UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

PROJECT REPORT

Incorporating Mobile Forensic Analysis and Reporting into Nugget

Daniel Mbugua

REG NO: P53/11790/2018

SUPERVISOR: Dr. Elisha Abade

A research project report submitted to the School of Computing and Informatics in partial fulfilment of the requirements for the award of the Degree of Master of Science in Distributed Computing Technology at the University of Nairobi, Nairobi, Kenya.

Submitted on 22nd July 2020

Declaration

STUDENT

This work, as presented in this report, is my original work and has not been presented for any award in any other university.

Signed ..... Date .....

Daniel Mbugua Itangi P53/11790/2018

SUPERVISOR

This research project is submitted as a partial fulfillment for the award of Master of Science Degree in Distributed Computing Technology with my approval as the university supervisor.

Signed ..... Date .....

Dr. Elisha Abade
School of Computing and Informatics

Dedication

I dedicate this project to God Almighty, the giver of wisdom, knowledge and understanding.

My humble efforts, I dedicate to my loving Dad & Mum, my siblings and friends.

Whose love, support, prayers and encouragement has led me this far.

Thank You.

Acknowledgement

I thank the Almighty God for the good health and for guiding me through to the completion of this project.

I am particularly thankful to my supervisor, Dr. Elisha Abade for being available and resourceful and for providing guidance during the undertaking of this project.

Special thanks to Chomba Ng'ang'a who ensured I got this far.

Finally, I would like to acknowledge my classmates and colleagues for their endless support and encouragement during the undertaking of the project.

List of Abbreviations

DFXML Digital Forensics XML

ANTLR ANother Tool for Language Recognition

DSL Domain Specific Language

UI User Interface

FTK Forensic Toolkit

XML Extensible Markup Language.

Abstract

The application of digital mobile forensic investigation has become an integral process both in private and public institutions (Sarwar Mir, Shoaib and Shahzad Sarfraz, 2016). This has led to the development of various forensic tools in order to bridge the gap in the business processes. The development however has come at a cost; most of these tools are heterogenous and hence pose the challenge of interoperability. (Stelly and Roussev) (2018) developed Nugget to address this in the spectrum of digital forensic. The development was limited to memory and network analysis. The exception of mobile forensic introduced a new area of research, which (Chomba and Abade 2018) explored in their research titled *Incorporating mobile forensic into Nugget*.

The research however introduced another gap, the application of mobile forensic into Nugget did not cover the last two phases of mobile forensics which are analysis and reporting. To address this, we incorporated mobile forensic analysis and reporting into Nugget. The output of this research was a software tool that can be used by forensic investigators when performing mobile forensics.

Keywords: *Mobile forensics, Nugget*

Table of Contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

1.1 Background

As the usage of digital media and devices become widespread, digital forensics has become an integral process not only in governments but also in other public and private organizations (Sarwar Mir, Shoaib and Shahzad Sarfraz, 2016). The application of digital forensics spans from carrying out organization audits to crime investigations for the purpose of providing admissible evidence premised on the cases under study. The people tasked with the responsibility of digital investigation are known as digital forensic experts.

To carry out any digital investigation, digital forensic experts rely on forensic tools to extract, analyze and represent their findings for decision making. With the increased need for forensic investigation, a broad variety of digital forensic tools have been developed. This may be perceived as an advantage but there are some drawbacks to this: the dilemma of what tool to use and why, what type of input does the tool take and what type of output the tool provides. These are just but a number of questions that forensic experts need to get answers to before they embark to selecting the best tool suited for the job.

To resolve the diversification of the digital forensic tools, Stelly and Roussev (2018) developed Nugget. Nugget is a digital forensics Domain Specific Language (DSL) targeted at digital forensic investigators. It provides a useful and formal description of digital forensic analysis (Roussev, 2015) that abstracts out its implementations. Nugget aims to address how investigators specify dataflow of a forensic inquiry from data source to final result as well as allowing fully automatic and optimized execution of forensic computation and to provide a complete, format and auditable log of inquiry (Roussev, 2015).

However, developing Nugget, did not resolve all digital forensic issues; specifically, in mobile forensics. As noted, Nugget supports hard disk forensics, network forensics and memory forensics only and therefore there was the need to integrate mobile forensics into Nugget (Chomba and Abade 2018). Chomba and Abade (2018) integrated mobile forensics into Nugget to actualize this. This research work emphasized on the first two phases of digital

forensics; identification and extraction. Given a digital forensic target, an output in form of DFXML would be produced to for analysis and reporting.

```
<mobile:location>
 <mobile:long>−77.11546951</mobile:long>
 <mobile:lat>38.87767624</mobile:lat>
 <mobile:source></mobile:source>
 <mobile:confidence>70</mobile:confidence>
 <mobile:timestamp>2012−06−13T22:01:21+03:00</mobile:timestamp>
 <mobile:cell_mcc>310</mobile:cell_mcc>
 <mobile:cell_mnc>410</mobile:cell_mnc>
 <mobile:cell_lac>7985</mobile:cell_lac>
 <mobile:cell_ci>160043533</mobile:cell_ci>
</mobile:location>
```

Listing 1.1: A sample DFXML output showing GPS coordinates extracted from a mobile phone

This presented another challenge to digital forensic investigators; getting meaningful results from the tool in order to make informed and timely decisions. As can be noted from the listing above, it would be hard for an investigator to make decisions because the information is hard to interpret. Decisions the investigator makes are heavily influenced by the last two phases of digital forensics which are analysis and presentation and therefore the need to bridge the gap.

1.1 Problem definition

Digital forensic investigation involves seizure of media targets, extraction of digital evidence, analysis of the data extracted and finally reporting on the forensic findings. The four phases are equally important as decisions are made based to the output of each phase. To address the need for mobile forensic standardization, Nugget was developed. However, the development and integration of Nugget into mobile forensics did not address the need for a complete model that would allow for a semantically rich analysis and presentation of the forensic findings which would help forensic experts in making informed and timely decision on the case under study.

As stated by Chomba and Abade, (2019) in their recommendations for future work, the current output representation does not exploit the depth of the information available and is hard to consume to the end user. Chomba and Abade, (2019) also notes that the current model does not allow for independent replication of forensic computations to validate the results of the computation which would be necessary at the final stage of forensic analysis which is reporting.

1.2 Research Objectives

The objectives of this study were:

1. To evaluate existing models used in digital forensic analysis and reporting of digital forensic information.
2. To design an architectural model that will inform Nugget integration to allow for rich, meaningful analytics of the forensic findings.
3. To develop a mobile forensic middleware comprising of a data service layer, analytics engine and a presentation layer from the derived architectural model.
4. To develop an integration layer between Nugget and available digital forensic tools for forensic data exchange.
5. To validate the architectural model using publicly available digital forensic targets.

1.3 Justification

The current implementation model of the integration of mobile forensics into Nugget identifies a gap in the last two phases of digital mobile forensics; analysis and presentation which are critical in digital forensic investigation. The implementation produces results that are hard to consume and interpret for the investigator and hence an opportunity for further research.

The inclusion of a forensic middleware between Nugget and forensic investigators which will introduce a data service layer, an analytics engine and a presentation layer will provide investigators with detailed information and analysis in order to make informed decisions on the case under investigation.

Lastly, the development of the integration layer between Nugget and the proposed middleware will ensure standardization of data exchanged and also an opportunity for further development without interfering with the Nugget implementation internals.

1.4 Scope of the study

This research limited its scope to the incorporation of a middleware which is made up of a data service layer, an analytics engine and a presentation layer between Nugget and other mobile forensic tools. Whereas mobile forensics is a wide area with fields such as network forensic, the current research was limited to mobile data forensics which include but not limited analysis of SMS and call logs.

Chapter 2: LITERATURE REVIEW

2.1 Introduction

Forensics is the systematic application of scientific techniques to gather and analyze findings for legal purposes (Roussev, Bertino and Sandhu, 2016). Digital forensics is a branch of forensic science that entails the recovery and investigation of materials from digital media confined on legal grounds. The recovery and investigation of materials go through identification, extraction, analysis and reporting of the findings. Mobile forensics is a branch of digital forensics which relates to collection of digital evidence from mobile devices for legal evidence purposes.

2.2 Why mobile forensics.

The growth in technology along with initiating integrative forces that offer improvement of quality of human life, have concurrently created prerequisites for individuals to exploit certain innovations for performing criminal activities (Spalevic, Bjelajac and Caric, 2012). Some of the innovations that have become candidates of exploitation are mobile devices, which have become part of our lives. This is because the mobile devices have been enhanced with the ability to store, view, transfer and print electronic documents among other capabilities. Some of the important information the mobile devices hold which is crucial in digital forensic investigations are SMS, call logs, location data and multimedia information (images, videos, email). This information would help an investigator reconstruct events in order to identify where the perpetrator of the crime was, communicating with who and what type of communication was made.

Toroitich (2020) points out that based on a report by National Kenya Computer Incident Response Team Coordination Centre, 26.6 million cyber threats occurred in Kenya between April and June 2019. Gravrock (2020) also notes that according to RSA more than 60% of fraud online is accomplished through mobile platforms. These cases point out the need for enhanced research and development in mobile forensics.

2.3 Phases of digital forensics

Digital forensics is the process of examing digital forensics artefacts to detect and collect evidence related to the case under investigation. Kent et. al (2006) categorizes digital forensic into four main phases.

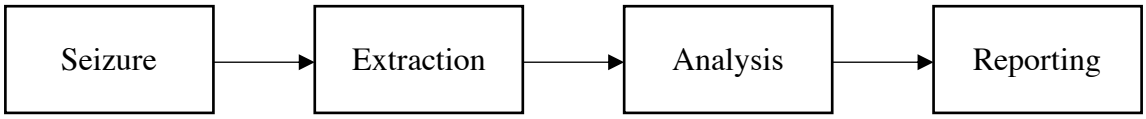

Figure 2.1: Digital forensic Phases.

2.3.1 Seizure

Prior to the actual examination, the mobile device is seized. This is a critical phase in digital forensics as it feeds the other phase in the process. The investigator preserves the device in its original state so that the evidence in kept intact. One of the many tasks performed in this phase is to prevent wireless connections which would either bring new data or overwrite the evidence. To facilitate the success of this phase, different approaches are taken which can be transporting the mobile device in shielded bags and placing the device in airplane mode.

2.3.2 Extraction

The goal of this phase is to retrieve data from the mobile device. Once the media has been acquired, a duplicate copy is created which is used in the rest of the process. This exercise is important as there's a need to preserve the original exhibit in case it needs to be referenced in the future. The acquired image is verified using hashing algorithms to ascertain that no alteration has happened.

Data extraction can be categorized into: a) Logical extraction and b) Physical extraction. Logical extraction involves using software to access the data from the device. The limitation to this method is that the software does not have access to all data (A. Khawla et al., 2012). This is due to the security measures applied in the current mobile devices. To overcome this limitation, physical extraction is applied. This method uses the hardware and software to extract information directly from the phone. Investigators are able to access raw memory and storage information.

2.3.3 Analysis

Investigation is done to identify evidence that either contradicts or support a given hypothesis. This is done after the acquisition phase. The recovery of evidence during analysis from digital

media is done using various tools and technologies. The analysis can even go as far as recovering deleted materials from unallocated disk space to operating system cache files depending on the objective of the analysis. For the recovery of forensic evidence, forensic analysts use tools such as EnCase, ILOOKIX and FTK among other tools. Data extracted and analyzed depends on the goal of the investigation. The data ranges from chat logs, emails, images to internet history or documents among others.

2.3.4 Reporting

This majorly marks the last phase of forensic analysis. Reporting involves presenting the findings of the analyzed data in a form that can be consumed by technical and non-technical persons. Various ways are used in this phase, with the goal being to support or disapprove a given hypothesis under study. This may include electronically presenting the results using forensic tools and having an accompanying written or e-report. The reporting is totally informed by the evidence collected and not by suspicions or unsupported materials.

2.4 Data acquisition from mobile devices.

This is the process of extracting important artefacts from digital devices to be used in a forensic process as evidence to prove or disprove a hypothesis. Data acquisition can be classified into four, a) manual acquisition, b) logical acquisition, c) physical acquisition and c) chip-off acquisition (Abdulla, Jones and Martin, 2012).

- Manual acquisition

  This is one of the easiest ways of retrieving data from a mobile phone. This process requires the investigator to manually browse through the device using its interface or keyboard to register the information displayed on the screen. This approach works for all devices and does not required any cable connection to extract the information. The drawback of this approach is that it does not maintain the integrity of the information and does not retrieve all information eg hidden files (Abdulla, Jones and Martin, 2012). This method also requires that the investigator is able to access the device.

- Logical extraction

  This process involves retrieving information that reside in the logical partition. A connection is created between the mobile phone and the forensic investigator PC using either, Bluetooth, infrared or a cable connection. The investigator can then use a set of AT commands to extract specific potential evidence from the mobile device (Willassen, 2005, p. 191-204). The drawback of this method is the inability to retrieve deleted files (Abdulla, Jones and Martin, 2012).

- Physical acquisition

  This method of acquisition entails copying the entire physical memory location of the device memory chip.

- Chip-off

  The purpose of this method is to get an image of the internal non-volatile memory. The process is done by de-soldering the non-volatile memory from the PCB (Mikhaylov & Skulkin, 2016). Then, cleaning chip pins of solder that has been left behind from the de-soldering stage. Finally, the content of the memory is read using a chip reader. This method is well suited for damaged devices or devices with a locked bootloader.

Further from the above classifications, (Brothers, 2011) proposed a way to group techniques applied to retrieve information from mobile devices as shown in the figure below.

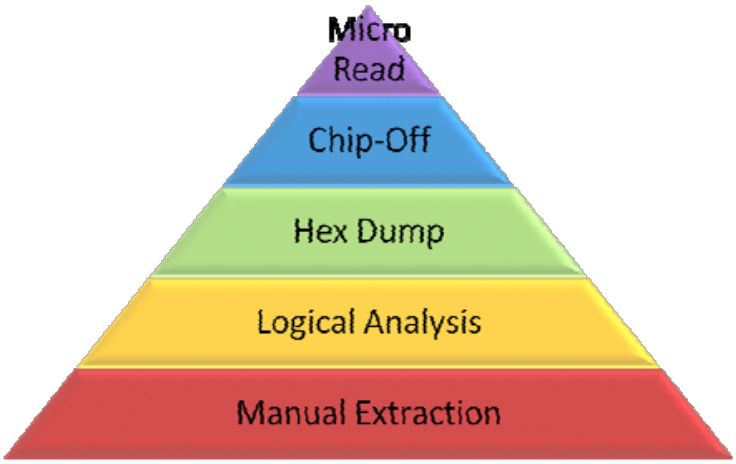

Figure 2.2: Tool hierarchy. Source: Brothers, 2011

The techniques used are highly specialized and technical moving up the pyramid, they take longer times to analyze, they require more sophisticated and expensive tools and are more invasive to the device (Murphy, 2011).

2.5 Digital Forensics Ontology

The ontology refers to a shared understanding of some domain of interest which is used as a unifying framework in solving problems (Uschold and Gruninger, 1996). Digital forensic analysis relies heavily on the review an analysis of unstructured information that is contained in the data extracted from the digital devices. In order to analyze the information, it's important to have a digital forensic ontology (Henseler and Hyde. 2019). The Cyber-investigation

Analysis Standard Expression (CASE) provides such an ontology. CASE is an open standard used to describe different types of digital evidence from various domains.

Once a semantic network has been identified, based on the ontology, it can be used in identifying possible crime scenarios (Henseler and Hyde. 2019) and testing hypothesis.

2.6 Forensic analysis models

Forensic analysis entails processing of data from digital forensic targets in order to retrieve information that is used in decision making to ascertain if there is proof to a hypothesis. This involves event reconstruction which allow investigators to understand the timeline of a crime (Chabot et al., 2014). To make an informed decision, investigators have to consider all factors and parties from the classified information they retrieve from a forensic tool. To achieve this, the investigator can choose to use the models below in the analysis (Sammons, 2015).

2.6.1 Timeline Analysis

Timelines are an essential part of forensic analysis. The visualization of a timeline combined with frequency analysis can be used to categorize the type of suspect and crime. It can be used to determine the pattern of device usage. The primary source of timeline information is the file system metadata including the modified (file metadata), accessed, changed and created file contents (Hoog, 2011).

Timeline analysis provide the investigators with a succinct view of the occurred events in a chronological order. While performing a timeline analysis, a forensic investigator seeks to understand more about the events surrounding the case under study. Some of the events would be (a) linking SMS to specific locations and time (b) linking calls with SMSs sent at a particular time.

To help analyze forensic data, Yoan et al (2014) Proposed a knowledge model which has four main composed relations; subject, object, event and footprint. In the model Yoan et al (2014) describes a crime scene as a space where a set of events $E = \{e_1, e_2 \ldots e_i\}$ takes place.

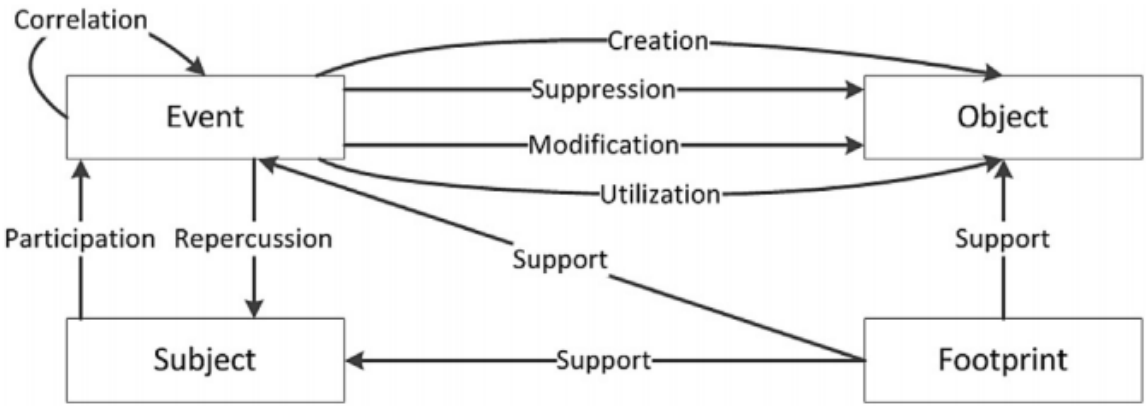

Figure 2.2 Knowledge model. Chabot et al. Digital Investigation

An event is a single action occurring at a given time and lasting a certain duration. A duration in a case of mobile forensic would be, making a phone call or a conversation via SMS.
An event involves subjects; where subjects in this case involves human actors and processes. A subject corresponds to an entity involved in one or more events. During the life cycle of an event, it can also interact with objects. An object in this case can be a webpage. A footprint is the sign of a past activity and a piece of information allowing to reconstruct past events.
The above model gives an incite of how timeline analysis can be modeled in mobile forensics to allow for meaningful information to investigators.

2.6.2 Link Analysis

Link analysis is a data analysis technique used to analyze the relation or connection between the network nodes or criminal cases. The technology used in link analysis enables examiners to build an immediate visual picture of communications, and helps them understand the relationship between those involved in a criminal case, and piece the puzzle together. Detection of critical communications via texts, calls, emails and social media apps puts the power in the hands of the field examiner at a vital stage of analysis; something that would have previously been unheard of with more traditional forms of forensic investigations (Moshe, 2015).
Examiners are able to visualize case data from multiple devices, analyze mutual device users on a single map and share findings. In most cases, a digital forensic case will involve more than two entities. In such a case, the investigators main goal is to identify if there is any link between the two entities.
The main motives of link analysis are (a) to identify patterns of interest between linked objects, (b) to identify for abnormal conditions by detecting profaned known patterns and to look for new patterns (Acquire Forensics – Learn In-Depth About Digital Forensics, 2020).

2.7 Nugget

Nugget is a domain specific language (DSL) which aims to enable the practical formal specification of digital forensic computations. A DSL is a computer programming language of limited expressiveness focused on a particular domain (Fowler, 2010). In the context of digital forensics, nugget aims to address the following: a) provide investigators with the means to easily and completely specify the dataflow of a forensic inquiry from data source to final results; b) allow the fully automatic execution of the forensic computation; b) provider a complete, formal and auditable log of the inquiry (Stelly and Roussev 2018).

2.7.1 Nugget DSL

Fowler, (2010) classifiers domain specific languages into two, internal and external. The difference between the two is that, internal DSL are an extension of their host languages; They add constructs to an existing general-purpose language. The advantage of internal DSL is that, the end users are able to invoke the full capacity of the host language. External DSL are completely independent in the context of parsing, compilation, lexical analysis and code generation. This is important as developers are free to create and extend their own syntax and semantics. Nugget started out as an internal DSL but switched to an external DSL after reaching its limit (Stelly and Roussev, 2018).

2.7.2 Nugget Architecture.

Nugget provides a single means of conducting forensic computations using forensic tools available and returning the results as needed (Stelly and Roussev 2018).

The representation of an operation is mapped out by the language runtime. An example of the operation is the extraction of list of processes to an actual command to be invoked to the target. To schedule computations and distributing the available resources to running processes, Nugget has a resource manager. It also ensures successful execution of processes. Stelly and Roussev 2018).

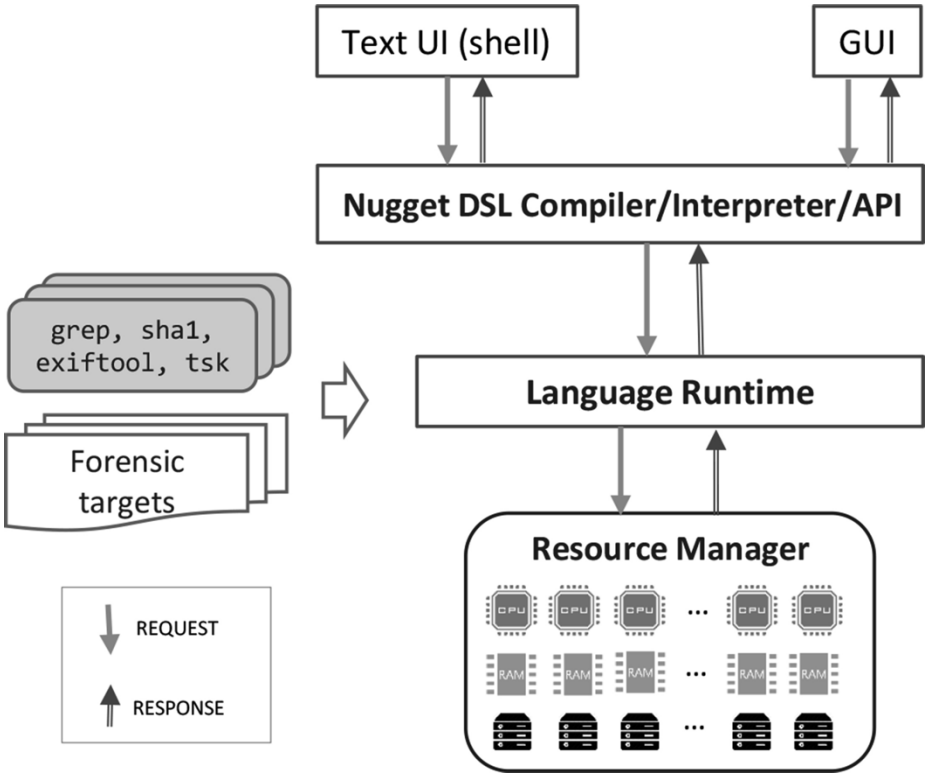

Figure 2.3: Nugget runtime architecture. Source: Stelly and Roussev, 2018

Nugget architecture as shown in Figure 2.3 disencumbers the concerns of) specifying the computation, b) mapping computations to available forensic tools and c) scheduling computations to available hardware resources Stelly and Roussev (2018).

2.7.3 Incorporation of mobile forensics into Nugget

As of 2018 Nugget only had support for hard disk forensics, network forensics and memory forensic. These were the only forensic tools containers integrated into Nugget (Stelly and Roussev 2018). Chomba and Abade, 2019 noted that there was need to integrate mobile forensics into Nugget for Android and iPhone as this was currently not implemented. The goal for this integration was to bring standardization in mobile forensics.

The communication between Nugget and forensic containers is via remote procedure calls (RPC). The forensic containers run their set of forensic tools on the given data type upon receipt of an RPC connection (Stelly and Roussev 2018). Utilizing the RPC communication has the advantage of extensibility as new instructions can be incorporated into containers be specifying functions that conform to a single convention. Below is a concept overview of the integration

of mobile forensics into Nugget. The communication between Nugget and the forensic container tool container was via RPC.

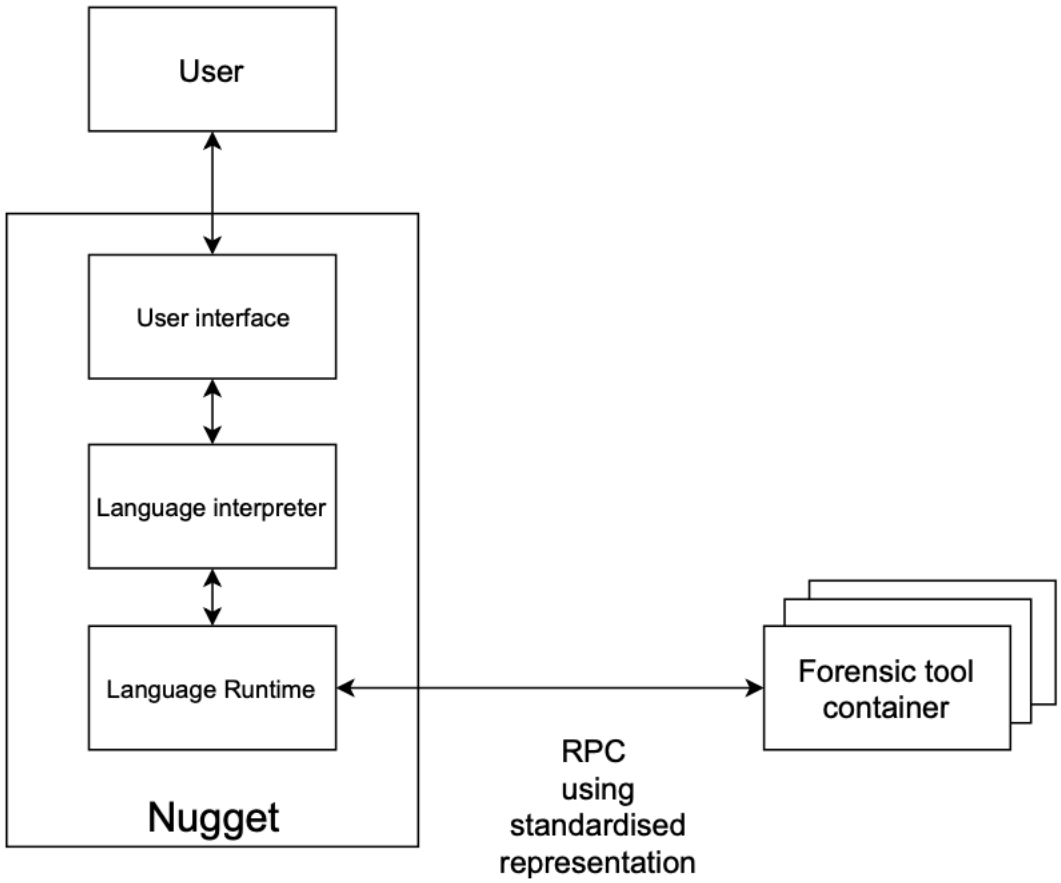

Figure 2.4: Concept Overview. Source: Chomba and Abade, 2018

The RPC interface provides its output in DFXML format, this was the output from the forensic tool. Below is a sample of the out.

```
<mobile:sms_mms>
  <mobile:kind>sms</mobile:kind>
  <mobile:sender>+15713083236</mobile:sender>
  <mobile:body>What are you up to this weekend? </mobile:body>
  <mobile:date_received>2012−06−13T00:25:04+03:00</mobile:date_received>
  <mobile:read>1</mobile:read>
  <mobile:country_code>us</mobile:country_code>
  <mobile:type>inbox</mobile:type>
</mobile:sms_mms>
```

Listing 2.1SMS and MMS DFXML extraction as DFXML

As can be deduced from the DFXML output and also from the researcher's recommendations there is need to explore and model how the output can be analyzed and presented to the investigator in order to fully utilize the incorporation of mobile forensic into Nugget.

2.8 Digital forensic corpora

A digital forensic corpus is a collection of standardized forensic material which are used for digital forensic operations. Having a reference set of representative corpora enhances the scientific evaluation of forensic methods beyond the obvious benefits of providing ready test data and enabling direct comparison of different approaches (S. Garfinkel et al., 2009). A specially crafted digital forensic corpus containing realistic data has a number of benefits, including testing correctness of a forensic tool's results, comparing two or more forensic tools, training and educating digital forensic experts (S. Garfinkel et al., 2009). Digital forensic corpora enable researchers to test techniques they have developed and also valid ate techniques other researchers have developed (S. Garfinkel et al., 2009).

2.8.1 Forensic reproducibility

There have been few attempts to enable digital forensic researchers to reproduce results. S. Garfinkel this to the manner in which digital forensics have evolved and the nature of forensic data. With the standardization of the forensic material S. Garfinkel states that this would help in forensic research operations, science research and training in digital forensics.

2.8.2 Corpora modalities

The current digital forensic corpora have the following different kinds of corpora:

1. **Disk images**; These are the fundamental kind of forensic corpora because they have a long-established use in forensics (S. Garfinkel et al., 2009).
2. **Memory images**; These include images of multiple operating systems and re needed for the development of both forensic tools and forensic training.
3. **Network packets;** Packet corpora can consist of traffic from one or more individual systems or networks (S. Garfinkel et al., 2009).
4. **Files;** can be productively collected and distributed as corpora. There has been considerable work on file and file fragment identification which would have benefited from standardized corpora of files. Work on metadata and text extraction would also benefit from such corpora. Although files can certainly be extracted from disk images, distributing files as stand-alone corpora significantly simplifies the effort for the intended users (S. Garfinkel et al., 2009).

2.9 Proposed concept overview

The diagrammatic representation below is the conceptual framework that shows how this research is geared towards modeling and implementing an analysis and presentation layer between Nugget and other forensic tools and therefore completing full cycle of digital forensic.

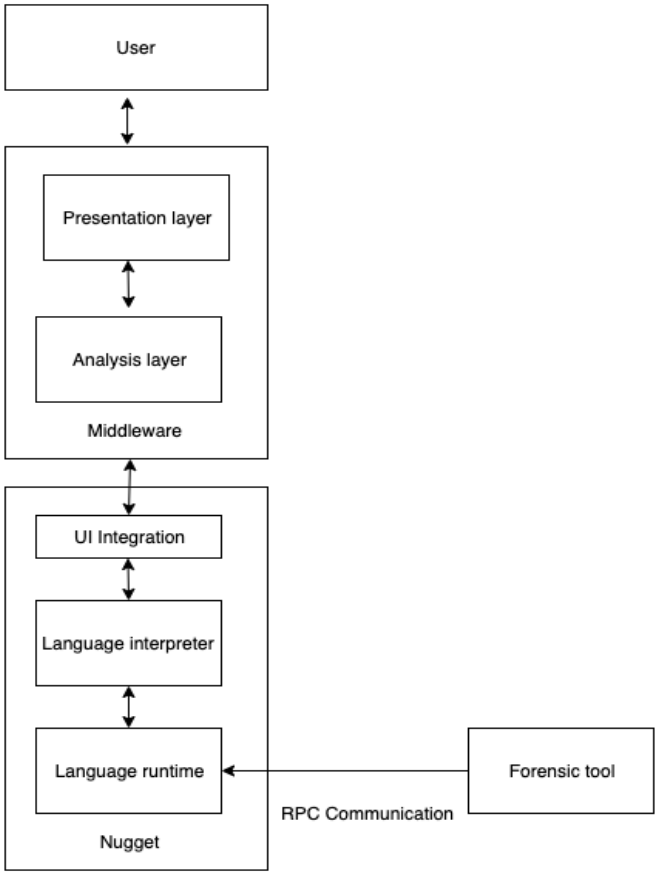

Figure 2.5: Proposed concept overview.

Chapter 3: RESEARCH METHODOLOGY

3.1 Research design

Research methodology outlines how the research was carried out to achieve the stated research objectives. The research was divided into four major phases: (a) Evaluate existing methods of analysis and reporting of digital forensic information. (b) Designing an architectural model that informed Nugget integration to allow for rich analytics of forensic findings (c) Design and implement an integration layer between Nugget and target forensic tools for data exchange (c) To implement a mobile forensic middleware comprising of a data service layer, analytics engine and a presentation layer (d) Test the implementation using publicly available digital forensic targets.

3.1.1 Evaluate existing methods of analysis and reporting of digital forensic information.

This phase of the research was be an exploratory study, focusing on existing methods of analysis and reporting of digital forensic information.

1. Identify a list of methods of analysis of digital forensic information.
2. Identify a list of methods of reporting the analyzed digital forensic information.
3. Identity the methods of analysis and reporting to integrate in the tool so as to achieve the third objective.

The output of this phase informed the research on what method to incorporate in the analysis and reporting layer in Nugget, satisfying the fourth research objective.

3.1.2 Design an architectural model that will inform Nugget integration to allow for rich analytics of the forensic findings.

In this phase the researcher designed an architectural model of a middleware that comprised a data service layer, analytics engine and a presentation layer. This design informed the roadmap to achieving the fourth objective. The methodology chosen for the above objective was evolutionary prototyping. Design iterations were made in order to come up with a suitable architectural model to be used in the integration.

3.1.3 Design and implement an integration layer between Nugget and target forensic tools for data exchange.

In this phase, the researcher designed, implemented and tested an integration layer between Nugget and target forensic tools for data exchange. For this research to be a success, digital forensic information needed to be exchanged from target forensic tools and Nugget. This phase was conducted in the following steps:

1. Design and implement and integration layer for data exchange between Nugget and target forensic tools
2. Identify target forensic tools to test the integration.

The methodology chosen for the above steps was evolutionary prototyping. The output of this phase helped in achieving the fourth objective.

3.1.4 Develop a mobile forensic middleware comprising of a data service layer, analytics engine and a presentation layer from the derived architectural model.

In this phase, the researcher designed, implemented and tested a model which comprised of the analysis and presentational layer for Nugget. To implement this phase, the researcher used applied experimentation (Edgar & Manz, 2017) techniques. This phase was conducted in the following steps:

1. Identified materials in digital forensic corpora to be used in testing the implementation of the analysis and presentation layer. Digital Corpora (Digital Corpora, 2017) as well as digital forensic challenges e.g. Digital Forensics Research Workshop (DFRWS) challenges were used to source samples for experimentation.
2. Designed and implemented an analysis and presentation layer for Nugget.
3. Identified a mobile forensic tool to integrate with the analysis and presentation layer.
4. Tested the integration with the digital forensic materials identified in step one.

The methodology chosen for the above steps was evolutionary prototyping. The output of this phase was the implementation of an analysis and presentation layer, satisfying the third objective.

3.1.5 Conceptual architecture

The conceptual architecture of the proposed implementation is illustrated below:

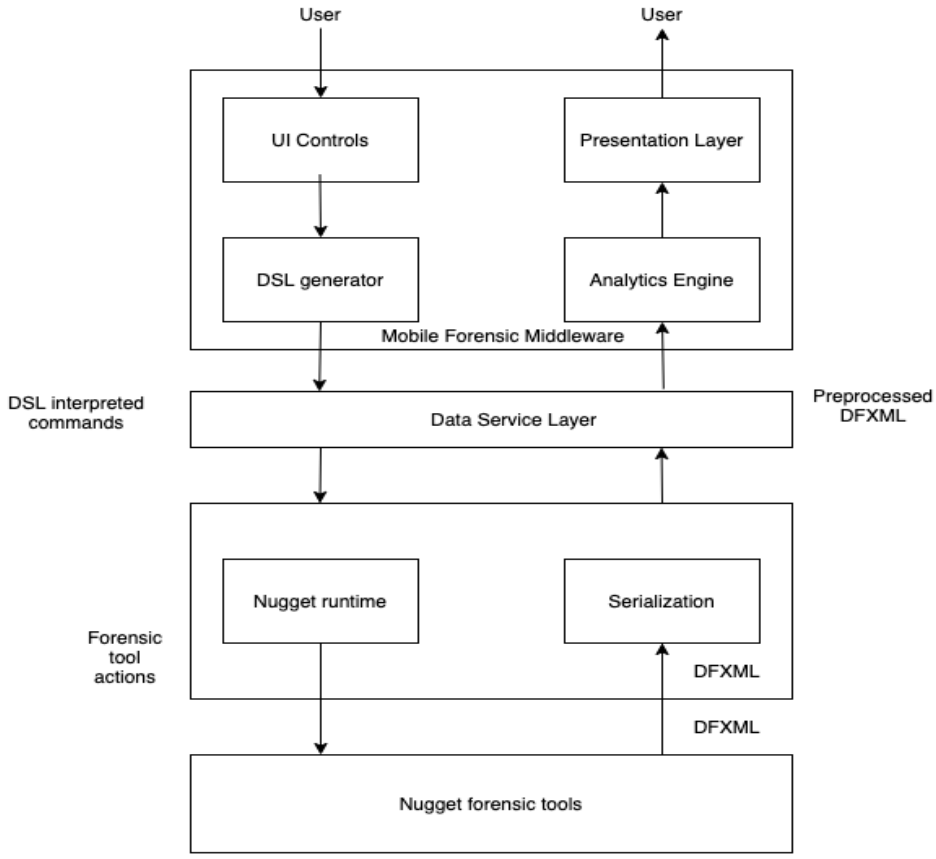

Figure 3.1: Proposed concept architecture.

The architecture consists of four main layers:

- Mobile forensic middleware

  This layer contains for main components which are a) the UI controls, b) DSL generator, b) the analytic engine and the presentation layer. The UI controls offer the channel for interaction between the users and Nugget forensic tools. The DSL generator component interprets the user actions into commands Nugget understands. The analysis layer contains the analytic engine which processes the results from Nugget to information the presentation layer can pick up in order it can be feedback to the user.

- Data service layer

  This is an abstraction layer for data access independent of what type of information is contained in the DFXML.

- Nugget

  Nugget consume DSL commands from the middleware in order to channel them to the forensic tools. These are commands Nugget understands. This layer has three main operators: a) *filters*, *transformers* and *serializers*. After receiving the DSL commands, Nugget will communicate with the forensic tools via RPC to perform the given action. The output from the forensic tools is feedback to Nugget in form of DFXML format for preprocessing.
- Nugget forensic tools.

  This layer provides forensic tools available for integration. This layer communicates to Nugget via RPC and the result is sent back to Nugget in DFXML format.

3.2 Source of Data

The dataset for this research was made up of mobile devices images from Digital Corpora (Digital Corpora, 2017). These images were made up of iPhone and android image platform and android device platform samples. Besides using data from Digital Corpora, creating image snapshots from volunteer users was considered. This was made possible by using tools such as FTK imager. From this mobile device images, the researcher expected to extract information which was used in the design and implementation of the analysis and presentation layer.

The reason data from Digital Corpora was preferred is because it has diversity and predictability of real data and therefore does not compromise the research findings (S. Garfinkel et al., 2009).

3.3 Data Collection

The mobile images from Digital Corpora are collected from individuals and companies on a voluntarily bases and must conform to certain formats (Digital Corpora, 2017). The acquisition of the data will entail finding the collect data formats needed for the mobile images which in our case can either be E01 Images (*. e01) or SMART Images (*.s01).Utilizing the same methodology as from the Digital Corpora dataset, selected volunteers will also offer cell phone dumps.

3.3.1 Data Collection Methods

The following are the data collection methods used to get the data needed to validate the modeled analysis and presentation layer.

3.3.2 Existing data

This method will entail finding existing data which in this case include mobile image dumps which contain actual messages and call logs. To get this data I will either create mobile dumps or either get the data from existing digital forensic corpus (Digital Corpora, 2017). Creating mobile dumps is made possible by having tools such as FTK imager (Courcier, 2020). This tool saves the content of the mobile details eg, messages and call logs in a specific format which can be absorbed by the tool.

3.4 Data Analysis and Evaluation

The cell phone images from Digital Corpora and the independently collected were utilized for the evaluation of the forensic tools. The images were passed through the various tools to identify difference and similarities in how data is extracted, analyzed and reported back to the user. In this research, the focus was how we extract data which includes call logs and messages in a standardized way that conforms to Nugget and pass the information through our analysis and presentation layer to validate out model. The goal will be to compare the output of the data extracted from the selected tools to find out if there are similarities and differences.

3.5 Ethical Considerations

The data used to test and validate the proposed model was sourced from Digital Corpora (Digital Corpora, 2017) and also from creating images from volunteers. The digital Corpora contains both sensitive and non-sensitive information and therefore the following taxonomies have been developed.

3.5.1 Test Data

This data is constructed specifically for purposes of demonstrating a specific forensic issue or for testing a specific feature in a tool (S. Garfinkel et al., 2009). Test data can be distributed freely on the Internet without any controls.

3.5.2 Realistic data

This data is similar to what an investigator might encounter in an investigation but is artificially constructed. Although there should be no privacy issues when distributing realistic data, there may be copyright issues (S. Garfinkel et al., 2009).

3.5.3 Real but Unrestricted

Data sets can be made available for unrestricted access. For example, the Enron Email Dataset (Klimt and Yang, 2004) is a corpus of 619,446 real email messages from the 158 users inside

the Enron Corporation. These email messages were entered as evidence in a court case by the US Government and, as a result, became publicly available without restriction. Another example of real but unrestricted data are photos that can be downloaded from the Flickr photo sharing website and user profiles on Facebook (Garfinkel et al., 2009). For the additional cell phone images, the researched will seek the appropriate approval for use from the different entities and then be de-identified the images.

Chapter 4: SYSTEM ANALYSIS, SYSTEM DESIGN & IMPEMENTATION

4.1 System Analysis

System analysis is the process of studying a system for the purpose of identifying its objectives, goals, challenges and also identifying areas to improve on. The process also helps understand if the proposed system is worth undertaking.

The incorporation of forensic analysis and reporting in to Nugget involved:

1. Feasibility Analysis.
2. System Modelling.

4.1.1 Feasibility Analysis

Feasibility analysis is the process that takes all of a project's relevant factors into account in determining if the project is worth undertaking. The factors include economic, technical, operational, legal and schedule.

The analysis assesses the practicality of the proposed project in meeting the proposed objectives.

4.1.1.1 Economic Feasibility

This assessment involves a cost/ benefits analysis of a project to determine the viability cost and benefits associated with a project before financial resources are allocated. In the development of an application that incorporates mobile forensic analysis and reporting into Nugget, all required resources were freely available and accessible. Below is a summary of the critical resources used in the development and a description on their availability and affordability.

Resource	**Comment**
Mobile forensic images	iPhone and android mobile forensic images were freely available from Digital Corpora (Digital Corpora, 2017).
RDMS	PostgreSQL RDMS was used to store all processed data. It's open source and hence no cost was incurred.
Open source programming languages.	All programming languages used in the development of the application all open source and hence no cost was incurred.
Server space	For testing purposes, server space from Digital Ocean was purchased. The cost incurred was affordable.

Table 4.1 Summary of some of the critical resource used in the development process.

4.1.1.2 Technical Feasibility

Technical feasibility entails reviewing the availability of software, personnel and hardware for implementing and deploying the system. All expertise and tools needed in the development were available. In terms of expertise, it was a learning process in the whole development process as some tools were new to the researcher.

4.1.1.3 Operational Feasibility

Operational feasibility involves undertaking a study to analyze and determine how well the current needs can be met by completing the project. Operational feasibility also examines how the project plan meets the requirements identified in the requirement analysis phase.

The application developed in this research will help in the analysis and reporting of mobile forensic cases.

4.1.1.4 Schedule Feasibility

This assessment is one of the most important for the project's success. The study seeks to estimate how much time is required to complete the project and if the solution can be designed and implemented within an acceptable time period. Though the initial schedule of the project was 6 weeks additional time was spent due to unexpected factors. One of the factors is the steep learning curve required in some of the technologies used.

4.1.2 System Modeling.

4.1.2.1 Use Case Diagrams

Use case diagrams summarizes the details of the system's users and their interaction with the system. The aim of use case diagrams is to illustrate the different ways that a user might interact with the system when analyzing any mobile forensic case.

The use case of this platform consists of one main actor:

a) Mobile forensic analysts.
   In this case a mobile forensic analyst is assumed to be any person interacting with the system with the purpose of analyzing a case.

The activities of a mobile forensic analysts include:

1. Creating a case.
2. Listing cases.
3. Analyzing a case.
4. Viewing case reports.
5. Interpreting case reports.

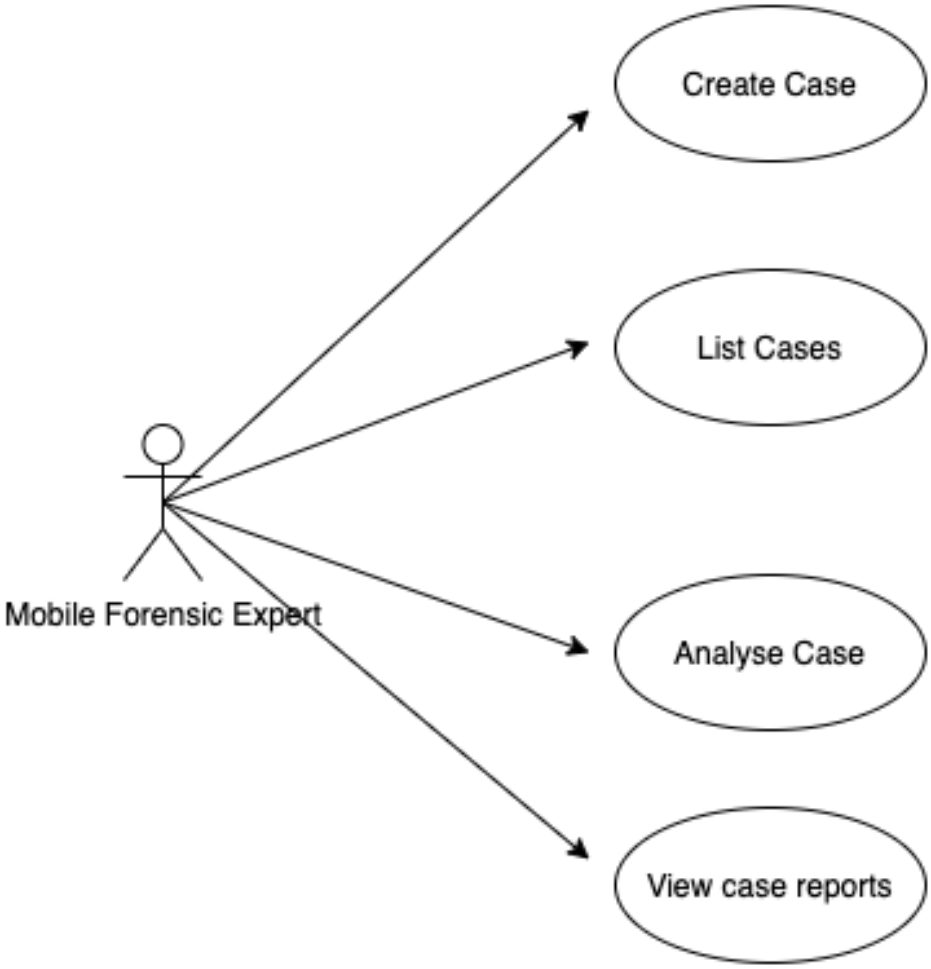

Figure 4.1 Use Case Diagram - Mobile Forensic Analyst

4.1.2.2 Functional Specifications.

Functional specifications describe and applications intended capabilities and interactions with the user. It also describes users' task and dependencies on other applications and the usability criteria. A functional specification document has two main categories which are described below.

1. Functional requirements.

   These are requirements are the functions that an application must perform in order to fulfil the business requirements.

2. Non-functional requirements.

   These are non-essential features not at the core of the application. They define the overall qualities of the application.

4.1.2.2.1 Functional Specifications.

They include:

1. The system should provide a way to create a case.

2. The system should generate mobile artifacts for analysis, includes SMS, call logs and location data.
3. The system should generate and analyze the case under study.
4. The system should report the analyzed findings to the mobile forensic analyst for decision making.

4.1.2.2.2 Non-Functional Specifications.

They include:

1. Ease of use of the application.
2. Efficiency in time creating and analyzing a case.

4.2 System Design

Designing a system entails defining elements that build up a system like modules, architecture, data flows, interfaces and their components according to the specified requirements. In this phase, the design process was broken down into the following categories.

1. System Architecture.
2. Schema design
3. Interface design.

4.2.1 System Architecture.

This architecture defines the structure and software components as conceptualized in the proposed application. In the conceptualization process, the C4 model was used to describe the various application software components. The C4model considers the software system in terms of four different levels (Brown, 2020):

1. System Context diagram

   The system context diagram enables the system architect to conceptualize the application round the user interactions and also other software systems. The details at this level are not important.

2. Container diagram

   A container is a specific module form the system context diagram. In the context of this research, the container diagrams included the web API, analysis server, nugget tools and the Nugget runtime. The definition of the high-level shape of the software architecture and how roles are distributed is presented by the container diagram (Brown, 2020).

3. Component diagram

   The component diagram indicates what the container is comprised of, what each of those components are, their responsibility and the implementation details. In this research, each of the containers were analyzed and this informed the implementation phase.

4.2.1.1 System Context diagram

Below is the system context diagram of the proposed application which incorporates analysis and reporting into Nugget. This is a high-level diagram which highlights how the proposed Nugget App interacts with the mobile forensic analyst, Nugget runtime and Nugget Tools.

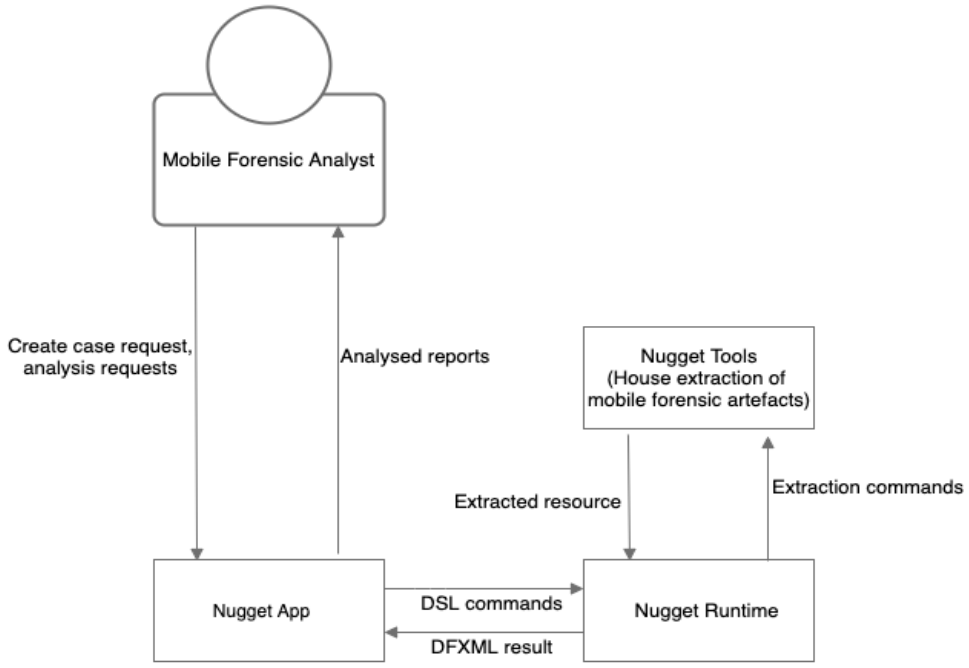

Figure 4.2 System context Diagram

4.2.1.2 Container Diagram

4.2.1.2.1 Nugget App

The Nugget app container diagram shows a detailed view of the proposed and implemented software application. The proposed design had three major components: Nugget web application, Nugget API server and a PostgreSQL relational database management system. The mobile interacts with Nugget web application which is responsible of making network calls to the Nugget API server. The model also indicates how calls to Nugget runtime and Nugget tools are made.

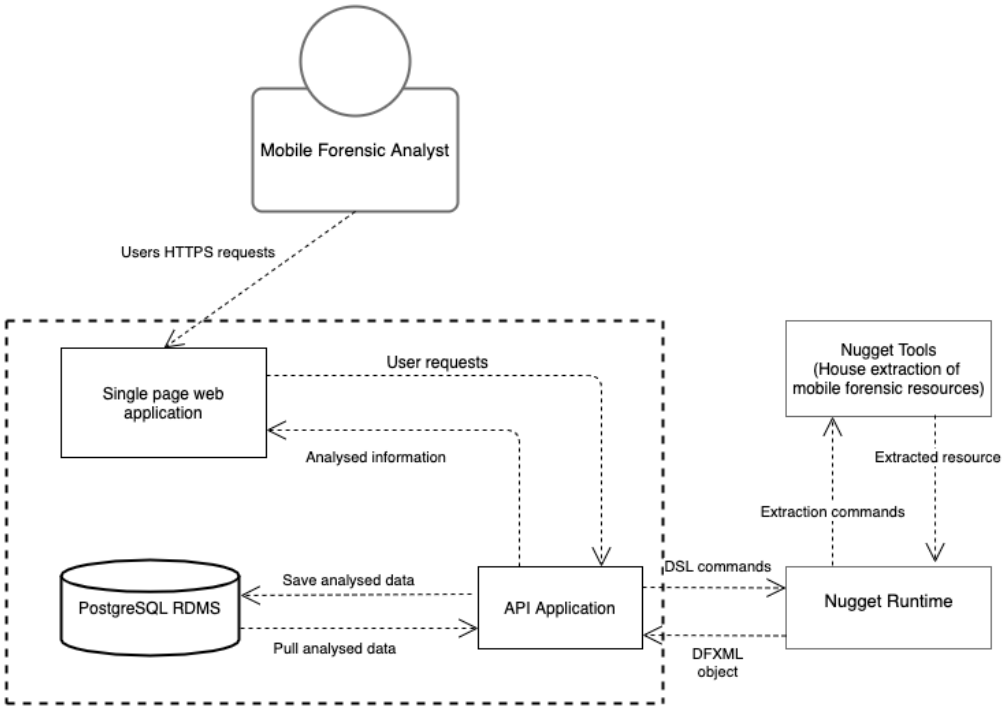

Figure 4.3 Nugget App container diagram

4.2.1.3 Component Diagram

4.2.1.3.1 API Application

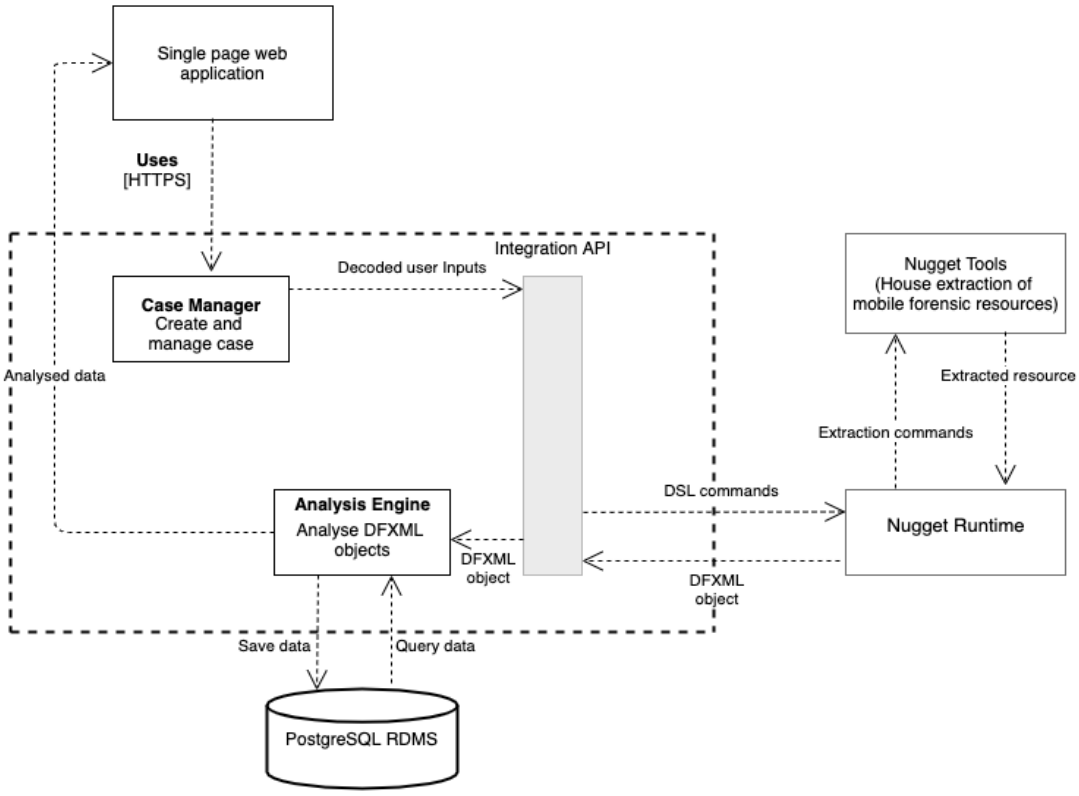

Figure 4.4 API Application component diagram

The above component diagram is a low-level proposed implementation of the Nugget API. The Nugget API comprises of 3 main components:

- Case Manager
- Analytics Engine
- Integration API.

4.2.2 Schema Design

The design approach taken was such that, the extracted mobile forensic data was to be used in a database before any analysis was done. To hold this data, schema designs were made to identify what models would be necessary to store the data.

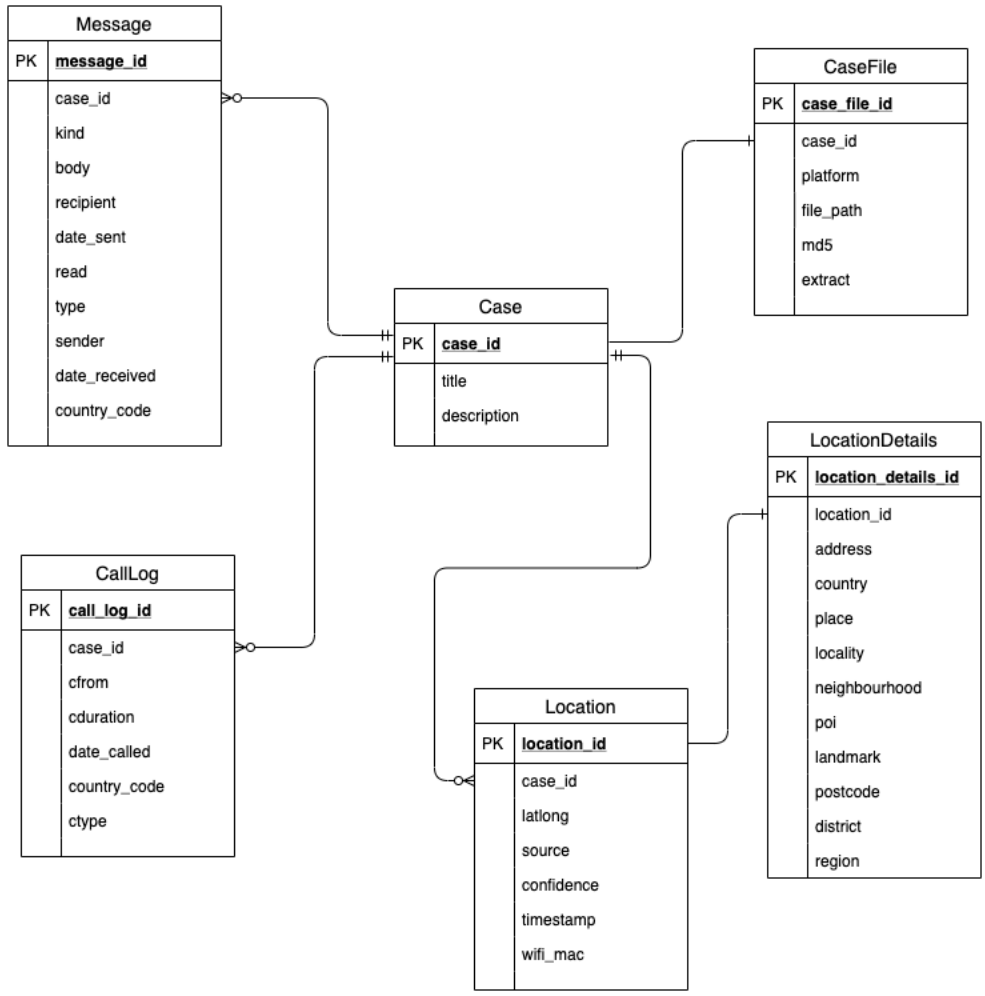

Figure 4.5 Schema Design

4.2.3 Interface Design

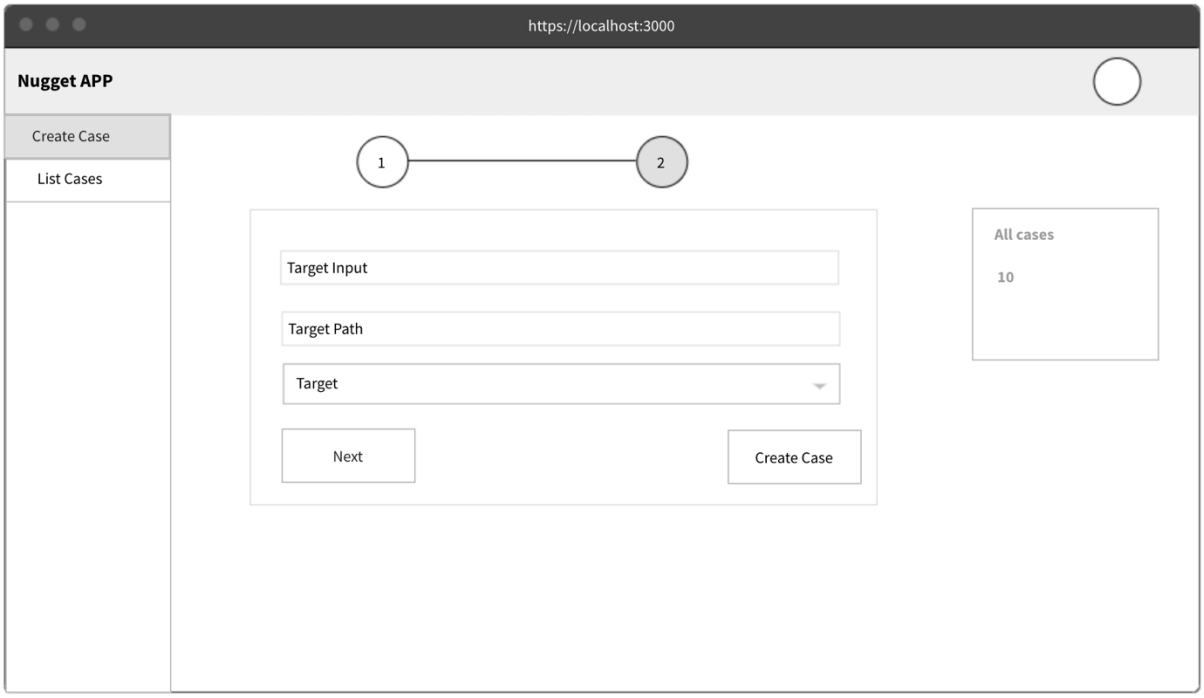

Figure 4.6 User Interface Design for Case creation

4.3 System Implementation

4.3.1 Resources

4.3.1.1 Hardware Resources

1. MacBook Pro Laptop
2. Processor 4 CPU's; 2.7GHz
3. RAM – 8GB
4. Hard Disk – 126 GB (SSD)
5. External Hard Disk

4.3.1.2 Software Resources

1. Operating System – macOS Catalina 10.15.4
2. VIM – Integrated Development Environment (Open Source)
3. Atom - Integrated Development Environment (Open Source)
4. Django Web Framework (Open Source)
5. Django Rest Framework (Open Source)
6. ReactJS (Open Source)
7. Draw.io for mockups design (Free)

4.3.2 Programming tools, techniques and technologies

4.3.2.1 Django

Django is a python based, open source web-framework with a model-template-view architectural pattern (Holovaty and Willison, 2020). Django was developed with an aim of faster prototyping and taking care of the hassle of web application development. Some of the notable features of Django include being:

1. Fully loaded

   Django include a collection of extra resources that makes web application development extremely fast.

2. Fast

   One Django design principle is to help software engineers take applications from concept to completion as quickly as possible.

3. Versatile

   Django have been used by companies and corporations to build all sorts of applications ranging from social network applications to scientific applications.

4. Scalable

   Django has the ability to quickly and flexibly scale to meet the customer's needs.

Django was utilized in the development of the backend analysis server. This also housed the implementation of the integration between Nugget API and Nugget runtime.

4.3.2.2 Django Rest Framework

Django Rest Framework is a flexible and powerful toolkit for building web APIs (Christie, 2020). Some of the notable features of Django Rest Framework are:

1. Serialization

   This is the process of converting complex data types into native data types that can be easily rendered into XML, JSON or other content types.

2. Authentication Policies.

   Django Rest Framework include authentication packages which makes authentication out of the box.

Django Rest Framework was used to create web APIs which were consumed by the web frontend application.

4.3.2.3 ReactJS

ReactJS is a JavaScript library for creating user interfaces. Its declarative nature makes it simple to create interactive UIs and easy to debug any programming errors (React – A JavaScript library for building user interfaces, 2020). ReactJS was used in creating UI components that rendered the analysis of the forensic tool.

4.3.3 Implementation

4.3.3.1 Nugget DFXML Extension

Digital forensic XML is an XML language that automates digital forensic processing and enables the exchange of structured forensic information. DFXML improves composability by providing a language for describing common forensic processes, forensic work products and metadata. DFXML implementations have been used for a variety of purposes:

- The XML format makes it easier to share data with other organization across different forensic tools.
- The DFXML format record which version of tool produces each file and therefore easy to reprocess.
- DFXML can be used to document provenance which include the computer on which the application program was compiled.

Though Nugget was capable of serializing the various extracted data from a mobile device, the exchange of the serialized data was missing. In this phase of the project implementation, an extension was implemented into Nugget in order to allow the exchange of the extracted Phone call logs, Messages and Location information in DFXML format from Nugget runtime to Nugget APAI. This was later consumed in the analysis and reporting phase.

- **Phone Call**

  The extracted phone calls were serialized into DFXML format with the representation being as below.

```
<?xml version=”1.0” encoding=”UTF-8”?>
 <mobile:call>
   <mobile:from>5713083236</mobile:from>
   <mobile:date_called>2012-07-06T18:18:50+03:00</mobile:date_called>
   <mobile:duration>244</mobile:duration>
   <mobile:country_code>310</mobile:country_code>
   <mobile:type>incoming</mobile:type>
 </mobile:call>
```

- **SMS**

  Messages extracted and serialized were represented in the format below: These were the important attributes which would were then consumed for analysis.

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile:sms_mms>
   <mobile:kind>sms</mobile:kind>
   <mobile:sender>+12027252124</mobile:sender>
   <mobile:body>I&#39;m almost there where should I meet you? </mobile:body>
   <mobile:date_received>2012-07-11T15:41:45+03:00</mobile:date_received>
   <mobile:read>1</mobile:read>
   <mobile:country_code>us</mobile:country_code>
   <mobile:type>inbox</mobile:type>
 </mobile:sms_mms>
```

- **Location**

  The location details extracted and serialized took the below form.

```
<?xml version="1.0" encoding="UTF-8"?>
<mobile:location>
   <mobile:long>-77.1114161</mobile:long>
   <mobile:lat>38.87981969</mobile:lat>
   <mobile:source>
   </mobile:source>
   <mobile:confidence>50</mobile:confidence>
   <mobile:timestamp>2012-07-05T19:32:46+03:00</mobile:timestamp>
   <mobile:cell_mcc>310</mobile:cell_mcc>
   <mobile:cell_mnc>410</mobile:cell_mnc>
   <mobile:cell_lac>7985</mobile:cell_lac>
   <mobile:cell_ci>158587935</mobile:cell_ci>
</mobile:location>
```

4.3.3.2 DSL generation

A computer language specialized to a particular domain is known as a domain specific language. To facilitate the interaction of the web application with Nugget, commands that

Nugget understands were created. User inputs from the web application, were interpreted and channeled to Nugget and formed the basis of the applications interactions.

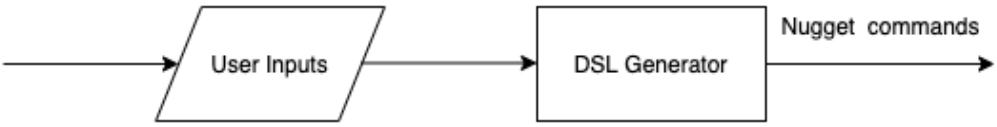

Figure 4.7 Nugget invocation flow

The user inputs which were consumed and translated to Nugget commands includes a combination of the following parameters:

- **Mobile device platform.**
  Supported mobile device platforms which in this case are Android and IOS.
- **What Data to Extract.**
  The data to extract which include, SMS, phone calls and location data.
- **Path to target.**
  This is the location path to where the mobile forensic target is stored.

Because Nugget does not understand the above parameters as they are, a translation was made to form the commands below:

```
iphonecall_dsl="iphone:/Volumes/Documents/CDan/Iphone/tracy-phone-2012-07-15-
final.tar" | extract as call
print construct_iphonecall as dfxml
```

Listing 4.1 Nugget command to extract iphone calls

```
iphonelocation="iphone:/Volumes/Documents /CDan/Iphone/tracy-phone-2012-07-15-
final.tar" | extract as location
print construct_iphonelocation as dfxml
```

Listing 4.2 Nugget command to extract iPhone location data

```
iphonemessage="iphone:/Volumes/LOVE-BANGA/CDan/Iphone/tracy-phone-2012-07-
15-final.tar" | extract as message
print construct_iphonemessage as dfxml
```

Listing 4.3 Nugget command to extract iPhone SMS data

4.3.3.3 Data Service Layer

A data service layer is an integration layer of computer programs which provides a simplified access of data and invocation commands. To integrate the three main applications, a DSL was implemented which facilitated the exchange of message invocation requests and data across the applications. User requests from the web applications were channeled to the mobile forensic nugget tools for extraction through Nugget and back to the analysis application for analysis and finally reporting.

The implemented data service layer served two major functions:

- Channel user translated commands to Nugget.
- Receive the output from Nugget for Analysis.

4.3.3.4 Subprocess Interface

The subprocess interface provides the communication between the Nugget App and Nugget runtime. Specific arguments are sent to Nugget runtime and the output received in DFXML format from Nugget runtime. Below is a summary of the arguments consumed by the subprocess interface.

Argument	Description	Required?
Nugget run command	The Nugget command to invoke Nugget runtime	Yes
Path to Nugget runtime	The path to Nugget runtime	Yes
Nugget DSL commands	User translated inputs from the web application	Yes

Table 4.2 Arguments passed onto the subprocess interface

The result of the subprocess interface is wrapped in a DFXML object and sent to the analysis server.

4.3.3.5 Mobile Forensic Data Analysis

To make meaning of the data received from Nugget which is in DFXML format, data analysis was done. The analyzed data was then used in the reporting phase. In the analysis phase, the focus was to capture the important aspects of information that a mobile forensic expert is interested in when carrying out a forensic investigation. The data analysis process was informed and guided by the data received and its attributes received from Nugget. Due to the

distinct mobile platforms used in this research, a study of the data attributes was done to harmonize the analysis.

4.3.3.5.1 IOS Platform

IOS forensic images from Digital Corpora were used in this research. The IOS images provided SMS, phone calls and location data for analysis. The data attributes under each category was as follows:

- **SMS**

  The attributes of the analyzed SMS object are discussed below.

Field	Type	Description
from	string	The category of the SMS, either sms or mms.
date_called	string	The party who sent the message.
duration	string	Content of the message
country_code	string	Sender phone number country code.
type	string	Indicates the statue of the message, sent or outbox.

Table 4.3 Attributes of the analyzed SMS object

- **Phone Call**

  From the phone call DFXML the following attributes were extracted which formed the basis of the analysis process.

Field	Type	Description
kind	string	The category of the SMS, either sms or mms.
sender	string	The party who sent the message.
body	string	Content of the message
date_received	string	When the message was received by the recipients
read	int	Status of the message on the recipient side.
country_code	string	Sender phone number country code.
type	string	Indicates the statue of the message, sent or outbox.

Table 4.4 Attributes of the analyzed Phone Call Object

- **Location**

  The location DFXML had the following attributes.

Field	Type	Description
long	float	Longitude coordinate.

lat	float	Latitude coordinate
confidence	int	Accuracy of the latlong coordinates
timestamp	string	Time the data was captured
cell_mcc	int	Cell mobile country code.
cell_mnc	int	Cell mobile network code
cell_lac	int	Cell local area code
cell_ci	int	Cell identity

Table 4.5 Attributes of the analyzed location Object.

4.3.3.5.2 Android Platform

Data extracted from the android platform was also in DFXML format. Information extracted under this were SMS and phone calls. Below is a summary of the attributes under each extracted information.

- **SMS**

Field	Type	Description
type	string	The type of the message eg inbox, draft, sent
address	string	The address of the other party
body	string	Content of the message
date	string	When the message was received by the recipients
read	int	Status of the message on the recipient side.
status	string	Indicates the status of the message, sent or outbox.

Table 4.6 Android SMS DFXML object attributes.

- **Phone Call**

Field	Type	Description
from	string	The category of the SMS, either sms or mms.
date_called	string	The party who sent the message.
duration	string	Content of the message
country_code	string	Sender phone number country code.
type	string	Indicates the statue of the message, sent or outbox.

Table 0.7 Android Phone call DFXML object attributes.

4.3.3.6 Timeline Analysis

From the data extracted, it was important to give a detailed analysis which would give insights to the mobile forensic investigator. Since the extracted data comprised of messages, phone calls and location information, a reconstruction of this information was made. The key questions in the implementation of the timeline analysis were:

1. How many calls were made in a specific day?
2. To whom were the calls made?
3. How many calls were received in a specific day?
4. What time were the calls made?
5. How many SMS were sent or received in a specific day?
6. What was the content of the SMS?
7. Is there any identifiable location information?

The above questions yielded a module that would show individuals contacted through messages or calls together with any location information available.

Chapter 5: RESULTS & DISCUSSIONS

We discuss and highlight results achieved in the incorporation of mobile forensic analysis and reporting into Nugget in this chapter.

5.1 iPhone Image Platform

For the iPhone platform, multiple images from the same iPhone device were used. The images were obtained from a simulated scenario about an attack in a city (Digital Corpora, 2018). The iPhone used was from a fictional character in this scenario called Tracy.

From the iPhone image the resources extracted included:

- SMS
- Phone Calls
- Location Information

5.1.1 SMS

The SMS DFXML object was analyzed to determine important information that a mobile forensic analyst might be looking for when carrying out a mobile forensic investigation.

5.1.1.1 SMS received by day analysis

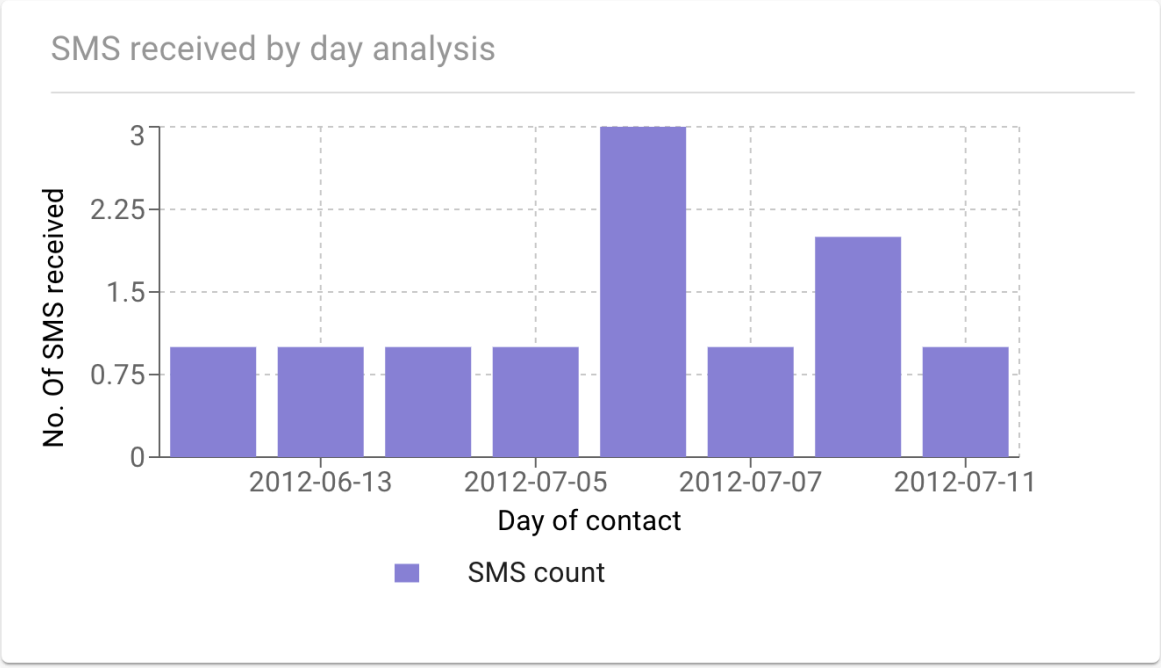

Figure 5.1 SMS received by day analysis

The graph above shows the breakdown of the number of SMS received by day. On clicking on each bar, one is able to know how many SMSs' were sent per day. In order to make this information useful to the mobile forensic analyst, a breakdown to determine the people who

sent the messages was done. This information is available on clicking on a specific bar in the graph.

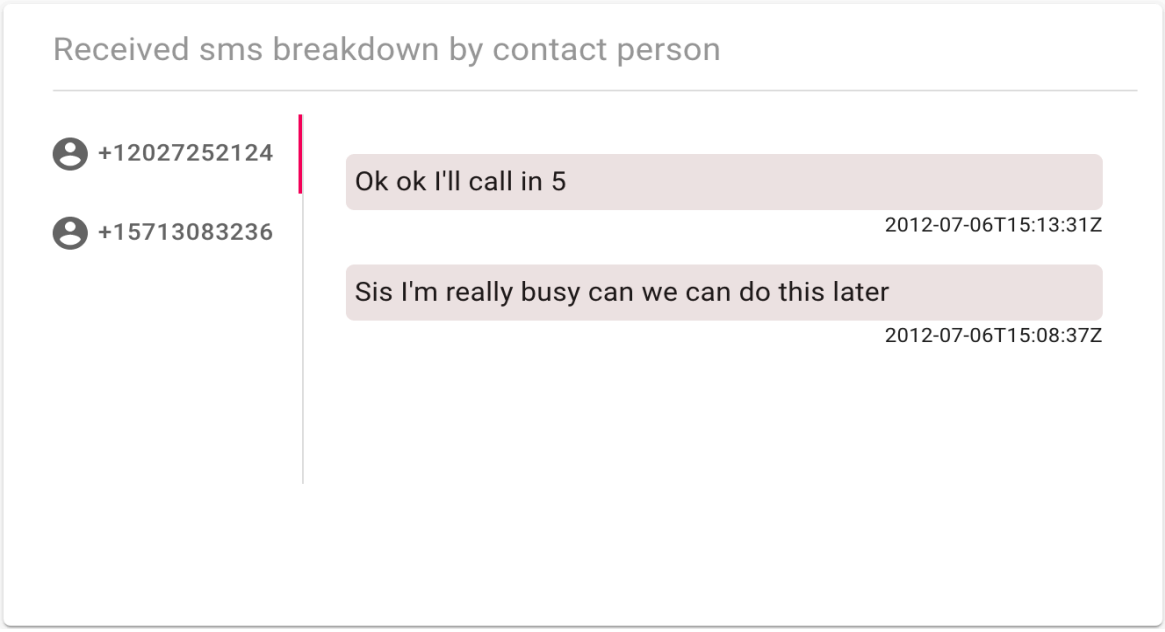

Figure 5.2 Received SMS breakdown by contact person

The above image shows the people who sent the messages and a reconstruction of the messages to view the exchanged information.

5.1.1.2 SMS sent by day analysis

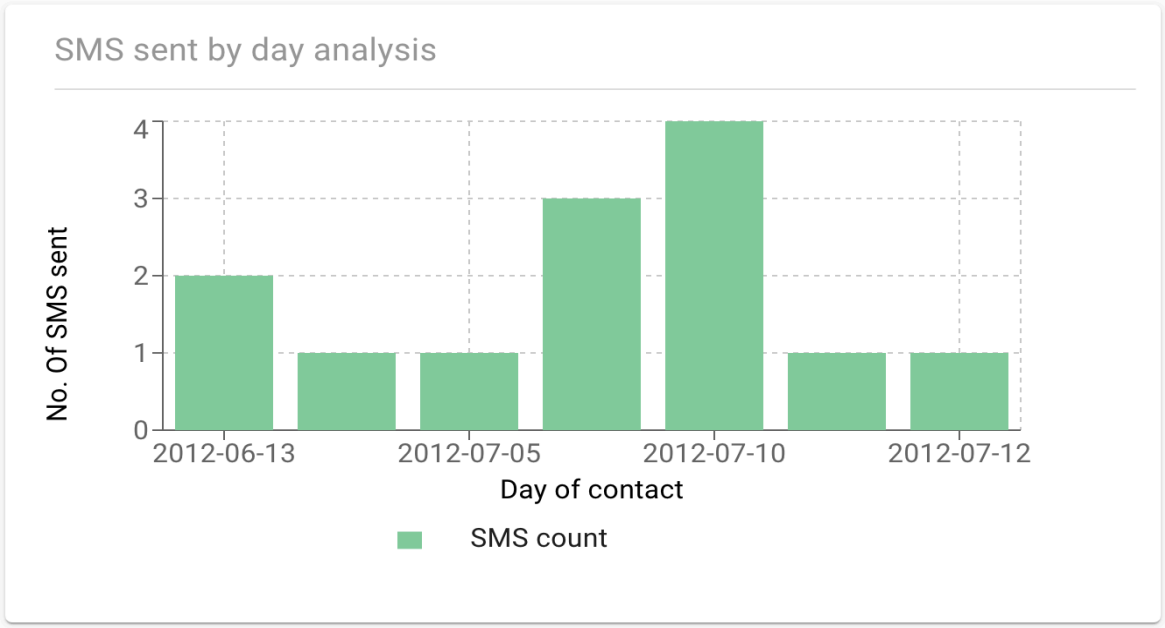

Figure 5.3 SMS sent by day analysis

This analysis was centric to the messages sent by day. From the above graph, the researcher was able to identify the number of messages sent by day. The graph also provided a channel of getting more information about the exchanges messages. On clicking any bar on the graph, we are able to get who the messages were sent to.

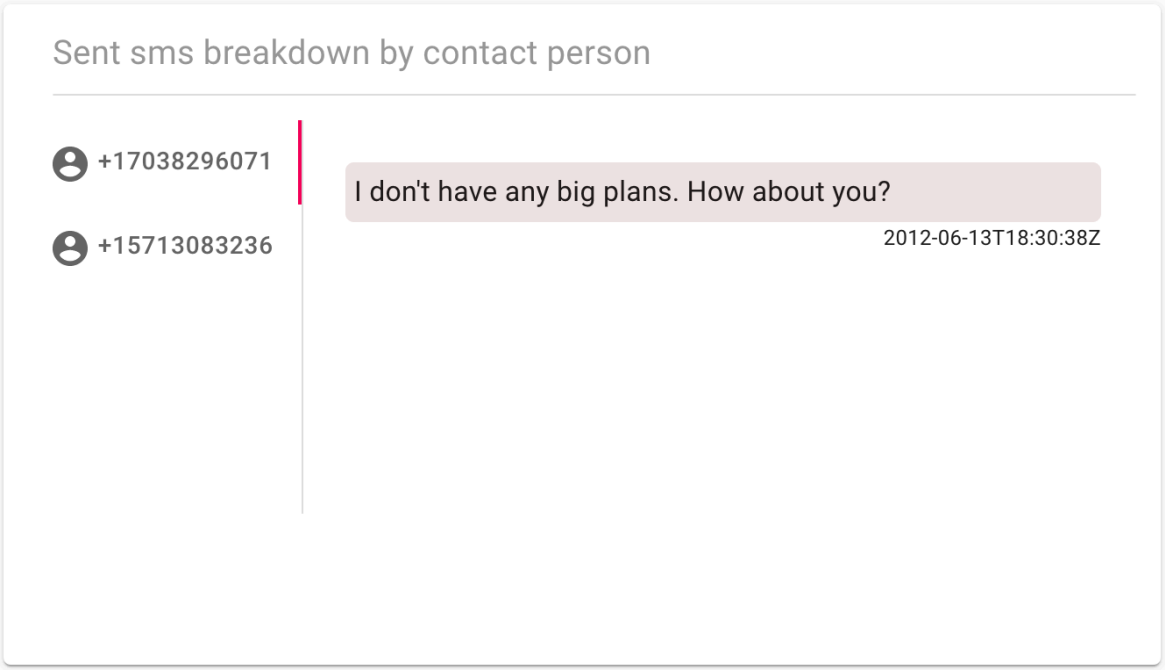

Figure 5.4 Sent SMS breakdown by contact person

The result of the sent and received SMS was combined into one graph. This was important in order to identify how many messages were sent or received per day. With this, a mobile forensic analyst is able to identity the activity per day in regards to messages. To differentiate the distinct categories, different color codes were used and on hovering on a specific bar, important information such as the count and date was made available.

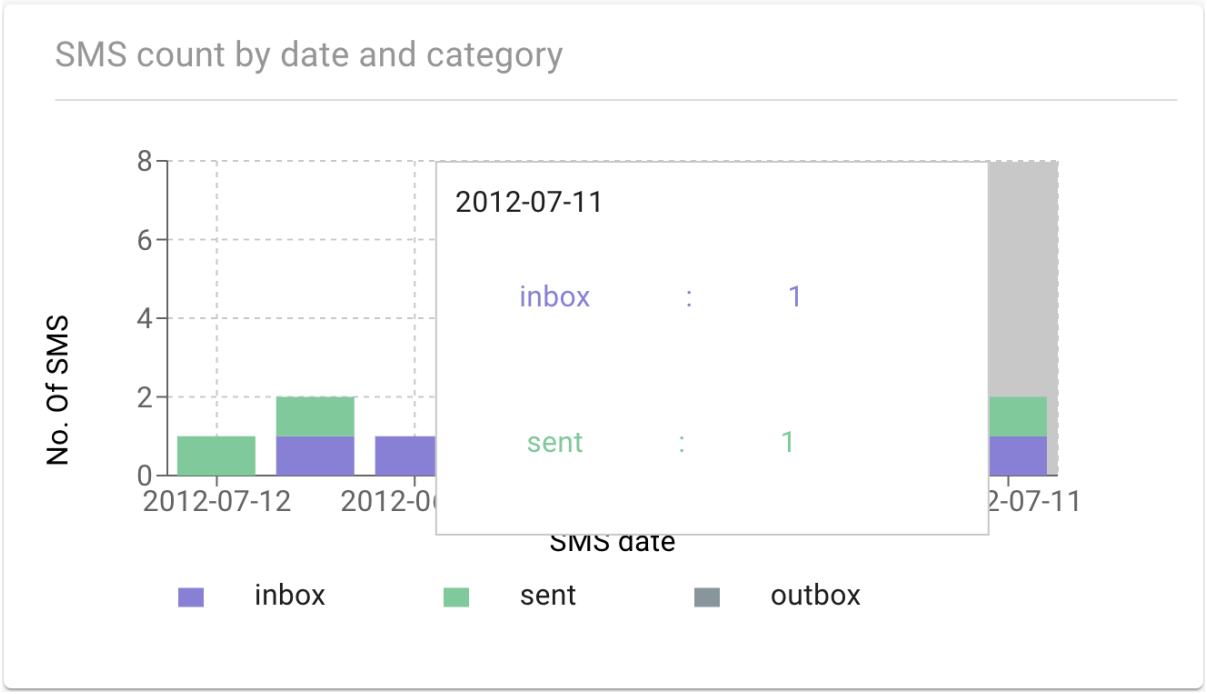

Figure 5.5 SMS count breakdown by date and category

5.1.2 Phone Calls

The analysis of the phone call DFXML object yielded important insights indicating when a phone call was made or received, from who and how long the call lasted. The categorization in this part of the research was:

- Incoming calls. These are call received.
- Outgoing calls. These are calls made to other persons

5.1.2.1 Incoming call count by day analysis

The result of this analysis was information on calls received by specific days. When plotted in a bar graph, we were able to identify how this information compared to other days, useful insights when making mobile forensic decisions.

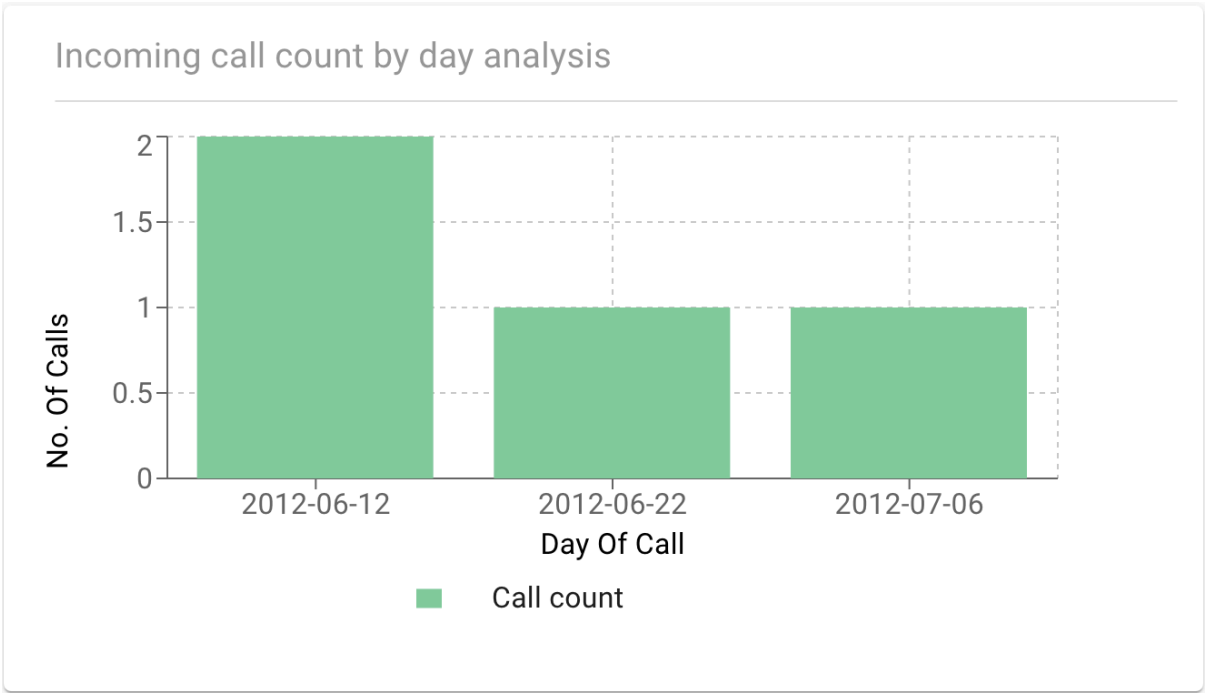

Figure 5.6 Incoming call count by day analysis

For meaningful information from the above graph, on clicking a specific bar on each day, one is able to know the contact persons on that day. Further information such as the country code, duration of the call and time of the call is also available against each contacted person.

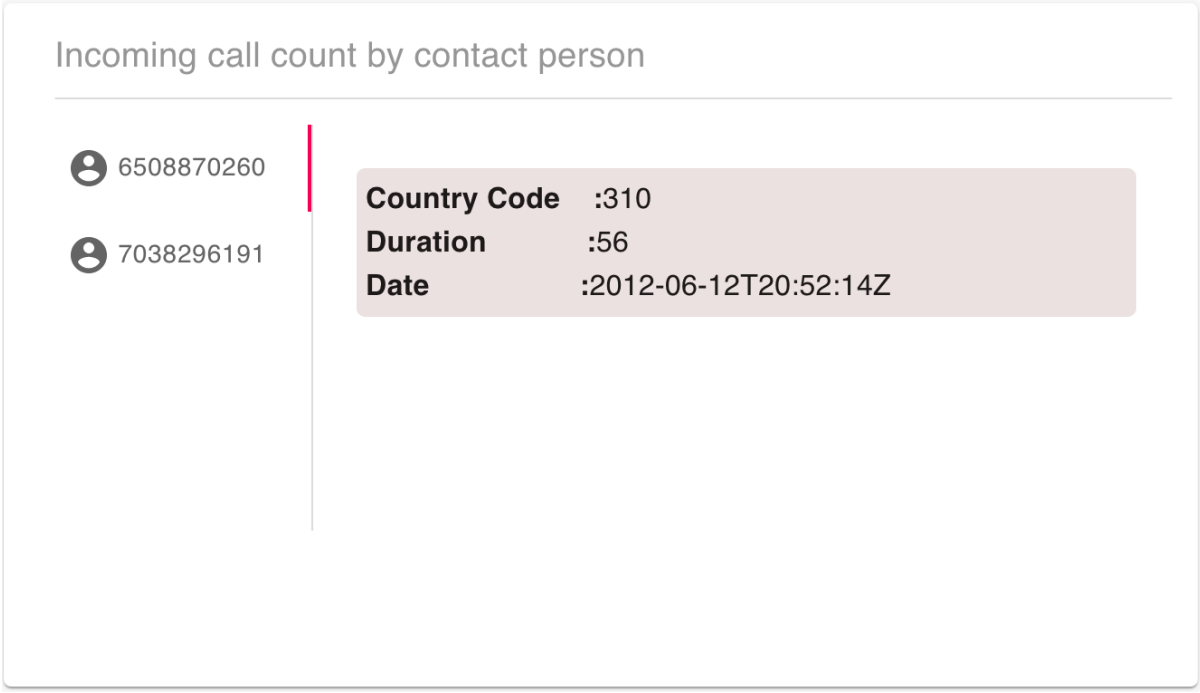

Figure 5.7 Incoming call count by contact person

5.1.3 Timeline Analysis

A timeline analysis takes into consideration more than one factor around a subject under investigation. Using a timeline analysis, the researcher was able to reconstruct activities such as sending and receiving SMS, phone calls and location information into one component. This serves to answer questions such as:

1. Who did the subject send the message to?
2. What conversation did the subject have with the contacted persons?
3. What location information can be associated to the specific time when the conversation happened.

The combination of the SMS, phone calls and location information provided a representation as shown below. The researcher was able to reconstruct conversations with each contact person including any phone calls made or received, messages sent or received and also location information.

It's however important to highlight that in the association of conversations and calls with location information, there was an acceptable margin of deviation in the computations. In the association, the available location information was after or before a conversation happened either through SMS or phone calls.

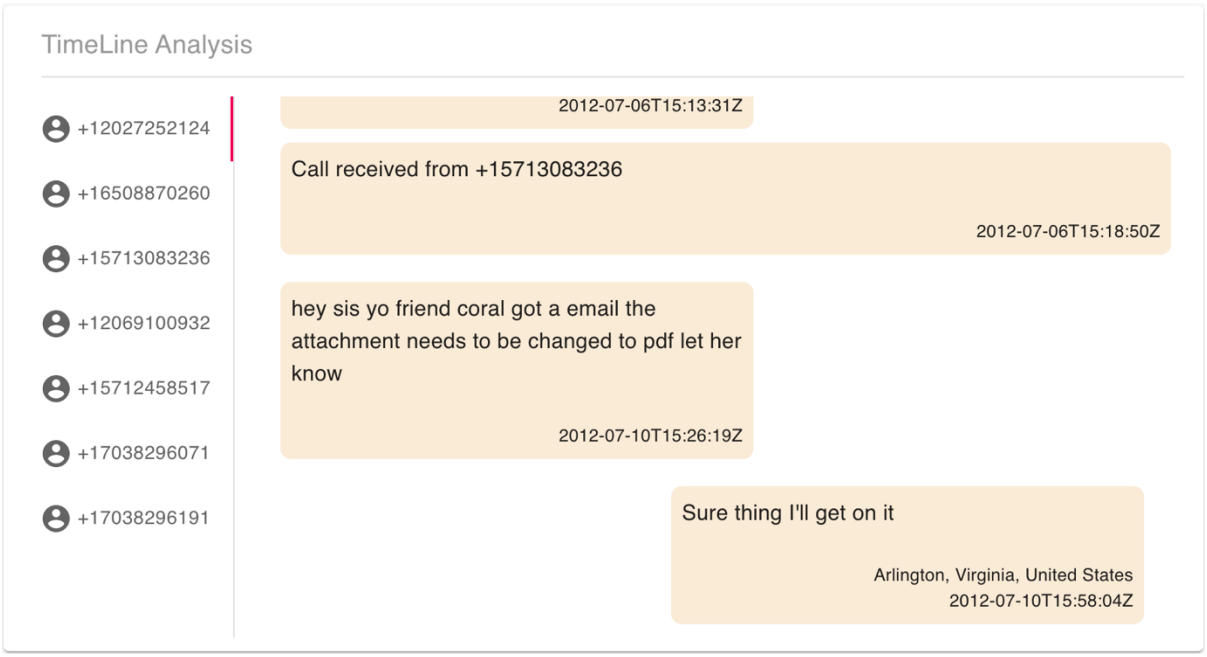

Figure 5.8 Timeline analysis

5.1.3.1 Timeline analysis on Map

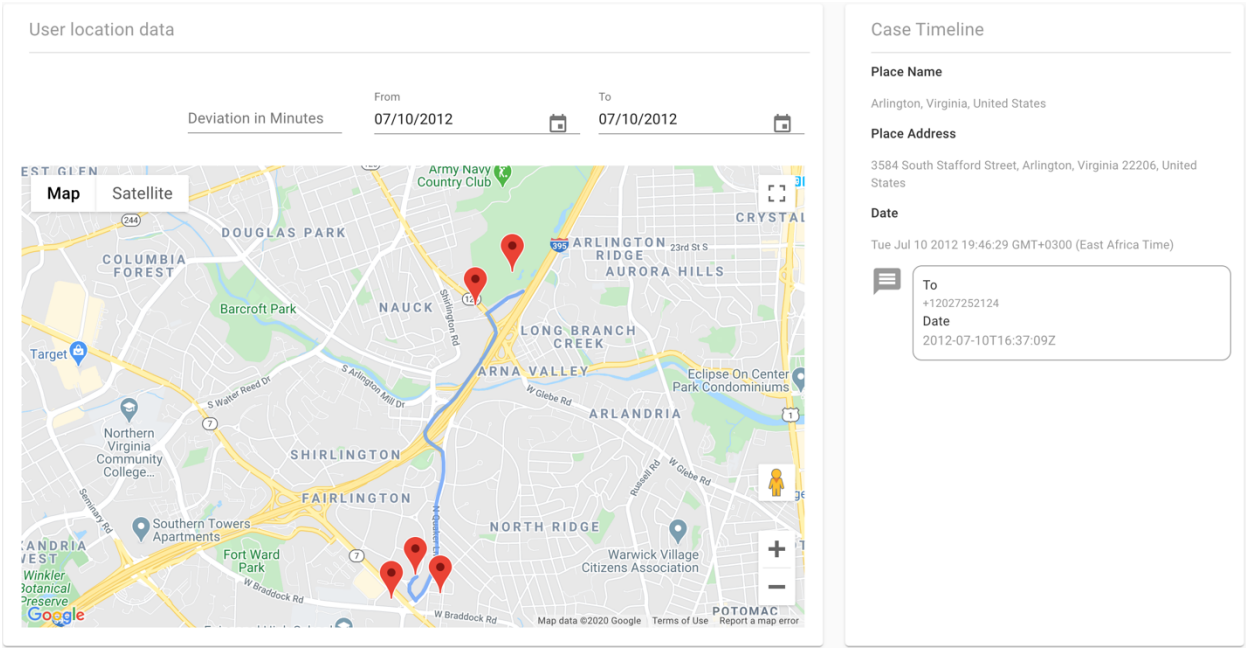

Figure 5.9 Timeline analysis on Map

Another implementation of a timeline analysis was visualizing the users' location on a map with extra information such calls sent out or received and messages received or sent out. The mapped points were received from the location information DFXML after which reverse geocoding was done to get the readable location names. Since a call can be made after or before getting the location information, an input box to get the deviation in minutes was provided.

Chapter 6: CONCLUSIONS AND RECOMENDATIOS

6.1 Summary of the research

The overall objective of the study involved incorporating mobile forensic analysis and reporting into Nugget. This objective was achieved by designing, implementing and testing a system that integrated with Nugget and Mobile forensic tools to extract, analyze and report on mobile forensic evidence given a mobile forensic target.

The research had specific objectives that were to be achieved. The first objective was to evaluate existing models used in digital forensic analysis and reporting of digital forensic information. This objective was met by identifying and studying existing models used in digital forensic analysis and reporting. From the research, two models were identified, 2) Timeline analysis b) Link analysis. These models were used in the study. The results of this objective informed the kind of analysis and what form of representations were carried out.

The second objective was to design an architectural model that informed Nugget integration to allow for rich, meaningful analytics of the forensic findings. This evolved from a high-level architectural design to a low level detailed architectural design. This objective was achieved by designing a model of the proposed system which defined the interaction of the different components. This design was important as it informed the implementation of the actual proposed system.

The third objective was to develop a mobile forensic middleware comprising of a data service layer, analytics engine and a presentation layer from the derived architectural model. Nugget and Mobile forensic tools are two distinct applications which are loosely coupled. The design makes it easy to update, upgrade or make any changes in either of the applications. To preserve this distinct feature, a mobile forensic middleware was designed and developed. This layer comprised of three major components, a) data server layer, b) analytics engine c) presentational layer. The data service layer ensured communication between the web application and Nugget. The information exchanged in this service are Nugget commands and mobile forensic evidence in raw form. The analytics engine was responsible of converting mobile forensic evidence in

DFXML format into information which can be presented to mobile forensic analyst for decision making. The presentation layer was responsible of displaying the analyzed data in form on graphs, tables and maps.

The fourth objective was to develop an integration layer between Nugget and available digital forensic tools for forensic data exchange. This integration was important as data from Nugget and mobile forensic tools to the data service layer had to be serialized. Data in this format also mean other applications can be integrated since the exchange of data is in a standardized format.

The fifth and the last objective was to validate the architectural model using publicly available digital forensic targets. This objective was important as its output informed if this research had fully met its objectives. In this context, publicly available digital forensic targets of Android and IOS mobile images from Digital Corpora were used and also from Android emulator SDK. The validation of the architectural model was through an interactive web application where mobile forensic analysts would create, list and analyze cases.

6.2 Challenges & Limitations

- **Inconsistences in evidence collected from mobile target images.**

  From the carried-out research, the goal was to extract any evidence from mobile devices which can be used to create and analyze a case. The extracted data was call log, SMS message and location data information. The expected behavior would be, e.g. when dealing with call logs, we should get all attributes which make up a single call log. The attributes are, a) the caller ID b) recipient ID, c) timestamp and the duration. In some instances, we had missing attributes which made it hard to fully rely on such data since it introduced inconsistences in the analysis.

  In regards to SMS messages, some missed the text body content or the sender and recipient ID. In such scenarios, the extracted information was not useful in making informed decisions.

- **Access to mobile target images and devices**

  To validate the implemented system, mobile target images were required. It's from these images that the researcher was able to extract evidence which include call log, SMS and location data. The target images used were accessed from Digital Corpora. The platform however does not have the diversity of various versions of a given platform. Android emulators were easily accessible though that was different for IOS platform.

6.3 Recommendations for future work

After carrying out this research, a number of items were identified that can be the basis of future work. The items include:

- **Extend information analysis capabilities.**

  This research was constrained to only analyzing call logs, SMS messages and location data. It's however important to note that we have other rich sources of important information that can be of great value in mobile forensics. These sources include modern messaging applications such as WhatsApp, Twitter and Facebook. These applications can be a rich source of information such as location data, messaging data. Other sources of data are in applications such as Calendar whose data can be correlated with other information to enhance the analysis.

- **Support analysis of more than one correlating target.**

  While performing forensic analysis, a researcher might be presented with a more complex case that requires analysis of more than one target at a go. This might be important if one would like to perform a link analysis to identify if different parties might be linked in one way or the other. E.g. Assuming two individuals were in a crime scene and a research would like to identify if there was any relationship between the parties.

6.4 Conclusion

In this study, we demonstrated the incorporation of mobile analysis and reporting into Nugget. This was achieved by converting serialized DFXML output from mobile forensic tools to meaningful information which mobile forensic investigators can consume.

To facilitate the process of mobile forensics, an interactive web application was developed from where a mobile forensic investigator can create a case and view case results. The study also demonstrated the relationship of the different digital forensic phases which are seizure, extraction, analysis and reporting.

REFERENCES

1. Abdulla, K., Jones, A. and Martin, T., 2012. Forensics data acquisition methods for mobile phones. *Research Gate*,.
2. Brothers, S. (2011). How cell phone "forensic" tools actually work–cell phone tool leveling system. In DoD Cybercrime Conference.
3. Brown, S., 2020. *The C4 Model For Visualising Software Architecture*. [online] C4model.com. Available at: <https://c4model.com/> [Accessed 18 May 2020].
4. Carrier, B. (2018). [online] The SleuthKit. Available at: https://www. sleuthkit.org [Accessed 2 Jan. 2020].
5. Chabot, Y., Bertaux, A., Nicolle, C. and Kechadi, M. (2014). A complete formalized knowledge representation model for advanced digital forensics timeline analysis. Digital Investigation, 11, pp. S95-S105.
6. Christie, T., 2020. *Home - Django REST Framework*. [online] Django-rest-framework.org. Available at: <https://www.django-rest-framework.org/> [Accessed 16 May 2020].
7. Chomba, E. and Abade, E., 2018. Incorporating Mobile Forensics into Nugget. *Unpublished manuscript*.
8. Digital Corpora. (2017, April 29). Digital Corpora. Retrieved November 5, 2018, from https://digitalcorpora.org/
9. Digital Corpora. (2018, July 27). National Gallery DC 2012 Attack. Retrieved November 5, 2018, from https://digitalcorpora.org/corpora/scenarios/national-gallery-dc-2012- attack
10. Digital Corpora. (2015, May 17). Retrieved November 5, 2018, from http://downloads. digitalcorpora.org/corpora/mobile/2011-android/
11. Fowler, M., 2010. *Domain-Specific Languages*. Boston, Mass: Addison-Wesley Professional.
12. Garfinkel, S., Farrell, P., Roussev, V. and Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. Digital Investigation, 6, pp.S2-S11.
13. Google. (2018). Android Debug Bridge (adb) | Android Developers. Retrieved December 20, 2018, from https://developer.android.com/studio/command-line/adb
14. Holovaty, A. and Willison, S., 2020. *The Web Framework For Perfectionists With Deadlines | Django*. [online] Djangoproject.com. Available at: <https://www.djangoproject.com/> [Accessed 16 May 2020].
15. Hoog, A. (2011). Android forensics. Waltham, MA: Syngress.
16. Karen, K., Suzanne, C., Tim, G. and Hung, D., 2006. Guide to Integrating Forensic Techniques into Incident Response. *NIST Special Publication*.
17. Khawla, A., Jones, A. & Martin, T (2012). Forensics data acquisition methods for mobile phones. 2012 International Conference for Internet Technology and Secured Transactions, ICITST 2012. 265-269.
18. Kothari, C. R. (2004). Research methodology methods and techniques (2nd ed.). New Age International Publishers.
19. Lillis, D., Becker, B. A., O'Sullivan, T. & Scanlon, M. (2016). Current Challenges and Future Research Areas for Digital Forensic Investigation. In Annual ADFSL

Conference on Digital Forensics, Security and Law (Vol. 6). Retrieved from https://commons. erau.edu/adfsl/2016/tuesday/6

20. Mazzetti, M., Tavernise, S. & Healy, J. (2010, May 4). Suspect, charged, said to admit to role in plot. Retrieved November 9, 2018, from https://www.nytimes.com/2010/05/05/nyregion/05bomb.html
21. Moshe, Y. (2015). [online] Intersecmag.co.uk. Available at: http://www.intersecmag.co.uk/wp-content/uploads/2015/07/Forensics-July15.pdf [Accessed 2 Jan. 2020].
22. Reactjs.org. 2020. *React – A Javascript Library For Building User Interfaces*. [online] Available at: <https://reactjs.org/> [Accessed 20 May 2020].
23. Roussev, V., 2015. Building a Forensic Computing Language. *2015 48th Hawaii International Conference on System Sciences*, pp.5228-5233.
24. Sammons, J. (2015). Digital forensics: threatscape and best practices. 1st ed. Syngress, pp.45-58.
25. Skulkin, O., Tindall, D. & Tamma, R. (2018). Learning Android Forensics (2nd ed.). Packt Publishing Ltd.
26. Sarwar Mir, S., Shoaib, U. and Shahzad Sarfraz, M., 2016. Analysis of Digital Forensic Investigation Models. *International Journal of Computer Science and Information Security*
27. Spalevic, Z., Bjelajac, Z. and Caric, M. (2012). The importance and the role of forensics of mobile. Facta universitatis - series: Electronics and Energetics, 25(2), pp.121-136.
28. Stelly, C. and Roussev, V., 2018. Nugget: A digital forensics language. *Digital Investigation*, 24, pp.S38-S47.
29. Svein Y. (2005). Forensic analysis of mobile phone internal memory, in Advances in Digital Forensics Springer, ch. 16, pp. 191-204.
30. The MITRE Corporation. (2016). CybOX Schemas and Schema Development. Retrieved December 30, 2018, from https://github.com/CybOXProject/schemas
31. The Volatility Foundation. (2018). Volatility. Retrieved November 3, 2018, from https:// www.volatilityfoundation.org/
32. Toroitich, D. (2020). Cyber-crime on the rise: Survey: The Standard. [online] The Standard. Available at: https://www.standardmedia.co.ke/business/article/2001346601/cyber-crime-on-the-rise-survey [Accessed 3 Jan. 2020].

Appendix A: Sample DFXML output

A.1 SMS and MMS DXML output

Listing A.1 SMS and MMS DXML

```
<?xml version="1.0" encoding="UTF-8"?>
<dfxml xmlns="http://www.forensicswiki.org/wiki/Category:Digital_Forensics_XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:mobile="https://github.com/ngash/dfxml/mobile" version="1.2.0">
  <metadata>
  </metadata>
  <creator>
    <program>nugget-tools</program>
    <version>0.0.1</version>
    <execution_environment>
      <os_sysname>Darwin</os_sysname>
      <os_release>19.4.0</os_release>
      <os_version>Darwin Kernel Version 19.4.0: Wed Mar  4 22:28:40 PST 2020;
root:xnu-6153.101.6~15/RELEASE_X86_64</os_version>
      <host>CDans-MacBook-Pro.local</host>
      <arch>x86_64</arch>
      <uid>501</uid>
      <username>cdan</username>
      <start_time>2020-07-21T20:45:04.049512+03:00</start_time>
      <mobile:command_line mobile:sequence="1">/usr/bin/tar -t -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar</mobile:command_line>
      <mobile:command_line mobile:sequence="2">/usr/bin/tar -x -C
/Users/cdan/projects/cdan/nugget-monorepo/nugget-tools/stuff/workspace -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar
./private/var/mobile/Library/SMS/sms.db</mobile:command_line>
    </execution_environment>
  </creator>
  <rusage>
    <utime>0.000484707</utime>
    <stime>0.000566904</stime>
    <maxrss>49393664</maxrss>
    <minflt>22658</minflt>
    <majflt>12</majflt>
    <nswap>0</nswap>
    <inblock>0</inblock>
    <oublock>0</oublock>
    <clocktime>0.798237182</clocktime>
  </rusage>
  <mobile:sms_mms>
    <mobile:kind>sms</mobile:kind>
    <mobile:sender>+15713083236</mobile:sender>
```

```
    <mobile:body>What are you up to this weekend? </mobile:body>
    <mobile:date_received>2012-06-13T00:25:04+03:00</mobile:date_received>
    <mobile:read>1</mobile:read>
    <mobile:country_code>us</mobile:country_code>
    <mobile:type>inbox</mobile:type>
  </mobile:sms_mms>
  <mobile:sms_mms>
    <mobile:kind>sms</mobile:kind>
    <mobile:sender>+17038296071</mobile:sender>
    <mobile:body>I&#39;m going out with dad after school for pizza! Thought I&#39;d let
you know if you planned to cook. &#xA;T</mobile:body>
    <mobile:date_received>2012-06-13T20:30:28+03:00</mobile:date_received>
    <mobile:read>1</mobile:read>
    <mobile:country_code>us</mobile:country_code>
    <mobile:type>inbox</mobile:type>
  </mobile:sms_mms>
  <mobile:sms_mms>
    <mobile:kind>sms</mobile:kind>
    <mobile:recipient>+17038296071</mobile:recipient>
    <mobile:body>Hey honey. I&#39;m not sure if we can afford Prufrock anymore... What
do you think about maybe switching to someplace else?</mobile:body>
    <mobile:date_sent>2012-07-03T16:41:51+03:00</mobile:date_sent>
    <mobile:read>1</mobile:read>
    <mobile:country_code>us</mobile:country_code>
    <mobile:type>sent</mobile:type>
  </mobile:sms_mms>
</dfxml>
```

A.2 Call DFXML output

Listing A.2 Call DFXML output

```
<?xml version="1.0" encoding="UTF-8"?>
<dfxml xmlns="http://www.forensicswiki.org/wiki/Category:Digital_Forensics_XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:mobile="https://github.com/ngash/dfxml/mobile" version="1.2.0">
  <metadata>
  </metadata>
  <creator>
    <program>nugget-tools</program>
    <version>0.0.1</version>
    <execution_environment>
      <os_sysname>Darwin</os_sysname>
      <os_release>19.4.0</os_release>
      <os_version>Darwin Kernel Version 19.4.0: Wed Mar  4 22:28:40 PST 2020;
root:xnu-6153.101.6~15/RELEASE_X86_64</os_version>
      <host>CDans-MacBook-Pro.local</host>
      <arch>x86_64</arch>
```

```
      <uid>501</uid>
      <username>cdan</username>
      <start_time>2020-07-21T20:39:56.540995+03:00</start_time>
      <mobile:command_line mobile:sequence="1">/usr/bin/tar -t -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar</mobile:command_line>
      <mobile:command_line mobile:sequence="2">/usr/bin/tar -x -C
/Users/cdan/projects/cdan/nugget-monorepo/nugget-tools/stuff/workspace -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar
./private/var/wireless/Library/CallHistory/call_history.db</mobile:command_line>
    </execution_environment>
  </creator>
  <rusage>
    <utime>0.000167268</utime>
    <stime>0.000172257</stime>
    <maxrss>41918464</maxrss>
    <minflt>11395</minflt>
    <majflt>12</majflt>
    <nswap>0</nswap>
    <inblock>0</inblock>
    <oublock>0</oublock>
    <clocktime>0.896029496</clocktime>
  </rusage>
  <mobile:call>
    <mobile:from>6508870260</mobile:from>
    <mobile:date_called>2012-06-12T23:04:50+03:00</mobile:date_called>
    <mobile:duration>20</mobile:duration>
    <mobile:country_code>310</mobile:country_code>
    <mobile:type>incoming</mobile:type>
  </mobile:call>
  <mobile:call>
    <mobile:to>+15713083236</mobile:to>
    <mobile:date_called>2012-06-13T19:29:13+03:00</mobile:date_called>
    <mobile:country_code>310</mobile:country_code>
    <mobile:type>outgoing</mobile:type>
  </mobile:call>
  <mobile:call>
    <mobile:from>5712458517</mobile:from>
    <mobile:date_called>2012-06-22T20:34:26+03:00</mobile:date_called>
    <mobile:country_code>310</mobile:country_code>
    <mobile:type>incoming</mobile:type>
  </mobile:call>
  <mobile:call>
    <mobile:from>5713083236</mobile:from>
    <mobile:date_called>2012-07-06T18:18:50+03:00</mobile:date_called>
    <mobile:duration>244</mobile:duration>
    <mobile:country_code>310</mobile:country_code>
    <mobile:type>incoming</mobile:type>
  </mobile:call>
</dfxml>
```

A.3 Location DFXML output

Listing A.3 Location DFML output

```
<?xml version="1.0" encoding="UTF-8"?>
<dfxml xmlns="http://www.forensicswiki.org/wiki/Category:Digital_Forensics_XML"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:mobile="https://github.com/ngash/dfxml/mobile" version="1.2.0">
  <metadata>
  </metadata>
  <creator>
    <program>nugget-tools</program>
    <version>0.0.1</version>
    <execution_environment>
      <os_sysname>Darwin</os_sysname>
      <os_release>19.4.0</os_release>
      <os_version>Darwin Kernel Version 19.4.0: Wed Mar  4 22:28:40 PST 2020;
root:xnu-6153.101.6~15/RELEASE_X86_64</os_version>
      <host>CDans-MacBook-Pro.local</host>
      <arch>x86_64</arch>
      <uid>501</uid>
      <username>cdan</username>
      <start_time>2020-07-21T21:00:29.112556+03:00</start_time>
      <mobile:command_line mobile:sequence="1">/usr/bin/tar -t -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar</mobile:command_line>
      <mobile:command_line mobile:sequence="2">/usr/bin/tar -x -C
/Users/cdan/projects/cdan/nugget-monorepo/nugget-tools/stuff/workspace -f
/Users/cdan/Desktop/tracy-phone-2012-07-15-final.tar
./private/var/root/Library/Caches/locationd/consolidated.db</mobile:command_line>
    </execution_environment>
  </creator>
  <rusage>
    <utime>1.000675186</utime>
    <stime>1.00098375</stime>
    <maxrss>68505600</maxrss>
    <minflt>73876</minflt>
    <majflt>12</majflt>
    <nswap>0</nswap>
    <inblock>0</inblock>
    <oublock>0</oublock>
    <clocktime>0.830211845</clocktime>
  </rusage>
  <mobile:location>
    <mobile:long>-77.11546951</mobile:long>
    <mobile:lat>38.87767624</mobile:lat>
    <mobile:source>
```

```
  </mobile:source>
  <mobile:confidence>70</mobile:confidence>
  <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
  <mobile:cell_mcc>310</mobile:cell_mcc>
  <mobile:cell_mnc>410</mobile:cell_mnc>
  <mobile:cell_lac>7985</mobile:cell_lac>
  <mobile:cell_ci>160043533</mobile:cell_ci>
 </mobile:location>
 <mobile:location>
  <mobile:long>-77.11447036</mobile:long>
  <mobile:lat>38.8778367</mobile:lat>
  <mobile:source>
  </mobile:source>
  <mobile:confidence>60</mobile:confidence>
  <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
  <mobile:cell_mcc>310</mobile:cell_mcc>
  <mobile:cell_mnc>410</mobile:cell_mnc>
  <mobile:cell_lac>7985</mobile:cell_lac>
  <mobile:cell_ci>158138900</mobile:cell_ci>
 </mobile:location>
 <mobile:location>
  <mobile:long>-77.11500936</mobile:long>
  <mobile:lat>38.87595677</mobile:lat>
  <mobile:source>
  </mobile:source>
  <mobile:confidence>60</mobile:confidence>
  <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
  <mobile:cell_mcc>310</mobile:cell_mcc>
  <mobile:cell_mnc>410</mobile:cell_mnc>
  <mobile:cell_lac>7985</mobile:cell_lac>
  <mobile:cell_ci>160093723</mobile:cell_ci>
 </mobile:location>
 <mobile:location>
  <mobile:long>-77.11703056</mobile:long>
  <mobile:lat>38.87869298</mobile:lat>
  <mobile:source>
  </mobile:source>
  <mobile:confidence>50</mobile:confidence>
  <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
  <mobile:cell_mcc>310</mobile:cell_mcc>
  <mobile:cell_mnc>410</mobile:cell_mnc>
  <mobile:cell_lac>7988</mobile:cell_lac>
  <mobile:cell_ci>193465818</mobile:cell_ci>
 </mobile:location>
 <mobile:location>
  <mobile:long>-77.1184743</mobile:long>
  <mobile:lat>38.87763071</mobile:lat>
  <mobile:source>
  </mobile:source>
  <mobile:confidence>60</mobile:confidence>
```

```
    <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
    <mobile:cell_mcc>310</mobile:cell_mcc>
    <mobile:cell_mnc>410</mobile:cell_mnc>
    <mobile:cell_lac>7985</mobile:cell_lac>
    <mobile:cell_ci>160381399</mobile:cell_ci>
  </mobile:location>
  <mobile:location>
    <mobile:long>-77.11932128</mobile:long>
    <mobile:lat>38.87675511</mobile:lat>
    <mobile:source>
    </mobile:source>
    <mobile:confidence>60</mobile:confidence>
    <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
    <mobile:cell_mcc>310</mobile:cell_mcc>
    <mobile:cell_mnc>410</mobile:cell_mnc>
    <mobile:cell_lac>7985</mobile:cell_lac>
    <mobile:cell_ci>160395633</mobile:cell_ci>
  </mobile:location>
  <mobile:location>
    <mobile:long>-77.1198436</mobile:long>
    <mobile:lat>38.87582403</mobile:lat>
    <mobile:source>
    </mobile:source>
    <mobile:confidence>60</mobile:confidence>
    <mobile:timestamp>2012-06-13T22:01:21+03:00</mobile:timestamp>
    <mobile:cell_mcc>310</mobile:cell_mcc>
    <mobile:cell_mnc>410</mobile:cell_mnc>
    <mobile:cell_lac>7985</mobile:cell_lac>
    </mobile:location>
</dfxml>
```

Appendix B: Sample source code listing

B.1 Get case location details

Listing B.1. Get case location details

```
def get_case_location(case, date_from=None, date_to=None):
    """Get case location."""
    location_data = Location.objects.filter(
        case=case, active=True).exclude(
        location_locationdetails__address='N/A').order_by('timestamp').distinct('timestamp')
    if date_from and date_to:
        location_list = location_data.filter(
            timestamp__date__gte=date_from, timestamp__date__lte=date_to).values(
            'latlong', 'timestamp', 'id', 'location_locationdetails__country',
```

```
        'location_locationdetails__address', 'location_locationdetails__place',
        'location_locationdetails__locality', 'location_locationdetails__neighborhood',
        'location_locationdetails__region')
  else:
    location_list = location_data.filter(
        case=case).values(
        'latlong', 'timestamp', 'id', 'location_locationdetails__country',
        'location_locationdetails__address', 'location_locationdetails__place',
        'location_locationdetails__locality', 'location_locationdetails__neighborhood',
        'location_locationdetails__region')
  qualified_location = [
    {
        'lat': q['latlong'].coords[1],
        'lng': q['latlong'].coords[0],
        'id': q['id'],
        'timestamp':q['timestamp'],
        'place': q['location_locationdetails__place'],
        'address': q['location_locationdetails__address'],
        'locality': q['location_locationdetails__locality']
    } for q in location_list]
  return qualified_location
```

B.2 Create case

Listing B.2. Create case

```
import React, { Component } from 'react'
import Grid from '@material-ui/core/Grid';
import Paper from '@material-ui/core/Paper';
import AssessmentIcon from '@material-ui/icons/Assessment';

import { withStyles } from '@material-ui/core/styles'
import MultiStep from 'react-multistep'
import { steps } from './steps'
import axios from 'axios';


const styles = theme => ({
  root: {
    flexGrow: 1,
  },
  paper: {
    padding: theme.spacing(2),
    textAlign: 'center',
    color: theme.palette.text.secondary,
  },
  header2: {
```

```
    color: "#90caf9",
    fontSize: "1.2rem",
    fontFamily: "Roboto, Helvetica, Arial, sans-serif",
    textAlign: "left",
    fontWeight: "500",
    marginBottom: "2px",
    paddingLeft: "10px"
  },
  header4: {
    fontSize: "2.125rem",
    textAlign: "left",
    paddingLeft: "10px",
    marginLeft: "30px"
  },
  caseInfo: {
    display: "flex",
    alignItems: "center"
  }
})

class CreateCase extends Component {
  constructor(props) {
    super(props);
    this.state = {
      total_case_stats: {
          total: 0,
          android: 0,
          iphone: 0
      },
    }
  }

  componentDidMount() {
    axios.get(
        `http://127.0.0.1:8000/v1/case/cases/get_all_cases/`, {
          headers: {
            'Content-Type': 'application/json'
          }
        }).then((response) => {
          const case_stats = response.data;

          var total_case_stats = {...this.state.total_case_stats}
          total_case_stats.total = case_stats['case_stats']['total_cases']
          total_case_stats.android = case_stats['case_stats']['android_cases']
          total_case_stats.iphone = case_stats['case_stats']['iphone_cases']

          this.setState(
            {
              total_case_stats: total_case_stats
            })
```

```
      })
  }

  render() {
    const { classes} = this.props

    const createCase = {
       boxShadow: "0px 1px 5px 0px rgba(0,0,0,0.2), 0px 2px 2px 0px rgba(0,0,0,0.14), 0px
3px 1px -2px rgba(0,0,0,0.12)",
       padding: "10px",
       paddingBottom: "15px",
    };
    return (
        <div className={classes.root}>
         <Grid container spacing={2}>
           <Grid item xs={9}>
           <div style={createCase} className="multi-container">
            <MultiStep steps={steps} />
           </div>
           </Grid>
           <Grid item xs={3}>
            <Paper className={classes.paper}>
             <div className={classes.caseInfo}>
              <AssessmentIcon fontSize="small" />
              <span className={classes.header2}> All Cases</span>
              <span className={classes.header4}>{this.state.total_case_stats.total}</span>
             </div>
             <div className={classes.caseInfo}>
              <AssessmentIcon fontSize="small" />
              <span className={classes.header2}> Android</span>
              <span
classNameclasses.header4}>{this.state.total_case_stats.android}</span>
             </div>
             <div className={classes.caseInfo}>
              <AssessmentIcon fontSize="small" />
              <span className={classes.header2}>IOS</span>
              <span
classNameclasses.header4}>{this.state.total_case_stats.iphone}</span>
             </div>
            </Paper>
           </Grid>
         </Grid>
        </div>
    )
  }
}

export default withStyles(styles) (CreateCase)
```