Supporting Visual Exploration of Historic Records

Simon Maishman
B.A.I. Engineering
Final Year Project April 2012
Supervisor: Dr. Owen Conlan

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland
Declaration

I hereby declare that I am the sole author of this dissertation and that the work presented in it, unless otherwise referenced, is my own. I also declare that the work has not been submitted, in whole or in part, to any other university or college for a degree or other qualification.

I agree that Trinity College Dublin may lend or copy this thesis in whole or in part, on condition that any such use of the content of the dissertation is duly acknowledged.

Signed,

____________________________________________

April 4th 2012
Abstract

With the vast array of data available today, one of the main problems encountered by researchers is an inability to visualise trends in data. It is difficult for researchers to access both high level overall trends and mine the raw data. In this project these problems are solved by creating user driven software that visualises and mines historical financial data.

This software is split up into different sections in order to solve the various challenges. A rich internet web application allows users to customise the graphs and display the result easily from any platform. There are several intermediate layers which extract the correct data and create the visualisations.

One of the main problems facing modern software developers is not the lack of computing power, but delivering the software in a meaningful way to the end users. This system aims to rectify this by involving users closely in every stage of the design and development cycle. This method was successful in tailoring a user-centric system to specific non technical domain experts.
Acknowledgements

I would like to extend my sincere thanks my supervisor Prof. Owen Conlan. For taking on this project and providing such valuable assistance and enthusiasm throughout. Also thanks for keeping patience during the attempts at thesis writing and (more than) occasional lateness to meetings.

I'd like to thank all my expert users, who gave up their time to provide feedback for this project: Prof. Jane Ohlmeyer, Dr. Micheál Ó Siochrú and Dr. Patricia Stapleton.

Special thanks to Dr. Mark Sweetnam, who was invaluable in the regular users meetings and for setting up the evaluating meetings.

Thanks to the rest of the user meeting team including Prof. Seamus Lawless. The project designs were drastically improved by intelligent feedback and insightful comments in these meetings.

Thanks to Eoin Bailey for his help in the initial meetings, and participation in the user meetings. Bilal Yousuf for the help accessing the statues staples database.

Thanks to Alex O’Connor for guiding me through the murky waters of the Trinity College ethical approval guidelines, and for proofreading my various attempts without mocking my appalling English.

I'd also like to thank my friends and family.

Luke and Nick Maishman – for proofreading my sections of my thesis and providing such professional and helpful feedback.

John Cooper for listening to my presentation repeatedly and the computer engineers who inhabit LG12 for providing such a laid back and friendly writing atmosphere and helping with grammar and vocabulary questions.
# Table of Contents

1 Introduction ................................................................................. 6  
  1.1 Motivation .............................................................................. 6  
    1.1.1 End user Needs................................................................. 8  
    1.1.2 Technical challenges....................................................... 9  
  1.2 Research Question: ............................................................... 10  
  1.3 Objectives and Goals ............................................................ 10  
    1.3.1 Objective 1 Analyse ......................................................... 11  
    1.3.2 Objective 2: Design and Implementation ...................... 12  
    1.3.3 Objective 3: Evaluation ................................................. 12  
  1.4 Case study 1641 ....................................................................... 12  
  1.5 Report Overview .................................................................... 14  

2 State of the art ............................................................................. 17  
  2.1 Chapter Overview ................................................................... 17  
  2.2 Visualisation ........................................................................... 17  
    2.2.1 Introduction .................................................................... 17  
    2.2.2 Development of Data Visualisation ................................. 18  
    2.2.3 Definition ....................................................................... 19  
    2.2.4 Data Visualisation in the Information age ....................... 19  
  2.3 Technologies .......................................................................... 20  
    2.3.1 Graphing language ......................................................... 22  
    2.3.2 PHP and JPGraph ........................................................... 22  
  2.4 Rich internet applications ...................................................... 23  
    2.4.1 Introduction .................................................................... 23  
    2.4.2 RIA Overview ................................................................. 25  
    2.4.3 Background .................................................................... 25  
    2.4.4 Technologies used ........................................................... 26  
    2.4.4 RIA Conclusion ............................................................... 29  
  2.5 Human Computer Interface .................................................. 31  
    2.5.1 Introduction .................................................................... 31  
    2.5.2 Background ..................................................................... 31  
    2.5.3 Key Concepts of HCI ...................................................... 32  
    2.5.4 Implementing a HCI ....................................................... 33  
    2.5.6 HCI Conclusion .............................................................. 35  
  2.6 Case Studies .......................................................................... 36  
    2.6.1 Gapminder ...................................................................... 37  
    2.6.2 Polaris ........................................................................... 39  
    2.6.3 Case Studies Conclusion .............................................. 39  

3 Design and Implementation .................................................... 40  
  3.1 Chapter Overview ................................................................. 40  
    3.1.1 Introduction .................................................................... 40  
    3.1.2 Structure of this Section ................................................. 40
6 Appendix

5 Conclusion

4 Evaluation

3.2 Requirements

3.2.1 Requirements from the Research Question

3.2.1 Requirements from the State of the Art

3.3 High level Architecture

3.4 Methods of implementation

3.5 Phase 1 – initial design

3.5.1 Stage 1 – First wire frame and Technology Demonstration

3.5.2 Stage 2 - Rapid prototyping of core functions

3.6 Phase 2 Main Implementation phase

3.6.1 Stage 3 - Implementation of Graphical and Data layer functionality

3.6.2 Stage 4 – Application layer reimplementation

3.7 Phase 3 - Refinement

3.7.1 Stage 5 – Improved Query Organisation

3.7.2 Stage 6 – Final Alterations

3.8 Testing

3.8.1 Regression Testing

3.8.2 Application layer Error 1

3.8.3 Application Layer Error 2

3.8.4 Data layer Error 1

3.8.5 Data layer Error 2

3.9 Design and Implementation Conclusion

3.9.1 The Completed System

3.9.2 The Completed System

3.10 Conclusion

3.11 Conclusion

3.12 Conclusion

3.13 Conclusion

4 Evaluation

4.1 Chapter overview

4.2 Self evaluation

4.2.1 Introduction

4.2.2 State of art requirements

4.2.3 Research Question Requirements

4.3 External evaluation

4.3.1 Purpose of Users tests

4.3.2 Methods and Measurements of the User Test

4.3.3 Question and Answer Session

4.3.4 SUS

4.3.5 Participants

4.3.6 Results

4.3.7 Discussion of these results in general terms

4.3.8 Evaluation Conclusion

5 Conclusion

5.0.1 Objectives Achieved

5.0.2 Answering the Research Question

5.1 Future work

5.1.1 Current position

6.1 User Tasks

6.1.2 Areas for Improvement

6.2 Summary

6 Appendix

6.1 User Tasks

6.2 Summary
6.2 User SUS questionnaire................................................................. 124
6.3 User debriefing questions............................................................ 125
6.4 User Instructions........................................................................ 126
Instructions for using the Visualisation of Statues Staples Database:........... 126
Index of Figures

Illustration 1: Comparison of a Google Docs RIA (left) and a desktop application word processor -LibreOffice, (right) .................................................................................................................. 27
Illustration 2: Loading screen for flash game RIA as shown on the right. Note the dedicated loading screen on the left ............................................................................................................. 29
Illustration 3: Gapminder RIA ........................................................................................................... 38
Illustration 4: The Polaris System ....................................................................................................... 41
Illustration 5: System architecture ..................................................................................................... 45
Illustration 6: Data mining as initially proposed to users .................................................................. 48
Illustration 7: Web architecture as proposed to in the initial wire-frame ........................................... 49
Illustration 8: Ticker taping as initially presented to users ............................................................... 49
Illustration 9: Radio graph of geographical data manually extracted from the statues staples database ........................................................................................................... 50
Illustration 10: Graphing amount data, the GL at the end of stage 1 ................................................. 59
Illustration 11: Radio graph, with data from the database, shown in a html web-page, the application layer ........................................................................................................... 61
Illustration 12: graph with get requests: st=1600&DC=0&g=0&d=0&title=Custon%20Graph%20Name&X=0&numFeilds=1&SQL=0&yearsNum=10 .............................................................. 63
Illustration 13: different graph with settings: st=1620&DC=1&g=2&d=0&title=Changed %20Graph%20Name&X=0&numFeilds=1&SQL=2&yearsNum=20 ................................................................................ 64
Illustration 14: Start date text input .................................................................................................. 64
Illustration 15: Drop down graph select ............................................................................................. 64
Illustration 16: Drop down debtor/creditor select .............................................................................. 64
Illustration 17: display of area selected on graph ............................................................................. 65
Illustration 18: The URL field ........................................................................................................... 65
Illustration 19: Application layer as a result of the work in stage 2 ..................................................... 68
Illustration 20: Example of final ticker taping result ........................................................................ 75
Illustration 21: Slider ....................................................................................................................... 75
Illustration 22: Comparison of inversion of the axis label. Left has date on x-axis and right has query on x-axis .................................................................................................................. 77
Illustration 23: Application at the implementation phase .................................................................. 78
Illustration 24: Slider with start and end point .................................................................................. 79
Illustration 25: The three query option sets ...................................................................................... 80
Illustration 26: Final query select input layout .................................................................................. 80
Illustration 27: Example of graph with total ...................................................................................... 83
Illustration 28: Line graph with single duplicated query ..................................................................... 85
Illustration 29: Example of the exception catching in the application layer ...................................... 92
Illustration 30: The application input outlining the main features .................................................... 94
Illustration 31: Image indicating the the parts of the system including ticker taping ....................... 95
Illustration 32: Example of direct access to graphing layer visualisation where
http://localhost/sqlGraphBasisMultiWithMultiGraphsFieldSelect8.php is the URL of the graphing layer .................................................................................................................. 97
Illustration 33: Example of direct URL access of the data layer.................................100
Illustration 34: Three examples of selection user feedback. ......................................101
Illustration 35: Example of overview visualisation of all data from a query...............102
Illustration 36: Example of zooming in from from wide (left) to small range of dates (right)............................................................................................................................102
Illustration 37: Zooming of data from many queries (left), so a single query (right) over the same date period.................................................................103
Illustration 38: Example of details on demand.............................................................104
Illustration 39: Input selected and not.........................................................................105
Illustration 40: Date picker for start date.................................................................106
Illustration 41: Example of error alert..........................................................................107
Illustration 42: System open in Mozilla Firefox (Left) and Google Chrome (right) internet browsers........................................................................109
Illustration 43: Table of raw data displayed on the application.................................110
Illustration 44: SUS feedback box..............................................................................115
1 Introduction

1.1 Motivation

Visualisation tools need to accurately display the vast amount of historical data in a way that the end user understands and can interact with in a simple and intuitive manner. The system aims to solve these problems by providing a web interface which visualises the data and takes user input from buttons and other simple input devices.

The system aims to fulfil the needs of historical researchers into this period, in addition to casual users.

These historical researchers have domain specific expertise with regard to the content of the Statues Staples database; however the data is provided in such a form that users may lack the necessary expertise needed to access the data. This results in inaccessible data creating a knowledge gap and unused or under-used resources; which is the problem this project aims to solve.

These are tasks that are technical in their nature and time consuming to do on a record by record basis. Therefore it was useful to create a system that could deal with this noisy data and extract only the desired and reliable factual evidence to the end user.

The main problem to solve was an experience gap in the user group with regards to the use of the techniques and technologies required to extract the necessary data. This system aimed to bypass these technical challenges and isolate the user form this complexity.

Skills such as using SQL to query the required relational databases and tables in order to return the correct information and setting up the correct type of graph are challenges that are beyond...
the skill set and specific expertise of the historian users. Therefore it is valuable for this to be provided by the application. Users knew how to use the data, and what data they wanted to analyse but the restricting factor was accessing the data. This required multiple user defined SQL queries derived form user input. The user input was simplified as much as possible in order to make it as easy and intuitive as possible to find and create the correct type of visualisation as overcoming the lack of technical expertise was one of the main benefits the system was designed to deliver.

When designing the system is was important to understand and then provide what was required rather than what the user thought they wanted, since users often have difficulty expressing their needs. It is a lot clearer to see what need to be delivered from this system from outside. The knowledge of what was possible and what the technologies could deliver was vital. This is what this dissertation aims to describe, as a member of a technical discipline can evaluate the correct restrictions of the project and the underlying technology and can provide the most usable solutions to address these restrictions – providing features that wouldn't have occurred to result driven users. As Henry Ford famously said "If I asked my customers what they want, they simply would have said a faster horse."

When designing this system it was important to liaise with potential end users to ensure that the resulting application would be able to provide what was required. As with many tools in the modern world, the skills required for the end use of the device do not necessarily overlap with those required to design and create it. In particular, the system was designed to ensure that the “filtering effect” intended to clean up the data and display only the reliable
parts of the original dataset, was not likely to accidentally fudge the data in a manner indecipherable to the end user.

The cornerstone of this project was to provide a forum in which the raw data is as easy to observe as possible. However, interpreting the bare tabular results retrieved for the database is difficult. The lines of unconnected facts and figures contain valuable information but are time consuming to make sense of or gain overall understanding of what the data indicates. For this reason it was deemed useful for the application to show the overall trends of the data in easily interpreted visual representations, such as graphs. Using this top down view it is easier to see the trends of the data. From this it is possible to zoom in and mine the data down to the tabular original data.

**Resource used**

This project aims to provide an application that will display a visual representation of the statues staples database. This is a database containing the details of the debtors and creditors in Ireland during the period 1597 to 1687. The records have been compiled from the library records in Dublin and Britain.

1.1.1 End user Needs

The end user needs to be able to find and show the information that they desire from the resource in a simple and intuitive manner. There were several constraints that prevented this from occurring – which this project set out to overcome.

There are IT skill required just to access the database in a meaningful manner and extra the desired data. These SQL queries are often a barrier for the targeted end users, who have expertise in a different area.
In addition to making it possible for the end user to access the database, further motivation for this project was to provide a “filtering effect” to clean up the unhelpfully noisy or inconsistent data in as it was retrieved. This can be described as two separate issues:

Firstly the data in the statues staples was complex and noisy, as it was directly transcribed from a primary source (the British or Dublin Library transcripts). There are several tables that need to be joined on the correct column in order to connect the debtors, creditor and the details of the transactions. Many tuples, rows of data defining a single instance, are also incomplete, with missing data, incorrect labels or different levels of detail for many of the records. Therefore the data should be filtered in order to get the most accurate and most comparable values. The system I have developed performs this task automatically in order to extract only the relevant data for each query. For example many of the Gentry were further defined as Gentry: Gent or Gentry: Landowner etc, however there wasn't enough correlation or uniform connections between the labels to subdivide the profession any further than Gentry.

1.1.2 Technical challenges

The main technical challenges that had to be overcome in order to create this application were to extract the right data and show it in a appropriate visual form.

We chose to use the simplest and most recognisable visual forms. Therefore bar, scatter and line graphs were used as they show trends and are easily to interoperate.

We wanted to provide the maximum possibility for data mining. For academic or professional publications it’s important to be able to trace the figures to the exact person. The nature of the information
meant that is was important to be able to see the name of each member of every transaction. So the system needed to provide the relevant source data from each of the query’s to back up the visualisations.

Variable time lines were very important as users could survey a large time period and then concentrate on specific periods of interest within these sections, such as peaks and troughs.

There were different query's to be represented; so the user could show different sets of data on the visualisations. These needed to change to specific SQL queries, reliant on the variables selected to be queried. Edge cases from all available user inputs had to be worked around in order to provide the most stable experience to the user.

Finding the right language to create the back end was one of the most important steps. PHP provided the required server side functionality, most importantly the JPGraph library was ideally suited to provide the visualising of the data. A pre-made graphing library was chosen as the purpose of the project was to get the right visuals to the user rather than to create a perfect graphing utility. This project aims to build on the work up to this point so there was no point making functionality that had already be created.

1.2 Research Question:

The question that this thesis will answer is “to what extent can an appropriate web-based interface for non-technical domain experts to support visual querying /exploration of historic records be developed?”

1.3 Objectives and Goals

In order to answer the above research question the goal of this
thesis is to propose and evaluate a software application that will take user input, and using a independent layered organisation of scripts provide a visualisation of the required data.

There are several major objectives require in order to meet this goal; outlined as follows:

1.3.1 Objective 1 Analyse:

In order to provide the best possible solution it is important to build on the relevant work of others. Therefore there was extensive research conducted on the state of the current visualisation technologies and techniques.

The aim was to find the current state of visualisation, to see the areas from which this project will take inspiration, and pitfalls to avoid. In order to provide a new and successful application you must evaluate the current state of the technologies and the techniques.

The leading techniques will supply guidelines to follow and the technologies provide a means to accomplish the tasks. We needed to find the right techniques - those that provided the best user-centric environment and which could be successfully implemented with the current technologies.

The technologies were researched extensively in order to find those that best suited the needs of the system. It was important to find the right scripting language to create the best possible graphs types, extracting the data and delivering the minded raw data to the user. The client side functionality required a suitable language to make the most user friendly interface. The dual action of the web application meant that the client scripts had to provide user functionality and to successfully integrate with the back-end. A key feature of a successful or a user friendly interface is up to date
graphics so the correct language and implementation was required to ensure that a “web 2.0” interface was provided.

1.3.2 Objective 2: Design and Implementation

The system will be developed in conjunction with user meetings in order to keep the system in line with what they required from the system. The design phases will provide the goals needed to be achieved in the following implementation cycle. The result of this objective will be a system that satisfies the users in the evaluation phase and build on the current standards found in the state of the art and answer the research question.

1.3.3 Objective 3: Evaluation

As this is a user driven application this thesis needs to evaluate the performance of the application as seen by end users for which it was designed. The success of this application will be based on how well the research question was answered and the opinions of those who will be using it professionally. So this objective will determine the success of the project.

The evaluation took the form of several user-tests. Extensive meetings where the users were required to utilise the finished system to complete a set of tasks similar to those required by an average user of the system. Usability surveys and a question and answer session gathered feedback to provide an honest evaluation of the system.

1.4 Case study 1641

The Catholic uprising occurred between October 1941 and March 1642. This was the most important historical action during the time
period covered in statues staples database, and one of the reasons that the historical data from this period is of such interest. The relationship and trends of the lenders and creditors during this period of civil unrest provide historians with a great insight into this tumultuous period.

This rising was led by the Irish Catholic gentry class in Ireland due to religious oppression – mainly from the Protestant English and Scottish who had arrived in Ireland during the plantations. They attempted a coup in Ireland to control the Irish parliament. However after this failed the Irish Confederate Wars started - comprising a religious conflict between the protestants and Catholics.

The rebellion was a long term result of the plantation of large parts of Ireland (Ulster especially,) by protestants settlers from mainland Britain. This led to the domination of Irish parliament by protestants and the conflicts between the catholic majority and the protestant newcomers. The protestant minority had the majority of seats in parliament, financial power and the ear of the protestant dominated English parliament. This came to a head with the ongoing religion based conflict between king Charles 1 and English parliament. Charles made a deal with the Irish catholic upper class, but after he reneged on this agreement, they rose up in violence.

He needed to support of the Irish catholic upper class in order to raise an army to put down a protestant rebellion in Scotland, after the English parliament wouldn't allowing him to raise on in Britain.

During the rebellion the Irish took control of the majority of the country, killing or dispossessing thousands of settlers. The fight was fiercest in Ulster against pockets of protestant resistance.

The uprising and the confederate wars were ended after the civil
war in England was concluded. There was extensive retribution against the Catholics after the number of protestants killed in the Ulster slaughters and other areas were exaggerated.

1.5 Report Overview

This chapter discussed the motivation research question and objectives of this research. The reason for conducting this piece of work from an end user perspective and the technical challenges faced were described. The research question was outlined in addition to the objectives of this thesis.

The next chapter gives an overview of the state of art with regard to visualisation of data, both the technologies and the techniques that have been applied to this problem are discussed. The current standpoint of the technology is extensively researched in order to use the best tools to provide the quickest and most up to date solution. This will drastically affect how the system is both designed and implemented as it is vital to have some experience of how others have used the various technologies in order to solve similar problems. This will inform the decisions made and ensure that the system will use correct and standardised methodology to provide a unique and new solution. The techniques that have been applied to visualisation, especially those devised by Schniderman, provided the groundwork for systematic approaches to providing the most user-centric application and visual representation.

The way the human users interact with the application is critical to the usability of the software. Therefore the standards of human computer interaction needed to be researched and observed. This was to ensure that the application would have familiar use, so novice researchers feel comfortable using it and the controls feel natural.
Although this application provides new and exciting functionality it is important that it does not seem too new and have unnecessary departure from the familiar application iterations. This is why rich internet applications were also studied; in order that this application was similar and recognisable as one of them.

Chapter three deals with the design and implementation of the final product. The design and the implementation were interwoven due to shifting goals and regular tweaks to the design throughout the implementation. This was due to the user driven approach to the design. The system had several design and implementation iterations; each of which involved a design step – where the targets and solutions would be mapped out, a implementation phase – where these plans were implemented, evolving with the available technology to solve the problems and the feedback session. In these user meetings the users would be provided with the progress and return reactions, improvement ideas and recommendations. In each stage of the cycle both design and implementation choices were made resulting in a constantly evolving, intertwined design and implementation. This short feedback loop with the users meant that they were involved in every step of the design and development cycle resulting in a tailor made product that fulfilled all of the required functions and provided a forum to discover new hither to unimagined functionality.

The penultimate chapter outlines the evaluation of the system. This includes the focus, process, engagement and the findings from the user tests. The users tests are of particular importance as this is one of the main bases for the success of the project. Even with the close relationship with the user body throughout the development
the litmus test of the system was the reaction of new researchers who had no prior iteration with system. The efficiency, satisfaction and technical accuracy as evaluated by the users was integral to gauging the effectiveness of this solution. Discussion of these results is also included in this chapter.

The final part of this thesis is the conclusion of the key element of the user driven visualisation of historical data and a view on how this project will affect future work in this field.
2 State of the art

2.1 Chapter Overview

This section of the report will outline the state of research with regard to visualisation of data-rich internet applications and human computer interaction. It will examine current papers on the theory of visualisation, the history of visualising data and look at some of the best methods in use today. There have been many different approaches to visualising data; this chapter will examine the different techniques devised to tackle this problem, focusing on the work of Schniederman. The current technology used to visualise data will be discussed, in addition to some relevant solutions. The main reference for this is the Gapminder graphing application.

This project aims to produce a Rich Internet Application (RIA) for use by a relatively non-technical user base. Therefore the current standards of Rich Internet Application and Human Computer Interfaces (HCI) programmings will also be considered.

2.2 Visualisation

2.2.1 Introduction

"A picture is worth a thousand words"

Napoleon Bonaparte

As long as there have been people, there have been pictures. As far back as 32 thousand years ago early humans have been displaying their opinions and drawing conclusions from images. [Clottes, J]
As described in the quote by Napoleon above; understanding can be achieved far quicker using visuals than from raw data. There are theories that humans acquire as much information in visual form as from all other senses. (Ware). In addition to being information rich, visuals provide gravitas, authority and meaning to otherwise confusing and unconnected data.

"A picture is a fact"

Ludwig Wittenstein

“The most important addition that visualisation brings is the orientation or context, to enable selection of regions, and to provide dynamic feedback for identifying changes, that abstract information visualization has the power to reveal patterns, clusters, gaps, or outliers in statistical data, stock-market trades, computer directories, or document collections” (Schneiderman).

Overall, “the bandwidth of information presentation is potentially higher in the visual domain than for media reaching any of the other senses.” [Shneiderman, B. ]

2.2.2 Development of Data Visualisation

The earliest forms of images information encoding are maps. Dots dating from 16,500 BC have been found on the walls of caves mapping stars as references’[Cartography Hist] .

The oldest dedicated maps are from the 9th century BC. Maps were the first natural foray into graphical representation of data and are an example of the positive attribute of visualising data. Maps allowed people to easily describe the landscape and show the positions of interest. Explorers used the charts to quickly record large amounts of geographical data that otherwise would have taken the famous thousand words per image to convey.
As civilisation evolved so did the visualisation of data. Contours and isolines were introduced in 18th century to encode additional information in the images. These visualisations of data were also used for data analysis, much like data graphs and visualisations of today.

Maps were the basis of visualised data until the 18th century when scientific methods increased the use of graphical data to represent information. Such examples include Mulhall’s dictionary of statistics [Mulhall, M].

In the 1960s documents set out rules for graphs, a ‘grammar of graphics’ such as the Semiologie graphique [Bertin, 1967]. These documents were the first theoretical frameworks to guide the assessment and development of charts and data graphics.

### 2.2.3 Definition

According to Edward Tufte data visualisations:

“visually display measured quantities by means of the combined use of points, lines, a coordinate system, numbers, symbols, words, shading, and colour.”[Christopher Ahlberg, B. Schneiderman]

### 2.2.4 Data Visualisation in the Information age

The exploration of large information spaces has remained a challenging task even as parallel hardware architectures, high-bandwidth network connections, large high-speed disks, and modern database management systems have proliferated.

Modern tools such as the aforementioned powerful procession power of modern computers and the wide availability of information from resources such as statues staple and the internet make accessing and examining large amounts of data possible very quickly.
However the same problems remain – how to show this data to the end user in the simplest and most informative manner, if anything these improvements in hardware have led to a reduction in quality of graphical representation as engineers attempt to utilise more and better graphics and use increasing amounts of data.

"It's not about clarifying data, it’s about contextualizing it."

(Aaron Koblin)

Introduction

This section investigates the problems apparent in visualising financial database data and explores some solutions which are weighed on their various merits.

The problems are explored as well as the solutions for both the application and the database management.

Overview

The core function of this project is the presentation of the data in a visual form. In order to encapsulate the most data and present it in the most understandable way the current visualisation techniques must be discussed.

Data visualizations bring themes and ideas to the surface, where they can be easily discerned. (Educause learning initiative)

"Analyse First -
Show the Important -
Zoom, Filter and Analyse Further -
Details on Demand"

B. Sneiderman

Methods for encoding data

“...avoiding catastrophe becomes the first principle in bringing colour to information: Above all, do no harm.”
[Tufte, E]

Colour informs the visualisations with meaning. Used well colour can enhance and clarify, used poorly will obscure, muddle and confuse. While there is a strong aesthetic component to colour, using colour well in information display is essentially about function: what information are trying to convey, and how - or whether - colour can enhance it.

It is possible to rely purely on colour and position to visually represent data. For example to display data in [Curdt et al] a simple colour layering was superimposed on the Bing map of the area to display the database data visually.

However the most important use of colour in information presentation is to distinguish one element from another, a function Edward Tufte calls “to label.”

**Implementation**

According to Joel Lauman:

Data visualizations should communicate data in the most effective way; to truly reveal the data they should be **quick**, **accurate**, and **powerful**.

Schneiderman builds on that with a set of 7 features that are the key to good data visualisation.

1. **Overview**: Gain an overview of the entire collection.
2. **Zoom**: Zoom in on items of interest
3. **Filter**: filter out uninteresting items.
4. **Details-on-demand**: Select an item or group and get
5. **Relate**: View relations hips among items.
6. **History**: Keep a history of actions to support undo,
7. **Extract**: Allow extraction of sub-collections and of the details when needed.
These factors are what make the visualisation of data so useful using current information technology. It can be seen from observing different visualisation method which allow these functions and which do not. Universally the ones that adhere to Schneidermans principals are the most useful and easy to understand, (Eg Gapminder, visualizinglastfm.de) and those that don’t convey little or no relevant information (Aaron Koblin’s flightpatterns.)

The way the data is shown visually is the cornerstone of this project. The main problems are:

- Erroneously seeing correlations.
- Cognitive set-up costs.
- Bias by scatterplot.
- Confusion by hidden labels.
- Difficulties with 3+ attributes
- Errors in correlation questions.

[114x722]These factors need to be avoided in order to deliver the best visualisation to the end user.

2.3 Technologies

2.3.1 Graphing language.

This section discusses the current state of server side visualisation software. There were several factors that had be taken into account in order to select the most suitable software platform. Building on the existing technological solutions was one of the goals of this project.

2.3.2 PHP and JPGraph

The most important technology to use was the graphing platform.
This would implement the central function of visualising the data. JPGraph was used as:

- Open source.

Can be professionally developed without costs of licensing fees.

- Multiple graphing types available.

There are many different types of graph available with JPGraph. Therefore it is possible to easily change between graphing types depending on user input. This provides more control to the user and therefore addition attraction to the project.

- Supports multi-query-graphs.

So that more that one query per entry is available. This provides additional options for comparison to the user.

- Speed.

As speed and power are two of the central factors of good visualisation software this was a major consideration.

- Ease of graph display

As the graph is returned as an image it can be easily included in a HTML front end application. This was on of the main considerations in selection this software.

### 2.4 Rich internet applications

#### 2.4.1 Introduction

The internet was originally designed to allow users access to documents independent of their location. However many users are dissatisfied with the capabilities and performance of today's HTML-based web applications. Users want the functionality of desktop applications with the ease of installation and accessibility that web applications provide. This is what Rich Internet Applications (RIAs)
aim to offer. The use of RIAs is in line with the ideals of “web 2.0” which moves away from the original page based design paradigm towards user centred, distributed and highly interactive web pages.

Google applications comprise some excellent examples of RIAs. In particular, the short history of Google Maps demonstrates technological advances that fulfilled the user’s desire for desktop-application utility in internet applications: The early incarnations of Google Maps had to be downloaded to the desktop to achieve the desired functionality. However, successive upgrades have rendered Google Maps a fully internet-based application with all the functionality of a desktop program. Google docs is another example of a desktop-like application (word processing) which is now available online.

Illustration 1: Comparison of a Google Docs RIA (left) and a desktop application word processor -LibreOffice, (right)

2.4.2 RIA Overview

The best way to demonstrate the difference between a RIA ans a
HTML based web page is to use a compassion between Yahoo maps (2005 version) with Google maps then or now. To navigate in Yahoo maps the entire page is reloaded in a different location. However in Google maps the user is immersed in the application and each new section is individually loaded when required.

2.4.3 Background

The term RIA was coined by Macromedia in 2002, but the idea of a 'thin client' existed before that. The main drivers, leading to the rise of RIAs are:

1. Broadband
A broadband internet connection allows data to be transferred to the client's computer, more quickly than a dial up internet connection. This allows RIAs to download initial “thin client” applications without a prohibitive download duration.

2. Computing power shift
The increased processing power provided by modern computers, has closed the gap between server power and client computer power. This means that software that previously would only perform adequately on big servers can now be performed client-side.

3. Company pioneering
One of the biggest driving forces behind RIAs has been their adoption by leading multinational technology companies, such as Google. Developers are assisted by RIA technologies from large companies, such as Microsoft’s .NET frameworks and Mozilla’s XML User Interface (XUL)

4. Web services and SOA
The Service Orientated Architecture (SOA) and the improvements to web-service development, based on the emergence of XML, for
example XML based RIA language AJAX, this allows the presentation layers to be independent from the underlying logic layers, for example the XML databases containing feature points behind Google maps. This allows the user to be isolated from the computations of the web-services as the front page itself is also isolated.

2.4.4 Technologies used:

The technologies used to create RIAs are predominantly client side, this may lead to longer initial loading times as the application is transferred to the client side, however it accelerates subsequent real time responses and no further connection is needed to the server. For example flash games such as the one shown below have an initial loading time while data is transferred.

Illustration 2: Loading screen for flash game RIA as shown on the right. Note the dedicated loading screen on the left.

However other technologies such as AJAX (Asynchronous JavaScript and XML) facilitate fetching information from the server without reloading the page, as a HTTP based web page would do.
Below is a comparison of the three most popular RIA technologies, which together comprise over 50% of the RIA on the internet.

### 2.4.4 RIA Conclusion

An AJAX style was used in this project for the following reasons:
1. An initial installation was not needed to run the application. Both flash and Java require plug-ins in order to run. Although web browsers like Chrome update and natively support flash this was an added complication that wasn't desirable for this application. As this project is user-centric and aimed at a non technical audience, all technical requirements were kept to a minimum. The benefit gained from designing and excellent HCI for novice users would be undone if this users base was alienated by being unable to load the application. As an AJAX style doesn't require client-side downloads this was one of the reasons this approach was adopted.

2. A specific language was required for the visualisation in this application. In order to successfully implement the graphical layer and therefore the core functionality of the project; correct graphing technology was needed. Using an AJAX system meant that any server side technology could be used.

3. A three tiered implementation structure could be used. As stated earlier in this chapter this was the architecture chosen for this project. While an XML returning data-layer is supported by all three approaches, only JAVA and AJAX facilitate a defined middle layer.

4. Graphical technology doesn't require visual manipulation in the web page. The AJAX approach relies on the native support of the browser for graphical representation. This potential issue was resolved by the nature of our graphical layer – an image object is returned that can be displayed in a HTML front end.
2.5 Human Computer Interface.

2.5.1 Introduction

Human Computer Interfaces (HCIs) seek to discover the most efficient way to design comprehensible electronic messages. It involves the study, planning and design of the interaction between people and computers. The main focus of the HCI is the user interface as this is the focal point of the connection between the computer and the user.

The possibilities of computers, and therefore any application on them, are limited not only by their power to compute, but by their power to communicate with their human users. This is of great importance to this project as it aims to effectively reach a non-technical user base.

2.5.2 Background

This field of study was first seriously considered at the same time that computers became commercially available for non technical users, as they required a positive and intuitive HCI. This was in contrast with the academics and professionals who had dominated computer use before that point.

Donald Norman is a psychologist who pioneered research into HCIs. He took inspiration from industrial product design, the standard of user friendliness at that time. According to Norman, design should:

- use both knowledge in the world and knowledge in the head. Knowledge in the world is overt - we don't have to overload our short term memory by having to remember too many things (icons, buttons and menus provide us with knowledge in the world). On the other hand, while knowledge in the head may be harder to retrieve and
involves learning, it is more efficient for tasks which are used over and over again.[HARTSON and HIX]

Designs should also "make it easy to determine what actions are possible at any moment (make use of constraints)"[Keim].

For example, menus should only display the actions which can be carried out at that time.

Furthermore, HCI designs should "Make it easy to evaluate the current state of the system" [Keim]. For example by providing feedback from buttons.

Finally a HCI should "Follow natural mappings between intentions and the required actions, between actions and the resulting effect; and between the information that is visible and the interpretation of the system state" [Keim]. This means it should be obvious what the function of a button or menu is. Therefore, - common conventions settings and formats are to be used in this system. For example hyper-links are underlined.

**2.5.3 Key Concepts of HCI**

There are several important concepts of interface management:

1. Dialogue independence
   This is the keystone concept upon which all the other concepts depend. It is a characteristic that separates the design of the HCI from the design of the software components of an application. This prevents modifications in one area or layer causing changes in the other.

2. Structural modelling
   The HCI serves as a framework for understanding the elements of interfaces and for guiding the dialogue developer in their construction.
3. Representation
The Human Computer Interface is accomplished by a variety of notational schemes for describing the interface, mainly the User Interface (UI). Therefore the representation of items in the UI for the user to digest is one the key passive concept of the HCI.

4. Interactive tools
The variety of interactive tools for human-computer interface development that are available free the dialogue developer from much of the tedium of hard coding dialogue into their applications. For example there are JavaScript inputs that follow all of the HCI requirements such as the ability to be active or inactive and provide feedback when users select them.

5. Rapid prototyping
This provides the ability to observe the behaviour of the interface while it is still under construction – before design choices are concrete. This increases communication among system designers, implementers, evaluators, and end-users.

6. Development methodologies
These are methods used for interactive system development that consider interface management to be an integral part of the overall development process and give emphasis to evaluation in the development life cycle.

7. Control structures
These govern how sequencing among dialogue and computational components is designed and executed.

2.5.4 Implementing a HCI
There are two overall governing ideals behind creating a usable HCI.
These are recognising the diversity of users and the importance of preventing errors.

1. **Recognising Diversity** requires taking into account the type of user operating the system. Be it complete computer novice, computer literate user or expert and frequent user. Each type of user expects the screen layout to accommodate them and allow them to perform the tasks required. Beginners need obvious labels and layout while experts need the ability for fast and extensive use.

2. **Error prevention.** Errors must be prevented wherever possible as this negatively affects the users opinion and use of the interface. Actions such as organizing menus functionally and preventing irreversible user choices can solve error issues. Expecting users to make errors and attempting to anticipate them is important for error prevention.

There are several more relevant rules for creating a good HCI, including Shneiderman's eight Golden Rules of Interface Design:

1. Strive for consistency
   - Identical terminology should be used in prompts and menus for example.
   - Colour, layout, capitalization, fonts, and so on should be constant throughout the application.

2. Enable frequent users to use short cuts
   - To increase the pace of interaction

3. Offer informative feedback
   - For every user action, the system should respond in some way; for example buttons changing colour.

4. Design dialogues to yield closure
   - Actions should have a beginning a middle, and an
end.
5. Offer error prevention and simple error handling.
   • Design the form so that users cannot make a serious error; for example, prefer menu selection to form fill-in and do not allow alphabetic characters in numeric entry fields.
   • If users make an error, instructions should be written to detect the error and offer simple and specific instructions for recovery.
6. Permit easy reversal of actions
   • Cancel buttons and back buttons
7. Support internal locus of control
   • Experienced users want to be in charge. Surprising system actions, tedious sequences of data entries, inability or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction with the system
8. Reduce short-term memory load
   • Studies suggest that humans can store only 7 (+/-2) pieces of information in their short term memory. It is possible to reduce this by designing screens where options are clearly visible, and don’t require the user to remember a list of previous steps taken.

2.5.6 HCI Conclusion

The best way to sum up HCI is to once again quote Norman, HCI should: "In other words, make sure that the user can figure out what to do, and the user can tell what is going on."
2.6 Case Studies

There are two systems that are relevant to the current state of the art of visualisation systems. These are Polaris [Stolte C,] from Stanford University and [Gapminder] produced by Google.

2.6.1 Gapminder

Illustration 3: Gapminder RIA

Gapminder implements a similar RIA that this system aims to produce. It has a main visualisation section that changes independently of the rest of the page, discussed above. The layout of the control panel, in standard method with consistent labelling as highlighted in Shneiderman's eight golden rules in the HCI section above. This clean and usable interface can provide inspiration for this
The graphing function uses balloon charts to encode as much data as possible – using colour to label points, as suggested by Keim. The size of the balloon display data for multiple points as well as the X and Y axis positions. The graph changes over time, to represent an additional axis. This functionality can be useful but wouldn't be used in this project as it provided difficulty mining the date.

This application also mines data from independent resources – Excel spreadsheets – but does not have a specific user base. Although the design of the HCI and aims appeal non-technical users. Lectures and teachers are Gapminders target audience rather than the historians this system is aimed towards. The lack of specialised user knowledge and mining capabilities are two are the main motivation differences between this system and Gapminder.

2.6.2 Polaris
Polaris is not a web application, although it is data independent. The aim is again slightly different, their emphasis is on very specific technically proficient users that want an extensively customised visualisation of complicated data. This is a very powerful tool but lacks the HCI and RIA of Gapminder, the lack of colour in the visualisation reduces the impact of the visual and the labelling ability.

### 2.6.3 Case Studies Conclusion

Both of these modern approaches successful sole the visualisation problems in very different ways with distinct different user aims.

This system aims to incorporate inspirations from some elements of both of these current solutions to visualising data, and build on them.
3 Design and Implementation

3.1 Chapter Overview

3.1.1 Introduction

This chapter details the steps taken to design and create the final application. In addition to describing each phase of the design and implementation this chapter will outline the requirements that the system must obey, the overall structure of the system and the method that was followed to include each addition to the system.

The aim of this section of the project was to build an system that fulfilled all of the requirements set by the users and by the research questions using the techniques and technologies researched in the start of the art. This will result in an application that can customise graphs from simple user inputs such as buttons rather than typed input. There should be functionality that allows academics to use the software to assist with research, so additions such as supporting citations should be present.

3.1.2 Structure of this Section

The structure of the design and implementation of this project was split into three phases, each containing two stages.

Each phase is a distinct body of work, with a separate emphasis. Phase one concentrates on the rapid prototyping and building of specific design requirements, phase two concerns the development of the results from phase one into a stable application and phase three focuses on final alterations to solve bugs and further tailor the system to the desired user base.

Each stage is a self-contained design and implementation
iteration. Development work is done, presented in a user meeting, and further design plans evolved from the implementation or user feedback are set out for the next stage.

This organisation allowed constant feedback; provided in regular meetings with the users in order to demonstrate the changes made to the application. The frequency of the user meetings reduced the length of the feedback loop in order to implement changes and additions early in the design process.

3.2 Requirements

3.2.1 Requirements from the Research Question.

The core goals were set out in the research question; these are the minimum requirements that need to be implemented in order for the project to be is successful. Below is a list of requirements from the research question:

• **RQ Req 1:** The interface should be displayed as a web based application.
• **RQ Req 2:** The interface should connect to a resource of historic records, specifically the statues staples database.
• **RQ Req 3:** The interface should have a visual component displaying the information in a graphical format.
• **RQ Req 4:** The system should make it possible to query the database to create different visualisations.
• **RQ Req 5:** The system must have ability to explore data-mining, and return pure tabular information from the database.

3.2.1 Requirements from the State of the Art.

There were several design features that were produced from the state of art research. This was a great addition to the design of the
system as there were some concrete targets to hit from the start of the design and development cycle.

Most of these targets were concerned with the final rich internet application and the human computer interface that was to be included in the RIA. Below is a list of requirements from the state of the art.

• **SOA Req 1: Architecture:** The Service orientated Architecture of the product, including the three tiered web architecture was directly influenced by the research into RIAs. The application layer would be a presentation layer and therefore would be isolated from the logic of the computation and visualisation layer. The use of XML to link the layers was also inspired by the research into the standards of RIAs.

• **SOA Req 2: RIA implementation** Not reloading the application page is a feature of RIAs which should be incorporated into the application layer. This objected will affect much of the implementation of the application page and how it interacts with the graphing layer.

• **SOA Req 3: Feedback from application:** It was necessary for each user input action to return feedback to the users, as outlined by Donald Norman in his work on HCIs. This is the origin of the idea to include dynamic button input responses.

• **SOA Req 4: HCI application UI:** Most of the key concepts of HCI were implemented in the design of this project. Most notably the dialogue independence between the layers and the languages used in this project as well as iterative tools and rapid prototyping.
3.3 High level Architecture

In accordance with SoA requirement 1 the system is split into three distinct levels, the application layer, the graphing layer and the data layer.

**Application layer: (AL)** this is a presentation layer as described in the state of art. In accordance with SoA requirements 2 and 4 this is a rich internet application that follows the guidelines of a good human computer interface. In order to do this it must not refresh after user input.

This layer will provide the user with a portal to the functionality of the other layers. The aim of this layer is to present the information and customisation options to the user in a easy to use manner.
**Graphing layer (GL):** The function of this layer is to provide a graph image or table of tabular data to the application layer. The graphical layer takes in parameters from the application layer and queries the data layer for the necessarily values in order to populate the graphs or tables. This layer provides the core functionality of the project – the visualisation - to the application layer.

**Data layer (DL):** takes in http “get” requests and returns the desired data in XML format. The DL connects to the Statue Staple database in order to extract the information. It is also responsible for cleaning the data and ensuring that no false data is returned.

**3.4 Methods of implementation:**

For new functionality a consistent method of implementation was applied. This consisted of the following steps:

1. An example of the functionality was prototyped on its own, to test where it would function as desired.
2. The prototype was customised with certain inputs, in order to further test whether it would provide the correct usability.
3. This advanced prototype was then attached to any dependant parts, and the methods for implementation were finalised and tested.
4. This final prototype was then integrated into the overall system. Which would then be tested in order to ensure that all the functionality – that was previewed on its own – worked as desired with the rest of the system.

**3.5 Phase 1 – initial design.**

This was the major design phase in which the overarching structure and basic wire-fame of the project was constructed. This is the section most similar to a standard design phase. The architecture
for the wire-frame was drawn from SoA requirement 1, as was the choice of the graphing technology. Images that could be loaded in the application layer without refreshing the page were used in accordance with SOA requirement 2 and taking into account the SoA of visualisation technologies.

### 3.5.1 Stage 1 – First wire frame and Technology Demonstration

**Introduction**

In this stage the main server side technology was finalised and an example of the graphing utility and a basic wire-frame of the proposed end product were demonstrated to the expert users. The database was ported to a usable database management system and integrated with the graphical layer.

A demonstration graph was used to show that it was possible to extract data from the Statues Staples database. This was a rapid prototype of the main functionality of the project. It was important to do this in order to show that there would be no completely prohibitive problems and that the main goals were achievable using this technology and approach, in compliance with concept 5 of HCI development as stated in the SoA.

**Goals**

1. To finalise the desired output application with expert end users.
2. To prototype the method intended to visualise data and demonstrate it to the users.
3. To convert the resource from Microsoft Access (MS) to a accessible database format.
Goals achieved and Methods used:

1. In the user meeting several key plans were demonstrated in a presentation to the users:
   1. Data mining by selection of the area on the graph to mine.
   2. URL based Get requests in order to pass information to the graphing layer
   3. URL based citing support of the graphs. This means that a certain URL will always replicate the exact same graph from the graphing layer – this is due to the point 3 above and an important function for the end user.
4. Three layered web architecture.
5. Ticker tape list of old graphs

Illustration 8: Ticker taping as initially presented to users

2. The rapid prototyping of the graphing functionality of the data
was competed in three stages:

1. The production of a graph in a web browser. In order to satisfy RQ requirement 1 the graph must be contained in a web based application.

2. The graphing of user generated data. In this step the user data was extracted from the database manually using a custom SQL query - the data entries were hard-coded into the graphing layer. This was to show that graphs could be created using the type of data found in the Statues Staples database.

The following is the PHP JPGraph function that displays hard-coded data in order to produce the above graph.

Illustration 9: Radio graph of geographical data manually extracted from the statues staples database

The following is the PHP JPGraph function that displays hard-coded data in order to produce the above graph.
4. In this case RQ requirement 2 was fulfilled by directly accessing the database from the graphical layer. This short cut was necessary in order to prototype the use of the correct data before the data layer was implemented. Therefore the data was extracted directly using a PHP database access.

5. The provision of custom queries from unique http get requests in the graphing layer URL. This was done on the temporary SQL query in order to test that custom querying from the application layer was possible, as per RQ requirement 4. This customisation was demonstrated to the users using URL manipulation to change the resultant graphs.

3. Porting data from MS Access format into usable database software.

This was completed by converting the MS access files to CSV files and then importing them into the MySQL database. The MS Access files were opened using Microsoft Excel 2010, and then saved in a comma separated format. This type of file can be loaded into a MySQL data base. However the date format is different in the two database managers. This required using a function in excel to change

```php
// Setup graph titles
$graph->title->Set("Debtors and creditors per region");
$graph->title->SetFont(FF_FONT1,FS_BOLD);
//set y values
$graph->SetTitles(array("Dublin","Belfast","Kildare","Cork","Other","Louth"));
// Create the radar plots with values from the array
$plot = new RadarPlot(array(intval($row[0]),intval($row[2]),intval($row[1]),intval($row[3]),intval($row[4]),intval($row[5])));
$plot->SetLegend("Debtors");
$plot->SetColor("red","lightblue");
$plot->SetFill(true);
$plot->SetLineWeight(2);

Text 1: PHP graph creation, using hard-coded values
the format of the date columns and then cast them to date object in
the SQL import function.

The two most important design decisions made in stage 1 were
the decision to use PHP as the major server side language using the
JPGraph as the graphing utility and the main design of the application
as shown in the wire-frame.

The decision to use the JPGraph library was probably the most far
reaching technology decision of the entire project. Implicit in this
choice was the assumption that this library provided all of the
functionality that would be required by the graphing part of the
project. This decision committed the way in which the graphs would
be produced, in HTML image format that could be displayed in the
application layer.

**Design Outcomes:**

1. Number and type of graphs required.
In the demonstration there were five types of graph shown. Bar, line,
scatter, pie and radio. It was decided from both the users and from a
technology simplicity standpoint that it was only necessary to provide
the first three graphs options.

  ◦ User motivation:
The application was not intended to graph 100% of any variable. The
data was entirely independent of each other for example the number
of transactions per year. Therefore comparing them to each other
and to the total – as a pie graph was unnecessary. The same
argument also applies for the radio graph type.

  ◦ Development motivation:
The development argument against using these two graphs was
based on the aim to have data mining propagating from user access
to the graph image. This could be done in a standard method for bar, line and scatter graphs by splitting the image into distinct vertical “slices” depending on the amount of x axis items on the graph. This approach wouldn't work with radio or pie graphs as they are cyclical. Therefore this would require a completely different approach for these graph types. While viable, the limited usefulness of having these two extra graph types in the application was not considered worth the extra work required to make it feasible.

1. Porting data from MS access format into usable database software.

2. Database data access:
   ◦ This functionality and the method for doing so - user iteration with the graph image - was decided. This was prototyped directly from the graphing layer.

   **Implementation outcomes:**

   1. The wire-framing of the project was completed.
   2. The technology for the central function of the project, the graphical visualisation, was finalised and a rapid prototype was implemented which proved that this technology could be used without prohibitive errors.
   3. The MS Access resource was loaded into a MySQL database. The original format was kept rather than use a view that joined all of the tables, in order to reduce the manipulation of the primary resource as much as possible.
   4. This is an example of the graphical layer as shown to the users in the presentation.
3.5.2 Stage 2 - Rapid prototyping of core functions

Introduction

This stage focused on the basics of the application layer, getting a rapid prototype working in order to prepare for further execution of similar functionality knowing the core concepts were sound.

It built on the results of the previous stage. Further implementation aims, resulting from the design choices in stage 1, were prototyped or explored. The set up and customisation or URL based graphs were also further explored in addition to implementing a distinct data layer to mine the States Staples database.

Goals:

Illustration 10: Graphing amount data, the GL at the end of stage 1
1. Wire-frame the functionality of the core user inputs:
    1. Seed the result of the GL in a RIA framework with HCI enforcement.
    2. Basic queries to the database layer from the AP propagating to the visualisation.
    3. Manipulation of the graph type from the AP.
2. Extend the communication between the layers, through HTTP get requests.
3. Prototype templates data retrieval from the data base.

**Goals achieved and Methods used:**

The initial stages of the application layer were completed. The following tasks were completed

1. Displaying the graphical layer in a Html-based application page, as per SoA requirements.
   This solidified the relationship between the application layer and the graphical layer. The graph was displayed as an HTML image element contained in the application layer with its source as the URL of the graphical layer URL, the PHP file.
2. Formatted the graphing layer improved customisable URLs, the address of the graphical layer script with the custom options as HTTP get requests.

For example the following URL:

```
http://localhost/sqlGraphBasisMultiWithMultiGraphsFieldSelect.php?
st=1600&DC=0&g=0&d=0&title=Custon%20Graph
%20Name%X=0&numFeilds=1&SQL=0&yearsNum=10
```

encodes this information to the graphing layer and is in turn passed to the data layer.
<table>
<thead>
<tr>
<th>URL feature</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>sqlGraphBasisMultiWithMultiGraphsFieldSelect.php</td>
<td>URL of the graphical layer</td>
</tr>
<tr>
<td>st=1600</td>
<td>Start date is 1600</td>
</tr>
<tr>
<td>DC=0</td>
<td>Debtor option is to query both debtors and creditors</td>
</tr>
<tr>
<td>g=0</td>
<td>Graph type to be produced is 0 (bar graph)</td>
</tr>
<tr>
<td>d=0</td>
<td>Data of graph option is 0 (graph rather than tabular data returned)</td>
</tr>
<tr>
<td>title=Custon%20Graph%20Name</td>
<td>Sets the title of the graph</td>
</tr>
<tr>
<td>numFeilds=1</td>
<td>Set the number of fields used to 1</td>
</tr>
<tr>
<td>SQL=0</td>
<td>SQL query is of type 1 (amount)</td>
</tr>
<tr>
<td>yearsNum=10</td>
<td>Number of years to be displayed is 10 (therefore from 1600 – 1609)</td>
</tr>
</tbody>
</table>

These are the values that are altered by the user iteration with the JavaScript; for example when the user selects the button to change the graph type to line – the URL is changed so “g=1” and then the image is reloaded.
Illustration 12: graph with get requests: st=1600&DC=0&g=0&d=0&title=Custom Graph Name &X=0&numFeilds=1&SQL=0&yearsNum=10
3. Introduced prototyping of user tools to control and customise that resultant graph as discussed above, prototyping the HCI intergeneration in the web application.

The user inputs competed in this stage:

1. **Start date** – text input
2. **Graph type** – drop down
3. **Debtors, creditors or both** – drop down

Illustration 13: different graph with settings: 
\text{st=1620&DC=1&g=2&d=0&title=Changed}\%20Graph&x=0&numFeilds=1&SQL=2&yearsNum=20
4. Image map – shows which area is being selected. This was modelled by loading the image from the graphing layer into an image map that had mouse-listeners on different sections. The sections were split into horizontal sections so that the user can mine a particular year in the graph. A mouse listener was used to show which section is being highlighted and a mouse-clicklistener that would pass the selected section number to a function that displays the tabular data. This wasn’t implemented in this stage – only the listener that showed this method would be successful and could be pursued further.

Illustration 17: display of area selected on graph

5. Show URL on page for direct access to the graphing layer. This was the first step to citation support. It was a simple HTML function to display the current URL on the page, but provided important information to the users.

Illustration 18: The URL field

changeURLValue(){
    document.myform.URL.value = URLvalue;
}

Text 3: JavaScript function to update URL field with current URL

JavaScript function that updates the URL field of the application this is called every time the URL value is updated.

Outcomes

The main objectives completed in this stage were the prototyping of the interaction between the HCI and all of the different layers as
well as the basics of querying the data and graphing layer from the application layer.

Choosing the graph type, start data and type of person were the basics of the 3 different types of querying that were built on during the implementation phase. This meant there was a lot of communication between the layers, in order to get the correct data from the database and return the correct graph to the application layer.

**Outcome:**
Design outcomes:

1. **User Interface**: as a result of the work in this stage changes to the UI were outlined. In order to make the application more user friendly. This led to a new layout and eventually the use of JQuery to improve the UI.

2. As much of the core features had now been prototyped, customising the data and graph shown, more advanced
features were designed in further detail – now that goals were proven to be achievable with the current approach. The two main functions were the ticker taping of the graph and data mining as it was now possible to accurately detect which area on the graph the mouse in pointing to.

3. The type and amount of queries that would be included were finalised and roughly prototype within wire frames of the necessary frameworks.

**Implementation outcomes:**

While the user functionality from this stage wasn't extensive it covered most of the core interactions and could be built on in subsequent stages in the implementation phase.

1. The ability to use an independent RIA application was another important result. This meant the current plan would follow the SoA requirement.

2. As a result of this stage the main implementation phase could begin, sure that the designs and goals were possible to complete.

**3.6 Phase 2 Main Implementation phase**

**Introduction**

This was the major implementation phase, where most of the functionality was coded. The rapid prototyping from phase one was used as a base. The application was transformed to a stable product that fulfilled most of the requirement resulting from phase 1. The major design decisions in this phase resulted from the implementation rather than user feedback.
3.6.1 Stage 3 - Implementation of Graphical and Data layer functionality

Introduction

This stage predominantly focused on the graphical and data layers. Most of the functions for these layers - outlined in the wire-frame in stage 2 - were implemented in a meaningful and stable manner, expanding on the limited nature in the prototypes. This was the first stage with significant stalls due to development problems.

Goals:

1. Change the layout to better HCI application design
2. Implement data-mining
3. Create and exploding data-mining section on the AL
4. Provide the back end functionality to selecting sections of the visualisation.
5. Implement ticker taping.
6. Expand querying
   1. Provide access in the AL.
   2. Provide relevant querying and data cleaning in the data layer.
7. Support more that one query per graph.

Goals achieved and Methods used:

Application layer:

1. Layout change:
   In order to follow SOA rule 4, the “control box” containing the user inputs were reorganised to the left of the graph. This was also in order to make the ticker tape and the data display appear more natural.
2. Ticker taping – implementation 1
The first ticker tapping was implemented. This was a shallow implementation where the last graph and URL were appended onto the bottom the document using the JavaScript document objects.

3. Choose the different queries – implementation 2 – using drop downs
4. Getting tabular data – exploding mining area
5. New window for image button implementation
6. Multiple query’s on the same graph.
7. Produce a separate pop up window to set up the download of the CSV files
8. Add drop down
9. Allowing more than one query per graph was the biggest and most difficult change in the data layer. The arrays of point values were changed to two dimensional arrays which the data layer filled by mining the database multiple times, once for each query. The graphing functions had to support these two dimensional arrays and add and additional entries to the graph for each array.
function makeBarGraph($data,$labels,$years){
    //....code to set up bar graph
    // Create the bar plots
    //initialise an array of Bar Plots
    $points=array();
    //step through the data array and make a new Bar Plot for each data point
    for($i=0; $i<count($data); ++$i){
        $points[$i] = new BarPlot($data[$i]);
        $points[$i]->SetColor( $colours[$i]);
        $points[$i]->SetLegend($labels[$i]);
    }
}

Text 4: Data interaction in makeBarGraph to support multiple queries

Design outcomes

1. Ticker taping: the first edition provided the functionality needed, however the users required more control of the ticker – in case they wanted to print the entire page of graphs. This meant that instead of a single graph a control panned was designed to be tickered – with the option of detention, the time of creating, the unique URL and a input field for notes.

2. Modify query selection: The users found the current method of selecting queries to be cumbersome and awkward, so a new method using check-boxes was designed to solve these concerns.

3. Layout: The current layout was an improvement – however for the final application it needed to be more “web 2.0” and more styling was required, especially on the bare JavaScript inputs.

Implementation outcomes

1. Queies: all of the querying was implemented as desired, in the
data and graphical layers. Therefore the largest part of the project was completed – the core implementation of the visualisation.

2. Multi-query-graphs: this was a main change to the data layer and had a sizeable result as it provided an order of magnitude more information per graph.

3.6.2 Stage 4 – Application layer reimplementation

Introduction:

The second implementation stage purely concentrated on the application layer as the UI needed a complete overhaul. This required improving the existing graphics and functionality in addition to inserting new inputs using new technology.

Goals:

1. Added end date.
2. Input a graphical method to change the data.
3. Organise the user buttons.
4. Change the input graphics.
5. Facilitate deletion of ticker tape books.
6. Implement functionality to invert the x-axis on the graph.

Goals achieved and Methods used:

1. End date:
   An end date option was added using text box on the application layer. This passed on to the data layer as a high date to be used in the query, the high date had previously been the start date plus ten.
2. Improved ticker tape – deletion
   This was one of the most challenging tasks of the entire
Many implementations were tried, the main source of errors was the organisation, linking each entry to the others in order to be able to delete a single entry. A linked list would have been very useful but unavailable with the technology used. In the end a table-like structure was used. Originally empty, each entry was a new “row” but could be deleted without destroying the table.

In order to add functionality the “new row” function wrote pure HTML to the page, as using JavaScript objects didn’t allow the implementation required.

```javascript
number_of_images = 0;
$('div').click(insertNewImageRow);)
function insertNewImageRow() {
    refresh;
    number_of_images ++;
    tableVals[rowCount]= rowCount;
    $('td').append('<tr> ' +
        '<img src='+url+' border="0" / > <td>' +
        '<td> <h2>Graph number '+number_of_images+' </h2> '+
        '<textarea id="txtArea" rows="2" cols="20">This is row id'+rowCount+' which is in position'+tableVals[rowCount] +'</textarea><p>' +
        '<textarea  id="txtAreaSource" rows="2" cols="20">'+url+'</textarea><p>' +
        'Created at '+getTime()+'<p> '+
        '<button type="button" onclick="delRow('+rowCount+')">Del row</button>'+
    '</td> </tr>
    rowCount++
}
```

**Text 5: Function to draw new table entry in pure HTML**

This was poor programming practice but the only way to produce the required results, including additional feature of the ticker tape.
Note the “Delete row” button is a standard JavaScript button due to these restrictions.

As seen above the extra features added to the ticker tape are:
1. Notes section
2. Time created
3. Change implementation

3. Graphical changes – bringing in web 2.0
   This was done in order to comply further with SoA requirement 2 and improve the HCI by eliminating the need for typed user input as per HCI requirements.
1. Date picker:
2. This was the first attempt to replace the text input, this was too specific, and took too long to input the desired variables.
3. Slider

Illustration 21: Slider
This was the second and more successful graphical representation of the start date. This was based on a widget
from JQuery, customised to this application with an upper and lower date limit and resultant changes to both the displayed start date and the stored start data value to be passed to the graphical layer.

```javascript
// Slider instance
$( "#slider" ).slider({
  range: "max",
  min: 1597,
  max: 1687,
  //Initial values: [ 1600, 1610 ],
  //Method definitions
  slide: function( event, ui ) {
    $( "#amount" ).val( ui.values[0] );
    $( "#end_Amount" ).val( ui.values[1] );
  }
});

//User input results
$( "#amount" ).val($( "#slider" ).slider( "values", 0 ) );
$( "#end_Amount" ).val($( "#slider" ).slider( "values", 1 ) );

Text 6: JQuery Code that implements the slider
```

4. Change to the implementations of the following sections of the application layer:
   1. Buttons
   2. Drop downs
   3. Titles
   4. CSV window

5. X-axis inversion:
   This was the main cross layer addition in this stage.
   This was a method to all users to observe the data in a new way. It changes the way the data is visualised in order group the information by the query rather than the time. In order to implement this the order of the arrays of data for the x and y axis are inverted in the graphical layer as they are passed to the graphing functions. This approach limits the changes to the
data layer by manipulating the same data in a to produce a result that should have required a different data layer query.

if ($Xorien != 0) {
    makeGraph($BigArray, $years, $labels);
} else {
    makeGraph($BigArray, $labels, $years);
}

Text 7: Orientation JavaScript check

Illustration 22: Comparison of inversion of the axis label. Left has date on x-axis and right has query on x-axis.

As you can see from this image trends in the years for particular queries can be seen easier with inverted x-axis.

**Design outcomes**

1. Another point on the slider was required by the users in order to control the end data as well as the start date visually. This would be in compliance with SoA requirement 2.

**Implementation outcomes**

1. The majority of the outcomes of this stage were restricted to the UI and the application page. The UI received a completed overhaul and new inputs (slider, data-picker, as well as a facelift to the existing inputs (using JQuery) and the ticker
(using deletable boxes) were the main changes in this sage. There were many implementation outcomes from this stage, this was the beginning of the end; with an system that could be distributed.

2. X-axis rotation: this was the main change that affected all three layers.

All the major goals had be completed – at the end of this phase the changes to be made were small and aimed at tailoring the application to the specific user base.

Illustration 23: Application at the implementation phase

3.7 Phase 3 - Refinement

Introduction

This was the final phase, the main targets were completed resulting in an application that could be released. However as a user-centric project it was important to tailor the final product to the specific end users. This phase had the most user interaction as every effort was made to deliver the best and most suitable application to
the expert users in the evaluation phase.

The design and implementation choices were small and mainly minor, as the “heavy” implementation and designs was completed in the previous two phases.

3.7.1 Stage 5 – Improved Query Organisation

Introduction

In the first refinement stage after the much changed system was demonstrated to the feedback users. The biggest issue was changing the way the specific queries were selected – going from a number of drop down menus to the more user friendly selection of check-boxes for the user to select. This was on the face of it- a fairly minor cosmetic change, however it lead to a new get request and changes in the graphical layer as well as complex JavaScript manipulation.

Goals:

1. Improve end date selection.
2. Change the query selection in the AL.

Goals achieved and Methods used:

1. putting 2\textsuperscript{nd} point on the slider
   This was in order to let the user easily control the upper date – allowing easier mining from large date sections to small period.
   This meant that the add year/less year buttons had to affect both sliders and that the end point needed to connect to the end date in the same way the start date did.
   The JQuery slider object has point object so adding an endpoint
was trivial. The most difficult part of this task was syncing the changes of both upper and lower limits – especially as the slider object was changing this from JQuery and other inputs – like the text-box were JavaScript based.

2. Putting in a different method of selecting the different fields

Illustration 25: The three query option sets

This was the one of the most challenging AL tasks, specifically changing the check-boxes for each query selection. IN the end, the boxes are deleted and re-drawn for ever query select. This was further complicated by the actions the check-boxes had to perform – each had to pass a variable that was eventually used by the graphing URL. The different number of fields for each query were an added complication.

A design and implementation choice was made to keep selected boxes selected throughout a refresh of check-boxes. It would have been easier to reload with only the first field selected. A local list variable was kept with the selected box numbers. This was queried foe each reload.

The checked box numbers were now passed to the GL as comma separated variables, the number of queries is calculated from this by the GL.


This was a trivial aesthetic change.
Design outcomes

1. An additional query selection was required in order to select the total of the selected queries. This was needed in order to
2. A new title was required in order to highlight the meaning of the system. It needed to reflect to aim of the application rather than just a description of the system.
3. The addition of a library selector was a User based design change. This would allow the user to select the source that the data would come form. It was an additional query field that was discounted earlier as the importance was misjudged.

Implementation outcomes

During the implementation it was found that the total could contain additional functionality if it totalled the selected queries rather for the gross
The system was in a position for final user reviews after some final user-based additions as the new ticker tapping was completed and the redesign of the control panel was complete.

3.6.2 Stage 6 – Final Alterations

Introduction

This was the final stage and the emphasis was on perfecting what was currently contained in the application. There were no large change in the UI or a drastic change to the functionality. There were several tweaks in order to provide final alterations, including 2 additional functions resulting from user meetings.

The main aim was to get the project to the point where it was a stable platform that could be presented to the end users for evaluation.
Goals:

1. Add library change query.
2. Add total check box and functionality
3. Regression test
4. Fix bugs

Goals achieved and Methods used:

A lot of the time in this stage was concentrated on error capture and prevention. This regression testing session aimed to capture and fix all of the bugs in the software before the system went “live” to the evaluating experts users. The entire program was broken up into sections and a grid was drawn up in order to sufficiently analyse that quality of the application. A grid was used in order to ensure that the entire application was covered and that all functionality was tested. It also helped

After this regression testing session was completed, live tests were performed. This involved letting another random selection of users “play” with the product. The aim was to allow complete new comers to find errors which would not normally occur to people who developed or aimed to use the application for its intended use.

The ultimate structure was finalised by accurately tying together the three distinct levels of the architecture into the finalised product

The extra functionality required was the addition of the total check-box which calculated the total amount of the selected inputs and another query option – the ability to define the library the results are selected from.

This provides a lot more power to the user as they can effectively customise a query from by adding existing queries, while retaining the ability to calculate the actual total (by selecting all the queries
and the total.)

For example if a user want to see the amount of transactions between 100 and 300 then they would have to select the options for 100-200, 200-300 and total.

![Illustration 27: Example of graph with total](image)

As a result of user evaluation meetings ALT 20 was implemented, where library query options were added.

- Robustness – error capture
- Implementing in with the back end (plugging into the XML etc.)
- putting in the total option
- Options to select libraries.

### 3.8 Testing

#### 3.8.1 Regression Testing.

This was done on the entire system – all layers of the front end the back end. The main areas to test for each of the parts was

- Normal use
• Stress testing (maximum entries)
• Edge case
  ◦ Minims
  ◦ Maximum values
  ◦ Null values
  
This was done on the application layer first in order to get all of the obvious, and most likely to occur errors caught and dealt with.

Main errors found and fixed:

3.8.2 Application layer Error 1

Scenario: Changing the x axis for line and scatter with only one query selected.

Result: Originally this caused an null graph to be produced as there was a line/scatter graph with only one entry on the x-axis.

Solution: Two solutions provided:

1) When this scenario appears, a error box appear, warning the user that an additional query needs to be selected.

2) Automatically duplicate the query if this scenario arises. This means having an if function in the code to check for this exception.
Solution 1 was first produced, however this reduced the usability of the application, restricting the use of the inverted x-axis to queries of 2 or more for line and scatter graphs. Solution 2 was devised in order to solve this second problem. Note the URL for the graph image:

http://localhost/sqlGraphBasisMultiWithMultiGraphsFieldSelect8.php?
num=10&us=1&st=1600&DC=2&g=1&d=0&title=new&X=1&numFields=4&SQL=0&yearsNum=10&lib=2&string=0,0&totalLab=0

There are two inputs for the value get, even there is only one query selected.

Illustration 28: Line graph with single duplicated query

The JavaScript exception catching code:

```javascript
if(graphType!=0 && orientation==1 && numberChecked<2){
    SQL = copySQL();
}
```

Text 8: The JavaScript exception catching code:
Which calls this function:

```javascript
function copySQL()
{
    for(var q=0;q<isChecked.length;q++){
        if(isChecked[q] == "true"){
            numberChecked++;
            SQL = SQL +","+q;
        }
    }
    return SQL;
}
```

### 3.8.3 Application Layer Error 2:

**Scenario:** No queries selected.

**Result:** Error caused as there were no entries in the graph

**Solution:** Create a exception catcher in the graphing layer and the application layer. The application layer exception catcher prompts the user to select a query and doesn't call the graphing layer. The exception in the graphing layer cuts out the input, so no error is produced.

![Illustration 29: Example of the exception catching in the application layer](image-url)
3.8.4 Data layer Error 1:

**Scenario:** Details of the MySQL database included in the script which returns the XML. This is poor for the security of the database as well as bad programming practice.

**Result:** Keeping the password, root user and name of the database in the same script means it is difficult to switch between databases and that these important security details can be found with relation to the information they protect.

**Solution:** Create a different PHP file where the root name, password and database name are stored and import it into the file.

```php
<?
$username="root";
$password="simon";
$database="fyp";
?>
```

*Text 9: PHP script containing the details of the MySQL database*

**Design outcomes**

There were no design outcomes that weren't covered in this stage as this is the final design stage. An design decision that weren't completed weren't covered in this project brief and will be covered in further work.

**Implementation outcomes**

The application was in a position that all the required functionality was implemented and the system was stable. This meant application could be tested against initial goals and by the users in the evaluation phase.

1. The system was completely bug tested and all bugs found were solved.
2. User generated:
1. The changes to the control panel were implemented, to separate the miscellaneous inputs from the query and date sections.

2. A new check-box was added to the query selection which added a total entry to the graph.

3. A library entry was added to the query section.

3.9 Design and Implementation Conclusion

As stated in the introduction the aim of this section of the project was to produce a system that complete all of the necessary aims. This is demonstrated in the diagrams below. Therefore from a purely results driven perspective this was a successful section.

However the another outcome from this section was the success of the design and implementation paradigm, the intertwining design, implementation and user feedback meetings into each stage worked very well as a development method. Many changes to functions and design features resulted from this integration of user and developer. An example of this was in integration of a library query option that was highlighted by a user outside of the regular user meetings. The importance of this functionality for researchers – it allows primary source specification, British or Dublin library records – wasn’t conceived be technical orientated developer. This core functionality would have been lost without multiple user interactions

The short development stages also allowed inefficient features to be discarded in their infancy. Features such as the Perl based SOAP web service were discarded during the prototyping phase as it became obvious that the time delay was not worth the added level of data independence - the SOAP web service would have acted as an intermediate between the data layer and the database. As this
change was made in an early stage, the rest of the system was designed accordingly in the next design section, without another layer of the system depending on the rejected part.

3.9.1 The Completed System

*Illustration 30: The application input outlining the main features*
Illustration 31: Image indicating the parts of the system including ticker tapping

Copy of previous graph

Current graph and query options

Option to delete entire tape, all entries

Info section of “tick”

Entry Title

Notes section

Unique graph URL

Delete tick option
4 Evaluation

4.1 Chapter overview

As this is a user-centric project it is very important that the required targets are achieved and properly evaluated. Requirements that were observer as a result of the research question and the state of art. In addition to those set by the users who will end up using the system.

This chapter is split into two main sections:
1. Self evaluation which considers how the research question and state of the arte requirements were fulfilled.
2. User evaluation which explains the methods used and results gained from the user tests.

4.2 Self evaluation:

4.2.1 Introduction

It's important to evaluate the end system against the goals set throughout the project which will form the basis of the evaluation. This will compare the initial goals to the final product to ensure there has been no object migration during development. Continuous self evaluation was performed throughout the implementation section in the form of user meetings. However it is critical that the end product satisfies all of the requirements initially set out for the project, and they are examined independently in this section.

4.2.2 State of art requirements

• **SOA Req 1 - Architecture**
This was implemented in the form to two PHP scripts and an
application layer consisting of HTML, JavaScript and CSS files. The independence of each layer was implemented and aided development as different layers were changed with no effects on the other layers. The ability to directly generate graphing and data layer pages – using a unique URL – demonstrates this. It is possible to investigate all of the application layer customisations directly through URL get requests, as was demonstrated in the first user meeting. In the same way the pure XML from the data layer can be seen and manipulated through URL without input from other layers. Again this shows that the layers are entirely independent of each other and that this requirement was fulfilled by the system.

Illustration 32: Example of direct access to graphing layer visualisation where http://localhost/sqlGraphBasisMultiWithMultiGraphsFieldSelect8.php is the URL of the graphing layer

The corresponding data-layer is shown below – the URLs get requests and the data shown are the same.
SOA Req 2 – RIA implementation

The web application changes what is shown by distinct divisions of the page rather than reloading the page. The alterations are self contained within sections of the application page, such as changing the source address of an image. Similar to the loading method used in Google Maps, as referenced in the SoA, each section is loaded individually rather than refresh the entire page following RIA principals. Changes to the layout of the page – like the ticker tapping or

Illustration 33: Example of direct URL access of the data layer
displaying the mined data - are performed by dynamic JavaScript functions without reloading the page.

• **SOA Req 3 Feedback from application**

Feedback is shown in multiple ways:

1. The inputs on the control panel indicates when actions are performed and their state. For example check boxes are ticked, start and end dates are displayed and query options are highlighted when selected.

![Illustration 34: Three examples of selection user feedback.](image)

2. With default settings, auto on, the main functions change dynamically with user input. The visualisation and mining auto-updates when options are selected.

• **SOA Req 4 HCI application UI**

Most of the key concepts of HCI were implemented in the design of this project. Most notably the Dialogue independence between the layers and the languages used in this project. In addition to iterative tools constant language, simple layout, error prevention and rapid prototyping.

**SoA visualisation Requirements**

The graphing utility is evaluated in comparison with these requirements from Schneidermans seven features of good data visualisation, as examined in the state of the art.:

1. Overview: Gain an overview of the entire collection. The system can graph all of the data for a particular query set. This provides overview functionally as all possible data from this query is encoded in the visualisation.
Illustration 35: Example of overview visualisation of all data from a query

2. Zoom : Zoom in on items of interest
This feature is facilitated by the changing of dates, from a wide range to a small range, to zoom into a region of interest.

Illustration 36: Example of zooming in from from wide (left) to small range of dates (right)
This can also be applied to amount of data on the graph, for example number of queries:
3. Filter: filter out uninteresting items.

This is performed in the same manner as the data zooming displayed in the previous point.

The querying functionality also allows undesired output to be excluded.

4. Details-on-demand: Select an item or group

The mining functionality supports this, as users can select a section and view the raw data that forms this visualisation. As shown below, the same number of interactions displayed on the graph in year 1620, are shown in detail in the data mining table, as can be seen from the date column.
5. Relate: View relationships among items.

The relationships between the results can be seen by the highlighted query options in the control box.

6. History: Keep a history

The ticker-taping functionality keeps a history of the graphs produced, the time they were created and provides the ability to make notes about them.

7. Extract: Allow extraction of sub-collections and of the details when needed.

The zooming and data mining provides extraction in the graph while the options to open them in different tabs or download the data which supports further extraction.
HCI Requirements

AS mentioned in the SoA Sniderman provided golden rules for a good HCI. These were met by the system in the following manner:

1. **Strive for consistency**
   1. identical terminology should be used in prompts, menus, etc
   2. colour, layout, capitalization, fonts, and so on should be constant throughout the application.

This was performed by using the same CSS style sheet for the entire application layer.
Titles, headers and input styling were applied consistently.

2. **Enable frequent users to use short cuts**
   1. to increase the pace of interaction

The application of the unique graph URLs allowed frequent users to reproduce the same graph quickly, in addition to text inputs.
The use of a slider and distinct input boxes for the state and end dates allow speedily setting of specific dates.
The auto function allows expert users to customise selections without changes to the current graph – speeding up the process and eliminating distractions.

3. **Offer informative feedback**
   1. for every user action, the system should respond in some way; button changing colour etc

The use of JavaScript and JQuery buttons automatically provides this functionality.

4. **Design dialogues to yield closure**
   1. Actions should have a beginning, middle, and end.
Although there is only one dialogue box this yields to closure in the normal way. The extra tabs use predefined browser functionality as does the main application page.

The Ticker taping function also employs closure options (both for single entire and the entire “tape.”)

The mining section can also collapse.

5. **Offer error prevention and simple error handling**

1. **Design the form so that users cannot make a serious error; for example, prefer menu selection to form fill-in and do not allow alphabetic characters in numeric entry fields**

There are only two text field entries for querying, and in order to minimise errors there are two alternate ways to enter the same data (date-picker and slider). Error prevention was performed by regression testing the final system.

These text boxes also strip out any non-numeric inputs in order to reduce errors.

```javascript
function isNumberKey(evt) {
    var charCode = (evt.which) ? evt.which : event.keyCode
    if (charCode != 46 && charCode > 31 && (charCode < 48 || charCode > 57))
        return false;
    return true;
}
```

*Text 10: JavaScript function to check if input is a number*
2. If users make an error, instructions should be written to detect the error and offer simple and specific instructions for recovery.

All possible errors result in an alert, preventing the error committing and informing the user, as shown below.

Illustration 41: Example of error alert

6. Permit easy reversal of actions

1. Cancel buttons and back buttons

There are no back buttons – however the ticker-taping function store all the resultant graphs, prevention loss of data.

7. Support internal locus of control

1. Experienced users want to be in charge. Surprising system actions, tedious sequences of data entries, inability or difficulty in obtaining necessary information, and inability to produce the action desired all build anxiety and dissatisfaction with the system.

The system is designed so that all changes to the system are a result of specific inputs from the user.
As all inputs are button based there is no extensive data input cycles. The layout and access of data is designed to provide a intuitive actions in order to return desired results.

8. Reduce short-term memory load

1. Studies suggests that humans can store only 7 (+/- 2) pieces of information in their short term memory. It is possible to reduce this by designing screens where options are clearly visible, and don't need in depth knowledge to operate.

The ticker-taping allows a history and note-taking functionality in order to improve the system aided memory.

All options are clearly visible in the simple layout, resulting in a shallow learning curve for novice or non-technical user.

4.2.3 Research Question Requirements

The aim of this project was to answer the following research question:

“to what extent can an appropriate web-based interface for non-technical domain experts to support visual querying/ exploration of historic records be developed?”

This was answered well by the final result of the project that complied with the following requirement from the research question, as set out at the start of the design and implementation chapter:

• RQ Req 1: The interface should be displayed as a web based application.

Hing a HTML and JavaScript based front end; system can be accessed through all currently supported browsers. Therefore this can be said to be HTML based web application and this goal has been completed.
• **RQ Req 2**: should connect to a the resource of historic records, in this case the statues staples database. This has been successful, the resource is used in its original schema from a MySQL database.

```
$sql = "SELECT COUNT(*)
$tableQuery
WHERE (
    $whereQuery
    AND msdent_old.date
    BETWEEN '$low-01-01'
    AND '$high-01-01'
);
$result = mysql_query($sql);
$row = mysql_fetch_row($result);
if (!$result) {
    echo 'Could not run query: ' . mysql_error();
    exit;
}
```

*Text 11:* PHP code from the data layer to access the Statutes Staples database with custom queries

• **RQ Req 3**: should have visuals displaying the information in a graphical formatting
This can be seen from the following of Snidermans 7 graphing features in the above section.

• **RQ Req 4**: should support querying the database to create

Illustration 42: System open in Mozilla Firefox (Left) and Google Chrome (right) internet browsers
different visualisations.

There are four different query methods supported by this application.

1. Date
   - Start date
   - End date
2. Library selection
3. Debtor or creditor selection
4. Query selection (Occupational, location or amount)
   - RQ Req 5: must have ability to explore data-mining, and return the pure tabular information from the database.

Data-mining is supported by user iteration with the graphing image, and the tabular data can be returned to the page or downloaded in CSV format.

![Illustration 43: Table of raw data displayed on the application](image)

### 4.3 External evaluation

As a user-driven system, the feedback from those that will be using the system on a day to day basis is a cornerstone of the evaluation.

The reaction of external users, who are world experts in the contents of the database, to the system is a core guideline of the success of the HCI and the usability of the application.
As the developer is a technically orientated with little knowledge of the domain specific content of the resource, the users will judge how accurately the core elements of the resource have been accessed and provide a fresh and impartial evaluation of the RIA produced.

4.3.1 Purpose of Users tests

To provide feedback on the success of the visualization of the Statue Staples financial database and the application that accesses this data.

The aim of this study was to determine how successfully the aims of this project have been implemented. With emphasis on the usability of the UI and how the visualization is implemented with regard to non-technical domain expert users.

4.3.2 Methods and Measurements of the User Test

This study will provide the participants a short set of tasks to be completed on the provided web application. The tasks are designed to explore the full extent of the web application. There are distinct end conditions for all the instructions. The user will be asked to complete these tasks and then provide feedback about them.

User tasks:

1. Complete a set of instructions using the visualisation web application.
2. Fill out a questionnaire. This is to ascertain the reactions to using this tool.
3. Participate in an informal recorded question and answer session in order to provide some more in depth feedback on the possible improvements and areas of interest.
4.3.3 Question and Answer Session

There was a short, less than 20 minutes, debriefing session directly after the use of the application.

The aim of this debriefing is to gain any additional information about the participants experience with the application that may not be covered in the questionnaire, with an emphasis on possible improvements and areas of strength. This debriefing will be audio recorded.

The tests:

<table>
<thead>
<tr>
<th>No</th>
<th>Tasks</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify and save a line graph with 20 entries.</td>
<td>Change the type of graphs and the number if entries. Test HCI input intuitivity</td>
</tr>
<tr>
<td>2</td>
<td>Produce a graph of Gentry Creditors from 1635-45.</td>
<td>Test targeted time period and specific querying</td>
</tr>
<tr>
<td>2.1</td>
<td>Save this graph</td>
<td>Ticker taping</td>
</tr>
<tr>
<td>2.2</td>
<td>Make a note to describe the saved graph.</td>
<td>Ticker taping functions</td>
</tr>
<tr>
<td>3</td>
<td>Find the name of a single gent who was a debtor in this period (the period above 1635-45).</td>
<td>Data mining</td>
</tr>
<tr>
<td>4</td>
<td>Compare the debts by merchants in Dublin and the number of debts by merchants in Cork. Turn on Auto to see the evolution of these graphs</td>
<td>Comparing graphs – by opening one in a new tab or by using the ticker tapping. Further querying testing Use of auto.</td>
</tr>
<tr>
<td>4.1</td>
<td>Find the two graphs that represent this data best. Open theses two graphs in New tabs.</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Tasks</td>
<td>Motivation</td>
</tr>
<tr>
<td>----</td>
<td>-------</td>
<td>------------</td>
</tr>
<tr>
<td>5</td>
<td>Compare the number of transactions of 100 pounds or less for trader and soldier.</td>
<td>Different querying and graph customisation</td>
</tr>
<tr>
<td>6</td>
<td>Find the period where the amounts of debtors and creditors in Dublin were highest and lowest.</td>
<td>Changing the amount of inputs shown to solve a real world task. Using changing start and end dates to take wide view to the specifics.</td>
</tr>
<tr>
<td>7</td>
<td>Save these graphs and the date range of these two periods.</td>
<td>Ticker taping in conjunction with realistic problem</td>
</tr>
<tr>
<td>8</td>
<td>How does this compare with the total amount of debtors and creditors at that time? Note these in the description of these graphs. Ensure that only these graphs are saved on the application.</td>
<td>Comparison</td>
</tr>
</tbody>
</table>

### The feedback:

<table>
<thead>
<tr>
<th>Question</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>What is you overall feeling of the experience using this visualisation tool?</td>
</tr>
<tr>
<td>2</td>
<td>What, for you is the weakest part of this tool?</td>
</tr>
<tr>
<td>2.1</td>
<td>Could this be improved or would you prefer it to be stripped out altogether?</td>
</tr>
<tr>
<td>3</td>
<td>If you could add additional functionality/change one part of this tool, what would it be?</td>
</tr>
<tr>
<td><strong>Question</strong></td>
<td><strong>Motivation</strong></td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4. What is the strongest part of this project (what performs best)?</td>
<td>What went well</td>
</tr>
<tr>
<td>5. What part would you use most in your research?</td>
<td>What was most important and would be used most</td>
</tr>
<tr>
<td>6. Would you like this type of visualisation to be applied to other resources? Why?</td>
<td>Further work</td>
</tr>
</tbody>
</table>
| 7. Which of:  
  1. Zooming in and out over time (range: decades to days)  
  2. More graph options  
  3. Faster data retrieval  
  4. User defined queries (ie select 50-200 pounds)  
  Would you like to be added most? | Further work                                                                    |

### 4.3.4 SUS

System Usability Survey was used to find a repeatable and stable way to gauge the usability of the application.

A scale was used as:

Usability does not exist in any absolute sense; it can only be defined with reference to particular contexts. [Brook, SUS]

So a System Usability Scale (SUS) was used as it is a reliable, low-cost usability scale that can be used for global assessments of systems usability[Brook, SUS]

Six options were provided so that there were no neutral responses. This improves the feedback from the survey, as every response contains feedback.
4.3.5 Participants

Participants will be the members of the Trinity College humanities departments. There were four participants in the study, although one didn't complete the survey. Each member will be familiar research into historical data.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Area of expertise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participant 1</td>
<td>Digital Humanities</td>
</tr>
<tr>
<td>Participant 2</td>
<td>Statues Staples database</td>
</tr>
<tr>
<td>Participant 3</td>
<td>Historical research.</td>
</tr>
<tr>
<td>Participant 4</td>
<td>1641 depositions</td>
</tr>
</tbody>
</table>

4.3.6 Results:

Table of SUS results

<table>
<thead>
<tr>
<th>Question</th>
<th>5 4 3 2 1 0 Total Ave</th>
</tr>
</thead>
<tbody>
<tr>
<td>This application allows you to explore the Statues Staples data in a new way.</td>
<td>3 0 0 0 0 0 15 1</td>
</tr>
<tr>
<td>I found this tool easy to use.</td>
<td>1 1 0 1 0 0 11 0.7333</td>
</tr>
<tr>
<td>In my opinion this tool would be useful in creating a publication.</td>
<td>2 1 0 0 0 0 14 0.9333</td>
</tr>
<tr>
<td>I found some data hard to access (hidden from the user).</td>
<td>0 0 1 0 2 0 5 0.3333</td>
</tr>
<tr>
<td>I found no restrictions when trying to identify a individual source.</td>
<td>3 0 0 0 0 0 15 1</td>
</tr>
<tr>
<td>I was happy with the ability to zoom in and out in the data in the graphs</td>
<td>2 0 0 0 0 0 10 0.6667</td>
</tr>
<tr>
<td>I was happy with the speed of the system (time to access data – time to create graphs)</td>
<td>2 1 0 0 0 0 14 0.9333</td>
</tr>
<tr>
<td>I found the various functions in this system were well integrated</td>
<td>2 1 0 0 0 0 14 0.9333</td>
</tr>
<tr>
<td>I found the saved graph/history section useful.</td>
<td>3 0 0 0 0 0 15 1</td>
</tr>
<tr>
<td>I found the ability to reproduce (cite) the graphs at a later date from a unique URL</td>
<td>3 0 0 0 0 0 15 1</td>
</tr>
<tr>
<td>I would be happy to see this tool applied to other sources.</td>
<td>3 0 0 0 0 0 15 1</td>
</tr>
<tr>
<td>I was happy with the range of graph types (bar, line and scatter) available.</td>
<td>2 0 0 0 0 1 10 0.6667</td>
</tr>
<tr>
<td>I was happy with the number of queries (amount, occupation or location) available.</td>
<td>2 0 1 0 0 0 13 0.8667</td>
</tr>
</tbody>
</table>
4.3.7 Discussion of these results in general terms

SUS

From the results of the SUS questionnaire it can be concluded that the users found the system very intuitive and easy to use.

Five sections received unanimous 100% positive feedback, and all sections received average positive feedback, >50% approval.

The five sections that performed perfectly were:

- The ability to explore the Statues Staples data in a new way.
- Being able to identify a individual source i.e. the querying mechanics. (Name, location, amount, occupation, of debtor and creditor)
- The use of the ticker tapping.
- Citable unique graph URLs

And the proposition to apply this tool to other sources also received universal, complete backing.

This last factor is very significant as the motivation of one of the
main design goals (the three tiered architecture,) was to allow new data resources to be applied with the minimum effort.

From the results Questions 1,5,9,10 and 11 all scored 100% approval from the testers. This shows that the core functions performed exceptionally and that the expert users gave them their full support. This covers the main motivation function of the system – the access the resource is a new way.

These questions covered the main roles of the system – support of citation- (ability to reproduce a URL specific graph and the ticker taping), data-mining, ability to select a single entry.

Question 4,5, 8 and 9 all received very strong agreement and one distraction of the 2nd highest positive result. This is also excellent feedback, as it is as near perfect evaluation as possible.

The biggest concern from the SUS feedback is the basic usability. As can be seen from the results the functions and performance of the application were very well received. There were some outlining results due to personal user preferance.

4.3.8 Evaluation Conclusion

This system has fulfilled the goals set out by the research question , the state of the art and by the users. There were few areas of dissatisfaction and some suggested sections to be built on in the future. The user meeting showed the difference in approaches taken by different users and highlighted the need for continual user interaction. The desire for a library section option displays the additional features that some expert users require while the desire for pie graphs by one of the users demonstrates the difference in approaches as this feature was ruled out in an earlier user meeting.
5 Conclusion

Our ability to collect and store data is growing faster than our ability to analyse it. [Keim]

This project has successfully provided a system that enabled users from the History Department at TCD to visualise and mine data from historical records intuitively, and to it without needing advanced computer skills.

The motivation behind this project was to provide a system that would eliminate the technical experience gap between the end users and those which skills to access the resource. This would and provide clean data from an under used resource to casual and expert users.

5.0.1 Objectives Achieved

In order to complete this project and answer the research question several objectives were completed:

- Objective 1: Research
  The state of the art of techniques and technologies of visualisation was examined. Research was conducted on the evolution and continuing improvement of technologies for web-based applications and visualisation techniques using open source software, such as the rise of RIA in the last three years. This led to the use of the JPGraph technology which provided the most information rich graphs, encoding data in independent quantities. The graphs were delivered in multiple formats using position and colour to encode data.

- Objective 2: Design and Implementation
  The system case completed using a combined design and implementation cycle. There were six stages of development, in
each there was a development section, a user feedback meeting and then a design period to outline goals for the next iteration. This provided a very short feedback loop with the users, thus allowing the system to be specifically tailored to their needs.

The main tasks were wire framed and presented to the users – with the main graphing function. Then most of the functionality was rapidly prototyped and changes were made where necessary.

The capabilities of the functions were then extended and the user interface was redesigned. Additional features, such as the ticker taping system, outlined in the user meetings were implemented. The system was then finalised and perfected to meet the users wishes, following this regression testing was performed.

• Objective 3: Evaluation.

This objective was carried out by comparing the system produced to the requirement set out by the research question and the state of the art.

User tests were performed on four domain expert users who evaluated the systems ability to perform a selection of tasks. The user tests provided universally positive feedback, as all users completed the tasks with minimal problems. The results from system usability scale survey were equally positive with universal approval for a third of the queried sections including applying the system to other resources, ability to support citations and data mining.
5.0.2 Answering the Research Question

The aim of the project was to answer the question “to what extent can an appropriate web-based interface for non-technical domain experts to support visual querying/exploration of historic records be developed?”

The project answered the research question to the fullest extent.

The system is web based, it offers visual querying and exploration of historic records to non-technical domain experts. The results for the user surveys show this to have been the case, with overwhelming positive feedback. There visual querying was supported by four different query fields over and the exploration allowed the custom raw data to be accessed or downloaded from any of the graphs.

This solved the current problem of data overloading by taking a under-used resource and making it accessible by closing the technical experience gap between the domain experts and the software skills needed to access the resource.

Digitalising records is still a high priority however as this project has shown further steps can be taken to make the data more accessible to a wider audience. By using a three tired architecture the resultant system can be easily applied to other under-used data sets, and all of the users strongly agreed that this would be a desirable outcome. This report has proven that this functionality and expertise access is possible for a single system – it is envisioned that this application can be applied to other data sources, with no changes to the human computer interface.

This shows clear steps to ward off obsolescence of applications. By designing a system that can be partially upgraded as technologies
improves, for example using HTML 5 to create a new application layer or porting the application layer for use with mobile devices, this system acknowledges the continual evolution of web based technologies.

Software developers are facing a new challenge. Computational power and software efficiency are no longer the central tasks for developers to solve. Getting input and displaying results in a meaningful manner to an ever increasing non-technical web based audience on the is the challenge of the 22nd century. This thesis has outlined significant steps to further research in that area.

5.1 Future work.

In order to release this system to the public as a professional product there are several tasks that need to be completed.

5.1.1 Current position

At the moment the system is a final prototype. It has the functionality that is required of the released product however it needs improvement in scalability and security in order to be formally released.

6.1.2 Areas for Improvement

• Multiple Users

In the future this system should be able to support multiple users. In order to support more than one user at the same time it will be necessary to implement concurrency. This will provide a separate thread for each user to have. This is a result of the difference between having a standard internet page and an application. As 'normal' web-pages have no customisation or results from unique user input.
• Additional features

There are several additional features that were not covered by the research question but would be helpful to implement in the released version of the application. These were highlighted from development:
  ◦ Zooming in and out buttons.

These could be added in order to change the frequency of the selection of data. From years, as is the default setting currently, to six months to months, and out to two or five years. This will allow more accurate querying and complete zooming over different periods with the same amount of entries. For example a five year period with 10 entries at 6 month intervals and a fifty year selection with ten five year entries.

• Connecting to other databases.

This would provide the same data exploration and visualisation tools on other unused resources. This would increase the power of the application and take advantage of the designs decisions made to make this possible (dedicated data layer.) Having two different resources, so the system can switch between the two would create a very powerful front end and vastly increase the user base for a relatively small amount of time expended.

• Integration into other resources

In order to deliver this application successfully it will be necessary to implement this with other existing resources that are related to similar areas.

There are two areas that this project will be tied with; the trinity 1641 depositions and the CULTURA digital resources section.
  ◦ Trinity depositions is an online resource that covers the extensive primary resources related to the 1641 Irish depositions
owned by the college. As the statues staples database, and constitute this project covers this period of Irish history this is an obvious tie in for this project.

- CULTURA is a cross college digital humanities project. With work going on around Europe there is a certain amount of cross over, as part of the projects involve visualising other types of Irish data. For example there is a CULTURA project that involves mapping the territories controlled by landlords during the 1641 depositions. This is similar in nature to this project and so this project can be supplied to provide additional information on this period.

In it intended that during the summer of 2012 this project will be further developed and delivered as a finished resource. This intended work will address the main points described for improvement above with particular emphasis on multiple user stability. This will be integrated with the current CULTURA web applications and the trinity 1641 depositions on the Trinity College history department website.

5.2 Summary

The aim of this project was to produce a user centric system to visualise financial data for non domain specialist users. This was achieved by building a rich internet application, following correct human computer interface guidelines. The system fulfilled both the requirements set from the research question and those set by the expert users who will use the application professionally. The success of this project is demonstrated by the decision to continue working on the system in order to deploy it for public use.
Bibliography

Clottes, J: Jean Clottes, Chauvet Cave: The Art of Earliest Times., 2003
Shneiderman, B.: Shneiderman, B., The eyes have it: a task by data type taxonomy for information visualizations, 3-6 Sep 1996
Cartography Hist: , history of cartography, 2007,
Mulhall, M: Mulhall, Michael G, , 1884
Christopher Ahlberg, B. Schneiderman: Christopher Ahlberg, B. Schneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, 1994
Tufte, E: Edward Tufte, Envisioning Information, 1990
Curdt et al: Curdt, Hoffmeister, Waldhoff, Bareth, Management and visualisation of interdisciplinary Research data with arcgis server data,
HARTSON and HIX: H. REX HARTSON and DEBORAH HIX, Human-computer interface development: concepts and systems for its management,
Keim: Dabiel A Keim et al., challenges in Visual Data Analysis,
Stolte C,: Chrs Stolte and Pat Hannrhan, Polaris: A Systrem for Query, Analysis and Visualization f Multi-dimensional relational Databases,
Gapminder: Google Inc, Gapminder Word, 2008,
Brook, SUS: John Brook, SUS - A quick and dirty usability scale, 1991
6 Appendix

6.1 User Tasks

User Tasks

*note:* completing this questionnaire is entirely voluntary – you may leave at any time and there is no obligation to complete this questionnaire.

1. Identify and save a **line** graph with **20 entries**.
2. Produce a graph of **Gentry Creditors** from **1635-45**.
   1. Save this graph
   2. Make a note to describe the saved graph.
3. Find the name of a single **gent** who was a **debtor** in **this period** (the period above- **1635-45**).
4. Compare the debts by **merchants** in **Dublin** and the number of debts by **merchants** in **Cork**.
   1. Turn on **Auto** to see the evolution of these graphs
   2. Find the two graphs that represent this data best. Open theses two graphs in new tabs.
5. Compare the number of transactions of **100 pounds or less** for **trader** and **soldier**.
6. Find the period where the amounts of **debtor**s and **creditors** in **Dublin** were highest and lowest. Save these graphs and the date range of these two periods.
   1. How does this compare with the **total** amount of debtors and creditors at that time? **Note** these in the description of these graphs.
   2. Ensure that **only these graphs** are saved on the application.
6.2 User SUS questionnaire

Please note - completing this questionnaire is entirely voluntary - you may leave at any time and there is no obligation to complete this questionnaire.

**Questionnaire for user of the Visualisation of Statues Stables Database:**

This application allows you to explore the Statues Stables data in a new way.

I found this tool easy to use.

In my opinion this tool would be useful in creating a publication.

I found some data hard to access (hidden from the user).

I found no restrictions when trying to identify a individual source (Name, location, amount, occupation, of debtor and creditor)

I was happy with the ability to zoom in and out in the data in the graphs (manipulate the range of dates the graph spans)

I was happy with the speed of the system (time to access data – time to create graphs)

I found the various functions in this system were well integrated

I found the saved graph history section useful.

I found the ability to reproduce (cite) the graphs at a later date from a unique URL useful.

I would be happy to see this tool applied to other sources.

I was happy with the range of graph types (bar, line and scatter) available.

I was happy with the number of queries (amount, occupation or location) available
6.3 User debriefing questions

Debriefing questions for study participants

Please indicate that you are happy that this conversation will be audio recorded.
Signed Participant:

1. What is your overall feeling of the experience using this visualisation tool?
2. What, for you is the weakest part of this tool?
   1. Could this be improved or would you prefer it to be stripped out altogether?
3. If you could add additional functionality/change one part of this tool, what would it be?
4. What is the strongest part of this project (what performs best)?
5. What part would you use most in your research?
6. Would you like this type of visualisation to be applied to other resources? Why?
7. Which of:
   1. Zooming in and out over time (range: decades to days)
   2. More graph options
   3. Faster data retrieval
   4. User defined queries (ie select 50-200 pounds)
   Would you like to be added most?
6.4 User Instructions

**Instructions for using the Visualisation of Statues Staples Database:**

**Interaction - The control box:**

**Debtors and creditors** changes the tables whether to debtors, creditors or (exclusively) both would be considered in the selection of data.

**Amount|Occ|Location** sets the way the data will be categorised, by the size of the translation (Amount,) the occupation or the location of the debtors/creditors.

**Date|Field** selects how the graph is organised - by the year or the selected category on the x-axis.

**Save as CSV** saves the selected mined data as a Comma Separated file (opened by excel etc) – note the values are separated by a semi-colon. A popup appears so the name of the file can be opened.

**Mine tab** opens up the data mining table in a new tab.

**Graph tab** opens up the graph (only) in a new tab.

**Auto** toggles between automatically updating the graph to changes made, when off, the graph only changes when the Save/Change graph is selected. This feature is designed to allow the user a preview of the graph as that variables are adjusted.

**Changing the date** – there are three ways to change the date:

- enter the date into the date box- either using the date-picker or entering the year manually/
- Using the slider
- Using the add year/less year buttons – this allows fine tuning of the dates – both the upper and lower limits change when these buttons are pressed.

**Save/Change graph** – this changes the graph and propagates a copy into the following table.

**Data mining** – the section of the graph can be selected and all the data from this section(year). Is shown below the graph. This allows further investigation of the data – displaying real people behind the displays.