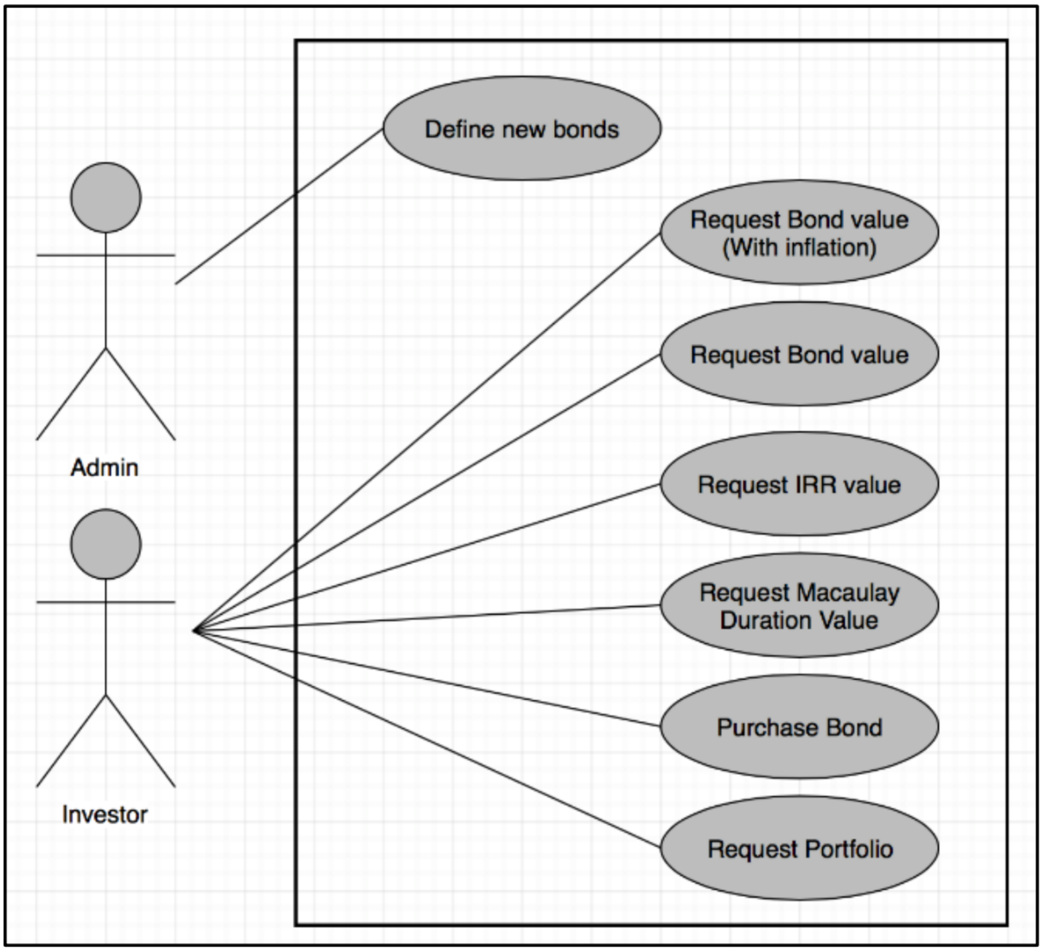

Figure 1: Use case diagram

The use case diagram clearly shows that our investment analysis system is primarily aimed at investors. The only use case an admin would take part in is actually defining the bonds.

The reason why the admin isn't depicted to be affiliated with any of the other use cases is as when playing the role of an admin, he should only define bonds from a company standpoint. If the admin defined and bought the bonds, the scenario that was described within the requirements' elicitation could occur (bonds being defined to maximise investor profit).

This use case diagram where the admins are separated from the investors is more realistic too. As commonly, investors do not get to define their own bonds.

Use Case Steps

This subsection briefly explains what steps are involved in completing each of the use cases.

Define New Bond

1. The admin asks the system to define a new bond
2. The admin enters name, price, term, coupon and an estimated inflation rate for the new bond
3. A new bond is created and added to the market

Purchase Bond

1. The investor selects a bond that they want to purchase
2. The investor confirms the intention to buy the selected bond
3. The system displays their newly purchased bond in their portfolio

Request IRR Value

1. Investor opens their portfolio
2. Investor can see IRR value displayed alongside their selected bond

Request Bond Value

1. Investor opens their portfolio
2. Investor can see the sum of pay-outs displayed alongside their selected bond

Request Bond Value(With Inflation)

1. Investor opens their portfolio
2. Investor can see the bond value displayed alongside their selected bond

Request Macaulay Duration Value

1. Investor opens their portfolio
2. Investor can see Macaulay duration displayed alongside their selected bond

Request Portfolio

1. Investor selects the *Portfolio* tab on the system
2. System displays all bonds within investor's portfolio

Class Diagram

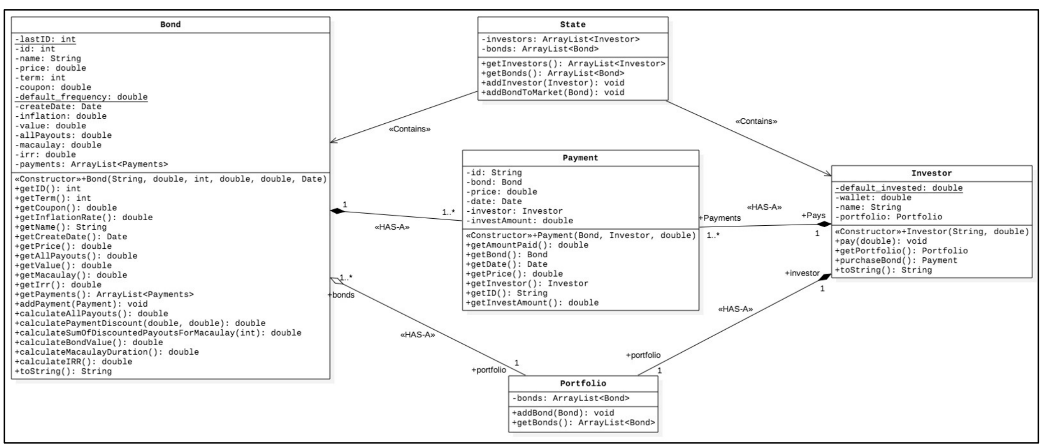

Figure 2: Class diagram

Architecture

The three-tier architecture is able to display how each level interacts/communicates with one another.

As the names of the two GUIs (Graphical User Interface) imply, the "Staff GUI" would be used by the staff of the system, and the second would be used by Investors.

Each GUI has its own set of operations within the functional core. This makes sense as they each play a different role in the system. If they did indeed have the same operations, there would be no point in separating them out, as this would just lead to redundancy.

The difference in access level for the two types of users has been stated on the arrow connecting to the bond system. As expected, staff have greater access clearance by being able to write as well as read from the data repository, whereas investors may only read.

Architecture Diagram

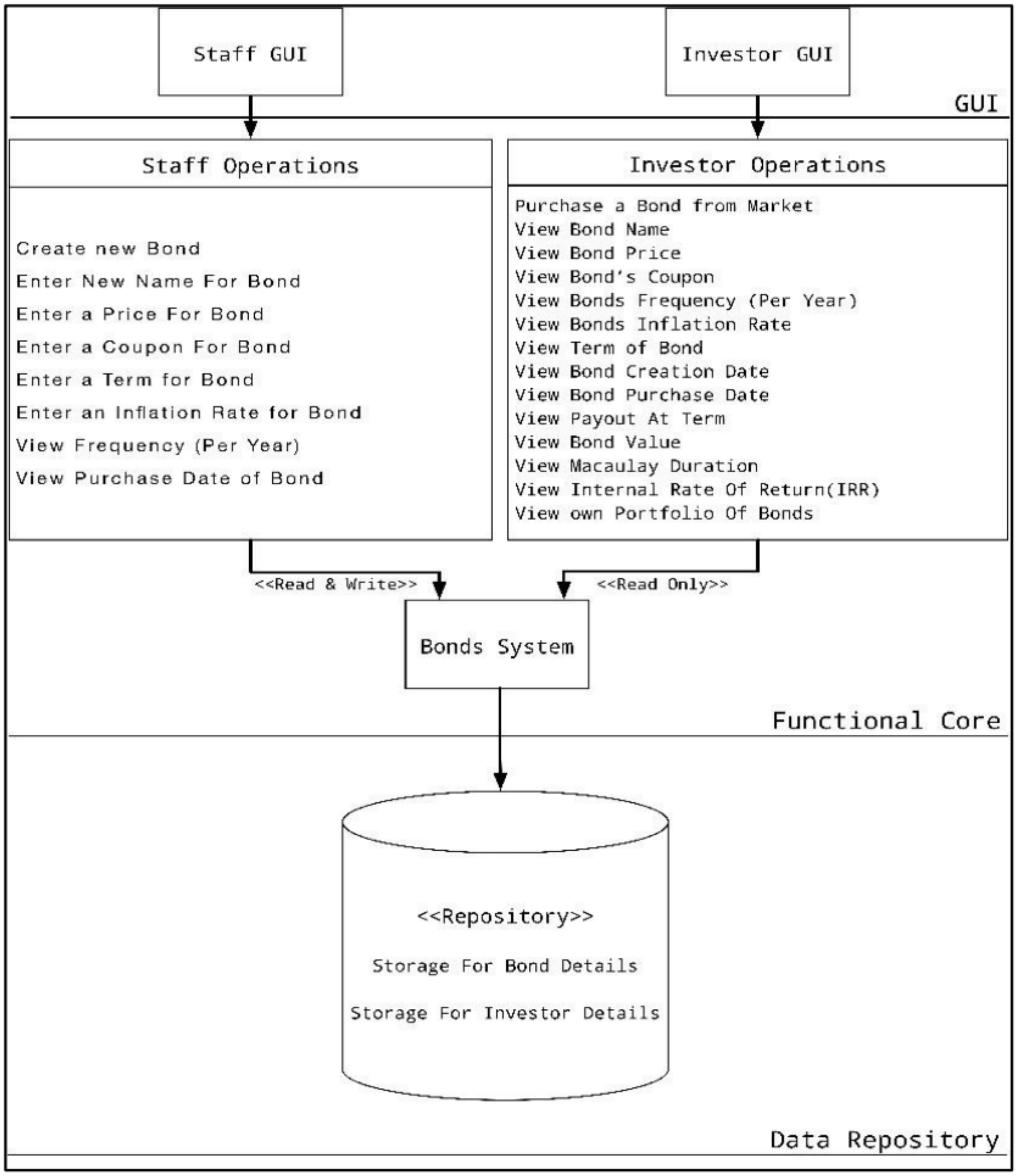

Figure 3: Architecture diagram

Operation Definitions

Below, the four main operations that the functional core offers have been defined. These are the operations that calculate valuable information for investors, such as calculating the bond value, the Macaulay duration, and computing the Internal Rate of Return.

Calculate All Pay-outs

This operation calculates the sum of the payments that would be received from a bond. This computation requires the bond's coupon and the amount invested (fixed at £100). Essentially, this function works out the amount that the bond annually pays from knowledge of the coupon. Then it sums up the payments by multiplying the amount paid by the term. As the investment is returned at the end of the term, the £100 is then added to the sum of the payments and the result is returned.

Calculate Bond Value

Calculating the bond value has the same algorithm as the operation above, the difference is that in order to obtain an educated estimate of the bond value, an inflation rate is taken into account. Therefore, each payment is re-evaluated using the inflation rate and the year that this payment would have been made. The formulae for this is:

$$Value = \frac{X}{(1+r)^n} \tag{1}$$

Where X is the payment, r is the inflation rate, and N is the year of the term of when this payment would be made. For instance, if a payment was £10 and was to be paid 3 years into the term; the value would be £8.64 assuming the inflation rate was 5%, and would be £9.42 if the inflation rate was 2%.

Calculate Macaulay Duration

To calculate the Macaulay Duration of a bond, the discounted payments are required. There are three calculations that need to be processed to output the

Macaulay Duration. Firstly, the bond value needs to be obtained (previous operation). Then, the sum of the discounted payments multiplied by their corresponding year is needed. Finally, the latter is divided by the first, giving us the Macaulay Duration.

Calculate IRR

The IRR was the most computationally expensive operation within the functional core (relative to the other operations). The reason being is that, as mentioned in the background study, analytically calculating the IRR is a very complicated mathematical operation. The formula for calculating IRR is:

$$0 = \sum_{t=1}^{T} \frac{C_t}{(1 + IRR)^t} - C_0 \tag{2}$$

Where:
C_t =Net cash inflow during the period t
C_0 =Total initial investment cost
IRR =The internal rate of return
t =The number of time periods

Analytically solving such equation would leave us solving a variable with rising degree in the denominator.However, there is a relatively neat workaround. It can be derived when some trial and error is involved. The algorithm for this operation was simply to start off with a blind guess of what the IRR could be, then the bond value is calculated using this IRR to see if it matches the price of the bond. Depending on how far off the result was from the actual bond price, the IRR is then adjusted to get a more accurate result. This is then repeated until it satisfies the pre-defined margin of error.

Pseudocode for Operations/Use Cases

Pseudocode for the key calculations within the functional core have been provided below. These include calculating the sum of the pay-outs for a bond, the bond value, the Macaulay Duration for a bond, and finally computing the internal rate of return for a bond.

Calculate All Pay-outs

Algorithm 1: query calculateAllPayouts() : double

Result: INVESTED + (term * FREQUENCY) * (INVESTED * coupon)

Calculate Bond Value

Algorithm 2: query calculatePaymentDiscount(amount : double , yearsPassed: double) : double

Result: result = amount/((1+inflation) ˆ yearsPassed)

Algorithm 3: query calculateBondValue() : double

```
value : int = 0;
for i : term do
    if i == term then
        value += calculatePaymentDiscount(INVESTED * (1 +
         coupon) , term)
    else
        value += calculatePaymentDiscount(INVESTED * coupon, i)
    end
end
return value;
```

Calculate Macaulay Duration

Algorithm 4: query calculateSumOfDiscountedPayoutsForMacaulay(x: int) : double

```
value : int = 0;
for i = 1:x do
    if i == x then
        value += calculatePaymentDiscount(INVESTED * (1 +
          coupon) , term) * i
    else
        value += calculatePaymentDiscount(INVESTED * coupon,
          i) * i
    end
end
return value;
```

Algorithm 5: query calculateMacaulayDuration() : double

Result: macaulay = calculateSumOfDiscountedPayoutsForMacaulay(term) / calculateBondValue()

Calculate IRR

Algorithm 6: query calculateIRR() : double

```
payments : ArrayList<double>;
payments.add(price * (-1));
for i = 1:term do
    if i == term then
        payments.add(INVESTED * (1 + coupon))
    else
        payments.add(INVESTED * coupon)
    end
end
guess : double = random value ;
error : double;
approximation : double = 0.0001;
isPositiveOld : boolean = true;
isPositiveNew : boolean;
multiplier : double = 0.1;
while true do
    error = payments[0];
    for i:payments.size() do
        error += payments[i]/((1 + guess)ˆi)
    end
    if error >0 then
        isPositiveNew = true
    else
        isPositiveNew = false
    end
    if isPositiveNew != isPositiveOld then
        multiplier = multiplier / 10
    else
    end
    if (error >0 then
        guess += multiplier
    else
        guess -= multiplier
    end
    if error >0 – approximation and error <0 + approximation
     then
        irr = guess;
        return guess
    else
    end
    isPositiveOld = isPositiveNew;
end
```

Graphical User Interface

Self-explanatory screenshots of the user interface have been taken and provided below