Final Report– Social Media
How big is our digital footprint?

Cardiff University School of Computer Science and Informatics

CM3203 – Individual Project – 40 credits

Author: Thomas Doyle

Supervisor: Dr Jianhua Shao

Abstract

There are currently around 45 million Facebook users in the UK alone [1] and the average person has 7 social media accounts [2]. Platforms such as Facebook, Twitter, Instagram and LinkedIn are becoming increasingly intrinsic to our everyday life and can contain a wealth of information about us. This has subsequently made social media an attack vector that we cannot ignore. Dr Mike McGuire, a Senior Lecturer in Criminology at the University of Surrey, conducted an extensive academic study to publish Social Media Platforms and the Cybercrime Economy. A key finding was that "Cybercrimes enabled by social media are generating at least $3.25 billion in global revenue annually"[3].

In regard to social media, these attacks can manifest from a number of issues. Often privacy settings are misunderstood and this leads to information becoming publicly available. In turn this leads to social media reconnaissance and user profiling to create a socially engineered attack. Even when a Facebook account is set to private there are still posts made when a user updates their 'profile picture' or 'cover photo'.

Reconnaissance could even go one step further and look for information publicly available on the internet. This could potentially uncover information such as a person's publications, businesses, membership to a society and even news articles they were mentioned in. This process of collecting publicly available information is known as Open-Source Intelligence (OSINT)[4].

In this project I will develop an OSINT application to perform Facebook and Google searches on individuals and explore the extent of information publicly available on us.

Acknowledgements

Firstly, I would like to thank my project supervisor Dr Jianhua Shao for his invaluable help and continued support throughout this whole project. He has provided guidance and mentorship at pivotal moments, which I greatly appreciate. I would also like to thank my placement company ZenRS, they have assisted me greatly in developing a variety of skills and attributes, such as communication, problem solving, creativity and self-management to name just a few. They also supported me to work with a great deal of autonomy during the COVID crisis and ultimately my experience with them has helped me greatly to take on the challenge of this project. Lastly, I would like to thank my family for their endless support throughout my time at University, they have not only provided financial assistance to me but have also given me moral support when the need has arisen. Again, I am grateful to them also.

Table of Contents

1. Introduction

1.1 Motivation

The motivation for this project has arisen as a result of an assignment I was tasked with during my ‘year in industry’. During my placement I was employed by the cyber security company ZenRS and on one occasion my manager tasked me to research an online resource called Hunter.io. He requested that I determine whether this tool could be beneficial to the company and thus utilised as part of our library of assets.

I established that Hunter.io was used primarily to assist individuals in detecting information associated with emails and domains. Whilst testing the efficiency and reliability of this tool I examined the ‘Email Verifier’ product within the program. This will provide information such as the inputted email's status, whether it accepts emails outside of its domain and if there is any reference to it being publicly available on the web. As part of my analysis, I inputted my University email address into the ‘Email Verifier’. To my amazement it provided a result and notified me that the email address had one reference. Upon further inspection I discovered that an email I had sent privately to a committee was now publicly available on the internet. I was concerned by this being freely available but also at the same time I was fascinated at how this could have occurred. What other information, which some people might construe as private and personal, is publicly available and can be obtained with relative ease. The result was that a seed was planted for my final project and my motivation was triggered.

1.2 Project Aim

This project intends to develop an OSINT application that would allow Facebook and Google searches to be performed. This will streamline the OSINT process by collating potentially useful data into one application where they can mark information as useful and export the findings. Through doing so the need to create documents, record findings, and the chance of revisiting webpages multiple times is removed. The project also intends to explore the extent of information in respect of an individual(s) being publicly available. Through conducting searches using volunteers’ details and requesting that those same volunteers evaluate the validity and accuracy of a profile created using the information obtained.

1.3 Beneficiaries

If this OSINT tool were to be made available outside of the scope of this project, some potential beneficiaries would be professions that already utilise OSINT tools, such as journalists, private investigators and private businesses working within the intelligence sector. Private businesses utilize OSINT tools for the detection of corporate data leaks, investigation of competitors, public perception forecasting and predicting future market trends. OSINT tools are also very commonly used within the profession of United States bondsmen. They use OSINT when investigating and attempting to locate individuals who have ‘skipped’ bail. OSINT is even used by military forces. The United States Air Force were able to locate an Islamic State headquarters through photos posted to social media and comments made within public forums [5].

1.4 Implementation Approach

By taking a phasal and sequential approach to implementation, development knowledge in initial stages can be used in subsequent phases. This waterfall method is suited for smaller projects where requirements are well defined. This approach also allows for flexibility in the timeline, allotting more time to stages if needed and in the case of failure, there is still evolution that may still be useable. Therefore, by firstly creating and testing one of the files that collects data, this would indicate features to be added in future development and how the following file and application should behave.

1.5 Important Outcomes

The project manages to achieve several outcomes:

- The development and implementation of a JavaScript program to collect data from a user's Facebook profile, when given a URL.
- The development and implementation of a JavaScript program to collect search results from Google, when given a search term.
- Development of a Java application to enable a user to action both the Facebook and Google programs, interact with the data and export the findings.
- Exploration of the extent of information in respect of an individual(s) being publicly available.

2. Background

2.1 OSINT

Open-source intelligence is intelligence that is collected from online sources that are openly and readily available to the public. It is a very valuable data source and the more information that is collated in respect of an individual then the fuller and more accurate the profile that can be built will be. It is one of the simplest, most rewarding and straightforward modes of obtaining intelligence. However, it does have it challenges. With such an abundance of data available via the internet it is often difficult to find known-unknown's and determine if they are relevant. OSINT is used by a wide variety of organisations who collect and analyse data to create information which would be of use to them. An example may be whereby a company conducts background checks on an individual who they are considering employing. With this information they can make an informed decision regarding the employability of the person concerned. Ultimately this could lead them on to either maintaining or creating a competitive edge against their competitors.

In the case of reconnaissance conducted on an individual, OSINT can arise from various sources but the way we go about procuring them is mainly through search engines. Google is the most commonly used search engine with a staggering 92.26% market share [6], it is clear from this statistic that most users have a preference to use Google. However, depending on the search engine used, search results can differ greatly. This is due to companies using different algorithms to obtain their search results. Also, with over 3.5 billion Google searches performed daily [7], you would believe that users have a good understanding of how to formulate queries. However, the way we create our queries can also have an impact on the results we get.

2.2 Social Media

Social media, such as Facebook, can provide a wealth of information. For example, it can supply details of both immediate and extended family members, it can furnish particulars of friends and relationships, places where individuals have lived and worked, schools attended and also links to other social media accounts. Settings are often confusing for users of such platforms and this can lead to a lapse in privacy and a user's esoteric information being publicly available to other users. Indeed, even in the case of a user having strict privacy settings implemented there are still ways of obtaining information.

The amount of people using social media is ever increasing, not only are users possibly unwittingly sharing information about themselves but they are also giving away information about others, again unknowingly on occasions. Even if a user's account has all privacy settings operating there are still a number of ways whereby more information about them can be discovered. For instance, when a user's friend list is visible and accessible to non-friends, we can extract information through their friend's accounts and use frequency analysis to make inferences about the primary target. For example, if in a target's friend list of 1,000 people there are 250 users who list their hometown as Cardiff then we can infer that the target themselves are from Cardiff. Furthermore, if there are 50 users who list their current location as Leeds and that they attend the University of Leeds, we can infer that the target has moved to Leeds to attend the University there. Even if a user's friend list is not visible, we can still perform social network simulation. This is where we assume any users that have liked the targets profile picture or cover photo are friends. With this partial friend list, we are then able to conduct further searches and over time we can start to build a graph comprised of nodes and edges. Once the friends have been exhausted and we have an extensive list we can then perform frequency analysis to attempt to retrieve information. The edges can have weightings and be loaded into a physics simulator, this will naturally form groups, and appear as shown below:

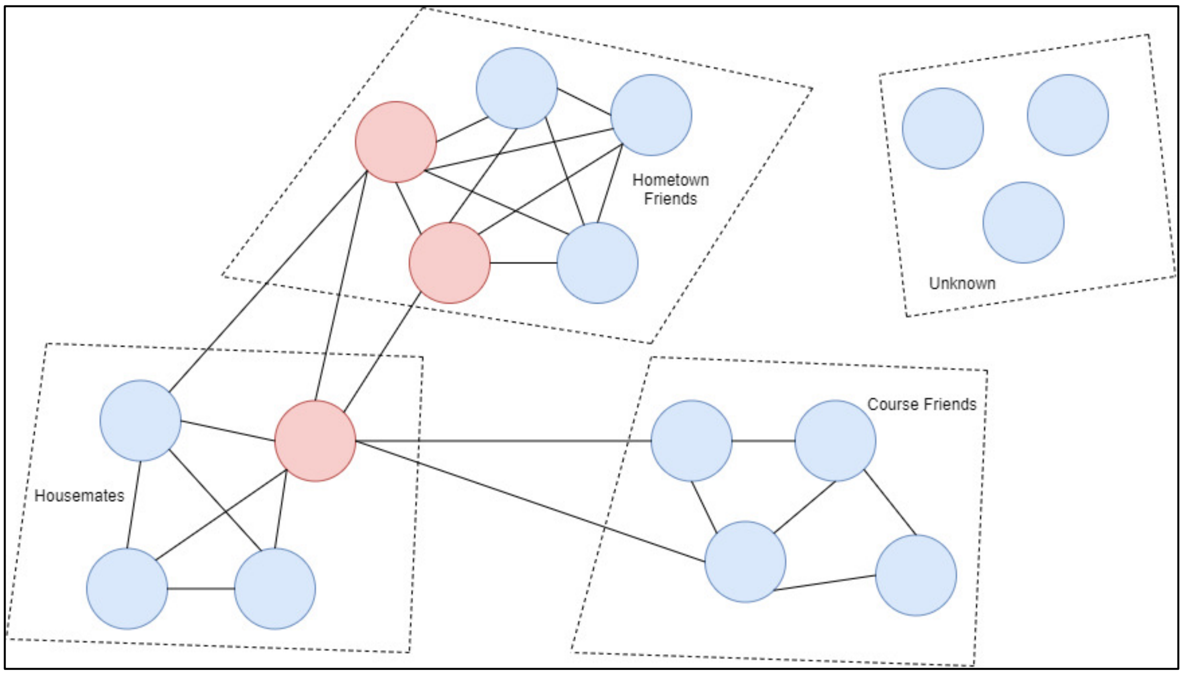


Figure 1: Example Social Media Simulation Graph

2.3 Existing Products

Most OSINT tools are created as a hobby by developers in their spare time, they utilise application programming interfaces (API's) and are uploaded to GitHub. This, unfortunately, usually leads to potentially useful software being left non-maintained, as APIs become deprecated and no longer useable. One tool that I did identify as potentially similar to my proposed software was 'skiptracer' on GitLab[1]. It allowed queries by phone, email address, screen names, real names, home addresses etc. and was aimed at United States users, where individual's information is more freely available. It was a popular OSINT tool due to the fact that it gathered data from a variety of sources. However, this project has not been updated for over 2 years, it encountered issues due to the code being written in Python 2.7 and many of the API's changing or becoming deprecated. One of the market leaders in OSINT tools is Maltego. It describes itself as "an open-source intelligence (OSINT) and graphical link analysis tool for gathering and connecting information for investigative tasks". Although there are options for a free 'community' edition, a lot of the useful functionality needed comes with the 'pro' version of the software and comes at a price of $1000 per user, per year.

2.4 Problems Identified

Currently there are no current OSINT tools that enable the collection of open-source information through Facebook and Google. Instead, it is achieved by manually visiting webpages to collect and record findings in a document. Information can come from a number of sources and often URLs appear in multiple search results. Users can end up visiting the same webpages multiple times. This can be avoided and the whole process streamlined by automating the collection, removing duplicate URLs and exporting findings.

3. Specification and Design

3.1 Requirements

The application and subsequent programs should fulfil the following requirements:

- The user should be able to input a Facebook profile URL and retrieve information about a user's hometown, current location, places of work and education if they are listed.
- The user should be able to interact with the data and modify it if necessary.
- The user should be able to input a search term and retrieve resulting URLs from Google.
- The user should be able to label URLs as relevant or not.
- The user should be able to add context to a URL.
- The user should be able to export their findings to a text file.
- The Facebook script should be able to be actioned from within the application.
- The Google script should be able to be actioned from within the application.

[1] https://gitlab.com/illwill/skiptracer

3.2 Architecture

The whole application is comprised of 3 files and follows an event driven master slave[2] architectural pattern. The ‘App.java’ file is run to start the OSINT tool. Whereas the 'GoogleScraper.js' and 'FacebookScraper.js' scripts are microservices that are tasked with the retrieval of information. Within the tool users can navigate between 3 screens, the home page, Facebook and Google. From within their respective screens the Facebook and Google collection scripts can be actioned. The OSINT tool will pause and allow time for the script to complete. When the script is finished it outputs text files containing the data. The application resumes and reads in this data from the text files outputted and the data is displayed for the user to interact with and export. The tool is intended to be used in collaboration with the Command Prompt as the current state of the system is regularly output to help the user understand the behaviour of the system.

3.3 Design

The OSINT tool’s graphical user interface is shown in Figure 2, Figure 3 and Figure 4, and has a brief explanation of the dynamic behaviour of the system:

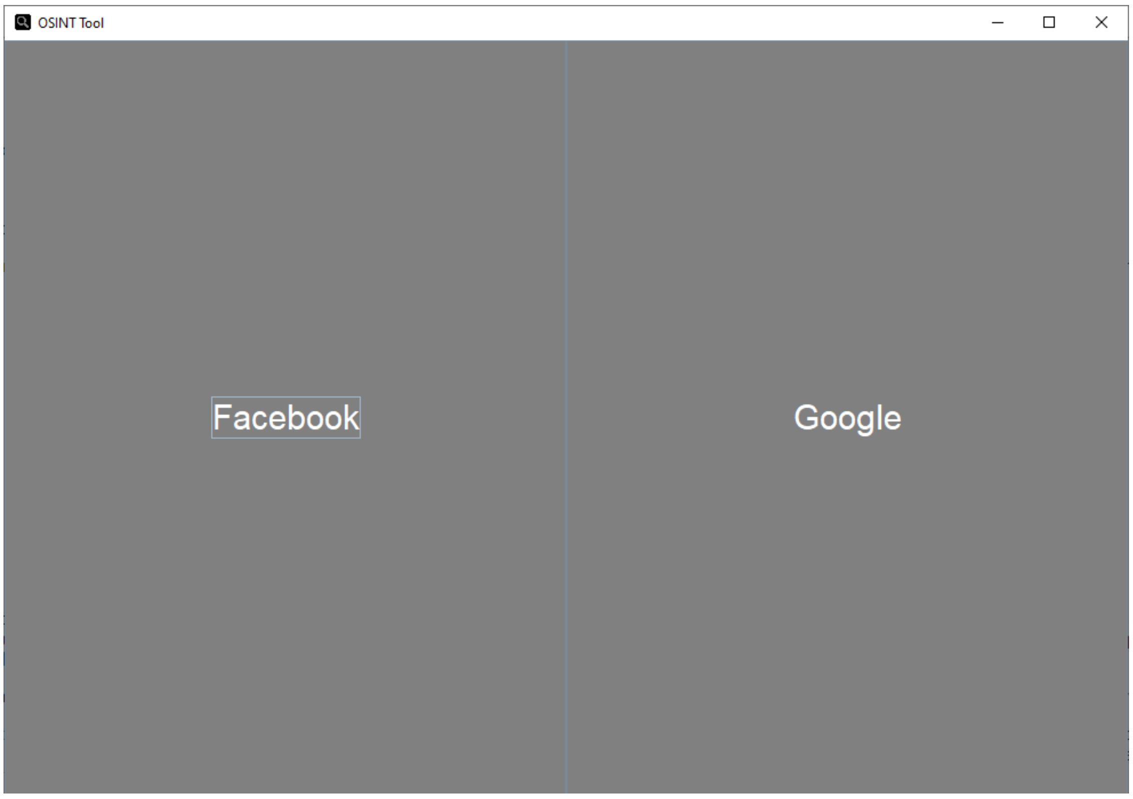


Figure 2: OSINT Tool Home Screen

The home screen is very minimalistic in that it only features two buttons, that when clicked will take you to the respective page.

[2] https://www.differencebetween.com/difference-between-master-and-vs-slave/

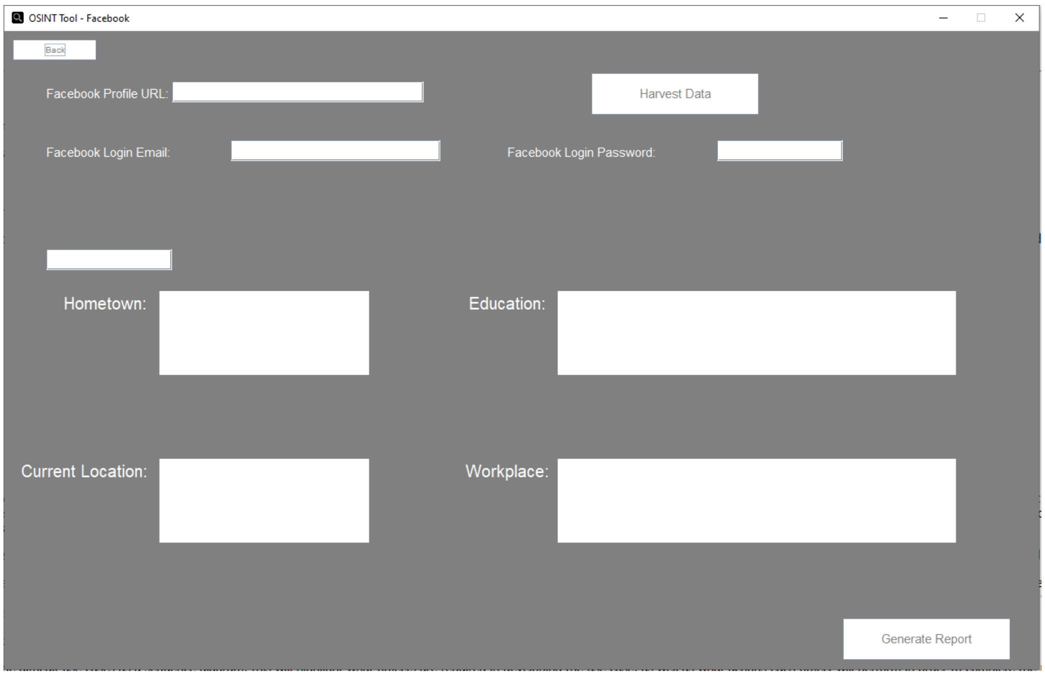


Figure 3: OSINT Tool Facebook Screen

The Facebook screen is made up of labels, text fields, text areas and buttons. To begin a search for information available on a Facebook profile the user must provide a URL to the profile, a valid email and password for the script to login to Facebook. Failure to provide one or more of these will result in a message being logged to the Command Prompt, informing the user of the missing requirements. Once the 'Harvest Data' button is clicked the application is paused and the script carries out its task. The application then resumes, data is read in and is displayed in the text areas ready for the user to interact with and export using the 'Generate Report' button.

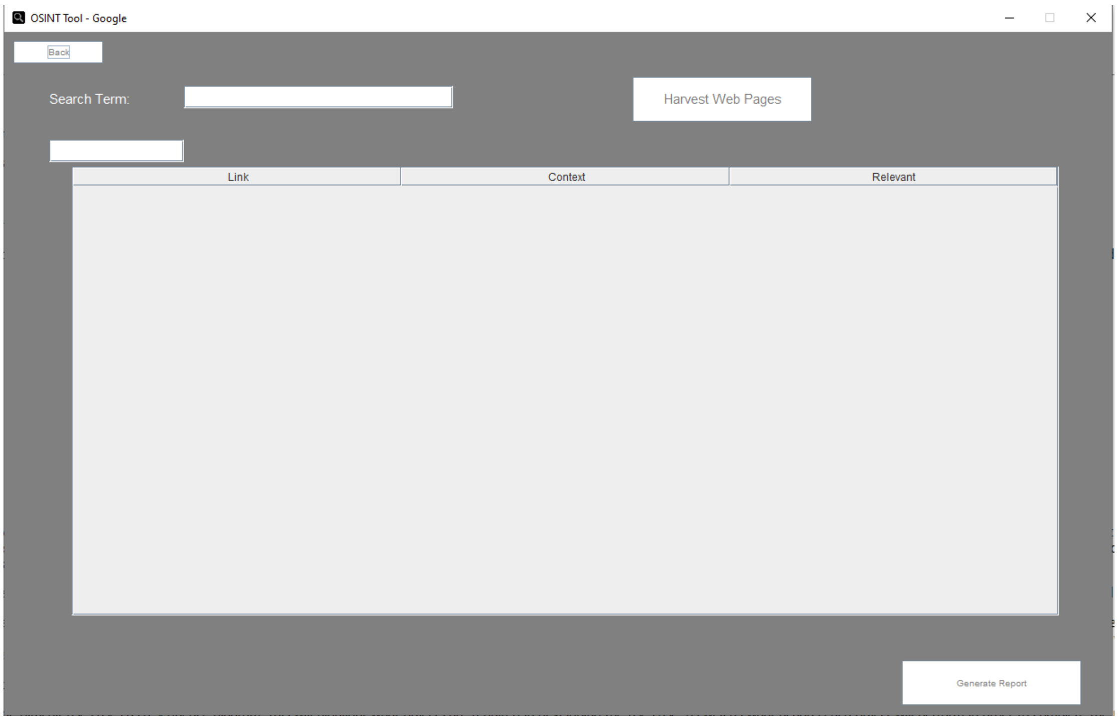


Figure 4: OSINT Tool Google Screen

The Google screen is made up of labels, text fields, buttons and a table. To begin a search a user must provide a search term. The application will check to ensure it is present before proceeding. Once the 'Harvest Web Pages' button is clicked the application is paused and the script retrieves web pages relating to the search term. Once the application resumes, the data is read in and displayed in the table. Within the table the users can add comments to the 'context' column which will be included when exporting findings. The 'relevant' column is a drop-down box with options 'Yes' and 'No'. If a row within the table is marked as not relevant it will not be exported when generating the report, it will only be exported if it is marked 'Yes' or left blank.

3.4 Justification

The OSINT tool achieves its aims through functional decomposition. The task of retrieving and outputting information is delegated to one of the 2 scripts. With an event driven master slave architecture that calls upon microservices, the system as a whole is simplistic and easy to understand. It is scalable as other microservices can be added in future. There is also fault isolation in that if one microservice fails, the application and other microservice remain mainly unaffected.

The design of the application is minimalistic and task orientated. Too much functionality on a page can be overwhelming to users. Both the Google and Facebook screens have a visual hierarchy aided by the proximity and spacing of elements into 3 stages. These being the search to be conducted, results and interaction with data and finally the exporting of findings. The use of a table within the Google screen means that large amounts of data are organised into a readable format, allows users to quickly see results, add comments into the context column and mark if they are relevant intuitively.

4. Implementation

4.1 Programming Languages

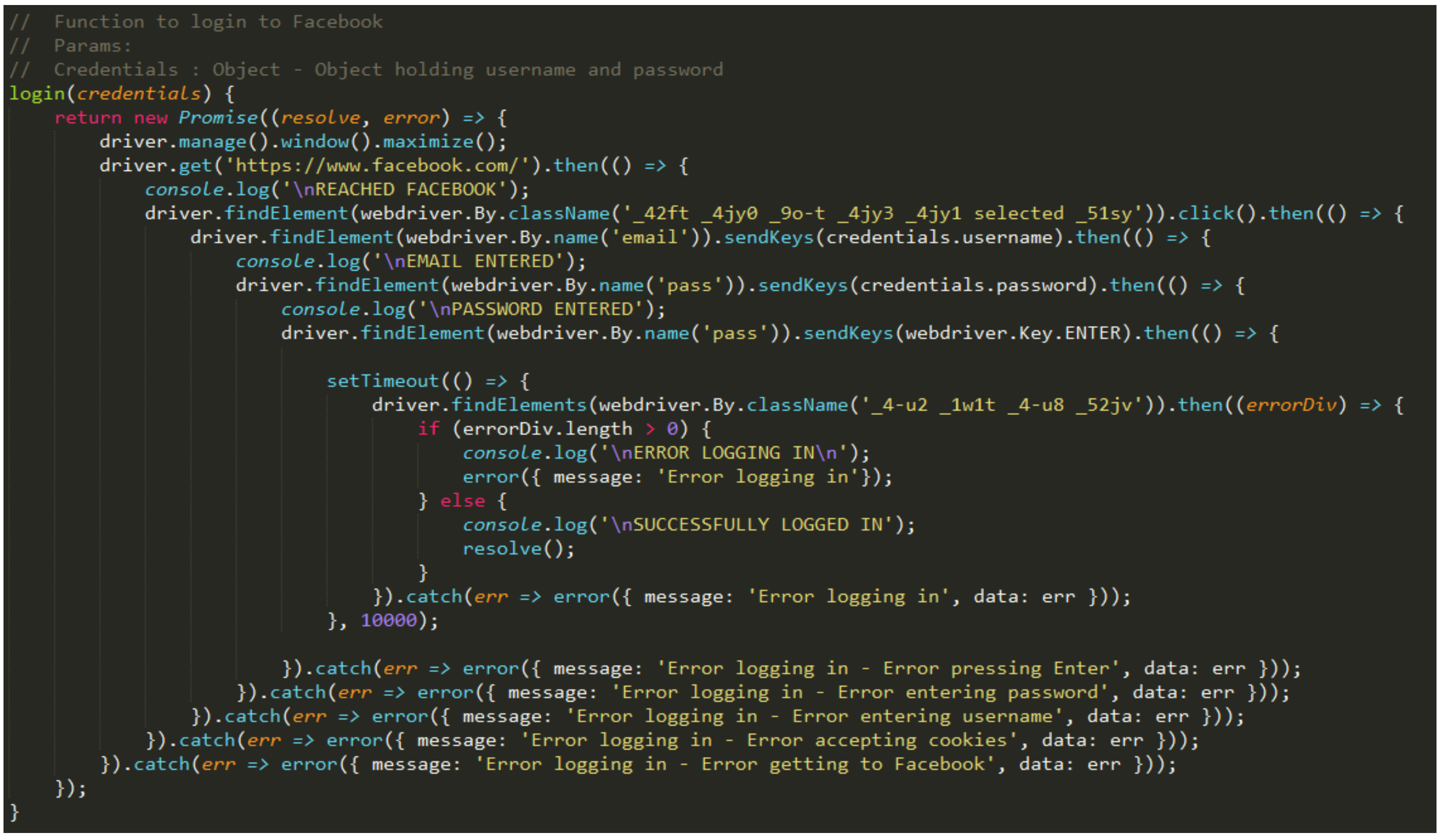
There a number of ways the application could be implemented. An ideal option is to utilise APIs in order to collect information. This would be efficient and provide ease of integration. However, there are a number of issues with the Facebook and Google API's. Firstly, there are rate limits and costs involved. Secondly, the Facebook API no longer supports API calls to retrieve places of work or education[3]. As these are key search areas, this option is rendered unusable in this project. In order to collect information from Facebook and Google the creation of purpose made scripts are needed. Having worked with NodeJS and Selenium WebDriver on my year in industry, I felt confident that I could achieve my aim by using these tools. NodeJS is a back-end JavaScript runtime environment that executes outside of a web browser. NodeJS also offers built in support for package management using Node Package Manager (NPM). Selenium WebDriver is a well-established tool for automating tasks, this factored into my decision to progress with this option.

[3] https://developers.facebook.com/docs/graph-api/reference/user

Extensive in-depth research was also made in respect of building an application and graphical user interface (GUI) to establish what option would best suit my objective. The viability of Angular was considered however, unfortunately, investigation discovered that Angular is used to build web applications and is not easy to learn. As a result, alternative options were discussed with my supervisor and he suggested that I explore using Java Swing as a more suitable. This is a Java GUI toolkit for building window-based applications. Java Swing is also a time-tested Java tool, it is well documented and offers a comprehensive amount of support online. Considering these attributes and factoring in my experience of completing a Java module in my first year at university, I felt more confident proceeding with this option.

4.2 Facebook Script

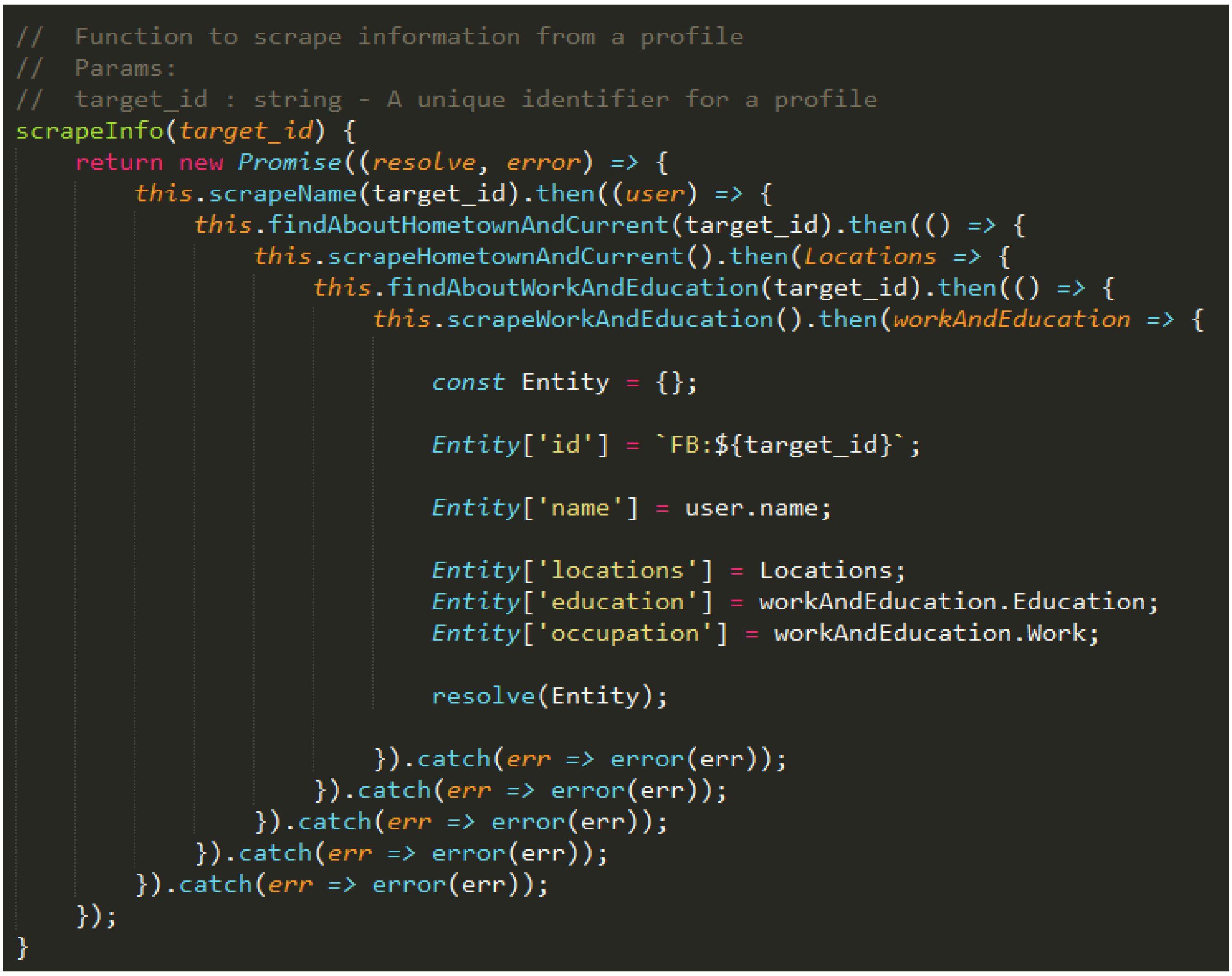
The 'FacebookScraper.js' file is comprised of 13 functions. The requirement of retrieving information from an individual's Facebook profile is achieved through functional decomposition. Sections of code that are of particular importance are shown in Figure 5 and Figure 6.



```
//  Function to login to Facebook
//  Params:
//  Credentials : Object - Object holding username and password
login(credentials) {
    return new Promise((resolve, error) => {
        driver.manage().window().maximize();
        driver.get('https://www.facebook.com/').then(() => {
            console.log('\nREACHED FACEBOOK');
            driver.findElement(webdriver.By.className('_42ft _4jy0 _9o-t _4jy3 _4jy1 selected _51sy')).click().then(() => {
                driver.findElement(webdriver.By.name('email')).sendKeys(credentials.username).then(() => {
                    console.log('\nEMAIL ENTERED');
                    driver.findElement(webdriver.By.name('pass')).sendKeys(credentials.password).then(() => {
                        console.log('\nPASSWORD ENTERED');
                        driver.findElement(webdriver.By.name('pass')).sendKeys(webdriver.Key.ENTER).then(() => {

                            setTimeout(() => {
                                driver.findElements(webdriver.By.className('_4-u2 _1w1t _4-u8 _52jv')).then((errorDiv) => {
                                    if (errorDiv.length > 0) {
                                        console.log('\nERROR LOGGING IN\n');
                                        error({ message: 'Error logging in'});
                                    } else {
                                        console.log('\nSUCCESSFULLY LOGGED IN');
                                        resolve();
                                    }
                                }).catch(err => error({ message: 'Error logging in', data: err }));
                            }, 10000);

                        }).catch(err => error({ message: 'Error logging in - Error pressing Enter', data: err }));
                    }).catch(err => error({ message: 'Error logging in - Error entering password', data: err }));
                }).catch(err => error({ message: 'Error logging in - Error entering username', data: err }));
            }).catch(err => error({ message: 'Error logging in - Error accepting cookies', data: err }));
        }).catch(err => error({ message: 'Error logging in - Error getting to Facebook', data: err }));
    });
}
```

Figure 5: 'FacebookScraper.js' Code

One of the first functions called is associated with navigating to Facebook and logging in with credentials provided by the user is shown in Figure 5. This involves identifying key elements within the HTML code of the webpage to input an email address and password, attempting to login and detect any errors that have occurred during the process.



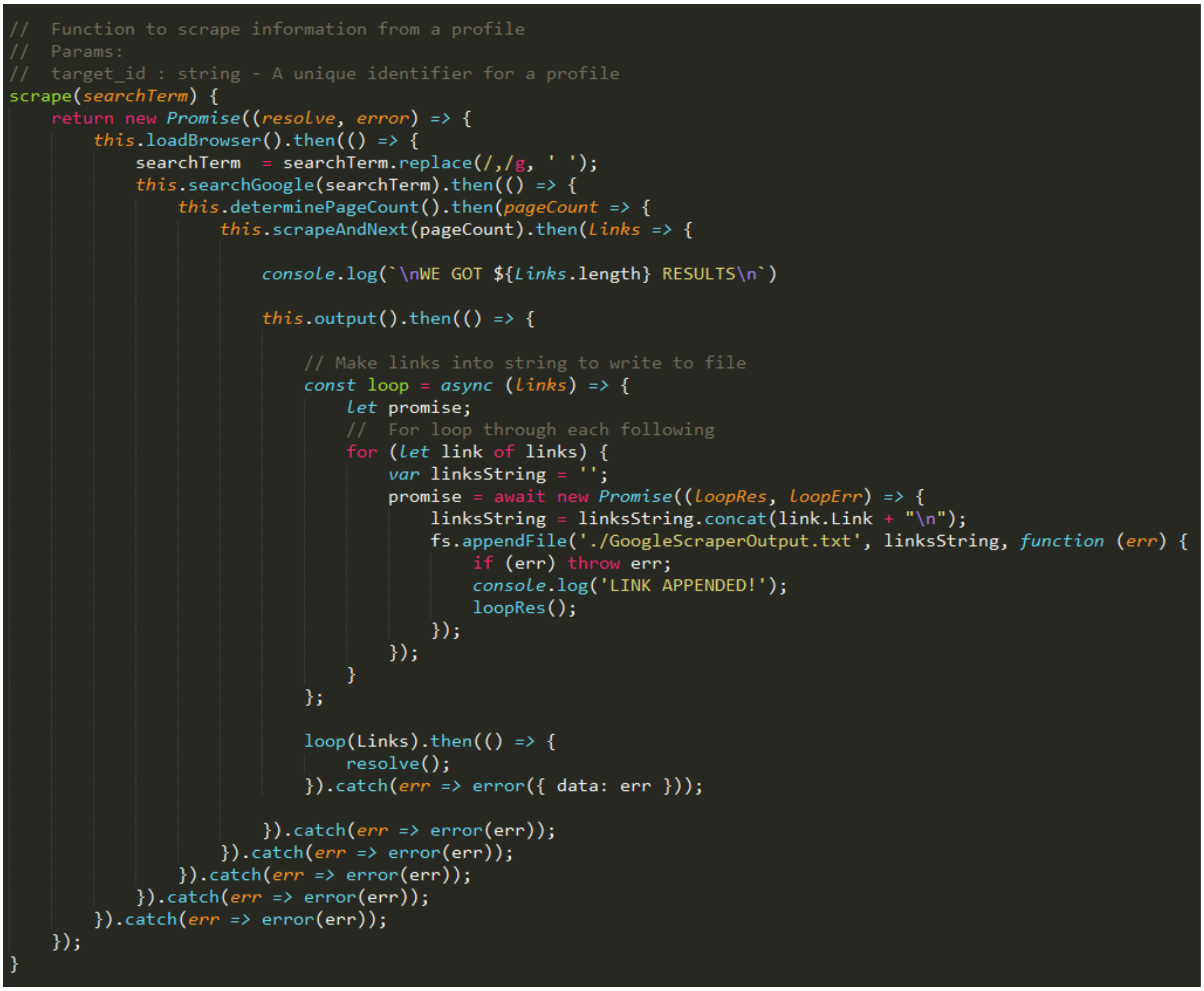
```
//  Function to scrape information from a profile
//  Params:
//  target_id : string - A unique identifier for a profile
scrapeInfo(target_id) {
    return new Promise((resolve, error) => {
        this.scrapeName(target_id).then((user) => {
            this.findAboutHometownAndCurrent(target_id).then(() => {
                this.scrapeHometownAndCurrent().then(Locations => {
                    this.findAboutWorkAndEducation(target_id).then(() => {
                        this.scrapeWorkAndEducation().then(workAndEducation => {

                            const Entity = {};

                            Entity['id'] = `FB:${target_id}`;

                            Entity['name'] = user.name;

                            Entity['locations'] = Locations;
                            Entity['education'] = workAndEducation.Education;
                            Entity['occupation'] = workAndEducation.Work;

                            resolve(Entity);

                        }).catch(err => error(err));
                    }).catch(err => error(err));
                }).catch(err => error(err));
            }).catch(err => error(err));
        }).catch(err => error(err));
    });
}
```

Figure 6: ‘FacebookScraper.js’ Code

One of the primary functions called is shown in Figure 6. Firstly, the name on the profile is captured, before progressing onto loading the web page containing the individual’s hometown and current location. If a hometown and/or current location are present these will also be captured, and then progress on to places of education and work to capture those. Once the collection of information is complete, the script will then output files containing the data for the OSINT application to read in and display to the user.

4.2 Google Script

The ‘GoogleScraper.js’ file is comprised of 9 functions and acts in a similar fashion to the ‘FacebookScraper.js’ file in that it achieves the requirement of retrieving potentially useful URLs through functional decomposition. A section of code that is of particular importance and demonstrates the functional decomposition is shown in Figure 7.



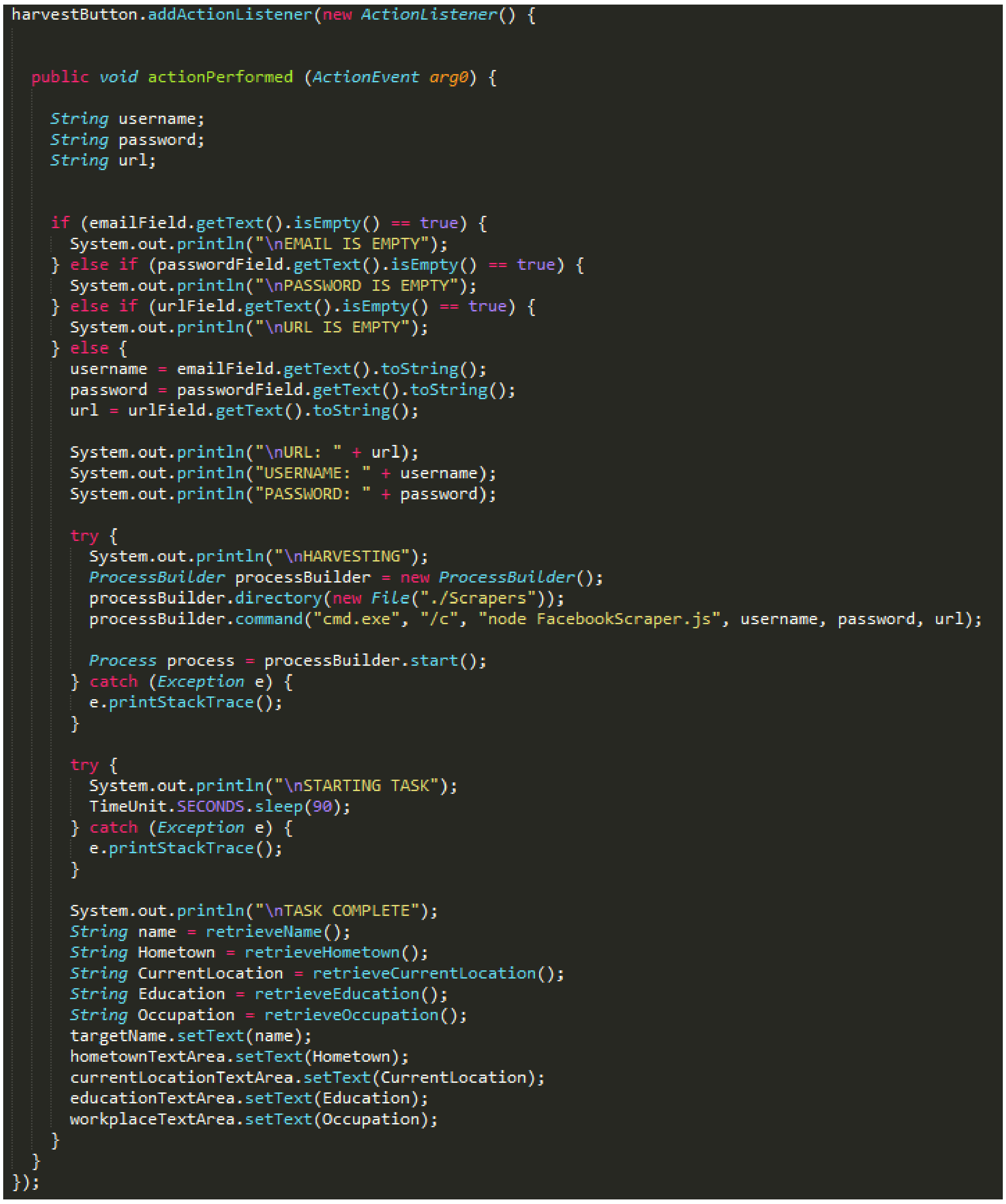
```
//  Function to scrape information from a profile
//  Params:
//  target_id : string - A unique identifier for a profile
scrape(searchTerm) {
    return new Promise((resolve, error) => {
        this.loadBrowser().then(() => {
            searchTerm  = searchTerm.replace(/,/g, ' ');
            this.searchGoogle(searchTerm).then(() => {
                this.determinePageCount().then(pageCount => {
                    this.scrapeAndNext(pageCount).then(Links => {

                        console.log(`\nWE GOT ${Links.length} RESULTS\n`)

                        this.output().then(() => {

                            // Make links into string to write to file
                            const loop = async (links) => {
                                let promise;
                                //  For loop through each following
                                for (let link of links) {
                                    var linksString = '';
                                    promise = await new Promise((loopRes, loopErr) => {
                                        linksString = linksString.concat(link.Link + "\n");
                                        fs.appendFile('./GoogleScraperOutput.txt', linksString, function (err) {
                                            if (err) throw err;
                                            console.log('LINK APPENDED!');
                                            loopRes();
                                        });
                                    });
                                }
                            };

                            loop(Links).then(() => {
                                resolve();
                            }).catch(err => error({ data: err }));

                        }).catch(err => error(err));
                    }).catch(err => error(err));
                }).catch(err => error(err));
            }).catch(err => error(err));
        }).catch(err => error(err));
    });
}
```

Figure 7: ‘GoogleScraper.js’ Code

It first loads a browser, handles popup iFrames and loads the Google search engine. The search term is entered and the number of pages containing results is determined. This helps the script in deciding the number of pages to scrape and therefore has the ability to know when it is on the last page of results and no more URLs are available. The script then begins collecting the URLs. It will navigate to the next page through the use of ‘next’ buttons and begin collecting URLs from the new page of results. Once collection is complete the script will then create a file to output the results to. Through use of a ‘for’ loop the results are appended to the output file, ready for the OSINT application to read in.

4.3 Java Application

The ‘App.java’ file is made up of 9 functions. 3 of these are responsible for creating the home, Facebook and Google screens. The other 6 carry out the operation of reading in data from the files outputted by the scripts. With the use of the GUI widget toolkit Java Swing, elements such as JLabel’s, JTextField’s and JButton’s are at fixed positions within a non-resizable frame. A section of code that is of particular interest and complexity is shown in Figure 8.



```java
harvestButton.addActionListener(new ActionListener() {


  public void actionPerformed (ActionEvent arg0) {

    String username;
    String password;
    String url;


    if (emailField.getText().isEmpty() == true) {
      System.out.println("\nEMAIL IS EMPTY");
    } else if (passwordField.getText().isEmpty() == true) {
      System.out.println("\nPASSWORD IS EMPTY");
    } else if (urlField.getText().isEmpty() == true) {
      System.out.println("\nURL IS EMPTY");
    } else {
      username = emailField.getText().toString();
      password = passwordField.getText().toString();
      url = urlField.getText().toString();

      System.out.println("\nURL: " + url);
      System.out.println("USERNAME: " + username);
      System.out.println("PASSWORD: " + password);

      try {
        System.out.println("\nHARVESTING");
        ProcessBuilder processBuilder = new ProcessBuilder();
        processBuilder.directory(new File("./Scrapers"));
        processBuilder.command("cmd.exe", "/c", "node FacebookScraper.js", username, password, url);

        Process process = processBuilder.start();
      } catch (Exception e) {
        e.printStackTrace();
      }

      try {
        System.out.println("\nSTARTING TASK");
        TimeUnit.SECONDS.sleep(90);
      } catch (Exception e) {
        e.printStackTrace();
      }

      System.out.println("\nTASK COMPLETE");
      String name = retrieveName();
      String Hometown = retrieveHometown();
      String CurrentLocation = retrieveCurrentLocation();
      String Education = retrieveEducation();
      String Occupation = retrieveOccupation();
      targetName.setText(name);
      hometownTextArea.setText(Hometown);
      currentLocationTextArea.setText(CurrentLocation);
      educationTextArea.setText(Education);
      workplaceTextArea.setText(Occupation);
    }
  }
});
```

Figure 8: ‘App.java’ Code

The code shown in Figure 8 is actioned when the ‘Harvest Data’ button is pressed within the Facebook screen. Firstly, it ensures that none of the text fields are left blank. It then actions the ‘FacebookScraper.js’ script using a ‘ProcessBuilder’ and passes parameters from the text fields.

The application pauses to give the script enough time to complete and then calls functions to read in the findings.

4.4 Problems Encountered

During the implementation process, there were a number of challenges that needed to be overcome. Initial plans were to action the Facebook and Google scripts with the use of Nashorn, a JavaScript engine, and receive the outputs as a HashMap. However, as of Java 11 the Nashorn engine was deprecated and then removed in Java 15. It is for this reason that the OSINT tool application actions the scripts through the use of a ProcessBuilder and has them output files so that the data can be read in. Whilst solving this challenge, the initial plan was to output a JSON object and have the application read from a single file. However, Java requires a library named org.json. With no prior experience working with libraries in Java, no Java Project within an IDE such as NetBeans or Eclipse and limited time I decided to have the scripts output text files and have the application read in the information by each new line.

5. Results and Evaluation

In order to test the viability of the OSINT tool and ensure it had met the primary project aim, I decided I would evaluate the tool by testing its ability to collect information from a number of volunteers. I did this by recruiting handpicked participants that I knew from attending Cardiff University. They were all aware of me working within the cyber security field during my year in industry and I was confident that this would provide them with belief and reassurance regarding the actions that I would be taking. I chose persons known to me as I believed recruiting random volunteers from Cardiff University, via a mailing list, would not be very successful. With no funding there was no incentive for individuals to participate and the project description could seem daunting or invasive at first glance. Participants would give me permission to carry out searches on them, using my OSINT tool application, and I would return a report containing information located which I believed to be pertaining to them. They would then mark its correctness and complete a questionnaire regarding the findings and their use of social media.

Participant 1 was given a report detailing a hometown and current location identified and a total of 24 URLs believed to be pertaining to them. The hometown, current location and 95.83% of the URLs provided were confirmed to be correct and accurate. Some of the information found included GCSE and AS level results, extra-curricular activities, hobbies and numerous social media accounts, some of which were identified as them using a ‘screen name’. One social media account found was a Reddit profile and contained 58 posts and over 900 comments publicly available on various topics such as career ambitions, politics and football.

Participant 2 was given a report detailing hometown, current location and schools attended. All of these were confirmed to be correct. As well as this 85% of the URLs believed to be pertaining to them were correct or accurate. From the search conducted numerous social media accounts were identified, news articles and blogs where they were mentioned for receiving awards, relatives were also identified and an academic article they published.

Participant 3 did not have information regarding their hometown, current location, places of work and education publicly available on their Facebook profile. Nevertheless 100% of the URLs suspected to be pertaining to them were correct. Findings included numerous social media accounts, a podcast they co-host, societies they are a part of and the position they held, a high school magazine they were mentioned in and finally a profile they have with a modelling and casting agency. This profile contains personal information such as their height, weight and ethnicity. This participant also responded in the questionnaire that they were surprised at the extent of information about them being publicly available on the internet.

In order to test the application's capability to contend with searches made in respect of more prominent public figures, information publicly available surrounding Louis-Rees Zammit was explored. This search uncovered a wide variety of articles he was mentioned in. These were not only pertaining to his rugby career but also various social media accounts. A 'Famous Birthdays' page and a number of his relatives were also identified. From the data retrieved we are able to summarise that Louis-Rees Zammit is a 20-year-old up-and-coming rugby player. He was born in Penarth, Wales and began playing rugby at the age of 7 for Llandaff Rugby and Football Club. He also attended The Cathedral School, Llandaff. Zammit started his youth career with the Cardiff Blues, but attended Hartpury University and College, Gloucester so he then attended the academy at Gloucester Rugby Club. Recently the Cathedral School published a video wishing the former pupil good luck for his debut in the Wales national team in a test match against France. Zammit has won numerous awards already in his professional rugby career. It is evidential from the information found, and the profile that was built in this case, that the OSINT tool can be utilised for a broad range of applications.

Overall, the aim of developing an OSINT application allowing Facebook and Google searches to be performed and streamline the OSINT process was achieved. The aim of exploring the extent of information in respect of individual(s) being publicly available was also met. Despite not having time for rigorous testing and more test cases, the results were certainly promising.

However, there were areas within this project that could be improved upon. Within the testing phase participants were only given a report containing information believed to be pertaining to them. It is possible that there was further information found but marked as not relevant and therefore not detailed in the report. The reasoning for this choice was that it would be impractical for participants to be given, in some cases over 100 URLs, and have them verify our decision, especially with the given timeframe. Another area within this project that could have been improved upon were the methods of retrieving information, namely through Selenium WebDriver. As mentioned prior in this report, OSINT tools often have a short life expectancy as they require lots of maintenance. With each browser update and every time a search engine or social media makes changes to their HTML code, the scripts need to be updated. Applications that use Selenium WebDriver are highly volatile to change.

6. Future Work

Although the OSINT tool has met the project aims, there is extra functionality that could be added if the project were to be continued.

6.1 Collect More Data from Facebook

Currently, the Facebook script only collects information regarding, hometown, current location and places of work and education. Facebook profiles can contain substantially more information, including:

- Relationships
- Gender
- Contact Information

- Interests
- Religious and Political Views
- Family Members
- Other Names
- Images

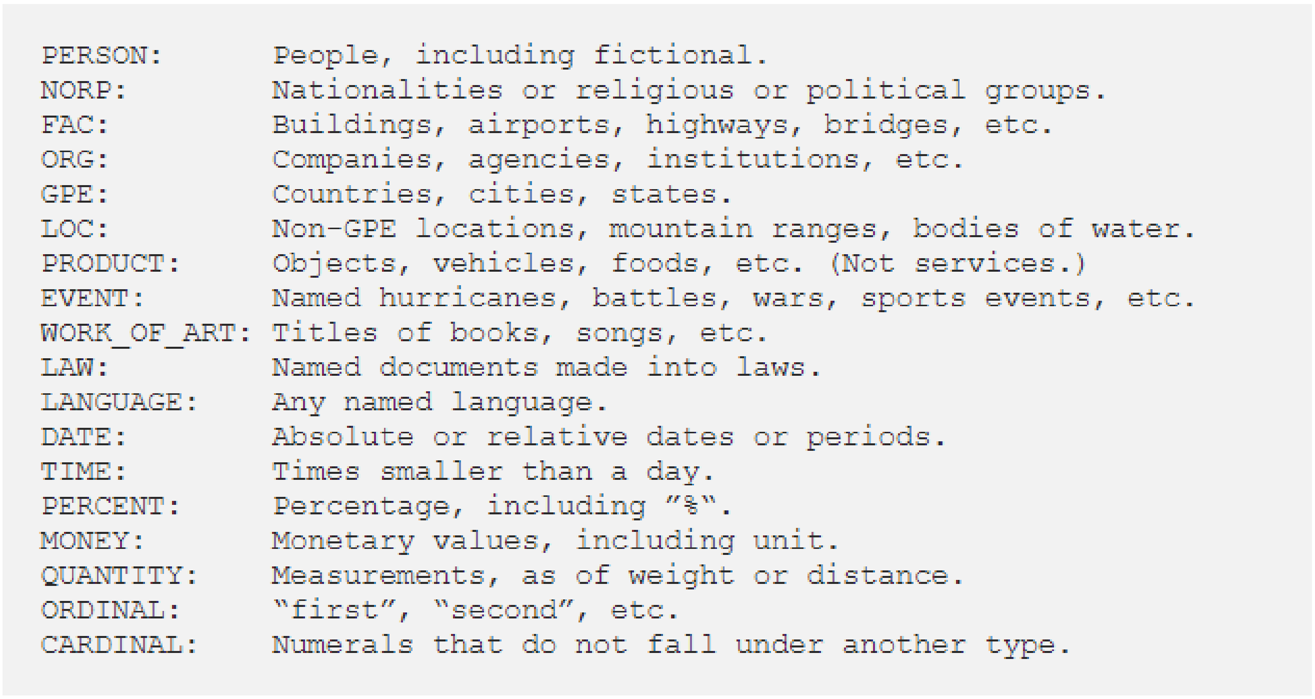
Collecting these and including functionality to interact with them within the OSINT tool could build a more accurate initial profile of the individual and help in creating better quality searches, receiving more relevant URLs and making decisions on whether or not information is relevant.

6.2 Incorporate More Search Engines

As mentioned earlier in the report, depending on what search engine you are using to conduct a search, you can receive quite different results. Search engines such as Bing, Yahoo, DuckDuckGo and Yandex could be included. With the OSINT tool's already existing ability to remove duplicate URLs, it would give the user a wider variety of information and possibly uncover material that would not have been found through Google alone. The use of 'Google Dorks' can also be explored. This is a search string that uses advanced search operators.

6.3 Natural Language Processing

One particular piece of functionality that could be explored is the use of Natural Language Processing (NLP) to completely automate the process by determining if information in a webpage is relevant. NLP is a subfield of artificial intelligence that can automatically understand, interpret and manipulate natural language. It is likely that this would be achieved with the use of Python as it has packages such as BeautifulSoup for parsing HTML and spaCy for working with human language data. With text extracted from a webpage using BeautifulSoup, spaCy can then be used for named entity recognition. A comprehensive list of entities spaCy can recognise is shown in Figure 9.



```
PERSON:      People, including fictional.
NORP:        Nationalities or religious or political groups.
FAC:         Buildings, airports, highways, bridges, etc.
ORG:         Companies, agencies, institutions, etc.
GPE:         Countries, cities, states.
LOC:         Non-GPE locations, mountain ranges, bodies of water.
PRODUCT:     Objects, vehicles, foods, etc. (Not services.)
EVENT:       Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART: Titles of books, songs, etc.
LAW:         Named documents made into laws.
LANGUAGE:    Any named language.
DATE:        Absolute or relative dates or periods.
TIME:        Times smaller than a day.
PERCENT:     Percentage, including ”%“.
MONEY:       Monetary values, including unit.
QUANTITY:    Measurements, as of weight or distance.
ORDINAL:     “first”, “second”, etc.
CARDINAL:    Numerals that do not fall under another type.
```

Figure 9: spaCy Entity Recognition Capabilities

Recognising and labelling entities would allow for comparison to data known to be correct. The idea of giving webpages relevance scores could also be explored. If a webpage is given a confident relevance score, the data identified within it can be added and used to compare with other webpages.

7. Conclusions

The project's primary aim was to create an OSINT application which would allow Facebook and Google searches to be performed. It was intended to streamline the OSINT process by collating potentially useful information into one application whereby users could mark information as useful and then export these findings. I believe that this aim has been satisfied. All of the functionality that was set out to be achieved has been implemented and helps in solving the issues of needing to create documents, record findings, and revisiting webpages multiple times. Despite choosing to work with Java and JavaScript a number of issues did arise and it is possible that these could have been identified at an earlier stage with more research being conducted. However, despite this these challenges were overcome and the application still works as intended.

The secondary aim of this project, which was to explore the extent of information in respect of an individual(s) being publicly available, has also been achieved. This was done by testing the application using volunteer details. It was successful in that it did indeed find a plethora of data and confirmed that we leave trails about various aspects of our personal life on the internet. Even when we think information is protected by privacy settings on social media, such as Facebook, there are still ways of obtaining this material. I believe that users of social media and the internet should be educated in respect of this. As a consequence, they could then make informed decisions, which would allow them to determine whether or not they would like to reduce the amount of personal information pertaining to them being publicly available. Only by doing so can we reduce the amount of personal information being available and then used for malicious purposes, such as a socially engineered attack.

8. Reflection on Learning

This entire project has been an extremely valuable learning experience. It has allowed me to develop both my technical and soft skills, both of which are essential for my personal development and which will undoubtedly help me in both my future learning and career.

Having never built an application or graphical user interface (GUI) before I was apprehensive about the process. I conducted extensive research in respect of this in order to establish what option would best suit my objective. Despite experiencing numerous difficulties in finding the appropriate solution I persevered and eventually arrived at a solution in the form of Java Swing. My programming has allowed me to build upon my previous Java and JavaScript skills, obtained from my second year at University and my year on placement. I am more aware of the libraries available to use in Java and will utilise them in future.

I have also been able to build upon my time management skills. With a short timeframe for this project, it was essential to plan my time effectively. I decided to tackle tasks that I was more confident of completing first in order to prevent procrastination. My intention was to build momentum which I believed would be beneficial for the more difficult tasks within the project.

On the whole I adhered to the work plan proposed in my initial report, with very few tasks taking longer than expected, however upon reflection I now realise that I could also have obtained a great deal of conviction and confidence by getting to grips with the most demanding pieces of work first. Undoubtedly these were at the back of my mind whilst completing the areas that I was assured of and it may have been beneficial to have completed these first in order to have given me peace of mind and further positivity.

On reflection one of the tasks that took considerably longer than expected was the participants evaluation of findings. This can be ascribed to the participants also attending Cardiff University and being committed to their own assignments and working to deadlines of their own. In retrospect I would have sought interaction from these volunteers at a far earlier stage in order to meet the deadlines issued or alternatively sought volunteers with less
time constrained commitments.

The project has also taught me the importance of the vigilance needed in respect of the volunteers participation. Obtaining approval for my project from the School of Computer Science and Information Ethics Committee took more time than expected. I underestimated the need to clarify important aspects, such as how participant data was going to be stored and the length of time it would be retained. This was an invaluable insight into what is needed to have volunteers participate in a project ethically and is something which I have learned from and can take forward in future work that I carry out.

The project has also developed my knowledge of the software development cycle. Initially I was very eager to begin implementing my application however with the guidance of my supervisor Dr Jianhua Shao, I quickly realised that I needed to take the time to plan, analyse and design my application before implementing the creation phase. Undoubtedly this will help me in the future, I now realise that planning and preparation are the cornerstone of all work that is carried out.

Overall, as the project's principal investigator, this role required a lot of multitasking and organisation. Whilst planning and designing the application I needed to apply for approval from the ethics committee. With regular meetings with my supervisor, I was able to prioritise work effectively.

9. References

[1] Tom Bendix, 2021, Facebook UK Statistics 2021, Available at: https://www.socialfilms.co.uk/blog/facebook-uk-statistics#:~:text=with%2073m%2B%20views-,How%20many%20people%20use%20Facebook%20in%20the%20UK%3F,seeing%20considerable%20growth%20over%20time [Accessed January 2021]

[2] Stephen Phillips, 2019, Social Media Statistics UK, Available at: https://hostreviews.co.uk/social-media-statistics-uk/#:~:text=On%20average%2C%20each%20individual%20social,from%2066%25%20to%2067%25 [Accessed January 2021]

[3] Laura Butler, 2019, Cyber Criminals raking in over $3bn a year from social media crime, Available at: https://www.surrey.ac.uk/news/cybercriminals-raking-over-3bn-year-social-media-crime [Accessed January 2021]

[4] John Breeden II and Josh Fruhlinger, 8 top open source intelligence tools, Available at: https://www.csoonline.com/article/3445357/what-is-osint-top-open-source-intelligence-tools.html [Accessed January 2021]

[5] Brian Everstine, Carlisle: Air Force Intel uses ISIS 'moron's' social media posts to target airstrikes, Available at: https://www.airforcetimes.com/news/your-air-force/2015/06/04/carlisle-air-force-intel-uses-isis-moron-s-social-media-posts-to-target-airstrikes/ [Accessed May 2021]

[6] Alex Chris, Top 10 Search Engines In The World (2021 Update), Available at: https://www.reliablesoft.net/top-10-search-engines-in-the-world/ [Accessed April 2021]

[7] Maryam Mohsin, 10 Google Search Statistics You Need To Know In 2021 (Infographic), Available at: https://www.oberlo.co.uk/blog/google-search-statistics#:~:text=Google%20processes%20over%203.5%20billion,searches%20per%20year%20have%20progressed. [Accessed April 2021]