Identifying an emerging pattern between investment in Youth Services and Knife Crime in London

Lee Wallace | 000938568

Supervisor: Maria Pretorius

Word Count: 13,196

May 2020

A report submitted for the degree of

BSc (Hons) Business Information Technology

Acknowledgements

The basis for this research stemmed from my personal experiences living in London and passion for making the streets that I use, safer. Statistics show that Knife Crime in London has been on a steady rise for many years. It is not only my passion to gain an insight into this problem, but also to develop a tool that can be used for identifying areas that need immediate support.

I would like to express my gratitude to my supervisor Maria Pretorius and second marker Aditi Rawal, for their guidance and direction. The meetings and conversations were vital in inspiring me to think objectively throughout this project.

Finally, I would like to thank my family who supported me with their time, knowledge and understanding throughout the duration of my research.

Thank you all for your unwavering support.

Abstract

This report focuses on reviewing current research on the correlation between increased knife crime and reduced youth service funding. Increased youth service funding highlights benefits for a local community whilst reduced funding reveals correlations with violent knife crime. To ensure that these findings are presented in a digestible format, research on human retention of information from interactive visualizations using raw data also offers positive results. This report concludes by presenting a high-level prototype and its development process, in which authorities can use to identify London boroughs that need immediate financial intervention, in order to curve increasing rates of knife crime.

Contents

Glossary **vii**

1 Overview **1**

1.1 Aim 1

1.2 Report Structure 1

2 Project Framework **3**

2.1 Neilson's Heuristics 3

2.2 Planning 5

3 Literature Review **6**

3.1 Approach to Literature Searching 6

3.2 Knife Crime 6

3.3 Youth Services 7

3.4 Correlation between rising Knife Crime and reduced Youth Service Funding . . . 9

3.5 Visualising Data 10

3.6 Conclusion 13

4 Product Research **15**

4.1 Scoring Existing Products 16

4.2 Conclusion 25

5 Legal, Social, Ethical and Professional Considerations **26**

6 Requirements Analysis **27**

6.1 Functional Requirements 27

6.2 Non-functional Requirements 30

6.3 MoSCoW Analysis 31

6.4 Timeboxing 32

7 Design and Implementation **33**

7.1 Use Case Diagram 33
7.2 Entity Relationship Diagram 34
7.3 Low-level Prototypes 36

8 Development **41**

8.1 Review of Technology 41

9 Prototyping **44**

9.1 Timebox One 44
9.2 Timebox Two 51
9.3 Timebox Three 58

10 Testing **60**

10.1 White-Box Testing 60
10.2 Usability Testing 60

11 Critical Evaluation **61**

11.1 Product Evaluation 61
11.2 Personal Evaluation 62

12 Conclusion **64**

13 References **65**

Appendices

A Project Proposal **72**

B Gantt Chart **79**

C Uses of visualisations **80**

D Code Explanations **81**

E Prototype One **92**

F Prototype Two **96**

G White-Box Testing **104**

H Usability Testing **118**

I Email Examples **121**

J Further Information **123**

K Additional Low-level Prototypes **124**

Glossary

WYSIWYG . . . . . A What You See Is What You Get editor or program allows a developer to see what the end result will look like while the interface or document is being created (Rouse, 2020).

RDBMS . . . . . A Relational Database Management System is a program that allows you to modify a relational database.

LSEPI . . . . . Legal, Social, Ethical and Professional Issues.

POST Request . . . . Appends submitted form data from a user in the body of the HTTP request (W3schools, 2020). A server typically handles the response.

Accessing the Prototype

The full code used to implement the high-fidelity prototype can be found in the zip file uploaded with this report.

A live version of the prototype can be accessed using the following link:

https://fyp-000938568.herokuapp.com/

Application Login Credentials

User Type	Username	Password
Normal User	user	123123
Administrator	admin	123123

1

Overview

Statistics reveal that violence involving a bladed weapon is rising in London, especially involving people aged under 25 (GLA Intelligence, 2018). A study by the Greater London Authority (GLA) Intelligence Unit found that knife crime reported to the Metropolitan Police (MET) had increased by 67% from 2013 to 2018 (GLA Intelligence, 2018). Across the same period, there was a real-terms funding cut of 40% in youth services (Barnardos, 2019).

Furthermore, research shows that a preventative approach to knife crime is essential (Byc.org.uk, 2019). Youth centres provide not only a preventative approach but also a safe environment for young people to build relationships within the community and participate in recreational activities. A significant reduction in funding for youth services since 2013 has resulted in over half of these centres being closed in London alone.

Across the UK, initial research has shown that the common council has reduced funding for youth centres by 40%, with some authorities reporting a 91% decline in funding. Recent research by the All-Party Parliamentary Group (APPG) has found that areas suffering the most significant cuts to youth spending have seen more substantial increases in youth crime (Dearden, 2019).

This report aims to identify this emerging pattern and provide a tool for authorities for users to decide where to prioritise youth service funding to tackle knife crime. Using a visualisation tool to identify crime hotspots will allow the MET to pinpoint areas that need priority in resources. A visualisation tool can be used by MPs to identify areas that need additional resources that will tackle youth crime. This project will also design, develop and implement a comprehensive visualisation dashboard for the use by the MET which present visualisations of the data and its findings. This tool will identify key areas in London have seen significant youth service cuts and provide funding suggestions which will aid in the reduction of knife crime.

Keywords: [youth crime, London crime, knife crime, visuali?ation, “police use of visualisation”]

1.1 Aim

The aim of this project is to create a comprehensive visualisation dashboard for the MET to identify areas that would benefit from youth service funding to reduce knife crime.

1.2 Report Structure

The rest of this report is structured as follows: Chapter 2 introduces the development framework that this project follows and how its characteristics are beneficial. Following this, Chapter 3 reviews current literature to identify findings across different mediums of research. Chapter 4

scores similar products against a critereon. Chapter 5 discusses various Legal, Social, Ethical and Professional Issues that may apply to this project and these are considered.

In Chapter 6, functional and non-functional requirements for the high-fidelity prototype are introduced, discussed and prioritised for development. In Chapter 7, prerequisites for development are introduced, such as database schema's and low-fidelity prototypes. In addition to this, Chapter 8 reviews available technologies. The advantages and disadvantages of various database engines and web frameworks are discussed.

Chapter 9 documents the process of prototyping the final prototype, with various key features and their implementations discussed. Chapter 10 documents the Testing that has been carried out. Chapter 11 discusses personal and project evaluations, with further developments highlighted. To conclude this report, Chapter 12 summarises the findings over the duration of this project and the effectiveness of the high-fidelity prototype.

2

Project Framework

The project will use DSDM (Dynamic Systems Development Method). DSDM is an agile software development framework based on the Rapid Application method (SolutionsIQ, 2020). This methodology has been selected because Rapid Application Development is needed for this project due to limited time constraints. To manage the time constraints of this project, the high-level requirements will be prioritised using MoSCoW – this is so essential requirements will be implemented in the first iteration, allowing more time for the remaining requirements to be developed.

In addition to using prioritisation, this project will utilise timeboxing as a technique. Timeboxing will be used to allocate a fixed unit of time to be spent on every activity (Scruminc, 2020), following this technique will allow the project to be completed within the given time constraints. Utilising MoSCoW prioritisation and timeboxing will mean the beginning of the project will have time spent scrutinising which high-level requirements are the most important, and thus, what requirements should be implemented in the first iteration. Further requirements which are not categorised as 'MUST' in the MoSCoW method will be developed in a new iteration of the product.

This chosen project framework will be followed very closely throughout this project in order to rapidly-produce a deliverable that is high quality. Further techniques used will also be referenced, and the processes described thoroughly in the project documentation.

DSDM has also been selected because one of its eight principles encourages iterative development (agilekrc, 2020). As a result, the prototype can be prepared in timeboxes which will keep the project schedule on track. In addition to this, iterative development provides scope for development of multiple prototypes. Iterations can result in a more well-rounded prototype being developed.

Using a framework that does not support prototyping for this project would be counter-productive. Another framework such as the Waterfall Model for example, is referred to as a Linear-Sequential Life Cycle Model (Sharma, 2016). This type of model does not support reactive changes being made to the application, such as that from user feedback. However, user feedback is crucial for this project in making positive changes for future prototypes.

2.1 Neilson's Heuristics

Neilson's Heuristics are popular usability heuristics for user interface design (Liyanage, 2016). These can be used to identify issues within a user interface, which can be improved on using iterative development, which follows the chosen DSDM methodology of iterative development. They will be used in this report for evaluating the design of products in the product research chapter. It will also be used as a guideline to ensure that the product developed in this project follows these heuristics. It is essential that this product follows these heuristics to provide a professional standard throughout.

In addition to this, research has shown that multiple evaluators are beneficial to find more issues that might not have been identified by one individual, for this reason, this project will use another person to evaluate the user interface through testing. This will improve the effectiveness of this method (Neilson, 1994). Neilson's Heuristics are composed of ten usability principles; these can be seen in **Table 2.1**.

Table 2.1: Neilson's Usability Heuristics

Heuristic	Description
Visibility of a system status	The System should not include dialogue that is irrelevant to the user. E.g. a minimalist error message that asks the user to try again later.
Match between the system and real-world	The System should use words, phrases and concepts familiar to the user, including familiar designs. Following this Heuristic will mean the system will not use technical jargon or system-orientated words.
User control and freedom	The System should offer an 'escape' function, should they accidentally use a function by mistake. E.g. Cancelling an action whilst in progress.
Consistency and standards	To promote efficiency and a clear design standard, the system should have quickly recognisable platform conventions. E.g. Menu bar/sign in button in the same location on every page.
Error prevention	This Heuristic describes a 'proactive' method to errors rather than waiting for an error or problem to appear. E.g. When Google mail detected when you are trying to send an attachment when no document has been attached. This prevents unnecessary communication in the future and saves time.
Recognition rather than recall	The System should allocate space in its interface to make objects, actions and options visible to the user. This includes making website instructions visible. E.g. Amazon remembering your search history and recommending items to you.
Flexibility and efficiency of use	The System should be designed so that novice and experienced users can use the interface efficiently. Experienced users can use 'accelerators' such as filters to enhance their experience using the interface.

Continued on next page

Table 2.1 – *Continued from previous page*

Heuristic	Description
Aesthetic and minimalist design	The System should not include dialogue that is irrelevant to the user. E.g. a minimalist error message that asks the user to try again later.
Help users recognise, diagnose, and recover from errors	Error messages should be clear to the user without any unnecessary jargon, although it should be able to indicate the problem and suggest a solution.
Help and documentation	A system should be able to be used without any documentation although it is always helpful to provide. Help and Documentation provided should be easy to find, digestible and focused on the users' task.

2.2 Planning

For this project, DSDM has been selected for the methodology. Techniques that will be utilised throughout this project include Iterative Prototyping, MoSCoW Prioritisation and the use of UML diagrams. Approaching my project using these techniques will result in being able to effectively manage the time constraints throughout the project using requirement prioritisation and timeboxing. In addition to managing time, I will also be able to produce a high-quality product which has been designed using UML diagrams and iterative prototyping.

At the beginning of the project, datasets will be acquired using publicly available sources such as data.gov.uk. From these datasets, the project aims to deliver a comprehensive visualisation dashboard which will incorporate developing a web application. Research into what technologies will be most suitable for this purpose, including the requirements of the user will be carried out before developing prototypes to ensure that the product meets the needs of the user. In order to identify suitable technologies, secondary research of advantages and disadvantages of web frameworks, database engines and visualisation software will be discussed. This will allow this project to select technologies that will be able to implement all functional and non-functional requirements.

In addition to this, the application will be thoroughly tested using a variety of techniques; including White-Box Testing and Usability Testing. Using White-Box Testing will be critical in testing the internal code structure of the application, this is identify logical or performance errors within the code (codefirst, 2020). White-Box Testing provides an opportunity for a developer to test user inputs for unexpected outcomes or operations.

Neilson's Heuristics will be used as the criteron for scoring all user interfaces in this report. This includes reviewing similar products and the final high-fidelity prototype.

A Gantt Chart can be seen in **Appendix B** to detail how this project will be scheduled.

3

Literature Review

3.1 Approach to Literature Searching

This Literature Review aims to discover a pattern emerging from investment in youth services and knife crime and will be split into three sections. The first section of research aims to review the literature concerning knife crime and the involvement of young people. The second section will discuss the significance of youth services, and the third section will review visualisation and how it aids people to read data.

The conclusion of this Literature Review will influence the product being developed. This includes potentially altering requirements or the design; this is so that the most effective product can be developed for the right users. This chapter uses Articles, Journals, and relevant sources to support discussions. Scholarly search engines such as Google Scholar and the Greenwich Library were used to find these sources.

3.2 Knife Crime

Knife crime includes the use of a knife to stab or in any way, attack another person. It is also illegal to carry a knife, threaten with a knife, commit a crime with a weapon (such as a robbery) and commit a crime using a fake knife (nidirect, 2019).

In the previous 12 months ending March 2019, there were 43,516 knife crime offences recorded across England and Wales, where it is mostly proportioned in London (Shaw, 2019), whose authorities have documented that for every 100,000 people in the capital, there were 169 offences involving a knife. This is followed by the North West of England with 93 knife offences per 100,000 people, resulting in London having a considerably higher offence rate.

According to data regarding gang crime in London, there has been a substantial increase from 2014 to 2018 in Knife Crime with injury for people under the age of 25 (Bulman, 2018), this is further backed up by findings by (Greater London Authority, 2018) whose report states that the "peak age for knife crime for carrying knifes is getting younger", and is "currently between 13 and 17 years old". Bulman (2018) also states that knife crime in London has surged by 69% in the same period. Other sources also show an observable link between youths and knife crime; the MET whose figures released in early 2019 show that half of all knife crime offenders in London are teenagers or younger children (Bentham, 2019), with four out of ten suspected of knife crimes aged between 15 and 19.

This rise in youth crime involving knives is especially worrying in terms of safety in London's boroughs as another study by GLA (2019) found that these incidents have increased from 67% from 2013.

It can be identified that knife crime involving youths started to increase dramatically in London from 2014 onwards. A study by (Greater London Authority, 2018) explains that factors such as deprivation and poor mental health are top indicators into the rise of SYV since 2014 . This identification of possible causes presents an issue for Members of Parliament (MP) as more than a quarter of all youths living in London live in the most deprived boroughs of the capital (London City Hall, 2019), in which deprivation is cited as the leading cause. Continued lack of support for these deprived boroughs in London may result in a further rise of crime. Further research by the GLA shows that socioeconomic factors such as social class or poverty increase the likelihood of being in a situation where an SYV incident could occur (GLA, 2019).

Overall, it can be identified that teenagers between 15 to 19 amounts to half of all knife crime offences across London, with knife crime offences being the most common SYV offence.

3.3 Youth Services

A youth service refers to the engagement of young people in an organised activity that contributes to the local or national community; this is mostly facilitated by Youth Centres (Youth Service, 2019). Youth Centres are open to all people aged 15-21, with some clubs accepting people as old as 25 (HYPP Youth Clubs, 2019).

3.3.1 Youth Centres

Youth Centres are a place where young people can meet and participate in activities with each other; activities include football, table tennis and video games (Mills, 2014). Youth Centres offer both advantages for the young people attending and the local community, advantages for young people include facilitating the development of personal and interpersonal skills, gain confidence and reduce the risk of becoming involved in 'unsafe activities' (King, 2019). Furthermore, it is also found that while young people are being involved in group activities with each other in a safe environment, they have a "safe identity" and are unlikely to engage in activities that are 'unsafe'.

Concerning advantages for Youth Centres, (Robertson, 2019) suggests that in adolescence, peer relationships are critical in growing an individual socially, allowing them to try out new roles and helping with developing a personality at a young age. Alongside developing a young person socially, youth centres also allow for young people to make healthy choices with an atmosphere to encourage them to stay healthy; this includes growing a strong support network.

Benefits can also be seen in local communities; these include further development of children's social skills in the local area, potentially resulting in a safer environment with reduced anti-social behaviour. Additionally, by involving young people in the community, they can develop leadership skills and young people who are more thoughtful of other members of society.

Overall, it can be recognised that Youth Centres provide a safe environment for young people, which statistically reduces their involvement with a crime of any kind. Further discussion can be had into whether reduced financial support in youth services has impacted the availability of youth centres and as a result, a safe environment for children, teenagers and young adults.

3.3.2 Reduction in Funding

A study by the youth persons charity YMCA found that from 2013, Local Authorities have reduced expenditure from £1.2bn on youth services by more than £750m resulting in a current budget of £448m, which is 60% of the initial budget (Youth and Consequences, 2018). The study also discusses that such a significant proportion of reduced expenditure on youth services in just six years would have needed more scrutiny by members of parliament in terms of the long-term impact on young people and local communities.

In terms of youth service funding in London, a study titled 'London's Lost Youth Services' published by the London Assembly shows that youth service funding has halved throughout London's boroughs since 2012. As a result of this, 104 out of 234 youth centres have closed since 2012 (Berry, 2019), seen in **Figure 3.1**.

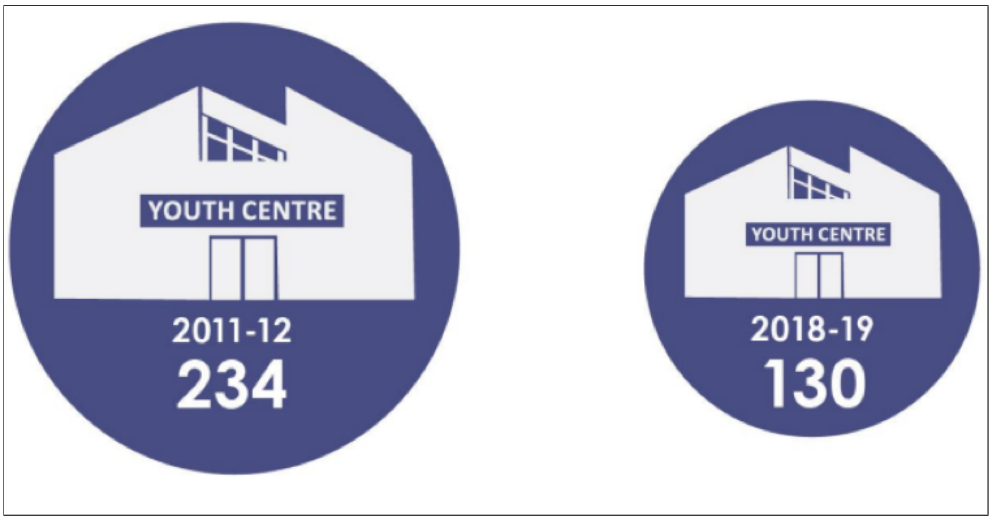

Figure 3.1: Decreasing number of Youth Centres (Berry, 2019)

Berry (2019) also discusses that most young people wanted investment in youth centres and services, in addition to outreach workers and mental health support. This revelation supports what was found by GLA (2019), where deprivation and poor mental health were top indicators amongst knife crime in youths – showing a clear correlation between two separate studies that young people use youth centres for outreach and mental health support, a leading indicator for rising knife crime in London.

External grants are granted to councils from external sources to contribute to their youth services budgets. **Figure 3.2** shows that the average external grants received by councils in London have been on a steady decline since 2012, which is also when the government-backed youth service funding began to decline.

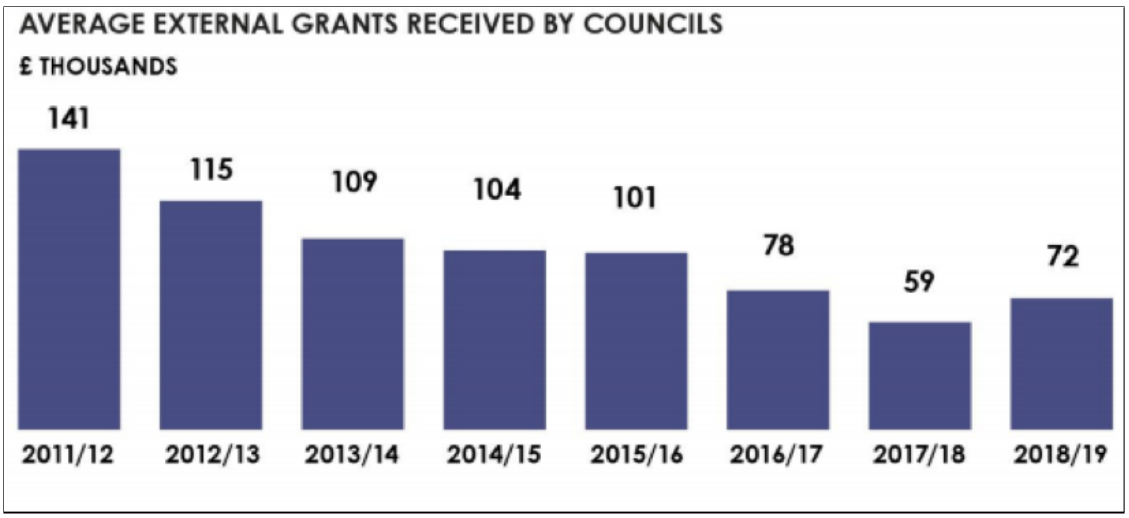

Figure 3.2: Average external grants received by councils (Berry, 2019)

In summary, it can be asserted that youth centres are the most popular investment of youth services, which not only provides a critical platform for people aged 15-25 to get help and support but also facilitates the development of future social skills in young people which can positively impact the community. Furthermore, it can also be identified that funding for these centres has been reduced substantially across London since 2012, including from government-provided funding and external sources to councils.

3.4 Correlation between rising Knife Crime and reduced Youth Service Funding

The previous two sections of this Literature Review have found that half of all knife crimes committed in London are by teenagers or younger children. Additionally, the advantages that youth centres provide for young children, including to keep them in a safe environment away from crime can no longer be utilised across 103 locations in London, as these have closed due to reduced youth service funding.

The APPG is a group of 40 Members of Parliament; they aim to "develop cross-party consensus from parliamentarians around new approaches to tackling knife crime, with particular focus on prevention and early intervention" (Dearden, 2019). APPG provides a reliable source as having a cross-party consensus means that their findings are purely factual and indicative of the entire government.

In a critical report, Deardon (2019) found that "councils with large cuts to youth services were more likely also to have seen an increase in knife crime in the area's police force". The report outlines that in terms of real-terms spending, the average council has reduced funding in youth services by 40% since 2016, with a handful of local authorities significantly reducing funding by up to 91% (Booth, 2019).

In correlation to this, the report finds that over the same period, there was a 68% increase in knife crime across England and Wales (Deardon, 2019). **Figure 3.3** provides a visual representation of this information; which shows a positive correlation between the two data points where it can be identified that the more youth centre closures there are, there is also an increase in knife crime.

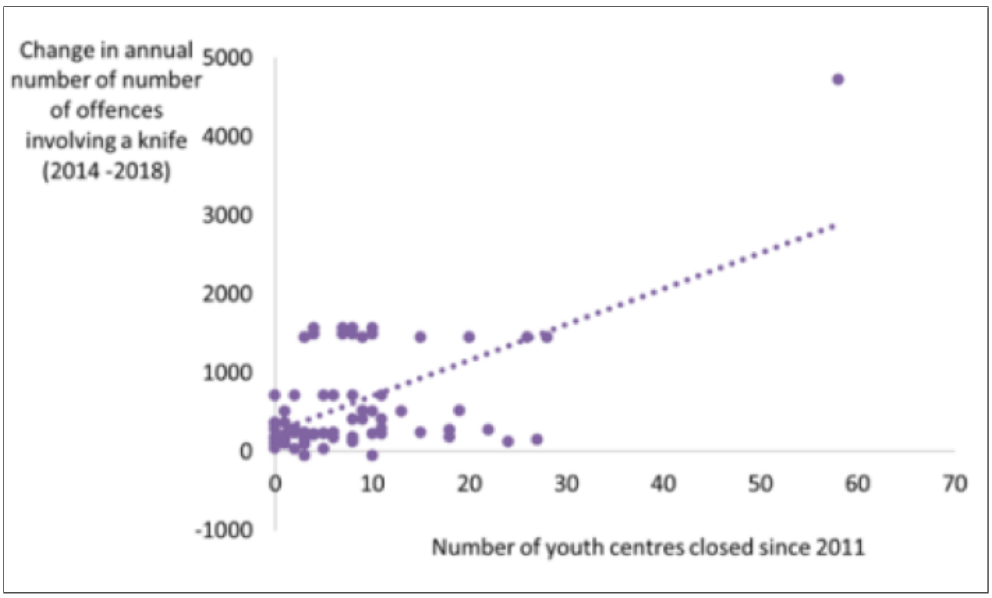

Figure 3.3: Correlation of annual knife crime offences and Youth Centre closures (Deardon, 2019)

Findings by the APPG can be supported by a study by the National Youth Agency, who also found that a primary cause of rising knife crime is a 'loss of year-round and voluntarily accessible youth services' (National Youth Agency, 2019). It goes further to explain that a well-funded youth service is necessary to prevent knife crime. The study continues to discuss a survey that was conducted on young people with the question;

"**If there were one thing you could change that you think would make young people safer, what would it be?**"

Two thousand two hundred participants responded - with the most popular response being the provision of more youth centres (The Youth Violence Commission, 2020). Overall, it can be determined that since reductions in youth service funding began, there has been a significant rise in knife crime over the same period, this observation is supported by multiple charities and government sources discussed in this section.

3.5 Visualising Data

Data Visualisation is used to help people to identify patterns, trends and correlations that would go undetected with data that is text-based. Data Visualisation software such as Tableau or SiSense facilitates the seamless transformation of text-based data into interactive and non-interactive visualisations (Rouse, 2012).

There are various types of visualisations, some of which are more suitable for specific purposes rather than general data. Infographics, geographic maps, heat maps and the commonly used bar and pie charts are the most popular variants of visualisations to show data. Modern computer-based visualisations allow for interactivity; the visualisation software Tableau, for example, allows the user to exclude parts of the data, filter data and select data which is used for analysis and querying.

Additionally, visualisation software allows for connection to various data sources; these data sources range from cloud storage platforms such as Hadoop or even a Microsoft Excel spreadsheet (Gour, 2019).

To build further on Rouse (2012), a report by Azzam et al. (2013) titled *'Data visualisation and evaluation'* provides a process to create an effective visualisation; where three requirements must be fulfilled. These requirements are;

1. Identifying if the data is qualitative or quantitative.
2. Ensuring the results in an image is representative of the raw data.
3. Ensuring the visualisations are readable by viewers and supports the exploration of the data.

Qualitative data and quantitative data can be visualised in various ways, with selecting an effective form of visualisation based on the different characteristics and attributes of the data. Qualitative data can include pictures, videos, transcribed interviews or even recorded conversations. Quantitative data can consist of geographical-based data, Likert-scale items (1=Strongly Disagree, 2=Disagree, etc.) and other numerically based sets of data. Numerically based sets of data are typically displayed with bar charts, pie charts or line graphs. As a result of these differences, varying approaches should be used to accurately display informative information from both data types. Using a visualisation technique that does not correspond with the kind of data will result in an uninformative visualisation.

After selecting an appropriate visualisation method, Azzam et al. (2013) continue to outline the importance of the second criterion, which is to ensure that the results of the visualisation are representative of the raw data being used. An inaccurate data set, for example, could omit important information or over represent a value. Legends on a visualisation should also be reviewed to ensure that visualisations are not unintentionally manipulated to provide an accurate conclusion.

Azzam et al. (2013) state that the third criterion exists not only to ensure accompanying discussions of the data should support the readable results of the visualisation by the reader, but, also to ensure that the visualisation itself is readable and is communicated effectively. He explains that even if criteria 1 and 2 are achieved, but criteria 3 has not been achieved - then the creator has failed to create an informative visualisation successfully.

Azzam et al. (2013) affirm that completing these three requirements will result in successfully producing a compelling visualisation.

3.5.1 Using different visualisations to present data

To build on Azzam et al. (2013), further research by Datalabs Agency (2014) outlines two basic types and common categories for visualisations. The two basic types of data visualisation are exploration and explanation. Exploration aims to help find a story that the data is showing the user, typically accompanied by an explanation. Explanation aims to tell a story to the user (Datalabs Agency, 2014). The study by Datalabs Agency (2014) also continues to provide standard categorisations of visualisations; these categories are Relationship, Comparison, Composition and Distribution, these categories can be seen in **Figure 3.4**.

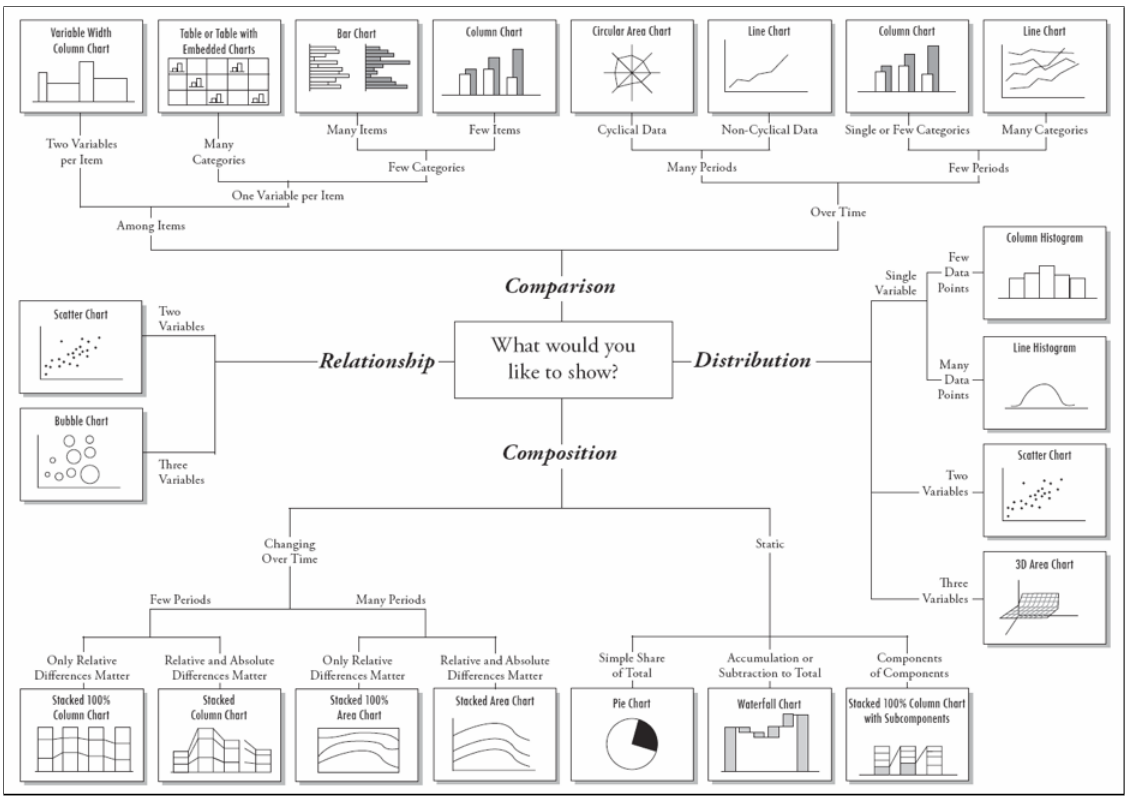

Figure 3.4: Choosing an appropriate visualisation (Datalabs Agency, 2014)

Datalabs Agency (2014) provides an infographic in **Figure 3.4** that can be used to identify what visualisation should be chosen based on varying different types of data. The findings in this infographic builds on the criterion of a successful visualisation discussed by Azzam et al. (2013). A scatter chart, for example, can be used to visualise a relationship between two types of data, this is more suitable than using a line chart for example as shown in **Figure 3.4**, this is especially important as scatter charts are used to show correlation (Chartio, 2019).

However, Datalabs Agency (2014) also indicates that different types of visualisations can be used in different contexts while still ensuring that they are appropriate, it also demonstrates that a scatter chart, for example, can be used for both relationship and distribution visualisations.

Bar graphs and geographical charts are the most popular visualisations. Bar graphs represent data using bars at different heights, proportional to the values that they represent (University of Leicester, 2019); this simple format allows for a straightforward interpretation of the data.

Geographical charts are used to present data in the form of a map, typically when the data is geographically based such as by using postcodes or areas in a dataset. This allows for a clearer understanding of distribution or proportion in a geographical region (Chou, 2019). Further research outlines that they have a high efficiency of transmitting information alongside aesthetics, Chou (2019) noted that geographical charts are commonly used in conjunction with other visualisations such as line graphs and bar graphs to emphasise the findings in the data further.

3.5.2 Authorities use of visualisation

To build on Chou (2019), the Department of Communities (DoC) published a report titled 'improving data visualisation for the public sector'. This report outlines that effective visualisation aids people to explore, understand and communicate data to others. (Smith et al., 2019). This report also explains that people can use datasets to identify patterns, trends and associations where visualisation is a vital tool to carry out an analysis. The table found in **Appendix C** shows what chart types authorities are using to convey the exploration of data using different visualisations. This report shows that authorities categorise data into what they're trying to display or what narrative they're trying to convey. This is comparable to what was found by Datalabs Agency (2019), where specific chart types are more effective when categorised based on what they're trying to show.

Further research into the authorities use of visualisation shows that the Scottish government employs a data visualisation manager and that public sector visualisations are typically used to figure out the best way to allocate resources to citizens (Aiton, 2018). This credible resource is especially valuable in justifying the use of visualisation for this project.

Furthermore, Aiton (2018) explains that data used for visualisations within the Scottish government need to be factual and accurate to ensure that visualisations produced are objective and impartial. This is especially important as previous visualisations used in Scottish government directly affected and supported societal changes, most notably of which was the Scottish governments review into the gender pay gap.

A visualisation for this was produced for Members of Scottish Parliament (MSP) by the data visualisation manager, with its recommendation to be viewed by an MSP to other members of parliament in a speech. As a result of this, it could be submitted that in addition to the private sector, visualisations are useful for presenting impartial and accurate information for the public sector which members of parliament can review and make essential changes.

Overall, research shows that by categorising visualisations into what narrative you are trying to show with a dataset, you will be able to provide an effective way to present data, especially for identifying patterns, trends and associations. Additionally, the use of visualisation has been useful in the public sector, as it was used to effectively convey data to MSP's about the gender pay gap.

3.6 Conclusion

In conclusion, this literature review has discovered crucial research that is necessary to link together the rise in knife crime and reduced funding for youth services. Over 50% of youth centres have been closed in London since 2012, which provides a safe and supportive environment for young people of ages 15-25. Over the same period, it was found that knife crime has risen to 32,986 offences in 2019, compared to 18,900 offences in 2012. In addition to statistical analysis of data involving youth centre closures and knife crime which shows a positive correlation, it can be submitted that from research, there is an emerging pattern developing in terms of rising knife crime and reduced funding in youth services.

Research into visualisation techniques has shown that visualisations are useful when categorised into what data they are trying to display, which allows people to gain a better understanding of data through visualisation, rather than in a text-based format. Additionally, visualisation is currently used by the Scottish government to provide visualisations to explain data for MSP's with effective results.

This project believes that research in this chapter has justified the development of a web application that authorities can use to identify areas of London where funding for youth service funding should be prioritised, tackle rising knife crime.

4

Product Research

This chapter will analyse the usability of similar existing products through introducing scoring their features against a criterion. This is beneficial as requirements can be elicited from features of these products that score highly against the criterion, which will result in this report building upon successful implementations from other similar products. The products that will be reviewed are web-based visualisation dashboards which provide crime statistics for London.

London Assembly Dashboard (LA)

Figure 4.1: (London Gov., 2019)

The London Assembly consists of 25 members of parliament, situated in various political parties to ensure that visualisations produced are politically unbiased and informative. The purpose of this dashboard is to allow the public to visualise various crime-related data measures for London. The measures are shown in a geographical visualisation across London in an interactive environment for the user, filters which can be used include knife crime, gun crime and acid attack offences.

MET Police Dashboard (MET)

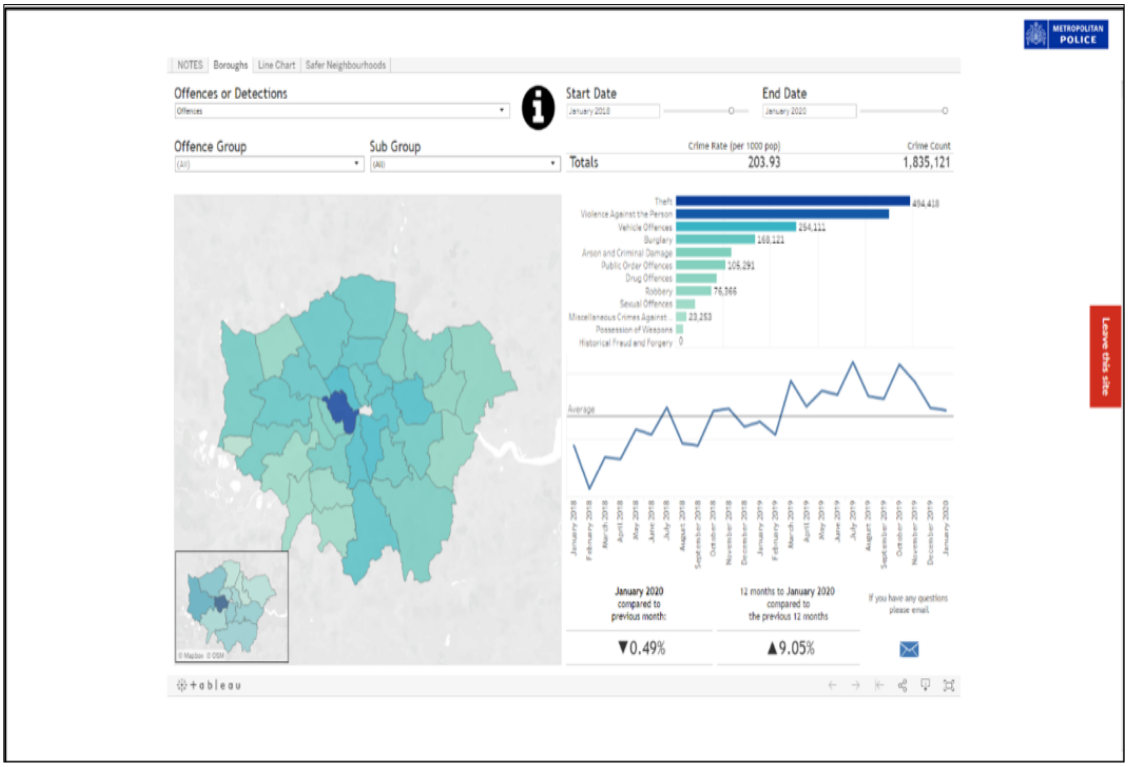

Figure 4.2: (MET Police, 2019)

The MET Police is the police force responsible for all 32 of London's boroughs. They have produced an interactive visualisation dashboard which is publicly available for use. It provides detailed information for users which can be filtered; filters include 'Offence Group' and 'Subgroup' which users can utilise to display specific information in hundreds of combinations.

4.1 Scoring Existing Products

Introduced in Chapter 2.1, Neilson's Heuristics consist of ten usability principles which a System can be evaluated against in order to identify issues within a user interface. These principles will be used in this report to identify problems within the existing user interfaces of available products.

By using these heuristics as the criterion, the usability for both products can be scored to gauge how well their user interface is designed, and if there are any issues that could be improved upon. This report will score how well each product meets each principle on a scale of 1-5, with a total score calculated at the end.

Measuring attitudes towards a statement based on intensity on a scale of 1-5 is commonly known as a Likert Scale. This has been chosen as it is the most widely used rating scale to measure how much an individual agrees or disagrees with a statement, with the strength of an attitude to be linear and measured numerically (Mcleod, 2020).

1 = Strongly Disagree 2 = Disagree 3 = Undecided 4 = Agree 5 = Strongly Agree

If this scoring system finds a functionality that scores highly against a principle in Neilson's Heuristics, the functionality can be used as a basis for a requirement in the deliverable of this report.

Analysis of the products meeting the criterion for Neilson's Heuristics can be seen in **Figures 4.3 - 4.6**. These annotations of the product are numbered according to the heuristic of the same number in **Table 4.1**.

Table 4.1: Scoring existing products

Heuristic	LA	/5	MET	/5
1. Visibility of system status	Feedback is represented using a dashboard-specific loading icon, with a grey tint over the dashboard with no indication of loading times. This is especially a concern as Tableau can run poorly on slow computers and is very memory intensive in a browser. With prolonged loading times for a small action, users could think that their computer is experiencing a problem such as a crash.	3	Feedback is represented using a dashboard-specific loading icon, with a grey tint over the dashboard with no indication of loading times. This is especially a concern as Tableau can run poorly on slow computers and is very memory intensive in a browser. With prolonged loading times for a small action, users could think that their computer is experiencing a problem such as a crash.	3
2. Match between system and the real world	Familiar words, concepts and phrases are used within the dashboard. No overly technical jargon is used which may confuse the user.	5	Familiar words, concepts and phrases are used within this product apart from the slight misspelling of 'sub-group'. With most of its text found within the visualisations away from the dashboard itself – the axis of the visualisations is very clear to the user and it would be difficult for the information to be mis-communicated.	4

Continued on next page

Table 4.1 – *Continued from previous page*

Heuristic	LA	/5	MET	/5
3. User control and freedom	Navigation away from the page - in any manner, is seamless and does not present any issues for a user.	4	If a user was to click something by accident, they would find it very simple to revert to the previous state of the application thanks to clear filters and controls. Furthermore, users have been provided a red button which can be clicked which will take them back to the previous website they were on, allowing the user to 'escape'.	5
4. Consistency and standards	This dashboard prevents the user from experiencing an error within the application, the most likely of which is a logic error in filtering in the dashboard. This is prevented by removing filters for the user where they are not available. If a user selects an 'Offence Group' filter for example, subgroups will be restricted to selections which encapsulate the chosen Offence Group.	2	This dashboard prevents the user from experiencing an error within the application, the most likely of which is a logic error in filtering in the dashboard. This is prevented by removing filters for the user where they are not available. If a user selects an 'Offence Group' filter for example, subgroups will be restricted to selections which encapsulate the chosen Offence Group.	5
5. Error prevention	In terms of consistency, the navigation pane at the top of the page is consistent throughout any dashboard page. However, contrasting colours in the line graphs and red/green visualisations makes the dashboard appear confusing and unpleasant.	5	This product provides information and a layout in a very clear structure. All colours are consistent throughout the pages and visualisations which allow for quick analysis of data.	5

Continued on next page

Table 4.1 – *Continued from previous page*

Heuristic	LA	/5	MET	/5
6. Recognition rather than recall	This product provides no recognition for the user, when the website that hosts the dashboard is reloaded, it reset the page that the user is on, along with any filters that they have chosen, meaning that the user will have to recall their steps.	2	This dashboard allows the user to redo and undo their actions. This can be highlighted in the bottom right of Figure 7. This allows the user to step through their actions, so they do not have to remember their path to the destination within the visualisation.	5
7. Flexibility and efficiency of use	This report believes that this product is catered towards more advanced users and provides less scope for 'beginners' to fully utilise its potential capabilities. This is because of its confusing layout and contrasting colours.	2	This product allows for quick adoption from beginner and advanced users, providing the tools needed for both groups of users to enhance their experience with the product. This is maintained through a simple layout of visualisations, navigation and clear filters.	5
8. Aesthetic and minimalist design	A lot of information is presented onto the visualisations and features poor minimalist design. The layout however is logical and allows for quick navigation between tabs.	2	This product follows a minimalist design, where information is always clear and readable, especially with the large and easy to read visualisations. The intensity of the blue colours in the dashboard indicates the volume of the measurement. Dark blue indicating a greater number than a lighter shade of blue for example.	5
9. Help users recognise, diagnose, and recover from errors	No errors are presented in normal operation. Although it appears that the dashboard is heavy in terms of memory usage, and on a less powerful computer it could be liable to freezing – which the user may be confused at.	4	No errors are presented in normal operation. Although it appears that the dashboard is heavy in terms of memory usage, and on a less powerful computer it could be liable to freezing – which the user may be confused at.	4

Continued on next page

Table 4.1 – *Continued from previous page*

Heuristic	LA	/5	MET	/5
10. Help and documentation	Suitable instructions are provided at the top right of the dashboard, which is visible in all visualisations.	5	No instructions are provided for the user in how to use the dashboard. Although the product is relatively simple to use, this would provide an additional way to help beginner users.	4
Totals:		**35**		**45**

Scores: LA Product = 34/50 | MET Product = 45/50

To provide a visual representation of the implemetation of Neilson's Heuristics, annotations of product screenshots can be seen in **Figures 4.3 - 4.6**.

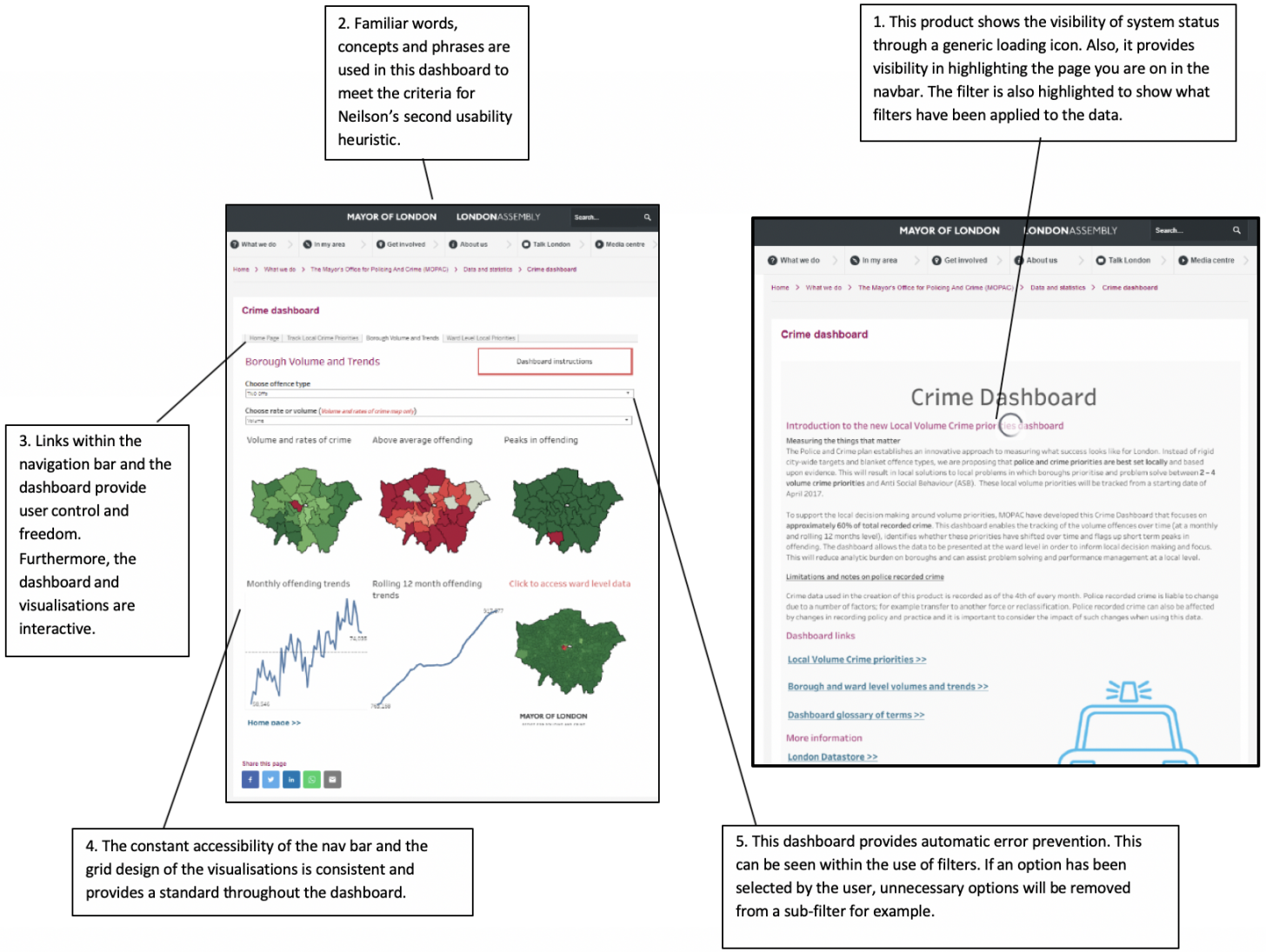

Figure 4.3: London.gov Product Annotations

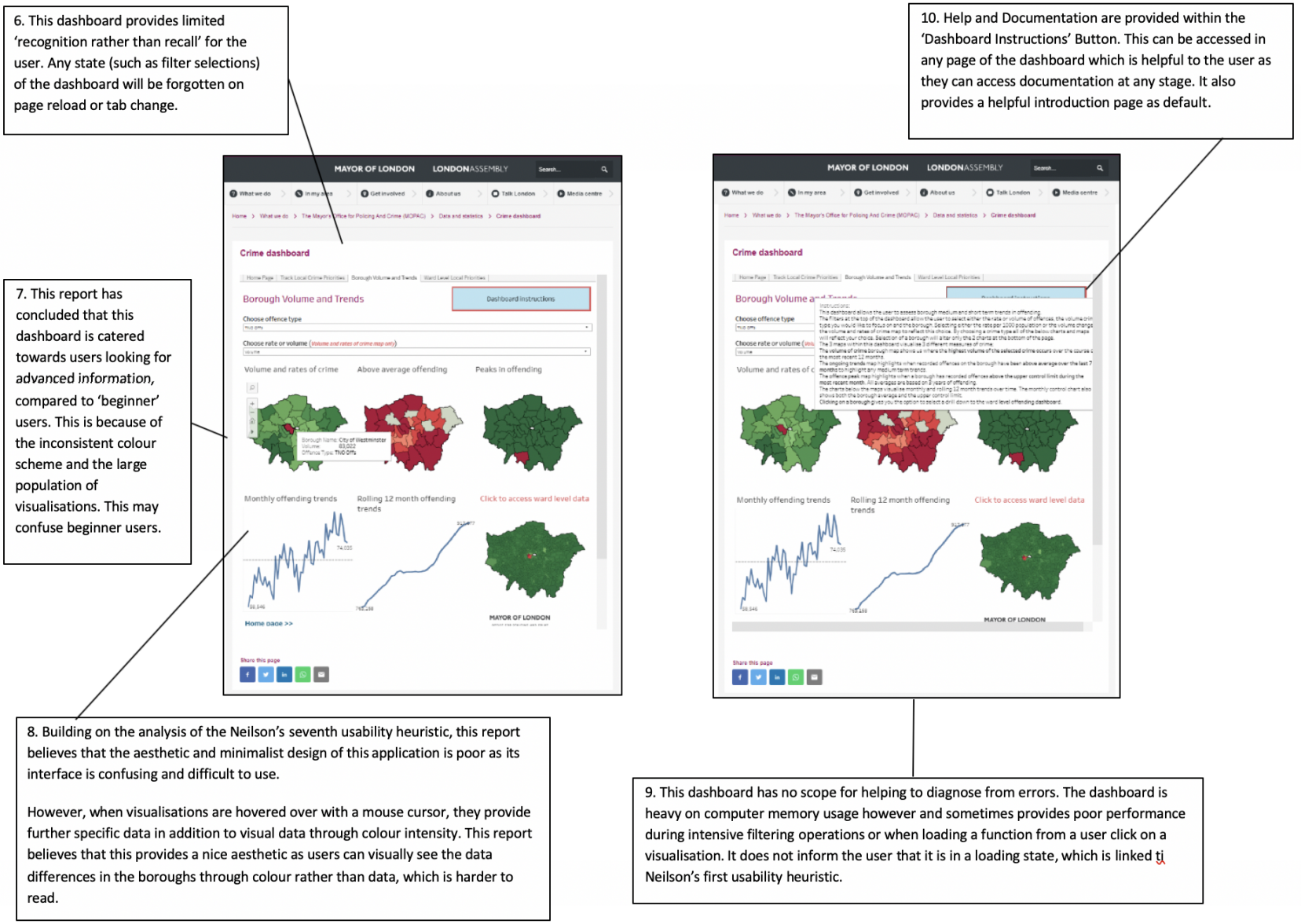

Figure 4.4: London.gov Product Annotations

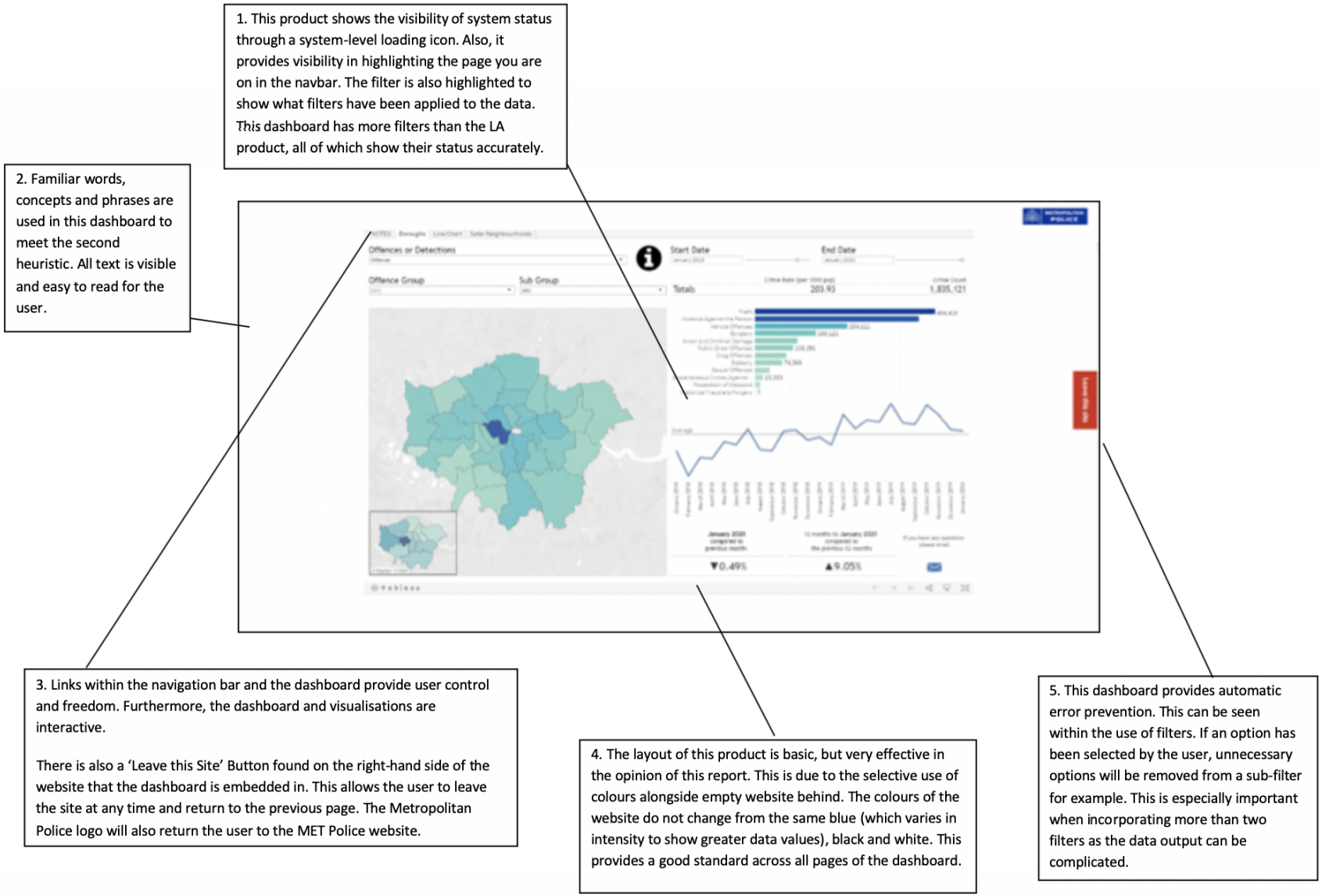

Figure 4.5: MET Police Product Annotations

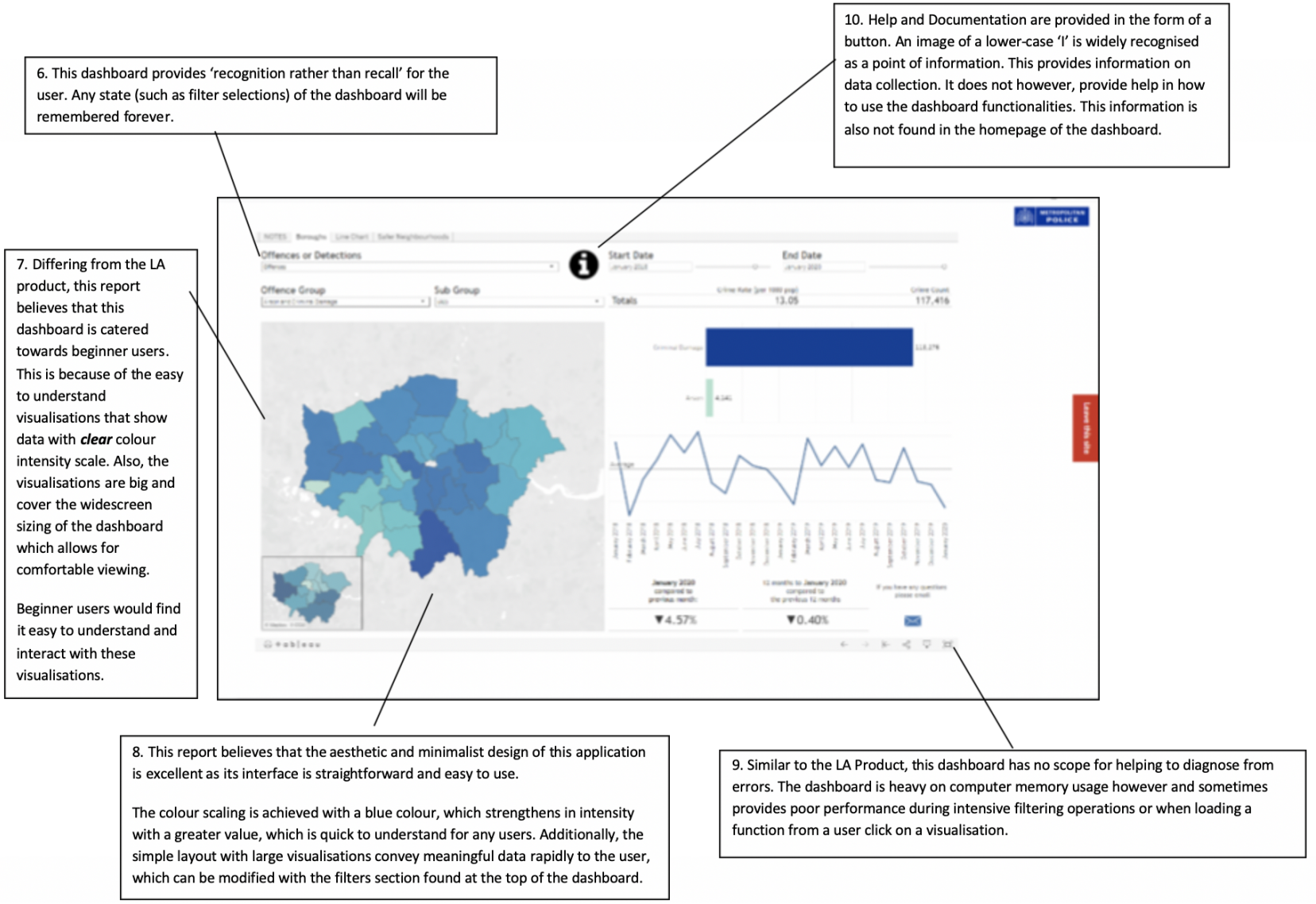

Figure 4.6: MET Police Product Annotations

Figure 4.3 and **Figure 4.4** show annotations made to screenshots of the London Gov. product during research. These annotations show how Neilson's usability heuristics have been implemented into the product, and how effective they are in providing an effective user experience. This project found that this product scores poorly in terms of aesthetics and user experience because of its confusing colour scheme and complicated layout.

Figure 4.5 and **Figure 4.6** show annotations made to screenshots of the MET Police product during research. In terms of Neilson's usability heuristics, this product scored highly in user experience because of its aesthetically pleasing colour scheme and 'open' layout. Visualisations in this product are clearly defined and follow a consistent layout throughout all pages.

4.2 Conclusion

In summary, this chapter has identified that both dashboards provided visualisations in the form of geographical charts, line graphs and numerical data also. These charts display current trends for varying types of crime, categorised in London's boroughs. This is especially advantageous for a user who understands the geography of London, as they will be able to instantly locate boroughs where knife, gun or acid attack crime is most prevalent.

Furthermore, both dashboards provide line graphs that are linked with the geographical visualisations; selecting a specific borough for example will filter this data in the line graph also. Interestingly, Chou (2019) noted that geographical charts are commonly used in conjunction with other visualisations such as line graphs to emphasise findings. The MET product follows this finding to emphasis data by incorporating both graphs in their dashboard.

In terms of usability, this report found that the MET Product provided a user interface that closely met Neilson's usability heuristics, scoring highly in against all of the critereon, whereas the LA Product scored poorly in sections relating to aesthetic and minimalist design and flexibility and ease of use. As a result, this report is more likely to elicit requirements from the MET Product.

5

Legal, Social, Ethical and Professional Considerations

This chapter will identify any potential Legal, Social, Ethical or Professional Issues that may affect this project. Additionally, these issues will be discussed to consider how they will be addressed in the report and the prototype.

The developed product will include login functionality for users. Typical login information that is stored for a user contains the following sensitive information; Name, Username and Password. As a result, this project will need to ensure that the seventh principle of the DPA is adhered to which states that data must be processed in a manner that ensures appropriate security of the personal data and login data (ICO, 2019). Additionally, a cookies notice should display to a user on the system to declare what cookies are active and what they track, this is currently required by the European General Data Protection Regulation (GDPR) (GDPR, 2020).

Additionally, this project will need to ensure that connected databases are secure and are suitable for the product. This can be achieved by encrypted communications between the user and the login server, for example, to ensure security. Building on the use of encrypted communications, further security measures could be implemented, including cryptographic password hashing to prevent storing plain text attributes in a database (Brown, 2013).

Hashing acts as a one-way mathematical function that is near impossible to reverse. This is used for turning plaintext passwords into a scrambled cypher (Greenberg, 2016). The system checks to see if this hash matches the users hash in the database during the login process - if they match, the user can login.

Throughout development, this project will make use of third-party open-source frameworks, languages, and copyrighted software to achieve its objectives. Software that was not produced by this project but adapted will be acknowledged; this is to ensure that it cannot be miscommunicated what code this project has developed and what has not been developed, this is to avoid any form of plagiarism. External images or scripts used on the application will be clearly cited in this report.

In terms of further ethical considerations, the Literature Review (see Chapter 3) and Product Research Chapter (see Chapter 4) of this report has found that usability is critical in enabling positive use of interactive software. As a result of this, accessibility features will be implemented into the deliverable to ensure that users with visual impairments, for example, will be able to experience the product without any issues in terms of usability. During implementation, the popular Web Content Accessibility Guidelines (WCAG 2.1) document will be referred to frequently. Used globally for web accessibility advice, WCAG 2.1 is updated regularly by the World Wide Web Consortium (W3C). It contains recommendations to make web content more usable and accessible for users with disabilities (W3, 2020). As a result, an accessibility page will be implemented to provide links to recommendations for users; this is so that they can have a comfortable experience using the system.

6

Requirements Analysis

This Chapter explores the functional and non-functional requirements necessary for achieving the development of a system that will allow users to interact with data in relation to London knife crime and Youth Centres. Following DSDM techniques, the requirements will be prioritised for development using MoSCoW prioritisation, this will enable requirements to be ranked in terms of importance.

Requirements can be elicited from information gathered from the Literature Review and Product Research. The Literature Review has produced requirements such as visualisation organisation and how this affects the readability of data to users. This requirement can also be enforced by information gathered from Product Research, where a minimalistic and consistent approach to presentation was favoured for usability. Further requirements have been added to implement in the Literature Review and to improve upon the capabilities of existing products. This chapter contains all of the functional and non-functional requirements for the system.

6.1 Functional Requirements

Functional requirements are features or functions that are implemented to allow the user to accomplish their tasks (Functional and Nonfunctional Requirements: Specification and Types, 2020). Requirements elicited from the product research (see Chapter 4) can be seen below.

These requirements have been selected as this Project believes that they were essential for both products in providing a complete user interface in line with Neilson's Heuristics and as a result, they have been elicited as requirements for this project. Requirements that this project has elicited follow this list.

6.1.1 Requirements elicited from Product Research

- Dashboard: Geographical Visualisations
- Dashboard: Line Graphs
- Dashboard: Bar Graphs
- Dashboard: Year Filters
- Dashboard: Blue Colour Scheme
- Dashboard: Percentage Statistics
- Dashboard: Help Button
- Dashboard: Support Email
- Dashboard: Home Page

These requirements will be incorporated into the visualisation dashboard as the Product Research Chapter (see Chapter 4) found that these aspects of existing, similar products functioned well to effectively convey information from raw data to the user.

6.1.2 Other Functional Requirements

- Users can log in and out
- Users can change password
- Users can register for a new account
- Login information is hashed + salted
- Dashboard: Borough Filters
- Dashboard: Navigational Tabs
- Dashboard: Tree Maps
- Dashboard: Research Page
- Administrators can login
- Contact Us Page
- Accessibility Page
- About Us Page
- Account Settings Page
- Web App Home Page
- Administrator Settings Page
- Dashboard: Total amounts
- Dashboard: Page Introductions

In addition to requirements elicited from similar products, requirements listed above will also be implemented in order to create a web-based platform that users can access to use the visualisation dashboard. Creating a web application for the visualisation dashboard to be hosted on is essential so that users can access the dashboard at all times.

6.2 Non-functional Requirements

Non-functional requirements are necessary to enable the implementation of functional requirements outlined for the system. Non-functional requirements typically detail the characteristics of functional requirements and therefore, compliance an important aspect of this section as it outlines the operation of a system and how it should operate to fulfil the principles of relevant laws, acts and guidelines.

In conjunction, relevant LSEPI issues that were discussed in Chapter 5 will be a further basis of selecting useful non-functional requirements. Chapter 5 discovered the importance of the DPA, BCS Code of Conduct and the WCAG guidelines and how these should be used for various objectives such as for safely storing user information or accessibility. The non-functional requirements for the system have been described below.

- Information must be stored securely
- Data must be accurate, with explanations concerning consistency
- Consistent colour scheme of visualisations
- Prevention of unauthorised access (check if user is signed in)
- Able to run on all systems (scale to viewport)
- All Fields must validate user input
- Database transactions must follow ACID principles
- External use of code must be credited
- Consistent Design / Layout
- Application must be accessible for all users
- Restricted access to administrator settings
- Documentation in how to use the system
- System state must always be explained to a user, to prevent them from being confused during long operations.
- Errors must be handled well throughout so the application does not crash

6.3 MoSCoW Analysis

In order to produce the deliverable of this project within its time constraints, MoSCoW Analysis will be used to rank the importance of every functional requirement. After deciding importance using a scale of 'Must Have, Should Have, Could Have and Won't Have', this project will be able to prioritise requirements that provide the most immediate benefits for the deliverable; these requirements will be implemented in the first prototype. After these requirements have been delivered, there will be more time to develop the remaining, more complex requirements.

Furthermore, the MoSCoW analysis in this project will follow the '80/20' Prioritisation Rule. By implementing requirements that provide the most immediate benefits in the first prototype (such as the core login system and website in this case), 80% of the project can be delivered in the first 20% of time into development, allowing more time for the implementation of complicated requirements.

Table 6.4: MoSCoW Analysis of Functional Requirements

Requirement	MoSCoW	Timebox
Users can log in and out	Must Have	1
Account Settings Page	Must Have	1
Users can change password	Must Have	1
Accessibility Page	Must Have	1
Users can register for a new account	Must Have	1
Login information is hashed + salted	Must Have	1
Web App Landing Page	Must Have	1
Contact Us Page	Should Have	2
Dashboard: Geographical Visualisations	Should Have	2
Dashboard: Line Graphs	Should Have	2
Dashboard: Bar Graphs	Should Have	2
Dashboard: Tree Maps	Should Have	2
Dashboard: Borough Filters	Should Have	2
Dashboard: Year Filters	Should Have	2
Dashboard: Percentage Statistics	Should Have	2
Dashboard: Support Email	Should Have	2
Dashboard: Total amounts	Should Have	2
Dashboard: Navigational Tabs	Should Have	2
Dashboard: Page Introductions	Should Have	2
Dashboard: Help Button	Should Have	2
Dashboard: Home Page	Should Have	2
Dashboard: Research Page	Should Have	2
Dashboard: Blue Colour Scheme	Should Have	2
Administrators can login	Could Have	3
Administrator Settings Page	Could Have	3
About Us Page	Wont Have	3

Table 6.4 shows that prioritisation has been focused on the core functionality of the system.

This core functionality is primarily based on the login system for users, with the visualisation dashboard and administration-specific systems being delayed until a further iteration of the system. With its core functionality implemented in the first prototype, visualisations and other functions can be implemented at a further stage in development.

6.4 Timeboxing

To successfully meet the project deadline, the finalised requirements discussed during the MoSCoW analysis have been organised into three timeboxes, each of which contains requirements that need to be implemented within a pre-determined frame of time, which in turn allows this project to follow the DSDM principle 'deliver on time'. The planned development time of each timebox are discussed in **Table 6.5**. The allocated time for each timebox follows the Gantt project schedule seen in **Appendix B**.

Table 6.5: Timebox Objectives

Timebox	Objectives	Time
1	Once complete, the first timebox of the system will include the login and registration system for the user. This also includes the architecture of the web application, such as the templates for all pages and working hyperlinks between them, this provides a foundation to develop the remainder of the requirements.	3.5 Weeks
2	The second timebox will include the development of the visualisation dashboard. This includes linking the data from the database to the dashboard, and to design the visualisations, such as the graphs, colour schemes and other functionalities.	3.5 Weeks
3	The third timebox will look to include administration functionalities, including an administrator login system and a way to allow an administrator to update the data in the database.	3.5 Weeks

7

Design and Implementation

This chapter of the report documents the methods used to design logical approaches to the system in addition to user interface design. The process of designing Use Case Diagrams, ERD Diagrams and Low-level Prototypes will be discussed.

7.1 Use Case Diagram

This section describes how a UML (Unified Modelling Language) Use Case Diagram can be used to visually model the functionality of a system, through displaying actors within and outside of the system and their interactions with the system. A use case includes any different actions, services and functions that the system can execute (Uml-diagrams, 2020).

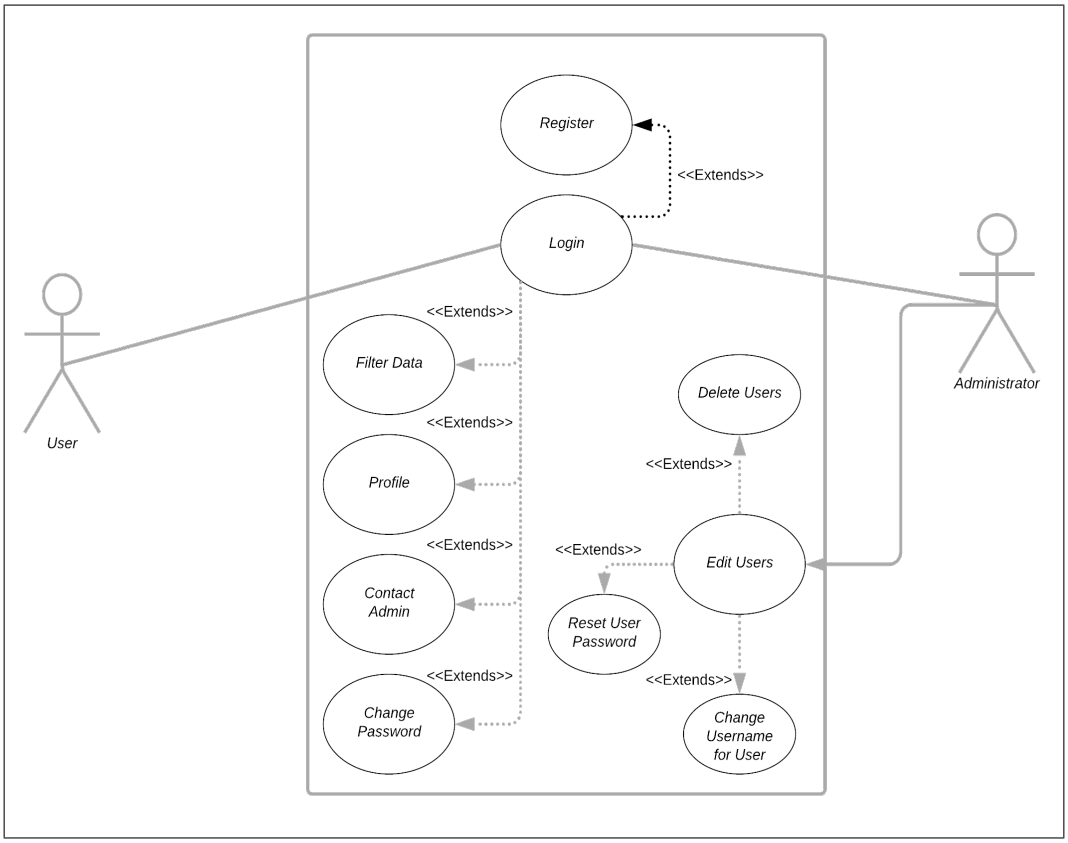

Figure 7.1: System Use Case Diagram

Figure 7.1 summarizes that the system has two main actors, the Users and the Administrator. Various use cases are available to the user within the 'logged in' state in the environment. They can access their profile for example and change their password. However, not all use cases within the system environment are available to the user, as some of these are reserved for the Administrator. These functionalities include being able to edit database and user records.

7.2 Entity Relationship Diagram

This project will need to design and implement a database in order to have a system that can continually access data. As well as visualisation data, a database will also be used for storing login information for users. Databases are typically modelled using Entity Relationship Diagrams (ERD), these diagrams provide conceptual information which a developer can use to implement the architecture in an ERD. An ERD defines the relationships between various entities (People, Concepts or Objects) (Lucidchart, 2020). Entities have attributes that define the properties of that entity. For example, an entity 'user' will have an attribute of 'password' associated to it.

The relationships in **Figure 7.2** are presented in Crows Foot Notation. The connection used between entities in this ERD is called 'One or Many'.

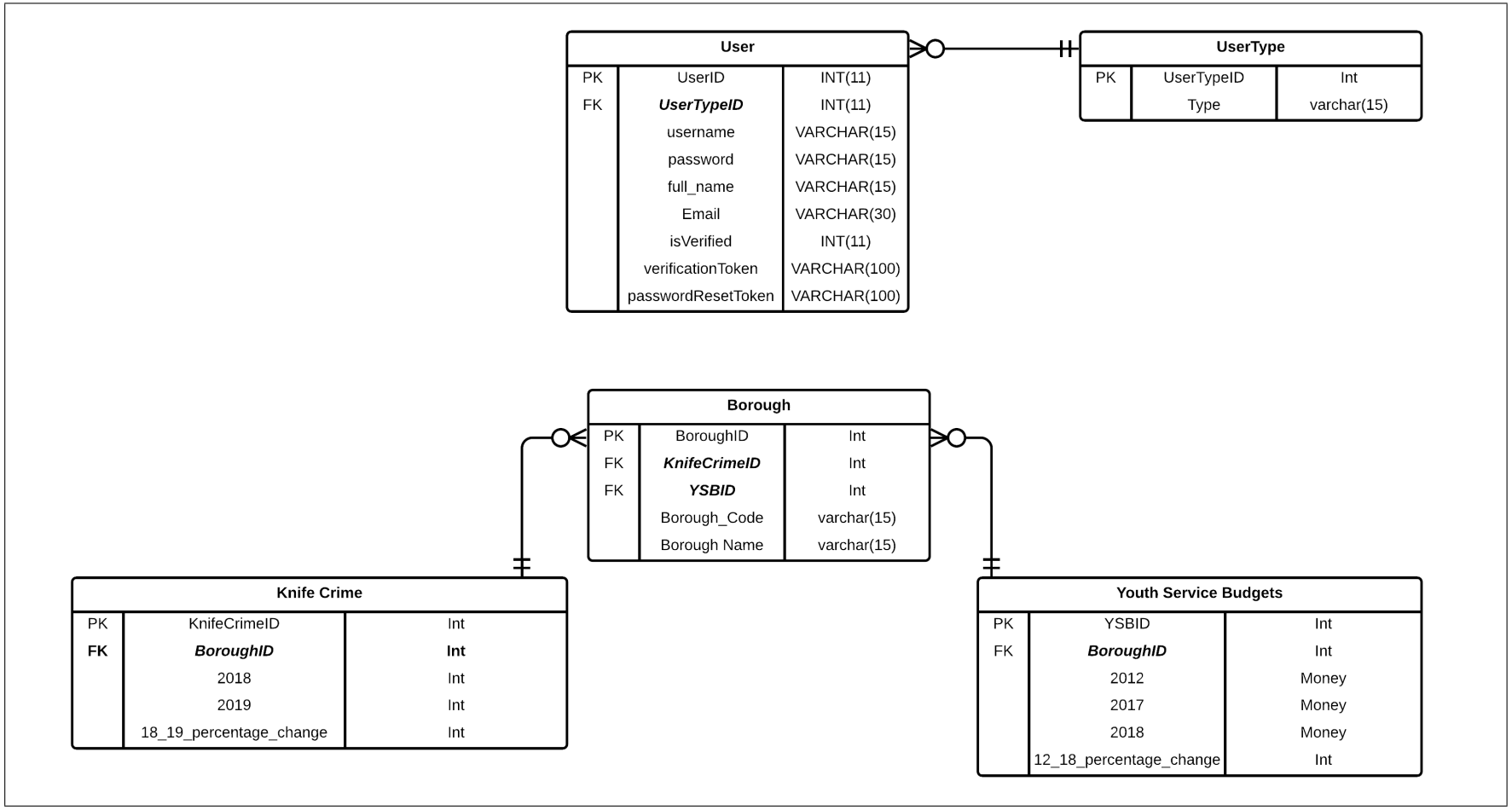

Figure 7.2: System ERD Diagram

The entity "User" in **Figure 7.2** contains attributes such as first name, last name, password and email. It can be identified that this entity has a relationship associated with UserType. The UserType entity will be used to store information regarding a user's level of access, such as if they are an Administrator, who has greater system privileges. In terms of the visualisation data, various relationships can be made, with the common attribute being the borough that the data was recorded in. For this reason, the borough is the link between all relevant data within the database as it is the common denominator across all related data sets, this can be identified in **Figure 7.2**.

7.3 Low-level Prototypes

Low-level prototypes will be used in this section to produce a draft visual representation of the concepts and designs discussed previously in this report. Through performing requirements analysis, product research and producing high-level conceptual designs, this report is now able to translate these concepts into an expression of design intent (Babich, 2020), through creating low-level prototypes.

Moreover, scoring similar products has enabled this report to gain a greater understanding in terms of ensuring that Neilson's Heuristics are met for the high-fidelity prototype. Low-level prototypes in this section will discuss how the criteron of various heuristics are met within the designs.

These prototypes can be used as a reference for designing the user interface during development of the high-fidelity prototype. **Figures 7.3 - 7.7** show the hand-drawn low-level prototypes used for this project.

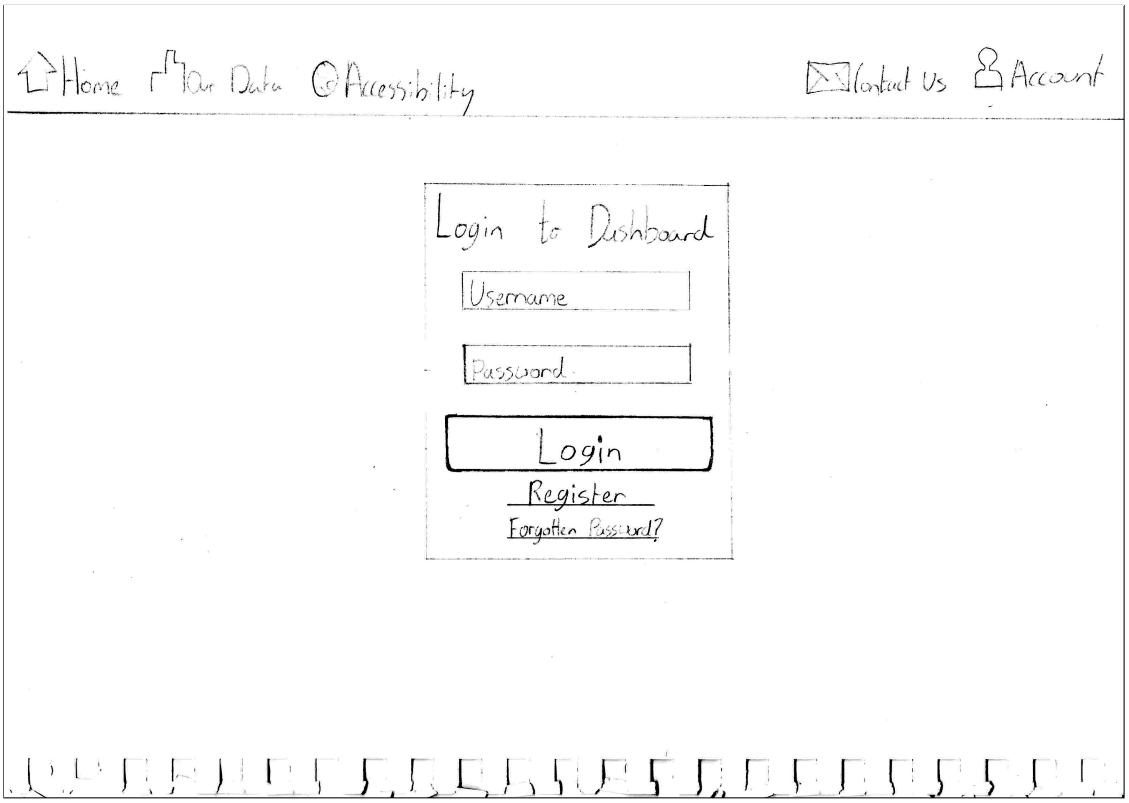

Figure 7.3: Login Page

Figure 7.3 shows the landing page for the web application. Here, users will be met with a sign in form which can be used to login to the application. In terms of additional Neilson Heuristic considerations, this page will inform the user of the current system status during the login process, informing the user if they have entered in an invalid username or password etc.

This landing page will also help to introduce a consistent theme throughout the prototype in terms of design and layout. A navigation bar will be introduced in the landing page for example, and will be static at the top of every page from this point onwards. By introducing the navigation bar in this instance, users can quickly identify that they are able to navigate to other pages of the web application at any point during their use of the system.

Another consistent theme is introduced here in the form of 'cards'. A card in this prototype will act as a container for content. **Figure 7.3** shows that the login form is situated in a rectangular white 'card' in the middle of the screen. Displaying page content in a container provides a consistent format for the prototype to deliver information to a user in an appealing, center-screen format. Additional wireframes follow this same consistent theme and can be seen in **Appendix N**.

The content of this landing page was discussed with the supervisor and second-marker of this report. Most significantly of which was the lack of information regarding the purpose of the web application for new users. Changes were suggested to the landing page to include a body of information about the web application and its purpose, and to move the sign in form to its own dedicated page which users can navigate to.

Additionally, changes to the design of the landing page were also discussed, these changes included to 'modernise' the layout, as currently, its design could be considered very basic. This project feels that this feedback is invaluable in providing a comfortable experience for a user, and will be taken forward during development of the high-fidelity prototype.

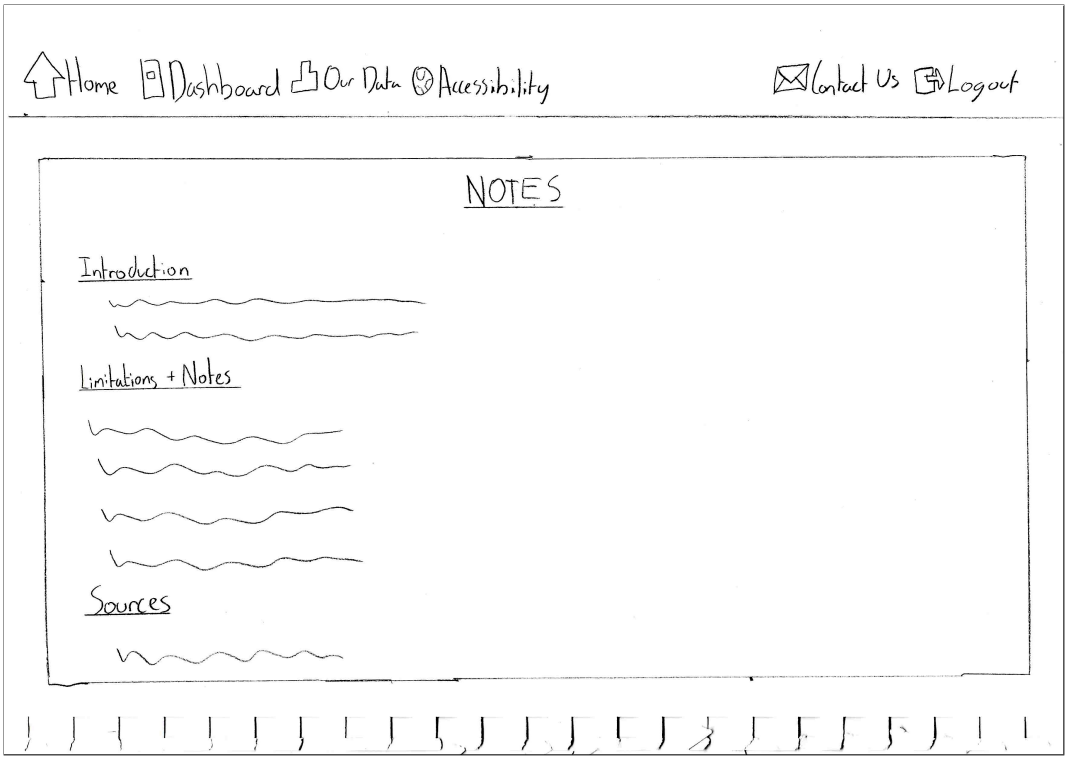

Figure 7.4: Dashboard Home Page

Figure 7.4 shows the home page for the visualisation dashboard. This provides information for the user on the intended target market and what the dashboard can be used for. In addition to usage information for the web app, this page introduces the user to how the dashboard functions, what actions can be taken and how to proceed with interacting with the visualisations.

Following this will be a Research Page, which will quickly summarise findings from the literature review in this report. Available articles will also be provided so the user is able to view secondary sources used for research.

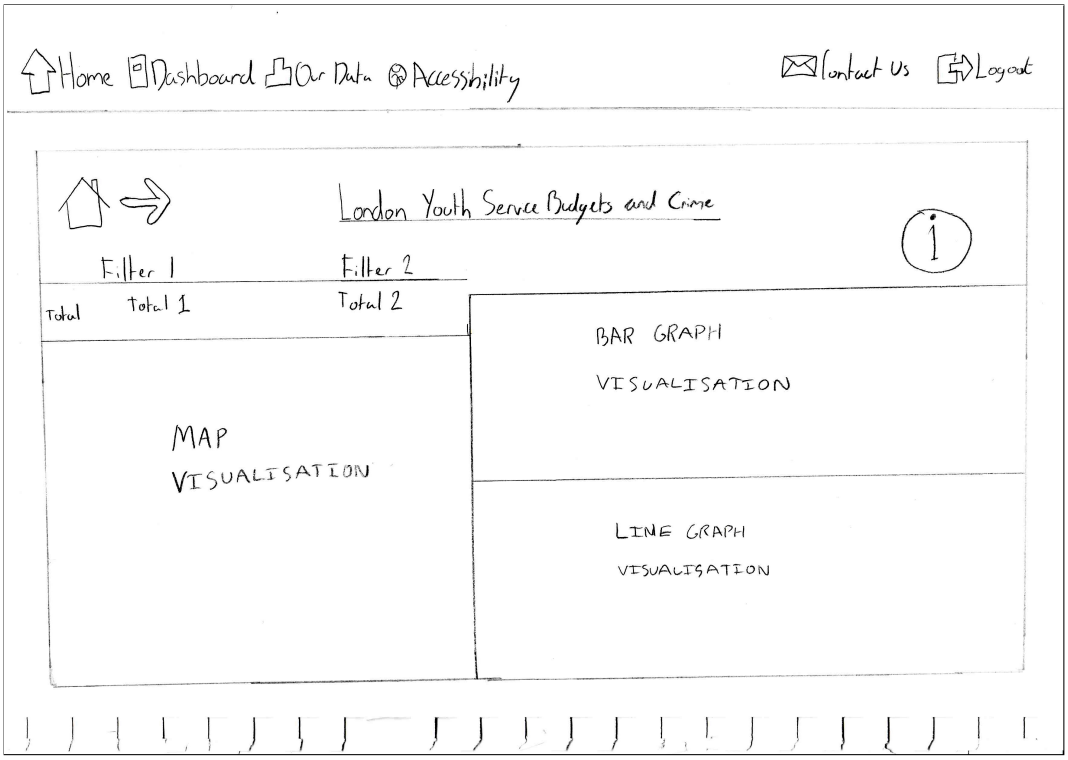

Figure 7.5: Dashboard Main Visualisation Page

Figure 7.5 shows the main visualisation page of the dashboard. By inner-joining Youth Service Budgets and Knife Crime data sets together, this page can display and link data together based on London's Boroughs.

This page contains a geographical map visualisation, bar graph visualisation and a line graph visualisation. Filters have been included on this page to allow the user to filter data to specific boroughs or a year range. A similar product (MET Police) discussed in Chapter 4 followed a similar layout of visualisations and scored highly against Neilson's Heuristics. This same layout of visualisations has been included in **Figure 7.5** as a result.

Additionally, product research demonstrated that the inclusion of an information button is critical in assisting new users. Consequently, this project has implemented an information button in all of the dashboard wireframes containing visualisations. This can be hovered over to provide information to assist new users regarding functionality.

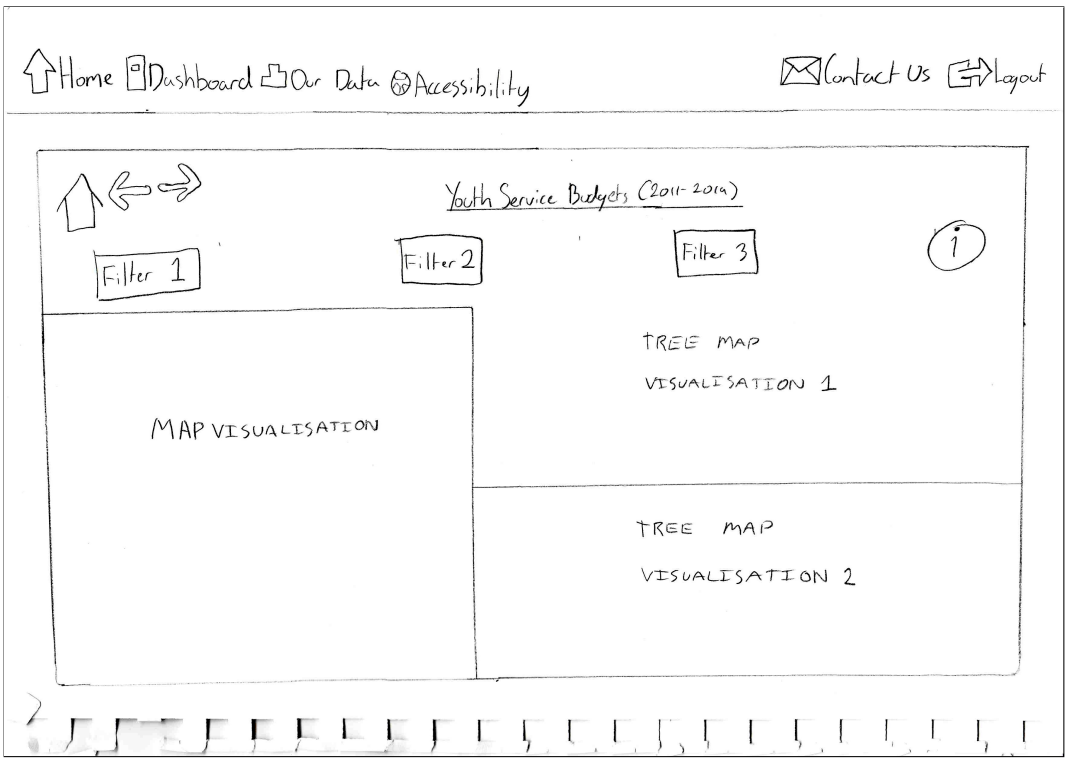

Figure 7.6: Dashboard Youth Service Budgets Page

Figure 7.6 shows the second page of the visualisations, which focuses on Youth Service Budgets data. This page provides visualisations that users can interact with to gain further insights of youth service budgets on specific London Boroughs.

Similar to **Figure 7.5**, this page contains a geographical map visualisation, which, aesthetically, allows the user to identify what borough is being selected and its position in comparison with other London boroughs. Additionally, two tree map visualisations are included here to demonstrate the proportional size of budgets compared to neighbouring boroughs. Filters are included to select specific boroughs.

Figure 7.7 shows the third page of the visualisations, which focuses on Knife Crime data. This page provides visualisations, that can be used to interact with London Knife Crime data from 2011 to 2019. Data can be filtered to years and specific boroughs in this page.

Moreover, this page of the dashboard will use a combination of a geographical map, line graph and bar graph to highlight year on year changes in knife crime in London over the past decade. Filters can be used in this page to select boroughs and a year range also.

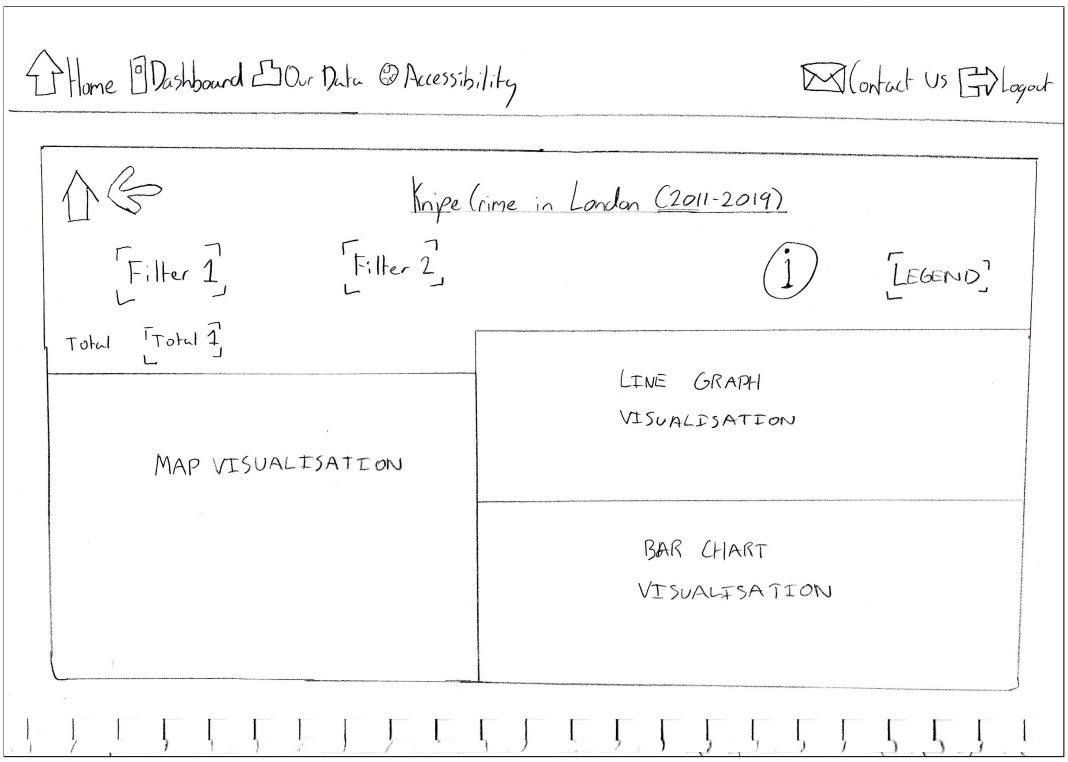

Figure 7.7: Dashboard Knife Crime Page

This section has introduced four key low-level prototypes that will be used as the basis for designing the user interface for the high-fidelity prototype. Additional wireframes for the prototype were produced for every page and can be seen in **Appendix K**.

8

Development

This chapter describes how the application was developed and what techniques were used to implement the finalised requirements of the system within the timeboxes. A review of technology is used to assess available technologies which can be applied to the project. In addition to this, code and screenshots of the prototypes will be annotated throughout to highlight functionalities.

8.1 Review of Technology

Web-based applications can be developed in a variety of languages. This section will start by comparing existing languages and frameworks that can be used for the development of the prototype. Secondly this section will review existing database engines and their suitability for development. Finally, this section will conclude by reviewing available visualisation software.

8.1.1 Development Frameworks

The advantages and disadvantages for popular web application frameworks are discussed in this section, the results of which can be seen below in **Table 8.1**.

Table 8.1: Review of development frameworks. (Macdonald, 2015), (Rubygarage, 2017), (AltexSoft, 2019), (Hasan, 2017), (Goel, 2020).

Framework	Advantages	Disadvantages
ExpressJS	- Provides asynchronous NodeJS performance. - Minimalistic. - Very fast, due to the NodeJS engine. - Increasing in popularity.	- Not suitable for heavy computational tasks. - New framework compared to alternatives. - Confusing code organisation. - Excessive use of callbacks.
ASP.NET	- Written in C#, a popular language. - Powerful and flexible. - Built-in windows authentication. - WYSIWYG Editing.	- Requires extensive MVC knowledge. - Limited control over HTML. - No design page preview. - Complicated deployment.

Continued on next page

Table 8.1 – *Continued from previous page*

Framework	Advantages	Disadvantages
Ruby on Rails	- Library support is extensive. - Clear and easy to read. - Highly Scalable. - Secure language.	- Referred to as being 'slow'. - Slow booting time. - Poor documentation. - Better alternatives right now.

The proposed system does not require the use of computationally-intensive capable frameworks or an extensive use of third-party libraries – and for this reason the chosen framework should be lightweight in terms of memory use and very responsive. Moreover, learning complex programming architectures such as MVC is not necessary to build the application for this project.

Table 8.1 shows that the Express framework accurately fulfils the requirements for the development of the system. Express is the 'go to' framework for developing web applications using the Node.js runtime, which is minimalistic and lightweight – a concept that this report has discussed frequently that is necessary for the product.

Node.js is a cross-platform environment that executes JavaScript code outside of a browser (Carnes, 2020), the architecture aims to optimise scalability in web applications and real-time web applications whilst being lightweight in memory usage. It allows for a web application centred entirely around using JavaScript.

Using only one language (instead of a different server and client-side language, which is the typical system for web applications) allows for a consistent approach to development, debugging and testing.

Table 8.2: Review of Database Engines. (Chand, 2019), (Arsenault, 2017), (Guthe, 2019).

Engine	Advantages	Disadvantages
Oracle12c	- Sets the bar for other RDBMS's - Strong Performance - Robust management tools - Backup and recovery	- Not open source - expensive. - Significant resources needed. - Complicated engine. - Difficult to learn quickly.
MongoDB	- Easy to scale. - Easy to install and setup. - No complex joins needed. - Extensive documentation available.	- Document database, not RDBMS. - Inconsistent, by design. - Data joins are not possible. - Excessive memory use.
MySQL	- Free to use. - Extensive functionality. - Fast, reliable and scalable. - Documented well.	- Stability issues. - Not full SQL-compliant. - Poor performance scaling. - Open Source Limitations.

According to db-engines (2020), Oracle is consistently ranked as the most popular database engine; because it offers strong performance and provides features that are unmatched by other engines. However, an Oracle database is expensive to maintain and consumes significant resources, it also is generally not as quick to get online than that of an open-source MySQL database – as these can be hosted anywhere.

MongoDB is known as a NoSQL database; it uses documents similar with the characteristics of JSON (JavaScript Object Notation) to unstructured data. This document-based system differs from the usual database format of SQL, where data is stored in tables (MongoDB, 2020). Due to the use of documents instead of tables, a NoSQL database does not support 'joins', which are crucial in this project to match data from different data sets to London Boroughs.

Table 8.2 shows that the MySQL engine fits the objectives of this project appropriately as it is reliable, scalable and supports data joins. MySQL's scaling capabilities are an advantage as significant data input would not have a dramatic effect on its performance. Furthermore, it is the most popular open-source SQL database engine, so it has plenty of support and documentation available during development.

8.1.2 Using Tableau

Tableau is a powerful data visualisation software used within the Business Intelligence Industry (Intellipaat, 2014). Data can be connected to and extracted from, using data found in a variety of places including any database engines such as Oracle or MySQL, where the latter will be used in this project.

Importantly, Tableau provides such an intuitive interface that it allows non-technical users to create complex visualisations and dashboards, as programming skills are not needed – this simplicity in creating visualisations is reflected upon analysis from any user; visualisations created are easy to read and can be understood at all levels within an organisation (Guru99, 2020).

Tableau visualisations and dashboards are created using the desktop application of Tableau, these can be shared online via Tableau's website, or embedded into a webpage using JavaScript. The use of JavaScript is advantageous as it is the primary development language and as a result, embedded JavaScript code can be implemented quickly.

9

Prototyping

This chapter introduces system features and provide brief discussions of their implementations in the final high-fidelity prototype.

During development, this project iterated through multiple prototypes, changing designs and functionalities based on user feedback. This section shows the final prototype and describes the process of implementing of the functional requirements (see Chapter 6) during development. In order to track changes to the system code, local development machines had a connection to the master repository hosted in GitHub (www.github.com) via the git versioning control system. Changes that were 'committed' to the master repository were logged and allowed this project to track changes to effectively produce prototypes for the system. Three prototypes were made in total, previous prototypes and brief descriptions of their functionalities can be seen in **Appendix E** (Prototype One) and **Appendix F** (Prototype Two).

9.1 Timebox One

9.1.1 Landing Page

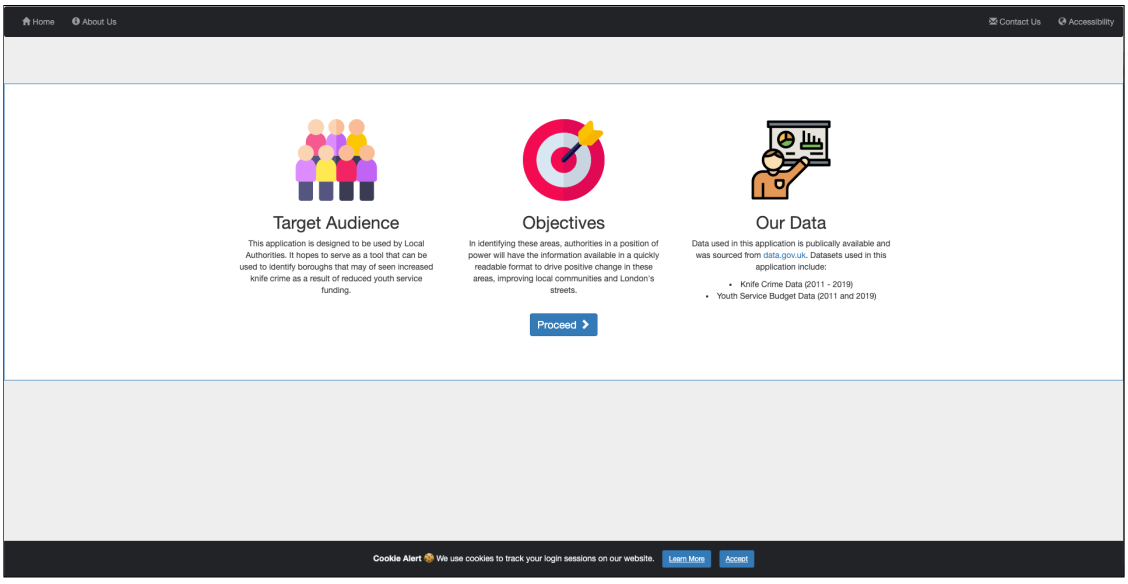

Figure 9.1: Login screen for the prototype

Figure 9.1 shows the default landing page for the application, this page provides information regarding the use of the website for new users, set out in a modern 'carousel' format. Feedback from the project supervisor and second marker inspired various iterations to this screen, where it was discussed that this page provided limited information for a new user initially. Past versions of the landing page can be seen in the previous prototype appendices, **Appendix E** and **Appendix F**.

Most components on this page such as buttons, fields and the navbar were built using the Bootstrap CSS framework, which focuses on responsive front-end web development and rapid prototyping (Bootstrap, 2020).

The navbar contains hyperlinks to other pages in the application. Some of these links are protected by authentication with server code. Attempting to access the /user page without being logged in will redirect you to the sign in page for example. This is so protected routes cannot be accessed by unauthorised users. An explanation of this can be found in **Appendix D.1**.

When a user is logged in, the navbar will display an additional dropdown option to reflect this named 'My Account'. This dropdown will allow a user to access their account settings or to log out.

Furthermore, a 'cookies policy' notice is included in the application in line with considerations from the LSEPI research (see Chapter 5). Users can click to 'Learn More', which will inform the user of the use of cookies to track login sessions, or to accept the notice. Upon accepting the cookies policy, the response will be saved for 365 days in the users browser to prevent further notifications.

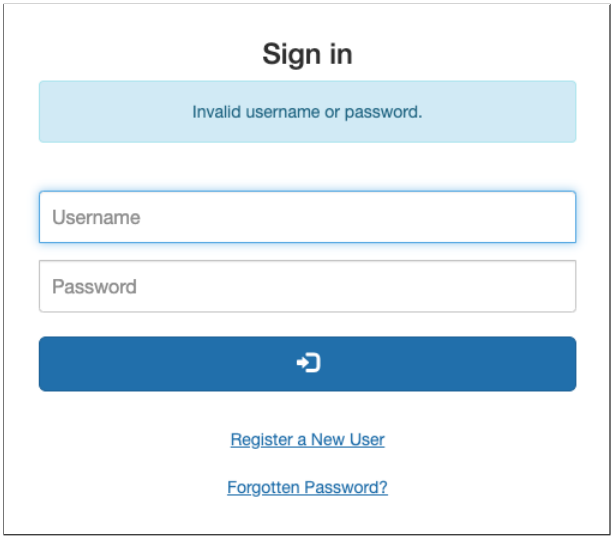

Figure 9.2: Sign in page showing the current system status

Figure 9.2 shows the alert message that will appear to inform the user that their login was unsuccessful due to invalid credentials. Producing alerts like this keep the user informed of the current system state, in line with Neilson's Heuristics. Throughout the system, alerts are presented to the user including error and success messages.

The login script is handled using PassportJS authentication middleware. Unfamiliarity with this module meant that boilerplate code was adapted from William (2020). Their example implementation was critical in providing a secure login process for this prototype.

Following discussions of LSEPI considerations from Chapter 5, this project will implement an advanced security technique known as *hashing and salting* in order to prevent exposing users password as plaintext in HTTP transfers and the database.

To define *hashing*, Jung (2020) states that the process of hashing involves putting a password into a hashing algorithm (this project will use SHA-1), which can turn plaintext passwords into a series of numbers and letters. To login, modern systems will "hash" this series of numbers and letters and compare it against that found in the database, if they match, the user can login. This comparison against the database for the prototype can be seen in Line 8, **Listing 9.1**

Jung (2020) further explains that *salting* is the process of adding a string to the plaintext password *before* it goes into a hashing algorithm. This makes its hashed output even more unpredictable to brute force attacks, for example.

Listing 9.1: Authenticating a hashed and salted SHA-1 password

```
1
2       var salt = '7fa73b47df808d36c5fe328546ddef8b9011b2c6';
3       salt = salt+''+password;   // Concatenate salt and password
4       var encPassword = crypto.createHash('sha1').update(salt).digest('hex');
            // Create sha1 hash
5       var dbPassword = rows[0].password; // Crawl database to see if password
            exists
6
7       // If the hashed password not found, then produce message.
8       if(!(dbPassword == encPassword)){
9         console.log("\n *** Login Unsuccessful *** \n")
10          return done(null, false, req.flash('message','Invalid username or
                password.'))}
11        console.log("\n *** Login Successful *** \n")
12        req.session.user = rows[0];
13      return done(null, rows[0]);
```

Listing 9.1 shows how passwords are hashed and salted to be submitted into the database for login. A SHA1 hash of a user's password is compared against the password in the database with that username. If the values match, then the user will be logged in, otherwise an error will be thrown.

Maintaining password security is especially important to fulfil the principles of the Data Protection Act. Encrypting passwords means that they are never seen in plain text in any point of the backend login process. Hashed passwords are a cryptographically secure string of numbers and letters, which will be held in the database, meaning that anybody who has access to the database cannot see the user's passwords in plaintext and thus, they are not able to login with that information as the hash cannot be reversed back to plaintext for login.

A plaintext password of 'London' concatenated to the salt with the value of;

"7fa73b47df808d36c5fe328546ddef8b9011b2c6"

Will result in;

"7fa73b47df808d36c5fe328546ddef8b9011b2c6London"

This new string will be hashed using SHA-1 encryption into;

"df31ec3e7336846b0882f874aed5c12565247a9b"

A full explanation of the login process can be found in **Appendix D.1**.

9.1.2 Registration Page

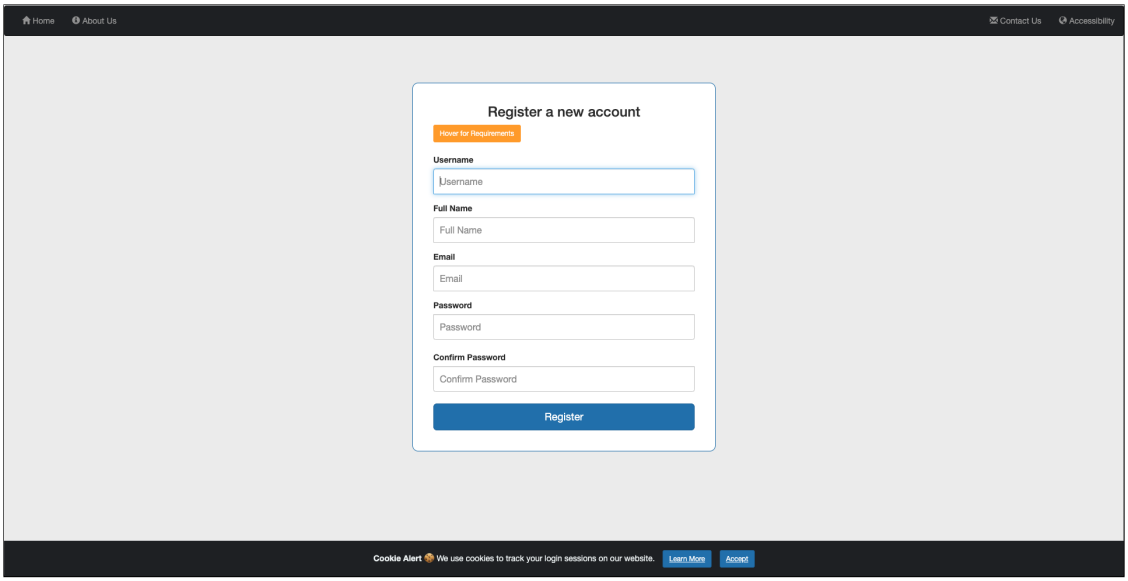

Figure 9.3: Registration Page Screenshot

A user having the ability to register for a new account is a requirement for the first timebox. Similar to the login process, registering a new user is accomplished by using a post form. This is supported by validation and field sanitization to ensure that the data submitted for a new user maintains its integrity for the database. For example, entering a password shorter than five characters is restricted as well as entering an email without an '@' symbol. Field sanitization is useful for ensuring that inputs from users are 'free of noise' (express-validator, 2020). An explanation of the registration process can be seen in **Appendix D.4**.

Listing 9.2: Registration Code

```
1  app.post("/register", [
2    check('username', 'Please enter a valid
         username').not().isEmpty().trim().custom(async username => {
3      const value = await isUsernameInUse(username);
4      if (value != 0) {
5          req.flash('message', 'Username count is ' + value)
6      } }),
7    check('fullname', 'Name must be over five characters
         long.').not().isEmpty().isLength({ min: 5 }).trim().escape(),
8    check('email', 'Please enter a valid email
         address.').not().isEmpty().isEmail().normalizeEmail().trim().escape()
9    .custom(async email => {
```

```
10      const evalue = await isEmailInUse(email)
11      if (evalue != 0) {
12        done();
13      }}),
14    check('password', 'Password must have a minimum of 5
          characters').not().isEmpty().isLength({ min: 5 }).trim().escape(),
15    check('confirmPassword', 'Passwords do not
          match').not().isEmpty().custom((value, {req}) => {
16      if (value !== req.body.password) {
17        return false;
18      } else {
19        return value;
20      }})], function (req, res) {
21
22  \\... Other Code
```

The backend code handling the validation and sanitisation functionality of the registration page is shown in **Listing 9.2**. Any errors are returned as a JSON response – which is then parsed to catch and handle validation or sanitisation errors.

In order to validate users, this system utilises an 'email verification' system. When a user successfully registers, they will be prevented from logging in until they have clicked the verification link, sent to their registration email. This link will send the user to 'https://www.HOST.com/verify'. In order to match a link click to the correct users account, every user will be matched with a unique salted SHA1 string string, which appends onto the /verify link sent to the user. An example of a link a user could receive in their verification email could be;

'https://www.{HOST}.com/verify?=**651b86124e72659b4987cdaec1ed8ba2a9cffad2**'

The hashed value after '/verify?=' is passed as a query parameter to the /verify endpoint. This hashed query parameter will be checked to see if the value exists in the database. The database value is known as the 'validationToken', which can be seen in the ERD for the system in **Figure 7.2** (see Chapter 7). If the value exists, the matching user record will have their 'isVerified' value set to '1', which is the MySQL Boolean equivalent to the boolean value 'TRUE'. Setting the isVerified value to true will enable the user to login successfully to their account as they have completed the account verification process. A full explanation of the email verification process can be seen in **Appendix D.4**.

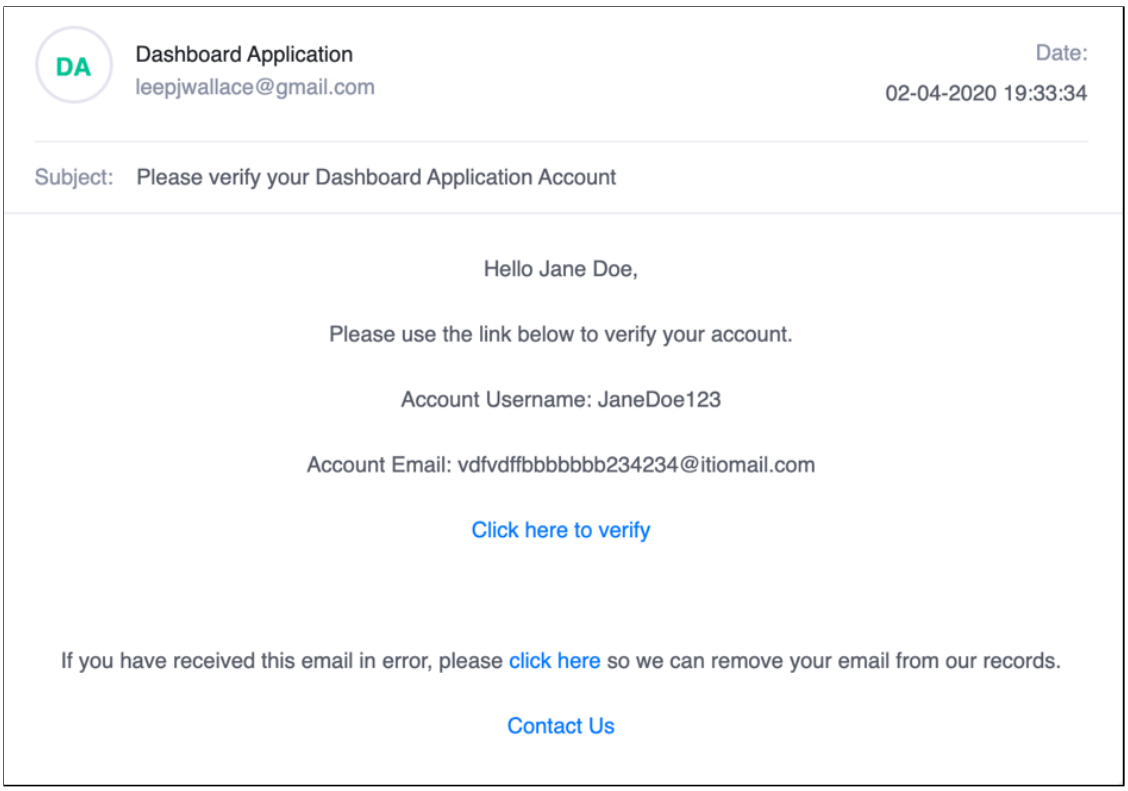

Figure 9.4: Email received after registering for an account

An example email that a user would receive after registering for a new account can be found in **Figure 9.4**. It provides a link for account verification, deleting the account and a contact us link. A full walkthrough of the registration process can be found in **Appendix D.4**.

9.1.3 Account Settings

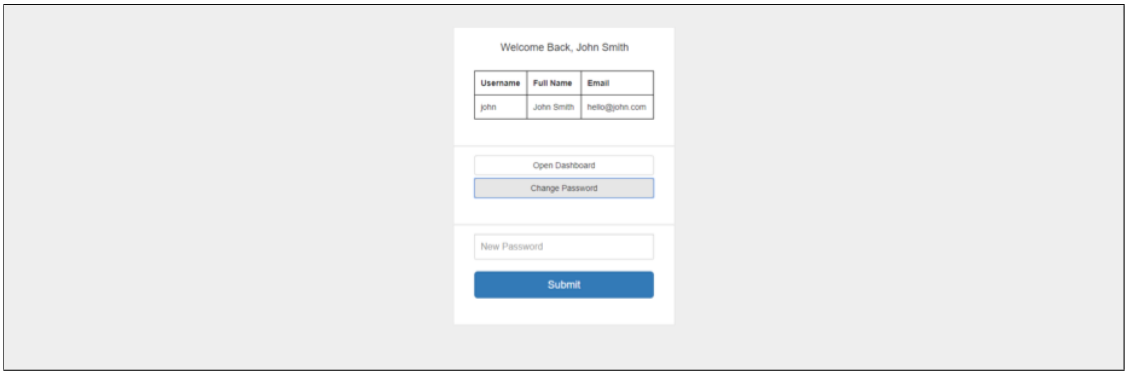

Figure 9.5: Account Settings Page Screenshot

The Account Settings page allows a user to see their account information and provides options to change this information where possible, such as an account password. Information from the account is parsed as variables into the page using router command in **Listing 9.3**. Beneath this table of information, there are various buttons which can be used for instance access to relevant pages, such as the Dashboard, Account Settings and Administrator Settings if applicable.

Listing 9.3: Rendering Account Variables

```
1 res.render('user', { username: username, full_name: full_name, email : email,
      user : user });
```

9.1.4 Logging Out

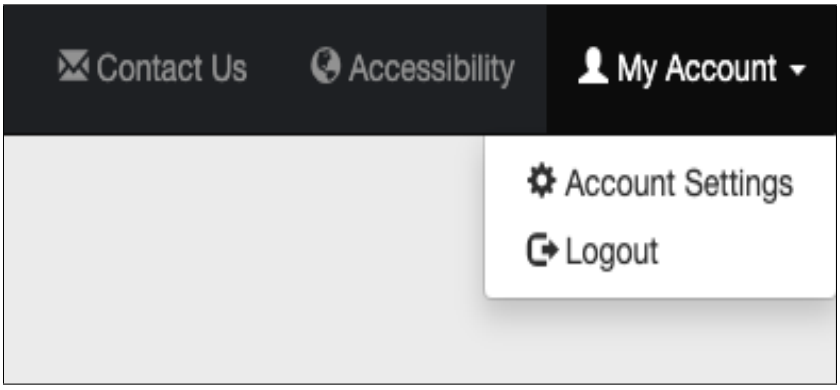

Figure 9.6: "My Account" dropdown options

When logged in, the user will be able to access a 'logout' button, under the new 'My Account' dropdown in the navbar. Which will safely log them out of their account. This destroys the

passport.js authentication session, logs the user out and redirects them to the sign in page.The JavaScript function used to execute this can be seen in **Listing 9.4**.

Listing 9.4: Logout Function

```
app.get('/logout', function(req, res){
    req.session.destroy();
    req.logout();
    res.redirect('/signin');
});
```

9.2 Timebox Two

9.2.1 Dashboard

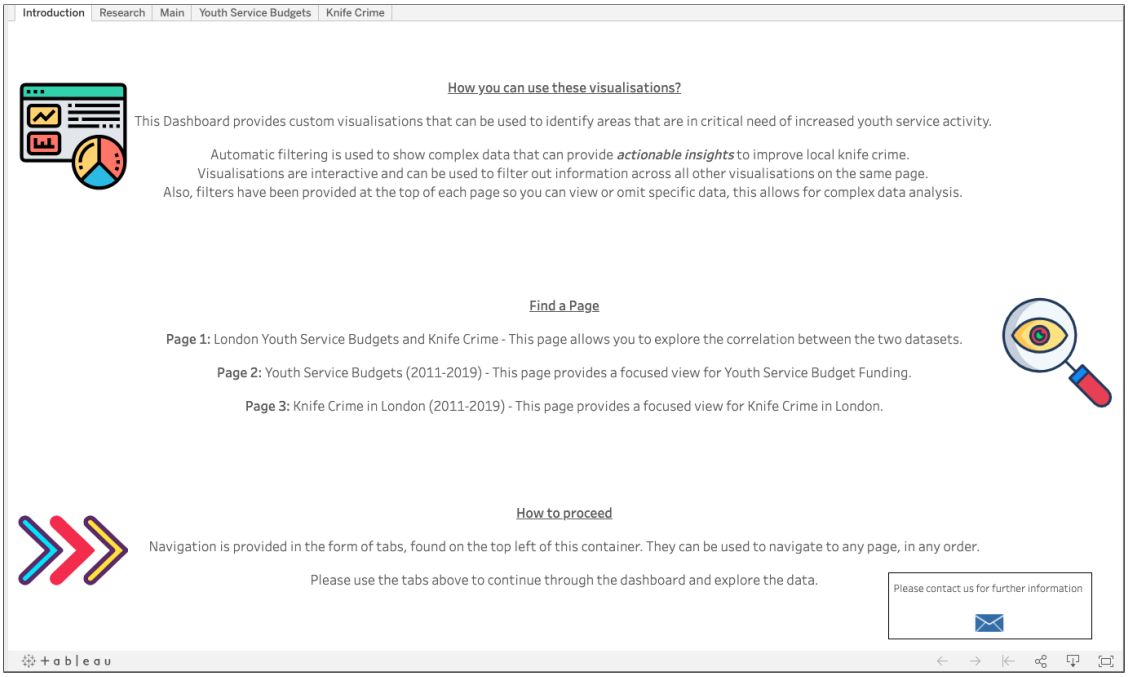

Figure 9.7: Dashboard Landing Page

Figure 9.7 shows the first page users will see when loading up the Dashboard. This page introduces the user to the contents of the dashboard and contains information about the target market and what they can gain out of using the system.

If a user has any questions or would like further information regarding the dashboard, they have been provided with an interactive e-mail button, which when pressed, will produce a template email to an administrator.

Figure 9.8 shows the research page in the dashboard. This page includes information about research discovered during the Literature Review in this report (see Chapter 3). Articles are also provided for a user to view.

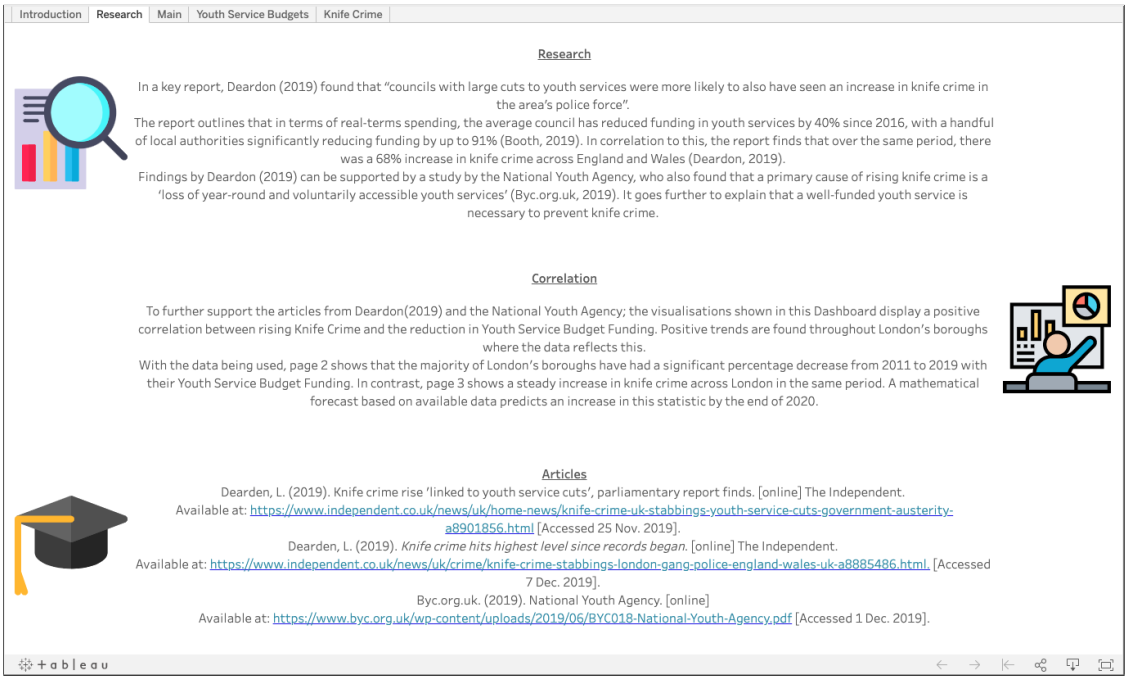

Figure 9.8: Dashboard Research Page

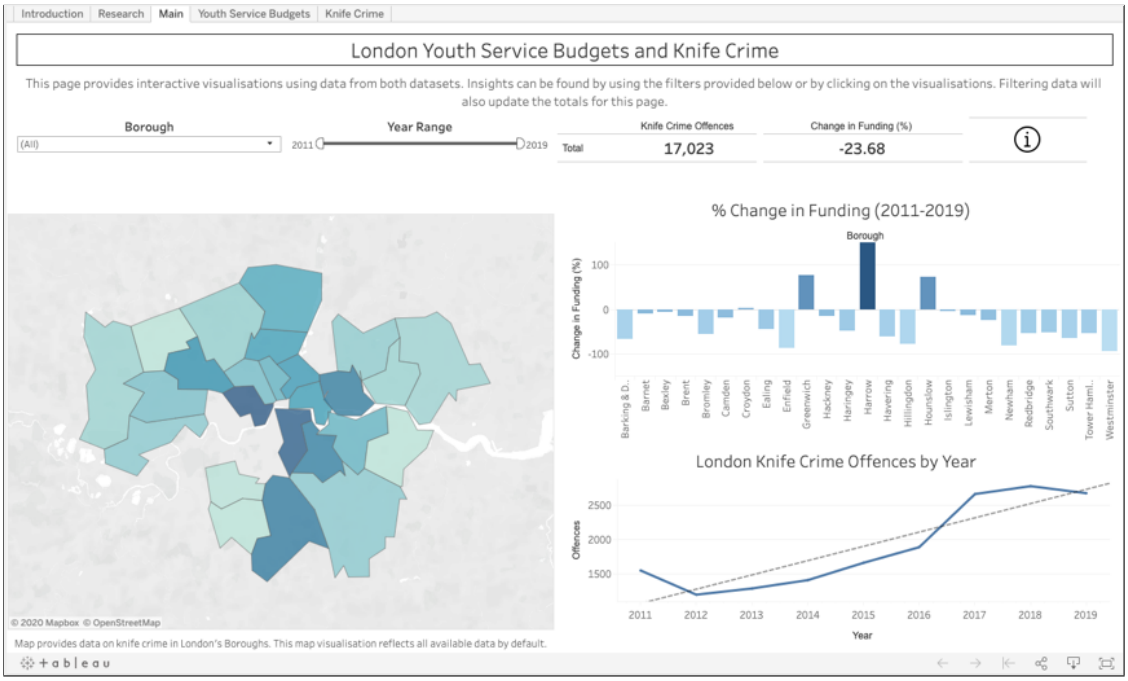

Figure 9.9: Dashboard Main Visualisation Page

Figure 9.9 shows the main page of the dashboard. This page provides a short description to introduce the user to the data used within this page. Filters on the page include the Borough and the Year Range. These filters allow the user to view a specific portion of data, especially when used together.

Fields have been added to indicate totals, which change with the data selected. Every main page of the dashboard includes an 'Information' Button to help the user in using the interface.

The visualisations include an interactive map of London, as well as a bar graph and a line graph. All three are interactive and can be used as filters, clicking a borough on the map will filter data from the line graph and the totals for example. This page of the dashboard displays information regarding knife crime and the % change in youth service funding for boroughs in London.

This page of the dashboard keeps a consistent theme of blue, white and black in the layout as an aesthetically pleasing colour scheme was elicited as a requirement during product research (see Chapter 4).

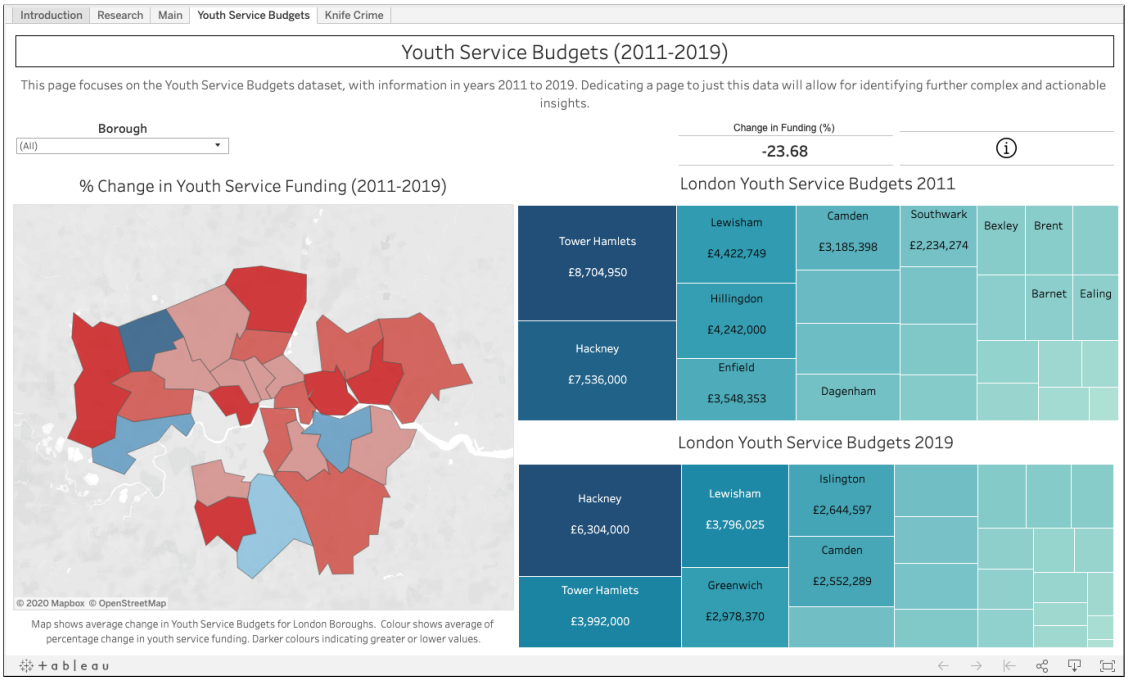

Figure 9.10: Dashboard Youth Service Budgets Page

The second page of the dashboard is shown in **Figure 9.10**. This provides a detailed overview of Youth Service Budgets. It also includes filters and an information button. This page has a map visualisation and two tree maps visualisations. The map visualisation shows the percentage change in youth service funding in London from 2011 to 2019. Positive values are shown using blue, negative values are shown using red with darker shades indicating greater values.

Although including the colour red in the dashboard does not follow the blue, white and black colour scheme that is prevalent in other pages, during development this project understood that it was important to highlight the significance of reduced youth service funding and found that using this technique is effective in conveying a disparity in values to a user.

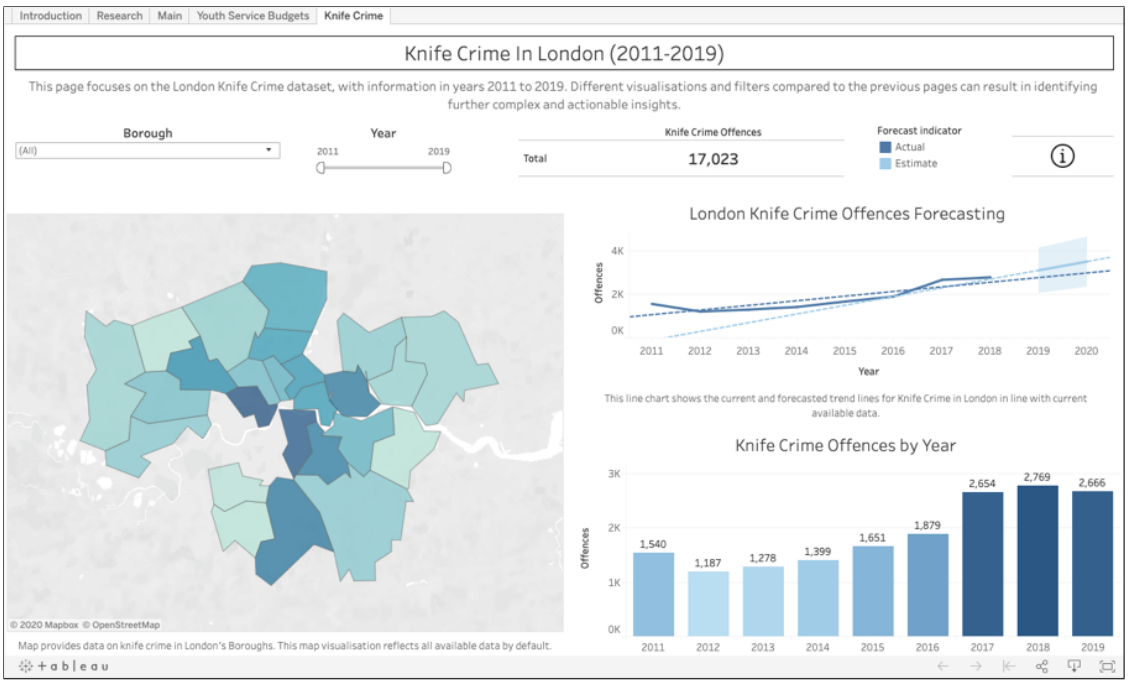

Figure 9.11: Dashboard Knife Crime Page

The third page of the dashboard is shown in **Figure 9.11**. This provides a detailed overview of Knife Crime in London. This page allows a user filter data by borough and year. Totals are also included. Furthermore, this page has a map visualisation, a line graph and a bar graph. The map shows knife crime intensity in London's boroughs. The line graph shows knife crime offences from the previous nine years.

The bar graph shows the knife crime offences from the previous nine years in an interactive bar graph, displaying specific totals for the user.

Tableau also includes the feature of enabling forecasting on connected data. **Figure 9.11** shows a forecasting of future knife crimes in London for 2020. In order to deliver this prediction, Tableau uses the 'Simple Exponential Smoothing' Equation (Tableau, 2020); which can be seen below (NIST, 2020).

$$S_{t+1} = \alpha\gamma_t + (1-\alpha)S_t, \quad 0 < \alpha \leq 1, \quad t > 0$$

This Simple Exponential Smoothing equation is used to make short term forecasts of data which uses a weighted moving average, which is typically used within the Finance and Economics sector (Andale, 2018). As a result of its popular use for forecasting unclear patterns, this project finds that the use of this equation is suitable for forecasting potential future values in the dashboard.

9.2.2 Accessibility Page

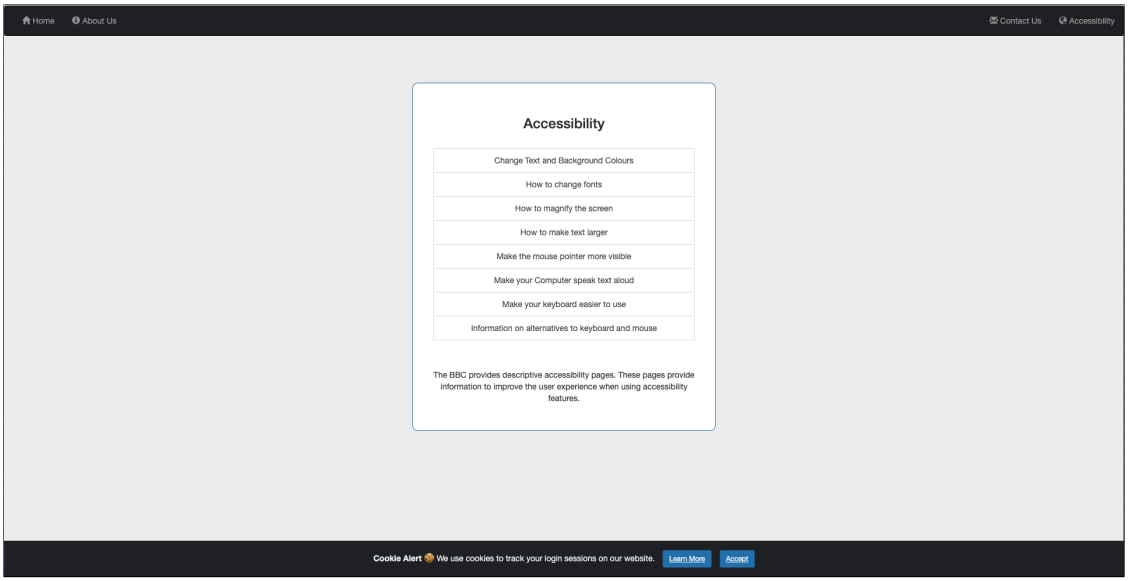

Figure 9.12: Accessibility Page Screenshot

An Accessibility Page was elicited as a requirement after research into Legal, Social, Ethical and Professional Issues (see Chapter 5) of this report. This hopes to provide information so that all possible users can interact with the system comfortably. In order to produce this page, the developer referred to the WCAG 2.1 Accessibility Guidelines document to identify common accessibility options. Various articles from the BBC have been included to guide users for information to improve their experience.

9.2.3 Forgotten Password Page

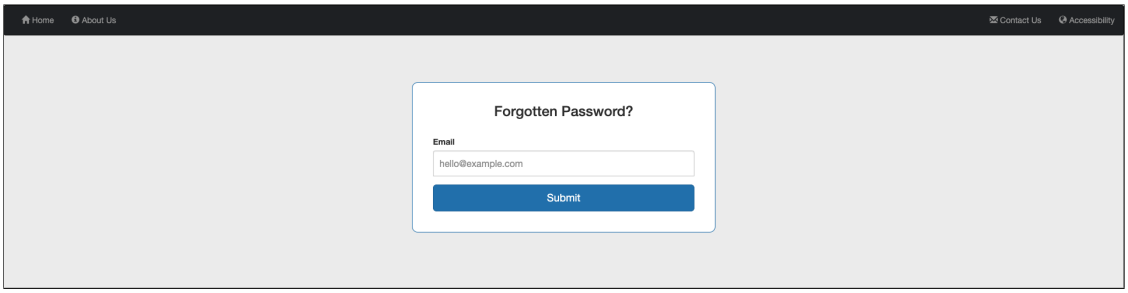

Figure 9.13: Forgotten Password Page Screenshot

The Forgotten Password Page simply asks a user to enter their email address to begin the forgotten password process. After a user has entered their e-mail address, they will receive an email with a password reset link. This link follows the same query parameter functionality of the registration process, where, backend code will handle database operations upon receiving this link (such as linking a user accounts to the query parameter of the link and resetting their

password). Once the query has been verified against a user, the system will reset the user's password to a random string, and email this to the user. Examples of this and other e-mails sent to a user can be seen in **Appendix I**.

9.2.4 Contact Us Page

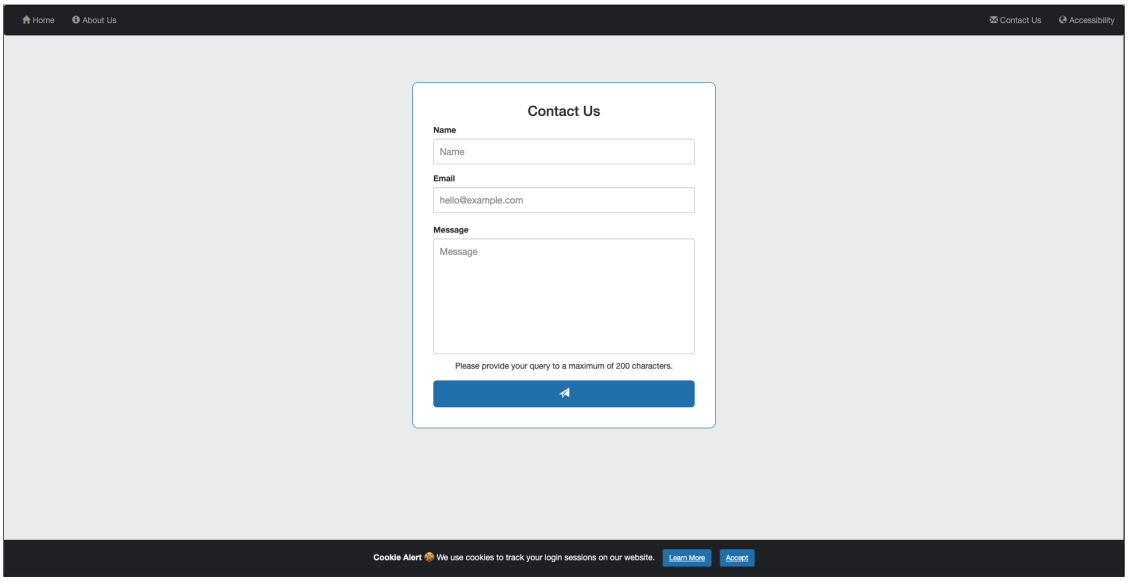

Figure 9.14: Contact Us Page Screenshot

This page provides a form that a user can fill in, if they want to contact the web app administrator. The administrator will be able to answer any queries that a user has and is the direct contact in this web app in terms of support.

Listing 9.5: Contact Us Page Form Code

```
app.post("/contactus", function (req, res, done) {

  // Set variables from req.body
  var formName = req.body.name;
  var formEmail = req.body.email;
  var formMessage = req.body.message;

  // Get the current date and time.
  var today = new Date();
  var date = today.getDate()+'-'+(today.getMonth()+1)+'-'+today.getFullYear();
  var time = today.getHours() + ":" + today.getMinutes() + ":" +
      today.getSeconds();
  var dateTime = date+' '+time;

  // Contact Us Form Mail Options
  mailOptions = {
    from: '"Dashboard Application" <leepjwallace@gmail.com>',
```

```
    to: 'leepjwallace@gmail.com',
    replyTo: formEmail, // When the administrator replies to the email, they
        will reply directly to the email address provided in the form.
    subject: "Contact Us Form Submitted",
    html: "<center><h3><u>Contact Us Form</u></h3><h4>Name: </h4>" + formName +
        " <br><h4>Email: </h4>" + formEmail + "<br><h4>Message:</h4> '" +
        formMessage + "' <h4>Sent at:</h4> " + dateTime + "</center>"
  }

  // Handle smtp Transport Error Handling
  smtpTransport.sendMail(mailOptions, function (error, response) {
    if (error) {
      console.log("smtpTransport ERROR: " + error);
      req.flash('message', 'smtpTransport ERROR.')
      return res.redirect("/signin");
    } else {
      console.log("Message sent: " + response.message);
      req.flash('message', 'Contact Form Submitted! Please expect a response
          within 24 hours.')
      return res.redirect("/signin");
    }
  })
});
```

Figure 9.14 displays the contact us page user interface for the final prototype. **Listing 9.5** shows the backend code for handling the contact form.

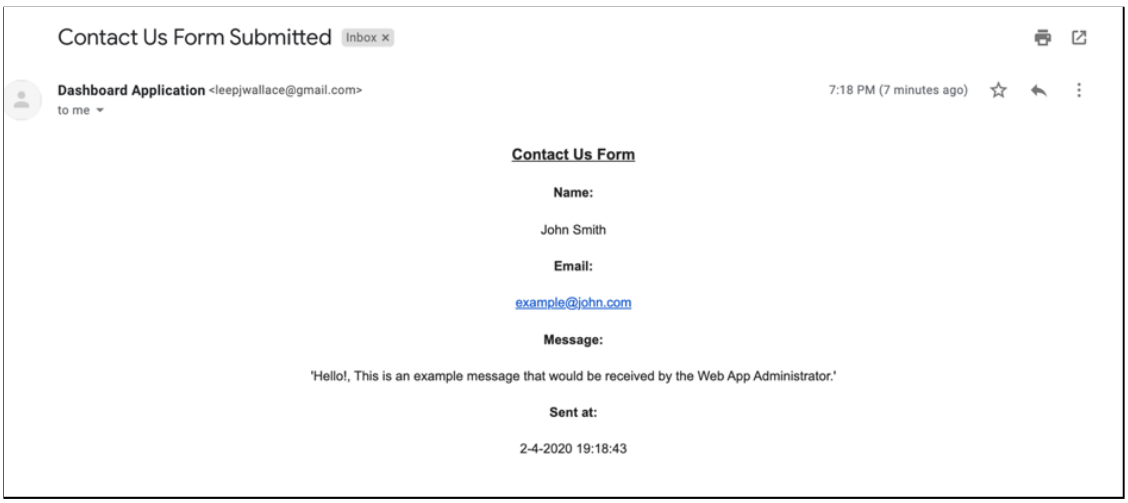

Figure 9.15: Example e-mail received by an Administrator

An example of what the Web App Administrator will see when a user has submitted the form on the Contact Us Page can be seen in **Figure 9.15**. Examples of e-mails that users may receive from the system can be seen in **Appendix I**.

9.3 Timebox Three

9.3.1 Administrator Page

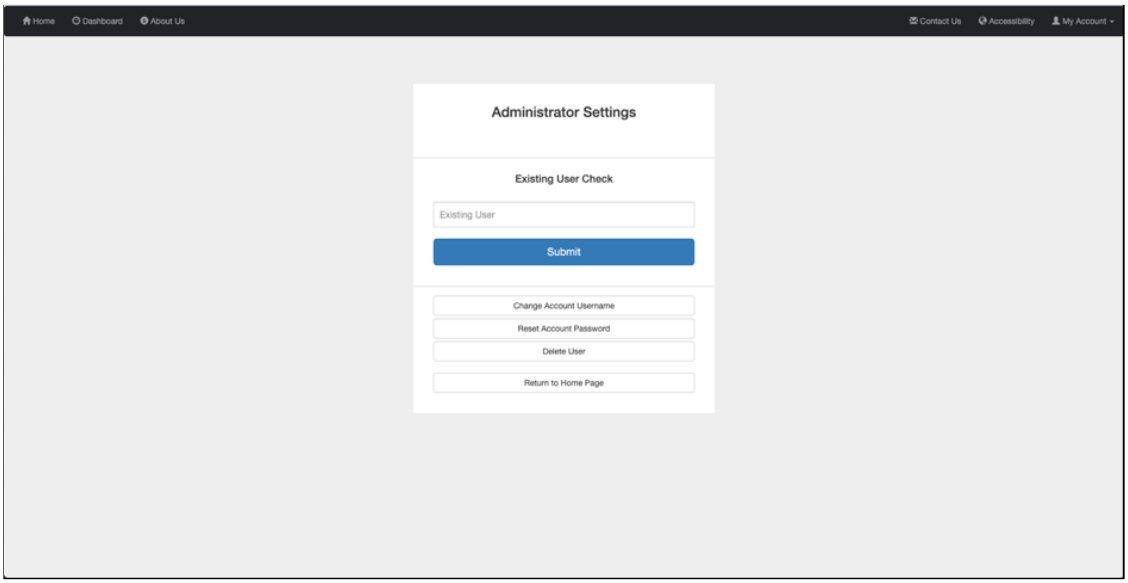

Figure 9.16: Administrator Settings Page Screenshot

The Administrator page is restricted to users who have the Boolean value **True** set to the field **isAdmin** in the MySQL database. If this value is not set to true for a user, they will not be presented with an option to access this page.

From this page, administrators can change various properties of other users accounts. For example, they have the ability from this page to reset a users password or delete their account entirely from the database. This page allows the system to be self-contained, without any need to login directly to the database to perform these administrative tasks.

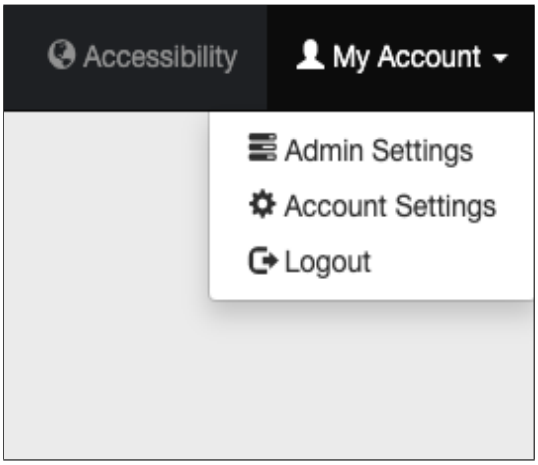

Figure 9.17: Administrator Settings Page Screenshot

Figure 9.17 shows that a user with administrator privileges is presented with the option to access the Administrator Settings Page. **Figure 9.6** shows the typical dropdown menu for a user, where they are not able to access the administrator settings page.

9.3.2 About Us Page

The About Us Page (see **Figure 9.18**) is designed to give users a quick overview of the web application, what it is designed to do and who its target market is.

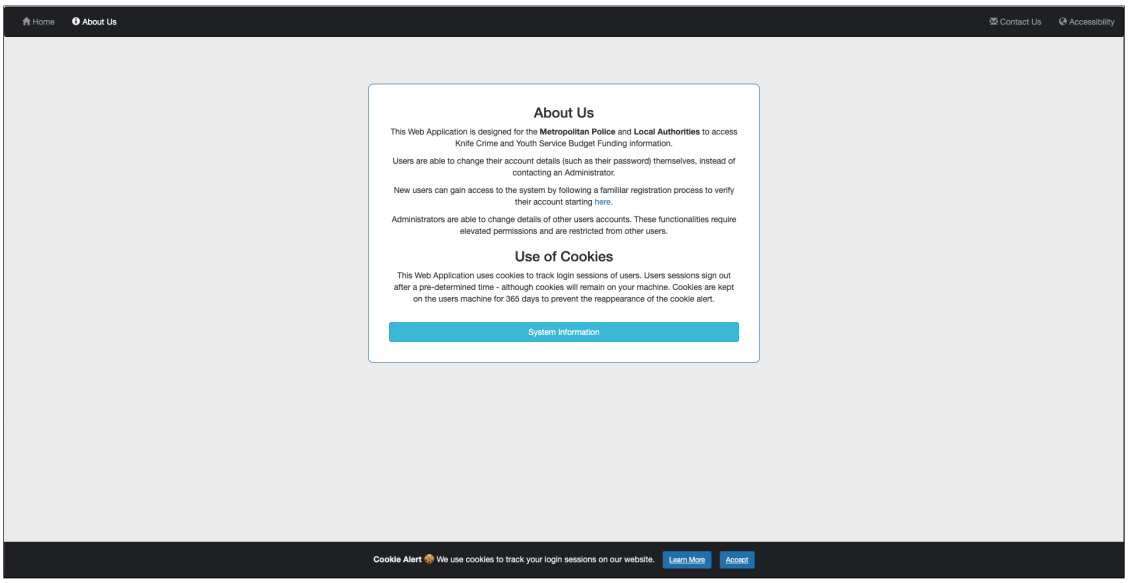

Figure 9.18: About Us Page Screenshot

10

Testing

Testing was performed consistently over the course of development. This helped to ensure that all functions and response were working correctly, including the catching and handling of errors. Consistently testing during development resulted in a final web application prototype that had form validation, sanitisation, escaping and error handling. Consistent testing during the development of the dashboard resulted in preventing logical errors and ensuring that a new unfamiliar user would understand the system and be able to use it instantly.

This project performed extensive testing for the system to ensure that all functionalities and possibilities were performing as expected for users.

10.1 White-Box Testing

White-Box Testing is used to allow a developer to identify issues within the system – with full access to internal code structures and implementations. Inputs and functionalities of the system can be tested with varying input data to test responses from the system. This is crucial in deploying a well-built system that catches errors and response accordingly. The results of the White-Box Testing can be found in **Appendix G**.

10.2 Usability Testing

Usability Testing is a technique used to identify if all potential users will be able to operate all functionalities of the system easily.

In Chapter 4, this report conducted usability testing using Neilson's Usability Heuristics to score similar existing products. The final prototype developed will be scored against the same criterion to provide a consistent approach to usability testing in this project. In addition to these heuristics, usability testing will also include aspects of research found during the literature review to identify how these were included in the user interface. The results of the usability testing can be found in **Appendix H**.

11

Critical Evaluation

Project and Personal Evaluations will provide an opportunity to review the success of the project based on objectives stated earlier in this report (see Chapter 1). Providing a reflective approach to the project will allow the developer to identify critical issues and how they were overcome.

11.1 Product Evaluation

This project used the agile framework DSDM to successfully deliver a product that has progressed through multiple prototypes based on user feedback. A Gantt Chart (see Appendix B) was introduced in the Project Proposal, which was used throughout the project to organise the development of the prototype. This organisation included ensuring that development objectives were met at specific points in time in order to meet the project deadline.

Iterations of the product were developed consistently throughout the project, with changes being implemented to improve the user interface and functionality. Low-level prototypes were designed and can be seen in Chapter 7.3 and **Appendix K**. However, through prototyping and iterative development, the final prototype has significant changes in design, as feedback was used to improve the designs, in line with DSDM.

This project recognises that data was sometimes incomplete (such as youth service funding data for various London Boroughs, discussed further in **Appendix J**). As a result, incomplete or poorly formatted values were omitted from the data used in the visualisations to provide the most accurate information as possible, the downside of this process was that data was not able to be accurately supplied for all of London's Boroughs.

In terms of the Web Application, it utilises the templating engine "PUG". This templating engine is used alongside the ExpressJS framework to produce HTML pages written in PUG syntax. This syntax is less complicated than HTML, which compiles in full HTML at runtime. Using the PUG syntax to render pages instead of HTML saved a lot of time during development in prototyping layouts and designs, an example of PUG syntax compared to HTML can be seen in **Appendix D.6**. However, because uses whitespace to seperate and end statements (similar to Python syntax), this caused issues when trying to create complex layouts with multiple elements on the same page. As a result, the developer would choose to use an entirely different front-end delivery architecture to improve productivity.

The dashboard layout was based on similar products discussed earlier in this report (see Chapter 4) and findings from the Literature Review (see Chapter 3). A report from Datalabs Agency (2014) which found that line graph and bar chart visualisations excel in showing data that change over time (see **Figure 3.4**). Chou (2019) found that linking bar, line and geographical charts together allows for further emphasis, which was crucial during this project. The Visualisation Dashboard begins with an overview page with information about how to use the dashboard, followed by a research page – which summarises some of the findings in this

report. Additionally, the visualisations are displayed in a consistent format which ensures that navigation is easy for the user.

The visualisation dashboard brings all of the data sets together to show correlations between youth service funding and knife crime, specific to London's boroughs. On reflection, this project understands that restricting access to the dashboard for local authorities could be counter-intuitive, as the data is publically available and may be beneficial to other domains such as purchasing a property, renting a property or the travel industry. If this prototype was open access, it could be beneficial to these domains as it provides a digestable summary of a boroughs history of knife crime and youth service budget funding from local councils and Government, that could be applicable to other domains. Collating all of this information together could impact an individual's decision to purchase, rent or travel to an area of London, for example.

In conclusion, this prototype was able to deliver all of the requirements elicited in Chapter 6 to produce a system that can aid users in identifying locations that could benefit from increased youth service funding, as well as discussing other domains where the prototype could be beneficial.

11.1.1 Future Developments

Further Developments could be made to improve the existing functionality of the final prototype. These developments include;

- Currently, the dashboard only uses Youth Service Budget data from two specific years, 2011 and 2019. Having a data source that provides year on year Youth Service Budget data would be beneficial to the system, as it would allow the platform to provide clearer insights. Users could benefit from detailed insights and compare this against year on year knife crime data to accurate identify where knife crime started to rise in correlation with youth service budgets.

- Currently, the prototype is designed to be restricted to the use of Local Authorities. On reflection, this dashboard could provide a wealth of information to other domains such as the housing and travel market. Opening this dashboard for public use by removing the login process may result in this dashboard being beneficial to other various sectors.

- Adding an additional data set that includes information regarding the closure of youth centres, which are the main beneficiaries of Youth Service Funding, as discussed in the Literature Review in this report (see Chapter 3); could further highlight local communities that have seen a dramatic decline in youth services. Locating where these have been shut down in the past decade could shed some light for greater analysis.

11.2 Personal Evaluation

In terms of my personal experience in this project, I feel that discovering potential causes behind the year on year rise in knife crime in falling youth service cuts was both rewarding and concerning. Rewarding in the sense that a direction for positive change can be made and concerning in the sense that youth service funding has seen such as a dramatic decline in the past ten years, which clearly has an impact in youth safety in London.

This project aimed to develop a tool for authorities to use to quickly gain insights into this problem based on available datasets. I chose the visualisation software Tableau, as I had previous experience with this, and it was preferred over other options during the technology review. This software is excellent at producing complex and interactive visualisations which can be embedded into webpages. However, I found that this software was volatile in terms of performance and found that it was difficult to maintain, as sometimes my changes would not be saved. This made the development of the dashboard difficult and troublesome. I would perhaps consider another option in the future when developing visualisations dashboards.

I decided to use a modern web application framework in NodeJS to host the dashboard. However, I was relatively new to this technology and had only developed standalone applications using Java and statically rendered webpages beforehand. This meant that I had some knowledge of standard programming paradigms but limited understanding of web development, especially using modern frameworks. However, I felt that I was able to rapidly adapt to producing software in a systematic and well-documented way. I achieved this by logging every source code change using versioning-control software, which was used to automatically deploy changes I made to the source code to a Heroku Dyno.

I feel that I have gained a lot of skills in Tableau, web development and the ability to effectively deliver a complex web application in a structured manner, using sophisticated development tools, services and techniques. Also, I have maintained a constant interest in improving the system since I started development and plan to develop it further in the future as the topic remains a personal interest.

12

Conclusion

This project aimed to develop a tool that authorities can use to identify similar patterns in rising knife crime and falling youth service budgets in London, which, during initial preliminary research into the subject, identified a consistent rise in knife crime since records began in in 2011. This same period has experienced a significant drop in youth service budgets across London's Boroughs with few anomalies. The objective was to create a tool where this research could be presented in a digestible format, to drive positive change and increase youth service funding to improve this problem across London.

Research from the Literature Review found various benefits in using visualisations to improve a user's insight into data, ensuring that it is easily digestible. As a result of this, the system was a visualisation dashboard developed using the popular visualisation software Tableau, hosted on a fast, modern web stack using NodeJS, ExpressJS and MySQL. These systems used in conjunction allowed me to design, prototype and develop a tool that supports a sophisticated accounts system, with multiple levels of access and a live data connection to a MySQL database.

This project also ensured that the system addressed the various LSEPI issues that were discussed earlier in this report (see Chapter 5). LSEPI issues ranged from the implementation of an accessibility page to hashing user passwords.

In providing this system, this project hopes to provide a tool that authorities can use to gain insights into both datasets. Users can discover information about boroughs regarding their knife crime statistics over the past ten years, alongside their youth service budget changes in the same period to identify any correlations. Also, filters have been included within the dashboard so users can select specific portions of data to review, such as the information about particular boroughs. As the dashboard is interactive, similar, related data is filtered on the page to provide the user with detailed insights.

Overall, the developer feels that they have produced a system that follows the requirements elicited earlier in this report. They hope that it serves as a tool for authorities and other relevant domains for gaining insights into the correlation between Knife Crime and Youth Service Funding in London.

13

References

A Public Health Approach to Serious Youth Violence: Supporting Evidence. (2018). [ebook] London: GLA Intelligence. Available at: https://airdrive-secure.s3-eu-west-1.amazonaws.com/london/dataset/a-public-health-approach-to-serious-youth-violence/2019-10-10T15 [Accessed 14 Oct. 2019].

agilekrc. 2020. What Is DSDM And The 8 Principles. [online] Available at: https://agilekrc.com/resource/168/what-dsdm-and-8-principles [Accessed 14 May 2020].

AltexSoft. (2019). The Good and the Bad of Node.js Web App Development. [online] Available at: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/ [Accessed 13 Feb. 2020].

AltexSoft. 2020. Functional And Nonfunctional Requirements: Specification And Types. [online] Available at: <https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/> [Accessed 5 April 2020].

Andale, S., 2018. Expontential Smoothing. [online] Statistics How To. Available at: https://www.statisticshowto.com/exponential-smoothing/ [Accessed 14 May 2020].

Arsenault, C. (2017). The Pros and Cons of 8 Popular Databases - KeyCDN. [online] KeyCDN. Available at: https://www.keycdn.com/blog/popular-databases [Accessed 13 Feb. 2020].

Azzam, T., Evergreen, S., Germuth, A. and Kistler, S., 2013. Data Visualization and Evaluation. New Directions for Evaluation, 2013(139), pp.7-32.

Babich, N. (2020). Prototyping 101: The Difference between Low-Fidelity and High-Fidelity Prototypes and When to Use Each | Adobe Blog. [online] Adobe Blog. Available at: https://theblog.adobe.com/prototyping-difference-low-fidelity-high-fidelity-prototypes-use/ [Accessed 13 Feb. 2020].

Babuta, A. (2017). Big Data and Policing. [ebook] London: Stephen Austin and Sons. Available at: https://rusi.org/sites/default/files/201709_rusi_big_data_and_policing_babuta_web.pdf [Accessed 15 Oct. 2019].

Barnardos. (2019). New Research Draws Link Between Youth Service Cuts And Rising Knife Crime. [online] Available at: https://www.barnardos.org.uk/news/new-research-draws-link-between-youth-service-cuts-and-rising-knife-crime [Accessed 13 May 2020].

BBC News. (2019). 'Poverty link' to youth violence - London mayor. [online] Available at: https://www.bbc.co.uk/news/uk-48982989. [Accessed 15 Oct. 2019].

BBC News. (2019). Reality Check: Have youth service cuts led to more crime?. [online] Available at: https://www.bbc.co.uk/news/education-43668167 [Accessed 13 Oct. 2019].

Bentham, M. (2019). Children and teenagers behind half of London knife crime. [online] Evening Standard. Available at: https://www.standard.co.uk/news/crime/half-of-london-knife-crime-carried-out-by-teenagers-and-children-as-young-as-ten-police-figures-a4056596.html [Accessed 13 Dec. 2019].

Bentham, M. (2019). Children and teenagers behind half of London knife crime. [online] Evening Standard. Available at: https://www.standard.co.uk/news/crime/half-of-london-knife-crime-carried-out-by-teenagers-and-children-as-young-as-ten-police-figures-a4056596.html. [Accessed 16 Oct. 2019].

Berry MP, S. (2019). London's lost youth services 2019. [online] London.gov.uk. Available at: https://www.london.gov.uk/sites/default/files/sian_berry_am_london_youth_services_2019.pdf [Accessed 13 Dec. 2019].

Booth, R. (2019). Do youth clubs really deter violence? I've seen for myself that they do | Robert Booth. [online] The Guardian. Available at: https://www.theguardian.com/commentisfree/2019/mar/11/violence-gangs-youth-clubs. [Accessed 15 Oct. 2019].

Booth, R. (2019). Youth club closures put young people at risk of violence, warn MPs. [online] The Guardian. Available at: https://www.theguardian.com/society/2019/may/07/youth-club-closures-young-people-risk-violence-mps [Accessed 13 Dec. 2019].

Brown, P. (2013). Why do you need to Salt and Hash passwords?. [online] Culttt. Available at: https://culttt.com/2013/01/21/why-do-you-need-to-salt-and-hash-passwords/ [Accessed 25 Nov. 2019].

Bulman, M., 2018. Police Failing To Solve Two-Thirds Of Knife Crime As Figures Soar. [online] The Independent. Available at: https://www.independent.co.uk/news/uk/home-news/knife-crime-young-people-unsolved-stabbings-soar-police-figures-a8410491.html [Accessed 14 May 2020].

Byc.org.uk. (2019). National Youth Agency. [online] Available at: https://www.byc.org.uk/wp-content/uploads/2019/06/BYC018-National-Youth-Agency.pdf [Accessed 1 Dec. 2019].

Carnes, B. (2020). Learn Node.js and start executing JavaScript outside of the browser. [online] freeCodeCamp.org. Available at: https://www.freecodecamp.org/news/getting-started-with-node-js/ [Accessed 13 Feb. 2020].

Chand, M. (2019). Most Popular Databases In The World. [online] C-sharpcorner.com. Available at: https://www.c-sharpcorner.com/article/what-is-the-most-popular-database-in-the-world/ [Accessed 13 Feb. 2020].

Chartio. (2019). A Complete Guide to Scatter Plots. [online] Available at: https://chartio.com/learn/charts/what-is-a-scatter-plot/ [Accessed 13 Dec. 2019].

Chen, H., Violette, C., Atabakhsh, H., Tseng, C., Marshall, B., Kaza, S., Eggers, S., Gowda, H., Shah, A. and Petersen, T. (2005). Visualization in law enforcement. CHI '05 extended abstracts on Human factors in computing systems - CHI '05. Chou, L. (2019). Top 10 Map Types in Data Visualization. [online] Medium. Available at: https://towardsdatascience.com/top-10-map-types-in-data-visualization-b3a80898ea70 [Accessed 13 Dec. 2019].

codefirst. 2020. The Difference Between Black Box And White Box Testing. [online] Available at: https://www.codefirst.co.uk/blog/difference-black-white-box-testing/ [Accessed 14 May 2020].

Code of Conduct for BCS Members. (2019). [ebook] BCS, The Chartered Institute for IT. Available at: https://www.bcs.org/upload/pdf/conduct.pdf [Accessed 25 Nov. 2019].

Cookiebot.com. 2020. Cookie Policy For GDPR. [online] Available at: https://www.cookiebot.com/en/cookie-policy/ [Accessed 17 April 2020].

Cps.gov.uk. (2019). Youth crime | The Crown Prosecution Service. [online] Available at: https://www.cps.gov.uk/youth-crime [Accessed 25 Nov. 2019].

Data.gov.uk. (2019). Find open data - data.gov.uk. [online] Available at: https://data.gov.uk/. [Accessed 11 Jan. 2020].

Datalabs Agency. (2014). 15 Most Common Types of Data Visualisation — Datalabs. [online] Available at: https://www.datalabsagency.com/2014/12/22/15-most-common-types-of-data-visualisation/ [Accessed 13 Dec. 2019].

Davidson, R. (2019). Crime and environment. London: Routledge. [Accessed 15 Dec. 2019].

Db-engines.com. (2020). DB-Engines - Knowledge Base of Relational and NoSQL Database Management Systems. [online] Available at: https://db-engines.com/en/ [Accessed 13 Feb. 2020].

Dearden, L. (2019). Knife crime hits highest level since records began. [online] The Independent. Available at: https://www.independent.co.uk/news/uk/crime/knife-crime-stabbings-london-gang-police-england-wales-uk-a8885486.html [Accessed 17 Nov. 2019].

Dearden, L. (2019). Knife crime rise 'linked to youth service cuts', parliamentary report finds. [online] The Independent. Available at: https://www.independent.co.uk/news/uk/home-news/knife-crime-uk-stabbings-youth-service-cuts-government-austerity-a8901856.html [Accessed 25 Nov. 2019].

Dsdm.org, (2019). [online] Available at: http://www.dsdm.org/content/4-atern-principles. Express-validator.github.io. 2020. Sanitization · Express-Validator. [online] Available at: https://express-validator.github.io/docs/sanitization.html [Accessed 2 April 2020].

Expressjs.com. (2020). Express application generator. [online] Available at: https://expressjs.com/en/starter/generator.html [Accessed 10 Feb. 2020].

Getbootstrap.com. 2020. Bootstrap. [online] Available at: https://getbootstrap.com/ [Accessed 14 April 2020].

Goel, A. (2020). 10 Best Web Development Frameworks [Updated]. [online] Hackr.io. Available at: https://hackr.io/blog/top-10-web-development-frameworks-in-2020 [Accessed 13 Feb. 2020].

Gour, R. (2019). Top 5 BI Tools that You must use for Data Visualization. [online] Medium. Available at: https://towardsdatascience.com/top-5-bi-tools-that-you-must-use-for-data-visualization-7ccc2a852bd3 [Accessed 13 Dec. 2019].

Greater London Authority, 2018. Progressing A Public Health Approach To Violence Prevention And Reduction, Appendix A: Proposed Public Health Approach. [online] London: Greater London Authority. Available at: <https://data.london.gov.uk/dataset/a-public-health-approach-to-serious-youth-violence> [Accessed 14 May 2020].

Greenberg, A., 2016. What Is Password Hashing?. [online] WIRED. Available at: https://www.wired.com/2016/06/hacker-lexicon-password-hashing/ [Accessed 14 May 2020].

Guru99.com. (2020). What is Tableau? Uses and Applications. [online] Available at: https://www.guru99.com/what-is-tableau.html [Accessed 13 Feb. 2020].

Guru99.com. 2020. What Is BLACK Box Testing?. [online] Available at: https://www.guru99.com/black-box-testing.html [Accessed 11 April 2020].

Guthe, N. (2019). [online] Quora. Available at: https://www.quora.com/What-are-the-advantages-of-Oracle-databases [Accessed 13 Feb. 2020].

Hasan, A. (2017). 10 Best Web Development Frameworks [Updated]. [online] Hackr.io. Available at: https://hackr.io/blog/top-10-web-development-frameworks-in-2020 [Accessed 13 Feb. 2020].

Ico.org.uk. (2019). The principles. [online] ICO. Available at: https://ico.org.uk/for-organisations/
guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/principles/
[Accessed 25 Nov. 2019].

Intellipaat Blog. 2014. What Is Tableau - Uses And Applications. [online] Intellipaat. Available at: https://intellipaat.com/blog/what-is-tableau/ [Accessed 15 May 2020].

Itl.nist.gov. 2020. 6.4.3.2. Forecasting With Single Exponential Smoothing. [online] NIST. Available at: https://www.itl.nist.gov/div898/handbook/pmc/section4/pmc432.htm [Accessed 14 May 2020].

Jung, J., 2019. What Are Salted Passwords And Password Hashing?. [online] Okta. Available at: https://www.okta.com/blog/2019/03/what-are-salted-passwords-and-password-hashing/ [Accessed 14 May 2020].

kids.org. (2019). HYPP Youth Clubs. [online] KIDS.
Available at: https://www.kids.org.uk/hypp-youth-clubs [Accessed 13 Dec. 2019].

King, T. (2019). What Benefits do Kids get from Youth Groups?. [online] Youthgroupgames. Available at:
http://www.youthgroupgames.co.uk/what-benefits-do-kids-get-from-youth-groups.html
[Accessed 13 Dec. 2019].

le.ac.uk. (2019). Bar charts — University of Leicester. [online] Available at:
https://www2.le.ac.uk/offices/ld/resources/numerical-data/bar-charts [Accessed 13 Dec. 2019].

London City Hall. (2019). Revealed: full links between poverty and violent crime in London. [online] Available at: https://www.london.gov.uk/press-releases/mayoral/full-links-between-poverty-and-violent-crime [Accessed 13 Dec. 2019].

Lucidchart.com. (2020). ER Diagram (ERD) - Definition. [online] Available at:
https://www.lucidchart.com/pages/er-diagrams [Accessed 13 Feb. 2020].

Macdonald, R. (2015). Pros and Cons of Ruby on Rails. [online] Madetech.com. Available at: https://www.madetech.com/blog/pros-and-cons-of-ruby-on-rails [Accessed 13 Feb. 2020].

Mackinlay, J.D., Hanrahan, P. & Stolte, C., 2007. Show me: Automatic presentation for visual analysis. Visualization and Computer Graphics, IEEE Transactions on, 13(6), pp.1137–1144.
Mcleod, S. (2020). Likert Scale Definition, Examples and Analysis | Simply Psychology. [online] Simplypsychology.org. Available at: https://www.simplypsychology.org/likert-scale.html [Accessed 2 Feb. 2020].

Medium. (2017). What are the Benefits of Using Ruby on Rails for Your Startup. [online] Available at: https://medium.com/@RubyGarage/what-are-the-benefits-of-using-ruby-on-rails-for-your-startup-f3d6f6b9938d [Accessed 13 Feb. 2020].

Mills, S. (2014). Informal Education. Basingstoke, Hampshire: Palgrave Macmillan. Nationalarchives.gov.uk. (2019). Open Government Licence. [online] Available at: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/ [Accessed 11 Oct. 2019].

MongoDB. 2020. The Most Popular Database For Modern Apps. [online] Available at: https://www.mongodb.com/ [Accessed 22 April 2020].

Neilson, J. (1994). Heuristic Evaluation: How-To: Article by Jakob Nielsen. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/ [Accessed 25 Nov. 2019].

Neilson, J. (2020). 10 Heuristics for User Interface Design: Article by Jakob Nielsen. [online] Nielsen Norman Group. Available at: https://www.nngroup.com/articles/ten-usability-heuristics/ [Accessed 13 Feb. 2020].

nidirect. (2019). Knife crime. [online] Available at: https://www.nidirect.gov.uk/articles/knife-crime [Accessed 13 Dec. 2019].

npm. 2020. Pug. [online] Available at: <https://www.npmjs.com/package/pug> [Accessed 13 April 2020].

Reynolds, C. (2019). Chorus Intelligence: Data Visualisation Tool is Helping Catch Criminals. [online] Computer Business Review. Available at: https://www.cbronline.com/feature/chorus-intelligence. Robertson, S. (2019). A warm, safe place: an argument for youth clubs | infed.org. [online] Infed.org. Available at: http://infed.org/mobi/a-warm-safe-place-an-argument-for-youth-clubs/ [Accessed 13 Dec. 2019].

Rouse, M. (2012). What is data visualization? - Definition from WhatIs.com. [online] SearchBusinessAnalytics. Available at: http://searchbusinessanalytics.techtarget.com/definition/data-visualization [Accessed 13 Dec. 2019].

Rouse, M., 2020. What Is WYSIWYG (What You See Is What You Get)? - Definition From Whatis.Com. [online] WhatIs.com. Available at: <https://whatis.techtarget.com/definition/WYSIWYG-what-you-see-is-what-you-get> [Accessed 16 April 2020].

Sas.com. (2019). Data Visualization Techniques. [online] Available at: https://www.sas.com/en_us/whitepapers/data-visualization-techniques-106006.html [Accessed 21 Oct. 2019].

Schraer, R. (2019). Have police numbers dropped?. [online] BBC News. Available at: https://www.bbc.co.uk/news/uk-47225797 [Accessed 19 Oct. 2019].

Scopatz, A. and Huff, K., 2015. Effective Computation In Physics. O'Reilly. [Accessed 16 Oct. 2019].

scruminc. 2020. What Is Timeboxing?. [online] Available at: https://www.scruminc.com/what-is-timeboxing/ [Accessed 14 May 2020].

Sharma, L., 2016. Waterfall Model In Software Developement Life Cycle | SDLC. [online] TOOLSQA. Available at: https://www.toolsqa.com/software-testing/waterfall-model/ [Accessed 14 May 2020].

Shaw, D. (2019). The key facts about rising knife crime. [online] BBC News. Available at: https://www.bbc.co.uk/news/uk-42749089 [Accessed 13 Dec. 2019].

Smith, T., Noble, S., Annell, D. and Lally, G. (2019). DataViz: improving data visualisation for the public sector. [online] Assets.publishing.service.gov.uk. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/7611/1318351.pdf [Accessed 13 Dec. 2019].

SolutionsIQ. 2020. Dynamic Systems Development Method (DSDM). [online] Available at: https://www.solutionsiq.com/agile-glossary/dynamic-systems-development-method-dsdm/ [Accessed 14 May 2020].

SpyCloud. 2019. Spycloud. [online] Available at: https://spycloud.com/how-long-would-it-take-to-crack-your-password/ [Accessed 14 May 2020].

Stevens, M., Bursztein, E., Karpman, P., Albertini, A. and Markov, Y., 2020. The First Collision For Full SHA-1. [online] Amsterdam. Available at: https://shattered.io/static/shattered.pdf [Accessed 14 May 2020].

Tableau.com. 2020. How Forecasting Works In Tableau. [online] Available at: https://help.tableau.com/current/pro/desktop/en-us/forecast_how_it_works.htm [Accessed 14 May 2020].

TAYLOR, R. (1995). The Impact of Crime on Communities. The ANNALS of the American Academy of Political and Social Science, 539(1). [Accessed 21 Nov. 2019].

The Youth Violence Commission, 2020. The Youth Violence Commission. [online] Warwick: YVC. Available at: http://yvcommission.com/wp-content/uploads/2018/07/Interim-Report-FINAL-version-2.pdf [Accessed 14 May 2020].

Trust for London. (2019). Poverty across London. [online] Available at: https://www.trustforlondon.org.uk/data/poverty-across-london/.

Tutorialsteacher.com. 2020. MVC Architecture. [online] Available at: <https://www.tutorialsteacher.com/mvc/mvc-architecture> [Accessed 16 April 2020].

Uml-diagrams.org. (2020). Use case diagrams are UML diagrams describing units of useful functionality (use cases) performed by a system in collaboration with external users (actors).. [online] Available at: https://www.uml-diagrams.org/use-case-diagrams.html [Accessed 13 Feb. 2020].

William, J., 2020. Nodejs Passport Login Script With Mysql Database. [online] Programmerblog. Available at: http://programmerblog.net/nodejs-passport-login-mysql/ [Accessed 14 April 2020].

W3.org. (2020). Web Content Accessibility Guidelines (WCAG) 2.1. [online] Available at: https://www.w3.org/TR/WCAG21/ [Accessed 2 Feb. 2020].

W3schools.com. 2020. HTML Form Method. [online] Available at: <https://www.w3schools.com/tags/att_form_method.asp> [Accessed 16 April 2020].

Ymca.org.uk. (2018). Youth and Consequences. [online] Available at: https://www.ymca.org.uk/wp-content/uploads/2018/04/Youth-Consequences-v0.2.pdf [Accessed 13 Dec. 2019].

Youth Crime Action Plan. (2009). London: Home Office. Youth Service. (2019). [online] Available at: https://www.revolvy.com/page/Youth-service [Accessed 13 Dec. 2019].

Appendices

A

Project Proposal

A.1 Overview

Statistics reveal that violence involving a bladed weapon is rising in London, especially involving people aged under 25. A study by the Greater London Authority (GLA) Intelligence Unit found that knife crime reported to the Metropolitan Police (MET) had increased by 67% from 2013 to 2018 (A Public Health Approach to Serious Youth Violence: Supporting Evidence, 2018). Across the same period, there was a real-terms funding cut of 40% in youth services. Furthermore, research shows that a preventative approach to knife crime is essential. Youth centres provide not only a preventative approach, but also a safe environment for young people to build relationships within the community and participate in recreational activities. However, there has been a significant reduction in funding for youth services since 2013 which has resulted in over half of these centres being closed in London alone.

Across the UK, initial research has shown that the average council has reduced spending on services such as youth clubs by 40%, with some authorities reporting a 91% decline in funding for youth services. Recent research by the All-Party Parliamentary Group (APPG) has found that areas suffering the largest cuts to youth spending have seen bigger increases in youth crime (Dearden, 2019).

This report aims to identify this emerging pattern and provide a solution for authorities to use to decide where to prioritise youth service funding to tackle knife crime. Using a visualisation tool to identify crime hotspots will allow the MET to pinpoint areas that need priority in resources. A visualisation tool can be used by MPs to identify areas that need additional resources that will tackle youth crime.

This project will also design, develop and implement a comprehensive visualisation dashboard for the use by the MET which present visualisations of the data and its findings. This can be used to identify key areas in London have seen significant youth service cuts and provide funding suggestions which will aid in the reduction of knife crime.
Keywords: [youth crime, london crime, visuali?ation, “police use of visualisation”]

A.2 Aim

The aim of this project is to create a comprehensive visualisation dashboard for the MET to identify areas that would benefit from youth service funding in order to reduce knife crime.

A.3 Objectives

A.3.1 Research Report

A.3.1.1. Research into relevant datasets available regarding London's youth crime [2.0]
A.3.1.2. Research into other relevant datasets [2.0]
A.3.1.3. Analyse London's youth crime patterns [7.0]
A.3.1.4. Identify any links between my findings on London's youth crime patterns and relevant datasets [4.0]
A.3.1.5. Research into current uses of visualisations for or produced by the police which relate to crime [2.0]
A.3.1.6. Research into techniques used to prevent or reduce youth crime [3.0]
A.3.1.7. Research into appropriate Methodology's to use [1.0]

A.3.2 Research designs and implementation

A.3.2.1. Research into relevant DBMS Systems [3.0]
A.3.2.2. Research into front-facing web application frameworks [1.0]
A.3.2.3. Research into back-end systems for web applications [1.0]

A.3.3 Research Legal, Social, Ethical and Professional Issues

A.3.3.1. Research into any legal issues that may concern the project [1.0]
A.3.3.2. Research into any social issues that may concern the project [1.0]
A.3.3.3. Research into any ethical issues that may concern the project [1.0]
A.3.3.4. Research into any professional issues that may concern the project [1.0]

A.3.4 Design Documentation

A.3.4.1. Design an ERD (Entity Relationship Diagram) [6.0]
A.3.4.2. Design a Use Case Diagram [7.0]
A.3.4.3. Create a Conceptual Class Diagram [7.0]
A.3.4.4. Low Fidelity Prototypes [4.0]
A.3.4.5. High Fidelity Prototypes [4.0]
A.3.4.6. Conclusion of design process [2.0]

A.3.5 Implementation

A.3.5.1. Selecting a front end for the Web Application – Discussion [1.0]
A.3.5.2. Selecting a back end for the Web Application – Discussion [1.0]
A.3.5.3. Identifying the most appropriate method to provide visualisations [1.0]
A.3.5.4. Identifying the DBMS for the System [1.0]

A.3.6 Implementing the DBMS and Visualisations

A.3.6.1 .Implementing the DBMS using the method(s) chosen in A.3.5.4 [7.0]
A.3.6.2. Implementing a live data connection [1.0]
A.3.6.3. Implementing the visualisations using the method(s) chosen in A.3.5.3 [7.0]

A.3.7 Implementing the front-end

A.3.7.1.Creating a Front End for the web application using the method(s) chosen in A.3.5.1 [3.0]
A.3.7.2. Test for any bugs in the Front-End System [2.0]

A.3.8 Implementing the back-end

A.3.8.1. Creating a Back End for the Web Application using the method(s) chosen in A.3.5.2 [4.0]
A.3.8.2. Test for any bugs in the Back-End System [2.0]

A.3.9 Testing the System

A.3.9.1. Create a test plan to identify functionality in the system [3.0]
A.3.9.2. Test using Black + White box testing [4.0]
A.3.9.3. Peer Evaluation and Feedback [2.0]
A.3.9.4. Implement peer feedback with A/B Testing [6.0]
A.3.9.5. Evaluation and Further Research [2.0]

A.4 Project Framework or Methodology used

The project will be developed using DSDM (Dynamic Systems Development Method). This has been chosen as it is an agile software development framework based on the Rapid Application method. This methodology has been chosen because Rapid Application Development is needed for this project due to limited time constraints. To manage these time constraints, high-level requirements will be prioritised using MoSCoW – this is so essential requirements will be implemented in the first iteration, allowing more time for the remaining requirements to be developed.

In addition to using prioritization, this project will utilise timeboxing as a technique. Timeboxing will be used to allocate a fixed unit of time to be spent on every activity, following this technique will allow the project to be completed within the given time constraints. Utilizing MoSCoW prioritisation and timeboxing will mean the beginning of the project will have time spent scrutinizing which high-level requirements are the most important, and thus, what requirements should be implemented in the first iteration. Further requirements which are not categorised as 'MUST' in the MoSCoW method will be developed in a further iteration of the product.

This chosen project framework will be followed very closely throughout the course of this project in order to rapidly produce a deliverable that is high quality. Further techniques used will also be referenced and their processes described thoroughly in the project documentation. DSDM has been selected also for the reason that one of its eight principles are for the product to be developed iteratively. This will mean the product can be developed in timeboxes which will keep the project schedule on track. In addition to this, iterative development can allow for further

iterations of the final product. Further iterations can result in a more well-rounded product being developed, an example of this is if during user testing, they highlighted a function that should be changed – this function could then be added to the next iteration of the product to improve it.
These important development principles cannot be found in the Waterfall Model, for example, which is why DSDM has been selected.

A.5 Neilson's Heuristics

Neilson's Heuristics are popular usability heuristics for user interface design (Liyanage, 2016). These can be used to identify issues within a user interface, which can be improved on using iterative development, which follows the chosen DSDM methodology of iterative development. They will be used in this report for evaluating the design of products in the product research chapter. It will also be used as a guideline to ensure that the product developed in this project follows these heuristics. It is important that this product follows these heuristics to ensure a professional standard throughout.

In addition to this, research has shown that multiple evaluators is beneficial to find more issues that might not of been identified by one individual, for this reason, this project will use another person to evaluate the user interface through testing. This will improve the effectiveness of this method (Neilson, 1994).
Neilson's Heuristics are composed of eight usability principles (Liyanage, 2016), these include;

Table A.1: Neilson's Usability Heuristics

Heuristic	Description
Visibility of a system status	The System should not include dialogue that is irrelevant to the user. E.g. a minimalist error message that asks the user to try again later.
Match between the system and real world	The System should use words, phrases and concepts familiar to the user, including familiar designs. Following this Heuristic will mean the System will not use technical jargon or system-orientated words.
User control and freedom	The System should offer an 'escape' function, should they accidently use a function by mistake. E.g. Cancelling an action whilst in progress.
Consistency and standards	In order to promote efficiency and a clear design standard, the System should have quickly recognisable platform conventions. E.g. Menu bar / sign in button in the same location on every page.

Continued on next page

Table A.1 – *Continued from previous page*

Heuristic	Description
Error prevention	This Heuristic describes a 'proactive' method to errors rather than waiting for an error or problem to appear. E.g. When Google mail detected when you are trying to send an attachment, when no document has been attached. This prevents unnecessary communication in the future and saves time.
Recognition rather than recall	The System should allocate space in its interface to make objects, actions and options visible to the user. This includes making website instructions visible. E.g. Amazon remembering your search history and recommending items to you.
Flexibility and efficiency of use	The System should be designed so that novice and experienced users can use the interface efficiently. Experienced users can use 'accelerators' such as filters to enhance their experience using the interface.
Aesthetic and minimalist design	The System should not include dialogue that is irrelevant to the user. E.g. a minimalist error message that asks the user to try again later.
Help users recognise, diagnose, and recover from errors	Error messages should be clear to the user without any unnecessary jargon, although it should be able to indicate the problem and suggest a solution.
Help and documentation	A system should be able to be used without any documentation although it is always helpful to provide. Help and Documentation provided should be easy to find, digestible and focused on the users' task.

A.6 Legal, Social, Ethical and Professional Considerations

In terms of Legal Issues that concern this Project, the developed product will include functionality to login. Typical login information that is stored on a user includes the fields; Name, Username and Password.

This project will need to ensure that the seventh principle of the DPA is adhered to which states that data must be processed in a manner that ensures appropriate security of the personal data and login data (Ico.org.uk, 2019). This project will need to ensure that connected databases are secure and are suitable for the product. This can be achieved by encrypted communications between the user and the login server for example to ensure security.

Further security measures could be implemented including cryptographic password hashing in order to prevent storing plain text attributes in a database (Brown, 2013). This project will make use of third-party open source frameworks, languages and copyrighted software in order to achieve its objectives. Software that is not mine will be acknowledged, this is to ensure that it cannot be miscommunicated what software this project has developed and what has not been developed, this is to avoid any form of plagiarism. External images or scripts used on the application will be clearly highlighted in this report.

Furthermore, this project will extensively cover youth crime and therefore, "youths". The term "Youth" concerns individuals whose age is below 18 (Cps.gov.uk, n.d.). In terms of Ethical and Legal issues that concern this Project, I will need to ensure that no youths can be identified using information that is sourced from Data used in this project. This will be achieved by, where available, omitting names of youths from references used in this project and datasets, this will ensure that data used is anonymous and impartial.

In addition to this, this project will also need to ensure that all data has been retrieved with legal sources. Data from the data.gov.uk website for example will need to hold the Open Government License. (nationalarchives.gov.uk, 2019) I will ensure that the British Computer Society (BCS) code of conduct is followed at all stages through this report and the product design, this is because this report is based on a University course accredited by the BCS. This code of conduct sets the professional standards of competence, conduct and ethical practice in the United Kingdom (Code of Conduct for BCS Members, n.d.), Following this will allow myself, and the project to conduct itself professionally and ethically.

Another person will be used in the testing section of this report in order to provide feedback that can be used to improve the product. Communication with this individual will need to be professional and socially acceptable, following the BCS code of conduct will allow me to achieve this.

A.7 Planning

For this project, DSDM has been selected for the methodology. Utilizing an agile framework includes the use of certain techniques to manage time and expectations. Techniques that will be utilised throughout the course of this project include Iterative Prototyping, MoSCoW Prioritisation and the use of UML diagrams. Approaching my project using these techniques will result in being able to effectively manage the time constraints throughout the project using requirement prioritisation and timeboxing. In addition to managing time, I will also be able to produce a high-quality product which has been designed using UML diagrams and iterative prototyping.

At the beginning of the Project, datasets will be acquired using publicly available sources such as using data.gov.uk. Software will be used to analyse and cross-examine this data so that

an analytical report can be produced with the findings. This will include using data analysis to cross-examine the data, to identify a link between youth crime and data from other agencies.

From these findings, the project aims to deliver a comprehensive visualisation dashboard which will incorporate developing a web application. Research into what technologies will be most suitable for this purpose, including the requirements of the user will be carried out before any building of the application(s) to ensure that the product meets the requirements of the user.

Development will be fully documented including all code used in the project. In addition to this, the application will be thoroughly tested using a variety of techniques, these techniques includes White-Box Testing and Usability Testing. Furthermore, I will include user testing in the testing section. Including user testing in this section will allow functional errors / user requirements to be found which may not have been picked up on during the normal course of testing. A Gantt Chart which can be found in **Appendix B** to detail how this project will be scheduled. This Gantt Chart has been split into four key sections which will be timeboxed.

B

Gantt Chart

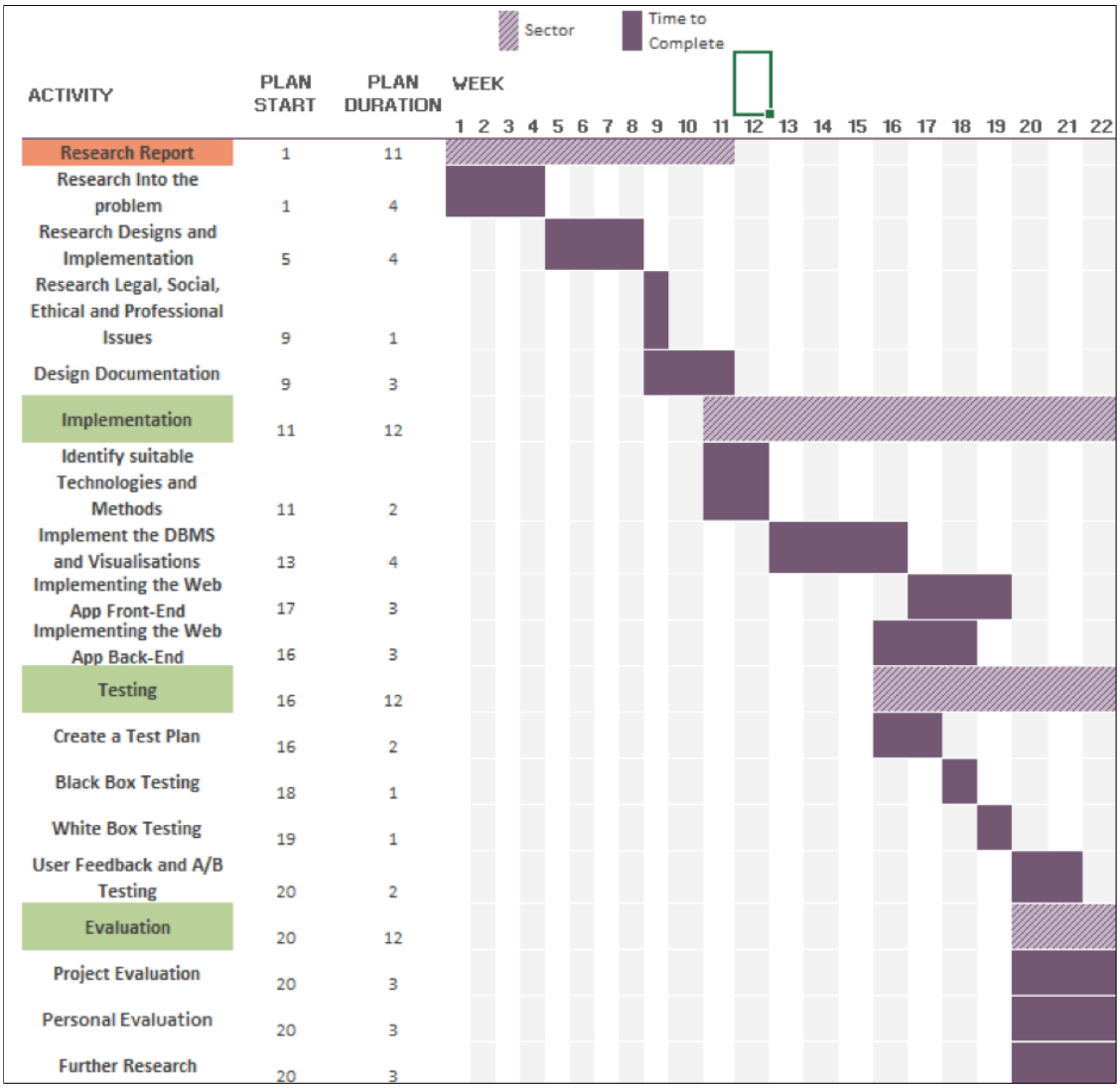

Figure B.1: Project Gantt Chart

This Gantt Chart has grouped objectives into four main groups; the Research Report will be complete in term one. The remaining sectors, Implementation, Testing and Evaluation are all due to be completed in term two.

C

Uses of visualisations

Research questions	Chart types (used to link to examples)
How do we **compare** (with other areas)?	• bar chart • choropleth map
What is **changing over time**?	• animations • flow maps • time line • line chart
What is the **distribution** (of socio-economic indicators)?	• histogram • scatterplot • bubble map • smooth statistical map • choropleth map
What are the **components** of an indicator (for example, how does the workless total split between jobseekers, incapacity benefits, lone parents etc)	• proportional area chart
What is the **relationship between** different socio-economic factors?	• scatterplot • quadrant plot • clustered bar chart • choropleth map • comparative chart
How **significant are differences**?	• confidence intervals • funnel
How can I **interactively explore** a dataset?	• Interactive
How can I visualise **qualitative (text) data**?	• qualitative
How can I visualise **categorical data** (for example, neighbourhood area classifications or service locations)	• categorical (data-type tag) • point location (data-type tag)

Figure C.1: Different uses of visualisations

D

Code Explanations

D.1 Login Code

The login system is essential in enabling users to gain access to restricted areas of the system.

The Login System is faciliated by the Passport.JS middleware and uses a technique called the 'LocalStrategy'. This technique connects to a database with provided credentials. In this instance, a MySQL Database is used for this system. The credentials for this database connection are set as environment variables, which are found in an .env file - this is to protect sensitive information being uploaded to the GitHub repository, as the .env was ignored during commits. The credentials file named 'dbconn.js' can be seen in Listing D.1.

Listing D.1: MySQL Connection

```
1  var mysql = require('mysql'); // mysql module is used as a driver.
2  require('dotenv').config()
3
4  // This block of code contains mysql database connection credentials for the
       database.
5  var connection = mysql.createConnection({
6    supportBigNumbers: true,
7    bigNumberStrings: true,
8    host     : process.env.DB_HOST,
9    user     : process.env.DB_USER,
10   password : process.env.DB_PASSWORD,
11   database : process.env.DATABASE
12 });
13
14 module.exports = connection;
```

The function in Listing D.2 shows the function that is called after a user submits their username and password. Line 9 checks if the username and password have been submitted in the login form, if they are not – this will produce a message for the user. Line 19 checks if the username entered exists, if it doesn't then an error will appear. Line 24 creates a SHA1 hash into variable 'encPassword' of the concatenated salt and password found in the updated salt variable in Line 22.

Following this, Line X will check if the sha1 hash matches the hashed value in the database, if it doesn't match, then an error will appear. Otherwise, the user will be logged in, in Line 32 with 'req.session.user'.

Listing D.2: Login Code

```
1
2  passport.use('local', new LocalStrategy({
3    usernameField: 'username',
4    passwordField: 'password',
5    passReqToCallback: true      //passback entire req to call back
6  }, function (req, username, password, done) {
7    console.log("\n *** Username Entered: " + username + " ***");
8    console.log("\n *** Password Entered: " + password + " *** \n");
9
10   if (!username || !password) { return done(null, false, req.flash('message',
         'All fields are required.')); } // If username + password fields null,
         then throw err.
11
12   var salt = process.env.SALT;
13
14   connection.query("select * from tbl_users_test where username = ?",
         [username], function (err, rows) { // Find username entered by the user.
15     console.log(err);
16     if (err) return done(req.flash('message', 'Database Error'));
17
18     if (!rows.length) { return done(null, false, req.flash('message', 'Invalid
           username or password.')); } // Identify if username exists
19
20     salt = salt + '' + password;   // Concatenate salt and password
21
22     var encPassword = crypto.createHash('sha1').update(salt).digest('hex'); //
           Create sha1 hash
23     var dbPassword = rows[0].password; // Crawl database to see if password
           exists
24     var dbIsVerified = rows[0].isVerified;
25     var stringJson = JSON.stringify(rows);
26     console.log(stringJson)
27     console.log("Database Verification BOOLEAN: "+dbIsVerified)
28
29     // Checks users hashed form password against the hashed password in the
           database.
30     if (!(dbPassword == encPassword)) {
31       console.log("\n *** Login Unsuccessful *** \n")
32       return done(null, false, req.flash('message', 'Invalid username or
             password.'));
33     }
34
35     // Checks if the 'isVerified' Attribute is set to true.
36     if (dbIsVerified == '0') {
37       console.log("\n *** User has not verified their account *** \n")
38       return done(null, false, req.flash('message', 'Account has not been
             verified'));
39     }
40     console.log("\n *** Login Successful *** \n")
41     req.session.user = rows[0];
42     return done(null, rows[0]);
43   });
44 }
```

```
45 ));
```

In order to protect routes from unauthenticated users, the function seen in Listing D.3 can be found in the router JavaScript file of every protected link.

Listing D.3: Protecting routes

```
1 function isAuthenticated(req, res, next) {
2   if (req.session.user)
3       return next();
4
5   // IF A USER ISN'T LOGGED IN, THEN REDIRECT THEM SIGNIN PAGE
6   res.redirect('/signin');
7 }
```

This function checks if a user is currently logged in, if this value is TRUE, then the system will return next(), e.g. allowing the user to access the page. Otherwise, it will redirect the user to the sign in page (see **Figure 9.1**).

D.2 Account Settings

In order to display user information on the account settings page, the function seen in Listing D.4 is used to assign variables from current session information. For example, it will request the current user's username, and assign it to a local variable of the same name. This will then render the page they are trying to access, in this case the user (account settings) page, and load the variables into the corresponding page, as local variables as seen in Line 8. The PUG template then loads these variables into the table on the account settings page (see **Figure 9.7**).

Listing D.4: Loading session variables to page

```
1 router.get('/', isAuthenticated, function(req, res, next) {
2
3   var username  = req.session.user.username;
4   var full_name = req.session.user.full_name;
5   var email = req.session.user.email;
6   var isAdmin = req.session.user.userTypeID
7
8   res.render('user', { username: username, full_name: full_name, email : email,
        isAdmin : isAdmin});
9 });
```

D.3 Registration Code

```
app.post("/register", [
  check('username', 'Please enter a valid
      username').not().isEmpty().trim().custom(async username => {
    const value = await isUsernameInUse(username);
    if (value != 0) {
        req.flash('message', 'Username count is ' + value)
    }
  }),
  check('fullname', 'Name must be over five characters
      long.').not().isEmpty().isLength({ min: 5 }).trim().escape(),
  check('email', 'Please enter a valid email
      address.').not().isEmpty().isEmail().normalizeEmail().trim().escape()
  .custom(async email => {
    const evalue = await isEmailInUse(email)
    if (evalue != 0) {
      done();
    }
  }),
  check('password', 'Password must have a minimum of 5
      characters').not().isEmpty().isLength({ min: 5 }).trim().escape(),
  check('confirmPassword', 'Passwords do not
      match').not().isEmpty().custom((value, {req}) => {
    if (value !== req.body.password) {
      return false;
    } else {
      return value;
    }
  }
  )], function (req, res) {

    // Read input from post form
    regUsername = req.body.username,
    regPassword = req.body.password,
    regFullName = req.body.fullname,
    regEmail = req.body.email,
    regUserTypeID = '0',
    regIsVerified = '0'
    regVerificationToken = ''

    // Username is appended with the "ValidationToken" String for a hashed
        token. Good as it uses a unique value everytime.
    hashedUsernameSalt = "ValidationToken" + '' + regUsername
    var regVerificationToken =
        crypto.createHash('sha1').update(hashedUsernameSalt).digest('hex');
    console.log("Verification Token for" + regUsername + "is:" +
        regVerificationToken);

    const errors = validationResult(req);
    console.log(req.body);

    if (!errors.isEmpty()) {
```

```
44        console.log('Validation Errors');
45        console.log('');
46        console.log(errors)
47        console.log('')
48
49        var valErrors = (JSON.stringify(errors)); // JSON is stringified here
50        var parsed = JSON.parse(valErrors) // Stringified JSON is parsed here
51
52        displayPassErr = (parsed.errors[0].msg) // .msg locates the msg attribute
              in the JSON String.
53        console.log(displayPassErr.includes(substring)); // Prints to console to
              see if the string 'must match' is found in the JSON Response.
54        console.log('')
55        console.log(parsed.errors[0].msg) // Test parsed JSON String
56        console.log(req.body.confirmPassword)
57
58        // Check to see if the stringified JSON response contains the substring
              variable 'subString'.
59        var substring = 'match';
60        if (displayPassErr.includes(substring)) {
61          req.flash('message', displayPassErr)
62          return res.redirect("/signin");
63        }
64
65        // Check if the stringified JSON response contains the substring variable
              'usrSubString'.
66        var usrSubString = 'req';
67        if (displayPassErr.includes(usrSubString)) {
68
69          req.flash('message', '* Username Already Exists. Please choose another.
                *')
70          return res.render('register', { message: req.flash('message') });
71        }
72
73        var emailSubString = 'done';
74        if (displayPassErr.includes(emailSubString)) {
75          req.flash('message', 'Email Address is already in use')
76          return res.redirect("/signin");
77        }
78
79        var printout = 'Name and Password must exceed five characters.'
80        return res.redirect("/signin");
81      }
82
83      var regSalt = process.env.SALT; //Salt will be kept as environment variable
            in the future.
84      regSalt = regSalt + '' + regPassword;
85      var encRegPassword =
            crypto.createHash('sha1').update(regSalt).digest('hex');
86      console.log("*** The Encoded password is: " + encRegPassword + " ***")
87
88      // this establishes a connection with the database and inserts the parsed
            data above into tbl_users with variables.
89
90      var sql = "INSERT INTO tbl_users_test (username, password, full_name,
```

```
        email, userTypeID, isVerified, verificationToken) VALUES ('" +
        regUsername + "', '" + encRegPassword + "', '" + regFullName + "', '" +
        regEmail + "', '" + regUserTypeID + "', '" + regIsVerified + "', '" +
        regVerificationToken + "')"
91      connection.query(sql, function (err, result) { //values inserted into the
        query
92        if (err) throw err;
93        // req.flash('message', 'Registration successful!' + '\n' + ' Please
            login and enter the code found on the e-mail to continue.')
94      })
95
96      //var rand, mailOptions, host, link;
97
98      //rand = Math.floor((Math.random() * 100) + 54);
99      //console.log("Random verification number: " + rand);
100     host = req.get('host');
101     link = "http://"+req.get('host')+"/verify?id="+regVerificationToken;
102     contactUsLink = "http://"+req.get('host')+"/contactus";
103     ErrEmailLink =
            "http://"+req.get('host')+"/verifydelete?id="+regVerificationToken;
104
105     mailOptions = {
106       from: '"Dashboard Application" <leepjwallace@gmail.com>',
107       to: regEmail,
108       subject: "Please verify your Dashboard Application Account",
109       html: "<center>Hello "+ regFullName +",<br><br> Please use the link below
            to verify your account.<br><br>Account Username: " + regUsername +
            "<br><br>Account Email: " + regEmail + "<br><br><a href=" + link +
            ">Click here to verify</a><br><br><br><br>If you have received this
            email in error, please <a href=" + ErrEmailLink + ">click here</a> so
            we can remove your email from our records.<br><br><a href=" +
            contactUsLink + "> Contact Us</a></center>"
110     }
111     // console.log("These are the mailOptions " + mailOptions);
112     smtpTransport.sendMail(mailOptions, function (error, response) {
113       if (error) {
114         console.log("smtpTransport ERROR: " + error);
115         req.flash('message', 'smtpTransport ERROR.')
116         return res.redirect("/signin");
117       } else {
118         console.log("Message sent: " + response.message);
119
120         req.flash('success', 'Registration Successful! Please check your email
              to verify your account.')
121         return res.render('register', { success: req.flash('success') });
122
123       }
124
125     });
```

To verify user inputs when registering for a new user, the system uses the 'express-validator' module. Various functions are used against user inputs, these can be seen in Lines 2 to 16.

The trim() command trims whitespace characters from both sides of the input, this is helpful

in preventing spaces in entries in the database. The escape() function is used to remove certain characters which could be used for cyber-attacks on the web app, such as SQL Injection through data input. This allows the Project to operate within the DPA principles. The not().empty() functions ensure that data inputs are populated and are not empty.

Additionally, the password and confirmPassword input fields are checked to see if they match when the form has been posted, this will prevent a user from inputting an unmemorable password accidently. If the values in these fields do not match, the system will alert the user of this error and will not submit the data inputted to the database as it failed the validation process.

Line 49 parses the JSON response from express-validator and checks for errors. This helps the system provide the appropriate response to a user and keep them informed of the current system state.

If there are no issues during registration and the user enters in information that pass the field requirements, then the rest of the function handles inputting the data into the database and sending the user an email to confirm their registration.

D.4 Registration Process

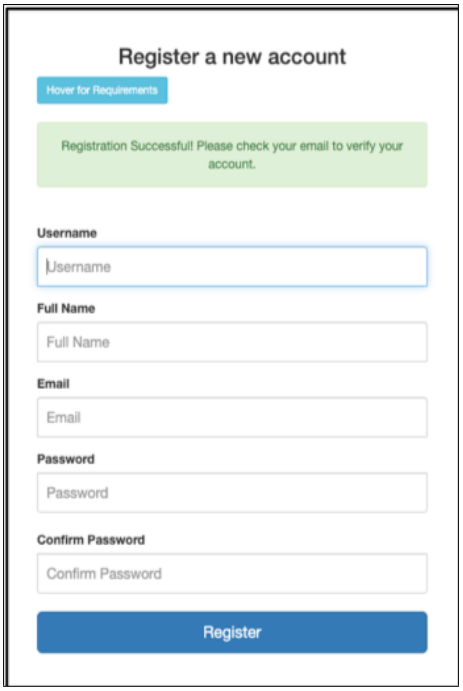

Figure D.1

In order to successfully register for an account, a user will have to follow a two-step process. This is to ensure that the new account can be verified. **Figure D.1** shows the response for when a user has successfully registered for a new account. It is in green to indicate a positive message. Errors are presented in Red for a negative response.

Figure D.2 shows an example email that a user will receive upon registering for a new account. Within the 'Click here to verify' link, there is a unique token generated using a salted SHA1 algorithm. This link takes the user to the /verify link. This token is used by the system as a query parameter when accessing .verify. The backend code will match the token with its associated user account to verify the user. If a match cannot be made between the token and a user account – then the system identifies that somebody is trying to verify an account that doesn't exist or a token that doesn't exist.

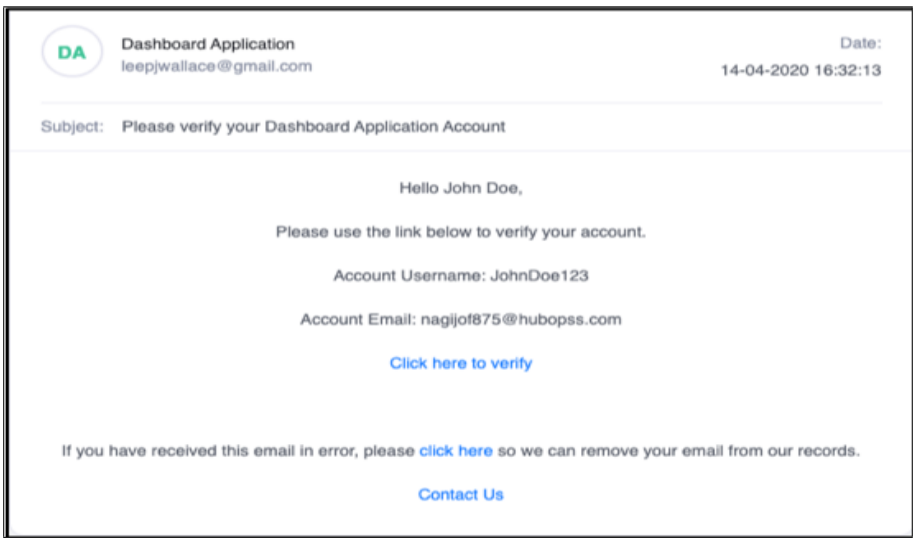

Figure D.2

Figure D.3 shows the response a user will receive after successfully verifying their account.

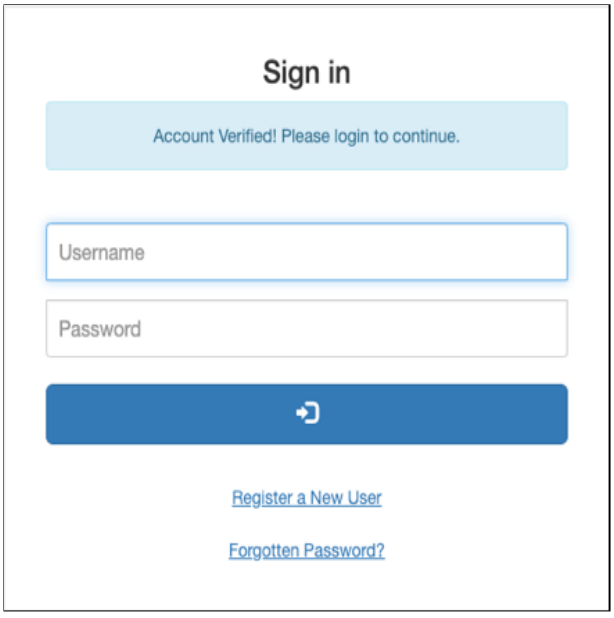

Figure D.3

D.5 Tree Directory Map of System

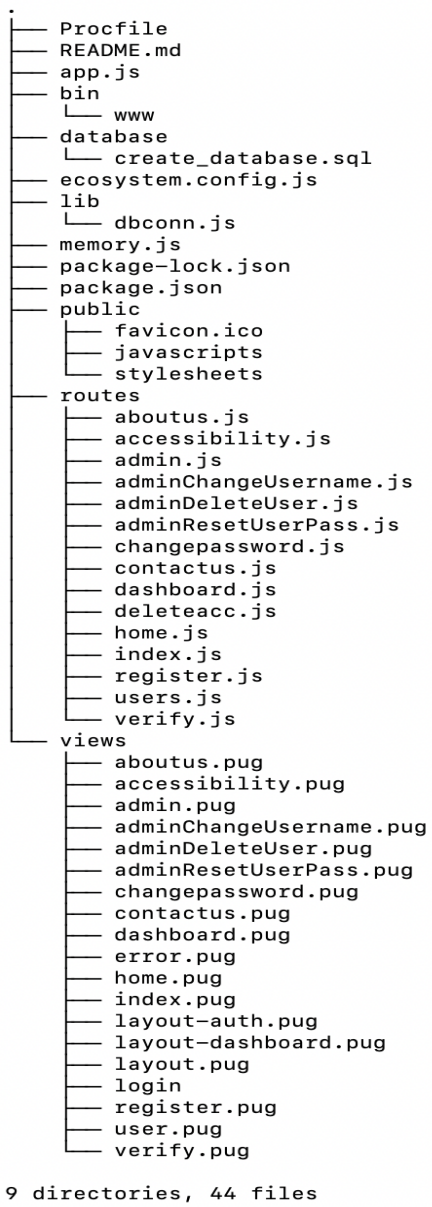

Figure D.4

D.6 Pug Template vs HTML Code

```
 doctype html
html
    head
        title My Page
    body
        h1 Heading
        p My paragraph here.
```

This will get translated to the following HTML content on demand.

```
<html>
   <head>
      <title>My Page</title>
   </head>
   <body>
      <h1>Heading</h1>
      <p>My paragraph here.</p>
   </body>
</html>
```

Figure D.5

Source:
Hossain, P., 2020. PUG Vs HTML. [online] Stack Overflow.
Available at: https://stackoverflow.com/questions/57992623/pug-vs-html-which-one-is-better-to-render-data-to-the-view-page-in-node-js-exp
[Accessed 13 April 2020].

E

Prototype One

This Appendix shows the first prototype that was developed during this Project. Functions shown below were developed further for the final prototype of the Project. This Prototype was completed after the third timebox. This prototype was the first to be developed in this Project.

E.1 Login System

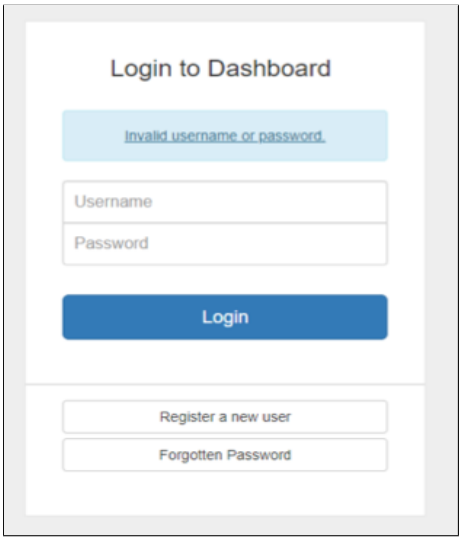

Figure E.1

Figure E.1 shows the login system found in prototype one. Headings were not aligned properly in this prototype and there were no labels for the input fields. Additionally, the link to the registration page and forgotten password button did not function in this prototype.

E.2

Registration Page

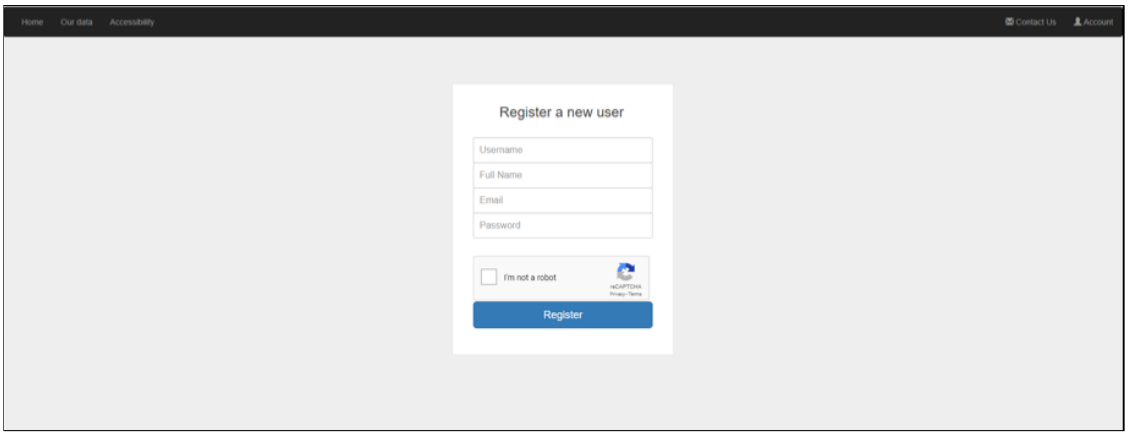

Figure E.2

Figure E.2 shows the Registration Page in prototype one. It doesn't check for a user in the database with an existing username or password. Also, this iteration of the Registration Page in this prototype does not provide the functionality for a user to re-enter their password and match it against the first password field. A user will typically be prevented from entering a password which they later cannot remember if there is a second field for the user confirming their password.

The final prototype of this Project has the above functionalities and is therefore an improvement of this prototype for a user. Furthermore, this iteration of the registration system allows the user to register, and then log in immediately without any verification. This is not a secure system as people can register and log in immediately, which presents various security issues. This verification system can be found in the final prototype and is based on email verification, which is popularly seen in most modern websites with an account functionality.

E.3 Home Page

Figure E.3

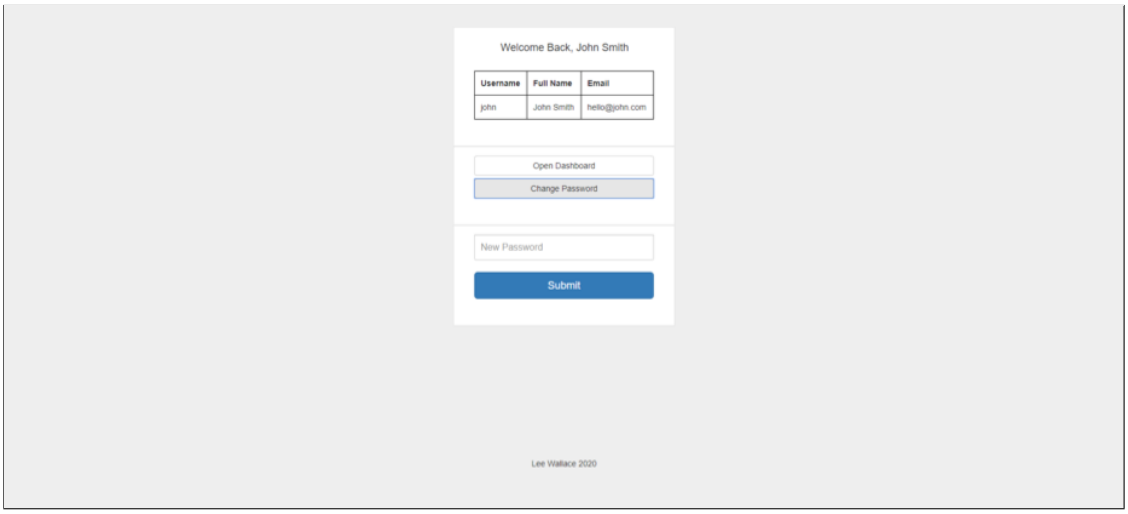

Figure E.4

Figure E.3 and **Figure E.4** show the home page in prototype one. When changing password in this prototype, **Figure E.4** shows that the Profile Page for a user does not provide a confirmPassword field for changing their password or field sanitisation and verification. This functionality was improved through iteration and can be found in the final prototype.

E.4 Accessibility Page

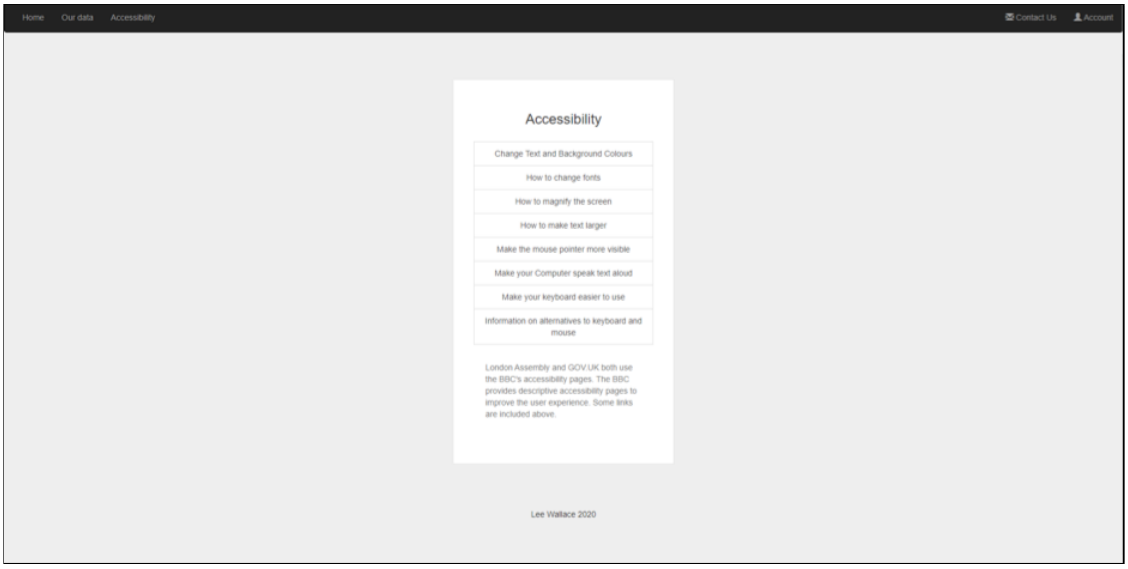

Figure E.5

Figure E.5 shows the Accessibility Page in prototype one. This page contains links to accessibility pages found on the BBC (British Broadcasting Company) website, which are widely used. The content on this page is however too narrow and will be changed in a future iteration of the system.

F

Prototype Two

F.1 Sign in Page

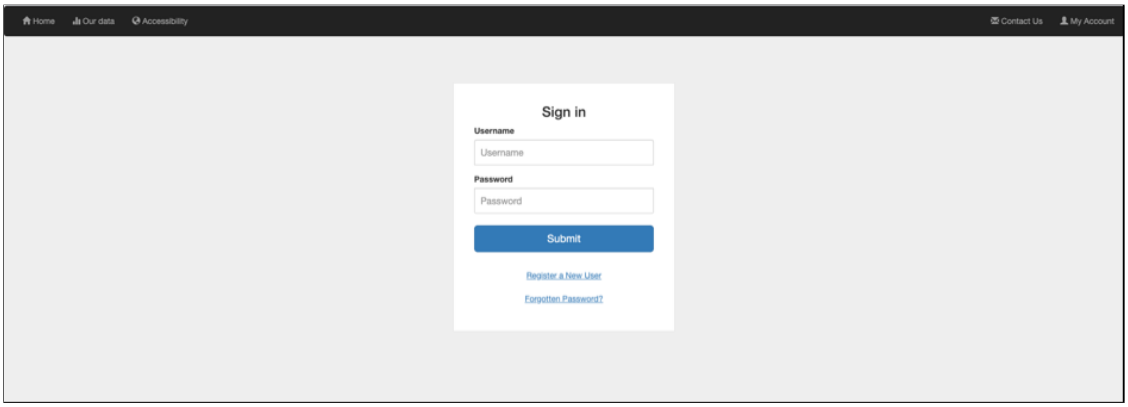

Figure F.1

Figure F.1 shows the sign in page for prototype two. This was changed based-on user feedback as the 'card' in the middle of the screen was very thin, which would sometimes render very small on some users' screens. Additionally, there were broken links at the top of this page. The 'My Account' navbar option on the top right would only redirect an unauthorised user to the sign in page, similar to the home button. The inclusion of this button was therefore considered redundant for an unauthorised user. As a result of this, code was changed in the template engine to ensure that the 'My Account' navbar option would only display for a logged in user.

F.2 Contact Us Page

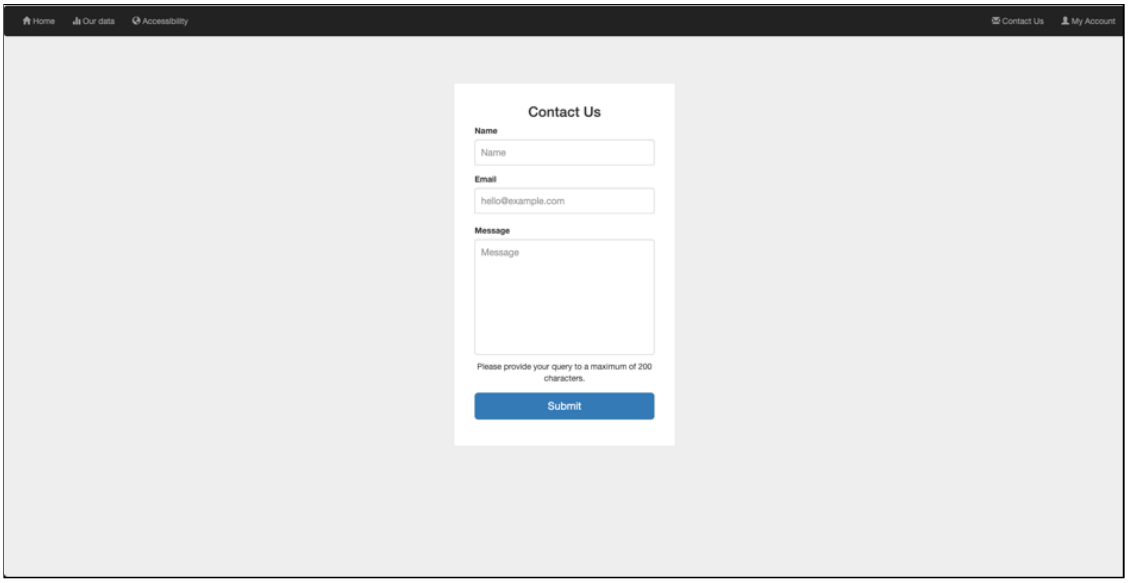

Figure F.2

Figure F.2 shows the contact us page in prototype two. The card on this page is very narrow and uncomfortable to use. The final prototype resized this to ensure that it was comfortable to use. Furthermore, the 'Message' box on this page did not limit input from a user, which could have created problems as there was no limit to how big a message a user could send. As a result, a maxlength attribute for the field input within the pug template was changed for this page in the final prototype, to limit a user's message to 200 characters.

F.3 Accessibility Page

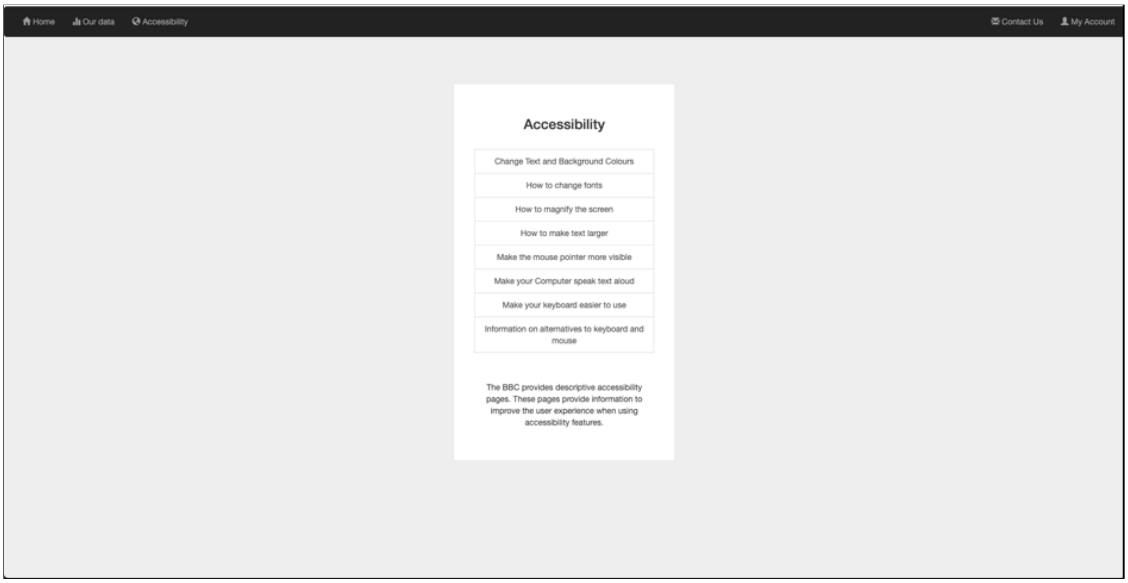

Figure F.3

Figure F.3 shows the accessibility page in prototype two. This page was considered to be too narrow and was widened in a further iteration.

F.4 Home Page

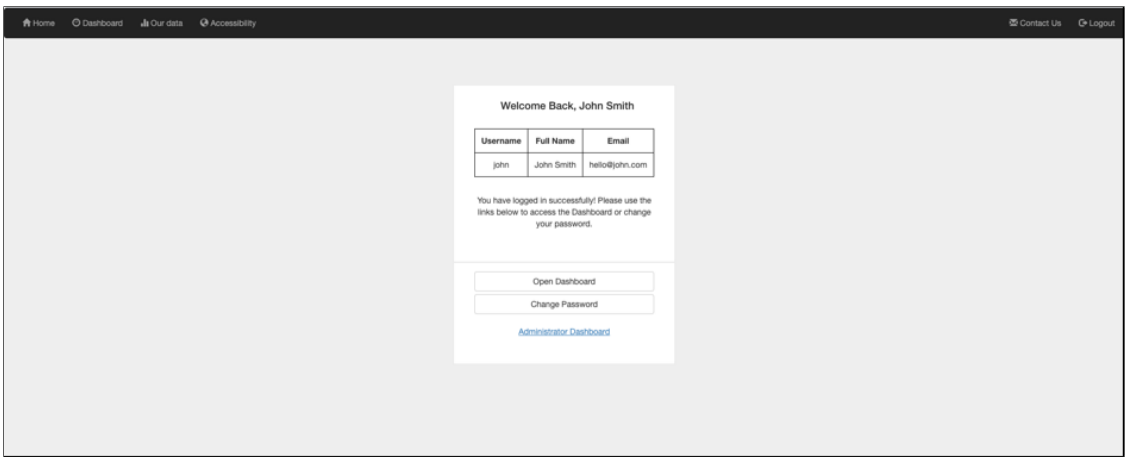

Figure F.4

Figure F.4 shows the home page in the second prototype. After reviewing user feedback, the developer decided that this page did not provide descriptive information about the website and

was more suitable for an ‘accounts’ page. Therefore, the final prototype introduced an accounts page and moved this account information to there.

The final prototype introduced a landing page, which provides information for new users before they log into the system. New users will be able to discover more information about the use of the site immediately in this way. Feedback from this project’s supervisor and second marker suggested to move this informative information *before* the log in process and was developed for the final prototype.

Additionally, the ‘logout’ navbar option was placed into a dropdown ‘My Accounts’ Menu option. This is commonly seen on most websites and is recognisable for a user. This change was introduced in the final prototype.

F.5 Visualisation Dashboard Page 1

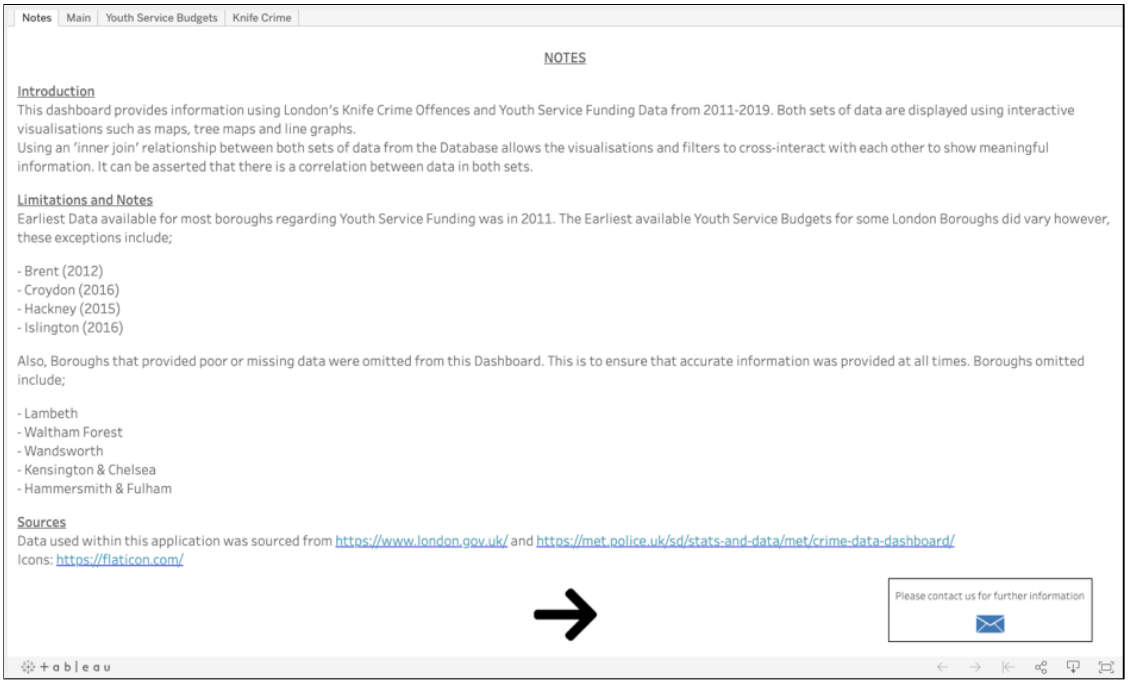

Figure F.5

Figure F.5 shows the first page of the visualisation dashboard in prototype two. Following user feedback, the developer decided that this page was not informative for a new user and should be moved elsewhere. Also, having two options for navigation (arrow at the bottom and tabs at the top left) was confusing for a user and should be changed.

The navigation options and the text on the starting page of the dashboard was changed to accommodate this feedback in the final prototype and provide a welcoming start page for a user.

F.6 Visualisation Dashboard Page 2

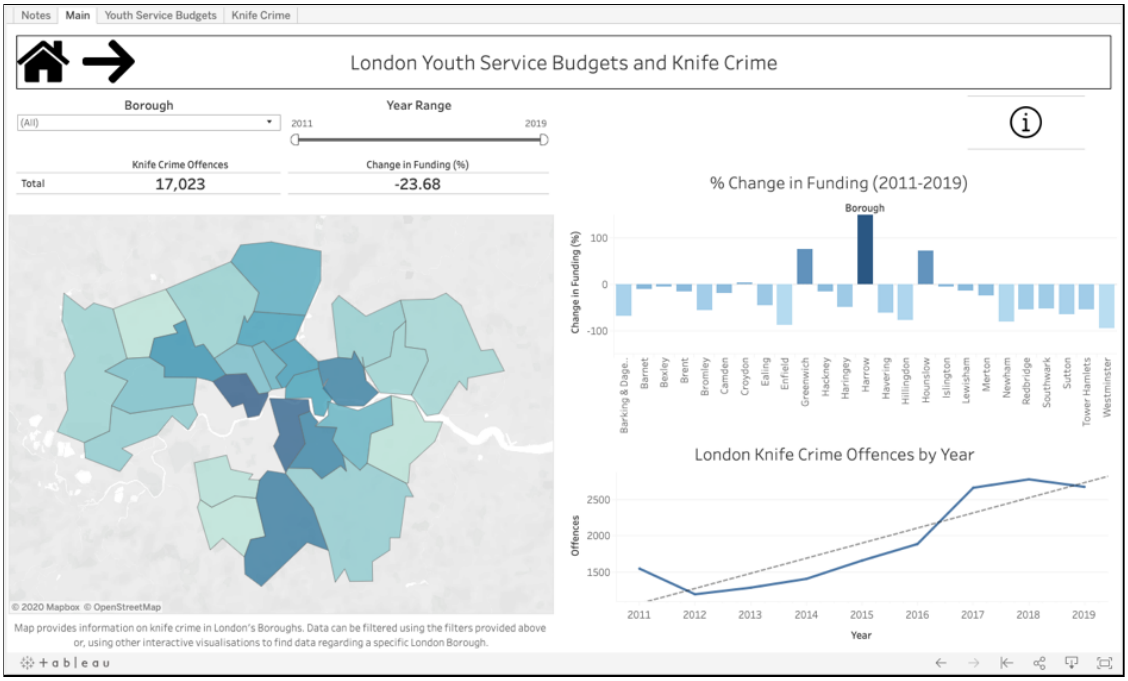

Figure F.6

Figure F.6 shows the second page of the visualisation dashboard in prototype two. User feedback showed that it was not clear for a new user in how to navigate the system due to having two navigation systems in place. Also, it was not clear for a user in how to proceed with the visualisations on the page. A descriptive paragraph was introduced in the final prototype to explain to users what information they can gather from the page that they are on.

F.7 Visualisation Dashboard Page 3

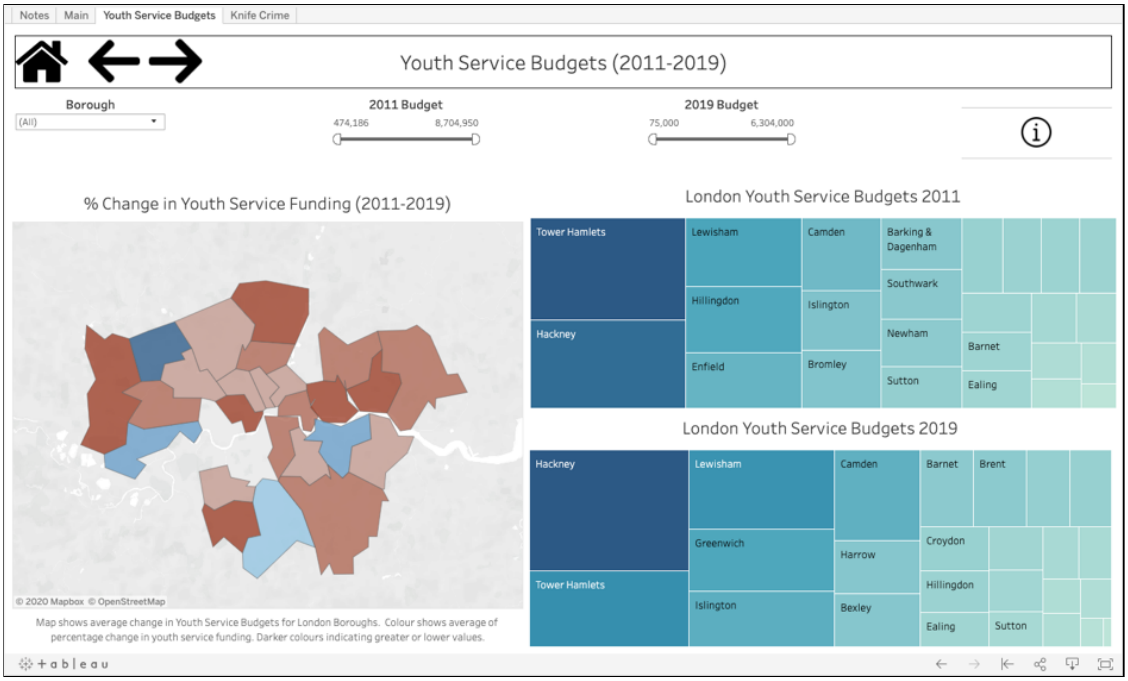

Figure F.7

Figure F.7 shows the third page of the visualisation dashboard in prototype two. Following user feedback, it was identified that the year sliders did not function as expected and provided a confusing experience for the user. The 'information + help' button at the top right did also not provide descriptive help information for a new user also.

As a result of this feedback, the year sliders were removed, and the information button was updated with informative information. The navigation options were changed. A descriptive paragraph was introduced at the top of the page in addition to this to introduce users to the data on the page.

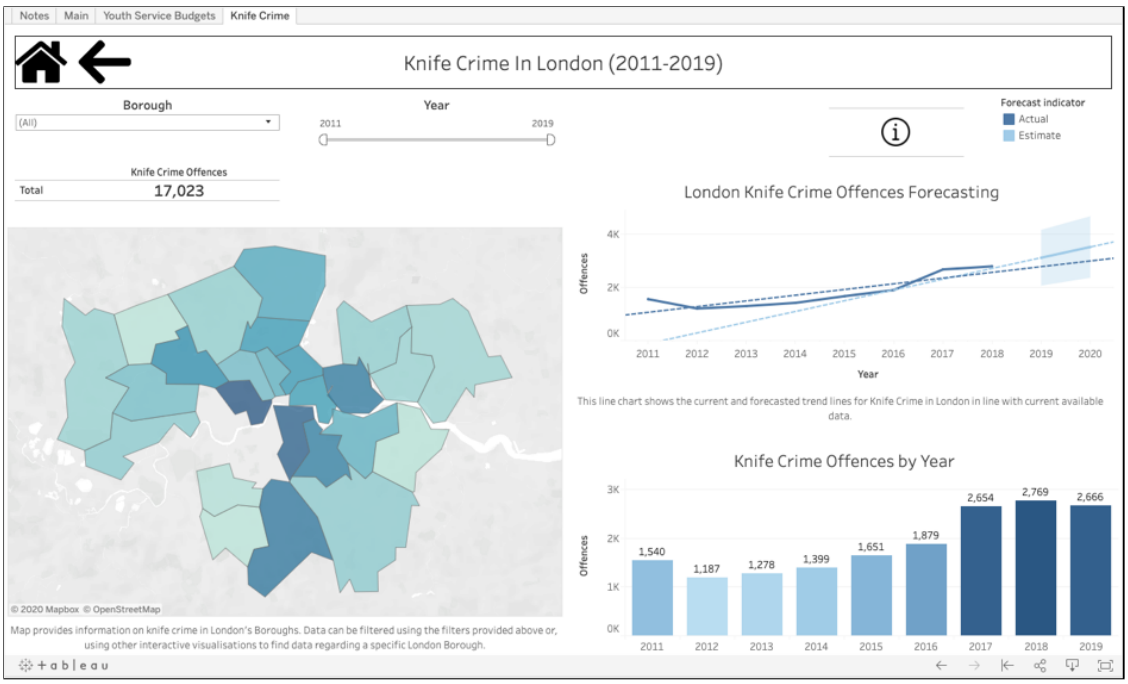

Figure F.8

Figure F.8 shows the fourth page for the visualisation dashboard in prototype two. User feedback shows that two navigation options were confusing. As a result, navigation options were fixed, and a descriptive paragraph was added to the top of the page to introduce the data for the user.

G

White-Box Testing

This Appendix shows the outcome of this project's white-box testing, which aims to test the functionality of the system. This testing was conducted with access to the internal code structure and implementations. Mainly, this method aims to test user inputs and actions to identify responses, with and within validation limits to gain a full understanding of the system and what its limitations are. This project used 82 test cases against the system to test its responses. The outcome of this testing can be seen in **Table I.1**.

Scoring System: 1 = Strongly Disagree, 2 = Disagree, 3 = Undecided, 4 = Agree, 5 = Strongly Agree

Table G.1: Final Prototype White-Box Testing

Test No.	Subject	Action	Expected Outcome	Actual Outcome
1	Navigation Bar	The Navigation Bar will display [Home, About Us, Contact Us, Accessibility] for user that is not logged in.	This standard Navigation Bar shows correctly for a user that is not logged in.	Successful
2	Navigation Bar	The Navigation Bar will display [Home, About Us, Contact Us, Accessibility, My Account (Account Settings + Logout)] For a user that is logged in.	For a logged in user, the 'my account' option will be added to the navbar so users can access their account settings or log out.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
3	System Routing	The System correctly routes all pages to their destinations for access	The system correctly routes pages	Successful
4	System Routing	The System rejects unauthorised access to pages across the application. Users who are not signed in cannot access the account settings page for example.	The System correctly redirects users away from pages with restricted access	Successful
5	Logging In	No data inputted into field (either field)	A bootstrap initiated popup with the text 'field required'	Successful
6	Logging In	Inputting >100 characters in either login field	A popup response with the text 'Invalid username or password'	Successful
7	Registration	The / registration link correctly displays the Registration Page.	Navigating to this link will open the / registration page.	Successful
8	Registration	Username less than five characters	A popup response with the text 'Password must be over five characters'	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
9	Registration	Username exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful
10	Registration	Username is less than 5 characters long	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
11	Registration	Fullname exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful
12	Registration	Email is not in a correct format (does not include an @ symbol or a .com/.co.uk etc).	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
13	Registration	Email exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful
14	Registration	Password entered by a user is less than 5 characters long	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
15	Registration	Password exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
16	Registration	Password and confirmPassword fields do not match	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
17	Registration	When moving a cursor over the 'hover for requirements' button. A tooltip will appear informing the user of registration requirements.	The tooltip works as expected.	Successful
18	Registration	A new users email is successfully hashed and salted for database storage.	The password is hashed successfully.	Successful
19	Registration	The system will inform the user where there is a successful registration.	The system will immediately notify the user of a successful registration via a popup and inform them to check their emails to continue the registration process.	Successful
20	Registration	A user will receive a verification email upon successful registration.	A user will receive a verification email upon successful registration.	Successful
21	Registration	A user can click the verification link to verify their account.	A user can use the verification link sent to them to verify their account. It will pass a token to the website which links to a user account. The system will then verify the linked user.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
22	Registration	A user can delete their account using the verification email.	Once a user has clicked the delete account link, their account will be deleted.	Successful
23	Cookies Policy Notice	Cookie Policy shows for a new user.	Cookies Policy works as expected and shows for a new user.	Successful
24	Cookies Policy Notice	Cookie Policy 'Learn More' button routes a user to the About Us page but stays present unless a user has accepted the cookie policy.	The 'Learn More' button on the Cookies Policy routes the user to the About Us page and stays present.	Successful
25	Cookies Policy Notice	Cookie Policy 'Accept' button ensures that a user will not see the cookie policy for 365 days. Any subsequent logins to the site will not display the notice	Upon clicking accept, a confirmation cookie will be placed on the users machine. This cookie is read when the site loads to check if a user has accepted the policy.	Successful
26	About Us Page	The /aboutus link correctly displays the About Us Page.	Navigating to this link will open the /aboutus link.	Successful
27	About Us Page	The /aboutus link correctly displays the About Us Page.	Navigating to this link will open the /aboutus link.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
28	Contact Us Page	The /contactus link correctly displays the Contact Us Page.	Navigating to this link will open the /contactus page.	Successful
29	Contact Us Page	Name exceeds maxlength limit of 40 characters.	Input field will prevent input exceeding 40 characters from a user.	Successful
30	Contact Us Page	Email exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful
31	Contact Us Page	Message exceeds maxlength limit of 200 characters	Input field will prevent input exceeding 200 characters from a user.	Successful
32	Contact Us Page	Email is not in a correct format (does not include an @ symbol or a .com/.co.uk etc).	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
33	Contact Us Page	Form correctly sends email to a pre-determined administrator account.	The pre-determined administrator email address will receive the contents of this form. (**See Appendix K**)	Successful
34	Contact Us Page	Administrator account correctly receives this email with the contents formatted for readability.	The pre-determined administrator email address will receive the contents of the form correctly.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
35	Forgotten Password Page	The /changepassword link correctly displays the Forgotten Password Page.	Navigating to this link will open the /changepassword page.	Successful
36	Forgotten Password Page	Email exceeds maxlength limit of 40 characters	Input field will prevent input exceeding 40 characters from a user.	Successful
37	Forgotten Password Page	Email is not in a correct format (does not include an @ symbol or a .com/.co.uk etc).	Upon submission of the registration form – the page will refresh with a message informing the user of this validation error.	Successful
38	Forgotten Password Page	Email does not exist in database	Upon form submission, the system will provide the user with an error stating this. This should be changed to make it more secure however.	Successful - Changes needed
39	Forgotten Password Page	Email exists in database and form is submitted successfully.	The user will receive an email asking for them to confirm that they want their password to be changed with a link provided (**See Appendix K**)	Successful
40	Forgotten Password Page	Upon clicking the link in the forgotten password email, the user will be redirected to the website, which will tell them that they have received a new password in their email.	User successful redirects, with information asking them to check their emails for their new password.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
41	Forgotten Password Page	User is able to log in with their new password	User is able to log in with new password.	Successful
42	Accessibility Page	The /accessibility link correctly displays the Accessibility Page.	Navigating to this link will open the /accessibility page.	Successful
43	Accessibility Page	All links correctly route the user to the corresponding BBC accessibility information page.	Links to all functions correctly and redirects the user.	Successful
44	Home Page	The /home link correctly displays the Home Page for a logged in user.	Navigating to this link will open the /Home page when a user is logged in	Successful
45	Home Page	The Open Dashboard button will open the /dashboard link.	Navigating to this link will open the /dashboard page.	Successful
46	Home Page	The My Account button will open the /users link.	Navigating to this link will open the /users page, which displays account information.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
47	Home Page	The Administrator Settings button will open the /admin link when a user has administrator privileges.	Navigating to this link will open the /admin page, which provides administrator settings. This can only be accessed by users who had administrator privileges.	Successful
48	Home Page	The Administrator Settings button will not display for a user, if they do not have administrator privileges.	The Administrator Settings button does not show for a user who does not have administrator privileges.	Successful
49	Account Settings Page	The /users link correctly displays the Account Settings.	Navigating to this link will open the /users (Account Settings) page.	Successful
50	Account Settings Page	Page correctly displays user information from the database.	The Account Settings page has a table that displays user information.	Successful
51	Account Settings Page	Page allows the user to change their password with all validation functionality working.	User can change their password, providing that they meet the validation requirements (both passwords entered match etc).	Successful
52	Account Settings Page	Page allows the user to delete their account, upon clicking this button, a modal will appear asking them to confirm their decision.	When clicking the 'Delete Account' button, the page will display a modal on screen asking them to cancel or confirm this process.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
53	Account Settings Page	Once a user has confirmed their account deletion, the account will be logged out to prevent logical errors and deleted from the database.	User will be logged out and returned to the signin page, with the account deleted.	Successful
54	Dashboard	The /dashboard link correctly displays the Dashboard Page.	Navigating to this link will open the /dashboard page.	Successful
55	Dashboard	Tableau Dashboard is sized correctly to screen at a resolution of 1280x720.	Tableau Dashboard is correctly sized and positioned in the centre of the screen/viewport.	Successful
56	Dashboard	Tableau Dashboard provides tabs on the top left of the container for navigation.	Page tabs are located correctly and can be used to navigate to any page at any time.	Successful
57	Dashboard	Tableau Dashboard will correctly display the Introduction Page as the first page.	Introduction Page is the first page displayed for a user.	Successful
58	Dashboard	Tableau Dashboard Introduction page allows a user to press an email icon to contact the administrator for further information.	This ‘mailto:’ link works successful and prompts the user with a draft email using their default email service.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
59	Dashboard	Tableau Dashboard Research page correctly scales according to the sizing of the dashboard in the viewport and shows the research information.	Research pages correctly scales and shows information.	Successful
60	Dashboard	Tableau Dashboard Main page correctly scales according to the sizing of the dashboard in the viewport and shows the visualisation dashboard.	Main Page correctly scales and shows information	Successful
61	Dashboard	Tableau Dashboard Main page visualisations are cross-interactive with each other and automatically filter data.	When interacting with the visualisations, they correctly filter out data.	Successful
62	Dashboard	Tableau Dashboard Main page filters function correctly to filter data.	Filters change the values/contents of the data correctly.	Successful
63	Dashboard	Tableau Dashboard Main page totals are shown correctly by default and when using filtered data.	Totals change correctly according to the default or filtered data.	Successful
64	Dashboard	Tableau Dashboard Youth Service Budgets page correctly scales according to the sizing of the dashboard in the viewport and shows the visualisation dashboard.	Youth Service Budgets page correctly scales and shows information.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
65	Dashboard	Tableau Dashboard Youth Service Budgets page visualisations are cross-interactive with each other and automatically filter data.	When interacting with the visualisations, they correctly filter out data.	Successful
66	Dashboard	Tableau Dashboard Youth Service Budgets page filters function correctly to filter data.	Filters change the values/contents of the data correctly.	Successful
67	Dashboard	Tableau Dashboard Youth Service Budgets page totals are shown correctly by default and when using filtered data	Totals change correctly according to the default or filtered data.	Successful
68	Dashboard	Tableau Dashboard Knife Crime page correctly scales according to the sizing of the dashboard in the viewport and shows the visualisation dashboard.	Knife Crime page correctly scales and shows information.	Successful
69	Dashboard	Tableau Dashboard Knife Crime page visualisations are cross-interactive with each other and automatically filter data.	When interacting with the visualisations, they correctly filter out data.	Successful
70	Dashboard	Tableau Dashboard Knife Crime page filters function correctly to filter data.	Filters change the values/contents of the data correctly.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
71	Dashboard	Tableau Dashboard Youth Service Budgets page totals are shown correctly by default and when using filtered data	Totals change correctly according to the default or filtered data.	Successful
72	Dashboard	The /admin link correctly displays the Administrator Settings Page for a user with administrator privileges.	Navigating to this link will open the /admin page for a user with administrator privileges.	Successful
73	Dashboard	Existing User Check (Valid User).	If a user exists, this check will return an update to the user, informing them that the user exists.	Successful
74	Dashboard	Existing User Check (User does not exist).	If a user doesn't exist, this check will return an update to the user, informing them that the user exists.	Successful
75	Dashboard	Change Account Username (Valid User).	If a user exists, this function will change the username for the selected user.	Successful
76	Dashboard	Change Account Username (User does not exist).	If a user does not exist, this function will not be carried out, and will return an update to the user informing them that the selected user does not exist.	Successful

Continued on next page

Table G.1 – *Continued from previous page*

Test No.	Subject	Action	Expected Outcome	Actual Outcome
77	Dashboard	Change Account Password (Valid User).	If a user exists, this function will reset the password for the selected user.	Successful
78	Dashboard	Change Account Password (Valid User) Success Message.	The user will receive a new password via email following this function	Successful
79	Dashboard	Change Account Password (User does not exist).	If a user does not exist, this function will not be carried out, and will return an update to the user informing them that the selected user does not exist.	Successful
80	Dashboard	Delete User (Valid User).	If a user exists, this function will delete the selected user.	Successful
81	Dashboard	Delete User (Invalid User).	If a user does not exist, this function will not be carried out, and will return an update to the user informing them that the selected user does not exist.	Successful
82	Dashboard	Return to Home Page Button.	When clicked, this button will redirect the user to the Home Page.	Successful

H

Usability Testing

This Appendix shows the outcome of this project's usability testing, using Neilson's Heuristics. In order to ensure that this testing is consistent throughout this project, the same linear scoring system of 1-5 will be used to score various heuristics against the final prototype. The outcome of this testing can be seen in **Table J.1**.

Scoring System: 1 = Strongly Disagree, 2 = Disagree, 3 = Undecided, 4 = Agree, 5 = Strongly Agree

Table H.1: Final Prototype Usability Testing

Heuristic	Score	Implementation
Visibility of System Status	1 2 3 **[4]** 5	The system keeps users informed through the use of flash messages. This can inform a user of an invalid username or password when trying to log in for example. These messages are found in every user function that returns a response.
Match between system and the real world	1 2 3 4 **[5]**	Words and phrases found during common conversation in the real world are used throughout the application.
User control and freedom	1 2 3 **[4]** 5	The user is able to navigate to any page, at any time during browsing the application. It also asks the user for confirmation in certain actions, such as deleting their account.

Continued on next page

Table H.1 – *Continued from previous page*

Heuristic	Score	Implementation
Consistency and standards	1 2 **[3]** 4 5	Words and phrases are used consistently across the application, confusing language is not used. Clear and concise throughout.
Error prevention	1 2 3 4 **[5]**	Error prevention in the system is excellent. Errors are caught all throughout the codebase where possible to prevent the application from crashing. It also asks the user for confirmation in certain actions, such as deleting their account.
Recognition rather than recall	1 2 **[3]** 4 5	Recognition is used throughout the application where possible, account settings will be populated with relevant user information. Login fields will autofill if a user was to use the website more than once. The 'Cookies Policy' notice also stores the users response for 365 days to ensure that they don't have to continuously confirm the cookie policy.
Flexibility and efficiency of use	1 2 3 **[4]** 5	The system is flexible and allows novice and advanced users to navigate and use the system as quickly as possible, or as slowly as possible.
Aesthetic and minimalist design	1 2 3 4 **[5]**	The system keeps a consistent design throughout to prevent confusing users. The minimalist design is easy to understand, and the system looks appealing to a user, especially the visualisation dashboard.
Help users recognise, diagnose, and recover from errors	1 2 3 4 **[5]**	Informative errors are produced where necessary throughout the system. An unsuccessful login attempt will prompt the user that their username or password is invalid for example.

Continued on next page

Table H.1 – *Continued from previous page*

Heuristic	Score	Implementation
Help and documentation	1 2 3 4 **[5]**	The system provides scope for users to access help and documentation throughout. An 'About Us' page is found on the web app and an 'Information Button' is found in the visualisation dashboard.
Total:	**43**	

Score: 43/50

Overall, the System scored similar to the MET product reviewed during Product Research (**See Chapter 4**) in this report. The developer understands that implementing a consistent layout at all times was a challenge, especially when having to adapt to different viewports. Furthermore, implementing a feature to fulfil the 'recognition rather than recall' heuristic was difficult as this required more time for development.

I

Email Examples

For the purpose of prototyping, this project assumes that the System mailer account is **'leepjwallace@gmail.com'**. In a production environment, this would be something like **'no-reply@company.com'**.

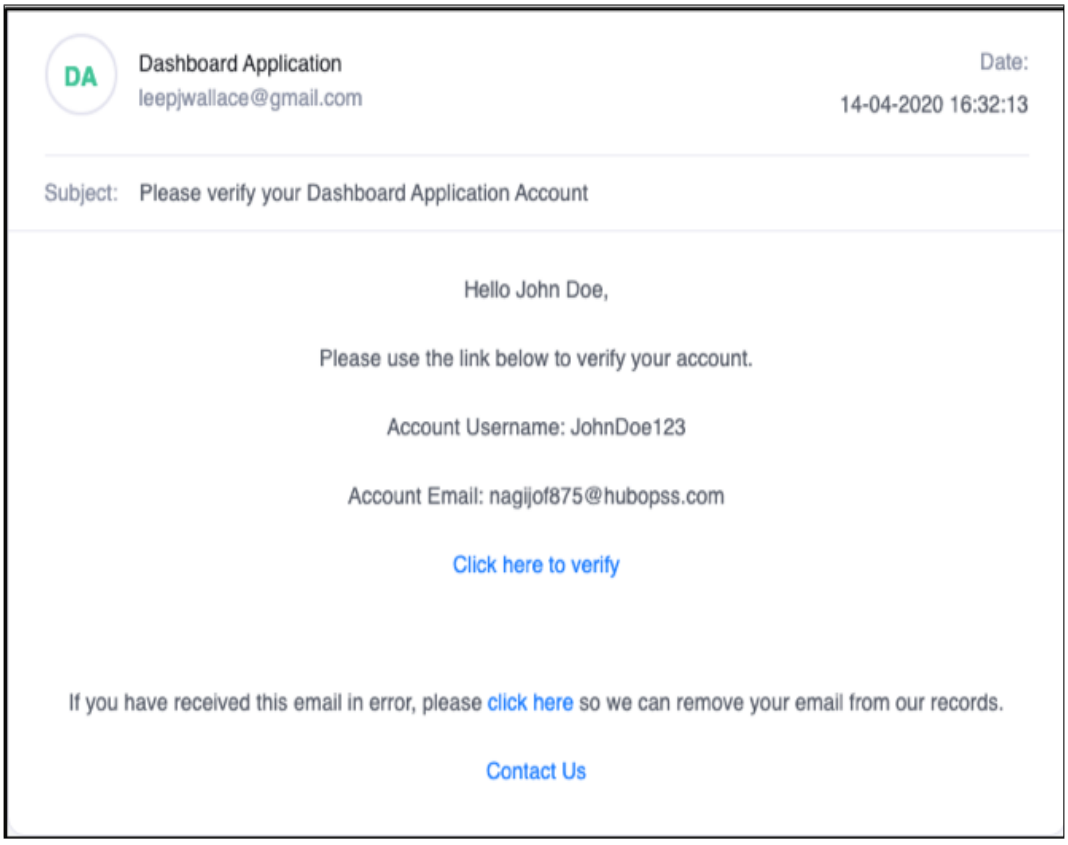

Figure I.1

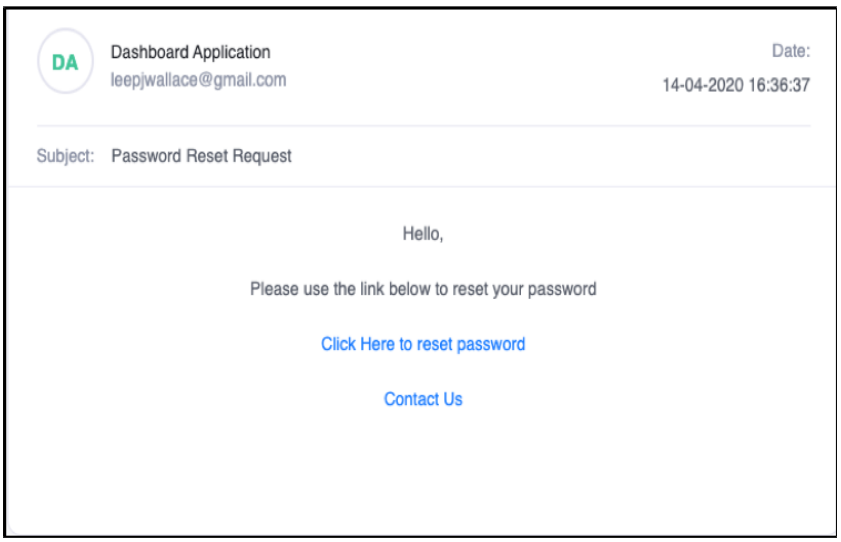

Figure I.2

Figure I.1 shows an example email that a user would receive when registering for a new account. **Figure I.2** shows what a user would receive when attempting to reset their password via the forgotten password page.

Figure I.3

Figure I.3 shows an example contact us form response that would be received by an administrator. It contains the senders name, reply-to email and their message. Additionally, the message will generate the current date and time at the time of submission and include this within the email.

J

Further Information

J.1 README.md Documentation

A README.md file contains information about files in the same folder. This is typically used in software development for information regarding software use. This file is presented in a markdown(.md) file type, which allows for special formatting.

This project has written a README.md file for documentation regarding installing dependencies, running the code locally and further developments. This file can be found in the project code folder.

J.2 Limitations of Data

Earliest Data available for most boroughs regarding Youth Service Funding was in 2011. The Earliest available Youth Service Budgets for some London Boroughs did vary however, these exceptions include;

- Brent (2012)
- Croydon (2016)
- Hackney (2015)
- Islington (2016)

Also, Boroughs that provided poor or missing data were omitted from this Dashboard. This is to ensure that accurate information was provided at all times. Boroughs omitted include;

- Lambeth
- Waltham Forest
- Wandsworth
- Kensington & Chelsea
- Hammersmith & Fulham

Data used within this application was sourced from https://www.london.gov.uk/ and https://met.police.uk/sd/stats-and-data/met/crime-data-dashboard/

K

Additional Low-level Prototypes

This appendix shows all low-level prototypes produced as a basis of design for the high-fidelity prototype.

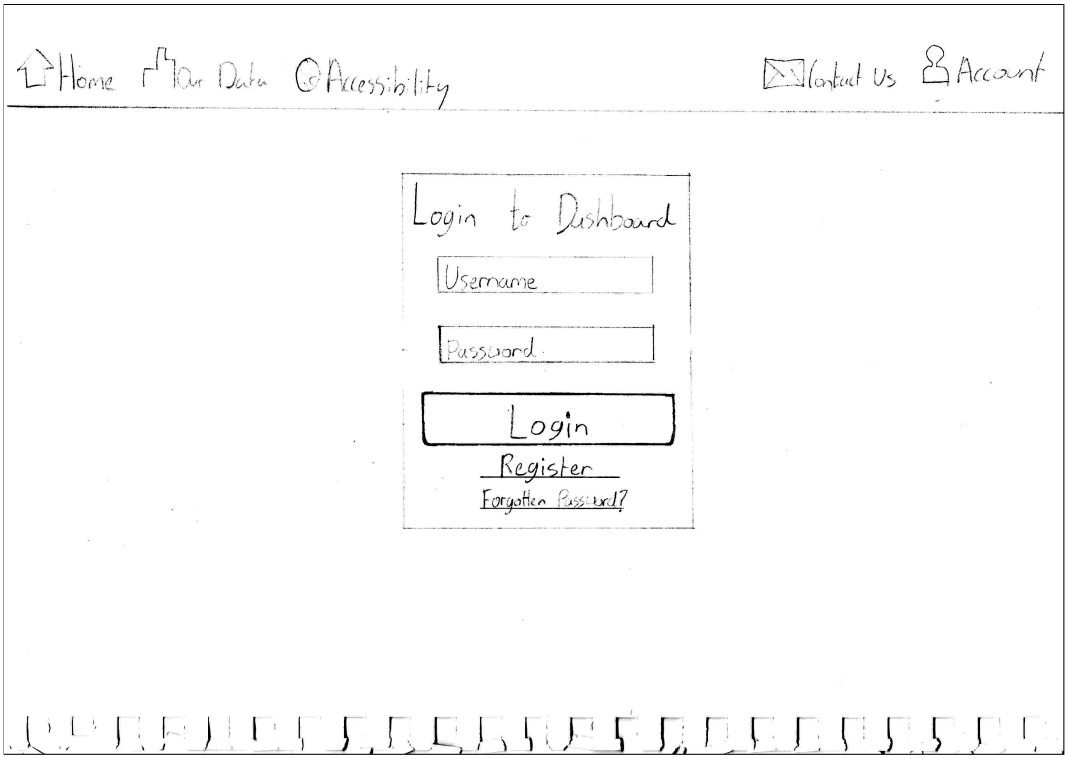

Figure K.1: Login Page

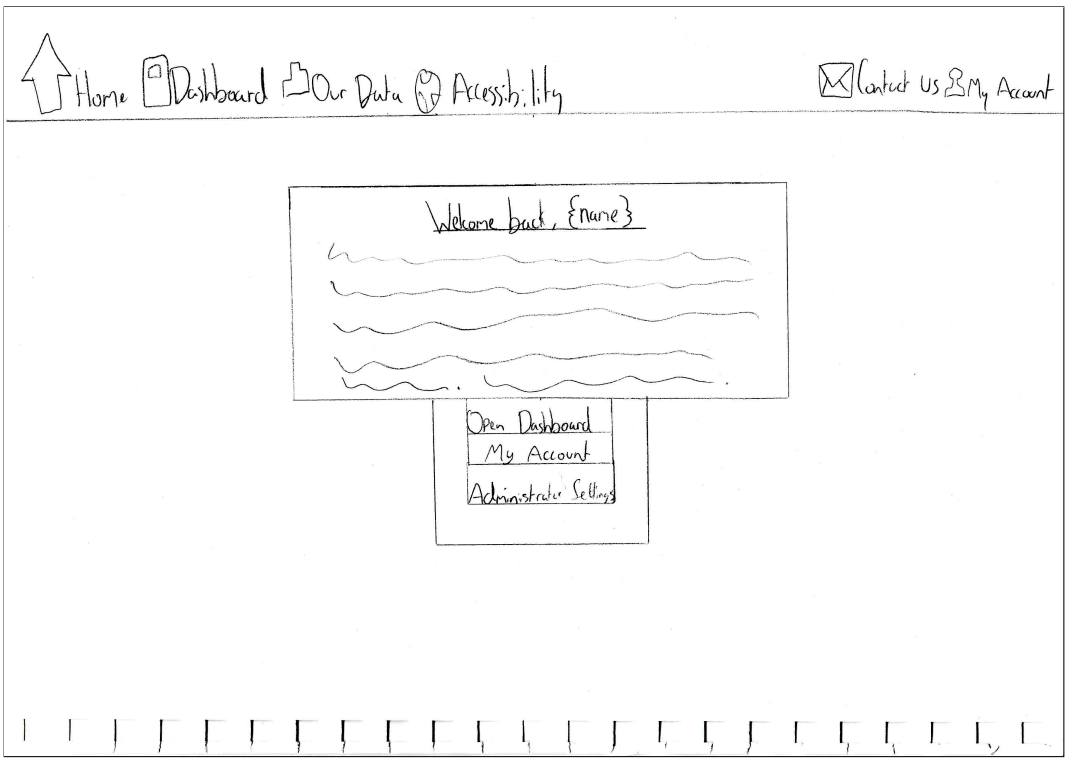

Figure K.2: Web App Home Page

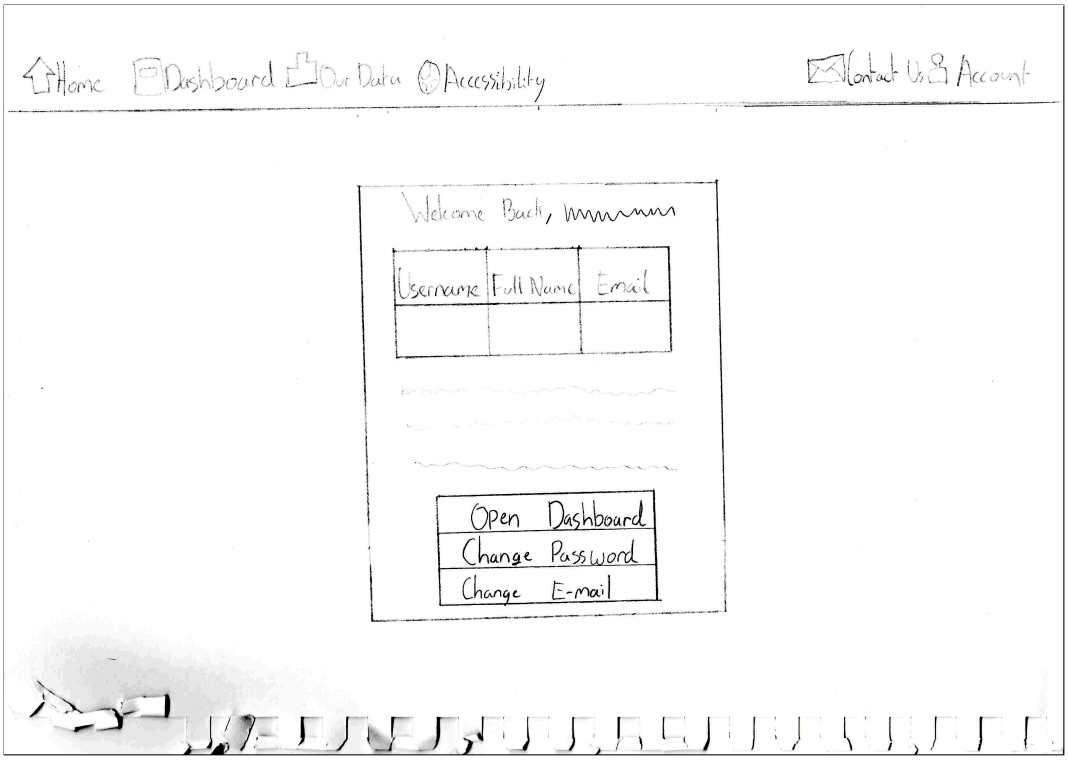

Figure K.3: Account Settings Page

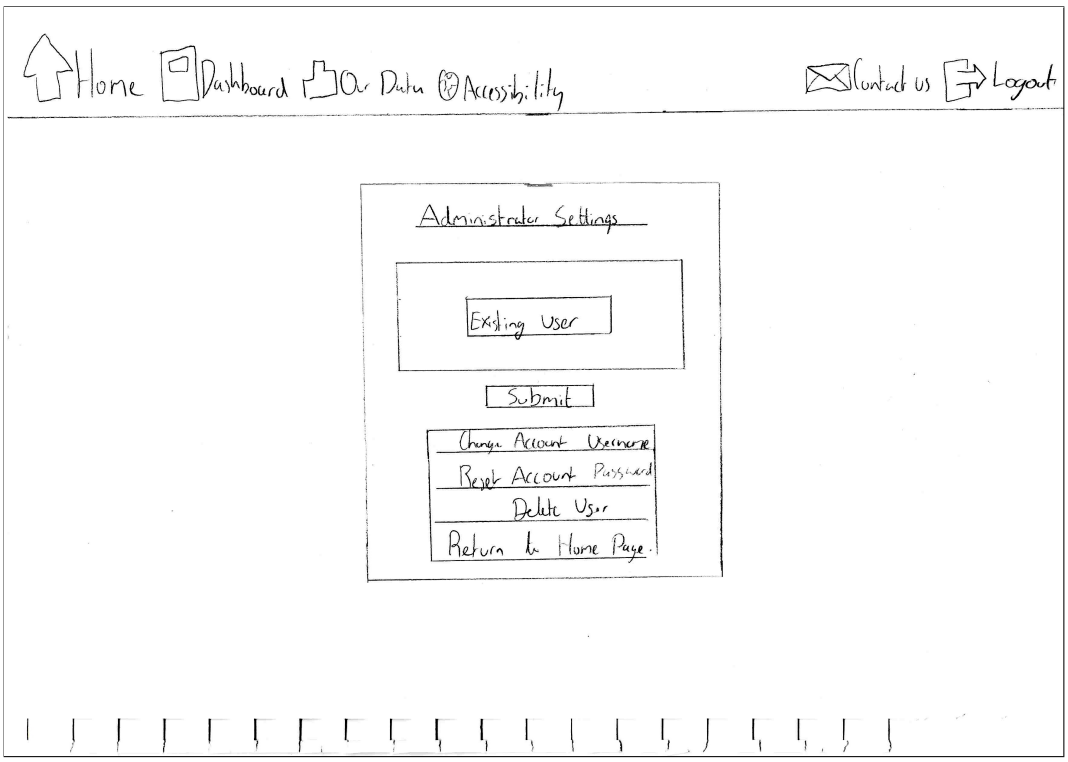

Figure K.4: Administration Settings Page

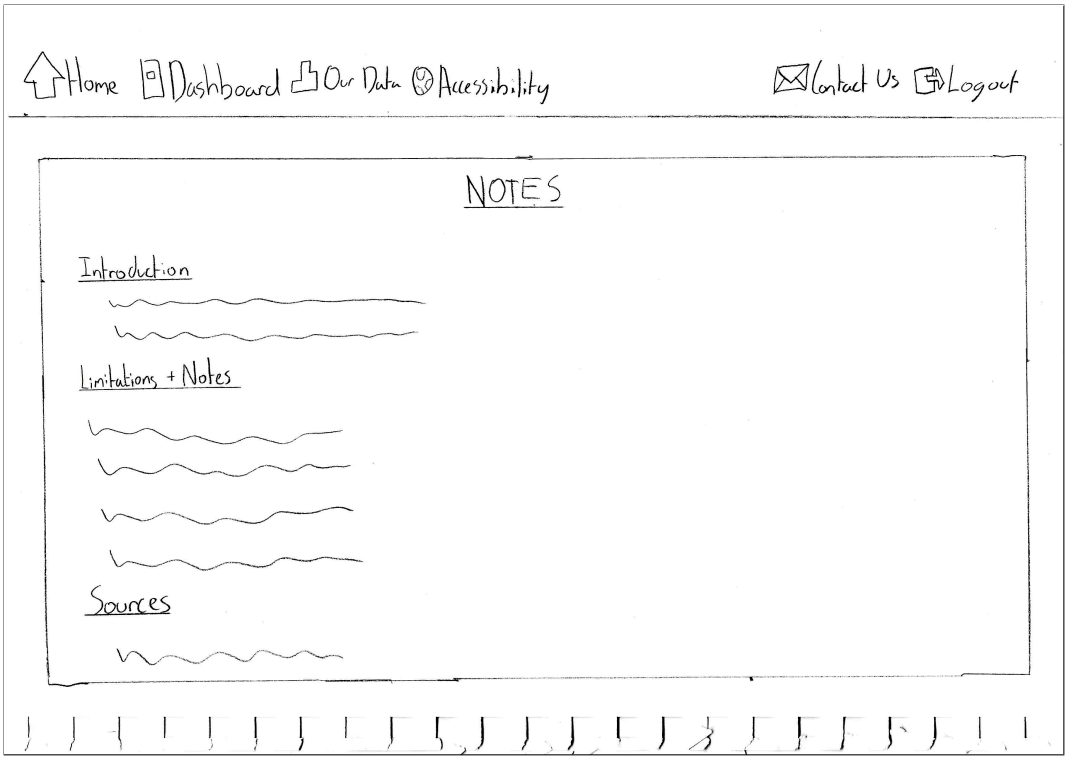

Figure K.5: Dashboard Home Page

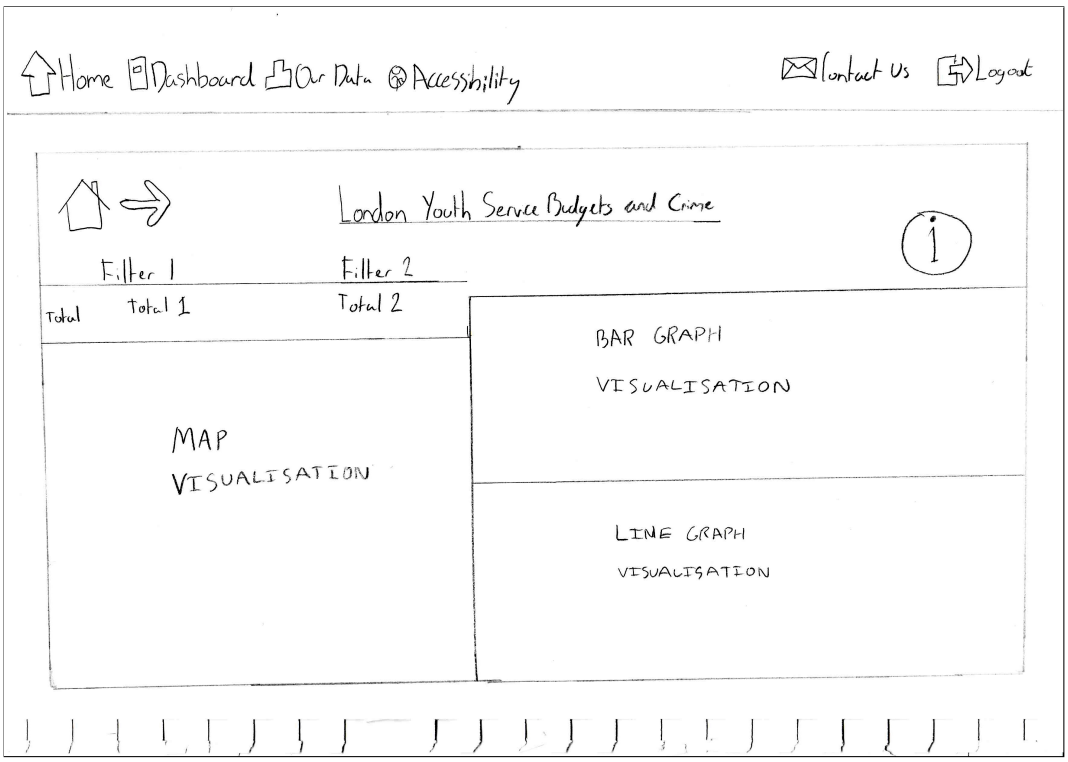

Figure K.6: Dashboard Main Visualisation Page

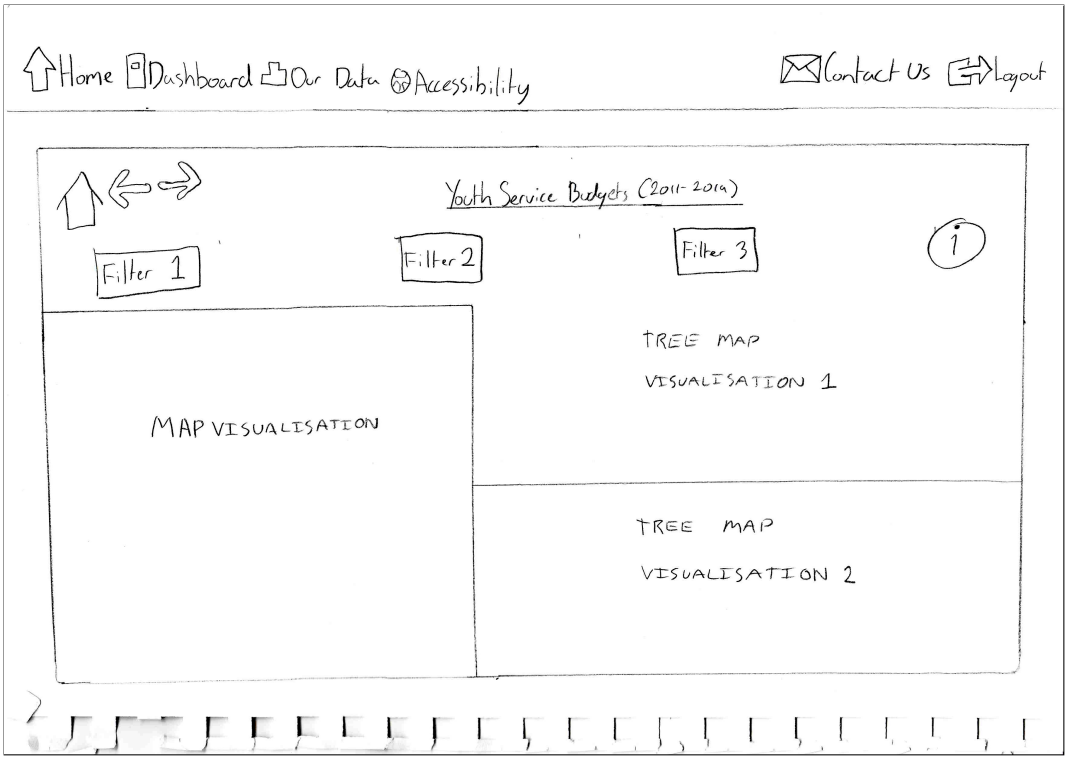

Figure K.7: Dashboard Youth Service Budgets Page

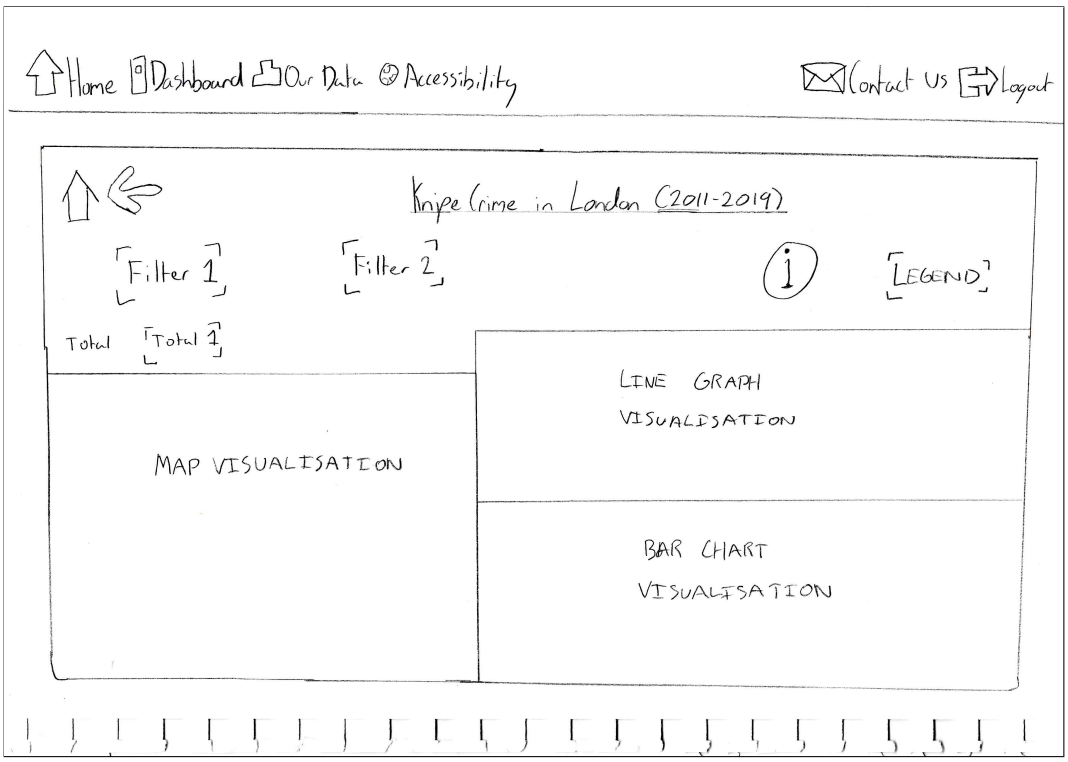

Figure K.8: Dashboard Knife Crime Page

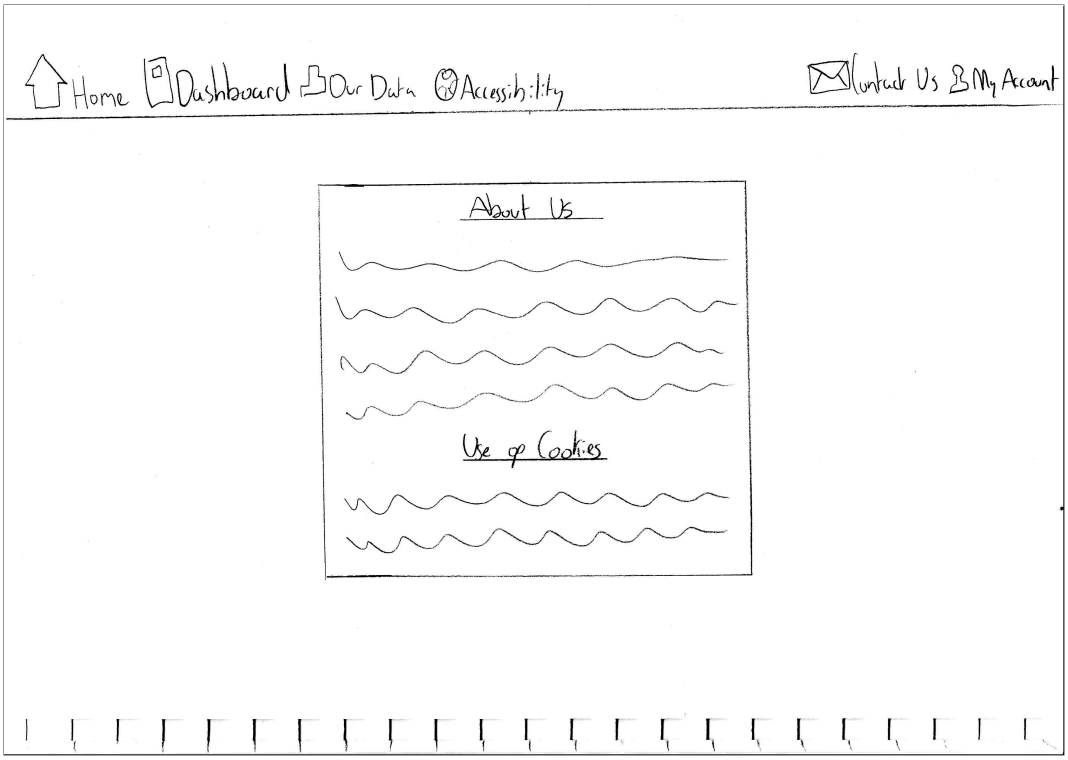

Figure K.9: About Us Page

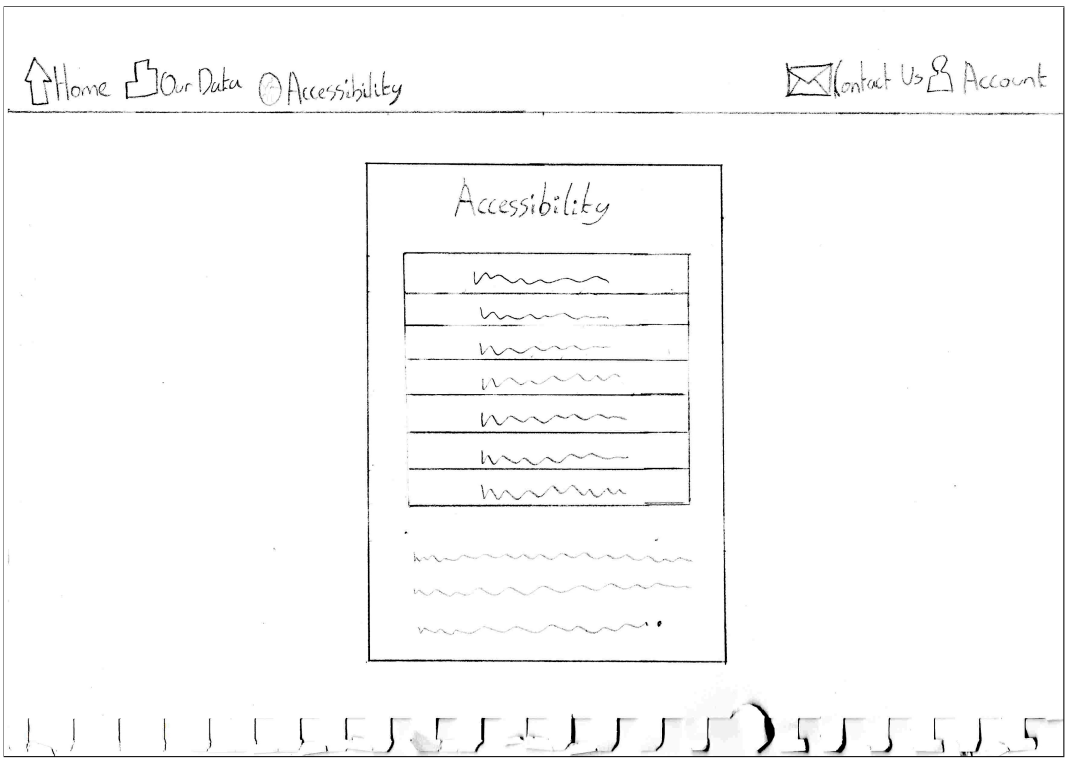

Figure K.10: Accessibility Page

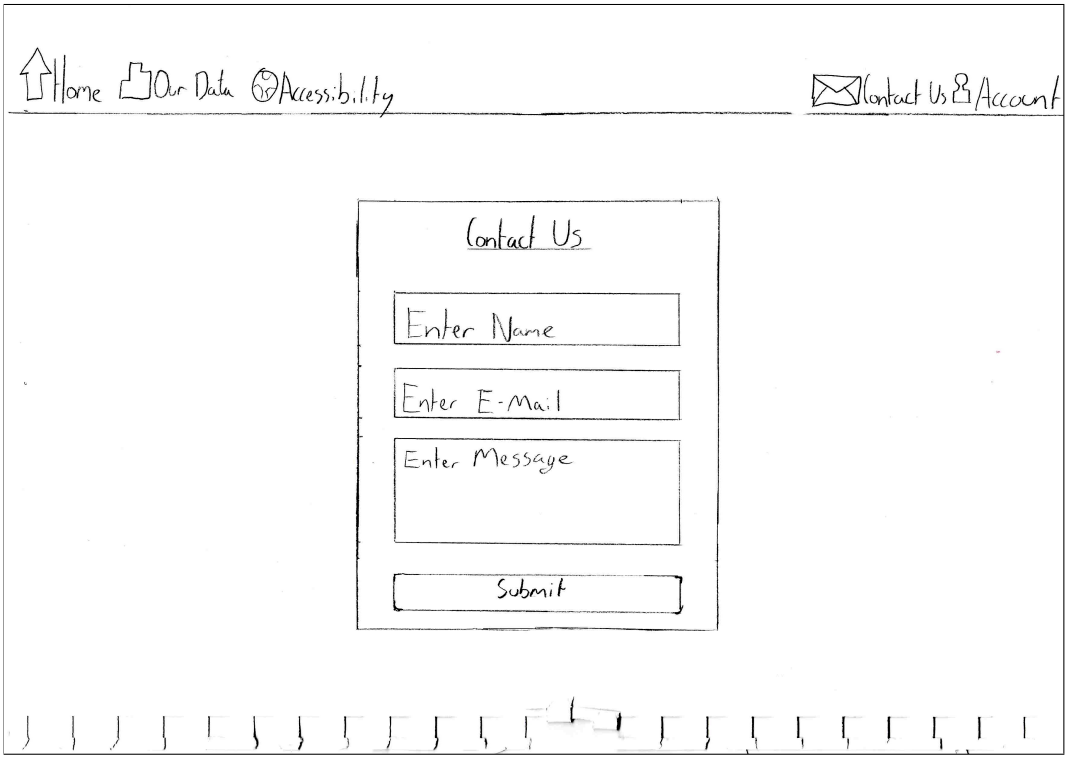

Figure K.11: Contact Us Page