UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

Computer Science Honours
Final Paper
2016

Title: Visualisation of Accident Data in South Africa.

Author: William Lumala

Project Abbreviation: M.O.V.E.R

Supervisor(s): Tommie Meyer

Category	**Min**	**Max**	**Chosen**
Requirement Analysis and Design	0	20	10
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	8
System Development and Implementation	0	15	12
Results, Findings and Conclusion	10	20	15
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	**80**

Visualisation of Accident Data in South Africa

William Lumala
University of Cape Town
lmlwil001@myuct.com

ABSTRACT

Previous research on road safety in South Africa has shown that a high number of accidents with numerous fatalities still occur despite campaigns aimed at identifying solutions to the problem. The main causes of road accidents are speeding and drunken driving. The Council for Scientific and Industrial Research of South Africa has identified the need for a visualisation tool to pinpoint the exact location of an accident to aid in their research on road safety. This project also sought to collect and visualise data for running accidents. Running accidents, in the context of this paper, are defined as accidents that occur while someone is casually jogging or running.

In this paper, we present the design of a geo-spatial visualisation showing the location of car and running accidents along with other relevant information. Such a visualisation is an improvement in the way data about road accidents is presented in South Africa. An interactive visualisation may help researchers to easily identify patterns and trends in the data faster than viewing the data in static graphs and tables. Several dimensions (variables) of data are displayed in addition to the location, such as the date and time of the accident. Since the Visualisation is interactive, users are able to interactively change the data that they view.

A user-centred design approach was used to develop the visualisation and usability tests conducted to evaluate the design in terms of effectiveness and usability. Usability testing showed that the application provided an efficient means to visualise accident data. Furthermore, evaluation done the System Usability Scale showed that the application developed had a score of 78.75 which translates into a usable user interface. [1]

1. INTRODUCTION

The formulation of this project arose from a request by the Council for Scientific and Industrial Research (CSIR) to assist them in their research on road safety in South Africa. The CSIR in South Africa is one of the leading scientific and technology research organisations in Africa. Through its Defence, Peace, Safety and Security (DPSS) unit, the CSIR plays an integral role in the security of South Africa. The CSIR have identified the need for a system to track road accidents (i.e. car accidents and running accidents) and the data associated with these road accidents. This system consists of two Android-based applications; one to detect car accidents and the other to detect running accidents. The two Android applications, developed by my group-mates, are frequently referenced in this paper. In addition to the applications, the CSIR would like to have a web-based application to visualise the data collected by the Android applications. All these three systems working in tandem will help help in the research on road safety especially with regard to road accidents. The proposed system may be used to identifying dangerous locations and find out why accidents occur in those locations. This could preempt more accidents from occurring in the said locations. In this paper, we present the design of a geo-spatial visualisation which displays road accident data in South Africa.

1.1 Project Significance

Previous research on road safety in South Africa has shown that the number of road accidents remains high the campaigns aimed at improving road safety. In 2011, 1080 fatal crashes were recorded in the month of December alone [1]. For the year 2014 to 2015, more than 4500 deaths were attributed to road accidents, with data showing that drunken driving and over speeding are two of the major causes of road accidents [1]. The Road Traffic Management Corporation (RTMC), tasked with compiling, researching and releasing data on car crash statistics in South Africa, releases data on car accidents annually. However, this data does not show the physical location of an accident; a key aspect in the research on road safety. Furthermore, most of the data is displayed using tables and the visualisations are not interactive. By presenting accident data in a visual and interactive manner, patterns and trends may be identified [17] [3]. Such a visualisation would be useful in identifying vital information, such as the most dangerous locations and at the time of the day that accidents are most likely to occur. As mentioned by Keim et al. [17] , another advantage of using interactive data visualisations is that a user can directly interact with the visualisation and change the data that they are viewing based on the data variables (dimensions). A visualisation of this nature could be useful in identifying black spots, that is, locations that have a high number of car crashes.

1.2 Project Aims

This project had the primary aim of developing a web based application that presents car and running accident data in a visual and interactive manner. Meetings were held frequently with a CSIR representative in order to get a better understanding of the functionalities required from the system. The system (a web-based application) should be

[1]This thesis was completed as part of a Bachelor of Science Honours degree in Computer Science at the University of Cape Town in 2016.

able to pull data from a database and display it, mapping the GPS location (latitude and longitude) of an accident onto Google Maps. With the physical location of a data point being vital to the application, a geo-spatial visualisation was deemed necessary and appropriate.

Secondly, we aimed to develop a scalable application with a good user interface that can be used to visualise data from a more extensive source. Essentially, we aimed to develop a system that could use data showing a lot more information than what we had to work with. This could highlight more attributes about road accidents which is vital in research such as the type of collision, number of vehicles involved and the number of fatalities.

1.3 Approach and Structure of Report

The project was approached as software development project and was developed over two main iterations. Usability testing was conducted to determine both the effectiveness of the visualisation and the usability of the application in general especially the user interface.

Section 2 introduces a number of key concepts in the field of data visualisation as well as two projects done on visualisation of accident data. Sections 3 discusses the methodology that we used while Section 4 explains the implementation phase, that is, the development and of the application. Section 5 explains the results from the usability testing. Section 6 discusses findings from the entire project, such as the use of prototyping in the design of visualisations and contrasts it with the research done by previous work. The paper ends with a conclusion in section 7.

2. LITERATURE REVIEW

2.1 Data Visualisation Design

Information Visualisation, more commonly known as Data Visualisation, is the presentation of data in a graphical format [3]. Visualisations can be used to identify trends and patterns that would otherwise be unclear [3, 6]. Data visualisations can also be used to generate a new hypothesis or verify one [17].

2.1.1 Design and Framework

The Visual Information-Seeking Mantra, coined by Ben Shneiderman, is a good starting point for the design and development of data visualisation applications [24]. This mantra consists of four main principles namely; overview, zoom, filter, and view details-on-demand [24]. Shneiderman explains that a user should be able to get an overview of the entire data set (overview) and zoom in on particular subset of the data (zoom) [24]. Users should also be able to remove any unwanted data based on the data attributes (filter) as well as get the details of a data point when needed (details-on-demand) [24].

Yi et al. [27] presented similar findings in their research on interaction in Information Visualisation (Infovis) systems. They mention seven different categories that one ought to consider while designing an interactive Infovis application. These categories are; select ("mark a data item"), explore ("view something else"), reconfigure ("use a different arrangement"), encode ("show a different representation"), abstract/elaborate ("show less/more detail"), filter ("show something conditionally"), and connect ("show related items"). A key takeaway from both these papers is that different formats can and should be used to show the same data. This is due to the fact that different graphical formats make it easier to grasp different concepts and identify different patterns.

In his research on Visual Design, Treinish [25] dispelled the notion that a generalised mechanism can be used to address the diversity of visualisation strategies. He introduces the concept of task-based visualisation and the required steps which are:

1. Defining the application in terms of user needs
2. Selecting interface elements and designing actions to implement the definition
3. Establishing different techniques for various user goals

Lastly, Wassink et al. [26] also supported the view that the users should be a key factor in the design of scientific visualisations. They argue that it is important to analyse what kind of visualisation and interaction techniques best fit the user group [26].

2.1.2 Data Representation

Geo-spatial data, a key aspect of this project, is data that contains a physical location [13]. Nollenburg [20] described geo-spatial visualisations as the use of visual representations to understand geo-spatial data. Representation of geo-spatial data should include everything such as land masses and water bodies among others to give users enough context [13]. Google Maps, a common web mapping service, is a good option for geo-spatial visualisations as it provides a lot of information as well as five different views, that is, Street View, Traffic, Map, Satellite, and Hybrid [28]. The most conventional way of representing a location on Google Maps is through the use of a marker icon [13].

The choice of which graph or visualisation is used greatly depends on the nature of the variables of the data points (the dimensions of the data) [6]. This notion is emphasised by Knight [18] in her research on visualisation effectiveness who stated that the structure of a data set is vital to the selection of the graphical format used to represent it. Most of the data visualisation done today comprises multidimensional data with some of the dimensions being hierarchical such as time (which can be broken down into years, quarters and months etc.). Another factor one ought to consider while choosing a graphical format or visualisation is the message being conveyed to the users [6]. An example of this is the use of line graphs to show a trend over time [6].

2.1.3 Interactivity of Data Visualisations

Interactive visualisation takes the concept of information visualisation even further, enabling users to change the data that they see in order to identify more trends and patterns in the data [3]. Interaction with the data enables users to dynamically change the visualisations according to their objectives and visual queries [17]. Some of the principles of the Visual Information-Seeking Mantra such as zoom and filter cannot be adhered to without users interacting with the data. Interactivity is vital to gaining insight into big data today as the data is often multivariate (contains more than one variable) and the data sets are increasingly very large [12]. Furthermore, interactivity can be used to reveal more details about the data that cannot shown with the overview [3].

Visual clutter, caused by having too much data on a small display area, diminishes the potential usefulness of a visualisation [12]. By interacting with the data, users can reduce on the amount of data on the display area to uncover patterns and trends within overcrowded displays. Ellis et al. [12] discusses the concept of clutter reduction and suggested a few criteria for clutter reduction including:

- Keep spatial information (maintain the geographical location of the data points)
- Show details of a data point
- Scalability (clutter reduction techniques should be able to cope with large data sets)

2.2 Data Visualisation Evaluation

According to Knight [18], a visualisation can only be considered effective if users can answer their visual queries as early and as easily as possible. Furthermore, Knight [18] stated that the effectiveness of a visualisation is considered from two main perspectives; the suitability of the graph or visualisation for the tasks it is intended to support and whether the representation is appropriate for the data set [18].

A key issue surrounding the evaluation of visualisations is the interaction between the usability of the interface and the effectiveness of the actual visualisation [18]. A visualisation could answer all the visual queries conclusively but a poor user interface design could hamper its effectiveness. The evaluation of visualisations should consider both the usability of the interface as well as effectiveness of the visualisation.

2.3 Related Work

Inquiron, a data visualisation company based in Dubai, designed a visualisation (mapsdata) for car crashes in the UK [4]. Using data from the UK government, this visualisation tool uses four different maps to display the data:

- Marker map - Displays the precise location of each entry and shows information about it
- Heat map - Shows the concentration of data points in every location of the map using different colour gradients
- Cluster map - Shows the concentration using numbers instead of colour gradients
- Bubble map - Displays an extra value in addition to the location of each data point e.g. the number of cars involved in an accident

This visualisation does not provide much interaction with the exception of the Bubble Map. Therefore, users cannot change the data that they are viewing. The lack of interactivity makes it difficult to identify patterns in the data, as users cannot change the data to view a specific subset of the accidents based on the attributes, such as the time of the accident. However, it is easy to pinpoint potential black spots in the UK and prevent further accidents from happening at these locations.

The Organisation for Economic Co-operation and Development (OECD) regularly collects data in its bid to promote policies that will improve the economic and social well-being of people around the world. OECD designed a visualisation using the data they collected between 1947 and 2014 [2]. This representation does not display geo-spatial data but rather uses an interactive map to show the number of accidents that occurred in different countries. This visualisation uses three representations to present the data i.e. a line chart, a map and a table with each serving a different purpose. The line chart displays the number of accidents per year, showing how the trend has changed over time while the map shows the number of accidents represented by a "bubble" in each country. The size of the "bubble" is vital to the visualisation. The bigger the bubble, the higher the number of accidents and vice versa.

The foregoing discussion of the two visualisations shows the importance of using more than one representation to visualise the same data set. Different graphical formats / visualisations can be used to identify different patterns and trends in the data. The examples also highlight the importance of other concepts, such as colour and size. An example of this is the Bubble map mentioned in the first example [4], that uses the size of a bubble to show the number of vehicles involved in an accident and the heat map that uses different colour gradients to show different levels of concentration of data points.

3. METHODOLOGY AND EVALUATION

The main aim of this project was to develop a web-based application that can visualise geo-spatial data being collected by two Android-based application that detect car and running accidents. Therefore, the project was approached as a software development project with Human Computer Interaction (HCI) being a major aspect as the user interface is key to any visualisation. This section explains the methodology used and the requirements analysis conducted.

3.1 Approach

The project used both a User-Centred Design (UCD) approach along with an iterative methodology for development. A UCD approach, one in which users are involved throughout the design of the visualisation, ensures that the final design addresses all of the users' needs [26]. In this approach, the end user of the application/system along with the functional requirements are the starting point of the design process [11]. Wassink et al. [26] proposed a UCD approach for designing visualisations that consists of three phases; the early envisioning, the global specification phase and the detailed specification phase. The early envisioning phase consists of gathering requirements and definitions of tasks. During the last two phases, the global and detailed specification phases, solutions are presented to users and feedback is received [26]. The main users of the application we developed were researchers at CSIR who needed to gain more insight in their research on road safety in South Africa. The project was developed in multiple iterations which have been summarised into two main iterations, discussed in section 4.2.

3.2 Requirements Analysis

Functional requirements of a system define what a user can do with the system, while the non-functional requirements are usually classified as system qualities, such as reliability and usability [14]. With no prior system in place, we did not have to investigate an existing system that required improvement but rather design a new from scratch.

Thus, an set set of functional requirements were required before the design and implementation could commence. All the requirements were specified by the external supervisor, Mr Francois Mouton, a CSIR representative. The main requirement was to map GPS locations (i.e longitude and latitude) to Google Maps. The key functional requirements, highlighted the need for a web-based application with a usable user interface that would efficiently display the data collected by the Android applications. These requirements are:

- Authenticating users (to maintain the privacy of the data)
- Showing the location of an accident on Google Maps using the GPS location
- Showing all the relevant information about an accident
- Displaying statistics to help identify any patterns and trends in the data

The non-functional requirements, not specified by the CSIR representative, were based on research done by Chung et al. [10]. The key non-functional requirements for this application are scalability, reliability, usability and data integrity [10].

4. IMPLEMENTATION

4.1 Development Framework

This subsection covers the programming language and framework used in the development of the application.

4.1.1 Programming language and framework

Development of the web application was done using the Vaadin framework [2], an open-source Java web framework. On the server side, the framework is optimised for productivity and encapsulates all the scripting languages required for web development such as Java-Script. Furthermore, it is compatible with a number of web servers including Tomcat [3] and Jetty [4]. Java was chosen as the development language because of its numerous libraries that are easy to use. Using Java for development enabled us to use the Object-Oriented Programming paradigm which was advantageous for two reasons. Firstly, each accident could be stored as an object with its details stored as the variables. Secondly, some of the code was re-used through inheritance. Maven [5] was used for building and managing the project while Github was used for version control.

4.1.2 Integration

The web server hosting the application and the database (MySQL) from which information is pulled are on the same server. This made integration with the data source much easier. Java Database Connectivity (JDBC), a Java Application Programming Interface (API), was used to connect to the database.

[2] https://vaadin.com/home
[3] http://tomcat.apache.org/
[4] http://www.eclipse.org/jetty/
[5] https://maven.apache.org/

4.2 Software Development Iterations

The development of the application was done using an iterative methodology which involved the end users as much as possible. This is discussed in detail in the following subsections (sections 4.2.1 and 4.2.2).

4.2.1 Iteration 1: System Design and Early Envisioning

The main focus of this iteration was to get a clear understanding of the system requirements, design a few prototypes and have a software deliverable ready for the initial feasibility demonstration.

Based on the system requirements and the selected framework (Vaadin as earlier discussed), we decided on a multi-layered architecture for the system. This architecture consisted of three layers namely; the Presentation layer, Application layer and Data Access layer. The Presentation layer consists of classes and methods used to develop the user interface while the Application layer consists of Java objects to represent the accident data as well methods to filter and change the data. Lastly, the Data Access layer provides methods to connect to the database and get the data. Essentially, these three layers can be described as the back-end (application and data access layer) and the front-end (presentation layer) of the system. A multi-layered architecture was used because it maximises cohesion and minimises coupling among the different modules of the system. The use of this architecture also enabled us to test the different components independently. The system design (i.e. a multi-layered architecture) also inadvertently ensured that we had a separation of concerns. Separation of concerns, a fairly, new paradigm in software engineering, tries to separate the different algorithms based on their purpose [15]. This separation allows for the locality of different kinds of information in the programs, making them easier to write, understand, reuse and test [15].

With a clear system design in mind, we researched into different data visualisations and looked at applications that had been developed with the Vaadin framework. Based on our findings, we designed a few low fidelity prototypes. These prototypes were not evaluated by the end users as is the norm but were rather used to give us ideas of the visualisation and user interface design. Low fidelity (lo-fi) prototypes were used because of their simplicity [23]. Lastly, lo-fi prototypes ensured that we could go through as many different designs as fast as possible.

This iteration concluded with the initial feasibility demonstration. The deliverable at this stage consisted of an interface displaying Google Maps and an icon showing the location of GPS coordinates (longitude and latitude) entered by a user. At this point, there was no major feedback as the visualisation did not show much and was not interactive.

4.2.2 Iteration 2: Global Specification

In the global specification phase, solutions are presented to end users [26]. This phase (iterations) comprised many short iterations in which solutions were presented to users (supervisors) and a few students.

Prior to the implementation phase of this iteration, more requirements analysis was carried out to find out what kind of data would be stored about each accident and how best it could be represented. Shniederman's [24] Visual Information-Seeking Mantra, discussed in section 2.1.1, was applied when designing the visualisation. Another key aspect considered during this iteration was the navigation through the application from one page (interface) to another. This is vital, as a poor and unintuitive navigation can lower the usability and effectiveness of a visualisation application.

Based on the requirements gathered, we deemed it necessary to use tables to show the data; one table showing the car accidents and another showing running accidents. Each row in the table represents an accident. A user can interact with the table to get more details about an accident. The location of each accident is also represented on Google Maps. Users can zoom in on a particular subset of the data and filter out any data they are not interested in.

Informal usability testing was conducted during this iteration. Two students were observed as they used the application. They were not given any specific tasks but rather did a walk-through of the application to identify any problems with the user interface. Demonstrations were done for both supervisors and changes were made based on their feedback.

4.3 Final Visualisation Design

Figures 1, 2 and 3 show the final visualisation design, a culmination of the two discussed iterations.

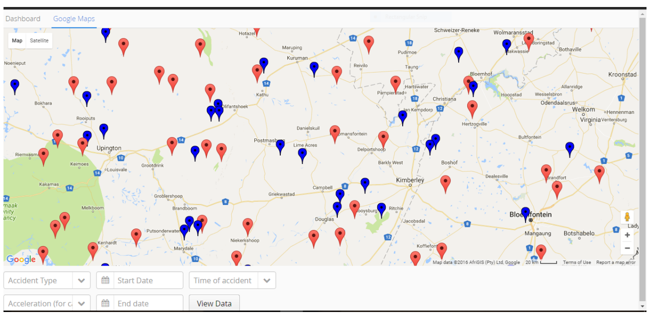

Figure 1: A screen-shot of the Final Visualisation Design: Google Maps

On the Google Maps page (Figure 1), users are presented with an overview i.e all the data points in the data set are shown. Hovering over a single data point displays all the relevant information about the accident such as the date and time of the accident. The drop down menus at the bottom of this page allow a user to select which data they would like to see, filtering out the rest. This is based on the different dimensions of the data. Users can select to see one type of accidents i.e. only car accidents or running accidents. Google Maps also has a zoom feature which enables users to zoom in and out of particular regions.

The table in Figure 2 shows all the data in the database. These tables can be sorted by any of the columns, enabling a user to view the accidents in order of occurrence or acceleration at impact (only for car accidents). Right clicking on a row in the table gives users options to view any details not shown in the table and the location of the accident on Google Maps (as demonstrated in Figure 2).

Two distinct symbols were used to represent the two types of accidents on the map; car accidents are represented by red icons while the blue icons represent running accidents. These symbols differed on only one feature, that is, colour because the marker icon is the common way of representing locations on Google Maps. This is different from items on

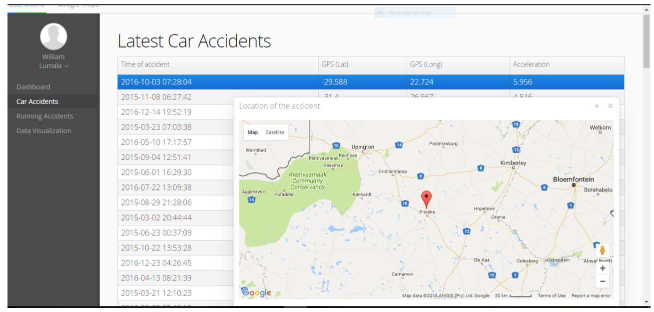

Figure 2: A screen-shot of the Final Visualisation Design: The tables

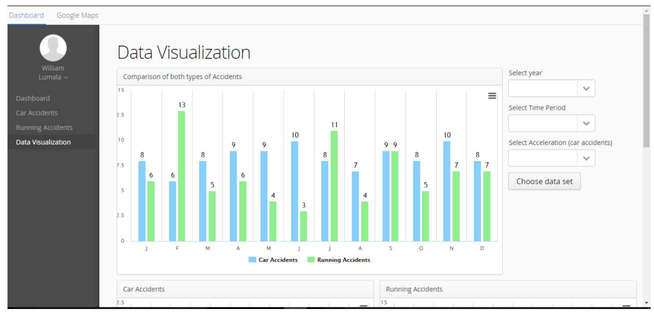

Figure 3: A screen-shot of the Final Visualisation Design: Data Visualization Section

the line chart (not in image) which differed by two features, that is, colour and shape. This was done to make the items on the line chart more distinguishable from each other.

5. FINAL EVALUATION AND RESULTS

5.1 Evaluation Metrics

Knight [18] suggested that the whole package of a visualisation application (i.e the user interface of the application and the actual visualisation) has to be considered in the evaluation of data visualisations. Therefore, the evaluation of a visualisation application should cover both the usability of the interface and the effectiveness of the visualisation. Users' subjectives feedback were used to determine the effectiveness of the visualisation and the usability of the application in general.

5.1.1 Effectiveness of the visualisation

Effectiveness, with regard to visualisations, focuses on the cost-benefit of using the visualisation to locate information [9]. Knight [18] had a similar proposition as she stated that a visualisation can only be considered to be effective if users can achieve results as easily as possible.

Visual queries, associated with the user goals laid out in the requirements, were set up. Users were asked to answer the visual queries using the visualisation. This tested a variety of issues including; the efficiency (how fast and easily they got an answer) of the users, the correctness of the user responses (whether users can extract the right information) and whether the user interface was easy to use. Users were also asked to complete a set of tasks using the system. Essentially, using the visual queries and tasks, we evaluated both the effectiveness of the visualisation as well as the usability of the user interface.

5.1.2 Usability

Usability in the context of a user interface, assesses how easy it is to use an application. Nielsen [16] suggests that usability is defined by five components; learnability (how easy it is for users to complete tasks the first time), efficiency (how quickly users perform tasks), memorability (how easy it is to use the design after not using it for a while), errors (how many errors users make), and satisfaction (how pleasant it is to use the design).

John Brooke [8], in his research on usability, suggests that the usability of a user interface should cover efficiency (the level of resource consumed in performing tasks), effectiveness (the ability of users to complete tasks coupled with the quality of the output of those tasks), and satisfaction (users' subjective reaction to using the system).

During the usability tests, users were observed as they interacted with the system. The users performed a set of predefined tasks and performance measures including success rate and task completion were collected. The tasks for the test, as explained in section 5.2, were based on the key features of the system.

The System Usability Scale (SUS) [8] was used to gauge the users' satisfaction and opinions of the systems. The SUS, a questionnaire designed by John Brooke [8], is a reliable tool for measuring usability of a system [8]. It comprises 10 items and users' responses are rated on a Likert Scale which ranges from 1 ("Strongly disagree") to 5 ("Strongly agree") [8]. SUS measures two of the five usability components discussed by Nielsen [16], that is learnability and satisfaction; is independent and can be used on a variety of applications [22]. Furthermore, the scale (SUS) provides valid results on small user sample sizes [8]. These two factors make the SUS a good choice for evaluation of the project, where there was limited access to users. The ten items on the SUS questionnaire are in the appendix of the paper.

5.2 Usability Tests

5.2.1 Participants

Convenience sampling, a type of non-probability sampling, was used to select 6 of the 8 participants while the other 2 were selected randomly. Non-probability sampling was used to enable us to get participants that fit a specific criteria, that is, Computer Science Honours students who had done both the Data Visualisation and HCI modules. This was done to maximise our chances of getting informative and extensive feedback about both the User Interface (UI) and the effectiveness of the visualisation while also getting some feedback from novice users.

5.2.2 Apparatus

Most of the tests were conducted in a controlled environment of the Computer Science honours laboratory. Participants accessed a web-page that contained the visualisation using a laboratory computer. Two of the tests were conducted in a university residence where users accessed the visualisation on their laptops. Data was collected during the experiment by observing users as they completed each task and taking down notes.

5.2.3 Procedure

Participants were given a brief overview of the system, its purpose and the goal of the project. They then completed three tasks and used the system to answer two visual queries. These tasks and visual queries are explained in the Section 4.3. These tasks were essentially a walk-through of the key functionality. This allowed the users to use visualisation before they completed the System Usability Scale (SUS) questionnaire. Users were observed as they completed the tasks. While this has its downsides e.g. users could get intimidated, it enabled us to see their reaction and get immediate feedback about any issues they had with the system.

5.2.4 Task Design and Visual Queries

The tasks and visual queries for the usability testing were designed with two main objectives in mind. Firstly, we wanted to establish whether users could effectively find a subset of the data i.e could users zoom into an interesting data and / or filter out unwanted data (essentially, testing the interactivity). Secondly, we wanted to determine how efficiently users could answer a visual query. Furthermore, these tasks were designed to give the users a feel of the entire system before completing the SUS questionnaire. The two visual queries were:

1. Which month had the highest number of running accidents in 2016? View these accidents on Google Maps.
2. Which month overall (over all the years) had the highest number of car accidents? I.e. what is the most dangerous month?

The users were also asked to complete the following four tasks:

1. Display all the locations of car accidents that occurred in Dec 2016 with an acceleration value of 20 mph and above (On Google Maps).
2. Display all the accidents that occurred from 1st April 2016 to 30th August 2016, between 6 pm (18hrs) and midnight.
3. Using the car accidents table, view the location of any one of the accidents
4. Display/view the details on the any accident on the map (using Google Maps).

5.3 Analysis of Usability Testing

5.3.1 Visual Query Accuracy and Task Completion

Successful task completion is defined as the ability of a user to obtain information when carrying out a task [19]. Participants who obtained the right answer were deemed to have successfully answered a visual query, while those who needed more than one trial to get the correct answer were deemed to have partially passed the visual query. This is a slight variation of the usability metric proposed by Nielsen [19]. Nielsen [19] stated that the success measure categorised into three groups; successful, partially successful and failed. The same metric was used for the analysis in the four tasks. Tasks completed on the first trial without help from us were categorised as successfully completed and the rest were categorised as failed. All the participants eventually completed the visual queries and task. This was vital as it helped us to understand which tasks were easier to learn and more intuitive to users.

Figure 2 shows the percentage of the two visual queries successfully answered, that is, at the first time of asking and with no help from the team members.

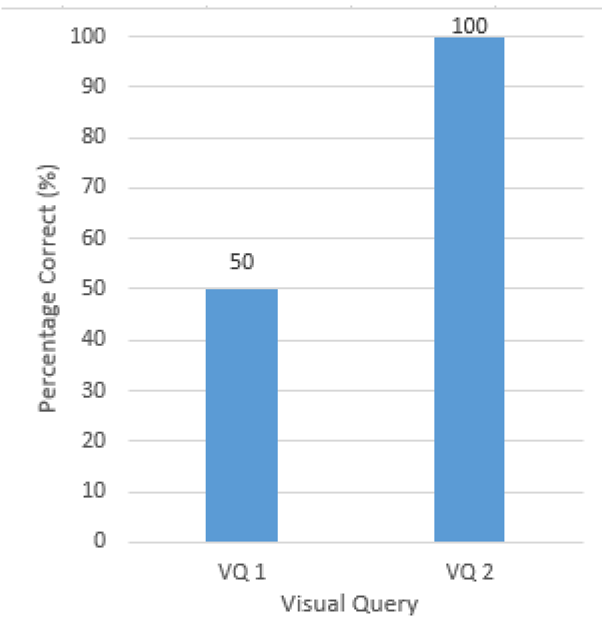

Figure 4: Successfully answered Visual Queries

Results showed that only half of the participants answered the first visual query correctly while all the eight participants answered the second one correctly. One possible reason for this could be that the second visual query is a simple variation of the first one. When examining the results of visual query 1 and the feedback from the users, two of the four participants who gave a wrong answer for visual query 1 explained that the instructions were not clear and that they required a further explanation. The other 2 participants stated that they were not sure which data was displayed when a user logs in initially. The application had been designed to show an overview of the data set before users could change the data to suit their needs. Visual query 1 also involved locating data on the map after retrieving an answer from the graphs. All of the participants clicked on the graph with the expectation that they would be redirected to the Google Maps interface. However, they had to manually go to the Google Maps (Figure 1) and then locate the data by filtering out the unwanted data. The navigation through the web application was a major point in the user feedback as discussed in Section 5.3.3.

Figure 3 shows the percentage of the four tasks completed i.e. which tasks were completed successfully.

Both tasks 1 and 2 involved filtering out some data-points based on their variables (dimensions). Therefore, a user had to select a date, time, type of accident among other variables (Figure 1). As shown in the graph above, half of the eight participants successfully completed task 1, while the other half made the same mistake, that is, they left some of the input controls (UI elements for data entry) empty. This prompted the system to notify them that they had not filled

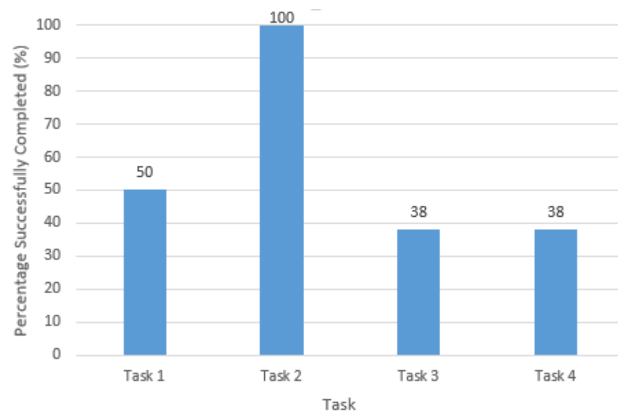

Figure 5: Successful Tasks Completed

in some information. As task 2 was unquestionably similar to task 1, there was an improvement in the task completion rate with all the participants successfully completing this task at the first time of asking. This showed that the participants performed well with the learnability component (metric) of usability described by Nielsen [16].

Based on the results, tasks 3 and 4 were significantly more difficult than the first two tasks. Task 3 involved using the table to get more information about an accident (Figure 2). Only three of the participants successfully completed this task three. Four of the other five participants were only able to complete this task once they had read the instructions on the homepage and one needed further explanations. Upon further analysis of the feedback, we discovered two possible reasons to explain why more than half of the users could not complete this task. Firstly, there was no visual cue to signify that users could interact with the data in the table. Secondly, most of users of web applications normally left-click on an element or item to get more information. This was evidenced in the usability testing as a number of users left-clicked on a row in the table (Figure 2) to get more information about a data-point (an accident). Right-clicking on a row in the table reveals more information, such as the location on Google Maps while left-clicking simply highlights the row. Both of these effects are shown in Figure 2.

We settled on the lack of visual cues as the possible reason because most of the users gave positive feedback about the number of clicks required to locate information and stated that the UI was simple and easy to use. Furthermore, users did not have to navigate from one page to another to complete this task and stated that they could not have known that they had to right-click.

Task 4 involved simply hovering over a data-point on the map to show all the relevant information about it. This task had a similar completion rate to task three with only 3 participants completing successfully. All the participants (including the three that completed the task successfully) initially clicked on the icon. This was also put down to the lack of a visual cue, as there was nothing on the UI to suggest to users that could get more information by hovering over an icon on the map. Feedback from the users that showed a majority of them expected to click on an accident to reveal further information and that although hovering over a data-point was also a familiar method to them, clicking was more intuitive.

Based on the all results mentioned above, the visualisation is effective as users are able to easily obtain correct information. Some of the users had to read the instructions but they eventually got the right information. The time taken per task was not recorded but the participants did not take more than 2 minutes on completing any of the tasks. This is deemed to be efficient especially since they were using the system for the first time.

5.3.2 System Usability Scale Score

The 10 items on the SUS questionnaire are in the appendix of the report. Each participant's SUS score was calculated as follows; odd numbered questions (positively phrased) are calculated as the response minus one while even numbered questions (negatively phrased) are scored as five minus the response [8]. The sum of the score is multiplied by 2.5 to obtain the overall value of System Usability. This score was obtained for each user to get an average of 78.75. [7]

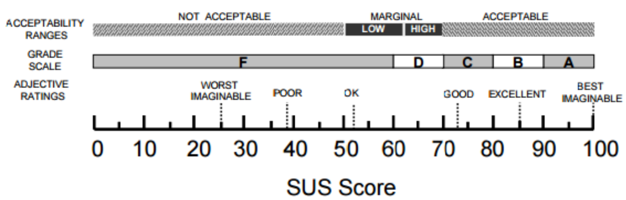

Figure 6: Interpreting SUS scores by comparing them to an adjective-based scale [6]

Based on the scale shown above, the usability of the product developed can be classified as good and is of an acceptable standard.

5.3.3 User Experience Feedback

The SUS method simply determines the usability of a system but does not reveal the usability problems. Free-response questions were appended to the System Usability Scale questionnaire to determine what usability problems participants experienced during the test. However, it was not compulsory for users to fill in these fields. Furthermore, users were encouraged to give feedback about the system as they were using it.

The major feedback received from the participants was to do with the input controls. There was both positive and negative feedback. Most users were happy with the drop down menus. This ensured that they knew all the options and simply had to select one. Furthermore, the use of drop down menus instead of other UI elements such as text-fields made it easier to input data. However, users would like to be notified about any compulsory fields that they have to complete before confirming.

One user highlighted the use of colour in the visualisation. The graph and the Google Maps interface used different colours to represent the same kind of information. It was pointed out during the usability testing that the use of a uniform colour throughout the entire visualisation would make comprehension easier.

The lack of a legend was highlighted by one of the users as it was not clear which icon represent which set of accidents. However, most users, upon completing task one were sure what each icon represented.

Lastly, an issue discussed by a majority of the users was the navigation through the application. Some of the tasks involved moving from one UI to another. This involved recalling information and users suggested that they should be able to navigate through the application without having to specifically remember anything.

6. DISCUSSION

This section explains the type of prototyping used and how a different approach could have been used. It also discusses the visualisation; how it meets the various principles and its limitations. The limitations of the usability testing are also highlighted.

6.1 Prototyping

As explained in section 4.2.1, paper prototypes were used in the design phase before implementation was done. While these prototypes were not evaluated by potential users as is the norm in software development, they still provided a good idea on how to proceed. Paper prototypes were used because at this stage we were more interested in the functionality of the visualisation rather than the design. One key advantage of using paper prototypes is that they are easy and fast to develop and do not require the use of any software [21]. Furthermore, the fact that it is easy to modify paper prototypes ensured that the cost of mistakes was low and we could get through a number of variations within a short time.

A significantly different approach to the prototyping phase would have entailed the use of high fidelity (hi-fi) prototypes. Hi-fi prototypes have a number of drawbacks such as the fact that most users tend to comment on the fit and finish issues i.e [21]. In hindsight, hi-fi prototypes, would have been valuable to this project despite all the drawbacks. Hi-fi prototypes show the UI elements and the spacing, that is, as close as possible to a true representation of the user interface [5]. They are also assumed to be much more effective in collecting true human performance data, such as task completion time [5].

Ben Shneiderman [24], in his research on Data Visualisation, does not mention the importance of the HCI aspect in data visualisation. Yi et al. [27] also neglected this aspect in their research on the role of interaction in information visualisation. The HCI aspect is vital in data visualisation because interacting with the visualisation through the interface is key to uncovering the hidden patterns as well as changing the data. Designers of visualisations should at least focus on three key aspects, that is, the tasks and purpose of the visualisation, key principles of data visualisation such as the Visual Information-Seeking Mantra [24], and the design of the UI.

6.2 Data and Visual Representation

The final visualisation design in this project covers the four main principles described by Ben Shneiderman in the Visual Information-Seeking Mantra [24]. The interactive nature of the visualisation allows users to change the data dynamically, filtering out any unwanted data points. Google Maps allow users to zoom into particular geographic regions. One advantage of this feature is that visual clutter in a small area of display can easily be overcome.

One of the key principles mentioned by Yi et al. [27] is connect ("link related data points"). About accident data, this would entail showing accidents with similar attributes when a user select an accident. This feature is not included in the visualisation as a user cannot click on an icon on the Google Maps interface. This is one of the major limitations of the final visualisation design.

The final visualisation design also lacked visual cues as mentioned in Section 5.3. This made it more difficult for participants to learn how to use some features of the application. Learnability, one of the five components discussed by Nielsen [16], is characterised as how easy it is for a user to complete a task the first time. Based on the results analysed in Section 5.3, users easily learnt how to use the system as all of them successfully completed the tasks that were variations of previous tasks. This was good especially given the lack of visual cues.

6.2.1 Comparison to prior work

In this section, we compare and constrast the visualisation developed on this project to the MapsData [4] visualisation discussed in Section 2.3.

The data used for the final visualisation design of this is multidimensional, where each data point contains more than one attribute (dimension). The attributes include the GPS coordinates (latitude and longitude), day and time of the accident, and the acceleration at impact for car accidents. When compared to mapsdata, a similar visualisation, there are two key differences. Firstly, mapsdata is not interactive which makes it difficult to identify any hidden trends in the data, while our visualisation is interactive. Secondly, the data used for the MapsData visualisation contains more dimensions such as the location, time, weather conditions, number of vehicles and number of causalities, which provides a lot of insight.

Both visualisations use more than one representation to present the same data. The MapsData [4] visualisation uses four different maps to show the same data, that is, a Marker map, a Heat map, a Cluster map, and a Bubble map while the project visualisation uses Google Maps, a table and graphs (bar charts and a line chart). The different representations cannot be compared, but they all add value to the two visualisations.

6.3 Limitations of Usability Testing

Although the experiment tried to use participants with as much knowledge about data visualisation as possible, they were not a true representative of the final users of the application. Furthermore, heuristic evaluation of the application could not be done because the project did not have access to any experts.

The usability tests conducted consisted of only eight participants to ensure that the tests were completed within a short time. Only eight participants were used for usability testing due the the instability at the university at the time. While the SUS is designed to cater for a small number of participants [8], getting feedback from more users could have provided more insight into the usability problems.

Lastly, the visual queries and tasks were as short as possible to ensure that users could complete the entire test within 15 minutes. This was done based on seeing users' reactions to similar experiments that took 30 or more minutes. Similar to the point mentioned above, giving users more tasks would have provided better feedback.

6.4 Design Methodology of Visualisations

In this subsection we propose a new framework / methodology for designing data visualisations. This methodology is based on the literature we reviewed as well as some findings from the project. We propose an iterative methodology for the development of software that is used to visualise data. This iterative methodology contains 3 main phases; 1) Requirements Gathering and Design 2) Prototyping and Evaluation, and 3) Implementation and Testing. Designers do not have to particularly go through the second and third phases more than once unless it is necessary. The three phases are further discussed in the subsequent subsections.

6.4.1 Requirement Gathering and Design

This phase entails getting and understanding the functional and non-functional requirements of the visualisation application. Understanding both the functional and non-functional requirements is vital in the development of software. Treinish [25] in his research on task-based visual design, highlights the importance of understanding the end users of the visualisation and the tasks that they would like to complete with the system (visualisation application). This information is vital, as the background and experience with technology of potential users should be considered when developing interface-based applications. With the users and requirements in mind, a system architecture and design can be produced. Essentially, this phase involves finding out as much information as possible about the system and the users and then designing the system.

6.4.2 Prototyping and Evaluation

This phase entails designing and evaluating prototypes. Although lo-fi prototypes are cheaper and simpler to design [21], hi-fi prototypes are recommended. This is because interactivity with the user interface is vital to identifying different patterns in a data-set. Using Hi-fi prototypes will also ensure that the prototypes are as close as possible to the final visualisation design. Developing and evaluating prototypes with end users helps developers identify mistakes early on. It also gives the end users a chance to give their opinions and feedback on the system. Lastly, it is easier to make changes during this phase than in later phases. Depending on the results of the evaluation, more than one prototype can be developed.

Both the HCI aspect and the design principles for Visualisations, such as the Visual Information-Seeking Mantra [24] should be at the forefront during this phase.

6.4.3 Implementation and Testing

The final phase involves developing and testing the visualisation application with end users. This can done more than once depending on the results from the user testing and the feedback. At this point, developers should have a clear system and user interface design. Developing fully functioning prototypes makes this phase shorter and more efficient as a number of mistakes have been identified and resolved in the previous phase. Similar to phase 2, both the HCI aspect and the design principles for Visualisations, such as the Visual Information-Seeking Mantra [24] should be at the forefront during this phase. The methodology proposed is iterative, therefore allows for changes to be easily implemented. It also ensures that a visualisation application project is approached like a software project with the user interface being a key aspect.

7. CONCLUSIONS

In this paper, we presented the design of an interactive geo-spatial visualisation to present accident data in South Africa. The visualisation maps the GPS location of an accident (data point) onto Google Maps and also shows all the information captured about an accident. In comparison to similar visualisations such a mapsdata [4] (discussed in section 2.3), this visualisation provides interactivity, enabling users to dynamically change the data that they view. This is particularly useful when users want to zoom and filter the data. In contrast, mapsdata [4] shows much more information about each data point than this visualisation. More information about the accidents could provide more information about why different types of accidents or collisions occur at different locations.

An interactive and user-centred design approach was used in the development of this application. The final visualisation was a culmination of two prior iterations. In the evaluation of the visualisation, we evaluated the effectiveness of the visualisation's ability to provide insight into accident occurrences. Users were given visual queries to answer and tasks to complete. The correctness of the visual queries was used to determine the effectiveness of the visualisation. Based on the results discussed in section 5.3, this visualisation was found to be effective at presenting data.

Usability tests also highlighted the importance of the UI in any visualisation. The interface is important, as human-computer interaction is vital to getting more information from a visualisation. Both sets of principles described by Yi et al. [27] and Ben Shneiderman [24] focus on the design of the visualisation and do not mention the importance of designing a good user interface. A better UI makes the visualisation more effective, enabling users to attain information more efficiently.

We also proposed an iterative methodology for the development of visualisation applications. This methodology is based on the literature we reviewed as well as some findings from the project. This methodology comprises three main phases; 1) Requirements Gathering and Design 2) Prototyping and Evaluation, 3) Implementation and Testing. This methodology highlights the importance of the end users as well as the user interface of the visualisation application.

8. FUTURE WORK

Visualisation of accident data is a novel way of presenting accident data in South Africa. This visualisation can be improved on in several ways. One avenue for future work would entail using official accident data from the South African government including attributes such as number of fatalities and the number of cars involved in an accident. Visualisation of more extensive data would reveal more insight and patterns that would be helpful in the research on road safety in South Africa.

Another avenue for future would involve adding a feature to enable users to link similar accidents. Essentially, a user would select one accident and then view all the similar accidents. This similarity would be based on the dimensions of the data. For example accidents that occurred in the same time or with the same acceleration value could be displayed depending on the data attributes and the algorithms.

9. ACKNOWLEDGEMENTS

I would like to thank my project partners, Luke Bell and Gavin Wiener, for their help, as well as Francois Mouton (from the CSIR) for providing valuable input and feedback throughout the course of this project. Finally, my sincere appreciation is extended to my project supervisor, Professor Tommie Meyer, for his constant guidance, support, and sound advice.

10. REFERENCES

[1] Arrive alive : Accident crash statistics. https://arrivealive.co.za/Accident-Crash-Statistics. Accessed: 2016-10-15.

[2] Data: Road accidents. https://data.oecd.org/transport/road-accidents.htm. Accessed: 2016-10-09.

[3] Data visualization (what it it and why it) matters. http://www.sas.com/en_us/insights/big-data/data-visualization.html. Accessed: 2016-10-15.

[4] Mapping car crashes in the uk. http://www.mapsdata.co.uk/portfolio-items/traffic-accidents-uk/. Accessed: 2016-10-09.

[5] Prototyping. https://www.usability.gov/how-to-and-tools/methods/prototyping.html. Accessed: 2016-11-05.

[6] Visualize data. http://betterevaluation.org/en/plan/describe/visualise_data. Accessed: 2016-10-15.

[7] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.

[8] John Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[9] Stuart K Card, Peter Pirolli, and Jock D Mackinlay. The cost-of-knowledge characteristic function: display evaluation for direct-walk dynamic information visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 238–244. ACM, 1994.

[10] Lawrence Chung and Julio Cesar Sampaio do Prado Leite. On non-functional requirements in software engineering. In *Conceptual modeling: Foundations and applications*, pages 363–379. Springer, 2009.

[11] OMF De Troyer and Corneli Jan Leune. Wsdm: a user centered design method for web sites. *Computer Networks and ISDN systems*, 30(1):85–94, 1998.

[12] Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics*, 13(6):1216–1223, 2007.

[13] Stephen Few. Introduction to geographical data visualization. *Visual Business Intelligence Newsletter*, pages 1–11, 2009.

[14] Daniel Gross and Eric Yu. From non-functional requirements to design through patterns. *Requirements Engineering*, 6(1):18–36, 2001.

[15] Walter L Hürsch and Cristina Videira Lopes. Separation of concerns. 1995.

[16] Nielsen Jakob. Usability 101: Introduction to

usability. https://www.nngroup.com/articles/usability-101-introduction-to-usability/. Accessed: 2016-10-28.

[17] Daniel A Keim. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[18] Claire Knight. Visualisation effectiveness. In *2001 International Conference on Imaging Science, Systems, and Technology (CISST 2001)*, pages 249–254. Citeseer, 2001.

[19] Jakob Nielsen. Usability metrics. https://www.nngroup.com/articles/usability-metrics/, 2001. Accessed: 2016-11-11.

[20] Martin Nöllenburg. Geographic visualization. In *Human-centered visualization environments*, pages 257–294. Springer, 2007.

[21] Marc Rettig. Prototyping for tiny fingers. *Communications of the ACM*, 37(4):21–27, 1994.

[22] Jeff Sauro. Measuring usability with the system usability scale (sus). http://www.measuringu.com/sus.php. Accessed: 2016-11-11.

[23] Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. Paper prototyping-what is it good for?: a comparison of paper-and computer-based low-fidelity prototyping. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 778–779. ACM, 2003.

[24] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[25] Lloyd A Treinish. Task-specific visualization design. *IEEE Computer Graphics and Applications*, 19(5):72–77, 1999.

[26] Ingo Wassink, Olga Kulyk, Betsy van Dijk, Gerrit van der Veer, and Paul van der Vet. Applying a user-centered approach to interactive visualisation design. In *Trends in Interactive Visualization*, pages 175–199. Springer, 2009.

[27] Ji Soo Yi, Youn ah Kang, John Stasko, and Julie Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231, 2007.

[28] Jingyuan Zhang and Hao Shi. Geospatial visualization using google maps: a case study on conference presenters. In *Computer and Computational Sciences, 2007. IMSCCS 2007. Second International Multi-Symposiums on*, pages 472–476. IEEE, 2007.

APPENDIX

The 10 SUS questions used in the evaluation are:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.