Vienna University of Technology

Institute of Software Technology and Interactive Systems

IT-based Management WS2017

User Guide - Group 21

Team Members	Matr.Nr.
Blagoy Panayotov	01643722
Diego Montano	01429264

November 2017

Contents

1 Executive summary

With regards to providing a software solution, group 21 will strive to design and implement a call-option replication. Our goal is to establish a new module for the Shiny-ERP application and to document the module which will ultimately solve the financial engineering problem. In order to achieve our overall mission of how to solve the problem of a call-option replication, we will focus on strengthening the following:

- Providing a new module for the Shiny-ERP application implementing the Plan-Do-Check-Act (PDCA) management cycle.

- Providing a brief documentation of the module itself.

2 Management team

Group 21's Management Team is responsible for the strategic direction and the day-to-day operations of the company. Covering all aspects of operations, management, implementation, and design, the team works together to ensure the solution of the financial engineering problem.

- **Diego Montano** *Project Manager*

  As group 21's project manager, Diego is responsible for planning the project with respect to timely and functional fulfillment.

- **Diego Montano & Blagoy Panayotov** *Business Analyst*

  Diego and Blagoy have the responsibility of design the solution (Semi-automated model driven development MDD).

- **Blagoy Panayotov** *Developer*

  Blagoy is scaling Engineering to address the challenges that arise with group 21's implementation of the design.

These main tasks are defined in a way that best utilizes the skills of the team members, however during the initial set-up of the business all the team members will participate in all the necessary tasks and will provide their support further beyond this basic responsibility definition. The same regarding the other tasks from the set-up phase, everyone will participate when necessary in the ongoing activities.

3 Iceberg List

The following table breaks down the development cycle into tasks. Every task is marked with an identifier, a categorization ("Feature"), the work, an expected time effort, a version when the feature will be implemented and the principal responsible team members for the implementation.

ID	Feature	Technical Task	Expected time effort (h)	Version	Responsible
1	Preparation	Iceberg list	1	1,1	Entire Team
2	Preparation	Planning and understanding of the project	4	1,1	Entire Team
3	Preparation	Setting up R-studio	1	1,1	Entire Team
4	Preparation	Repository setup	1	1,1	Entire Team
5	Preparation	Setting up existing IT infrastructure	1	1,1	Entire Team
6	Preparation	Getting up with Black-Scholes Model (formula)	2	1,1	Entire Team
7	Preparation	Getting up with Data layer	2	1,1	Entire Team
8	Preparation	Getting up with the Business layer	2	1,1	Entire Team
9	Initial pricing	Implement UI on the OntoREA Prototype	2	1,2	Blagoy
10	Initial pricing	Implement storing of the initial data in masterdata database	2	1,2	Blagoy
11	“Do” step	Getting up the input from the user	1	1,2	Blagoy
12	“Do” step	Storing the processed input of the database	2	1,2	Blagoy
13	“Plan” step	Calculating the new weights N(d1)	3	1,2	Diego
14	“Check” step	Retrieve N(d1, t-1) from database	3	1,3	Diego
15	“Check” step	Compare N(d1, t) to N(d1, t-1) and propose rebalancing of portfolio	2	1,3	Diego
16	“Act” step	Perform the proposed rebalancing of portfolio	3	1,3	Blagoy
17	“Act” step	Calculate the portfolio fair value and store	2	1,3	Blagoy
18	“Act” step	Show graph of the changes of the replication portfolio	3	1,3	Diego
19	Testing	Testing database	1	1,4	Entire Team
20	Testing	Testing calculations	1	1,4	Entire Team
21	Testing	Testing the graph	1	1,4	Entire Team
22	Testing	Fixing bugs	4	1,4	Entire Team
23	Testing	Final project test	1	1,4	Entire Team
24	Final step	Do the documentation	2	1,5	Diego
25	Final step	Do the user guide	2	1,5	Diego
26	Final step	Final presentation	1	1,5	Entire Team
		Total effort	50		

Figure 1: Iceberg list

4 Implementation & Design

Our starting point will be focused in the creation of the repository on GitHub. The repositoy is https://github.com/bogy159/IT-based-Management/

4.1 Data model

The data model is provided and will be used to implement the business logic regarding the PDCA management cycle.

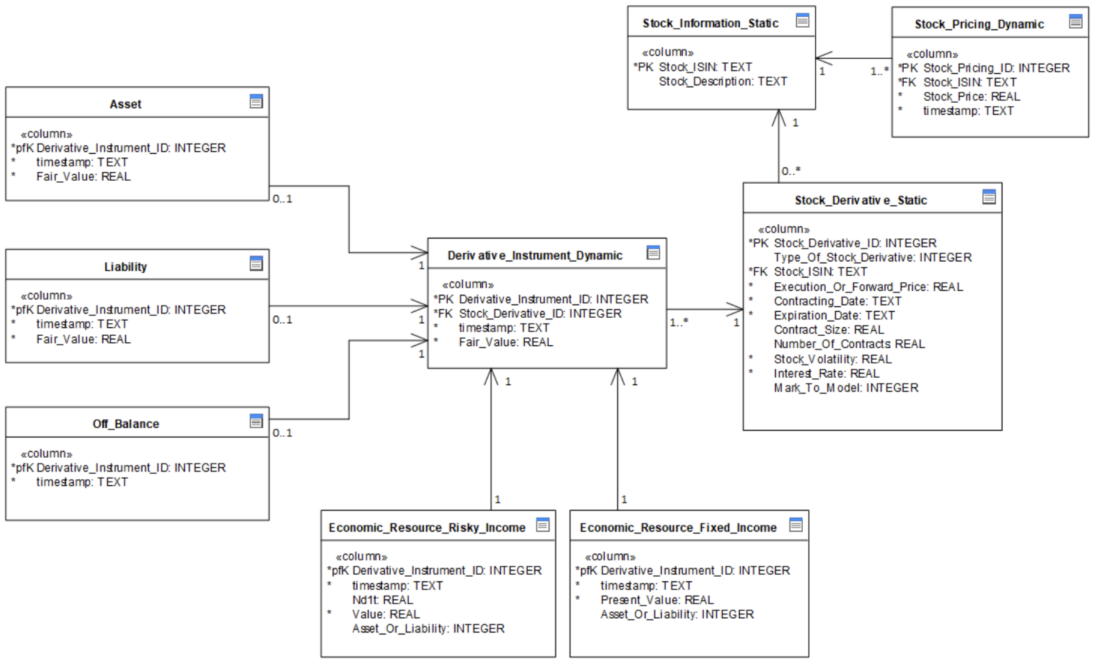

Figure 2: Datamodel of Shiny-ERP.

4.2 Architecture

In order to create the Plan-Do-Check-Act management cycle, we will use R Shiny and the SQLite database to implement the required call-option replication policy.

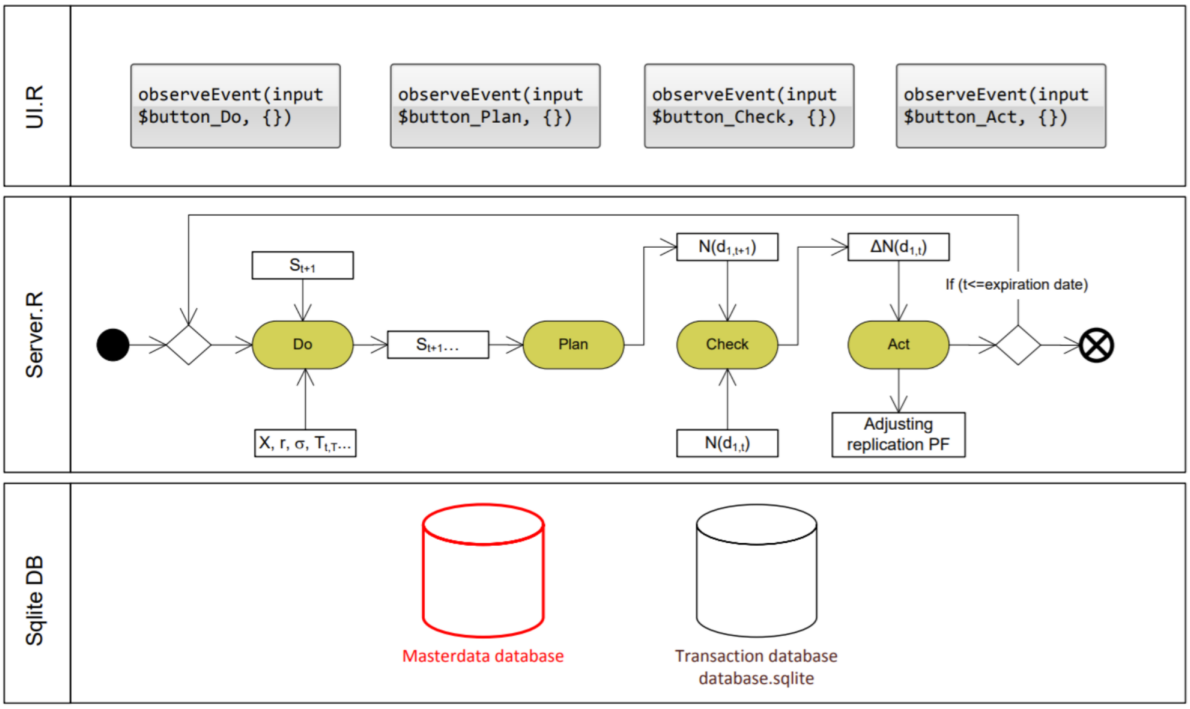

Figure 3: Layer Structure of Shiny-ERP.

4.2.1 Data layer

The data layer consists of transaction and masterdata. Master data remains unchanged over a period of time and is used to structure transaction data.

4.2.2 Business logic layer

Now, the PDCA management cycle is implemented within the Shiny-ERP application as it follows.

First, we select the "Table Explorer", and we load our data from the database for every table.

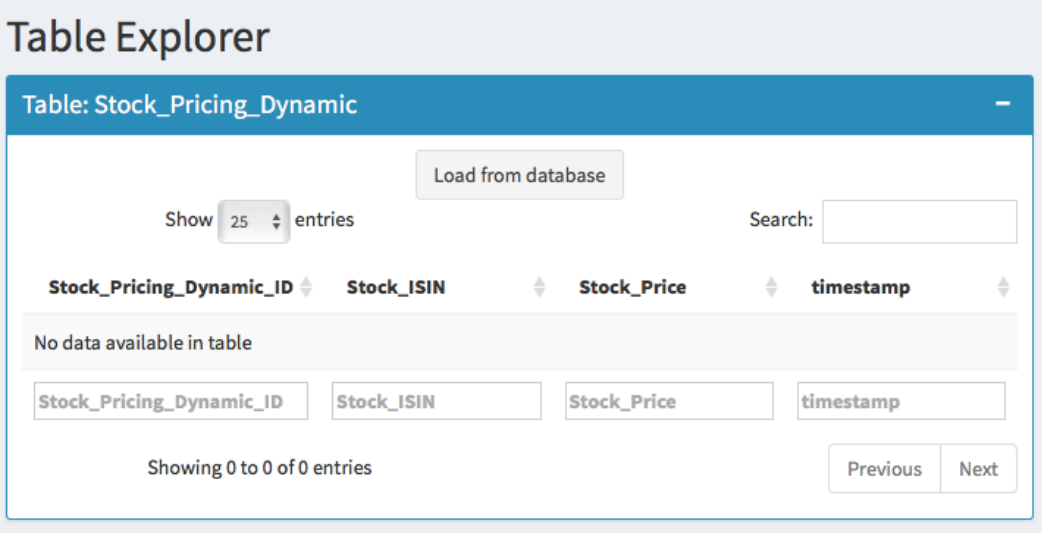

Figure 4: Load of the data from database.

The replication policy is implemented according to the activities in the PDCA management cycle:

- **"Do" activity.** In this step, the data input from the user is asked to fill in a timestamp in the form of date and a stock price. To do this, we just have to save the data that the user insert and keep it in a variable called t with a specific format (See Figure 5).

```
temp_db_draw <- dbReadTable(sqlite, "Stock_Pricing_Dynamic")
temp_db_draw$Pricing_Date <-
  as.Date(as.POSIXct(temp_db_draw$timestamp))

t <- as.numeric(difftime(as.Date(input$ti_Expiration_Date2), as.Date(input$ti_Do_timestamp2), unit="weeks"))/52.25
t <- round(t, digits = 2)
```

Figure 5: Storing the date as a variable.

The stock price is information given, and this variable depends on the date we are analyzing. So for the first one (31.03.2020), the Stock price is 100€.

Then the app writes down the result into the database in Stock_Pricing_Dynamic table.

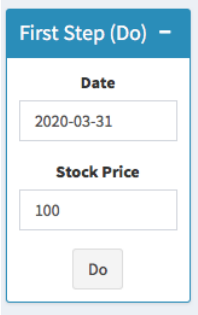

Figure 6: First step (Do).

- **"Plan" activity.** In this step, we focus on the calculation of the new weights $N(d_1)$ for the risky income security (stock). To do this calculation, we apply the formula to determine the amount of the portfolio weight $N(d_1)$ as it follows:

$$d_{1,t} = \frac{ln(\frac{P_{A,t}}{X_{0,T}}) + (r_t, T + \frac{\sigma^2_{0,T}}{2}) \cdot T_{t,T}}{\sigma_{0,T} \cdot \sqrt[2]{T_{t,T}}}$$

Figure 7: Formula to calculate d1.

For the calculation, first we need to "convert" the time (t). By contract, it's defined that the duration is for one year (Starts on the 01.01.2020), so we get the time by checking how much time is the expiration from the starting until the first period that we are calculating (For the first one, it's on 31.03.2020).

```
t <- as.numeric(difftime(as.Date(input$ti_Expiration_Date2), as.Date(input$ti_Do_timestamp2), unit="weeks"))/52.25
t <- round(t, digits = 2)

d1 <- (log(as.numeric(input$ti_Do_Stock_Price2)/as.numeric(input$ti_Exercise_Or_Forward_Price2)) + (as.numeric(input$ti_Interest_Rate2)/100 +
Nd1 <- pnorm(d1,lower.tail = TRUE)
d2 <- (log(as.numeric(input$ti_Do_Stock_Price2)/as.numeric(input$ti_Exercise_Or_Forward_Price2)) + (as.numeric(input$ti_Interest_Rate2)/100 -
Nd2 <- pnorm(d2,lower.tail = TRUE)
```

Figure 8: Formula to calculate d1.

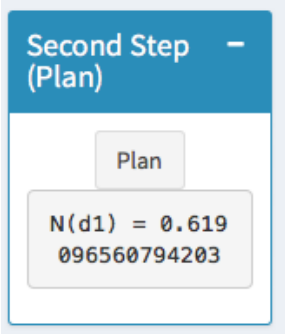

Figure 9: Second step (Plan).

- **"Check" activity.** In this step, the new weights are compared to the old weights. This is, the variable $N(d_1)$ of the first period minus the $N(d_1)$ of the starting period. The new variable is called delta $N(d_1, t\text{-}1)$. Once we compared these two variables, we proposed rebalancing of portfolio. For the calculation, we have to load previously the value of the $N(d_1, t)$ from the Economic_Resource_Risky_Income table. The changes determine the adjustments of the risky income security in the replication portfolio. Now we do as well the calculation of the necessary variables following the other formulas.

Value of the call option at time t	$FV_{C,t} = ASSET - LIABILITY$
Asset side of the portfolio	$ASSET = P_{A,t} \cdot N(d_{1,t})$
Liability side of the portfolio	$LIABILITY = X_{0,T} \cdot exp(-r_{r,T} \cdot T_{t,T}) \cdot N(d_{2,t})$

Figure 10: Rest of formulas.

The required changes lead to changes in the fixed income position of the replication portfolio, to finance increasing risky income positions as well as to the redemption of the fixed income position in the case of decreasing risky positions.

```
t <- as.numeric(difftime(as.Date(input$ti_Expiration_Date2), as.Date(input$ti_Do_timestamp2), unit="weeks"))/52.25
t <- round(t, digits = 2)

d1 <- (log(as.numeric(input$ti_Do_Stock_Price2)/as.numeric(input$ti_Exercise_Or_Forward_Price2)) + (as.numeric(input$ti_Interest_Rate2)/100 +
Nd1 <- pnorm(d1,lower.tail = TRUE)

d2 <- (log(as.numeric(input$ti_Do_Stock_Price2)/as.numeric(input$ti_Exercise_Or_Forward_Price2)) + (as.numeric(input$ti_Interest_Rate2)/100 -
Nd2 <- pnorm(d2,lower.tail = TRUE)

Liability <- as.numeric(input$ti_Exercise_Or_Forward_Price2) * exp((-1) * t * (as.numeric(input$ti_Interest_Rate2)/100)) * Nd2
Asset <- Nd1 * as.numeric(input$ti_Do_Stock_Price2)
FairValue <- Asset - Liability
```

Figure 11: Calculation of delta N(d1, t-1).

Figure 12: Third step (Check).

- **"Act" activity.** Finally, the changes of the replication portfolio are executed. To do this, first we perform the proposed rebalancing of portfolio and we calculate the Portfolio fair value, the N(d1, t) (See figure 11) and we store this values as follows in the Asset_table, Liability_table and in the Off_Balance-Phase table.

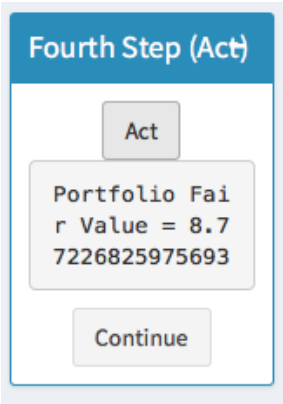

Figure 13: Fourth step (Check).

5 Outcome

The replication policy for a European call-option is designed and implemented according to the activities in the PDCA management cycle as it could be see in the last figure.

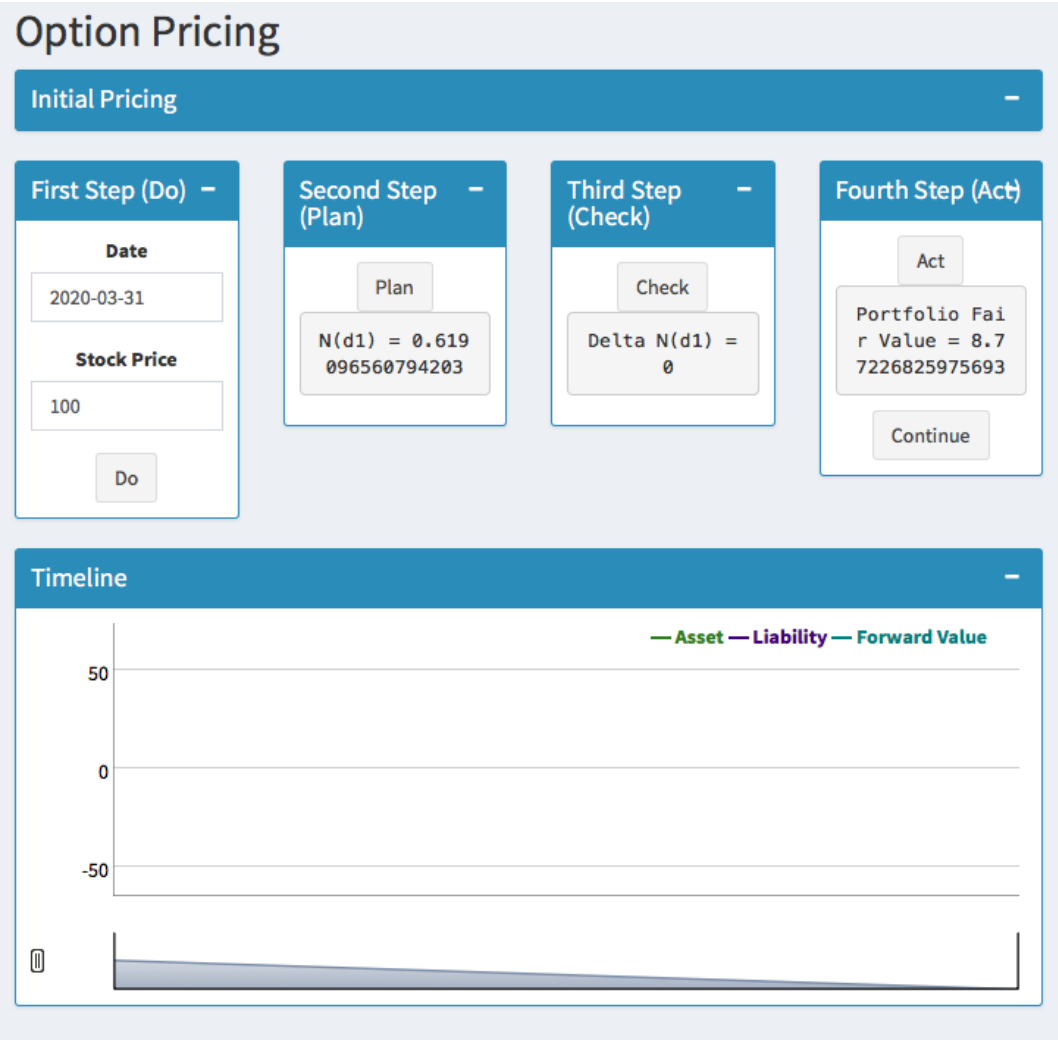

Figure 14: Fourth step (Check).

When the "Act" activity is performed at least twice, for different timestamps, the Timeline graph below shows how the values of Asset, Liability and Forward Price change over these periods of time. Every consecutive time that the "Act" is executed the timeline is extended to show the newly added data.

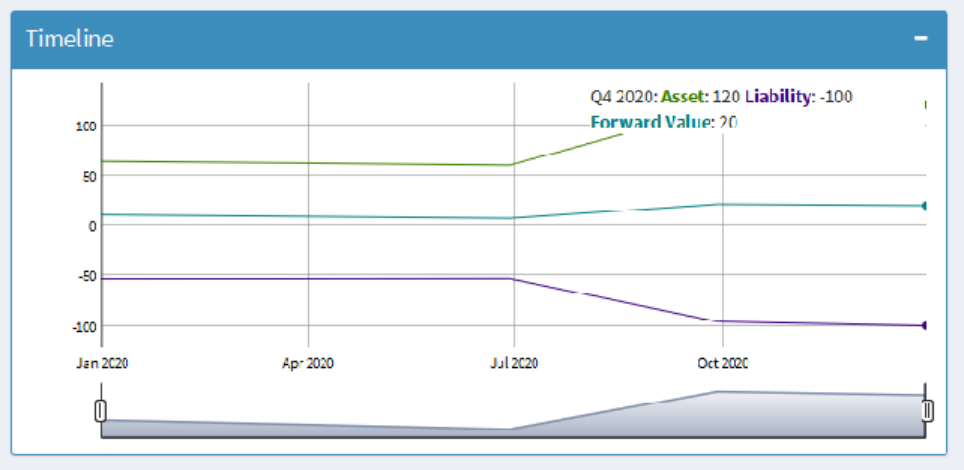

Figure 15: Timeline graph.