DEVELOPMENT OF A HOMICIDE MEDIA ANALYSIS TOOL

Syed Mohammad Khizar Anas (2166010)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This report presents the development and evaluation of an open-source digital platform, the Homicide Media Tracker, designed to collect and analyse homicide-related data from media outlets. The platform aims to provide transparent insights into patterns and trends in homicide reporting. This investigation compares the two libraries used to make the dashboard, Dash library and the streamlit library. The platform's primary components include a data entry system, a robust data analysis engine, and a user-friendly dashboard for interactive visualizations. The complexity of the Dash API and the streamlit API was compared to check which is less complex and uses up less resources of a computer. The results show that the Streamlit library is less complex then the Dash library.

Key words: API, Dash, Streamlit, Dashboard, GUI.

1. INTRODUCTION

In South Africa, homicide is a significant issue, with over 27,000 murders recorded between April 2022 and March 2023 [1]. Humanitarian researchers aim to investigate the causes of these homicides, their frequency, locations within South Africa, and how the media portrays these incidents to the public, including identifying flaws in the media's reporting process [2]. Addressing these research goals requires a comprehensive approach to data collection, analysis, and visualization. To support researchers in this effort, the development of a homicide media analysis tool is essential.

Therefore, the project aims to develop a homicide analysis tool that allows humanitarian researchers and journalists to store homicide data from news reports and articles, analyse it, and visualise it. This is done by creating a GUI that allows researchers to enter data on homicide cases reported in news articles/newspapers and store it in a database. It then allows them to clean the already stored data and visualise it using different visualisation techniques. This tool is required to be free and open-source.

One of the main objectives of the project was to investigate if the GUI has a very low complexity and uses fewer computer resources to allow humanitarian researchers with low funds or no funds at all to access the tool and perform their respective research in homicide cases in South Africa. This is done by designing two GUI, both using different Python libraries (One GUI is built using the Dash library and the other is built using the Streamlit library) to test which GUI is less complex and more user-friendly while having the same functionality between the two. Moreover, the GUI will also be tested to check which one requires the least amount of resources to run on a computer. Both Dash and Streamlit are open-source Python libraries that are used to build web applications, dashboards and GUIs.

This report is split into different sections beginning with the `Background' section that presents information relating to this project. The `Methodology' details and justifies the techniques used in the project. The results of testing are presented in `Results'. The `Discussion' section explores the findings of the results, outlines the benefits and other uses of induction heating, and provides future recommendations. Finally, the report is concluded in the `Conclusion' section.

2. BACKGROUND

2.1 Homicide in South Africa

The South African Police Service (SAPS) and Stats SA have published the latest official crime statistics for South Africa. These statistics show that 12734 people were murdered from January to June 2024 [3]. This shows that an average of 70 murders a day have happened over the past 182 days [3].6536 people we killed from January 2024 to March 2024 and 6198 people were murdered between April 2024 and June 2024. This is worse than the same period in 2023 where 12517 were murdered, thus showing that 217 more murders were recorded in 2024.

2.2 Existing Homicide media analysis tools

This shows that there is an increase in homicides in South Africa and humanitarian researchers would like to research it and also visualise the correlation between different aspects of the homicide cases. To do this, they required a data collection tool.

United Nations has a data collection tool but it is not open source, researchers can not input their data into the tool and the data visualisation that is done is basic [4] [5]. Moreover, in South Africa, there is Crime Hub is a source of regularly updated, credible information and analysis on crime, violence, and the functioning of the criminal justice system in South Africa. Crime Hub was developed by ISS (Institute of Security Studies) which is an African organisation which aims to enhance human security on the continent.[6] It does independent and authoritative research, provides expert policy analysis and advice, and delivers practical training and technical assistance. Crime Hub has data visualisation, and data analysis tools built into it. Moreover, researchers can also see the correlation between different crimes happening in a municipality in South Africa. The problem with Crime Hub is that it is not an open source tool and it's not free. The data used in Crime Hub comes from the annual crime statistics and other public releases made by the South Africa Police Service (SAPS). Researchers can not input their collection of data into Crime Hub. As one of the important aspects of the project is to allow researchers to collect the data they have, therefore, Crime Hub can not be used.

2.3 Constraints

The constraints in the project were that the database and the GUI are built using free, open-source tools to allow any humanitarian researcher with no funds/low funds to be able to access the tool, input homicide cases into it, and analyse and visualise the data. Another constraint is that the homicide media analysis tool needs to be a low-complexity tool so that it uses the least amount of resources of the user's computer. This will allow a computer with very low specs to run the tool. This allows humanitarian researchers with low-spec computers to use this tool to research homicide cases.

Other constraints for the tool are that the homicide media analysis tools need to allow researchers to enter data and store them, it needs to be able to find the duplicates stored in the data and allow the user to delete them and be stored in a duplicates table. It also needs to allow researchers to delete the data present in the database. Moreover, it needs to be able to visualise the data using different visualisation techniques which would allow researchers to see the correlation between different variables in the data.

2.4 Required Software used

Due to the constraints, the database is created using PostgreSQL. As discussed above GUI is created using the Python libraries as Python programming language is commonly used in machine learning and data science therefore it has a lot of libraries for data manipulation, data visualisation, data cleaning and data analysis. Plotly library is used for data visualisation because it allows for easy implementation of interactive graphs.

3. EXPERIMENTAL SETUP

3.1 Database

A database called `homicide_main` was created on PostgreSQL. To store the data, a table had to be created on the database. A `main.py` file is created in which the `psycopg2` Python library is used to build a connection between the database on PostgreSQL and the GUI. After the connection was established, SQL commands were sent from the `main.py` file to PostgreSQL to build the table called `homicide_news`. `homicide_news` is the main table where all the data from the researchers is stored. The different fields in the data provided by Dr. Nechama Brodie as a CSV file were added to the table and the data from the CSV file was also added to `homicide_news`.

3.2 Graphics User Interface (GUI)

3.2.1 Dash GUI: The dashboard.py file creates the GUI for the tool. The psycopg2 library was used to establish and maintain the connection between the GUI and the `homicide_main` database. In the Dash library, the dcc and html components are used in the GUIs creation. Dash Core components (dcc) contain components like graphs, dropdowns, sliders, and more. These are interactive elements that you can use to build the GUI. The HTML component provides components that correspond to standard HTML elements like <div>, <h1>, <p>, etc. These are used to structure the layout of the GUI.A Navigation bar is used to show the different functions of the GUI. These functions are represented in *Figure 2*.

3.2.2 Streamlit GUI: The streamlit GUI uses the psycopg2 library to establish and maintain a connection with `homicide_main`. The Streamlit library does not use standard web page frameworks such as HTML, or CSS to build. It instead has built-in functions to display texts, images, videos, widgets, progress bars, status messages, sidebar, containers and display graphs. A sidebar is used to organise the functionality of the GUI.*Figure 2* shows all the functionality implemented in the GUI.

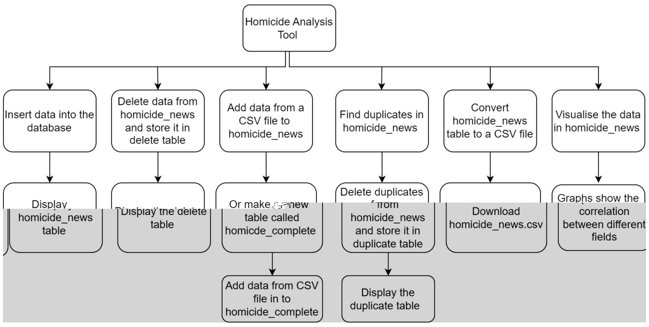
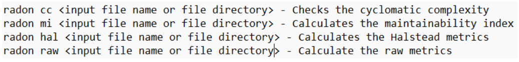

Homicide Analysis Tool
Insert data into the database
Delete data from homicide_news and store it in delete table
Add data from a CSV file to homicide_news
Find duplicates in homicide_news
Convert homicide_news table to a CSV file
Visualise the data in homicide_news
Display homicide_news table
Display the delete table
Or make a new table called homicde_complete
Delete duplicates from homicide_news and store it in duplicate table
Download homicide_news.csv
Graphs show the correlation between different fields
Add data from CSV file in to homicide_complete
Display the duplicate table

Figure 1 : GUI functionality.

4. METHODOLOGY

Three different tests are conducted to see the complexity of the two GUI to determine which GUI is less complex. Moreover, these tests also indicate which GUI will use fewer resources of a computer. The tests are as follows:

1. **API response size:** This test examines the API response size of the two GUIs (Dash and Streamlit) to assess their impact on the homicide analysis tool's performance and bandwidth needs. The tool is run, with Dash on port 8050 and Streamlit on port 8501. During testing, actions are performed in the GUIs, and the network tab in Google Chrome is monitored to capture the API response sizes for each action. These sizes are then recorded and tabulated.
2. **Code complexity:** This test checks for code complexity by finding the cyclomatic complexity of the code and the Halstead metrics of the code. Cyclomatic complexity measures the number of decision points in the code which helps to understand the flow of the code. Halstead metrics quantify the complexity based on the operation and operands in the code. To conduct these tests, the Python library called Radon is used. After installing the Radon library, specific commands are run to conduct the tests mentioned above. The commands are run on the terminal. The command snippets are shown below



```
radon cc <input file name or file directory> - Checks the cyclomatic complexity
radon mi <input file name or file directory> - Calculates the maintainability index
radon hal <input file name or file directory> - Calculates the Halstead metrics
radon raw <input file name or file directory> - Calculate the raw metrics
```

Figure 2 : GUI functionality.

3. **Error rate and handling of each API under stress testing:** In this test, both the Dash and Streamlit GUIs were deployed on a test server or local environment to handle multiple simultaneous requests [7]. Stress testing was performed using Apache JMeter, which simulates a high volume of users and concurrent requests to each API. The error rate, representing the percentage of requests that result in errors, was measured during testing. An increase in the error rate indicates that the APIs are unable to manage the load, leading to failures. Apache JMeter was selected due to its ability to effectively stress test and assess the performance of web applications, functional tests, and database servers.

5. EVALUATION AND RESULTS

The results from each of the tests are as follows:

5.1 API response size

The amount of memory and resources required to load the GUI on a server from each API is tested. This depends on the number of requests made for each action that takes place on GUI. The higher the number of requests to perform an action on the GUI, the higher the memory and resources required for the action to take place [8].

Table 1 : API response size

Action Taken	**Dash API response size**	**Streamlit API response size**
Insert function	888 KB	74.5KB
Display homicide_news table function	3.5 MB	159.7KB
Delete function	14.1 KB	3.2 KB
Duplicate function	545 KB	17.5 KB
Import and Export function	1.3 MB	50 KB
Data visualization function	645.7 KB	85.4 KB

Table 1 shows that for each action taken, the response size from the dash API is larger than the response size of the Streamlit API. Table 2 shows the total number of requests made while the experiment was taking place, the amount of memory transferred over the internet for these requests and the amount of resources used to load the page.

Table 2 : Summary results

API	**Number of requests**	**Memory transferred over the network**	**Resource loaded by the dashboard**
Dash	285	6.4 MB	6.4 MB
Streamlit	44	1.5 MB	4.8 MB

Table 2 shows that the dash API has a higher number of requests than streamlit. Due to the high number of requests, the dash API has a larger amount of memory being transferred over the network and the resources required to load the dashboard are higher. Therefore, the dash API requires more resources to make the dashboard than the streamlit API. This is because the Dash API uses a reactive, server-side model where all callbacks and components are handled on the server. This means that, during each interaction (like updating a graph or clicking a button), Dash often sends a significant amount of data back and forth between the server and the client. This can result in larger API responses because the server has to keep track of, render, and send the full layout or updates of each component involved in a callback. Furthermore, Dash maintains the entire application state on the server, which can result in more data being transmitted with every user interaction. When there is an interaction with the dashboard, Dash needs to send more data back to the client to ensure that all components are properly updated and rendered in sync with each other. This results in a high number of requests being made by the Dash API.

The Streamlit API, on the other hand, follows a more lightweight model where the UI is re-rendered from scratch on the client side, meaning fewer and smaller requests/responses are sent between the server and client. Streamlit's event model tends to avoid sending large state or layout data, which can lead to a smaller response size. Also, Streamlit is stateless. Streamlit doesn't store the state of the application on the server between user interactions; it rebuilds the app from scratch on each request. This often leads to smaller API responses since there's no need to send large amounts of state data back and forth. Therefore, resulting in a lower number of requests being made by the API.

5.2 Code Complexity

The cyclomatic complexity of each function in the code is tested for both APIs are tested. Moreover, the Halstead metrics on the API are conducted which tells the complexity of the code based on the operations done [9].

Table 3 : Cyclomatic complexity results for Dash and Streamlit

Function	**Dash code complexity rating**	**Streamlit code complexity rating**
render_plot	D(27)	C(19)
update_plot_type _dropdown	B(6)	B(6)
delete_record	B(10)	A(2)
update_data_display	B(9)	A(1)
display_selected_column	B(7)	A(4)
check_duplicates	B(7)	A(5)
upload_csv	A(5)	A(5)
update_custom_bar_graph	B(8)	A(4)

Table 3 presents the cyclomatic complexity test results for both the Dash and Streamlit dashboards. The function `render_plot` in the Dash dashboard is rated D(27), indicating high complexity due to its numerous conditional branches, decision points, or branching logic. In contrast, the `render_plot` function in the Streamlit dashboard is rated C(19), reflecting lower complexity[10]. This suggests that the Streamlit function contains fewer branches, loops, and logic than its Dash counterpart. Additionally, several functions in the Dash dashboard have B(6) and B(7) ratings, indicating they are relatively straightforward but beginning to introduce complexity[10].

The functions `delete_record` and `update_data_display` in the Dash dashboard have moderate complexity ratings of B(10) and B(9), respectively. While not overly complex, these functions could become harder to maintain if further expanded. In the Streamlit dashboard, `update_data_display` is rated B(6), signifying it is less complex and easier to manage than the corresponding Dash function.

Table 3 shows the Halstead metrics for both APIs. h1 is the number of distinct operators in the program. h2 represents the number of distinct operands(variables, constants). N1 represents the total number of operators in the code, including repeats. It indicates how often operations are being performed. N2 represents the total number of operands in the code, including repeats. A larger number of operands generally indicates more complexity in terms of data handling[10]. Vocabulary represents the total number of unique operators and operands that make up the code. A larger vocabulary indicates a broader range of components used in the program. Vocabulary is calculated using the equation Vocabulary = $h1 + h2$

Table 4 : Halstead metrics for Dash and Streamlit

Halstead metrics	Dash API	Streamlit API
h1	10	6
h2	113	48
N1	82	39
N2	154	71
vocabulary	123	54
length	236	110
calculated_length	803.8995	283.5879
volume	1638.4334	633.0376
difficulty	6.81415	4.4375
effort	11164.546	2809.104
time	620.25257	156.06135
bugs	0.546144	0.211012

Length is the total number of operators (N1) and operands (N2) in the program. It's a basic measure of how "long" the code is, considering the number of actions (operators) and variables/constants (operands) involved.

The calculated length is the theoretical length. It estimates the program's "theoretical" length based on the distinct operators and operands. If the calculated length is higher than the actual length, it suggests the code has more operators and operands therefore it is more complex. It is calculated using *Equation 1*

$$\text{Calculated length} = h1 \cdot \log_2(h1) + h2 \cdot \log_2(h2) \quad (1)$$

Volume measures the size of the implementation in terms of the information content of the program. It indicates how much mental effort it takes to understand the code. A higher volume suggests more complex logic or structure. It is calculated using *Equation 2*

$$\text{Volume} = Length \cdot \log_2(Vocabulary) \quad (2)$$

Difficulty measures how difficult it is to write or understand the program. It's based on the ratio of total operands to distinct operands and the number of distinct operators. The difficulty level is found using *Equation 4*

$$\text{Difficulty} = \frac{h1}{2} \cdot \frac{N2}{h2} \quad (3)$$

Effort is the total amount of mental effort required to develop or understand the program. A high effort means more work is required to maintain the code. The equation used for calculating effort is $Effort = Volume * Difficulty$

Time is an estimate of how long it would take to read and understand the program. It's based on the effort divided by a standard processing rate of 18 instructions per second. The equation used to find time is Time = $\frac{Effort}{18}$

Bug estimates the number of potential bugs in the program. Halstead's metric assumes that more complex code is more likely to contain bugs. The dash dashboard has a value of 0.54 which suggests there is a 54% chance the code might contain a bug. On the other hand, the streamlit dashboard has 21% chance it contains a bug.

Table 3 shows the Halstead metrics results for the Dash dashboard is very high when compared to the Halstead metrics results of the Streamlit dashboard. Therefore, this shows that the dash dashboard has more operators and operands which makes it a more complex code. Due to this difficulty required to understand and maintain the code for the dash dashboard is higher making it more difficult to maintain. It also requires a very high amount of time to read and understand the code and it is more prone to bugs.

5.3 *Error rate and handling of each API under stress testing*

Stress testing is done on both dashboards to see which one can have the least amount of error rate. Using Apache Jmeter, multiple threads/users are used to simulate concurrent users[7] [11]. These users then access the dashboard and do multiple GET or POST requests on the dashboard simultaneously. Table 4 shows the performance of dash dashboard and the streamlit dashboard using Apdex (Application Performance Index). Apdex score is a standard for measuring the performance of an application, particularly its responsiveness to users, during load testing. The Apdex score is calculated based on thresholds for acceptable response times, providing a single, normalized score between 0 and 1. 1 means that all users are satisfied and the dashboard is working in peak performance. 0 means that all users are frustrated and the dashboard is not performing. Apdex can be calculated using e

Table 5 : Performance and error rate results

Users	Number of Samples	Apdex of Dash dashboard	Error rate of Dash dashboard	Apdex of Streamlit dashboard	Error rate of Streamlit dashboard
10	700	1.000	0%	1.000	0%
100	7000	0.988	0%	1.000	0%
250	17500	0.353	62.67%	0.535	0%
500	35000	0.180	81.36%	0.391	16.24%
1000	70000	0.069	92.76%	0.000	100%

$$\text{Apdex} = \frac{\text{Satisfied Count} + \left(\frac{\text{Tolerating Count}}{2}\right)}{\text{Total Requests}} \tag{4}$$

Where:

Satisfied Count number of requests with a response time $\leq$ T

Tolerating Count number of requests with a response time $>$ T but $\leq$ 4T

Total Requests total number of requests made

Moreover, *Figure 3* and *Figure 4* show the request summary made by the simulated users and how many requests were successful during the stress test and how many failed. With the help of Table 4, *Figure 3* and *Figure 4*, it can be seen that the Dash dashboard's Apdex significantly decreases when 250 users are concurrently active and the dashboard fails consistently thus increasing its error rate. On the other hand,

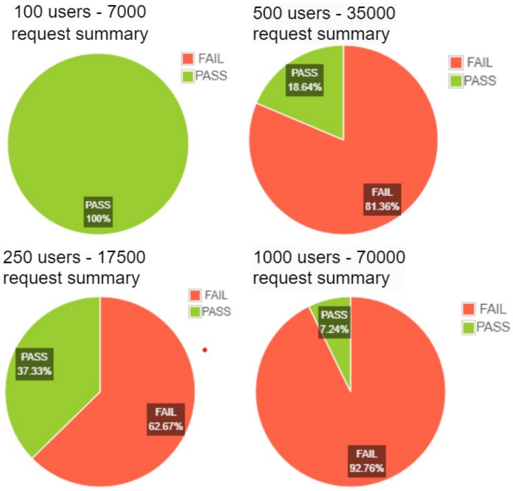


Figure 3 : Cyclomatic complexity results for Streamlit

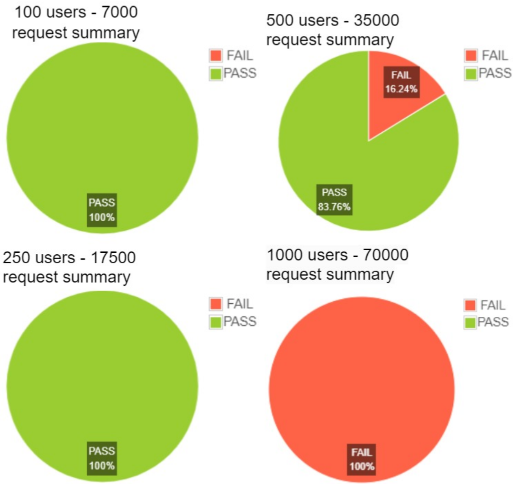


Figure 4 : Cyclomatic complexity results for Streamlit

the Streamlit dashboard is at peak performance when 250 users are active and its performance only starts degrading when 500 users are using the dashboard. The connection error for the Dash dashboard happens due to Dash's lower concurrency limits, as it typically runs on a single-threaded Flask server. Streamlit, on the other hand, uses Tornado, which handles higher concurrency more efficiently. Additionally, Dash applications tend to involve more complex interactions and larger responses, increasing the load on the server. The issue may also be linked to server configuration, as Dash may not be optimized for handling multiple requests simultaneously, while Streamlit is more robust in managing high traffic. This shows that the Streamlit dashboard has a higher tolerance for excessive users using the dashboard and has a lower error rate when compared to the Dash dashboard.

6. DISCUSSION AND FUTURE RECOMMENDATION

The results from each tests point out that the Dash dashboard is much more complex than the Streamlit dashboard. This can be seen from the results of the cyclomantic complexity tests, the API response size tests and the stress tests. Furthurmore, Dash dashboard only works on a single thread and it has long running callbacks which leads to the dashboard crashing when high number of users use the dashboard. Streamlit on the other hand, uses Tornado, a more scalable web server capable of handling higher concurrency levels by design and it is efficient at handling rapid request-response cycles under load. All these points show that streamlit is a much better library in creating a dashboard than Dash. But this does not mean that the Dash library can not be used to make dashboards. It is harder to implement the Dash library but it works well with Plotly.

Both dashboard are susceptible to SQL injection as they were not developed with security in mind, therefore, anyone could unauthorised could change valuable information in the database using the dashboard. Furthermore, the admin password and username can be seen in the front-end application code. This is a bad coding practice as if the code is made open-source then anyone can access the database directly from PostgreSQL which could led data corruption, deletion or data manipulation in an unintended manner. Therefore, separation of concern needs to be employed to refactor the dashboard code so that the dashboard is more modular, maintainable and easy to understand.

7. CONCLUSION

The homicide media analysis tool facilitates data collection, analysis, and visualization. However, while the Dash dashboard offers superior data visualization capabilities, it requires more resources and is significantly more complex and unstable compared to the Streamlit dashboard. This indicates that Streamlit may be preferable for the final product, despite its lower optimization for the Plotly library, which enhances visualizations. Consequently, a trade-off arises between having a more optimized dashboard with lower-quality visualizations and a less optimized dashboard with higher-quality visualizations. Additionally, both dashboards are vulnerable to security breaches, including SQL injection attacks, social engineering, and poor coding practices that expose database credentials. These security issues must be resolved before the homicide analysis tool can be safely made available to researchers and the public.

REFERENCES

[1] M. Damons and N. Geffen. \How bad is South Africa's murder rate?" https://groundup.org.za/article/how-bad-murder-in-south-africa/, June 2023. Accessed on 21 October 2024.

[2] N. R. Brodie. *Using mixed-method approaches to provide new insights into media coverage of femicide*. Ph.D. thesis, University of the Witwatersrand, Faculty of Humanities, 2019.

[3] G. S. Writer. \Shocking murder numbers in South Africa { 70 people killed every day." https://businesstech.co.za/news /government/788892/shocking-murder-numbers-in-south-africa-70-people-killed-every-day/, August 2024. Accessed on 21 October 2024.

[4] H. Canton. \United Nations Office on drugs and crime | UNODC." In *The Europa Directory of International Organizations 2021*, pp. 240{244. Routledge, 2021.

[5] UNO. \Office on Drugs and Crime." https://www.unodc.org/unodc/en/data-and-analysis/statistical-activities.html, March 2022. Accessed on 21 October 2024.

[6] ISS. \ISS CRIMEHUB." https://crimehub.org/what-is-the-crime-hub, January 2010. Accessed on 21 October 2024.

[7] B. Erinle. *Performance Testing with JMeter 3*. Packt Publishing Ltd, 2017.

[8] E. D. Demircioğlu and O. Kalipsiz. \API message-driven regression testing framework." *Electronics*, vol. 11, no. 17, p. 2671, 2022.

[9] M. P. Malhotra, M. K. Shah, J. Rathod, and M. Mehta. \Python based software for calculating cyclomatic complexity." *International Journal of Innovative Science, Engineering and Technology*, vol. 2, no. 3, pp. 546{549, 2015.

[10] D. De Silva, R. New Kandy, B. Sachethana, S. Dias, P. Perera, M. Katipearachchi, and T. Jayasuriya. \The Relationship between Code Complexity and Software Quality: An Empirical Study."

[11] D. Nevedrov. \Using jmeter to performance test web services." *Published on dev2dev*, pp. 1{11, 2006.

Appendix A: Group Work Reflection

This project served as the final endeavor for my Information Engineering degree and involved collaborating with another student to form a two-member group. My partner for this project was Muhammad Bux (2130436). We chose to work together due to our shared interest in the project and our long-standing friendship, which helped maintain a high work rate throughout our collaboration. Our established friendship made the partnership enjoyable, as we could engage in shared interests and hobbies. Additionally, we were able to meet outside of school hours, such as on weekends, to ensure we stayed on track with the project. Having previously partnered on other coursework also facilitated a harmonious working relationship, as we had developed a similar work pace and style.

Our communication was effective, thanks to our prior experiences and friendship. Clear communication ensured that both partners understood their assigned tasks and deadlines, allowing for honest discussions about any delays due to illness or personal commitments. We faced challenges related to understanding PostgreSQL, the Dash library, and debugging code. In such instances, we would collaboratively discuss solutions and plan to get back on track, providing support for each other when mistakes occurred.

The project tasks were divided evenly between us to maintain group morale and optimize efficiency. Tasks were assigned based on our individual interests and skills, with me focusing on the database while Muhammad concentrated on the dashboard. This allocation allowed us to work more effectively, as we could pursue separate objectives when possible. For tasks requiring both partners' involvement, we collaborated closely, sometimes tackling different aspects of the same task. We also offered assistance and advice on each other’s responsibilities to ensure that the output met both of our standards.

These factors contributed to a relatively smooth and stress-free experience while working with Muhammad Bux.

PROJECT PLAN FOR AN OPEN-SOURCE TOOL FOR HOMICIDE MEDIA ANALYSIS

Syed Mohammad Khizar Anas (2166010), Muhammad Bux (2130436)

School of Electrical & Information Engineering, University of the Witwatersrand, Private Bag 3, 2050, Johannesburg, South Africa

Abstract: This project presents the design and development of the data collection and data analysis tools for a homicide media tracker for humanities researchers who are researching homicide media. The primary objectives are to develop a user interface (UI) for researchers to input information regarding homicide cases and news articles that cover those corresponding cases, to create data visualisation tools that give researchers more insight into homicide cases and to make the homicide media tracker easily accessible to people. The project leverages information engineering principles to create the homicide media tracker a user-friendly website and to develop the homicide media tracker using open-source tools so that it is accessible and free to the researchers. The anticipated outcomes will significantly help researchers in storing and archiving homicide media data and in analysing the data. The project is expected to be completed in 6 weeks, 1 week ahead of the 7-week deadline, providing slack time for any unforeseen delays. This project showcases the interdisciplinary collaboration between engineering and humanities to address the storage of news articles on homicide cases.

Key words: User Interface (UI), newspaper, homicide, Python, Prototype

1. INTRODUCTION

This report presents a comprehensive plan for a seven-week investigation project commencing on the 26th of August and concluding on the 10th of October. The project aims at the development of data collection and analysis tools for a homicide media tracker. This is a critical tool for humanities researchers who are inexperienced in handling and representing data.

The primary objective is to design and develop a homicide media tracker website that can store and archive data on homicide cases that are captured by news articles in South Africa. This will be done by creating a UI that researchers will use to enter data on homicide cases that are captured in news articles present on the internet and newspapers. Moreover, data visualisation tools will be designed and developed which will help researchers out in data representation.

The anticipated outcome of this project is expected to help researchers capture and store data on homicide cases in newspapers and news articles in a systematic and fast way without copyright infringement and the data can be exported on any machine in CSV format. Moreover, it will also help them find news articles and newspapers on homicide cases that have happened in South Africa in the past more easily and efficiently. Furthermore, it will allow researchers who are not good at data handling to visualise the data more easily and efficiently.

Section 2 outlines the project scope in which the project background, objectives, assumptions and constraints will be discussed. Section 3 addresses research risks and mitigation strategies. Section 4 identifies the necessary resources, including personnel and equipment. Section 5 details the methodology used to design and develop the homicide media tool. Section 6 outlines project milestones and deliverables. Section 7 discusses the project implementation strategies. Section 8 talks about the project management strategies. After that, the next section concludes the report.

2. PROJECT SCOPE

The scope outlined below includes the project background, objectives, constraints, and assumptions.

2.1 Project Background

In the current digital age, the accessibility, organization, and analysis of large datasets, particularly from news archives and other document sources, have become increasingly important. However, significant challenges exist in accessing and systematically collecting this data, especially regarding news reporting on homicide cases. Many news archives are locked behind paywalls, restricting access for researchers and the general public. This limitation creates substantial barriers for those without the financial resources to subscribe to multiple news services. Even when accessible, commercial news publications often provide fragmented and incomplete data, making systematic analysis difficult.

The inconsistent nature of data collection further complicates this issue. News is not consistently collected or archived, and various organizations that offer news clipping or archive services are often unreliable. Articles can be lost due to changes in website URLs or the closure of news outlets, leading to incomplete datasets. News coverage does not always accurately represent the reality of crime, with certain types of homicides, particularly those involving male victims, being more frequently reported. Male homicides make up 85% of all murder cases. In South Africa, the high volume of murder cases makes comprehensive coverage unattainable. Less than 20% of female homicides are reported in the news, and coverage is often concentrated in specific areas, neglecting others.

Despite these limitations, news reports provide valuable qualitative insights that cannot be obtained from police reports or statistical data. They offer context, mode, and consequences of crimes, which are crucial for a deeper understanding of homicide cases. However, news creation is qualitative and lacks the systematic approach needed for comprehensive analysis. Additionally, many researchers worldwide lack funding and cannot afford paid news services, hindering their ability to conduct thorough research and analysis.

2.2 Project Objectives

The primary objectives of the project are to develop a platform designed as a homicide media tracker capable of capturing data and metadata from homicide-related news articles reaching as far back as 100 years, as well as the capability to visualize the data utilizing a user-friendly, customisable dashboard. The platform is to be available to as many people as possible, therefore necessitating the use of purely open-source tools.

2.3 Project Assumptions

The following assumptions have been made:

- It is assumed that the database provided is sufficiently filled with accurate information as a starting point.
- The data is sufficiently pre-processed and appropriately labelled, therefore suitable for analysis.
- The data provided complies with all legal and ethical standards.

2.4 Project Constraints

The investigation is limited to several constraints, which are outlined in the following:

- The project is to be completed within a 7-week time frame, therefore limiting the extent of features and functionality which can be implemented.
- The platform is to be developed using open-source tools, further limiting functionality.
- The platform must be able to run locally on machines. This limits the ability to use cutting-edge techniques due to computational infeasibility.

3. RISKS

These are the potential risks in this investigation project:

3.1 Low Data Quality and Lack of Availability

There may be limited access to news archives due to paywalls or restricted access policies. There may also be inconsistent, incomplete or incorrect data from news sources. These may hinder the platform's ability to provide accurate and meaningful insights. Mitigation strategies would include implementing robust data cleaning and preprocessing methods, as well as using multiple sources to cross-verify information.

3.2 Compliance Risks

There are no risks related to copyright infringement as the tool will just be used to capture data from articles and store it, rather than capturing the entire work of the author.

3.3 Dependency Risks

Third-party tools and libraries could become obsolete, therefore causing delays in project completion due to needing to search for and implement alternative measures. Mitigation strategies would include choosing well-supported and widely-used technologies, along with having contingency plans for replacing critical dependencies.

3.4 Validation Risks

The strict time frame in which the project needs to be completed may lead to insufficient time for testing should delays take place. This could lead to undetected issues surfacing during public use. Mitigation strategies would include allocating adequate time and resources for thorough testing, and the use of automated testing tools where possible.

4. REQUIRED RESOURCES

4.1 Hardware

- **Computing**: A computer with sufficient CPU and GPU resources is essential for the training of machine learning models and the execution of feature selection algorithms.

4.2 Software

- **Python**: The Python programming environment will be used for data analysis, feature extraction, data visualisation and the application of machine learning libraries. The Python programming language will also be used to create the UI of the website. It will also be used to develop the server required to run the website on.
- **Libraries and Toolkits**: Python offers extensive libraries for data analysis, feature extraction, UI design and development, server development and machine learning, which will be utilized throughout the project.

4.3 Data

- **News articles and newspaper on homicide cases**: Access to 5000 homicide cases that are covered by various new publications is required. These entries of data will be essential to check if the website is functioning properly and if all functions are available. This is crucial in the development of the project.

4.4 Development Tools

- **Version Control**: GitHub will be used for version control to manage code, track changes, and facilitate collaborative development.
- **Communication Tools**: Communication will be maintained through email, Microsoft Teams, and WhatsApp to ensure efficient collaboration and review processes.

4.5 Diagramming and Documentation Tools

- **Draw.io**: This free online tool which will be used for creating diagrams and visual representations of project workflows and results.
- **MATLAB**: MATLAB's built-in diagramming capabilities will also be leveraged. The university will provide the necessary licenses.
- **LaTeX**: Access to online LaTeX editors for documentation and report generation. The university provides free Wi-Fi, which supports the use of these tools.

4.6 Training and Tutorials

- **Machine Learning Tutorials**: Tutorials and instructional materials for the use of machine learning libraries in Python can be obtained from YouTube or other Web resources to ensure that all team members are proficient in the necessary techniques and tools.

5. PROJECT METHODOLOGY

The investigation project will employ a hybrid development process that combines the Agile Software Development Process and the Waterfall Development Process. This hybrid approach is chosen to leverage the strengths of both methodologies which provide a balance between flexibility and structure. This suits the nature of this project.

The Agile Software Development Process is characterised by its iterative and flexible nature, which allows for continuous improvements and adaptation throughout the project. This will minimise risks when adding new functionality to the project. This is because it will allow the project team to work in sprints which will be one week long[1]. In these sprints, the project team will design and develop a prototype of the homicide media tracker and do testing on it to check for bugs. If bugs are found then they will be resolved in the next iteration which will happen in the next sprint and the required functionality will be added to the prototype.

On the other hand, the Waterfall Development Process is a linear and systematic approach that involves progressing through a series of distinct stages, from the initial requirements gathering to final product delivery[2]. This approach provides a clear structure and road map for the project, ensuring that all necessary steps are completed logically and thoroughly.

Using these two approaches combined will allow the project team to proceed through the project easily and efficiently. A flowchart is presented below which summarises the methodology process.

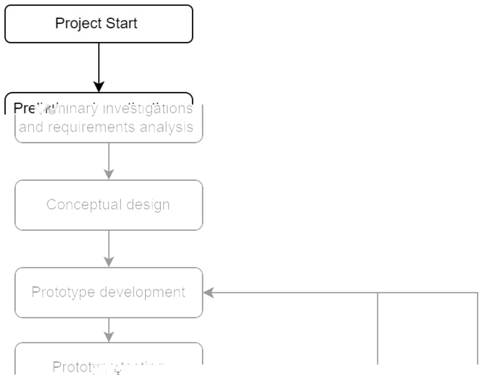


Figure 1 : Flowchart showing the methodology process.

5.1 Preliminary investigation and requirements

The first stage of the project involves a detailed study of existing apps and websites that store or archive news articles and identifies potential areas of improvement in these apps/websites. Moreover, different open-source libraries in Python will be explored to design and develop the prototype. Alongside this, the specific requirements, constraints and assumptions for the homicide media tracker will be defined by having meetings with the client. The scope of the project and deliverables will be defined as well. Access to the database will be given by the client with 5000 data entries in it.

5.2 Conceptual Design

This phase begins with the development of a preliminary design for the homicide media tracker which is informed by the findings from the initial step taken in the project. The design will integrate innovative solutions to enhance the design of existing homicide media archives/websites. The focus of this stage is to understand the system's functions and the relationships between its components.

5.3 Prototype development and initial testing

In this phase, a prototype of the homicide media tracker will be developed, incorporating all key design elements and operational parameters defined during the planning stage. There will also be a focus on data scraping and processing scripts, as well as the development of the layout and interactive elements of the dashboard. Initial validation tests will be conducted on this prototype to verify the initial functionality that is implemented and adherence to the design specifications. Moreover, the back end and front end will be connected, ensuring seamless data flow and interaction. This phase will also involve the implementation of callbacks and dynamic updates within the dashboard to reflect real-time data analysis

5.4 Iterative testing and improvements

Following the completion of the preliminary prototype, the project will transition into an iterative process of testing, feedback, and refinement. Each iteration will thoroughly test the homicide media tool's functionality and efficiency. This will be done to check for system bugs and to see if the functionality implemented is operating correctly. Bugs existing in the system will be fixed. If there are no bugs, then new functionality will be added to the prototype which adheres to the requirements of the project. This will then become the next prototype. Rigorous testing will be done again to check if any new bugs were introduced due to the implementation of new functionalities in the website. Each modification and its impacts will be meticulously documented, contributing to the comprehensive reporting of the project's progression and outcomes.

Potential risks will be identifed and managed proactively throughout the project life cycle. Regular risk assessment meetings will be held to identify new risks, evaluate their impact, and develop mitigation strategies.

5.5 Final evaluation, verification and reporting

Upon completion of the iterative testing and improvement phase, a final round of comprehensive testing will be conducted on the refined prototype. Specific details will be obtained from the client to determine the overall criteria for a successful product that can be validated. Another round of testing will be done to verify there are no bugs present in the final prototype and it adheres to the client's requirements.

Comprehensive documentation will be maintained throughout the project, including design documents, code comments, and user manuals. This will ensure that the platform is maintainable and extensible in the future. Regular progress reports will be prepared and shared with stakeholders, providing transparency and accountability.

All findings, methodologies, insights and bugs will be compiled into a comprehensive investigation report. This report will detail the investigation process, summarise the results, discuss any observed bugs or anomalies in the system, and propose recommendations for future research.

6. PROJECT MILESTONES

To track the project's progress and to ensure that project completion will be on time, a set of milestones has been chosen. The project is projected to be completed in 6 weeks which is 1 week ahead of the 7 weeks deadline. This allows for slack time to accommodate any unforeseen delays that may arise. Table 1 below, shows the list of milestones.

Table 1 : Milestone Estimation

Milestone	Deadline
Meet with researchers	26 August
Literature Review Complete	28 August
Become familiar with database	30 August
Begin prototype application development	2 September
Prototype complete	12 September
Testing commences	12 September
Testing complete	15 September
Start corrective actions	16 September
Corrective actions complete	21 September
Meeting with researchers	2 October
Poster preparations begin	5 October
Poster Complete	9 October
Open Day	10 October

7. PROJECT IMPLEMENTATION

The homicide media tool will be designed and developed using Python. In Python, the Dash library will be used which is an open-source framework specifically designed for building interactive web-based applications and dashboards [3]. It seamlessly integrates with Plotly's graphing library to provide a rich set of visualization capabilities, making it an excellent choice for creating complex and data-driven dashboards. Dash applications are web servers that run Flask, communicate with JSON packets over HTTP requests, and update web pages via React.js, offering a powerful and flexible way to develop interactive, user-friendly interfaces[4].

8. PROJECT MANAGEMENT

To facilitate the execution of the project, the processes outlined in the Project Methodology section are further decomposed into specific sub-tasks. These subtasks, which are guided by the milestones established in the milestone section which are detailed in the Work Breakdown Structure found in Appendix A. Moreover, the project's progression is monitored and managed in line with the Gantt chart in Appendix B specifically developed for this project.

8.1 Task distribution

The project will be managed collaboratively between the two primary developers, Muhammad Bux and Syed Anas, who will share the workload evenly to ensure efficiency and thorough coverage of all tasks and sub-tasks.

In the initial phase, Syed will be responsible for setting up the meetings with the client. These meetings will be held regularly to discuss project progress, address any challenges, and ensure alignment with the project objectives. Communication will be maintained through daily stand-ups and weekly review sessions, ensuring continuous feedback and adjustment of tasks as needed.

After the client meeting, the literature review and development of design specifications will be conducted in tandem by both Syed and Muhammad. The dual involvement will help in the thorough understanding and exploration of the research done on existing solutions and their shortcomings. It will also make it easier and more efficient to formulate the design specifications of the homicide media tracker.

After the design specifications are established, both team members will contribute equally to the design and development of the prototype. Collaboration at this stage ensures a fusion of skills and expertise, culminating in an initial prototype that meets the established design specifications. Syed will do the testing on the initial prototype. On the other hand, Muhammad will design and develop a new functionality that will be added to the prototype which will form the next prototype. Muhammad will also test for bugs in the new prototype. While Muhammad is testing for bugs, Syed will start the development process for the next functionality that needs to be added to the prototype forming the next prototype. He will then check for any bugs present in the new prototype. This will continue until the final prototype is developed which will be under the design specifications and will have no bugs in it.

Both Syed and Muhammad will try and identify potential risks during the testing phase for the homicide media tracker. If any risks are found then a meeting will be held between the two teammates in which the impact of the risk on the project will be discussed and a mitigation strategy will be drafted and implemented.

The data analysis and report drafting will be done individually. After this Syed and Muhammad will work together to create a video presentation of the project.

9. CONCLUSION

The Homicide Media Tracker project aims to enhance the systematic collection and analysis of homicide-related news data by developing an open-source digital platform. Leveraging Plotly's Dash for an interactive dashboard and advanced machine learning models for data analysis, the platform will provide valuable insights into homicide reporting patterns. Through a collaborative approach between Muhammad and Syed, rigorous testing, proactive risk management, and thorough documentation, the project ensures a reliable, user-friendly, and extensible tool. This initiative will significantly enhance data accessibility, foster community engagement, and support informed research and policy-making in the realm of crime data analysis.

REFERENCES

[1] I. wrike. \The Agile Software Development Life Cyle." https://www.wrike.com/agile-guide/agile-development-life-cycle/, 2024. Accessed: 14 July 2024.

[2] M. McGuire. \Top 4 software development methodology." https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies.html, 2024. Accessed: 14 July 2024.

[3] Plotly. \Dash documentation and user guide." https://dash.plotly.com/, 2024. Accessed: 14 July 2024.

[4] F. S. Ltd. \Dash Bootstrap Components." https://dash-bootstrap-components.opensource.faculty.ai/, 2024. Accessed: 14 July 2024.

Appendix A

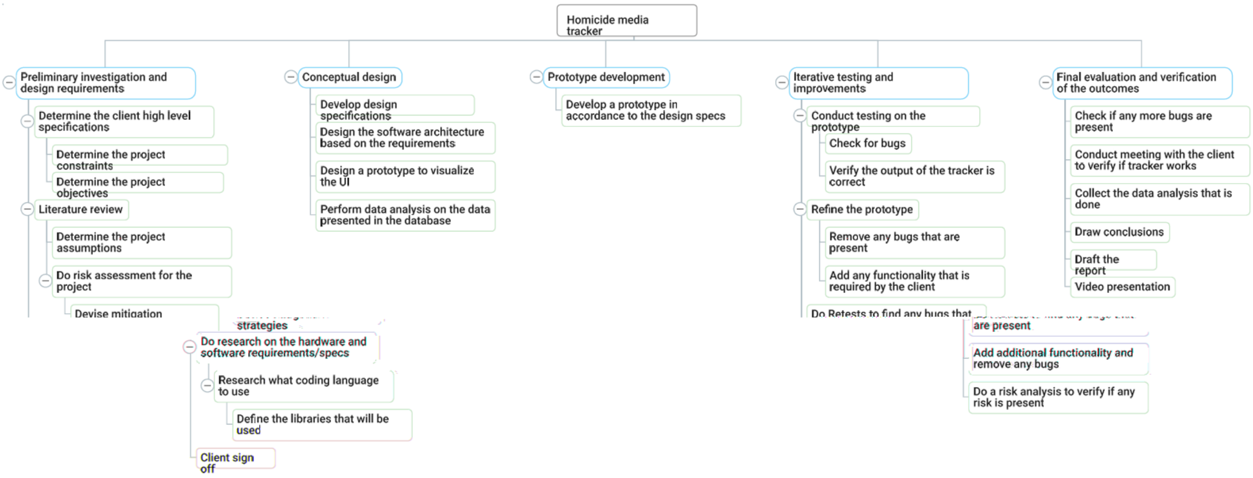
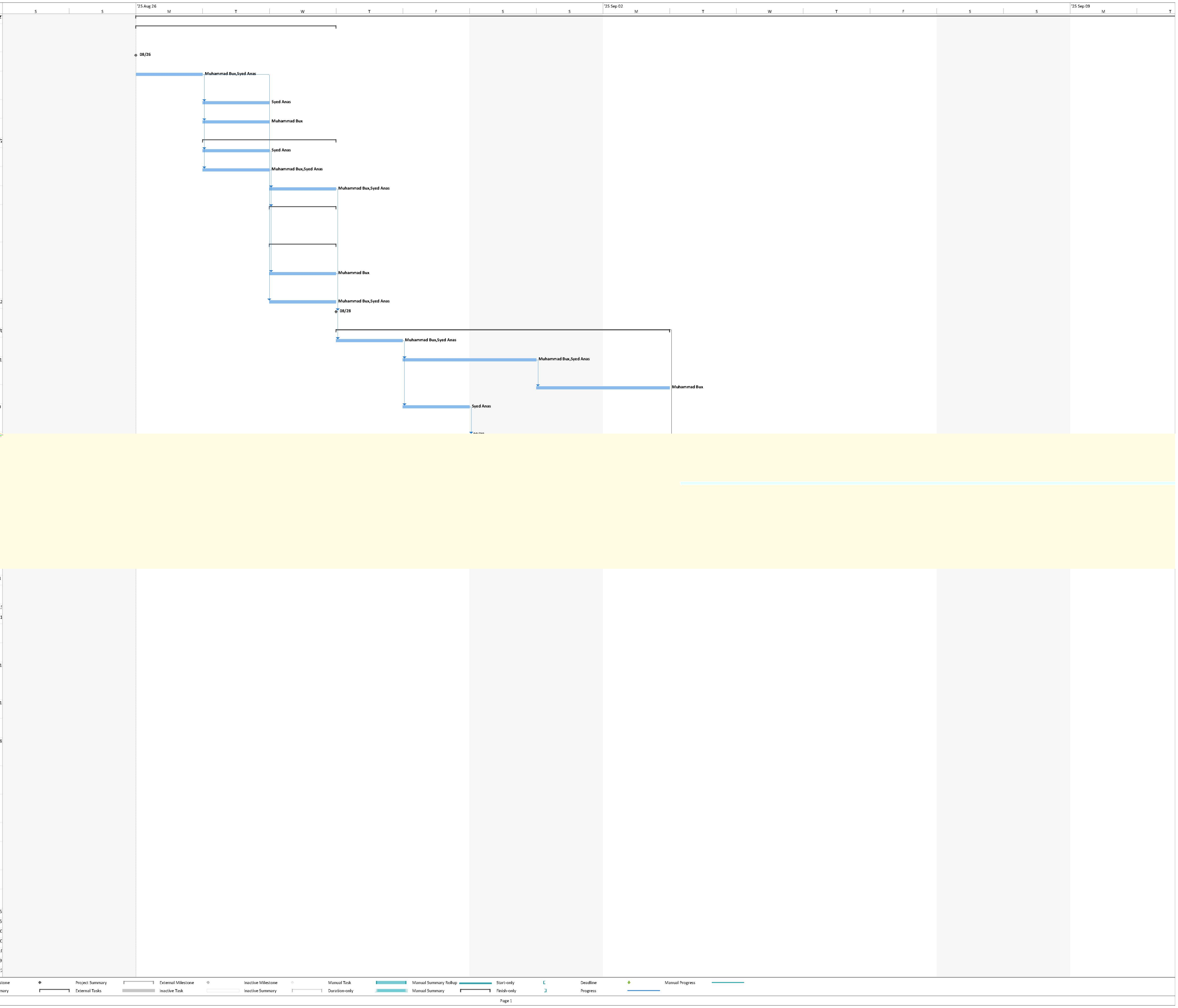

Homicide media tracker
Preliminary investigation and design requirements
Determine the client high level specifications
Determine the project constraints
Determine the project objectives
Literature review
Determine the project assumptions
Do risk assessment for the project
Devise mitigation
strategies
Do research on the hardware and software requirements/specs
Research what coding language to use
Define the libraries that will be used
Client sign off
Conceptual design
Develop design specifications
Design the software architecture based on the requirements
Design a prototype to visualize the UI
Perform data analysis on the data presented in the database
Prototype development
Develop a prototype in accordance to the design specs
Iterative testing and improvements
Conduct testing on the prototype
Check for bugs
Verify the output of the tracker is correct
Refine the prototype
Remove any bugs that are present
Add any functionality that is required by the client
Do Retests to find any bugs that
are present
Add additional functionality and remove any bugs
Do a risk analysis to verify if any risk is present
Final evaluation and verification of the outcomes
Check if any more bugs are present
Conduct meeting with the client to verify if tracker works
Collect the data analysis that is done
Draw conclusions
Draft the report
Video presentation

Figure A1 : Work breakdown structure of the project plan

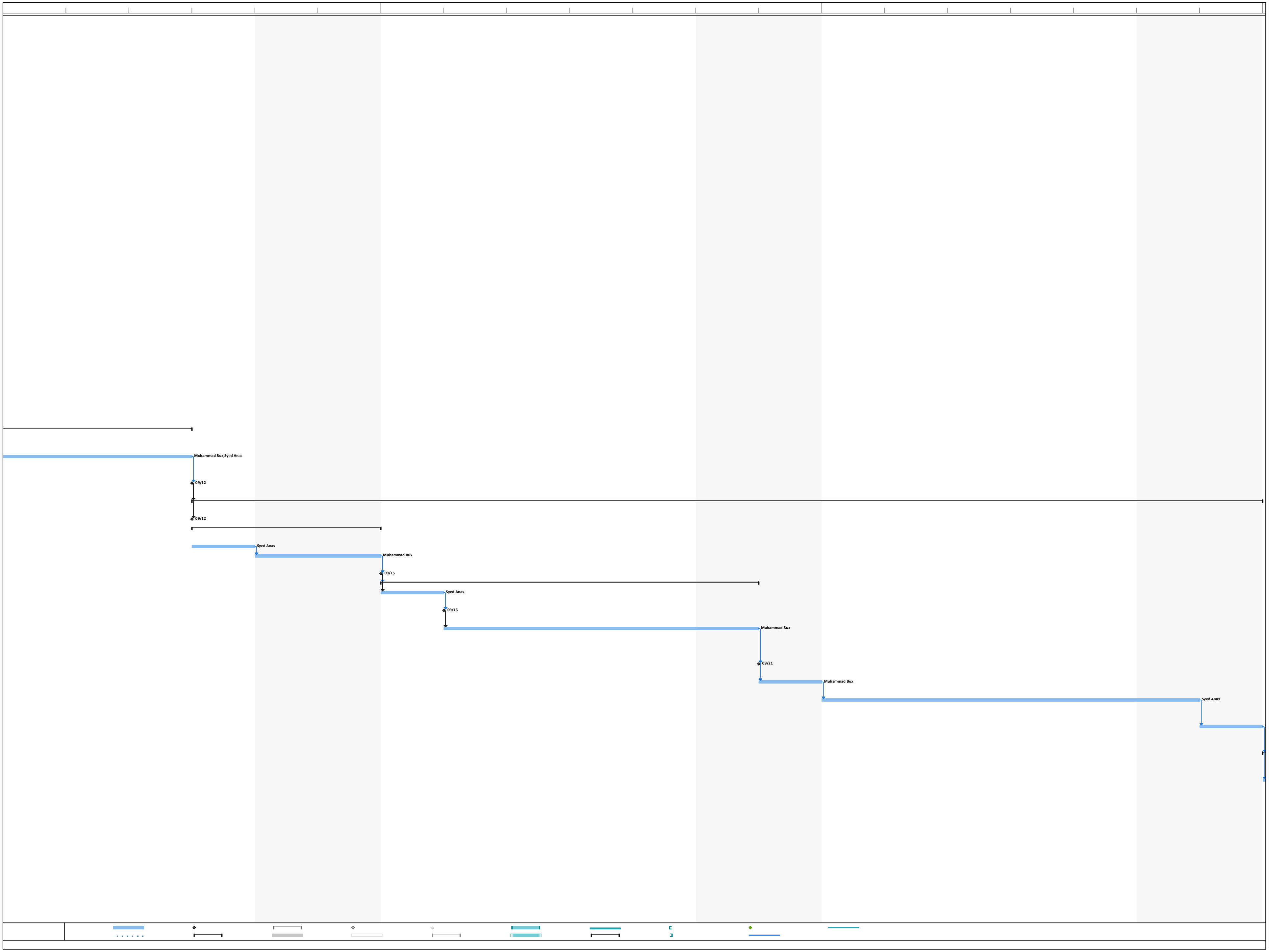
Appendix B

ID	ⓘ	Task Mode	Task Name	Duration	Start	Finish
1			**Homicide media trac**	**62,33 d**	**Mon 24/08/2**	**Sun 24/10/2**
2			**Preliminary investigation and design requirements**	**3 d**	**Mon 24/08/26**	**Wed 24/08/28**
3			Meet up with the researchers	0 d	Mon 24/08/26	Mon 24/08/26
4			Determine the client high level design requirements	1 d	Mon 24/08/26	Mon 24/08/26
5			Determine the project constraints	1 d	Mon 24/08/26	Tue 24/08/27
6			Determine the project objectives	1 d	Mon 24/08/26	Tue 24/08/27
7			**Literature review**	**2 d**	**Mon 24/08/2**	**Wed 24/08/2**
8			Determine the project assumptions	1 d	Mon 24/08/26	Tue 24/08/27
9			Do risk assessment for the	1 d	Mon 24/08/26	Tue 24/08/27
10			Devise mitigation strategies	1 d	Tue 24/08/27	Wed 24/08/28
11			**Do research on the hardware and software requirements**	**1 d**	**Tue 24/08/27**	**Wed 24/08/28**
12			**Research what coding language to use**	**1 d**	**Tue 24/08/27**	**Wed 24/08/28**
13			Define the libraries that will be used	1 d	Tue 24/08/27	Wed 24/08/28
14			Client sign	1 d	Tue 24/08/27	Wed 24/08/2
15			Literature review complete	0 d	Wed 24/08/28	Wed 24/08/28
16			**Conceptual design**	**5 d**	**Wed 24/08/2**	**Mon 24/09/0**
17			Develop design specifications	1 d	Wed 24/08/28	Thu 24/08/29
18			Design the software architecture based on the requirements	2 d	Thu 24/08/29	Sat 24/08/31
19			Design a prototype to visualize the UI	2 d	Sat 24/08/31	Mon 24/09/02
20			Perform data analysis on the data presented in the database	1 d	Thu 24/08/29	Fri 24/08/30
21			Become	0 d	Fri 24/08/30	[illegible]
			prototype			
29			Check for bug	1 d	Thu 24/09/1	Fri 24/09/13
30			Verify the output of the tracker is	2 d	Fri 24/09/13	Sun 24/09/15
31			Testing compl	0 d	Sun 24/09/1	Sun 24/09/1
32			**Refine the proto**	**6 d**	**Sun 24/09/1**	**Sat 24/09/21**
33			Remove any bugs that are present	1 d	Sun 24/09/15	Mon 24/09/16
34			Start corrective actions	0 d	Mon 24/09/16	Mon 24/09/16
35			Add any functionality that is required by the client	5 d	Mon 24/09/16	Sat 24/09/21
36			Corrective actions complete	0 d	Sat 24/09/21	Sat 24/09/21
37			Do Retests to find any bugs that are present	1 d	Sat 24/09/21	Sun 24/09/22
38			Add additional functionality and remove any bugs	6 d	Sun 24/09/22	Sat 24/09/28
39			Do a risk analysis to verify if any risk is present	1 d	Sat 24/09/28	Sun 24/09/29
40			**Final evaluation and verification of the outcomes**	**27,35 d**	**Sun 24/09/29**	**Sun 24/10/27**
41			Check if any more bugs are present	2 d	Sun 24/09/29	Tue 24/10/01
42			Conduct meeting with the client to verify if tracker works	1 d	Tue 24/10/01	Wed 24/10/02
43			Meeting with the researcher	0 d	Wed 24/10/02	Wed 24/10/02
44			Collect the data analysis that is done	1 d	Wed 24/10/02	Thu 24/10/03
45			Draw conclusion	2 d	Thu 24/10/0	Sat 24/10/05
46			Poster preparati	0 d	Sat 24/10/05	Sat 24/10/05
47			Prepare for oper	4 d	Sat 24/10/05	Wed 24/10/0
48			Poster complete	0 d	Wed 24/10/0	Wed 24/10/0
49			Open day	0 d	Thu 24/10/1	Thu 24/10/1
50			Draft the report	9 d	Thu 24/10/1	Sat 24/10/19
51			Video presentat	8 d	Sat 24/10/19	Sun 24/10/2

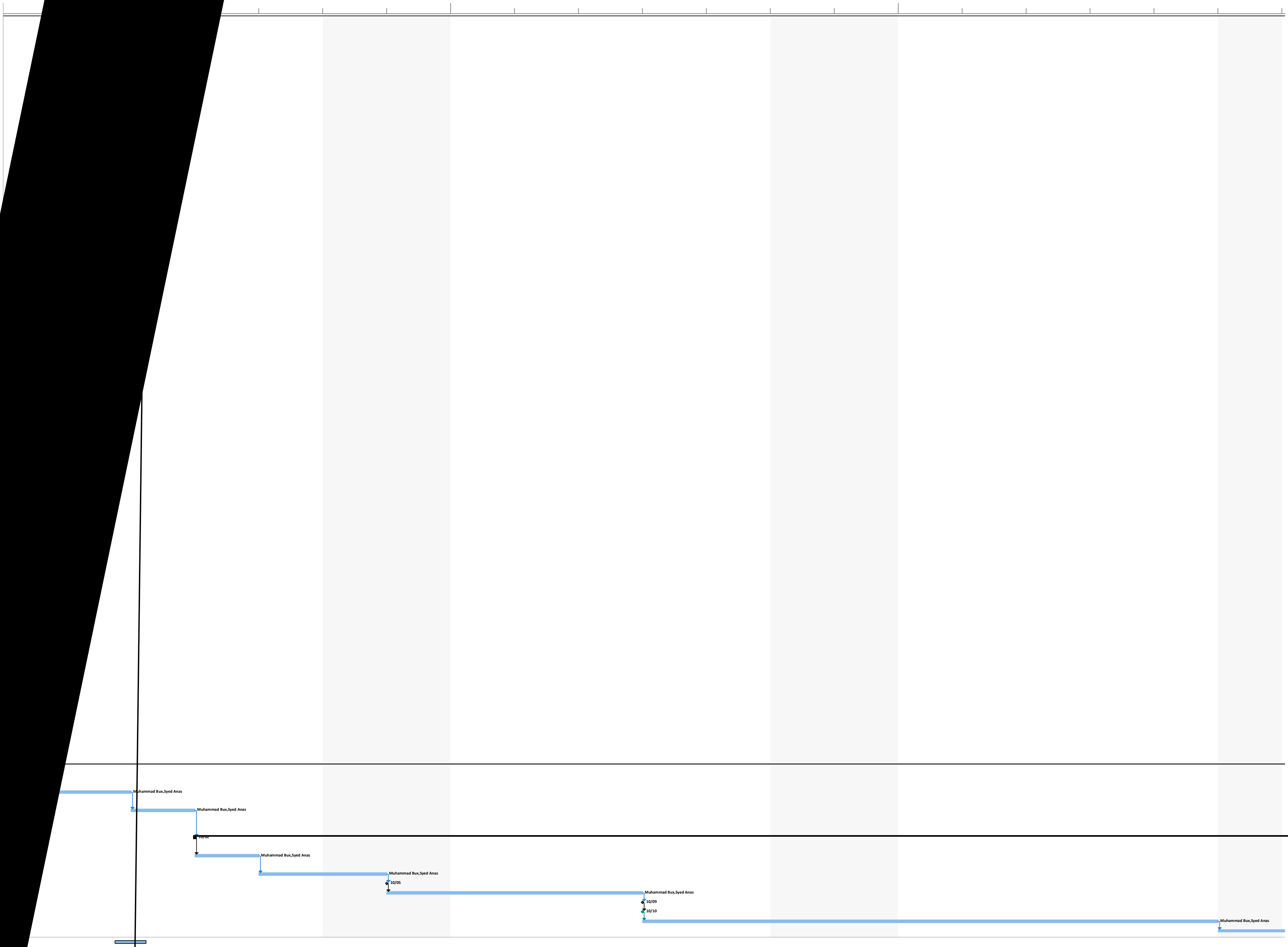


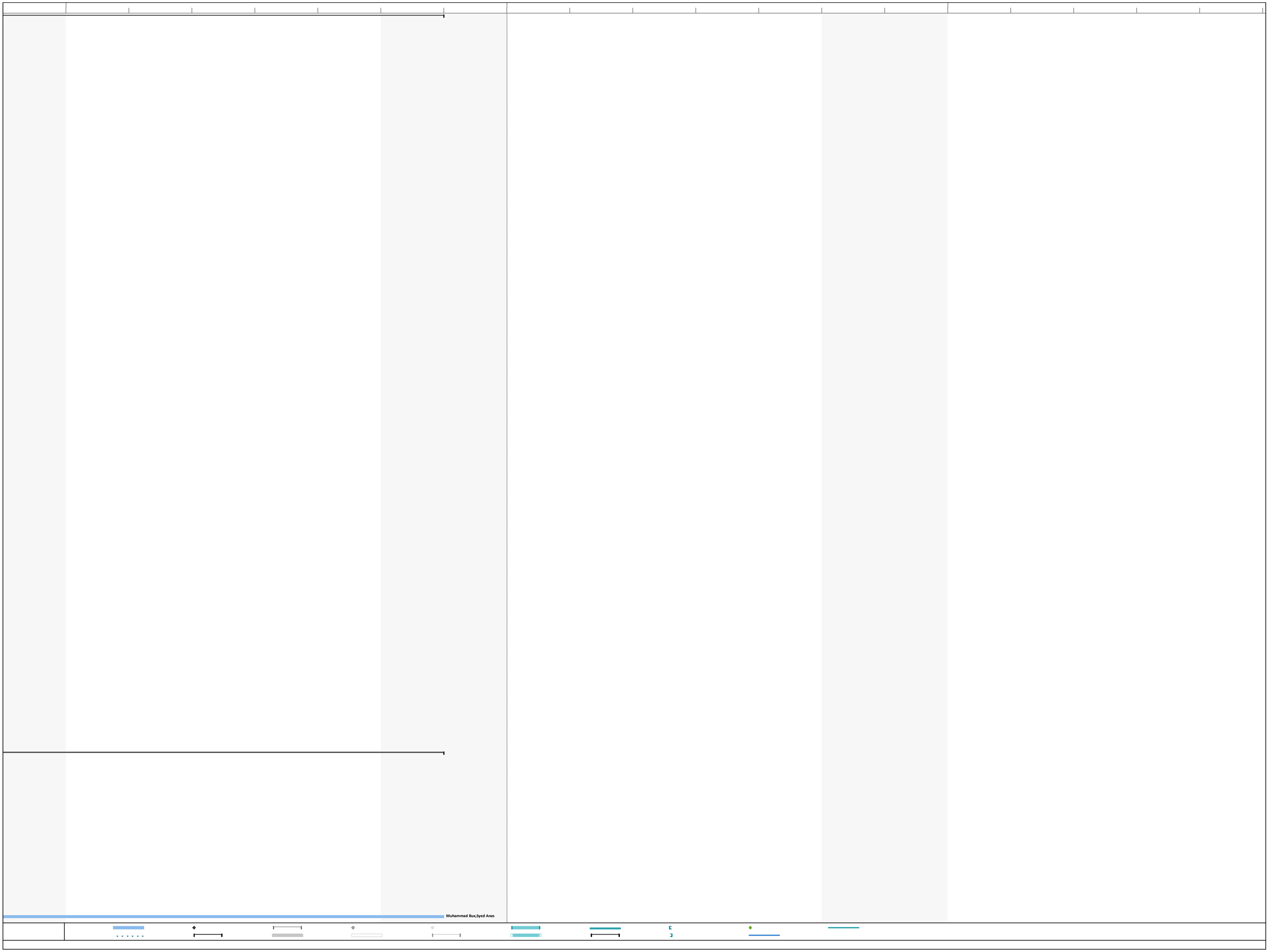
Project: Homicide media tracker
Date: Mon 24/07/15

Task	▬	Milestone	◆	Project Summary	⌐¬	External Milestone	◆	Inactive Milestone	◇	Manual Task	▬	Manual Summary Rollup	▬	Start-only	[	Deadline	⬇	Manual Progress	▬
Split	- - - - - -	Summary	⌐¬	External Tasks	▬	Inactive Task	▭	Inactive Summary	⌐¬	Duration-only	▬	Manual Summary	⌐¬	Finish-only	]	Progress	▬		


Muhammad Bux,Syed Anas
09/12
09/12
Syed Anas
Muhammad Bux
09/15
Syed Anas
09/16
Muhammad Bux
09/21
Muhammad Bux
Syed Anas

[redacted]


Muhammad Bux,Syed Anas
Muhammad Bux,Syed Anas
10/02
Muhammad Bux,Syed Anas
Muhammad Bux,Syed Anas
10/05
Muhammad Bux,Syed Anas
10/09
10/10
Muhammad Bux,Syed Anas


Muhammad Bux,Syed Anas

Appendix C

Cohort 9

Week 1: Meeting Minutes

Meeting Date: 28/08/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Akiva Levitt
Minute Taker: Joseph Kaplan

Overview: Introduction meeting introducing each other, discussing each project overview and finalising admin details. Group agreed to meet every Monday at 9-10 from now on.

Group 11: Edgar Ngoepe and Phuti Mokgehle

- Project Objective: Determining the optimum route up a sport or bouldering climbing wall
- This week: Visit the CityRock gym, capture high-quality images of the climbing wall from multiple angles and create a 3D Representation of the Climbing Wall.

Group 19: Akiva Levitt and Joseph Kaplan

- Building a VS Code extension that allows new programming C++ students to learn how pointers work, by visualizing the code provided.
- We have investigated various ways of going about doing the project, using JavaScript for backend, with Python to run Clang to interpret C++ code.
- Launch a VSCode extension that can be downloaded (1 step before an MVP)
- This week: Need to figure out how to launch a VSCode extension and subsequently decide the best way to take in the C++ code as input and output the corresponding visual representation of the pointer logic in the code

Group 22: Muhammad Bux and Syed Anas

- Building an open-source platform for tracking information retrieved from media about homicide cases and producing insights using various data visualization methods
- Database has been built previously, supervisor needs to provide access
- Using plotly dash to build an interactive dashboard for the data
- This week: get access to database and integrate database with the dashboard

Group 41: Shen Reddy and Dylan Baker

- Project Summary: Benchmarking TinyML systems. We will be comparing the performance of ML algorithms for anomaly detection in laptops and microcontrollers. We will also be comparing the energy consumption of both systems.
- Some research has been conducted into potential microcontrollers for the project as well as potential algorithms that can be used.
- This week: we will continue with our research and will aim to find datasets for anomaly detection as well as design an anomaly detection system for the laptop solution.

Group 47: Edward Ndlala and Lefa Mofokeng

- Project summary: investigation into power usage of IT Equipment based on varying computational load.
- Building algorithms that solves soduko and ticktactoe then measure Their computational power.
- This week: Research how the smart powerr distribution unit (PDU) device operates. Researching algorithms and soduko. This week is based on understanding the PDU and the algorithms

Week 2: Meeting Minutes

Meeting Date: 02/09/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Phuti Mokgehle
Minute Taker: Edgar Ngoepe

Overview: 2nd meeting each group shared what they worked on the previous week, plans for the upcoming week and any challenges encountered the previous week

Group 11: Edgar Ngoepe and Phuti Mokgehle

- We successfully visited the City Rock gym and captured high-quality images of the climbing wall from multiple angles.
- However, we faced challenges when attempting to create a 3D representation of the climbing wall. The software we obtained presented installation issues, and when we managed to execute the programs, our PCs struggled to handle the computations, with some tasks taking more than seven hours to complete.
- Plans for the Week: Finish the 3D reconstruction and start modelling the climber with the idea of developing a stickman.

Group 19: Akiva Levitt and Joseph Kaplan

- We have successfully published a VSCode extension, and it is now live. We successfully decided on and have implemented clang as the compiler that converts the C++ code into an AST that is stored in a JSON format. The visualiser aspect takes this JSON format and visualises basic pointer functionality using D3.js based on the AST.
- We need to create the google forms and the tutorial that will allow users to test our software. Our goal is to integrate the tutorial and questionnaires into a single streamlined environment so that it is plug and play for users.
- We have been struggling to figure out how to correctly integrate a video with audio into our VSCode extension. We think that there are VSCode settings that prevent this from being possible, but we will work more on this.

Group 22: Muhammad Bux and Syed Anas

- Last week: we were able to get access to the database and the data. We downloaded and initialised the database and found that some functionalities of the database do not work like adding data into the database, deleting data from the database and editing data in the database.
- This week: This week we will try to figure out how to make the functionalities that does not work in the database to work. If we cannot make the functionality work, then we will design

and develop the database from scratch and input the data provided to us. Moreover, we will do some research on user-friendly dashboard online which will help us design our dashboard this week.

- Challenges faced: The challenge we faced is the misconception we had about the project, but that misconception was cleared with the meeting with our supervisor. But this made us a bit behind on our project according to the project plan.

Group 41: Shen Reddy and Dylan Baker

- last week we investigated different options for our microcontroller and have come to a decision on which one we'll be using. We also investigated different machine learning algorithms that could be used for anomaly detection in our system, such as CNNs and KNNs. We also started development on the laptop/pc ML system.
- This week we plan to order and obtain our microcontroller, continue with development on our laptop system, as well as obtain multiple dataset sources for image anomaly detection
- Currently we have not encountered any challenges yet with our project development

Group 47: Edward Ndlala and Lefa Mofokeng

- We have successfully implemented the backtracking algorithm and constant propagation algorithm. We made an extension cord for the smart PDU device.
- This week, we will be benchmarking the project matrixs. Test the developed algorithms and get some insights on how we can change the computational power
- We will also be acquiring login details for access to the smart PDU metrics system and run test measurements to determine its capabilities
- Encountered challenges regarding acquiring exact baseline values of memory usage, cpu usage and drive storage.

Week 3: Meeting Minutes

Meeting Date: 09/09/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Muhammad Bux
Minute Taker: Syed Anas

Overview: This is the third meeting. In it each group shares the progress they have made in their project in the last week. The group also discusses their plans for the project for the week and lastly the challenges that are faced by each group in the project are discussed.

Group 11: Edgar Ngoepe and Phuti Mokgehle

- Progress this Week:
- We successfully created the basic stickman model and developed the 3D climbing wall. These components are ready for integration to simulate a climber's movement on the wall.
- Challenges:
- Simulating dynamic movements with high realism remains a challenge. We need to make further enhancements to the physics-based model to achieve more accurate results.
- Plans for the Week:

- Explore different machine learning algorithms or models to optimize the climbing route, focusing on determining the most efficient paths based on simulated movements.
- Integrate the stickman with the 3D climbing wall to allow us to visualize the climber's movements on the wall.

Group 19: Akiva Levitt and Joseph Kaplan

- Last week:
- We created videos that will help users learn about pointers and guide them through using the software. Adding the videos to the VS Code extension with audio was very difficult however we eventually got it working. We also improved the UI of the software making it more dynamic and user friendly.
- This week:
- Need to fix functionality regarding visualizing arrays and pointers to arrays. Need to finialise the UI look and feel. Need to finalise the Google forms and questionnaires along with the flow of the tutorial. We would like to finish functionality so we can get ready to start experimenting with our product.

Group 22: Muhammad Bux and Syed Anas

- Last week: We built our own database from scratch as the database provided was not functional. Moreover, in the database, three tables were built, relationship between the tables were developed and data in a csv file was imported into the database. Furthermore, the dashboard was designed and developed and the export to a csv file functionality was developed and implemented.
- Challenges: One of the challenges that we faced last week was implementing the insert functionality in the dashboard as it was giving an error therefore data cannot be inputted from the dashboard. Moreover, as three tables are used in the database therefore; to show all the data in the database on the dashboard as one table is a challenge.
- This week: We will work on the insert functionality on the dashboard to make it work. Other functionalities such as the delete, edit and import from a csv file will be developed and implemented into the dashboard. Also, a one table database will be developed to allow for easy representation of the whole data on the dashboard. Lastly, investigation will be done to see which Machine Learning techniques need to be implemented on the data.

Group 41: Shen Reddy and Dylan Baker

- Last week: We worked on implementing a CNN algorithm to perform basic image anomaly detection and were able to successfully implement it. We were also able to implement TensorFlow to work on our laptop GPUs instead of the CPUs. We also looked at other potential algorithm options to implement other than CNNs, such as K-Nearest Neighbours. We also briefly looked at options for energy measurement.
- This week: We will continue to investigate options for algorithms, continue implementing algorithms and start on our energy measurement system design.
- Challenges: So far, the only challenge we faced was getting the right version of TensorFlow to work with the Windows CUDA toolkit, but we managed to resolve the issue.

Group 47: Edward Ndlala and Lefa Mofokeng

- Last week:

- We have successfully implemented the two new algorithms which are dancing link and genetic. Moreover, we did benchmark for the smart PDU and worked on automatically logging the benchmarking of the cpu usage and memory usage
- This week.
- We will try to test the algorithms and metrics. We will develop a sudoku generator. We will also develop a revised work breakdown structure.
- Challenges:
- The challenges that we faced are that we had constraints in logging the consumption on one pc since it couldn't access the PDU metrics page. Also, we couldn't log other PC metrics on the computer as well as it was slow to retrieve the data we required.

Week 4: Meeting Minutes

Meeting Date: 16/09/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Shen Reddy
Minute Taker: Dylan Baker

Overview: 5th week of the investigation project. The groups briefly discussed what they did in week 3, what their plans are for this week, and if they are struggling with any aspect of the project.

Group 11: Edgar Ngoepe and Phuti Mokgehle

- Progress This Week: We successfully:
- Explored two different machine learning algorithms (A* search and Dijkstra’s pathfinding algorithms) to determine the optimal climbing route.
- Integrated the stickman with the 3D climbing wall to visualize the climber's movements on the wall.
- Challenges: No changes arose for the past week.
- Plans for the Week:
- Simulate dynamic movements of the climber/stickman with high realism.
- Make further enhancements to the physics-based model to achieve more accurate results.

Group 19: Akiva Levitt and Joseph Kaplan

- Last week we fixed the pointer functionality regarding pointers and arrays, made each element of the array a separate block and allowed for the incrementing and decrementing of array pointers. We fixed dereferencing functionality for various pointer operations. We added error handling for C++ code that doesn’t compile. We also improved the UI, flow and questions for the tutorial and experiment part of our extension. We also added a few more tutorial videos.
- This Week: We need to run a pilot session with a user and our software so that we can get feedback for finalising our software, the UI, and the tutorial flow. Then we will hopefully be able to start testing our software and send out messages asking people to test it. We need to research Cronbach's alpha and p values relating to data and statistical importance.

Group 22: Muhammad Bux and Syed Anas

- Past week: We got the dashboard working. We investigated multiple data visualization techniques and implemented some of them into our dashboard. We implemented a bar graph and line graph which shows the homicides that took place over time. Also, a choropleth was implemented to show which provinces, the homicides have taken place. Moreover, we developed the add data, delete data and export to CSV format functionality in the dashboard.

- Challenges faced: The challenge that we are currently facing is to find the investigative element in our project. The investigative element of the project needs to be different for both partners which is suggested by Dr. Yuval.

- This week: More research will be done on different types of data visualization techniques and how we can implement them in our dashboard. Moreover, some other functionality such as finding duplicates, delete data manually and import to CSV file format will be implemented in the dashboard. Furthermore, more research will be conducted to find the investigative element of the project.

Group 41: Shen Reddy and Dylan Baker

- Last week we successfully implemented a version of a Convolution Neural Network (CNN) algorithm, for our PC system, to detect defects in images of screws and to detect clouds in the sky. We also obtained our microcontroller for the project and started investigating the software IDE for it. We implemented a basic demo application on the microcontroller just to understand how the system worked.
- This week we need to implement 2 more ML algorithms for our PC system (potentially KNN, SVM or Isolation Forests). We also need to try get an ML model on the microcontroller and we also need to devise an energy measurement system for both of our systems.
- The only challenge we faced was implementing code on the microcontroller due to deprecated methods in the software IDE.

Group 47: Edward Ndlala and Lefa Mofokeng

- Executed 2 more algorithms and recorded their execution time for sudoku boards of different difficulties

- Recorded baseline Cpu time and memory usage using the performance monitor

- Used the Hardware Monitor software to record baseline cpu tenperature
- Created the sudoku board generator.

- Automatically logging the metrics in c++ and HW monitor.

- Constraints included being unable to record values overtime as HW Monitor only logs data at specific time and not over a period.

- This week we will focus on testing the algorithms based on different difficulty and the number of sudoku boards.

Week 5: Meeting Minutes

Meeting Date: 23/09/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Edward Ndlala
Minute Taker: Lefa Mofokeng

Overview: 6th week of the investigation project. The groups discussed the progress they have made in the past week, the issues they encountered and what their objectives are in the following week.

Group 11: Edgar Ngoepe and Phuti Mokgehle

- Progress this week: We integrated a basic stickman model capable of climbing the wall.
- The model successfully simulates hand and foot movements as it ascends the climbing wall.
- The A* search algorithm has been completed, and it successfully determines the optimal climbing routes based on the designated holds.
- Challenges: One of the challenges we encountered was the appearance and movement of the stickman.
- Plans for the coming week: Our next focus is on perfecting the stickman's appearance and movement.
- We will compare the two implemented algorithms (A* search and Dijkstra) and finalize our analysis on the most efficient route-finding algorithm for the climbing wall.

Group 19: Akiva Levitt and Joseph Kaplan

- Last Week: we ran a pilot on two people with our software, got great feedback and then implemented the changes necessary. Thereafter we ran our software experiment on 5 people so far and have seen great results showing that the software improves user understanding of pointers. We did research into Conbachs alpha and p values on how we will determine the validity of our experiment.
- This week we will be finding participants to partake in our research experiment testing out our software.

Group 22: Muhammad Bux and Syed Anas

- Last week: We added more functionality into our dashboard such as being able to export the database tables as csv, import csv into a table in the database, being able to find duplicates in the table and delete those duplicates as well as keeping a record of the deleted duplicates by creating a duplicates table and storing it there. We also implemented 9 visualisation plots to see the correlation between the different fields in the table.
- This week: We have a meeting with Dr. Yuval Genga in which we will show a demo of our dashboard and depending on his criticism and advise on future improvements which we will add into our project. We will also do more research on the investigative element of the project and work on it this week.

- Challenges faced: The only challenge we are facing right now is time management. Even though we have added most of the functionalities in our project and visualisation techniques, we still might have to add a few more depending on Dr. Yuval Genga’s suggestions. While we also need to do the investigative element of the project and work on it to get results for our project.

Group 41: Shen Reddy and Dylan Baker

- Last week we were able to implement a basic computer vision model using a pure CNN algorithm as well as an autoencoder CNN algorithm on our laptop systems. We were able to compress the computer vision model and put it on our microcontroller. We also started developing software to display the results of microcontroller models on the LCD screen on the microcontroller.
- This week we need to compress the autoencoder algorithm and put it on the microcontroller, and we need to start benchmarking the performance on microcontroller. We also need to develop one more ML algorithm for both systems.
- The only challenge we faced was being able to use the LCD display screen. We need to investigate more online tutorials to help us get the screen properly set up for use.

Group 47: Edward Ndlala and Lefa Mofokeng

- Last week: Switched to HWinfo instead of HW Monitor which produces hardware metrics overtime and stores, data in a .csv file
- Recorded the baseline temperatures of CPU Package and Cores
- Tested the execution of the algorithms, the speed, memory usage and CPU consumption on different difficulties.
- No challenges of note in the prior week.
- This week: we are doing extensive testing and measurement. We will be running each algorithm on 2 PCs and collecting and analyzing the data.

Week 6: Meeting Minutes

Meeting Date: 30/09/2024
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Syed Anas, Shen Reddy, Dylan Baker, Edward Ndlala and Lefa Mofokeng
Meeting Chair: Joseph Kaplan
Minute Taker: Akiva Levitt

Overview: Each group discussed the achievements and accomplishments of last week as well as the plans for the next week and the problems encountered. It seems all projects are making good progress and reaching the end of development, moving onto the investigation part of the project.

Group 11: Edgar Ngoepe and Phuti Mokgehle

- Progress this week: Integrated a CNN for hold detection and scoring based on shape and size.
  Optimized the pathfinding algorithm by incorporating the climber’s reach, the distance between holds, and hold scores.
  Compared the performance of A* and Dijkstra algorithms for route optimization.Challenges:
- Challenges: The CNN accuracy is currently 68%. We plan to improve it by expanding the dataset.

- Plans for the coming week: Increase the dataset size and perform hyperparameter tuning to improve CNN accuracy. Finalize the integration of CNN-based hold scoring into both the A* and Dijkstra algorithms.Continue improving the stickman model after route optimization.

Group 19: Akiva Levitt and Joseph Kaplan

- last week: we recruited various students from computer science and engineering degrees from 1st to 4th year, to join in our experiment. We help an online experiment where we broke people up into breakout rooms and allowed each student to do the experiment at their own pace while being able to call for assistance. So far, we have had 14 students partake in our experiment. The results look promising. The average of the first test is around 66% and the average of the second test is 86%. So there has been an improvement after using our software.
- This week we will be trying to get more people to do our experiment and consolidating out results

Group 22: Muhammad Bux and Syed Anas

- Last week: We finished up on cleaning our dashboard so it looks neat and our dashboard is now ready to be shown on open day. We implemented a custom visualise technique where the user can define the x and y axis for the plot and what plot they would want to see the visualisation in. We also did research on the investigative element of the project. We found two investigative element.
- This week: The two investigative element that we researched are that Syed Anas(me) is working on implementing the different database designs (normalization, indexing strategies, and query optimization) and how it will impact the performance of data entry, updates, and data retrieval on our dashboard. My partner (Salman Bux) is working on comparing different plotting libraries (e.g., Plotly, Matplotlib, Bokeh) to evaluate which is most efficient or provides the best user experience for data visualization in our dashboard.
- Challenges faced: The challenge we faced was deciding which is the best investigative topic in our project. As we had to see which investigative topic will give us the best results and good enough results which we can write in our reports.

Group 41: Shen Reddy and Dylan Baker

- Last Week we were able to adapt our basic computer vision model to work for anomaly detection. We tested the model on a laptop using a dataset of manufactured screws and were able to get an accuracy of around 77%. We were also able to compress the new model on run it on our microcontroller and have it accurately identify anomalies in our dataset.
- This week we need to optimise our 3 algorithms to try increase our accuracy to the 80% range as well as ensure that they can work for multiple datasets. We will need to recompress the models again and run them on the microcontroller. We will also be taking energy usage measurements for our 2 systems.
- We have not encountered any major challenges in the past week.

Group 47: Edward Ndlala and Lefa Mofokeng

- 1. Took temperature distribution for solving one board.
- 2. We had a setback that forced us to review our algorithms and methodology. This is due to us encountering inconsistent results with the algorithms' memory records.

- 3. This week we will be testing and performing analysis of the captured data using our updated methodology and reviewed DLX

- Challenges faced: The challenge we faced last week was cleaning the data set that was provided to us by Dr. Broadie but we were able to be done with it in the weekend. Except for that we did not face any other challenges.

Group 41: Shen Reddy and Dylan Baker

- Last week we were able to successfully run all of our algorithms on both our PC and microcontroller systems. We then collected performance metrics for both systems. Our results (in the form of a confusion matrix) showed that we were able to achieve the same ML performance metrics on both systems.
- This week we need to finish collecting our energy measurements for both systems and finish printing our poster for the open day.

Group 47: Edward Ndlala and Lefa Mofokeng

Last week:

- we reviewed our algorithms and did some refactering
- We implemented web API which can extract information from the Smart pdu faster.

This week: ρ □ M δ □

- analysing the results, we will look at creating our poster and preparing for presentation day.
- Challenges faced: the challenge we fast was that at first, we only logged at 1 minute. The web api can log at 1 second. This affected our testing results that we got. Thus we had to start testing afresh.

Week 8: Meeting Minutes

Meeting Date: 14/10/2024 **g Minuy**
Cohort Participants: Akiva Levitt, Joseph Kaplan, Edgar Ngoepe, Phuti Mokgehle, Muhammad Bux, Sy

! □ ρ

- Last week: We developed the homicide media tracker and implemented all the required functionality. While implementing the functionality, we found bugs which we then fixed. We ironed out the dashboard and made it look professional for open day. Lastly, we designed and printed out our poster for open day.
- This week: We are now finalising the testing and the results we are getting from it. After that we will do the report for the project in the week.
- Challenges: Last week, the challenges we faced were fixing some of the bugs in our code as well as finalising the testing methodology as our metrics to test our investigative element in the project needs to be independent of the machine we are running the dashboard on, therefore some of our testing methodology was not compiling to that need.

Group 41: Shen Reddy and Dylan Baker

- Last week we created our poster and presented it at the open day. We also spent some time revising some of the energy measurements we had taken for our microcontroller system.
- This week we are going to take one more set of energy measurements to ensure the integrity of the results we captured last week. Both of us will continue to work on our individual reports.

Group 47: Edward Ndlala and Lefa Mofokeng

- Last week we took final Power measurements to ensure that they are consistent with previously taken measurements and to account for any false readings such as when the PC is in its sleep mode while running the algorithms. We then created our poster and presented it at the open day using the final results.
- This week we are going to perform a broader analysis of our results and discuss them.
- Both of us will continue to work on our individual reports.