Design and Development of an Automated Dashboard for Fund Allocation

State Street Ireland

James O’Donoghue

April 2013
STATE STREET IRELAND
Design and Development of an Automated Dashboard for Fund Allocation

April 2013

Prepared by: James O’Donoghue        Supervisor: Aideen Keaney
Declaration

I declare that the work described in this dissertation has been carried out in full compliance with the ethical research requirements of the School of Computer Science and Statistics.

Signed: ___________________

James O'Donoghue
03/04/13
The aim of this project was to develop a dashboard which could be used to monitor the progress of fund administration in State Street’s Derivative Pricing Team. The purpose of this project was to design and develop a prototype dashboard that would monitor the status of funds and fund allocation within the Derivative Pricing Team. A dashboard was developed as the current system was deemed too labor intensive and prone to error.

The system consists of a front-end user interface, using dynamic web pages and a back-end database that stores the required information. Combinations of programming languages were used to compile this dashboard including HTML, Java Script, SQL, CSS, and Visual Basic etc.
The client for this project was State Street Ireland. Within State Street the primary point of contact was Brian Keelan, Assistant Vice President of the OTC Trade Processing and Derivative Pricing Team.

Throughout the course of the project I developed expertise in a number of software languages and built upon knowledge far beyond what I anticipated at the outset. The project can be deemed a success because it automates a manual system that is currently in place in State Street. The project shows the potential that lies within current software languages to rapidly develop applications that would provide real business value for corporate institutions like State Street.

Although the application needs to be adapted to be implemented within other teams in State Street, it has been demonstrated to a number of managers in State Street who provided positive feedback.

I would like to express my sincere gratitude to Mr. Ronan Ellard and Mr. Brian Keelan for their help and assistance throughout the project.

Finally, I would like to thank my project supervisor, Dr. Aideen Keaney. She provided an invaluable source of information, support and feedback throughout this project which greatly contributed to the completion of this project.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>NO.</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Client Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.4</td>
<td>Terms of Reference</td>
<td>2</td>
</tr>
<tr>
<td>1.5</td>
<td>Report Summary</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>SYSTEM OVERVIEW</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Current System</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Areas of Improvement</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>Purpose of New System</td>
<td>5</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary of New System</td>
<td>6</td>
</tr>
<tr>
<td>3.</td>
<td>DESCRIPTION OF WORK DONE</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>Initial Analysis</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>Problem Definition Phase</td>
<td>9</td>
</tr>
<tr>
<td>3.3</td>
<td>Design of the System</td>
<td>10</td>
</tr>
<tr>
<td>3.4</td>
<td>Design and Development of Database</td>
<td>10</td>
</tr>
<tr>
<td>3.5</td>
<td>Development of User Interface</td>
<td>12</td>
</tr>
<tr>
<td>3.6</td>
<td>Implementation and Testing</td>
<td>17</td>
</tr>
<tr>
<td>3.7</td>
<td>Development Tools</td>
<td>18</td>
</tr>
<tr>
<td>3.8</td>
<td>Familiarisation with software languages</td>
<td>19</td>
</tr>
<tr>
<td>3.9</td>
<td>Problems Encountered</td>
<td>20</td>
</tr>
<tr>
<td>4.</td>
<td>FUTURE EXTENSIONS</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>Conclusions</td>
<td>22</td>
</tr>
<tr>
<td>4.2</td>
<td>System Improvements</td>
<td>22</td>
</tr>
<tr>
<td>4.3</td>
<td>Rolling the System out to other Teams</td>
<td>23</td>
</tr>
<tr>
<td>4.4</td>
<td>Incorporating the System in State Street’s Intranet</td>
<td>24</td>
</tr>
</tbody>
</table>
# APPENDICES

<table>
<thead>
<tr>
<th>NO.</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>Project Outline</td>
<td>A.1</td>
</tr>
<tr>
<td>B.</td>
<td>Interim Report</td>
<td>B.1</td>
</tr>
<tr>
<td>C.</td>
<td>User Manual</td>
<td>C.1</td>
</tr>
<tr>
<td>D.</td>
<td>Design Documentation</td>
<td>D.1</td>
</tr>
<tr>
<td>E.</td>
<td>Data Tables</td>
<td>E.1</td>
</tr>
<tr>
<td>F.</td>
<td>Requirements</td>
<td>F.1</td>
</tr>
<tr>
<td>G.</td>
<td>Screenshots</td>
<td>G.1</td>
</tr>
<tr>
<td>H.</td>
<td>Sample Source Code</td>
<td>H.1</td>
</tr>
<tr>
<td>I.</td>
<td>Test Documentation</td>
<td>I.1</td>
</tr>
<tr>
<td>J.</td>
<td>Technical Manual</td>
<td>J.1</td>
</tr>
</tbody>
</table>

Glossary of Terms Used

References
1. INTRODUCTION AND SUMMARY

This chapter contains the introduction and background to this project. It also discusses the objectives of the project as well as the agreed terms of reference. Finally, it will outline the remaining sections of the report.

1.1. Client Background

State Street is a leading provider of financial services to institutional investors worldwide. Today, with over 2,000 employees in Ireland, they offer investors a complete range of financial services across the investment spectrum, including investment servicing, investment research and trading, and investment management.

The Client for this project was the Derivative Pricing Team (DPT) within the Market Data group. The DPT is responsible for analysing and processing the changes in stock prices for funds across a number of portfolios. They provide feedback from their analysis to other departments within the organisation.

1.2. Background

At present, the derivatives team has a large number of funds that need to be processed over the course of any given day. Currently, these funds are allocated to Fund Administrators via an Excel spreadsheet and when completed, are signed off by a Manager. When that process is completed the Fund Administrator must then sign the fund off manually on a Lotus Notes database. Once a fund has been allocated, the onus is on the Fund Administrator to memorise his/her funds for the given day and the various deadlines for each of their funds.

Once a fund has been administered and signed off by a Manager, the Lotus Notes database is manually updated to inform other teams that the DPT have completed their analysis. The current system does not allow for management reports such as the progress of the team. Similarly there is no process by which team members can individually examine their tasks for the upcoming day. Finally, there is currently no functionality in place to examine the number of missed deadlines in a given day, something that management has expressed concern about.

State Street would like that these tasks be automated and the results presented on a web front-end, which would be easily accessible by all members of the team – from administrative staff to managers and executives. The automation of this process should reduce the manual effort that is required to complete the current manual updates of the system. As a result this should reduce the number of errors that occur during the course of the day.
1.3. Objectives

The development of a dashboard should automate the processes mentioned above. The new system will provide visual representation in the form of tables to each of the team members for their daily task list. This should improve the efficiency of the team and reduce errors or discrepancies. Similarly, an automated system will provide the basis for future extensions.

The objective of the dashboard is to give management a real time view of their team’s progress at any given time. The system also provides management the functionality to make alterations in the team’s schedule to allow for unplanned events such as staff absences or an unexpected issue arising with a client.

1.4. Terms of Reference

- To investigate and document the problem through discussions with the client.
- To research possible solutions. The solution should allow historical data to be stored in a readily accessible database.
- To design and develop a dashboard that will:
  - Allow staff members to easily manage their assigned funds.
  - Provide managers with a view of funds progress.
  - Allow management to make adjustments easily to fund information.
  - Output results in a user friendly format.
- To produce both technical and user documentation.

1.5. Report Summary

This report will outline the design and development of the dashboard. It will examine areas for improvement within the current system and the possible extensions that could be applied to the proposed system.

- **Chapter 2:** Presents an overview of the current system and highlights areas that the new system will address as well as providing a description of the structure of the new system.
- **Chapter 3:** Contains a description of the work done over the course of the project.
- **Chapter 4:** Contains conclusions drawn from the project as well as recommendations for further development of the system.
2. **SYSTEM OVERVIEW**

This chapter outlines the current system that is in place within the DPT. It then examines the areas of improvement that need to be addressed in the new system. Finally, an overview of the new system is provided.

2.1. **Current System**

The current system is reliant on manual procedures and as a result, discrepancies and errors occur from time to time. The weekly schedule is constructed by one of the Managers in an Excel spreadsheet which is then stored on the team’s server and can be accessed by the entire team. Following this an email is circulated to inform the Fund Administrators that the spreadsheet has been updated and that they should examine their tasks for the week ahead.

The onus then lies on the Fund Administrator to memorise his or her funds for the upcoming week and the deadlines and other details that accompany each fund. There is no breakdown as to the assignment of funds per employee and as a result each Fund Administrator is faced with a large spreadsheet from which they must extract their funds. Occasionally, this results in certain details being missed regarding funds for the week which can lead to missed deadlines.

Once a fund has been completed by a Fund Administrator, there is a requirement for a sign off under the four eye principle of error checking, usually performed by a manager. Once the fund has been signed off, the Fund Administrator must then manually update the Lotus Notes database to inform other teams in the organisation that the fund is available for further analysis. The Fund Administrator must input a “completed time” in the Lotus Notes database, this information is then used for the team’s monthly statistical reports. This will be discussed further in Section 2.2. A high level overview of the current system can be seen in Figure 2.1.1 below.
2.2. Areas of Improvement

There are a number of areas which could be improved in the current system, some of which management have expressed a specific interest in being addressed. One of the key reasons that administrators miss deadlines is due to the fact that team members are expected to memorise their daily schedule and accompanying deadlines. To produce a table of tasks for each Fund Administrator would greatly improve efficiency and reduce the risk of missed deadlines. It will also allow Fund Administrators to focus on the completion of their funds and other tasks such as replying to any queries as opposed to memorising their schedule for the upcoming day.

From a managerial perspective, there is currently no scope to access the schedule and make alterations during the course of the day, if an unforeseen event was to occur. This functionality will be implemented in the new system and it is hoped that this will allow for a smoother running of the team’s daily tasks. This is explained in further detail in section 3.

Similarly, there is currently no opportunity for Fund Administrators to warn Managers that deadlines may be missed for various reasons. This can often lead to missed deadlines that could easily have been prevented if a Manager knew an issue existed. With the knowledge that a Fund Administrator would not be in danger of not completing the particular fund, the Manager could transfer ownership of that fund to a fellow Fund Administrator who could have completed the fund on time.
Recently, Senior Management in State Street has placed an onus on each team Manager across the organisation to present monthly statistical reports as to the efficiency of their respective team. With the current system, compiling these reports can be both tedious and time consuming and the manual time recording of completion can lead to inaccuracies in the data. With a new automated system, it is considered that these reports will now embody a more accurate reflection of the team's productivity.

The current fund allocation sheet is based on a weekly schedule. There is no further breakdown as to the allocation of funds. This system works adequately if the funds are equally allocated on each day because team members will assume that the funds that they were to complete on Monday will be the same for the remainder of the week. This is not always the case as changes can occur to the schedule with sick/holiday leave or unplanned changes in the team’s routine. As a result, a new method of displaying the fund allocation per team member per day will be installed under the new system. This will eliminate any excess information and provide the Fund Administrator with a quick easy to understand description of their tasks for the upcoming day.

2.3. Purpose of New System

The purpose of the new system is to automate the process of fund administration status reporting. This will in turn result in a number of benefits, including:

- Simple to use interface for end-users;
- The Excel sheet containing the schedule will still be compiled in the same manner;
- Reduced risk of human error resulting from a reliance on team members memorising their tasks;
- Data presented is up-to-date as it is updated every 5 minutes;
- Managers will be able to get a real time view of the team’s progress at any given time;
- Flexibility for management to make changes in the schedule in the event of unplanned circumstances;
- Summary statistics are readily available in a table. These represent progress in daily tasks;

The front end will be designed to be user friendly and intuitive to use. Unfortunately, the Lotus Notes system that the team currently uses does not provide the necessary functionality to accommodate these changes. Consequently the back-end of the system had to be constructed from scratch. A number of options as to the method for design were explored but a combination of PHP, HTML, CSS and MySQL was believed to be the most effective method to constructing this unique system. This will be discussed in more detail in section 3.4 and 3.5.

This enhanced system will integrate with the team’s current data entry system which is Microsoft Excel. It was noted by the client at the outset that the scheduling of the task
allocation would remain in Excel. However, the presentation of the scheduling was intended to be displayed in a separate software package. As a result, the new system will interact with Excel by extracting the weekly schedule and using it as the basis for the information contained within the dashboard. This is an automated process and will therefore not cause any disruption to the Derivatives team.

2.4. Summary of New System

Due to audit control and regulatory constraints the architecture of the new system will remain relatively static. The new system however, will be based on a dashboard which will interact with a back end database. A dashboard is an easy to read, real time user interface that displays a snapshot of the current status of an organisation’s or team’s key performance indicators and as a result enables instant and informed decisions to be made.

The system as a whole can be broken down into two stages. The data gathering and manipulation stage and the data dissemination & display stage. With the new system the Excel spreadsheet is still updated as in the previous system. However, the Manager will now click a button on the spreadsheet which will then manipulate the data into a usable format and automatically save it to a file location that the back-end database can access when the Import function is run from the Manager’s dashboard. This import function will be explained in greater detail in Section 3.4.

Once this has taken place, the Fund Administrators will be able to log onto their dashboard and view their upcoming funds and respective deadlines. Similarly, the managers will be able to log onto their respective dashboard to analyse their team’s progress for the day and to track the completion rates for the team’s funds. The manager’s main menu can be seen below in Fig 2.4.1. This shows the functions he/she can avail of and also displays some summary statistics as to the progress of their team at a given time.
Gathering the Data:

There are no changes to the format of the existing Excel Sheet. This should allow for a smooth transition into the team’s current daily routine because there are no changes to the processes that are currently in place. The new system gathers the relevant information from the Excel spreadsheet. Before the information can be input into the database, some data manipulation has to be carried out. This data manipulation is automated using Visual Basic Script. Once the data has been manipulated into a usable format it is then imported into the database using a PHP script that can be run from the Manager’s dashboard.

Interpreting the Data:

The dashboard is created using PHP and HTML, it then runs queries on the MySQL database using SQL. The data that these queries produce is then displayed in the form of tables which are manipulated to display information relevant to the user. The dashboard recognises when a certain user, Fund Administrator or Manager, is logged in and therefore modifies the results accordingly. For instance Figure 2.4.2 below shows the daily tasks for a given Fund Administrator and also displays the status of the funds in each case. The ‘Flag’ variable in Figure 2.4.2 represents the current status of a fund. A ‘Green’ flag indicates that a fund has been complete. A ‘Purple’ flag indicates that a fund is due in 30 minutes. A ‘Red’ flag means that it is due in 10 minutes and a ‘Black’ flag means that a fund has missed its deadline. This is further explained by the legend that can be seen in Figure 2.4.2 below.
Access to Database via the Dashboard:

The users of the database can be split into two different user groups, Fund Administrators and Managers. Fund Administrators have the functionality to read information from the database and display the information onto their screen. They have also the ability to update the status of their funds in the database. Managers on the other hand are given the additional functionality of adding information such as new funds to the database and editing existing records in the database such as changing the deadline time of an existing fund.

The system has the ability to recognise whether a user is a Fund Administrator or a Manager by the information they provide at the log-in screen. This adds a layer of security to the system as administrative staff cannot access the functionalities that are exclusive to management such as editing and adding records to the database as well as transferring ownership of funds.

Using the Application:

In order to use the application, the user must enter the URL of the dashboard’s homepage into their web browser and log in using the details provided to them by management. Appendix C.1 contains a detailed user manual with further explanation on how to use the system.
3. **DESCRIPTION OF WORK DONE**

This chapter contains a description of the work done and the systems methodology used when creating the online fund management system. It describes the initial analysis of the current system. It also examines the process by which the work was planned from the outset, the proposed design and functionality of the new system and the various types of software used to create the new system.

3.1. **Initial Analysis**

The initial analysis consisted of setting out the various tasks that would be involved with this project. This initial analysis resulted in the project being split into three main areas:

(i) Problem definition phase;
(ii) A design and development phase;
(iii) An implementation phase;

3.2. **Problem Definition Phase**

Although the outline of the project gave a good indication as to what was involved, a formal definition needed to be fashioned in order to eliminate the risk of misunderstandings regarding the deliverables of the project. This formal definition was composed after a number of interviews with the client. This resulted in a list of requirements.

**Interviews with the client:**

During the first meeting with the clients, Mr Brian Keelan and Mr Ronan Ellard, an initial set of requirements was created. These potential ideas and requirements and the feasibility of each were then discussed with the project supervisor, Dr Aideen Keaney. Further meetings were arranged with the client to discuss in detail the feasibility of the initial requirements and the possibility of additional features that could potentially be incorporated into the dashboard.

It was agreed that one of the main requirement of the database was as little interaction between the Fund Administrator and the system as possible. It was considered that with an increasing amount of interaction there would be an increasing level of rejection of the system. This would only add to the Fund Administrator's workload as opposed to reducing it. A clear and simple layout and easy navigation throughout the system was also viewed as a priority. A full list of requirements can be seen in Appendix F.
3.3. Design of the System

Once the requirements of the project were decided upon, the next step was to design how the system might work. Certain components of the fund administration process will remain manual due to audit and regulatory controls. The functionality of the dashboard was decided upon during the interviews with the client. This meant that the main areas of design for the project were the user interface and the data structure of the back-end database.

3.4. Design and Development of Database

In order to get a clearer understanding of the structure of the database, entity relationship diagrams were drawn. These diagrams proved beneficial and provided a clearer picture of the interactions between the various tables that existed within the database. Once the nuances of the relationships became clear, the structure of the tables was constructed using MySQL and phpMyAdmin. The dashboard would not create any additional tables. It would simply populate existing tables, transfer records from one table to another, query information from the tables or edit information in existing tables. An overall entity relationship diagram of the system was produced, see Figure 3.4.1.

![Entity Relationship diagram of the new System](image)

As you can see from Figure 3.4.1, the Managers table has control over the entire system whereas the Fund Administrator/Employees table can only interact with the day tables. Even
at that, the Fund Administrator can only add to information that exists within the day tables, it cannot edit or modify the records in those tables.

The Type of tables:

The system uses a combination of both static and dynamic tables. This was decided during the initial analysis with the client as it was believed that it would allow the system to hold information that would not need to be accessed except for unusual circumstances. For example, the Managers and Employees tables are both static tables and simply store the log-in details for the respective company employee. There is an additional field marked “Initial” in the Employees table and this was used throughout the system as a session variable that would enable the dashboard to recognise the current user and adapt the results of their queries accordingly.

Populating the tables:

The Manager still creates the Excel spreadsheet for the upcoming weekly schedule. He/she will then carry out data manipulation and export the manipulated spreadsheet to a file location to be chosen by the client. This process will be automated and will be carried out with the click of a button from the original spreadsheet. This is carried out using Visual Basic Script. The code to which can be seen in Appendix G.1.

The Manager then imports that file into the MasterData table from their dashboard. The MasterData table will contain the list of funds to be completed for the upcoming week and the allocated Fund Administrator for each. From a combination of the Funds table and the MasterData table, the 5 separate day tables are created. This is completed as a background operation when the Manager decides to import the upcoming weekly schedule from their dashboard. This import will be carried out using PHP Script. The code to which can be seen in Appendix G.2.

Once the week is completed, the Manager then has the ability to “Archive” the week’s schedule. At which point, the information that resides in each of the days tables is transferred to the Archive table. The remaining variable that resides in the day tables that has yet to be accounted for is the “Flag” variable. This is used to distinguish the progress of each fund and will be the key variable during the creation of the summary statistics. This field can be populated in two ways. The first of which occurs if the Fund Administrator confirms the completion of the fund from his/her dashboard which then triggers an update to the “Flag” value. The other possibility is the systems update function, which will be explained in greater detail in Section 3.7. These processes were made possible with the use of PHP script, see Appendix G.3.

Similarly, the Funds table stores a list of all the team’s clients and funds that must be processed as well as other pieces of information that accompanies the fund. This information includes variables such as deadline time and frequency, which refers to the regularity with which the fund must be administered. This information is then copied into each day’s table
from which the “Owner” of the fund is assigned. This value depends on how the funds have been allocated in the MasterData table. This process is carried out with the use of PHP script.

The remaining information in the tables is created by the Manager. He/she has the ability to add records to the static tables as well as the capability to edit information in the dynamic pages as they see fit. Figure 3.4.2 displays a screenshot of the “Edit Fund Menu” that can only be accessed by a Manager. He/she can make adjustments to the systems back-end database, authority that the Fund Administrators do not have. These functions were created using PHP script which can be seen in greater detail in Appendix G.3.

3.5. Development of User Interface

The development of the front-end user interface required a combination of various programming languages. It was also essential to predetermine the style and the flow of the interface, both of which are key components to good software design, (Shneiderman, 1998). As previously discussed, during the initial discussions with the client, certain characteristics of the dashboard were agreed upon such as user friendliness and minimal interaction with the dashboard.

As a result of these discussions, a system that minimised disruption to the current work day was seen as the optimal solution. The proposal was that the user could log onto their dashboard in the morning through their preferred browser. The dashboard will sit in the background while the Fund Administrator carries out his/her daily tasks. If the Fund Administrator needs a reminder of his/her funds they can simply check their screen and
within one or two clicks of the mouse have the desired information and carry on with their daily routine.

System Update:

One of the integral parts of this system is the update function that runs in the background which continually updates the status of the funds. This was created using UNIX script that runs from Monday to Friday and calls a PHP script periodically every 5 minutes. Although it has the flexibility to be updated at other intervals should this be required. The script for this function can be found in Appendix G.3.

This PHP script retrieves the current time and compares it to the funds deadline time. If the funds deadline is due within 30 minutes, the status or “Flag” of the fund will turn Purple, if due within 10 minutes it will turn Red and if passed its deadline it will turn Black. It is important to note, that at any stage if the Fund Administrator specifies that the fund has been complete the flag turns Green and the update function would no longer change its value.

If however, the fund has already passed its deadline and the Fund Administrator tries to update the status of the fund, he/she will be met with a warning message that their fund has already passed its deadline and therefore its status cannot be changed. This validation is included in the system so that when monthly statistical reports are compiled the information extracted from the database will be more accurate and precise. These reports will be based on the number of funds that missed their deadlines. If the user had the ability to alter the status of the fund even after a deadline had been missed, when these reports are assembled there is no way of knowing if the fund was in fact completed on time.

Dashboard Appearance:

With the requirements agreed, a suitable appearance of the dashboard was necessary. The appearance had to resemble a professional site and the navigation had to be intuitive to coincide with prior objectives of user friendliness being a priority. Each page had to be consistent with the next; for this reason a template was required. This template consisted of attributes such as background color, logo, and the placement of various aspects of the pages such as tables and summary statistics.

This template was developed using CSS in Microsoft Web Expressions 4. A key requirement was that the user only needed to glance at the page to quickly assess his/her situation. For this reason, the dashboard was broken into a number of different sections so that once the dashboard is integrated within the Derivative team and Fund Administrators and Managers become accustomed to its appearance, deciphering information from its screens will be instantaneous.

Each page can be broken down into the different sections of the template. The header contains the logo of the company which was one of the few design features that the client
requested for the dashboard. The main content of each page exists in the middle which is referred to as the body.

The body acts as a wrapper for the entire dashboard and encapsulates a number of subsections within the dashboard, such as the statistics, output, contents and buttons subsection. The background of this section is white and the text is black. The text is Arial and its size varies from 18 point to 30 point depending on the type of text from a heading to normal paragraph text, all with 2.0 line spacing. The aim of this is to ensure that every visitor can read the text displayed on the site.

The ‘statistics’ subsection refers to area on the dashboard that displays the current summary statistics to the user. These statistics are manipulated to accommodate the user. For example a Fund Administrator will only be concerned with his/her upcoming funds and statistics surrounding them. The Manager however, will be concerned with the entire team and therefore the section will be altered accordingly. This is facilitated by the “session variables” that was previously mentioned in section 3.4, which enables the dashboard to recognise the user and adapt its display accordingly.

The ‘output’ subsection refers to the area where results of queries run by the user are displayed. This area has been specially formatted to display the results in a user friendly format, primarily in tables. The background for this section is white, the text is black and the borders of the tables are blue in accordance with the State Street theme.

The ‘contents’ subsection is very similar to that of the output subsection except that the contents section refers to static information. Regardless of the events the information displayed will remain the same, whereas the content of the output section will change from time to time depending on the specific situation. It is formatted in the same way with a background colour set at white and the text set to Arial and black in colour.

The ‘buttons’ subsection refers to the navigation buttons that are displayed at the bottom of each HTML page and displayed above the output tables produced from MySQL queries. They have also been aligned at the centre of the page so that they should not be missed by the user. They have been coloured blue to maintain the state street theme. Figure 3.5.1 below illustrates the various sections that can be seen throughout the dashboard.
Figure 3.5.1 – A breakdown of the different sections throughout the dashboard

**HTML/ PHP Web Pages:**

Microsoft Web Expression 4 was the platform used to create the various pages that form the dashboard. These consist of a mixture of HTML and PHP pages. These pages form the basis of the user interface and accommodate the connection between the front-end design and back-end database. Essentially these pages form the basis of the dashboard and carry out the user commands in the background.

PHP files are just like HTML files, but they can include both HTML and PHP code. The PHP code is executed by the web server when the page is accessed and the resulting output is written as HTML within the web page. PHP allows dynamic content to be generated each time the Web page is loaded, based on the variables included in the PHP code. It is for this reason that the PHP pages were primarily used to connect with the back-end database and to run the SQL queries, while the HTML pages were primarily used to display lists of functions for the user to choose from. Examples of these scripts can be seen in Appendix G.

**Error Checking:**

There are a number of validations included throughout the system as well as navigation buttons that should minimise the risk of human error within the system. Figure 3.5.2 provides an example of one of the validations that exist within the system. A simple prompt to ensure that the user is happy with his/her decision to log-out. This functionality is also included if the Manager attempts to import next week’s schedule without first archiving the previous week’s schedule. This control was put in place because the import function is designed to erase the previous week’s schedule and then import the upcoming schedule. Therefore valuable
Information would be lost if the Manager were to import the impending schedule without first archiving the week’s task schedule. The script for these confirmation pages can be found in Appendix G.5.

Navigation buttons are also included at every screen for the dashboard. This is to minimise the risk of human error and ensure the system user can easily navigate through the dashboard. Similarly, the system will return the results of any changes to the system, this could come in the form of a Manager editing fund information or a Fund Administrator simply completing his/her fund. These confirmation messages are displayed with the intention of re-iterating to the user what changes have been made within the system.

![Figure 3.5.2 – An example of the systems confirmation pages](image)

Data Validation:

An additional integrity measure incorporated in the new system is data validation. This was primarily implemented throughout the dashboard so that any data inputted by the user had to be of a usable format and comply with certain characteristics. For example in the “Add Fund” and “Edit Fund Details” section of the dashboard, the user must enter the ‘Deadline’ of the fund in a specified time format. If the user does not input a time format the system will return an error. The code for this validation can be further examined in Appendix G.5.

This data validation ensures that the system works on clean, correct and useful data. Figure 3.5.3 below is an example of how the system deals with data being entered that does not comply with the validation rules set out for that particular piece of information.
3.6. Implementation and Testing

Implementation:

In order for the system to be implemented within the State Street systems, connection must be established with a web hosting site such as Blacknight Solutions. A root folder must also be set up so that the files containing the code can be stored and then accessed by the server. Once this has been complete the files need to be copied into this root folder, this will then allow the dashboard to become operational within State Street. The files needed for this site are included on the accompanying CD.

Design Methodology:

For the User Interface, it was decided that the most proficient form of development would be to implement an incremental model. An example of an incremental model can be seen in Appendix D.1.1. This meant that at regular intervals during the development process the system would be tested to ensure that work done to that point was robust. The queries run by the dashboard were retrieving the correct information from the database and displaying the results in the correct format with no errors or exceptions. This methodology was seen as the most effective form of development because it allowed for small gains to be made at regular intervals with safety in the knowledge that the system was working correctly.

The alternative to this would have been to continually develop the system without testing and once the system had been fully coded and designed, to carry out extensive testing at that point. This practice carries with it a number of inherent risks, the most serious of which...
would be, that once the system had been complete from a design perspective, there could be a minor error in the code and no clear understanding as to where the error came from. When dealing with a system that integrates a number of applications and programming languages it was considered that this risk would be amplified and therefore the iterative testing model was chosen.

**UNIX Code**

Prior to the creation of the dashboard, I had no previous experience with UNIX code and its uses. Having carried out research on the design and development of dashboards through online tutorials, it became clear that there would be a need for UNIX code to be used within the system. This code would allow the system to continually update itself at intervals specified by the system creator. For the purpose of this system, the update would occur every 5 minutes from Monday to Friday.

Initially this update was set to occur every minute however this turned out to be a major use of computing power and was therefore limited to every 5 minutes. This would not have major implications for the system due to the fact that the majority of deadlines occur on either the half hour or on the hour. An additional limit was included so that the update would only occur between 6am and 8pm. This meant that the system would not be working needlessly throughout the night.

### 3.7. Development Tools

Throughout this project, a number of different software tools were used to create this new system. The tables below list all of the software tools, applications and languages that were used and provide a brief description of their use within the project.

The table has been broken down into two sections, the software package and the programming languages that were used in the construction of the dashboard.

<table>
<thead>
<tr>
<th>Programming Language</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Visual Basic Script</td>
<td>• This was used to manipulate the Excel spreadsheet created by the manager into a usable format so that it could be imported into the database.</td>
</tr>
<tr>
<td>2. CSS</td>
<td>• Used to format the HTML pages and to ensure a continuous structure existed within the dashboard.</td>
</tr>
<tr>
<td>3. PHP</td>
<td>• Used to connect the front-end design to the database and to query the database</td>
</tr>
</tbody>
</table>
4. MySQL  • Used to set up the database.

5. HTML  • Used to create the design of the front-end system which acts as the user interface of the dashboard.

6. Java Script  • Used in the dashboard to create error checking functions in the dashboard.

7. Unix Script  • Used to create the Cronjob that continually updates the database and therefore the dashboard.

<table>
<thead>
<tr>
<th>Software Tool/Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Microsoft Excel 2010</td>
<td>• Used to create the visual basic script to manipulate the data into a usable format.</td>
</tr>
<tr>
<td>2. Microsoft Expression Web 4</td>
<td>• Used to develop the front end design of the dashboard. It was also used to create the interactions between the front-end design and the database.</td>
</tr>
<tr>
<td>3. phpMyAdmin</td>
<td>• Used to set up the database design.</td>
</tr>
<tr>
<td>4. Putty/SHS Telnet Configuration</td>
<td>• Used to create the Cronjob that updates the dashboard periodically.</td>
</tr>
</tbody>
</table>

Table 3.7.1 – List of the programming languages used and their contribution to the dashboard

Table 3.7.2 – List of development tools used and their description

3.8. Familiarisation with software languages

I had only undergone introductory courses in a number of the technologies which would be involved in the project. As a result, a large amount of research was necessary to familiarise myself with the nuances of the technologies and the functionalities that they could offer the new system. Without this additional research and knowledge development, it is doubtful that the project could have succeeded.
Database Introduction:

Due to my inexperience in terms of database design I researched the area and discovered the functionalities that could be exploited. My research proved fruitful and gave me realistic expectations of what to expect from the project. It also provided me with knowledge that I could then pass on to the client during our meetings in terms of the possible functionality of the database. This research was carried out using material that had been covered throughout the college course as well as PHP and MySQL Web Development (Welling & Thompson, 2008), which provided an invaluable source of information during the course of the project.

Dashboard Introduction:

Having undergone introductory courses in HTML and PHP during my college career I had some understanding as to the basics of these languages. For the purpose of the project a greater level of expertise would be required. I built upon my introductory knowledge of dashboards through reading PHP and MySQL Web Development (Welling & Thompson, 2008). This however was not sufficient to encapsulate the complexity of the proposed dashboard. Consequently, I completed a number of online tutorials that provided a better understanding of the languages and enabled me to create basic dashboards. The online tutorials built upon my existing knowledge particularly from a practical standpoint. This was beneficial to the project.

CSS Programming:

During the design process for the front-end dashboard, it became apparent that CSS would be integral to creating a design template and ensuring that this template was applied to the entire dashboard. After some online research, the language became far easier to implement into the design of the system and it allowed for a far more professional appearance. A link to these online tutorials can be found in the References section.

3.9. Problems Encountered

During the development of the dashboard a number of issues were encountered.

Access to State Street Servers:

It proved difficult to gain external access to the State Street servers. State Street, as with any large multinational company, employs a particularly strict IT policy. Therefore I was unable to gain access to their servers throughout the development of the dashboard. As a result the dashboard was created using the School of Computer Science and Statistics’ server in Trinity College, with the intention of transferring to State Street’s servers once access was granted.
This meant that any queries that arose during the development of the dashboard were unable to be viewed by the client in their offices. Instead, emails were exchanged and a description of the issue was provided, from which the client was expected to provide appropriate guidance as to the solution of the problem. Although this did not have a detrimental effect on the project, it did come with a level of frustration as it could take up to a week for an issue to be resolved.

**Systems Integration:**

The proposed system deals with a number of different software applications. Creating connections between these applications initially proved troublesome, in particular ensuring that the UNIX script would interact with the back-end database. Similarly, the college server was initially not set up correctly to allow the system’s database to interact with Microsoft Excel. This caused a delay in the development of a number of functionalities within the system.
4. **FUTURE EXTENSIONS**

This chapter describes recommendations for future development of the dashboard system as well as conclusions drawn from the project.

4.1. **Conclusions**

The most significant parts of this project were to identify the client’s requirements and to design the new system based on these requirements. The design and development of the new system was the most time consuming aspect of the project. However it was essential to be vigilant as to ensure that the end result completely satisfied the requirements. The requirements were identified through meetings with the client and the project supervisor. The success of the project also required researching content and analysing features provided by other example dashboard that were found online.

A fully operational and professionally designed dashboard was created and designed which fulfilled all of the client’s requirements, and currently resides within the school of Computer Science and Statistics’ server at [http://macneill.scss.tcd.ie/~jodonog/FYP/fyp1.html](http://macneill.scss.tcd.ie/~jodonog/FYP/fyp1.html).

The terms of reference that were agreed with the client were all completed successfully. The terms of reference for the project are provided in section 1.4.

This project has been challenging technically, creatively and academically. It has encompassed a wide range of the material covered in the Management Science and Information System Studies course and provided an education in website application development and implementation.

4.2. **System Improvements**

Some further works are recommended in order to make the system more reliable and robust.

*Modification 1: Backup Database*

One of the main refinements that should be considered for this system to improve the current architecture is to add a back-up database to the server which would contain a replica of the database proposed in this report. This back-up would act as a contingency allowing the dashboard to interact with it should the original database be down or unavailable for some reason.

*Modification 2: Adding Commenting Functionality*

The addition of a commenting function would also greatly enhance the effectiveness of the dashboard. This would allow Fund Administrators to attach comments to the fund for a
variety of reasons such as an explanation as to why a fund may be delayed or comments directed at other teams that further investigation into certain aspects of the fund may be necessary. It is believed that this would greatly enhance the effectiveness of the dashboard and its addition should be strongly considered. It would also enhance the auditing of issues relating to the Funds Administration by providing an integral real time audit trail. This function could be carried out in the "Finish Fund" screen. Commenting functionality would also need to be implemented into the database to allow for this modification.

**Modification 3: Improving Emergency Section**

Another feature that could be modified to enhance its current functionality is the Emergency section in the dashboard. Currently the dashboard has the functionality to simply email a Manager a predetermined generic message that will inform him/her that there is an issue. The Manager must then investigate the issue further and decide on the best course of action. If this functionality were to be developed to provide more detailed information to the Manager and provide him/her the option to address the issue with one click. This enhanced ability would result in speedier response times to issues, which would in turn reduce the risk of deadlines being missed. This could be carried out with the use of PHP script. The PHP script could add a message to the email that the Manager receives, once clicked the code will automatically assign the fund to a Fund Administrator who does not have a fund due in the near future.

**Modification 4: Handling of the Archive Table**

The current system is set up so that the weekly schedule is saved to an *Archive* table at the end of each week. This *Archive* file can then be used to carry out statistical analysis of the team’s performance over previous time periods. At some point the *Archive* file will reach its capacity and will no longer be able to be used to store the previous week’s schedule. Therefore, it is advisable that State Street devise a strategy to deal with such a circumstance. Be it, erasing the *Archive* table periodically once the statistical reports have been generated or else finding a way to store the information in a separate format that would allow for a greater amount of data to be. This could be achieved by transferring the information from the archive table into a separate static table periodically, for example every 6 months.

### 4.3. Rolling the System out to other Teams

It is recommended that this system be tested for more teams within State Street with the objective of establishing the system as the standard fund administration tool across the organisation.

With this in mind a number of requirements must be considered as to standardise the process of fund administration across all teams, as well as standardising how data will be displayed on the dashboard. This should standardise how the data will be stored in the
database tables. This unified approach will reduce the risk of inconsistency when integrating the new system into new teams.

Another approach may be to edit the current system to accommodate different needs in different teams. This system deals with fund administration which may not be applicable to other teams within the organisation. Nevertheless, the functionality of the system may still be deemed useful with certain alterations. In such a situation, the system could be copied and with a little refinement could become far more pertinent to some teams that may not be involved in fund administration. This could be done by modifying the tables to handle information more appropriate to the team.

These propositions will more than likely result in changes being made to the system that has been offered in this report. The system does however provide a sound basis for future development.

4.4. Incorporating the System in State Street's Intranet

It would be advisable to integrate this system into State Street's Intranet which would then eliminate the need for the user to have separate login details for the Intranet and for the dashboard. This has already been incorporated into a number of applications which are in use across the organisation.
APPENDIX A
Original Project Outline

Client: State Street Corporation
Project: Developing a Web based dashboard
Location: Sir John Rogerson’s Quay, Dublin 2
Client Contact: Ronan Ellard, Vice-President, r.ellard@statestreet.com

Client Background

State Street is a leading provider of financial services to institutional investors worldwide. Since its entry into the European market in 1970, State Street has built a strong presence in the region to better serve their clients. Today, with over 2,000 employees in Ireland, they offer local investors a complete range of financial services across the investment spectrum, including investment servicing, investment research and trading, and investment management. State Street in Ireland services more than US$550 billion in assets for its clients — making it one of Ireland’s leading fund services companies. For more information see http://statestreet.com

Project Background

We have a large number of sign-offs over the course of any given day, incorporating pricing and associated sign-offs as well as trade processing (+ sign-offs). At the moment, these tasks are allocated to team members via an Excel spreadsheet and signed off on a Lotus notes database. There are a number of issues with the process:

- Task allocation via Excel is an onerous process.
- Only tasks are allocated, not sign-offs.
- Once a pack is signed off, the Lotus notes database is updated to let other teams know pricing is complete. Files are attached with each sign-off.
- Reports can be run from Lotus for statistics, but this output is very poor and needs to be manipulated.
- There is no warning process showing a user what deadlines are approaching.
- Tasks have been missed due to the manual nature of the overall allocation process.
- There is no overall view available at any stage during the day showing what has been completed.

What is involved for the student?

The student’s role will be to develop an automated dashboard for task allocation/review over the course of a day. This application would ideally be web-based and would be accessible by each team member. Team members would see their allocated tasks in their own view, and an overall team view should be available. A traffic light system would let everyone know the status of tasks in relation to
deadlines - green for completed, yellow for pending completion and red for within 10 minutes of deadline. Authorised users should have permission to change allocations.
APPENDIX B
Interim Report

Management Science and Information Systems Studies
Project: Creating a web based dashboard for fund allocation
Client: State Street
Student: James O'Donoghue
Supervisor: Aideen Keaney

Review of Background and Work to Date

State Street is a leading provider of financial services to institutional investors worldwide. Today, with over 2,000 employees in Ireland, they offer local investors a complete range of financial services across the investment spectrum, including investment servicing, investment research and trading, and investment management.

The derivatives team within State Street have a large number of sign-offs over the course of any given day. At the moment, these signoffs are allocated to team members using an Excel spreadsheet and signed off on a Lotus notes database. Once a fund has been allocated, the onus is on the staff member to memorise his/her funds for the given day and the various deadlines of these funds. There is no warning process to show the user of upcoming deadlines and this can lead to missed deadlines as a result of confusion in their schedule.

Once a fund has been complete and signed off, the Lotus Notes database is updated manually to inform other teams that pricing has been complete. The current system does not provide any management reporting.

To date, a large amount of progress has been made.
- A requirement analysis has been conducted with the client.
- The various computer languages and technologies that could be used to develop the dashboard have been researched.
- The basic layout of the dashboard has been agreed upon in principle with the client.
- The basic layout of the database has been constructed using MySQL

Terms of Reference

- To investigate and document the problem through discussions with the client.
- To research possible solutions. The solution should allow historical data to be stored in a readily accessible database.
- To design and develop a dashboard that will:
  - Allow staff members to easily manage their assigned funds.
  - Provide managers with a view of funds progress.
  - Allow management to make adjustments easily to fund information.
  - Output results in a user friendly format.
- To produce both technical and user documentation.
Further Work

Over the coming weeks a number of different aspects of the project still need to be developed such as:

- Familiarisation with the software to construct the dashboard
- Construction of the dashboard
- Construction of the queries to extract relevant information
- An attempt will be made to involve the client more as the project comes closer to completion
- Produce relevant documentation

Once a pilot version of the dashboard is complete, it will undergo testing within State Street to examine any potential areas for improvement.

Conclusions

Developing the dashboard is time consuming but good progress has been made since beginning the project. The dashboard will improve various aspects of the working day in the State Street including improved efficiency, minimising errors and providing added functionality such as the managerial report. The client is enthusiastic and has been very helpful in terms of providing support throughout the project.
APPENDIX C
User Manual

Getting Started:

Before the dashboard can be used, the Manager must first import the upcoming week’s schedule. This process begins in the Excel sheet where the weekly schedule is currently created. As seen in Figure C.1 below, a simple button is added to the sheet which will automatically prepare the spreadsheet so that it can be imported into the new database. Once this button is clicked the Manager will be prompted by a message that will ask whether or not the schedule is complete. This prompt message can also be seen in Figure C.1. This has been included as a precautionary measure to insure that no blank information is imported into the new system.

Figure C.1 – Data Preparation for Dashboard

If the Manager confirms that the schedule is complete, the spreadsheet will then be stored in a location that will be accessible to the dashboard when the Manager clicks on the Import function from his dashboard. This import function will later be explained in greater detail on page C.12. If however, the Manager does not believe the schedule is complete and clicks the “No” option, the Manager will be prompted to ensure that the schedule has been checked before preparing the schedule for the dashboard. This message can be seen below in Figure C.2. Once this process has been completed by the manager, Fund Administrators can now access the dashboard.
In order for this dashboard to be integrated into State Street's Derivative Team, passwords must be allocated to each team member. To date, generic passwords have been temporarily created. However, in order to maintain the security of the dashboard, new passwords must be distributed by management. This will be a unique password that will allow the system to recognise an employee. These passwords will be entered into the database in the **Employees** table. Access to this database will be dependent upon State Street’s servers. However, on the Trinity server this is done through phpMyAdmin by altering the value for the password which can be seen below.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Function</th>
<th>Null</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>varchar(30)</td>
<td></td>
<td></td>
<td>Mark O Keefe</td>
</tr>
<tr>
<td>Password</td>
<td>varchar(7)</td>
<td></td>
<td></td>
<td>mok1</td>
</tr>
<tr>
<td>Initial</td>
<td>varchar(10)</td>
<td></td>
<td></td>
<td>MOK</td>
</tr>
</tbody>
</table>
Users must then enter in the correct log-in information in order to access the dashboard. If the correct details are not entered, an error message will appear prompting the user to re-enter the correct details.

![Logon Screen](image)

**Figure C.4 – Logon Screen**

The dashboard will now split into two different user groups, the Fund Administrator and the Manager. The system will automatically redirect you to one of them depending on your log-in details. For the purpose of this report, there is a walk through the dashboard as a Fund Administrator and then as a Manager.

**Fund Administrator:**

Once a Fund Administrator logs on he/she will be then met with the following main menu.

![Fund Administrator Main Menu](image)

**Figure C.5 – Fund Administrator Main Menu**

This main menu displays a list of the Fund Administrators functions on the left hand side as well as a quick overview of their task list for the day. If he/she would like a more detailed
overview of their tasks for the day they can click the “My Funds List” option on the menu page. This will then bring up the following page;

![Fund Overview](image1)

Figure C.6 – Fund Overview for the upcoming day

The user then has the option to go back to their main menu or to log-out as you can see from the navigation buttons in Figure C.6 above. If at any stage the Fund Administrator wishes to log-out they will be met with a confirmation page which can be seen below;

![Log-out Confirmation](image2)

Figure C.7 – Fund Complete
If he/she decides to go back they will be returned to Figure C.5. If throughout the course of the day, they wish to declare a fund complete they then click on the “Fund Complete” section of their main menu which will bring them to the following page;

![State Street](image)

**Figure C.8 – Fund Complete**

Once the user is brought to this screen they will be faced with a drop down box, populated only with funds that they have been assigned for the upcoming day. From that list they can chose the fund that has been complete and click the “Submit” button. This will then return a confirmation page as seen in Figure C.9.

![State Street](image)

**Figure C.9 – Fund Complete Confirmation Page**

Again the navigation buttons are present for ease of use and user friendliness and if the user choses to go back he/she will be reverted back to his/her main menu page. Finally, if the user has an issue throughout the day and they feel that for some reason they will not be able to complete one of their funds they can click into the “Emergency List”, this will then re-direct them to the following page;
From here the user has the ability to directly email the manager with one click and depending on the issue they currently have can chose either of the options that you can see in Figure C.10.

Manager:

When a Manager enters in correct log-in details they will be directed to their “Manager Home Page”. This will give a list of their functions on the left hand side and statistics of the team’s progress on the right hand side as can be seen below in Figure C.11.
If a more detailed view of the team’s progress was required they could click on the “Fund Overview” section of the main menu. This will then bring the manager to Figure C.12 below. This view coincides with that of a Fund Administrator’s “Fund Overview” except that the Manager can see the progress of the entire team’s funds.

![State Street](image)

**Figure C.12 – Manager Fund Overview**

If the Manager at any stage wants to log-out he/she will be met with a confirmation page similar to that of the Fund Administrator which can be seen in Figure C.7. If however the manager chooses to press the “back” button he/she will be brought back to the main menu.

If the Manager wishes to “Search Fund” they will be re-directed to a screen that will enable them to search a particular fund and display information regarding that fund. This process can be seen below in Figure C.13 and C.14.
If the Manager wishes to “Edit Fund” they will be re-directed to a screen that will enable them to carry out a number of operations that are exclusive to managers. They will be brought to Figure C.15 which displays the various operations available to them.
From this screen the Manager can take advantage of a number of functions exclusive to their dashboard. The Add Fund function, as seen in Figure C.16 below allows the Manager to add funds to the team’s daily task list in the database. All fields must be filled and there are some validations so that incorrectly formatted information cannot be entered. If the field is red the information is invalid and the system will not allow you to progress until it has been entered correctly, which will be indicated by a green box.
If the Manager wished to “Delete Fund” they would be directed to Figure C.17, from which they must select the fund they wish to erase from the database. Once a fund has been selected a confirmation page will appear to emphasize the action that has taken place.

![Figure C.17 – Manager Delete Fund](image)

If the Manager wished to “Edit Fund Details” they would be directed to Figure C.18, from which they must select the fund they wish to edit. Again there are validations in place to ensure the correct information is entered into each field. Once a fund has been edited a confirmation page will appear to emphasize the action that has taken place.

![Figure C.18 – Manager Edit Fund Details](image)

If the Manager wished to “Change Fund Administrator” they would be directed to Figure C.19, from which they must select the fund they wish to change administrator. They must then select the Fund Administrator they wish to transfer the fund too. Once a fund has been edited a confirmation page will appear to emphasize the action that has taken place, an email will also be sent to that Fund Administrator informing him/her that a fund has been added to their daily tasks and they should check their dashboard.
If the Manager did not wish to edit any Fund information, they could go “back” to the main menu. From this menu there are still two functions available to the manager. If he/she wished to “Archive” the week’s schedule, this would involve transferring information from the day tables and inserting it into the Archive table of the database. This function should only be carried out at the end of the week otherwise valuable information will be lost. Once this option has been selected by the Manager, a background operation will be carried out and there will be a confirmation page informing the Manager of changes to the system. This confirmation page can be seen below in Figure C.20.

The final function available to the manager is the “Import” function which can be found on the Manager’s main menu. This will allow the Manager to import the schedule for the week ahead. It is important that before the Manager tries to import the schedule for the week ahead that he/she archives the previous schedule so that valuable information is not lost. There is a confirmation page that occurs when the manager tries to import he/she will be asked if they have already archived the information and only if they have archived the information will the system allow them to import the upcoming weeks schedule.
Figure C.21 – Manager Import Function

Next week's schedule has been imported!
Appendix D
Design Documentation

This appendix outlines the steps undertaken while designing the system as well as descriptions of the processes involved in the final application. The appendix consists of 4 main sections:

D.1 Software Methodology (Incremental Model)

Software methodologies are useful for structuring the software development process. The incremental model was used to design, develop, implement and test the dashboard system. The model was selected as it ensures the client’s requirements are met at each stage therefore satisfying the client’s requirements. It also allows for any issues to be resolved at a lower level where it is far easier to recognise and resolve.

The incremental model combines the elements of the waterfall model and the philosophy of prototyping. An illustration of the incremental model can be found below in Figure D.1.1.

Figure D.1.1 – Incremental model that was used to design, develop and test the dashboard
D.2 Use Case Analysis

Use case diagrams are useful for modeling the interactions that exist between the user and the system. In this case there are two user groups, the Fund Administrator and the Manager. The interactions that the Fund Administrator would have include:

- Log into the system;
- View his/her funds for the day;
- Inform the system that a fund has been completed;
- Send his/her manager an Emergency E-mail;

The main interactions the Manager would have with the system were:

- Logging into the system;
- View the entire team’s funds for the day;
- Add funds;
- Edit funds;
- Change a fund’s owner;
- Delete a fund;
- Search a particular fund;
- Import next week’s schedule;
- Archive the previous week’s schedule;

The result of the use case analysis can be seen in the diagram found in Figure D.2.1 below.

![UML Use Case Diagram of the System](image-url)
D.3 Entity Relationship Diagram

An entity relationship model is an abstract way to describe a database. Figure D.3.1 below describes the link that exists between the various tables in the new system. The *MasterData* table is the core of the database. All other tables are related to the MasterData table. The data tables in Appendix E describe the tables and relationships in greater detail.

The database will be populated in two ways, manual data input and automatic data input resulting from scripting or coding.

![Entity Relationship Diagram of the new system](image)

Figure D.3.1 – Entity Relationship Diagram of the new system
D.4 Design of the dashboard

In order to achieve the preliminary objectives of user friendliness and intuitive use, the design of the dashboard was crucial. For this reason clear sections, as shown in Figure D.4.1 were set out in the dashboard and maintained throughout. This would allow for intuitive use.

Similarly it will be helpful for any further development on the system as changes can be made to certain sections by simply searching for the `<div id="section name">` in the HTML code. By changing characteristics within these sections the display for each will be modified.
## Appendix E
Data Tables

### APPLICATION DESIGN WORKSHEET TABLES

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>Managers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains Log-In details of managers</td>
</tr>
</tbody>
</table>

#### Related Tables

#### Data Name | Data Type | Description               | Validation Rule                              | Linked To |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 30 characters</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>Varchar</td>
<td>Unique Password</td>
<td>Must be less than 7 characters</td>
<td></td>
</tr>
</tbody>
</table>

---

### APPLICATION DESIGN WORKSHEET TABLES

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>Employees</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains Log-In details of Fund Administrators</td>
</tr>
</tbody>
</table>

#### Related Tables

#### Data Name | Data Type | Description                              | Validation Rule                        | Linked To |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 30 characters</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>Varchar</td>
<td>Unique Password</td>
<td>Must be less than 7 characters</td>
<td></td>
</tr>
<tr>
<td>Initial</td>
<td>Varchar</td>
<td>The Initials of Fund Administrator (must be unique)</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
**APPLICATION DESIGN WORKSHEET TABLES**

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>Funds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains details of funds</td>
</tr>
</tbody>
</table>

**Related Tables**

<table>
<thead>
<tr>
<th>Related Tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday, Tuesday, Wednesday, Thursday, Friday</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Fund Name</td>
<td>Must be less than 50 characters</td>
<td>Monday - Friday</td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td>Monday - Friday</td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Point_of_Contact</td>
<td>Varchar</td>
<td>Manager overseeing the fund</td>
<td>Must be less than 30 characters</td>
<td></td>
</tr>
</tbody>
</table>
# Application Design Worksheet Tables

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>MasterData</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains the daily task schedule for Monday</td>
</tr>
<tr>
<td>Related Tables</td>
<td>Monday, Tuesday, Wednesday, Thursday, Friday</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td>Monday</td>
</tr>
<tr>
<td>Tuesday</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td>Tuesday</td>
</tr>
<tr>
<td>Wednesday</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td>Wednesday</td>
</tr>
<tr>
<td>Thursday</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td>Thursday</td>
</tr>
<tr>
<td>Friday</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td>Friday</td>
</tr>
</tbody>
</table>
**APPLICATION DESIGN WORKSHEET TABLES**

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>Monday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains the daily task schedule for Monday</td>
</tr>
<tr>
<td>Related Tables</td>
<td>Archive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Varchar</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
# Application Design Worksheet Tables

**Table Name:** Tuesday  
**Brief Description:** Contains the daily task schedule for Tuesday  
**Related Tables:** MasterData, Archive

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Varchar</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
## Application Design Worksheet: Tables

**Table Name:** Wednesday  
**Brief Description:** Contains the daily task schedule for Wednesday

**Related Tables:** Archive

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Varchar</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
## Application Design Worksheet Tables

<table>
<thead>
<tr>
<th>Table Name:</th>
<th>Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description:</td>
<td>Contains the daily task schedule for Thursday</td>
</tr>
<tr>
<td>Related Tables</td>
<td>Archive</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Varchar</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
### Application Design Worksheet: Tables

#### Table Name: Friday

**Brief Description:** Contains the daily task schedule for Friday

**Related Tables:** Archive

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Varchar</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Varchar</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Varchar</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Varchar</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
## Application Design Worksheet Tables

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Archive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Description</td>
<td>Contains the historic task schedule from previous days</td>
</tr>
<tr>
<td>Related Tables</td>
<td>Monday, Tuesday, Wednesday, Thursday, Friday</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Name</th>
<th>Data Type</th>
<th>Description</th>
<th>Validation Rule</th>
<th>Linked To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fund_Name</td>
<td>Text</td>
<td>Full-Name</td>
<td>Must be less than 50 characters</td>
<td></td>
</tr>
<tr>
<td>Frequency</td>
<td>Text</td>
<td>The Frequency to which the fund must be administered</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Deadline</td>
<td>time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Owner</td>
<td>Text</td>
<td>The Fund Administrator assigned</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
<tr>
<td>Flag</td>
<td>Text</td>
<td>Describes the status of the fund</td>
<td>Must be less than 10 characters</td>
<td></td>
</tr>
</tbody>
</table>
Appendix F
Requirements

For the purpose of this project a number of requirements were set in order to deem this project a success.

1. To carry out research into the current problem and discover possible areas for improvement;
2. Research possible solutions for the design and development of a dashboard;
3. Construct a back-end database that would store the required information to be used by the Derivatives Team;
4. Construct a simple to use interface for end-users, Managers and Fund Administrators;
5. Ensure that the current construction of the Derivative team’s weekly schedule is carried out in Excel. The proposed system must then interact with Excel in order to extract the required information;
6. Eliminate the current reliance on Fund Administrators memorising their tasks;
7. The dashboard must be updated automatically so that it displays live information;
8. Managers will have the ability to get an overview of the team’s progress at any given time;
9. Flexibility for Managers to make changes to the schedule in the event of unplanned circumstances;
10. Include a function so that Fund Administrators can warn Managers of issues that have arisen that may impact their ability to complete their tasks;
11. Include data validations so that the system is working off clean, accurate data;
Appendix G
Screenshots

Figure G.1 – Logon Screen

Figure G.2 – Fund Administrator Main Menu
Figure G.3 – Fund Overview for the upcoming day

Figure G.4 – Fund Complete
Figure G.5 – Fund Complete

Figure G.6 – Fund Complete Confirmation Page

Figure G.7 – Emergency List
Figure G.8 – Manager Main Menu

Figure G.9 – Manager Fund Overview
Fund Search:

Fund Name: [SEB Grantifond]

Submit, Reset

Figure G.10 – Manager Fund Search

Manager Fund Search Record

Fund Name | Deadline | Frequency | Manager | Flag
---|---|---|---|---
SEB Grantifond | 07:30:00 | Daily | |

Figure G.11 – Manager Fund Search Record

Edit Fund Menu:

Add Fund
Delete Fund
Edit Fund Details
Change Fund Administrator

Figure G.12 – Manager Edit Fund Menu
Adding a Fund:

- Fund Name:
- Deadline: [Invalid format]
- Frequency:
- Manager in Charge:

Figure G.13 – Manager Add Fund

Deleting a Fund:

- Fund Name: SEB Grantfond

Figure G.14 – Manager Delete Fund

Editing Fund Details:

- Enter New Fund Name: SEB Grantfond
- Enter New Fund Manager's:
- Enter New Fund Deadline: [Format]
- Enter New Fund Frequency:

Figure G.15 – Manager Edit Fund Details
Change Fund Administrator:

Enter Fund Name:  SEB Grant Fund
Enter New Fund Administrator: Mark O Keefe

Figure G.16 – Manager Change Fund Administrator

There were 2822 rows added to the archive file!

Figure G.17 – Manager Archive Data

Next week's schedule has been imported!

Figure G.18 – Manager Import Function
Appendix H
Sample Source Code

Visual Basic Script

' Copying and pasting the tasklist to a separate spreadsheet as to make tables
' Author - James O'Donoghue
' Last Modified 28/03/13

Sub move()
' Moving the sheet constructed by the Manager to a separate sheet as to not affect the
' current system
answer = MsgBox("Is all the information fully entered?", vbYesNo)
If (answer = "Yes") Then
Do Until (Range("A5").Offset(i, 0).Value = "Early Morning")
i = i + 1
Loop
    Range("A5").Offset(i, 0).Select
    Range(Selection, Selection.End(xlToLeft)).Select
    Range(Selection, Selection.End(xlDown)).Select
    Selection.SpecialCells(xlCellTypeVisible).Select
    Selection.Copy
    Workbooks.Add
    ActiveSheet.Paste
    Columns("A:AX").EntireColumn.AutoFit
Else
    MsgBox ("Fill in all information before preparation for dashboard")
End
End If
End Sub

Sub manipulate()
' deleting all the unnecessary data and manipulating into tables i want
Columns("B:B").Select
Selection.Delete Shift:=xlToLeft
Rows("1:1").Select
Selection.Delete Shift:=xlUp
Do Until (Range("A1").Offset(i, 0).Value ="")
If (IsEmpty(Range("A1").Offset(i, 1).Value) = True) Then
Range("A1").Offset(i, 0).Select
Else
End If
i = i + 1
Loop
End Sub
Sub MasterData()
' Saving the overall Masterdata spreadsheet to import for Archive purposes
Dim w As Workbook
Set w = ActiveWorkbook
Columns("A:H").Select
Selection.Copy
Workbooks.Add
ActiveSheet.Paste
Application.CutCopyMode = False
ActiveWorkbook.SaveAs Filename:="Z:\www\FYP\Funds\MasterData.csv", _
FileFormat:=xlCSV, CreateBackup:=False
w.Activate
End Sub

Import Function PHP

<?php

ini_set ("display_errors", 1);
error_reporting (E_ALL);

include("Home.php");

$db = mysql_connect("$host", $user, $password);
mysql_select_db("$database", $db);

$d1 = "TRUNCATE table MasterData";
mysql_query($d1,$db);
$q1 = "LOAD DATA LOCAL INFILE 'Funds/MasterData.csv' REPLACE INTO TABLE MasterData FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY "'" LINES TERMINATED BY "\r\n";
mysql_query($q1,$db);

$aa = "Insert Into Monday (Fund_Name, Frequency, Deadline) Select Fund_Name, Frequency, Deadline From Funds";
mysql_query($aa,$db);
$aa1 = "Update Monday, MasterData Set Monday.Owner = MasterData. Monday Where Monday.Fund_Name = MasterData. Fund_Name";
mysql_query($aa1,$db);

$ab = "Insert Into Tuesday (Fund_Name, Frequency, Deadline) Select Fund_Name, Frequency, Deadline From Funds";
mysql_query($ab,$db);
$ab1 = "Update Tuesday, MasterData Set Tuesday.Owner = MasterData.Tuesday Where Tuesday.Fund_Name = MasterData.Fund_Name";
mysql_query($ab1,$db);
$ac = "Insert Into Wednesday (Fund_Name, Frequency, Deadline) Select Fund_Name, Frequency, Deadline From Funds";
    mysql_query($ac,$db);
$ac1 = "Update Wednesday, MasterData Set Wednesday.Owner = MasterData.Wednesday Where Wednesday.Fund_Name = MasterData.Fund_Name";
    mysql_query($ac1,$db);

$ad = "Insert Into Thursday (Fund_Name, Frequency, Deadline) Select Fund_Name, Frequency, Deadline From Funds";
    mysql_query($ad,$db);
$ad1 = "Update Thursday, MasterData Set Thursday.Owner = MasterData.Thursday Where Thursday.Fund_Name = MasterData.Fund_Name";
    mysql_query($ad1,$db);

$ae = "Insert Into Friday (Fund_Name, Frequency, Deadline) Select Fund_Name, Frequency, Deadline From Funds";
    mysql_query($ae,$db);
$ae1 = "Update Friday, MasterData Set Friday.Owner = MasterData.Friday Where Friday.Fund_Name = MasterData.Fund_Name";
    mysql_query($ae1,$db);

    echo 'Next week’s schedule has been imported!'
?>

Archive Funds

$t1 = "insert into Archive select * from Monday";
$r1 = mysql_query($t1,$db);
$t2 = "insert into Archive select * from Tuesday";
$r2 = mysql_query($t2,$db);
$t3 = "insert into Archive select * from Wednesday";
$r3 = mysql_query($t3,$db);
$t4 = "insert into Archive select * from Thursday";
$r4 = mysql_query($t4,$db);
$t5 = "insert into Archive select * from Friday";
$r5 = mysql_query($t5,$db);

Manager Editing Database

$ll = "Select Initial From Employees where Name like "$fadmin"";
$result = mysql_query($ll,$db);
$numre = mysql_num_rows($result);
if ($row = mysql_fetch_assoc($result)) {
    $initial = $row['Initial'];
$sql = "UPDATE $day SET Owner = '$initial' where Fund_Name like '%$fund'';
mysql_query($sql, $db);

Update, Combination of Unix script and PHP

1. Unix Script

*/5 06-20 * * 1-5 /usr/bin/php /users/ugrad/jodonog/www/FYP/Update_Flag.php

2. PHP Script

<?php

include ("Home.php");

$db = mysql_connect("$host", $user, $password );
mysql_select_db("$database",$db);

# getting the current time
$ti = date("H:i:s",strtotime("+0 minutes"));

# getting the time 10 minutes from now
$t10 = date("H:i:s",strtotime("+10 minutes"));

# getting the time 30 minutes from now
$t30 = date("H:i:s",strtotime("+30 minutes"));

# getting the current day
$t=date('d-m-Y');
$day = date("l",strtotime($t));

# updating the database when a deadline is within 30 minutes its flag turns 'Purple'
$t = "UPDATE $day Set Flag = 'Purple' Where Deadline < " . $t30 . " AND Flag <> 'Green'';
mysql_query($t,$db);

# updating the database when a deadline is within 10 minutes only on funds that are not already complete
$t2 = "UPDATE $day Set Flag = 'Red' Where Deadline < " . $t10 . " AND Flag <> 'Green'';
mysql_query($t2,$db);

# updating the database so that when a fund is passed a deadline it turns 'Black'
$t3 = "UPDATE $day Set Flag = 'Black' Where " . $ti . " > Deadline AND Flag <> 'Green'';
mysql_query($t3,$db);

?>
CSS

h1, h2, h3, p, li, a, th, td, ul, ol {
  color: black;
  font-family: Arial, Helvetica, sans-serif;
  line-height: 200%;
  font-weight: normal;
}

h1 { font-size: 30px; font-weight: bold; margin-left: 20px; }

h2 { font-size: 24px; font-weight: bold; }

p { font-size: 18px; }

li { font-size: 18px; list-style: none; }

body { width: 984px; margin: 0px auto; border: 1px #000000 solid; background: #ffffff; }

a { font-size: 18px; text-decoration: none; }

th { font-size: 18px; text-align: left; }

td { font-size: 18px; text-align: left; }

.auto-style1 {
  margin-left: 40px;
  text-align: left;
  border: 0;
  width: 50%;
}

#content { font-size: 18px; }

#content table { margin: 40px; table-layout: inherit; }

#logo img { width: 100%; height: 200px; }

#output { margin-left: 10%; margin-bottom: 10px; font-size: 18px; font-family: Arial, Helvetica, sans-serif; }

#output table { width: 80%; }

#output th { border: thin #336699 solid; text-align: center; }

#output td { border: thin #336699 solid; padding-left: 5px; text-align: left; font-size: 14px; font-family: Arial, Helvetica, sans-serif; }

#stats { float: right; margin-right: 60px; margin-top: 10px; }

#stats th { font-size: 12px; }

#stats table { margin-left: 10px; }

#stats h3 { margin-left: 8px; text-decoration: underline; }

#stats td { font-size: 12px; padding: 8px; margin-left: 5px; }

#buttons { clear: both; text-align: center; margin-top: 10px; vertical-align: baseline; }
### Java Script: Confirmation pages

```javascript
function log_out()
{
var s=confirm("Are you sure you want to log out?");
if (s==true)
{
    document.location.href = "fyp1.html";
}
else{ }
}
</script>
```

### Java Script: Data validation

```html
<tr>
<th>Enter New Fund Deadline:</th>
<td><span id="sprytextfield2">
<input name="fdead" type="text" id="fdead">
<span class="textfieldInvalidFormatMsg">Invalid format.</span></span></td>
</tr>

<script type="text/javascript">
var sprytextfield2 = new Spry.Widget.ValidationTextField("sprytextfield2", "time",{validateOn:["blur"],
hint:"##:##"});
</script>
```
Appendix I
Test Documentation

This appendix outlines the testing that was conducted on the dashboard during the development of the system.

I.1 Unit Testing

Unit testing involves testing individual software components. Every function within the system was tested once that particular function had been added to the system in accordance with an incremental model. As there are a large number of functions in the system, the results have not been included in this appendix. However, an example of such testing would include the interaction between the HTML pages and the CSS code. This was checked a number of times to ensure that the HTML pages were reacting to the changes in CSS code and displaying the sections accordingly.

1.2 System Testing

System testing tests major functions of the system rather than single units. All of these tests were eventually successful.

<table>
<thead>
<tr>
<th>Test</th>
<th>Expected Result</th>
<th>Testing Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Funds</td>
<td>New Funds should be added to the database</td>
<td>Creating a number of new funds. Ensured they were inserted into the database. Queried the funds from Managers Main Menu.</td>
</tr>
<tr>
<td>Edit Funds</td>
<td>Fund Details should be changed in the database</td>
<td>Edited a number of funds. Queried them from the main menu and double checked the information.</td>
</tr>
<tr>
<td>Delete Fund</td>
<td>Fund should be removed from the database</td>
<td>Deleted a number of funds from the main menu. Tried to search for the funds.</td>
</tr>
<tr>
<td>Email manager</td>
<td>An email should be found in an inbox, confirmation message on screen.</td>
<td>Ran the query a number of times, changing the email address and the content to ensure that it was working correctly.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
<td>Validation Details</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finish Fund</td>
<td>The flag status should be changed to “Green”.</td>
<td>Checked from a few different Fund Administrators accounts if you pressed complete fund the flag would change to Green in the database.</td>
</tr>
<tr>
<td>Finish Fund (past deadline)</td>
<td>The Flag will stay “Black” because the fund is already passed its deadline.</td>
<td>Checked from a few different Fund Administrators accounts if you pressed complete fund the flag would stay Black in the database and the error message would appear.</td>
</tr>
<tr>
<td>View Funds</td>
<td>The list of funds applicable to the person logging in.</td>
<td>Tested it using a number of different Fund Administrators and Managers log-in details. The information changed for each user and was displayed in the correct format.</td>
</tr>
<tr>
<td>Summary Statistics</td>
<td>The correct statistics displayed in the box on the right hand side of both the Fund Administrator and Managers Main Menu.</td>
<td>Logged in with several different log-in details which included a mix of both Managers and Fund Administrators. Then manually checked the statistics displayed against the information contained in the database.</td>
</tr>
<tr>
<td>Change Fund Owner</td>
<td>The “Owner” should be changed in the day table to the new “Owner” that was selected by the Manager.</td>
<td>Carried out the change fund owner process on a number of different funds on different Fund Administrators.</td>
</tr>
<tr>
<td>Import Schedule</td>
<td>The schedule created by the Manager in Excel will be imported correctly into the back-end database.</td>
<td>Ran the query a number of different times. Checked that all information was inputted into the correct value and correct table.</td>
</tr>
<tr>
<td>Archive Schedule</td>
<td>All records from the day tables should be transferred into the Archive table and</td>
<td>Ran the Archive function a number of times with different managerial log-in details.</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Details</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Confirmation Pages</td>
<td>A prompt would appear asking the user if he/she was sure of their decision and would act accordingly.</td>
<td>Tested this from various different pages and with different log-in details for both Managers and Fund Administrators. Made sure that the system would direct the user in the correct manner depending on their decision.</td>
</tr>
<tr>
<td>Data Validation</td>
<td>The system will not allow the user to proceed if incorrect information is entered</td>
<td>The deadline time input field was tested with a number of different values in the field and in different formats.</td>
</tr>
<tr>
<td>Web Browsers</td>
<td>All pages from the dashboard should look and behave correctly in various web browsers.</td>
<td>Tested multiple pages in Internet Explorer 9, Mozilla Firefox 18.0.1, Google Chrome 26.0.1410.13 m and Apple Safari.</td>
</tr>
</tbody>
</table>

Table I.1 – Tests carried out
Appendix J  
Technical Manual

Since the dashboard was created using Microsoft Web Expression 4, a freely downloadable software package, the entire dashboard can be managed cost free. The list of current pages that are included and their functionality can be found in Table J.1 below.

<table>
<thead>
<tr>
<th>Page Name</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add_Fund.Html</td>
<td>This form is where the Manager will enter in the details of the new fund. It will then run the Add_Fund_Record php script using the details provided.</td>
</tr>
<tr>
<td>Add_Fund_Record.php</td>
<td>This PHP page inputs the information that the Manager provided in the Add_Fund page into the back-end database.</td>
</tr>
<tr>
<td>Admin_Main_Menu.php</td>
<td>This PHP page is where the Fund Administrator will be re-directed once they log in. It contains the various functions available to the Fund Administrator and also includes summary statistics that have been customised to suit the Fund Administrator.</td>
</tr>
<tr>
<td>Archive.php</td>
<td>This PHP page gathers all the information from the day tables and transfers that information to the Archive table. It will not delete any information that is currently in the Archive table, it will simply add the data to the current records.</td>
</tr>
<tr>
<td>change_fund_administrator.php</td>
<td>This PHP page automatically generates a list of funds that are currently on file in a drop down menu. Similarly, it automatically generates a list of all the Fund Administrators. The Manager then selects the Fund that they wish to change owner of and select the Fund Administrator they wish to assign the fund to. This information is then passed onto the update_fund_admin page.</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Create_email.php</td>
<td>This PHP page automatically emails a manager an urgent mail informing him/her that an issue has arisen and that</td>
</tr>
<tr>
<td>Delete_Fund.php</td>
<td>This PHP page automatically displays a list of all the funds that are currently on file in a drop down menu. The Manager then selects a fund from the list that he/she would like to delete. This information is then passed onto the Delete_Fund_Record page.</td>
</tr>
<tr>
<td>Delete_Fund_Record.php</td>
<td>This PHP page gathers information that the Delete_Fund page and carries out a delete query on the back-end database using the information provided by the user.</td>
</tr>
<tr>
<td>Edit_Fund.php</td>
<td>This PHP page automatically generates a list of funds that are currently on file in a drop down menu. The Manager then enters information which he/she wishes to attach with the fund. This information is then passed onto the Update_Fund_Record page.</td>
</tr>
<tr>
<td>Edit_Fund_Menu.Html</td>
<td>This HTML page displays a list of functions that the Manager can choose from if he/she wishes to make any changes to the current fund list. The page then redirects the Manager to their desired location depending on the function they chose.</td>
</tr>
<tr>
<td>Emergency.Html</td>
<td>This HTML page displays a list of functions available to the Fund Administrator and then transfer the Fund Administrator to the correct page accordingly.</td>
</tr>
<tr>
<td>PHP Page</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finish_Fund_Record.php</td>
<td>This PHP page gathers the information from the finishfund page and updates the back-end database with the information. It then displays a confirmation message to the Fund Administrator.</td>
</tr>
<tr>
<td>finishfund.php</td>
<td>This PHP page automatically displays a list of all the funds that are currently on file in a drop down menu. The Fund Administrator then selects the fund which they have completed. This information is then passed onto the Finish_Fund_Record page.</td>
</tr>
<tr>
<td>fund_name.php</td>
<td>This PHP page gathers the information from the search_funds page and carries out a search query on the back-end-database. The page then displays information from the fund that the Manager wished to search.</td>
</tr>
<tr>
<td>Fund_Query_Admin.php</td>
<td>This PHP file queries the back-end database for fund information pertaining to the user. The user is identified by a session variable that was created when he/she logged on. It then displays this information in a table format for the Fund Administrator to analyse.</td>
</tr>
<tr>
<td>Fund_Query_Manager.php</td>
<td>This PHP file queries the back-end database for fund information for the whole team. It then displays this information in a table format for the Manager to analyse.</td>
</tr>
<tr>
<td>fyp1.Html</td>
<td>This HTML page represents the home page for the dashboard. It is here where the user, Fund Administrator or Manager will be asked to enter their log-in details.</td>
</tr>
<tr>
<td><strong>Home.php</strong></td>
<td>This contains log-in details for the server so that the remaining pages can access the server without having to re-enter their details repeatedly.</td>
</tr>
<tr>
<td><strong>Import.php</strong></td>
<td>This PHP page runs an import function, to extract the files created using the Visual Basic Script that have been stored in the “Funds” folder which can be found in the root folder.</td>
</tr>
<tr>
<td><strong>login_check.php</strong></td>
<td>This PHP page queries the log-in details entered by the user against the records in the back-end database and will re-direct the user to the correct menu depending on their log-in information.</td>
</tr>
<tr>
<td><strong>Manager_Main_Menu.php</strong></td>
<td>This PHP page is where the Manager will be re-directed once they log in. It contains the various functions available to the Manager and also includes summary statistics of the entire team’s progress.</td>
</tr>
<tr>
<td><strong>search_funds.php</strong></td>
<td>This PHP page generates a list of all the funds that are currently on file in a drop down list. The Manager then chooses the fund they wish to search and will then pass that information onto the fund_name page.</td>
</tr>
<tr>
<td><strong>Update_Flag.php</strong></td>
<td>This PHP page queries the funds deadline times against various time markers and determine whether or not their status should be changed. If their status does need to be changed, this file carries out those changes on the back-end database.</td>
</tr>
</tbody>
</table>
This PHP page gathers the information from the change_fund_administrator page. It then carries out a query to change the Ownership of the fund from its previous Fund Administrator to the one selected by the Manager. It then emails the new Fund Administrator informing them that a change to their schedule has occurred and that they should consult their funds list.

This PHP page collects the information from the Edit_Fund page and carries out an update query on the database, entering in the new information for the fund selected by the user.

Table J.1 – Description of all the web pages currently in the new system

Update Function:

Crontab is a feature of Linux servers which allows you to execute a specified command at set intervals. The intervals can be any valid combination of the arguments specified as the first five entries in a Crontab line. The 6th entry specifies the file to be executed by Crontab at each interval. The code for the Crontab function which will run every 5 minutes can be found in Appendix H.3. The format for this function is as follows:

1. Minutes interval;
2. Hour interval (specified between 0-23, 0 = 12 midnight, 23 = 11pm);
3. Day of month interval (specified between 1-31, 1 = 1st day of the month);
4. Month interval (specified between 1-12, 1 = January, 12 = December);
5. Day of week interval (specified between 1-7, 1 = Monday, 7 = Sunday)
6. File to be executed at each interval;

A * at any of these intervals indicates that the function is to be run for every possible value. For example, a function with * as a value for the minute interval will run every minute of every interval specified by the remaining values of the function. This will be the same for the remaining intervals. If however a minute value of 30 was used, the function would then run every 30th minute for every interval specified by the remaining values.

Data Manipulation – Excel:

The data manipulation occurs once the Manager has completed the weekly schedule. This manipulation is necessary so that the tasklist can be imported error free into the back-end
database. There are two main sub-routines that are carried out, the Move and the manipulate subroutine.

1. The ‘move’ subroutine has been created to extract the daily task-list from the spreadsheet and to eliminate information that is not relevant to the upcoming week’s schedule. This will only run after the Manager has confirmed that the spreadsheet is fully completed and that there is no missing information on the task list. The required information is then copied from the spreadsheet and is pasted onto a new sheet.

2. The ‘manipulate’ subroutine is to format the spreadsheet that has been created from the ‘move’ subroutine. This routine will delete any blank rows or any headers that may exist. Once this has been complete, the file will then be ready for the back-end database to import.

3. The remaining subroutines will simply save the newly created spreadsheet as a CSV file in the root folder. It will specify the file’s name so that when the import function is run from the dashboard, it will search for that specific name and import the spreadsheet.
Glossary of Terms Used

CSS – Cascading Style Sheets
- Presents the style of web pages written in HTML

Server
- A physical computer dedicated to serve the needs of other computers on a network.

HTML – Hypertext Mark-up Language
- A language used for the creation of websites.

MySQL – Structured Query Language
- MySQL is a relational database management system.

PHP – Hypertext Preprocessor
- A language embedded within HTML which is used to create dynamic content.

UML – Unified Modeling Language
- A language which is not specific to any individual code.

Java Script
- An interpreted computer programming language. Implemented as part of web browsers so that scripts could interact with the user.

Visual Basic
- Is a third-generation event-driven programming language and integrated development environment, developed for Microsoft Applications such as Excel.

Dashboard
- An easy to read, real time user interface that displays a snapshot of the current status of an organisation’s or team's key performance indicators.
REFERENCES


