

CS261 Requirements Analysis Report

Group 24

Stylianios Maimaris Tim Marshall-Cox Maciej Lulka Folabi Ayonrinde Josh Adams
Lewis Buttle

1 Preface

This document describes the requirements corresponding to the CS261 Software Engineering Project for 2020. This information is targeted towards the client (Deutsche Bank) and the module organiser (Dr. James Archbold).

2 Introduction

A derivative is a contract that derives its value from an underlying entity [1]. Important to note, derivatives are not the obligation, but only the right, to make a purchase, with the involved parties being able to not make a purchase at a later date. Being such a speculative purchase, mistakes can be very costly to the involved parties. As such, the proposed system will provide smart error checking capabilities, so that any incorrect entries or ‘fat finger’ errors are spotted before the trade goes ahead. This prevents situations where the notional value has been incorrectly reported, reduces the risk for involved parties, and removes costly errors, sometimes reaching into the billions. In addition, the software will create the daily report so that Deutsche bank operates at a transparent level about the trade of derivatives and adheres to the EU regulations, of which breaking would constitute breaking the law.

3 Glossary

- **Fat Finger:** Clumsy or inaccurate typing, typically resulting from one finger striking two keys at the same time.
- **Web application:** Is an application program that is stored on a remote server and delivered over the Internet through a browser interface.
- **Web Interface:** Allows for interaction between a user and software running on a Web server.
- **Query:** A request for data or information from a database table or combination of tables.
- **Machine Learning:** A method of data analysis that automates analytical model building.

4 User Requirements Design

The system is to function as a web application which will allow a user to select their desired operation through a menu as shown in Figure 1. The menu will prompt the user to select between creating a new derivative trade, altering an existing trade, deleting an existing trade or producing a daily report. When selecting to create a new trade, the user will be required to fill in the necessary information for said trade. This information will then be processed by the system to detect potential fat-finger errors, mathematical errors or syntax errors and will learn over time from user behaviour. Search functionality is provided for the user when selecting to alter or delete an existing trade. The attributes which must be filled when creating a trade also act as filters when searching for a trade.

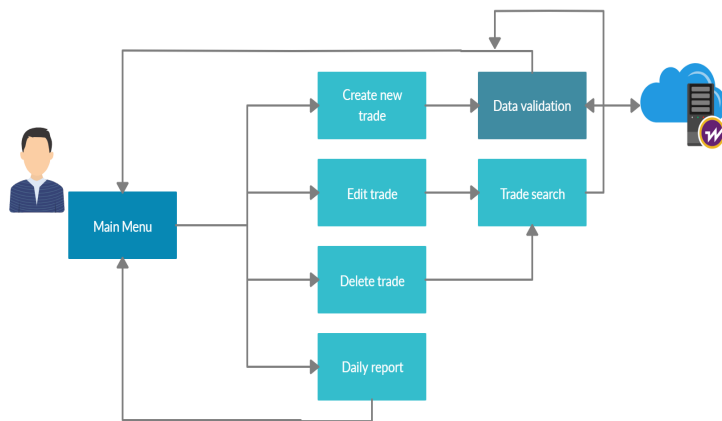


Figure 1: System Functionality from user's perspective

5 System Architecture & System Models

The system will be divided into three smaller subsystems: web interface, back-end processing and database as shown in Figure 2. Each will be developed separately and with care to ensure they will be able to function together to form a complete system.

- The **web interface** will be the only subsystem with which the user will directly interact. It will deal with the inputs and outputs of the entire system. For inputs, it should detect and handle user inputs correctly, such as button presses. For outputs, it should coherently communicate information to the user, such as ensuring clarity of the operation of buttons, or clarity of which information to insert into the input fields.
- The **back-end processing** subsystem will deal with all of the processing the system requires. As such, it will take data from a web form, process it, then pass it on to the database subsystem - and vice versa. It will validate user input by checking for errors using machine learning techniques built using a set of test data, and then act accordingly depending on if the inputted derivative data is valid or not. It will also prepare SQL commands for the database to both insert and query it, for when the user may need to change database data, or if processing requires information from the database.
- The **database** subsystem will act as storage for all valid data that the system will use, such as test data and also valid user inputs. It will need to store data in a coherent and organized fashion to allow for simpler querying, be able to recognize and handle SQL statements from the processing subsystems, and also return the queried data to the back-end processing.

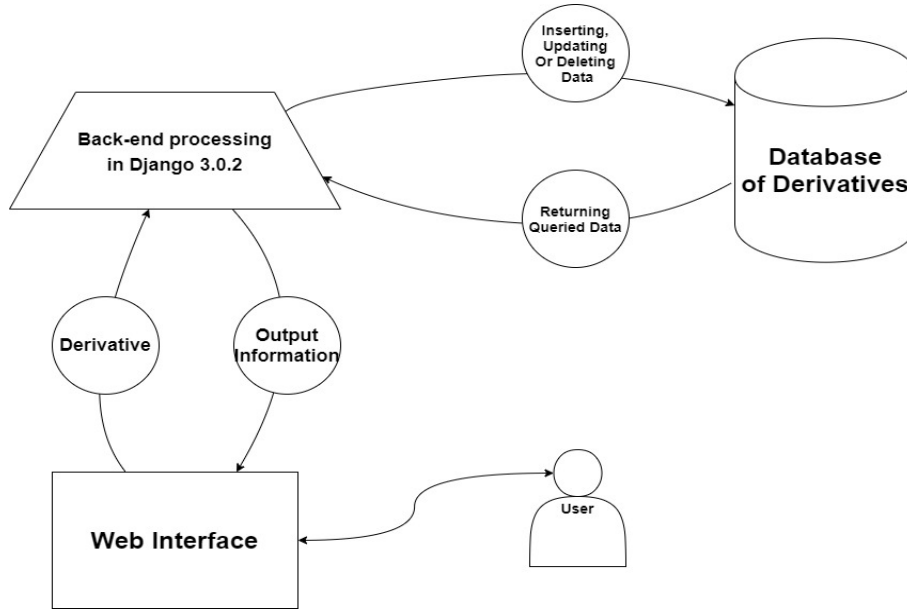


Figure 2: Relationships between system components

6 System Requirements Specification

This section outlines the main system requirements distinguishing between functional and non-functional requirements [2, 3]. Each requirement is presented as a customer facing (**C**) and developer facing (**D**) requirement. Some developer requirements are split into multiple parts to better explain what is required. The MoSCoW method was used in order to classify the importance of each requirement [2].

6.1 Functional Requirements

1. **C.** Must present the user with a menu when entering the web app consisting of five options for creating new derivative trades, updating/deleting existing trades, producing a daily report and updating/adding new companies/products for which trades can be created. **[MUST]**
- D.** The user must be able to access the system's five main functions through a menu by clicking the button corresponding to each function. Each button links to separate web page.

2. C. User must be able to create a new derivative trade. Allow the user to enter the details of the derivative trade(trade ID, date of the trade, the product, the buying and selling parties, the notional amount and currency, the quantity, the maturity date, underlying price and currency, and strike price). **[MUST]**
 - 1D. User must be able to select menu option of creating new trade.
 - 2D. The user must be presented with a web form which will allow them to enter the data detailed in 2C.
 - 3D. If a company or product does not exist, the user must be able to add them through the administrator account.
3. C. User must be able to edit an existing trade. The user must be able to search for a trade from the database and edit it by changing one of its data fields. Trades will **not** be editable after 24 hours from their creation. **[MUST]**
 - 1D. User must be able to select menu option of editing and existing trade.
 - 2D. User must be presented with a search form to allow them to find a trade based on its attributes: trade ID, date of the trade, the product, the buying and selling parties, the notional amount and currency, the quantity, the maturity date, underlying price and currency, and strike price.
 - 3D. The user will be able to edit any of the above data fields.
 - 4D. Trades will **not** be editable after 24 hours from their creation.
4. C. User must be able to delete an existing trade. The user will be able to search for a trade from the database of trades. Trades will **not** be delete-able after 24 hours from their creation. **[MUST]**
 - 1D. User must be able to select menu option of deleting existing trade.
 - 2D. User must be presented with a search form to allow them to find a trade based on its attributes stored in the database for each trade: trade ID, date of the trade, the product, the buying and selling parties, the notional amount and currency, the quantity, the maturity date, underlying price and currency, and strike price.
 - 3D. Trades will **not** be delete-able after 24 hours from their creation.
5. C. User must be able to select option to create a daily report. The system must produce a formatted PDF report detailing all trades of a given day, including information such as the product sold, the buying and selling parties, the quantity and the notional amount. Daily reports will be archived, and the user will be able to search through these and access specific ones. **[MUST]**
 - 1D. User must be able to select menu option of creating daily report.
 - 2D. Must present the user with a PDF file detailing all trades of the selected day.
 - 3D. The trades must be fetched from the database before they are used to create the PDF file to be presented.
 - 4D. The system will allow the user to search reports by their date (will ensure fast access to a daily report when the data on the report is archived and cannot change).
6. C. User input must be validated when creating a new trade.
 - D. System must ensure that the data inputted by the user matches the data type of the data field e.g. price must be numeric and not accept other symbols. **[MUST]**
7. C. The system must provide automatic validation and error correction on the data inputted when creating a new trade. The user must be notified if corrections have taken place and they will have to manually confirm as to whether the trade data that has been altered is correct. **[MUST]**
 - 1D. Use historical/archived trade data (stored in the CSV files) to train a machine learning model to identify potential boundaries for values of a trade.
 - 2D. Frequency of errors (common user errors) must be identified. Common errors must be corrected. Using user feedback, user input and new data the model must adjust its behaviour and way in which error correction is handled.
 - 3D. Utilise newly created trades, if these trades after 24 hours have not been deleted (archived), to train the machine learning component for common user errors and error corrections/pattern.

6.2 Non-functional Requirements

1. C. The system must be suitable for non-technical users. **[MUST]**
 - D. The system must only require basic knowledge of operating a web browser e.g. clicking on buttons to perform actions.

2. C. The system should remain responsive even after extended use and regardless of volume of data being input. [SHOULD]
 D. The system should process a user's action in time linear to the amount of data that needs to be processed before producing a result.
3. C. The system must be intuitive and require little to no training. [MUST]
 D. Navigation of the user interface must be intuitive by following web page standards e.g. consistent themes and fonts on each web page the system provides.
4. C. The system should remain up-to date. [SHOULD]
 D. The system should be easily adaptable to newer versions of software and hardware.
5. C. The system will be validated using appropriate techniques [MUST].
 D. Validation will be performed using industry standard techniques: unit testing, integration testing.

6.3 What the system will not do

- Will not implement any kind of explicit security mechanisms (i.e. authentication methods) [WON'T].
- Will not include the Deutsche Bank branding or corporate identity [WON'T].
- Will not connect to any external systems and can rely on the provided data [WON'T].

7 System Evolution

The system is designed with the following assumptions. The system will be based on an online web page, as such it is assumed a user will have a device with an internet connection and browser software, to access and use the system. The system will utilise some form of database to store derivative trades. Furthermore, the system will be developed and is expected to be used on a Linux based operating system. Moreover, the system may not function on later updates of certain browsers - meaning it is assumed (whilst unlikely it will cause an affect on system availability) that the web browser version the user uses will not change. It is also assumed the user will have a reasonable amount of experience with accessing websites via a web browser, to be able to gain access to the system - this is the only assumption made to the user's technical experience.

8 Group Management

Group roles within the group were decided through discussion and mutual agreement. The role of Project Manager [4] was assigned to Josh. Stylianos will act as Business Analyst [4]. In addition, the group was split into two teams, one responsible for the back-end and database (Tim Marshall-Cox, Maciej Lulka, Lewis Buttle) and one for the front-end (Josh Adams, Stylianos Maimaris, Folabi Ayonrinde). The group decided that any major design/implementation decisions will be made using a voting system. Lastly, group meetings will take place at least once a week in which the project details will be discussed.

References

- [1] J. Archbold, "Group software development project," Jan 2020. [cited 29 January 2020]. Available from: <https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs261/project>.
- [2] J. Archbold, "Topic 3: Requirements analysis," Jan 2020. [cited 29 January 2020]. Available from: <https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs261/14-1.pdf>.
- [3] G. Ahemdabad, "What is a functional requirement? specification, types, examples," Jan 2020. [cited 29 January 2020]. Available from: <https://www.guru99.com/functional-requirement-specification-example.html>.
- [4] J. Archbold, "Topic 3: Early team organisation," Jan 2020. [cited 29 January 2020]. Available from: <https://warwick.ac.uk/fac/sci/dcs/teaching/material/cs261/15-1.pdf>.