



The author(s) shown below used Federal funding provided by the U.S. Department of Justice to prepare the following resource:

Document Title: Targeted Forensic Data Extraction from Mobile Devices (TFDEMD)

Author(s): Sudhir Aggarwal, Tathagata Mukherjee, Umit Karabiyik, Hong Mei Chi

Document Number: 300697

Date Received: April 2021

Award Number: 2016-MU-CX-K003

This resource has not been published by the U.S. Department of Justice. This resource is being made publically available through the Office of Justice Programs' National Criminal Justice Reference Service.

Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

Final Report
Targeted Forensic Data Extraction from Mobile Devices (TFDEMD)
2016-MU-CX-K003

Sudhir Aggarwal, (PI)
Tathagata Mukherjee (co-PI), Umit Karabiyik (co-PI), Hong Mei Chi (co-PI)

E-Crime Investigative Technologies (ECIT)
Florida State University
Tallahassee, FL 32306

Period: January 1, 2017- December 31, 2019

1. Introduction

In this final report we describe three software systems that we were completed by 2019 year end: (1) the prototype system for targeted (selective) data extract (**TDES Version 3.5**) for iOS and Android Smartphones that was our focus based on the initial solicitation and work primarily during the first two years of the project; (2) a variation of the TDES system that is designed to support law enforcement in getting relevant data subsequent to a mass incident (**TDES-MI Version 1.0**) and was proposed for the third year (as extension work); and (3) An iOS Schema Evolution Analysis System (**SEAS Version 1.0**) that was also developed in the third year of the project. It was designed to support 3rd party iOS developers in understanding changes in the iOS database schemas for native applications and to assist in needed code changes when Apple updates / upgrades the iOS versions.

Before describing our research, prototypes and deliverables under this project, we would like to acknowledge an exceptional set of students, across our several universities who have worked on this project. They spent a great deal of time supporting our team in designing, implementing, testing and documenting the work of this project and are all to be commended. The students are:

- Florida State University – Gokila Dorai (PhD received Fall 2019), Manuel Hernandez Romero (M.S. Fall 2018), James Parsons (M.S. Spring 2018), Samiha Shimmi (M.S. Summer 2019), Kobra Amiri (PhD candidate), Reaz Masud (M.S. candidate), Juan Pablo Conde Curuchet (M.S. candidate), Aaron Pinto (M.S. candidate);
- Sam Houston State University – Nicholas Guerra (B.S. Spring 2018), Kushboo Rathi (M.S. Spring 2018);
- Florida A&M University – Rodney Wilson (M.S. Summer 2018), Temilola Ageribigbi (M.S. Summer 2018).

PI and Co-PI Staffing: Dr. Sudhir Aggarwal (Florida State University) worked on the project all three years, as did Dr. Umit Karabiyik (Purdue University, previously at Houston State University) and Dr. Tathagata Mukherjee (University of Alabama Huntsville, previously at FSU). Dr. Hongmei Chi (Florida A&M University) worked on the project for the first two years.

In Section 2 we describe our Targeted Data Extraction System Version 3.5. Our work was presented in *Proceedings 15th IFIP WG11.9 Int. Conf. on Digital Forensics*, Jan 2019:

- S. Aggarwal, G, Dorai, U. Karabiyik, T. Mukherjee, N. Guerra, M. Hernandez, J. Parsons, K. Rathi, H. Chi, T. Aderibigbe, R. Wilson, “Design and Implementation of a Targeted Data Extraction System for Mobile Devices.”
- We refer to this paper in Section 2 and it is included as a separate .pdf file titled *TDES-DigitalForensics 2019.pdf*. It is an integral part of this report.
- The paper has been published in as a chapter in Advances in Digital Forensics XV published by Springer. The full citation is:
Aggarwal S. et al. (2019) A Targeted Data Extraction System for Mobile Devices. In: Peterson G., Sheno S. (eds.) Advances in Digital Forensics XV. IFIP Advances in Information and Communication Technology, vol. 569.

In Section 3 we describe a variation of the TDES system that was designed and developed in the third year, called TDES-mi Version 1.0. This work focused on a mass incident situation in which law enforcement needs to swiftly analyze many phones (thousands) collected after a mass incident or mass-casualty incident. In addition to uploading selected data from bystander smartphones, the TDES-mi system also supports a backend analysis system for teams of law enforcement personnel.

In Section 4 we describe another system that was developed in the third year of the project. The Schema Evolution Analysis system (SEAS Version 1.0) was developed due to our own needs for modifying parts of our system code based on upgrades or version changes to iOS. Whenever Apple launches an update/upgrade to an iOS version, the database schema of native applications could be changed. Third-party developers typically have very little knowledge about what is changed in database schemas in an iPhone backup. We faced this problem ourselves when we needed to update our TDES code during an iOS upgrade. SEAS helps the developers and code maintainers deal with this issue. A research paper on this work has recently submitted to a conference. This is also a separate file called **SEAS.pdf**.

A patent on the TDES 3.5 work has been filed through the FSU office of Commercialization. For more information you can contact FSU. The basic information on this is:

- EFS ID: 38214491
- Application Number: 16735092
- Title of Invention: TARGETED DATA EXTRACTION SYSTEM AND METHOD
- First Named Inventor: Sudhir Aggarwal
- Filer: Mark R. Deluca
- Attorney Docket Number: 6624-06401
- Receipt Date: 06-JAN-2020

Deliverables

This report also includes user manuals for TDES (Android and iOS) and TDES-MI at the end of this report. Note that the full code is **too large** (about 500 MB) to send as an attachment. It can be obtained from the PI on a memory stick. A bootable hard drive of the working TDES system can be sent by the PI as a hard drive. The full code could also be uploaded from a server upon request. In each sections of our work we discuss some aspects of the deliverable code.

2. Targeted Data Extraction System Version 3.5

In this section we assume that the reader has read the published paper *TDES-DigitalForensics 2019.pdf*. We view this paper as an integral part of the current report. We first present an overview of our work describing the major capabilities we have implemented. Additional detail is available in the paper. The basic goal of the project was to build a system that could be used by a law enforcement investigation to do selective data extraction from iOS and Android smartphones. The goal was to do this quickly, effectively and with proper chain of custody support. Additionally we designed straight forward user interfaces and support for legal consent forms.

In this section we also discuss a useful addition to the content filtering capabilities of TDES. We have added the ability to select messages based on keyword search. This is a natural feature that can be useful when investigators need to analyze a large number of messages. It can be applicable to both TDES and TDES-MI. This work is not in the published paper. We have completed the keyword search and integrated it into TDES as part of our extension work in year 3. We were only able to complete this for the Android smartphone version and not for iOS. The work on keyword search is described in Section 2.3. Our deliverables for TDES, which are also similar in mechanism for TDES-MI and SEAS also, are discussed in Section 2.7.

2.1 Metadata Based Filtering

Metadata filtering refers to filtering with information about data categories that is both saved by the OS as well as possibly accessible through OS APIs. Column 1 of Figure 1 shows these categories of metadata.

Our system extracts categories in Box 1 (Photos to Reminders) for both iPhones and Android Phones using *on-device* extraction where our downloaded Apps first extract the data and then move the relevant data to the TDES Manager. This way only the desired data is removed from the phone (note that content filtering is also done on this data by the Apps which additionally reduces it). Thus on-device is preferable for targeted data extraction.

However, if we look at the Box 2 categories (Third party Photos to Call Logs) for iPhones we are not able to extract these categories on-device. We can do so for Android phones and we in fact do that. However, for iPhones, we were able to implement and use an iTunes backup and do off-device extraction using the TDES Manager. If there are any such categories of data that need to be extracted from an iPhone, we first create an iTunes backup in the TDES Manager environment. The TDES Manager extracts the relevant data and then deletes the backup. Note that we only create a backup if there is a request for a data category in Box 2.

Finally, there does not appear to be any way to extract categories of data in Box 3 from Android phones. However for iPhones it is possible to do some of this using the iTunes backup for iPhones. Extraction of Box 3 categories for iPhones would need to be done on a case by case basis since it depends heavily on whether the data from that application (for example Viber Messages) is encrypted, understanding the details of the storage scheme on the iTunes backup, etc. Use of iTunes backup for such

applications has been explored in work by others and we decided not to implement this only for iPhones as it would mainly be redoing work of others and would also be time consuming. Thus we did not focus on any of the categories in Box 3 for either iPhones or Android based phones. All the metadata based filtering possible for both TDES systems for categories in Boxes 1 and 2 however have been implemented and tested.

Category of Data	Metadata Type	iOS On-Device	iOS Off-Device	Android
Photos	Date & Time	Yes		Yes
Photos	Location	Yes		Yes
Photos	Album Type	Yes		Yes
Videos	Date & Time	Yes		Yes
Videos	Location	Yes		Yes
Contacts	Name	Yes		Yes
Contacts	Number	Yes		Yes
Contacts	Area Code	Yes		Yes
Contacts	Email	Yes		Yes
Calendar Events	Date	Yes		Yes
Reminders	Date	Yes		Yes
Photos	Origin - Third Party Apps	No	Yes	Yes
Messages/SMS/MMS	Date Time	No	Yes	Yes
Messages/SMS/MMS	Contact Number	No	Yes	Yes
Call Logs	Incoming Calls	No	Yes	Yes
Call Logs	Outgoing Calls	No	Yes	Yes
Call Logs	Missed Calls	No	Yes	Yes
Call Logs	Date Time	No	Yes	Yes
Notes	Search String	No	*	No
Notes	Date & Time	No	*	No
Voice Memos	Date & Time	No	*	No
Web History	Date & Time	No	*	No
Emails	Date & Time	No	*	No
Facebook Messages	Date & Time	No	*	No
Whatsapp Messages	Date & Time	No	*	No
LinkedIn Messages	Date & Time	No	*	No
WeChat Messages	Date & Time	No	*	No
Viber Messages	Date & Time	No	*	No

Figure 1: Metadata Based Extraction

2.2 Content Based Filtering

We explored using many types of ML frameworks to do our content based filtering. We ultimately chose to use TensorFlow (developed by Google) and Caffe (Berkeley AI Research) for our project. We focused on content filtering for Photos although the same approaches could be used for Videos. Trained *models* are developed through many ML techniques, including deep learning using neural nets. Models can be used that have been developed by others and are open source. For our project, we used the Inception v3 Model for many of the object classifications and OpenNSFW for skin exposure. A model can be viewed as a data structure that can be imported for use within a framework which is then combined (in our case) with the TDES mobile app. In order to make a prediction about an image, the app runs all the inference computations locally on the device - on its own CPU or GPU.

Figure 2 below shows the content based image filtering options we chose to implement in our prototypes. We arrived at the specific choices after discussions with law enforcement. These are only proof-of-concept choices and could naturally be

expanded in future work. Of course, for some new choices it may be necessary to do a significant amount of additional work. For example, if we wanted to have as choice “graffiti” it would likely necessitate first developing a large data set of such images and then training on these images. Often getting a sufficiently large data set to be useful is not easy.

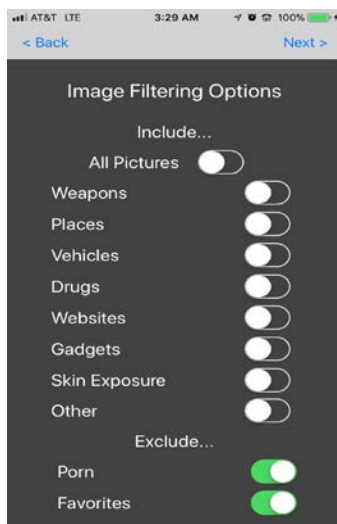


Figure 2: Image Content Filtering Options

2.2.1 Image Filtering Approach and Options

The image filtering screen allows one to choose image categories that one desires to *include* (Weapons, Places, etc.) or *exclude* (Porn, Favorites). “All Pictures” includes all images without content filtering. “Other” is a special category that is used if one wants to return other choices from the Inception model besides the ones indicated. The “Favorites” choice has not been implemented and is planned to be a set of favorites of the user that could be excluded (e.g. family members). Note that porn *exclusion* makes use of the NSFW model and the *include* choices use the Inception model.

2.2.2 Image Content Filtering Adaptation from Original Models

A model such as Inception will analyze an image and then return the probability values of all the *labels* (specific choices) in the image that it has been trained to recognize. Inception returns values for 1000 different labels. In order to determine whether or not to return an image for one of our *include* filtering choices (we use *Weapon* as a continuing example) we needed to develop a heuristic that decides what images should be put into our various categories based on the Inception data of probabilities of the labels for each candidate image. See Figure 3 for an example image and Figure 4 for the corresponding top 10 label probability values returned by the Inception model.



Figure 3: Image of man with rifle

Rifle (WEAPON)	0.613122940063477
assault rifle, assault gun (WEAPON)	0.255587637424469
revolver, six-gun, six-shooter (WEAPON)	0.0237733535468578
power drill (OTHER)	0.00991020817309618
Tripod (OTHER)	0.00468125101178885
chain saw, chainsaw (WEAPON)	0.00221126712858677
harmonica, mouth organ, harp, mouth harp (OTHER)	0.00172213651239872
horizontal bar, high bar (OTHER)	0.00172213651239872
violin, fiddle (OTHER)	0.00134120113216341
Screwdriver (OTHER)	0.00104452844243497

Figure 4: Top 10 Inception labels and values

The mechanism to determine our image categories is thus a heuristic we build above the Inception model. First we categorize the 1000 Inception labels into the 8 categories of the include choices of Figure 2. Note that our *Weapon* category includes the Inception label Rifle and the Inception label (revolver, six-gun, six-shooter) where all the last three terms reflect a single Inception label. Thus, we first aggregate the 1000 labels into 8 categories.

Next, there are still many ways we could recognize whether a *Weapon* is in the image based on the probabilities of the original labels. We experimented with many heuristics before finding one that we felt best balanced returning images to be expected for each of our categories while not returning undesired images.

Our Heuristic: For each of our categories, find the *label* with the highest probability. Classify the image as in all categories for which the highest probability label has a probability value that is greater than or equal to 10%.

For the category *Weapon*, the label Rifle has the highest probability label value at 61%. For the category *Other*, the label power drill has the highest probability at 1% (after rounding up). For the other categories none of the labels even appear in the top 10 and are certainly below 0.1%. Thus this image would be returned only if the *Weapon* category was chosen. Note that if a percentage threshold of 1% had been used, then this image will be classified as a *Weapon* and also as *Other*. If only category *Other* were indicated in Figure 2, then the image would have still been returned.

With respect to the *exclude* porn filter, if it is checked, all images will be evaluated by the NSFW model and any that have an NSFW score > 0.85 are *not* returned.

2.3 Content Based Filtering for Messages

In this subsection we explore retrieving messages based on content information of the message. Textual evidence is important for a variety of investigations. Our initial focus is on text messages stored on smartphones. We restrict ourselves to the native capabilities of text storage on smartphones as “messages” and do not investigate 3rd party apps storing text. We believe our approach could be equally applicable to such 3rd party apps but it would require specific access techniques to be developed for each such third party app.

2.3.1 Keyword Search

Text messages search used in digital forensic can be used to find hits relevant to the investigator’s objectives. In most forensic tools the main capability implemented is keyword search where each message is analyzed to determine if it contains a special keyword or phrase such as “gun” or “I will kill you.” Although many keyword search algorithms have been implemented for general text search, our goal is to insure that the keyword search techniques that we implement leave no residual trace on the smartphones and additionally do not use any permanent storage on the phones either. To implement text string search, there are typically two approaches: simple matching and indexing. There are various tools based on these two approaches such as Grep, Lucene, and FTS. Among these tools, we have selected FTS3 and FTS4 to implement exact keyword matching on the phone.

In text search, the goal is searching for desired keywords in a message or document in order to find the most relevant matches. To do the search, two approaches have been used in the literature: simple matching and indexing. In simple matching, each of the text messages is scanned one by one to find a match in the text message. In the indexing approach, all of the words in the text messages are extracted and an inverted index is created based on these words. During the search phase, the text messages that contain the “best matching” set of desired keywords would be returned in order. This would be done by using the inverted index approach. We first discuss a few tools that have been developed for keyword search and then discuss our approach:

- Grep: This tool was developed in the early 1970s by Ken Thompson. Grep performs pattern matching based on regular expressions. A variety of Grep implementation are available in different operating systems such as Linux and Windows.
- Lucene: Apache Lucene is a high performance full featured text search engine library written entirely in Java. It is one of the Jakarta projects of the Apache Software Foundation. Lucene is able to achieve fast search responses because, instead of searching the text directly, it uses the index search technique. The basic algorithm implemented by the library uses a set of documents (the messages) and supports defined sets of fields in the document to create the index. Lucene has a query language that allows the user to specify which field

to search on, which fields to give more weights to (boosting) and to perform Boolean queries (based on AND, OR, and NOT).

- SQLite FTS: SQLite developed a set of extension modules for full text search (FTS). The FTS3 and FTS4 extension modules allow users to create special tables with built-in full-text indices. The full-text index allows the user to efficiently query the database for all rows that contain one or more words, even if the table contains many large documents.

2.3.2 Implemented Approach for Keyword Search

We have added the ability of keyword search to TDES so that when the user types some desired keywords in our user interface, the text messages containing at least one of the keywords be visible for the user. Although our search is for specific strings in text, we do not implement regular expression search as implemented in Grep. We felt that typical users would not wish to try defining a regular expression. Our search implementation does not first produce an inverted index (as Lucene) which can be memory consuming for the application. It simply checks each individual text message and returns the ones containing requested keywords.

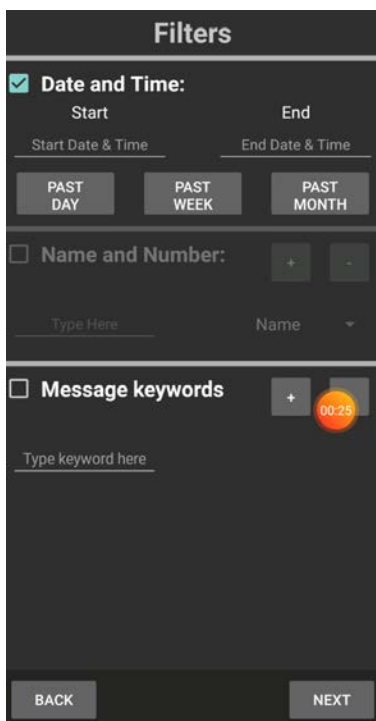


Figure 5: User Interface for Keyword Filtering

As shown in Figure 5, the search capability has been added to the user interface. The user types a keyword in the indicated space; for searching for more or fewer keywords, the add or remove button can be used. The message contact's name and set of text messages containing at least one of the keywords would be shown. An example is shown in Figure 6 below where the set of text messages that contain the keyword "just" is retrieved which are categorized based on name and number of the

contact. By clicking on a listed text message, the actual contents of the message are displayed.

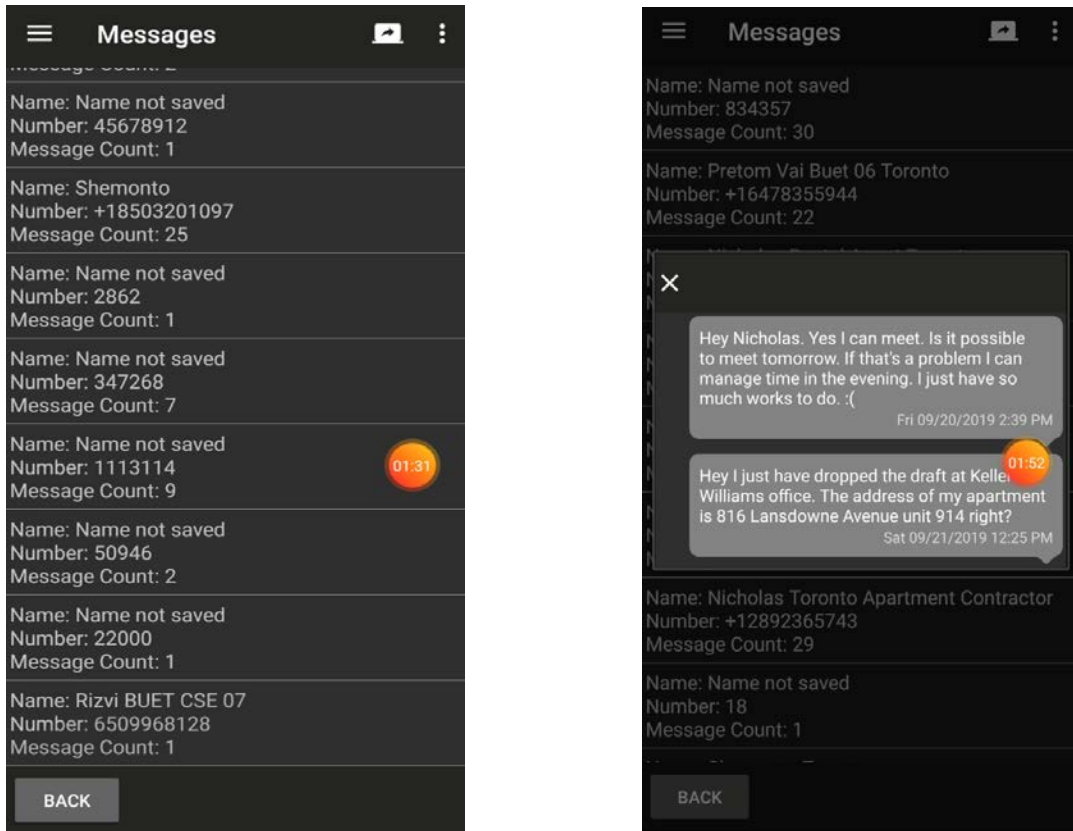


Figure 6: Messaged Retrieved with Details

Note that we have implemented exact keyword matching.

2.4 TDES Manager–App Communication Model

In order to extract data from the smartphones we chose to download an app to the smartphone. We briefly discuss this communication paradigm. See Figure 7.



Figure 7: TDES Communications Paradigm

2.4.1 Connections for Data Transfer

We implemented both wired and wireless connections for iOS and Android phones. A wired connection is faster and this is preferably used for the data transfer. For downloading the apps to the phones we use a wired connection as we need to do this for iOS in any case. For iOS smartphones, it is necessary to set up a personal hotspot also for authentication and we assume a law enforcement phone is available for setting up this hotspot. In most cases, using the evidence device as a personal hotspot is not advised since it might need additional consent from the owner of the device.

2.4.2 TDES Manager

The TDES Manager can be loaded onto many portable devices. Initially we used a USB stick which is quite convenient but it seems to degrade in speed with constant use and transfers then take more time, so we experimented with other device options. We moved to using a portable SSD (solid state device) as our preferred connection device. Of course, any device can still be used as our system only needs to have sufficient memory and speed for our purposes.

The TDES Manager is the initial program that is started when using our TDES system and it interfaces with the investigator. The TDES Manager supports both iPhones and Android based phones. The reporting tool is integrated as part of the TDES Manager and the report that is generated looks exactly the same for both iPhone and Android based phones.

Note that with any upgrade there may be changes that need to be done, particularly for iOS phones. For example, after the update to iOS 11.3, we had encountered a few inconsistencies with respect to the iOS App installation using Cydia Impactor. We had to separate out the process of App Certificate Revocation and App installation to resolve this problem. To keep the effort of inputting user credentials (in order to sign the iOS app that is downloaded) at a minimal level, we also use a script called AutoHotKey. This script will automate the process of revoking a license and re-signing the IPA file on the connected iOS device. The script assumes there is only one iOS device (iDevice) connected. This script is written into a file with extension ".ahk." Double-clicking the AHK file (or .exe if you decide to compile it) should revoke a license and resign the IPA on the connected device. The "username" used in this script is the Apple ID we had used in order to sign the iOS app IPA and "password" is the associated password. Simply double clicking the AHK file installs the iOS app on the iOS device.

2.5 User Interface

We have implemented somewhat different interfaces for both iOS and Android phones because it is very difficult to have exactly the same interface in any case. However, we have implemented similar functionality for both types.

2.5.1 Bookmarking

We do bookmarking for both iPhone and Android phones in a similar way in that the bookmarking does not require a separate iteration but is done as part of an iteration only. Thus, after a description of what is to be extracted is completed, the user can view the results on the phone and bookmark those items that are actually to be exported. Once the export button is hit, items that are bookmarked are extracted. The actual method of bookmarking is slightly different for the two interfaces but obvious from the interface.

2.5.2 Consent Form

We have implemented two versions of using consent forms associated with the two user interfaces. In iOS, the consent form is primarily associated with the export phase of the data extraction. Once the user has chosen what is to be exported and after bookmarking, a consent form appears that can be signed by both the investigator and the owner of the analyzed phone. This signature is digitally done on the phone. After this, the data can be exported. In the iOS version, we have also implemented having a consent form being generated and signed by the phone owner as part of the TDES Manager. For Android phones, the paradigm is slightly different. The user when starting using the Android app is asked as about the broad categories that are required to be extracted. Once this is done and signed, the actual data extraction definition phase begins. The system ensures that the detailed definitions are consistent with the initial “broad” consent form. Thus if only photos were to be extracted, the user could not subsequently even define any videos to be extracted. At the present time we are providing both an off-line capability of generating the consent form and an on-line capability for Android phones that could optionally check if the actual filtering is a subset of a consent form that is “known” by the system.

2.6 Reporting and Data Integrity

The reporting capability has been integrated smoothly into the TDES Manager so the investigator can use the Manager to then look at reports and files.

For validating the forensic soundness of our data extraction approach, we used the ideas of e-discovery and follow the Electronic Discovery Reference Model (EDRM) [6] to do the appropriate reporting, logging, hashing, etc. In the next section, we briefly review the EDRM model and then show how our TDES system supports this.

2.6.1 The EDRM Model Framework

The EDRM model was created to address the lack of standards and guidelines in the e-discovery field. The framework outlines standards for the recovery and discovery of digital data. Figure 8 shows the overall stages of the model and procedures that should be carried out within each stage.

The EDRM model further defines nine components that relate to the organization and preservation of electronically stored data (ESI). These are:

1. **Information Governance** – Getting your electronic house in order to mitigate risk & expenses should e-discovery become an issue, from initial creation of ESI (electronically stored information) through its final disposition.

2. **Identification** – Locating potential sources of ESI & determining its scope, breadth & depth.
3. **Preservation** – Ensuring that ESI is protected against inappropriate alteration or destruction.
4. **Collection** – Gathering ESI for further use in the e-discovery process (processing, review, etc.).
5. **Processing** – Reducing the volume of ESI and converting it, if necessary, to forms more suitable for review & analysis.
6. **Review** – Evaluating ESI for relevance & privilege.
7. **Analysis** – Evaluating ESI for content & context, including key patterns, topics, people & discussion.
8. **Production** – Delivering ESI to others in appropriate forms & using appropriate delivery mechanisms.
9. **Presentation** – Displaying ESI before audiences (at depositions, hearings, trials, etc.), especially in native & near-native forms, to elicit further information, validate existing facts or positions, or persuade an audience.

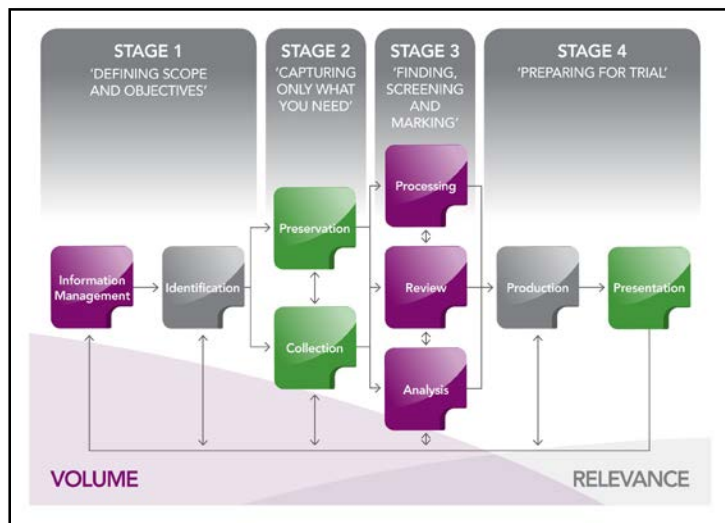


Figure 8: Stages of the EDRM Model

2.6.2 How the TDES System can Support Forensic Soundness

As part of defining scope and objective, an electronic consent form was established to be used with the TDES system. We have also defined a mechanism in the Android version that ensures that we are staying within the scope of the investigation. In the iOS version we only sign the consent form after the types of data to be transferred are defined. As part of capturing only what is needed, the TDES system clearly has this as its primary objective through the ability to describe the data to be extracted. As part of the screening and marking phase, we support bookmarking of the data so that only relevant data is extracted. Furthermore we support hashing to ensure that data is not changed during the export process. Additionally, to develop supporting evidence that our processes do only what they are supposed to do, we have been developing logging capabilities by our system so that an evidentiary trail also exists as to what our system does step-by-step. See Tables 1 and 2 for examples of log reports that we

have designed and in the process of implementing. Finally, as part of preparing for trial, we prepare a detailed report for the investigator about all the data that we have collected.

Table 1: TDES Manager Log Report (optional)**

Event Time	Event
(UTC)	TDES Manager Starts
(UTC)	**iTunes backup starts (based on investigator's selection on the GUI)
(UTC)	**iTunes backup ends (tentative)
(UTC)	TDES App installation starts
(UTC)	TDES App installation ends
(UTC)	Communication channel established
(UTC)	Data transfer from app starts
(UTC)	Data transfer from app ends
(UTC)	Consent form received
(UTC)	Messages/Call Logs filtering criteria received from app
(UTC)	Messages/Call Logs filtering starts
(UTC)	Messages/Call Logs filtering ends
(UTC)	Exit command received
(UTC)	App removed from iOS device
(UTC)	TDES Manager ends (followed by hash computation)

Table 2: TDES iOS App Report

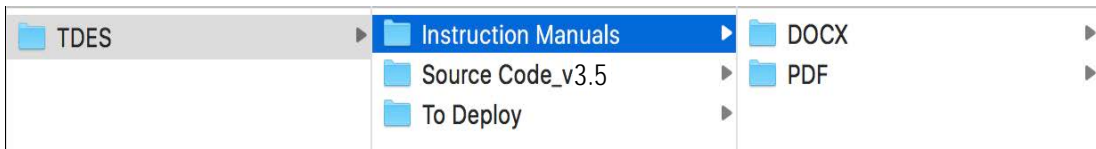
Event Time	Event
(UTC)	TDES App Starts
(UTC)	TDES Manager IP entered ("120.0.0.1")
(UTC)	Iteration-1 starts
(UTC)	Filtering criteria entered
(UTC)	Consent form completed
(UTC)	Filtering starts
(UTC)	** Photos filtering starts
(UTC)	** Videos filtering starts
(UTC)	** Contacts filtering starts
(UTC)	** Calendar filtering starts
(UTC)	** Photos filtering ends (tentative)
(UTC)	** Videos filtering ends (tentative)
(UTC)	** Contacts filtering ends (tentative)
(UTC)	** Calendar filtering ends (tentative)
(UTC)	Filtering ends
(UTC)	Items displayed
(UTC)	Items bookmarked
(UTC)	Data transfer (export) starts
(UTC)	Data transfer (export) ends
(UTC)	Iteration-1 ends
(UTC)	** Iteration-2 begins
(UTC)	[REPEAT]
(UTC)	** Iteration-2 ends
(UTC)	App Exit initiated

2.7 Deliverables: The TDES 3.5 Directory

The full set of all deliverables for this project is simply the TDES Directory. This contains everything related to the complete project: all source code, object code (packages), deployment instructions and manuals, user manuals, etc. The TDES Directory is too large to upload as a file and will can be obtained from the PI via a USB stick or other mechanism. In the next sections we describe the components of the TDES Directory file.

2.7.1 The Main Folder and First level Sub-Directories

The main folder is called TDES. It has three subdirectories: *Source Code*, *To Deploy*, and *Instruction Manuals*. If someone at NIJ wanted to test our system in the final draft phase, we could simply send the directory *To Deploy*. If they also wanted the source code we would additionally send *Source Code*. The initial directories / files are shown below. The instruction manuals for iOS and Android can be found in *Instruction Manuals*. For convenience we attach only the *user* manuals for Android and IOS titled “TDES Manager.”



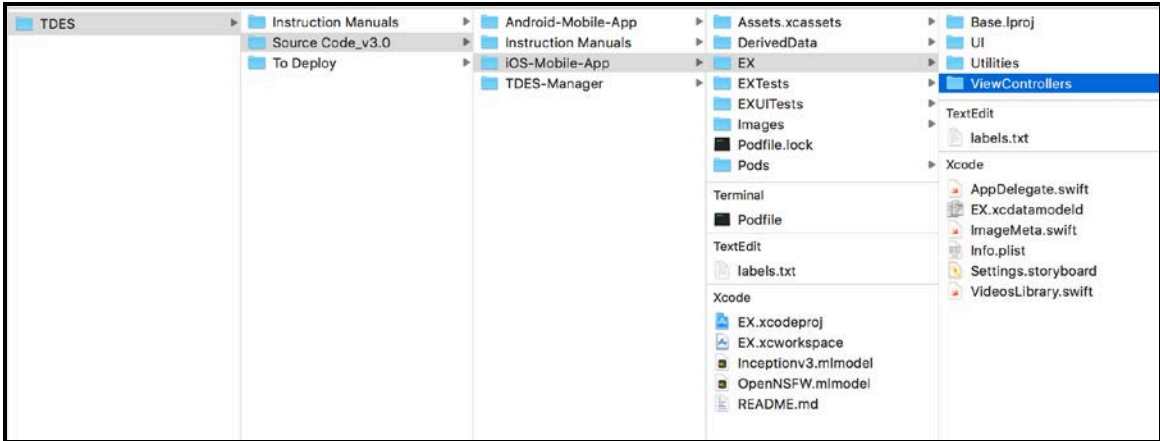
In the subdirectory Instructions Manuals (and also some in the other directories), there are manuals for investigators (users) explaining how to use our system. Various manuals contain information on initial setup of the TDES manager, connecting to the target device and installing the relevant app, running the TDES manager, entering information into the user interface on the app to extract requisite data, getting data back to the manager and using the manager report.

2.7.2 Source Code Sub-Directory

This directory contains all the source code of our system including various instruction and read-me files.

iOS Source Code

The source code files for iOS are in the subdirectory IOS-Mobile-App which itself consists of several directories. Using files in this directory the iOS app to be downloaded to the target device is packaged into an ipa file called EX.ipa. The files in these directories are primarily files in the programming language Swift used by apple developers. See the iOS source code file structure below.

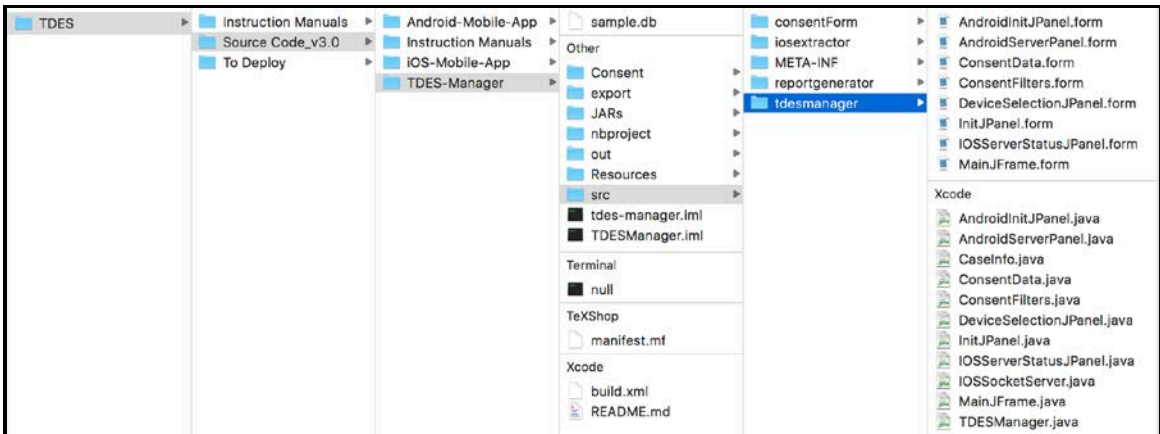


Android Source Code

The Android-Mobile-App subdirectory has the analogous information for the TDES android app as described above for the iOS App. In this case the programming language used is Java and the resulting packaged file is called an .apk file.

TDES Manager Source Code

Recall that the TDES manager is deployed on the Windows operating that is booted up by an investigator from the solid state drive, usb stick or other portable device. The directory structure of the source code for this program is shown below.



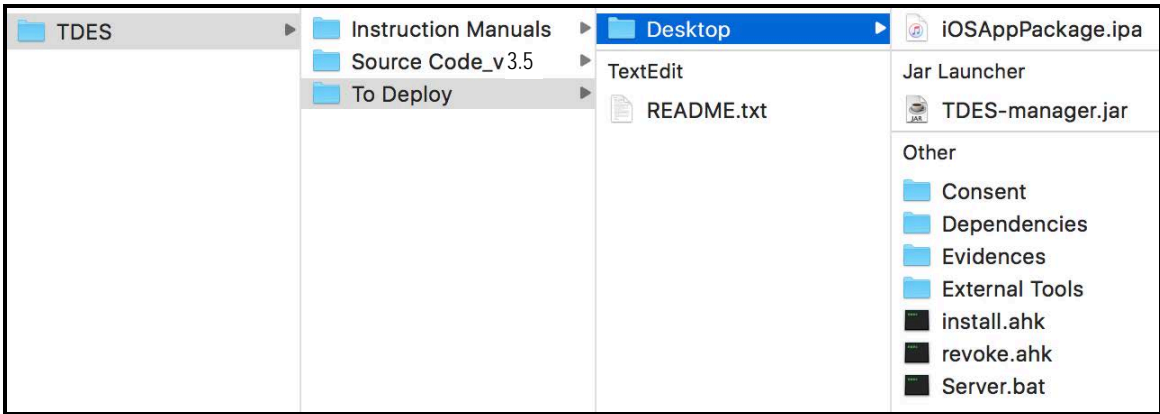
The source code in this directory is used to create the TDES-manager.jar file which must be moved to the portable device.

Compiling Instructions

In the subdirectory Source Code, there is also a directory *Instruction Manuals* which contains instructions on compiling the three major source code systems just mentioned: IOS-Mobile APP, Android-Mobile and TDES-manager. Note that these instructions are different from the user instructions in the first level sub-directories.

2.7.3 To Deploy Sub-Directory

We have essentially automated all the deployment aspects of our project through the files in this folder. See the structure below.



The file *README.txt* explains how to deploy the TDES system. It is designed to be fairly straightforward and can be done by law-enforcement using simple technical skills. The folder *Desktop* must be copied to the “Desktop” of the Windows OS of the bootable drive (SSD, USB stick, etc.). The README explanation file explains the rest of the deployment process.

3. TDES Mass Incident Version 1.0

In this section we discuss the goals achieved for the TDES Mass Incident project. As for the TDES project itself described in section 2, we have separate deliverable as a TDES-MI Directory that will contain all source code, user manuals, deployments instructions, etc.

4.1 Motivation and Initial Considerations

The key motivation behind proposing this subproject is to be able to quickly gather targeted data from a large number of phones during mass incidents such as shootings, bombings, etc. In order to do this, it cannot be done as our current version of TDES operates. Our current version is viewed as designed for an investigator working with a few phones in the field to get the targeted data from these phones with the consent of the phone owner.

In the Mass Incident (MI) case, we have developed an initial model that requires a TDES app to be already on the user's phone. A server in "the cloud" will be used to upload data from multiple phones on demand. At the current time we will view this as a separate version of our TDES system and call it the TDES-mi system for convenience. As previously, we will assume that we have consent of the phone owner to selectively extract data from the phone. In the MI environment, we want to have minimal interaction with the phone owners. Based on the incident, we send the same targeted extraction query to all of the phones in the region of the incident. The query extracts data from the phone and uploads it to the server. The user can see the data to be uploaded and gives consent by permitting the upload. Once we have gathered the data, we provide law enforcement with an analysis system that can integrate the data extracted from the multiple phones. Thus, an integral part of this project will be some analysis options available after selective data extraction. In our initial work we have focused on Android based phones as there is more complexity related to privacy issues when working with iOS based phones.

4.1.1 Mass Incident Model Design

The TDES-mi system has two major components. The first is an adaption of our TDES selective data extraction system, which we term *TDES-extract*. This system can upload images, videos, etc. using a *TDES-mi app* that the user has on their phone obtained from the Google Play Store. The second is an analysis system that uses a large data base gathered from multiple phones. We call this system *TDES-analyze*.

In our previous TDES system, we used the TDES manager on a portable device to work with the TDES app downloaded to the user's phone to extract the relevant data. The user interface on the TDES app was designed to support the investigator and the consenting user to specify the specification of the targeted data to be extracted on the TDES app itself. In the TDES-mi system, this specification is done on the equivalent of the TDES manager side, which is now running on a server in the cloud. Again, we still need consent from the user, and this is explicitly provided by the user when viewing what is to be uploaded. Note that the TDES-mi app has only a minimal user

interface and has no provisions for defining the upload filters. The TDES-mi design components are summarized as:

- An Android application, the TDES-mi app, which can be downloaded from the Google Play Store.
- A back-end server which is part of the TDES-extract system so that data from a TDES-mi app can be securely uploaded to the server.
- An investigator's panel that can be used to define the specified filters and can also run custom analyses, along with helping the investigator organize and classify the data obtained. The component defining the extraction specifications is part of the TDES-extraction system.

4.1.2 Mass Incident System Overview and Interactions

The server application needs to provide necessary API endpoints to support the TDES-mi app. We provide the investigators with a web panel interface that allows specification of the selected data to be requested and later uploaded by the witnesses, along with data processing and an interface for the investigator to work on the data. In the model we have developed, specification of filtering as in our original TDES system can be done, including all the types of metadata extractions and well as content-based extractions. However, we expect that images will be the primary data to be analyzed and we will focus on this type of data in our initial analysis system. To support this analysis, we propose a backend database that will store the data collected at a central site and will provide private access only to investigators through the (private) investigator panel (see Figure 9 for an overview of the components). The system can be either accessed from the private network and the public network. The private network gives the investigators access to the panel, while the public network only gives the app users (witnesses) access to the API endpoints that allow uploading the evidence. For security reasons, the investigator panel is not accessible from the public network. Users who are willing to support investigations of mass incidents will download the TDES-mi app. In our system design, we are proposing that the application would be downloaded from Google Play Store. After downloading the app, the user (witness) will be guided by the app so they will be able to choose the evidence to be uploaded and then upload it.

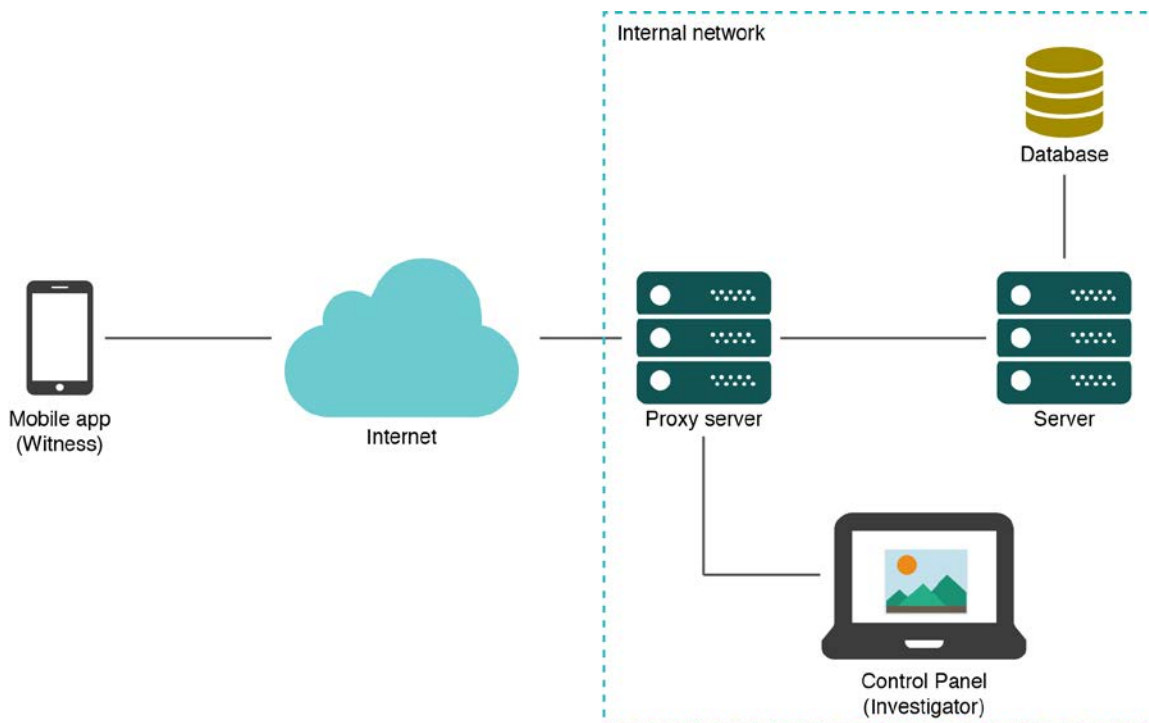


Figure 9: Overview of components

4.1.3 Notifications to the witnesses

We considered two alternatives for this solution: one consisted on having push notifications for the mobile app, and the other required the user to open the app in order to download the data related to the case. We finally decided to use the second approach, since it is more secure in terms of data privacy, as there is no third-party server interacting with the app (for push notifications, a Google server interacts with the app in order to deliver the notification).

Notification upon opening the app: The request for evidence will be shown to the user once they access the app. In this case the user will open the app and it will automatically update the information related to the cases, and in case there is a case related to the location of the user and within the timeframe defined for the incident, it will show the notification and request the user to upload the evidence. This is a more secure approach, since there is no interaction with Google servers but only with our own API backend server.

Note that the filtering options for TDES-mi are currently exactly as we have defined for the Version 3.0 TDES system. In TDES-mi however, the filtering options are only definable by the investigators through their control panel and sent to the TDES-mi app. Subsequently, the user can only see what is to be uploaded and delete any data that should remain private; however, the user is never able to modify the investigators filtering options.

A local database in the backend will keep track of the uploaded data, ensure that the same data is not uploaded twice and keep track of information related to each upload

from a phone. The TDES-extraction system will keep track of the unique device IDs of smartphones, time of update, the related data, etc. In our preliminary design of the system we are planning to *not* capture any other information about the user. For anonymity reasons, we could also not even correlate the uploaded data with a particular phone and delete the device ID once the upload is done.

We next briefly discuss the server-side application that will get the uploaded data and the MySQL database that we use for storage. For the Android TDES-mi on the server side we used Java as the programming language.

4.1.4 Mass Incident Server in the Cloud Application

The server application provides the necessary capability for defining desired filtering information for mobile apps in the location of the mass incident. It also provides the APIs for sending the filtering options to the mobile app and then allowing the app user to upload the resulting (and selected) data. The server also provides the private APIs for the investigators' panel (discussed in the next section) where the investigators will be able to create cases, define the data filters for the requests of evidence and run various types of analysis on the data.

The server application has two main components: the **backend** server that exposes the services (API) both for the investigator panel and the mobile app. On the other hand, there is the **frontend** server, which provides the investigator with the proper interface (web) to navigate through the panel. Details can be seen in Figure 10. Although the frontend and the backend server appear as separate servers, they can be installed on the same physical server for convenience.

For developing the server application, *Java Spring Boot* framework was used, backed by a MySQL database. This is a proven technology stack that guarantees both security and scalability. We also set up a separate FTP server as well for storing and accessing the data retrieved from the smartphones.

Note that the server application is responsible for providing two sets of APIs – an internal set within the cloud for the investigators analyses and an external set to interface with the mobile app. The internal APIs will not be exposed as public and cannot be triggered from outside of our private network. For setting up this environment we suggest a variety of technologies available that we are in the process of implementing. For example, an *NginX* server can be used for setting up routing configurations.

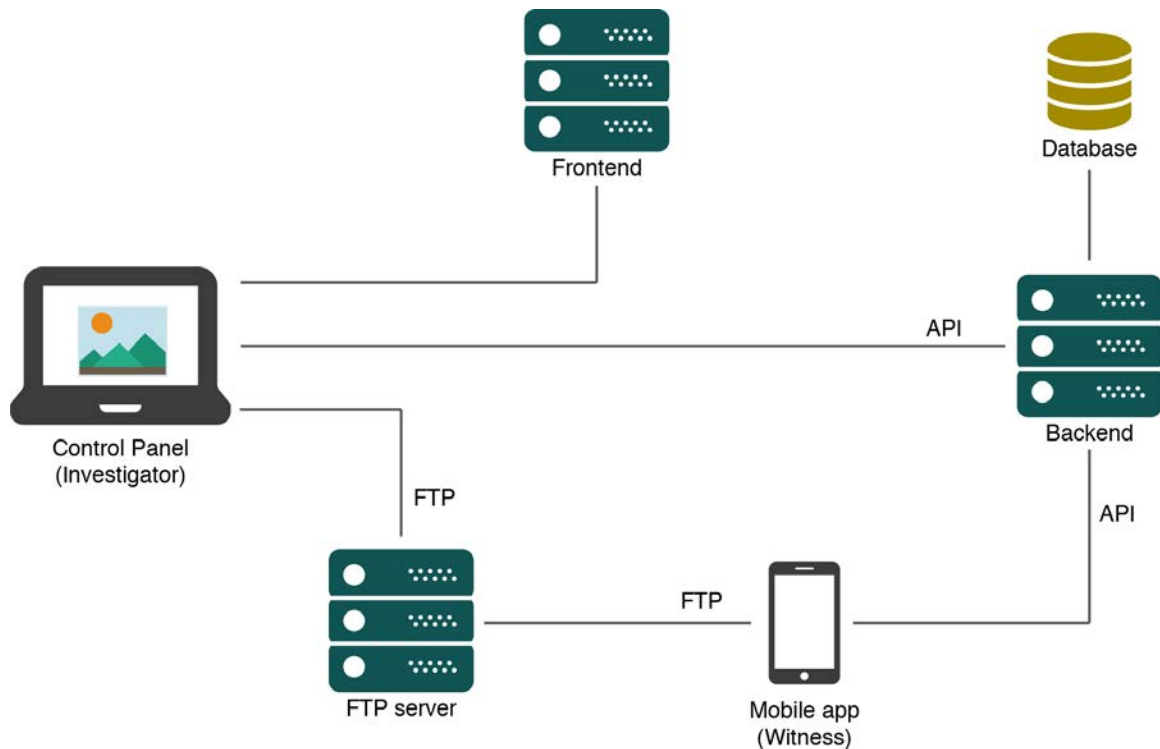


Figure 10: Details of the cloud server and components

The server application has different user management considerations for the internal users (the investigators).

4.1.5 Mass Incident Investigators Panel and Analysis

The (investigator) panel has been designed as a web application that can only be accessed from the private network. A browser-based application provides better usability and performance when doing large computations. The frontend application was developed using the Vue.js framework, which is a stable framework that integrates very well with HTML and so far it has a very good community support. The investigator panel will be used by the investigator as a tool for requesting evidence to the witnesses and to process it later. There are two types of evidence processing: automated and manual. Automated processing is the one performed by the server and will show the investigator the results (e.g.: photo tagging). Manual processing is any kind of work done by the investigator (e.g.: adding notes).

An important part of our proposed mass incident analysis system is aggregating data and collecting relevant digital forensics evidence from the data collected. Consider an example where the investigators want to query for evidence such as a bag set down at some place from the large set of images collected. There are many deep neural network models that could be used for this purpose. We have implemented one such model in our current TDES system where a concern was the computing power available on a smartphone. For the TDES-mi no such concern is necessary as we can implement any desired model on backend machines, which makes the system more scalable. We could use available models or develop new sophisticated models that

support the needs of investigators. Visualization techniques such as 3D could also be used with the large set of images likely to be available and taken from somewhat different locations, perspectives and time. This in fact could become a very broad area for exploration.

4.1.5.1 Investigator's panel use cases

Add, modify, and list cases

This page allows users to add/modify/delete the various case files. An investigator starts with adding a case number and description of the case. Cases will be displayed with their number and description. The investigator will be able to define a “filter request” based on the relevant mass incident which is limited to photos and with a small date range. Based on the filtering request and assuming witnesses have started to use their mi-app, the request will be available for users of the mi-app, so when they open the app, the request will show up. Note that the filtering request will filter the selected data from the witnesses’ phone and display this to the witness. At this point the witness will have only the option of unselecting any of the data that is deemed irrelevant or inappropriate or that the witness desires not to have uploaded. After the witness desired selection, the rest of the data will be uploaded to the investigators’ server in the cloud.

Process Mass Incident Data Received (in the cloud)

The system tags each specific item in a photo of interest to the investigators. Thus, filter request might have been fairly general: all photos during a specific period of time and within 10 miles of the incident. Once uploaded we run a machine learning model to look for anything similar to bags, weapons, and other suspicious objects in all the pictures and have them tagged.

Visualize and process case evidence (by investigators)

Visualize pictures

The investigators will be able to visualize pictures related to the mass incident. A list of pictures will be shown so an investigator can pick from that list and expand as desired. As the pictures will be pre-processed, once expanded, they will contain tags to indicate the objects detected by the machine learning model. This will ease the investigator’s work in a significant way, since they will not have to tag all the items that are present on the images. See Figure 11.



Figure 11: Photo visualization and tagging interface

4.1.6 Mass Incident System Security Considerations

We have explored several security issues related to the development and implementation of our system. For example, the TDES-mi app could potentially upload data from any location. The API endpoints in the cloud thus need to be secured to prevent any misuse from such an attacker. We also need to take necessary precautions to prevent DoS or DDoS attacks. We need to be concerned with both server-side security as well as mobile app side security. We have explored some of these issues in our Lab’s experimental public / private network modeling our problem environment. Standard security measures such as firewalls and configurations on the internal machines in the cloud should be implemented in the future in order to ensure a minimum security level of the system. We also considered a few issues we are considering with respect to security:

- Appropriately using HTTPs and SSL certificates
- Using *Cloudflare* to prevent DoS and DDoS
- Using Google’s *SafetyNet* attestation API for Android phones
- Shared private / public keys for iOS phones
- Using device ID and user-registration system.

We have implemented many of the idea we have discussed and consider we have produced a good prototype for the TDES-extraction system. We have been able to upload data from a phone app to a server in the “cloud” and tag it automatically by

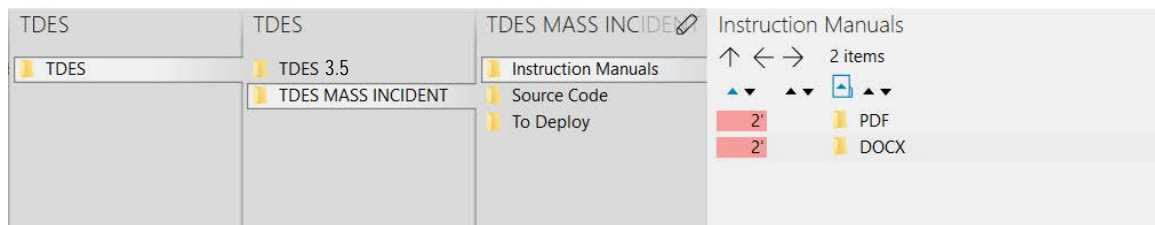
using a machine learning model, which will ease the work of the investigators in a very significant way and opens the door to implement many new features for further analysis. We have not yet focused much on the TDES-analysis system as our intent is to implement only basic analysis capabilities. Obviously a very sophisticated analysis system would require substantially more resources than we ones had available.

4.2 Deliverables: TDES Mass Incident Version 1.0

The full set of all deliverables for this project is simply the TDES Mass Incident Directory. This contains everything related to the complete project: all source code, object code (packages), deployment instructions and all related manuals, both for deployment and use. As mentioned before, these can be delivered by the PI as a memory stick upon request. In the next sections we describe the components of the directory file. For convenience, we attach as part of this report the TDES-MI deployment and user manuals titled “TDES-MI.”

3.2.1 The Main Folder and First level Sub-Directories

The main folder is called TDES. It has a subdirectory called TDES Mass Incident within it. There are three subdirectories: *Source Code*, *To Deploy*, and *Instruction Manuals*. The high-level directories / files are shown below.



In the subdirectory Instructions Manuals, there are three manuals for investigators (users) explaining how to use our system. These are TDES_MI_Android, Server_deployment, and TDES_MI_Investigator_Panel. These contain information on initial setup of the TDES Server, connecting to the target device and installing the relevant app, running the TDES Manager Dashboard, entering information into the user interface on the app to extract requisite data, getting data back to the manager and using the manager report.

3.2.2 Source Code Sub-Directory

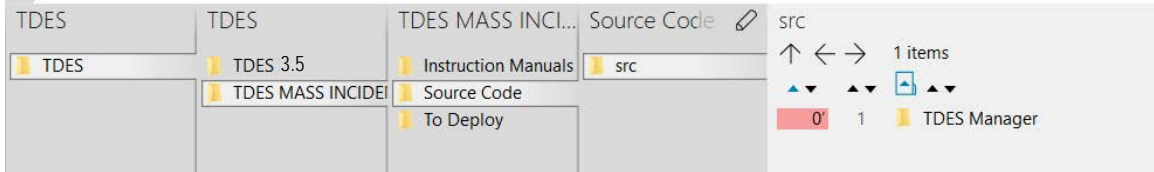
This directory contains all the source code of our system including various instruction and read-me files.

Android Source Code

The Android-Mobile-App subdirectory has the analogous information for the TDES android app as described above for the iOS App. In this case the programming language used is Java and the resulting packaged file is called an .apk file.

TDES Manager Source Code

Recall that the TDES manager is deployed on the Windows operating that is booted up by an investigator from the solid-state drive, USB stick or other portable device. The directory structure of the source code for this program is shown below.



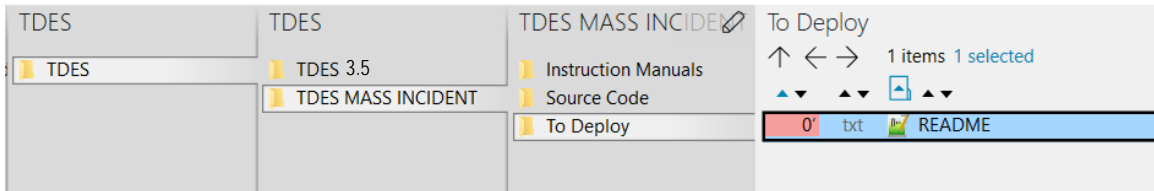
The source code in this directory is used to create the TDES-manager.jar file which must be moved to the portable device.

Compiling Instructions

In the subdirectory Source Code, there is also a directory *Instruction Manuals* which contains instructions on compiling the three major source code systems just mentioned: Android-Mobile app, TDES-Server and TDES Manager Dashboard. Note that these instructions are different from the user instructions in the first level sub-directories.

3.2.3 To Deploy Sub-Directory

Deployment information is described in the relevant folder. See the structure below.



The file README.txt explains how to deploy the TDES-MI system. It is designed to be straightforward and can be done by law-enforcement using simple technical skills.

4. SEAS Version 1.0

In this section we discuss our third deliverable, an iOS Schema Evolution Analysis System. The name of the file is **SEAS.pdf**. We briefly review the system as part of this final report.

4.1 iOS Schema Evolution Analysis System Overview

Logical acquisition of iOS data from an iPhone backup is widely used by many forensic tools. Apple primarily uses SQLite databases to store the backup information. Whenever Apple launches an update/upgrade to the iOS version, the database schema of the native applications could be changed. However, forensic tool developers have very little knowledge about what is changed in the database schemas as they have no control over how the iOS native app's database evolves. In our research and development of the TDES system we faced this very same problem. Whenever a major iOS release happens (typically once every year), forensic tool developers have to do substantial code rewrite to cope with the iOS schema evolution. In particular, we faced the problem of changing our code related to SQL queries in TDES Manager. This led to this subproject, which is designed to help developers, and software maintainers quickly rewrite code related to accessing Apple iPhone SQLite databases found in the iTunes backup.

We thus began the development of SEAS, a tool which analyzes the differences between two versions of SQLite schema. Although the tool can clearly help us in our needed code rewrites upon iOS version / upgrade changes, our system is designed to be sufficiently flexible so that the tool can be more generally used by other forensic tool developers trying to extract and analyze data on iOS backups. The backup of iOS is very rich and is frequently used for forensic data analysis. For our TDES system we can extract queries from the relevant TDES manager code automatically, analyze them and thus reduce the task of future maintenance of our system. In order to understand iOS changes, we investigated four consecutive versions of our iOS backups (iOS version 9 through 12), analyzed the changes manually and showed how we could do better with the help of SEAS. In extracting data using TDES, such as extracting call logs and messages from iOS devices, we do need to extract this data from the iOS backup. In TDES, for iPhones, we use the iTunes backup when we need to extract some types of data such as call logs, messages and data from third party apps. Our program to extract this data is written in Java (TDES manager) and uses SQL queries to access this data from a backup. If the iOS version changes, we have to make changes to this program. Thus, SEAS can help us in making these changes more quickly.

4.2 SEAS Goals

In our design we focused on two aspects of coping with the iOS updates. First, we provided information about changes in the SQLite backup database schema design. And second we have implemented a query analyzer to determine what has changed with respect to specific SQLite queries. An iOS backup is stored in a hierarchical format. Inside the main backup directories, there are several subdirectories. Inside these, the iOS backup information is stored in various SQLite databases. SEAS

supports the following features in order to provide information about schema changes:

- **Revision Changes Identification:** SEAS uses backup folders of two versions as input and displays the same named backup files of the two versions. It can find the following changes when given two versions of a SQLite database: added or deleted tables, renamed tables, added or deleted columns in a table, renamed columns and data type changes for a column. Note that in order to get column renaming and table renaming, the backup of two versions should have the same data. Thus, given an iOS revision, we first get a backup of an iPhone before the update and then get a second backup of the iPhone after the update.
- **Query Analysis:** An important feature of our SEAS tool is our query analyzer. Given a SQL query, we parse it to get the list of tables used in that query and then SEAS determines how to rewrite the query if needed. The query analyzer also runs the given SQL query in both versions and compares the results. Then it provides feedback on the results. If it gets the same results in the two versions, it gives a success message and it is likely the query does not have to be rewritten. If the query runs successfully but gives a different result, then the software tool maintenance team will have to rewrite the query by checking the differences shown by the query analyzer.

4.3 SEAS System

Figure 12 shows the design of SEAS. The system takes two iOS backup versions as input and identifies the differences in the SQLite databases in the files of the backup. Additionally, there is a *Query Analyzer* component that is designed to output the changes (tables and columns) relevant to a SQL query. It has three capabilities: (1) given a SQL query, output all changes related to that query; (2) given a program such as the TDES Manager, parse the file for all the SQL queries (which are only written inside the *executeQuery()* method in our code) and output all the changes; and (3) given a Java or Python program, modify it to instead make calls to the query analyzer (and thus output changes) for each SQL call made in the program.

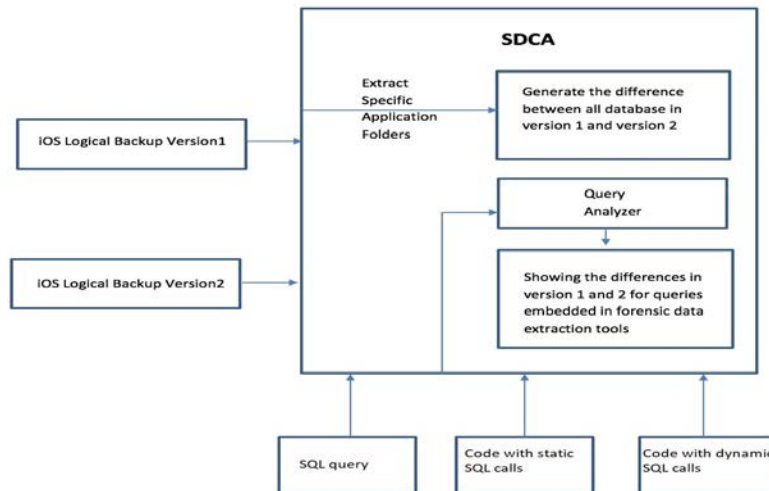


Figure 12: SEAS high-level design

4.4 Analyzing Differences in SQLite Databases

SEAS's GUI was developed using Python and PyQt5. It is designed for use on a Mac OS. The SQLite engine is used to retrieve information about the tables. Given the two input versions, we abstract the information into several tables as shown in Figure 13. For each version we have the database meta-information and the table meta-information. Information about each query is also stored. Using this information, we check the evolution of each table in version 1 as compared with version 2. If the name also appears in version 2, we assume it is the same table. If not, we can decide that a table has been created or dropped. A similar approach is used for the columns of each table to decide if the columns are the same or have been dropped or added. Renaming is a little more complex and data entries are used to decide if tables or columns were renamed.

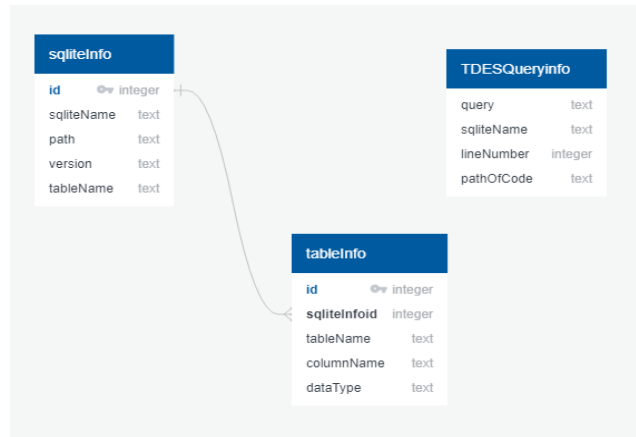


Figure 13: Internal Data Structures

4.5 Query Analyzer

As discussed previously we have developed three versions (or capabilities) of the query analyzer. In the simplest case, we simply allow the developer to enter a query and we then output whether tables or columns related to the query has changed and whether the results are the same or not. This provides some guidance to simplify the developer's task.

To assist TDES source code maintainers, we analyzed the queries used in the TDES manager. We noted that none of the SQL queries were formed at runtime but rather all of the queries were statically defined in the source code. Hence, we could simply parse the TDES manager application source code and check each SQL call to give information about what needs to be changed. Figure 14 gives a snapshot of an example where the iOS version was updated.

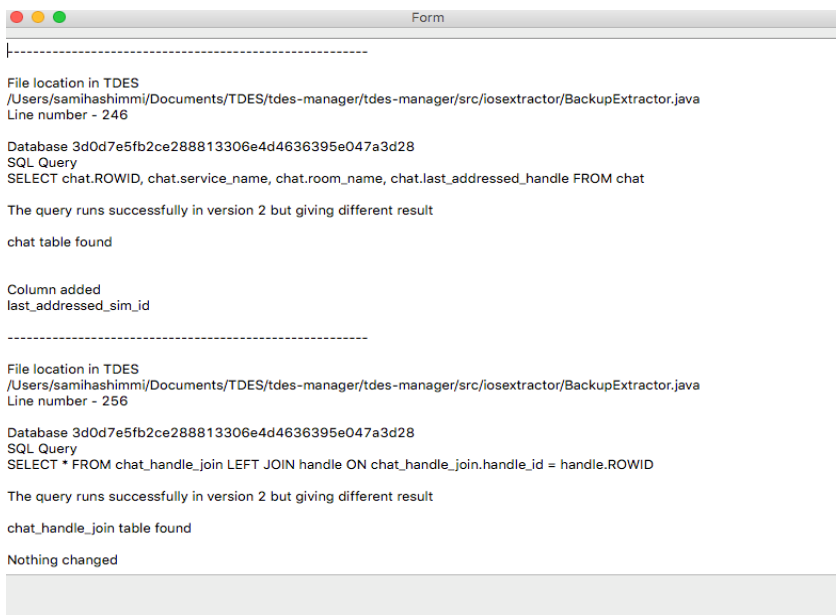


Figure 14: Query Analysis for TDES Manager

The query analyzer can also be invoked from an external source code at runtime. SEAS tool will be useful for developers and source code maintainers as frequent changes in the iOS versions are expected. And the ability to use SEAS as a stand-alone tool or as a static analyzer (where the source code of a forensic tool may contain hard-coded SQL queries) or as a dynamic analyzer (where the source code contains run-time values for SQL queries) gives more flexibility for forensic software maintenance teams.

4.6 SEAS Deliverables

We are delivering the SEAS tool as a separate branch of the TDES system. We decided to keep this as a separate branch since it can be used to maintain any forensic data extraction software tool where SQL queries are used as part of the source code. We deliver this as a SEAS Directory, which has all the code, documentation and instruction manuals, as previously described for the TDES system.

5. TDES Deliverables to NIJ

5.1. Memory Stick

We can deliver all code and documentation on a memory stick upon request by NIJ. The total amount of data (500 MB) is too large to upload as a file.

5.2. Solid-State Hard Drive

We have a fully working bootable SSD that an investigator would use for TDES (extract data from IOS or Android phones). We can also supply an SSD drive related to TDES-MI but it would require the deployment on a server of the backend system.

5.3. BitBucket and Github Repositories

Source code of TDES projects has been uploaded to BitBucket and Github repositories. The PI can work with the NIJ to allow access to these repositories if needed. The repositories are organized as follows. Note that, there is a “git clone” command prepended to all links of the repositories.

- TDES v3.5:
 - Manager Application (TDES v3.5):
git clone <https://gokiladorai@bitbucket.org/A01211341/tdes-manager.git>
 - iOS Application (TDES v3.0):
git clone <https://gokiladorai@bitbucket.org/gokiladorai/tdes-ios-app-v3.0.git>
 - Android Application (TDES v3.5):
git clone <https://gokiladorai@bitbucket.org/NicholasGuerra/shsu-android-mobile-data-extraction.git>
- TDES v1.0 Mass Incident:
 - Mass Incident:
 - Android Application (TDES v3.5):
git clone <https://gokiladorai@bitbucket.org/ap9292/tdes-mi.git>
 - Investigator Panel:
git clone <https://gokiladorai@bitbucket.org/juanpabloconde/tdes-panel.git>
 - Cloud Server:
git clone <https://gokiladorai@bitbucket.org/ap9292/tdes-cloud.git>
- TDES v3.5 Keyword Search:
 - git clone <https://gokiladorai@bitbucket.org/kobra1370/tdes-android-text-search.git>
- TDES v1.0 Schema Evolution:
 - git clone <https://github.com/SamihaShimmi/SqliteDif.git>

5.4. Delivered by File Uploads

Previously: Interim reports, a draft final report, manuals and version of published paper.

This document: Final report, including TDES Manager and TDES-MI manuals. Separately, .pdf of TDES paper and .pdf of SEAS submission.

TDES Manager

Android 3.5

iOS 3.0

ECIT Lab

Florida State University

Table of Contents

TDES Manager 3.5 (Android app)	2
Initial setup	2
Enabling USB Debugging Mode on Android	3
Connecting the device	4
Running the TDES Manager	4
Using the Report	12
TDES Manager 3.0 (iOS app)	13
Initial setup	13
Connecting the devices to a network	13
Trusting the device	14
Installing the application	15
Running the TDES Manager	16
Using the iOS App	18
Using the filters in the app	20
When Filter Screen	20
Where Filter Screen	21
What Filter Screen	22
Machine Learning Image Filtering	23
Who Filter Screen	24
Consent Form	24
Viewing and selecting evidence	25
Exporting evidence	26
Going back to the TDES Manager	28

TDES Manager 3.5 (Android app)

This manual is written to instruct users to properly use the TDES Manager on Android devices. This is an alpha version, which means that minor bugs may occur, requiring the user to perform certain tasks to reset the application.

Initial setup

The first step will be to boot the desired computer, with the TDES Manager installed, from the solid state drive. On most computers, with varying vendors, it is usually required to press the <F12> key when booting the computer. Some computers might require users to press different keys, for example, there are laptops that will require to press <Enter> and then <F12>. If system prompt to choose booting devices, select the "UEFI: Samsung portable SSD T5 0", then press <Enter>.



Figure 1

Once the computer boots from the SSD, a Windows 10 login screen will be prompted. The user is "TDES-SYSTEM" and the password to login is "TDES2018".

Once logged in, a lists of files, which are used to run both TDES Manager on IOS and Android, will appear on the desktop. Select the icon named server with the following icon. (Figure 2)



Figure 2

Figure 1.2

Enabling USB Debugging Mode on Android

Prior to connecting the Android device to the computer, Android device is required to be in the USB debugging mode ADB. The ways to enable USB Debugging mode varies from one Android version to another; here we describe the general procedures.

Navigate to **Settings > Developer** options configuration screen on the Android device

1. If Developer options is not enabled, navigate to **Settings > About** device and tap on the **Build number** seven times. Return to previous screen, **Developer options** should be visible now.
2. Click to open the **Developer options** screen, select “ON” mode and enable the USB debugging.

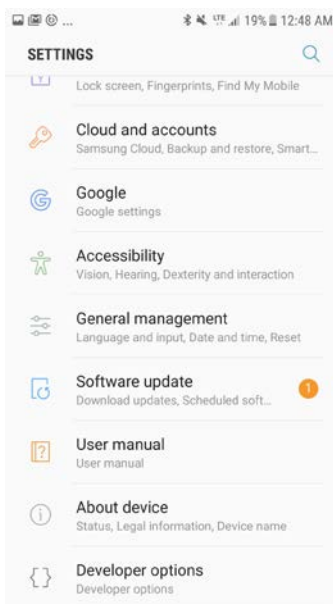


Figure 3



Figure 4

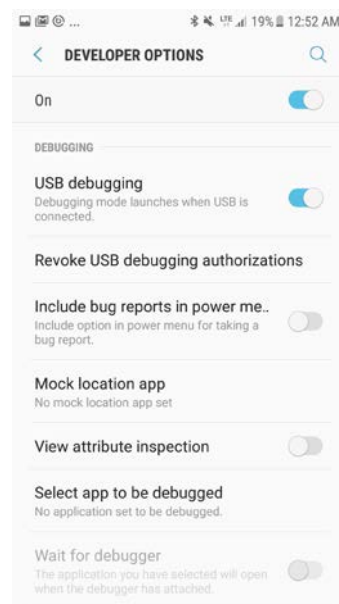


Figure 5

Connecting the device

Connect the Android device to the computer with a USB cable. The phone will prompt to allow access to device data, select <ALLOW>.

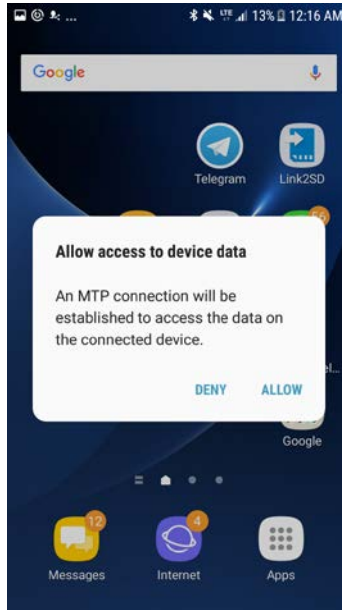


Figure 6

Running the TDES Manager

After ensuring that the device is properly connected with display screen turned on, double-click on the previously mentioned "server" icon on the computer desktop to open the application.



Figure 7

1. Select "Android" on the following displayed window.

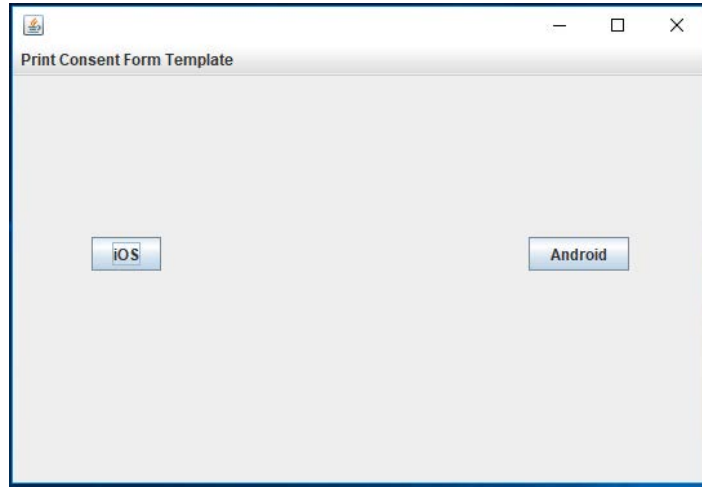


Figure 8

2. The Android Initialization window will then prompt for users input on case number, investigator's Name, Device Owner's Name, and Export Directory. These information will later be used to record and export a investigation report.

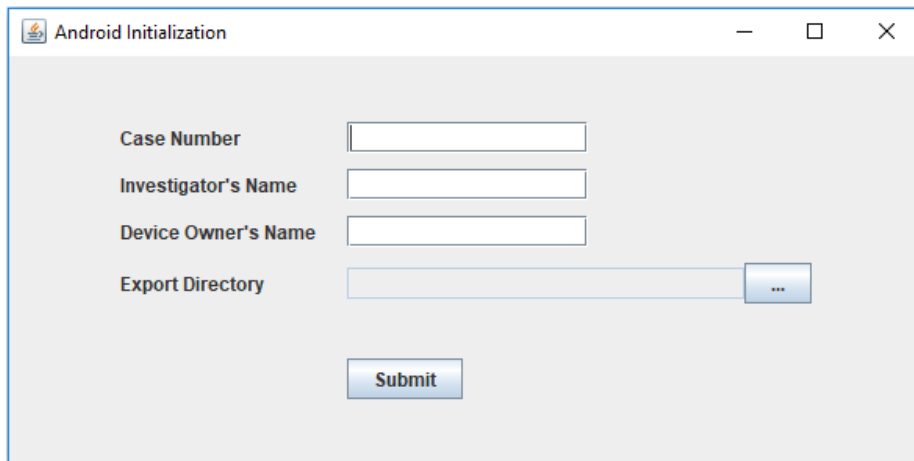


Figure 9

3. Once you click submit, TDES Manager will install the application on the Android device. A window displaying server status will show up to indicate when installation is done. Upon completion of the installation, screen labeled as Figure 11 will pop up on the Android device. Select device in order for the user to sign a consent form on the Android device prior to data extraction

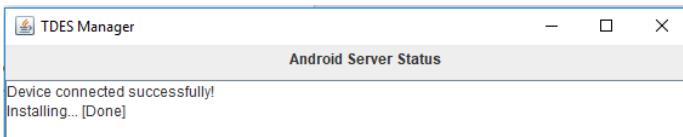


Figure 10

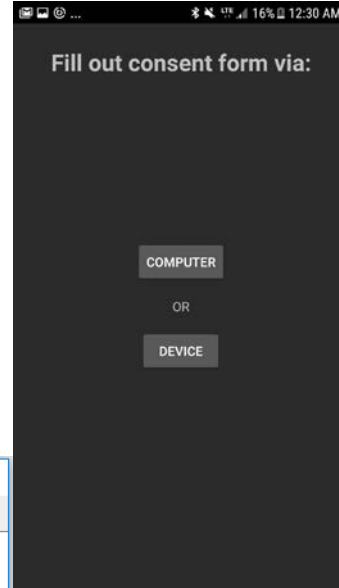


Figure 11

Using the Android TDES

1. After the “Device” option is selected, the TDES will prompt the user to grant permission to access specified data. The options are “Call logs”, “Contacts”, “Messages”, “Videos”, “Photos”, and “Calendar”.

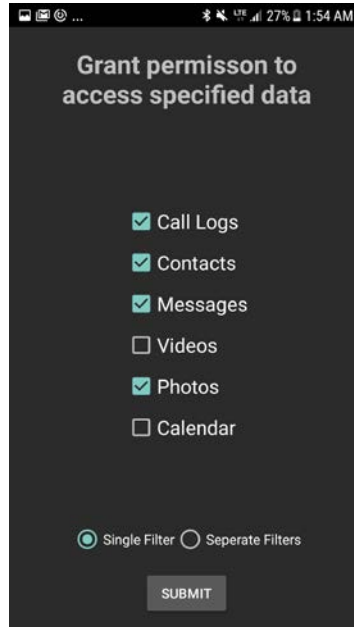


Figure 12

2. After the data types are selected, TDES will prompt the user to filter the selected data types by, “Data and Time”, “Name, and Number:”, “Location”, and “Machine Learning”. The filtering option will vary depending on the data types selected. The filtering options listed here assumes that the user granted permission to all the access specified data in Figure 12.

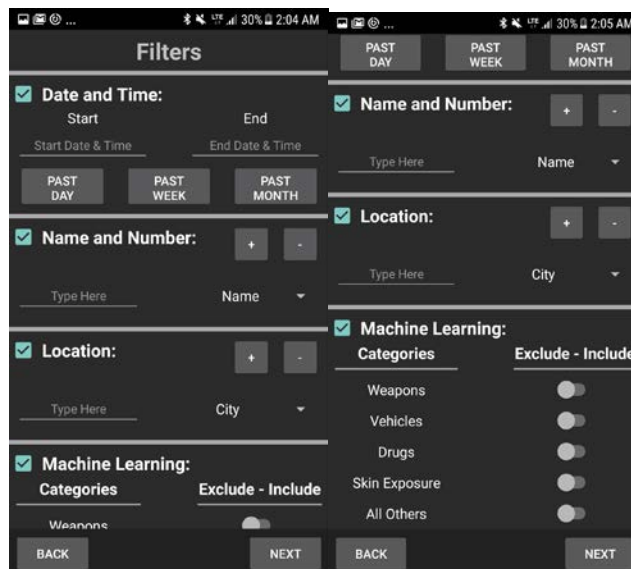
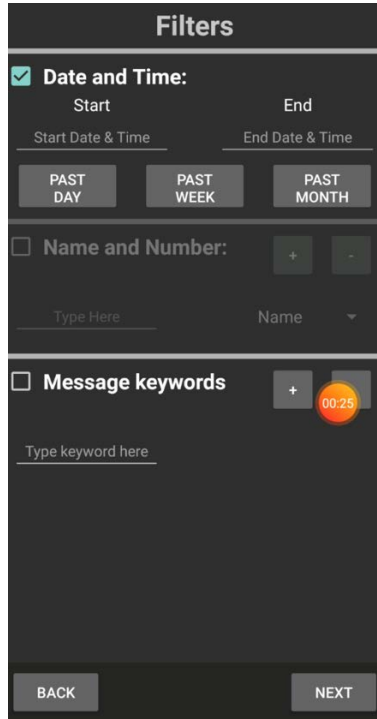


Figure 13

Figure 14

3. The user will also be able to filter messages by keyword by defining the keywords they want to use for the search on the keywords field. Keywords have to be separated by spaces.



4. Once the filtering option is completed, the TDES will prompt the user for the signature. This signature proves that the user has given consent to export the information selected and filtered by the user.

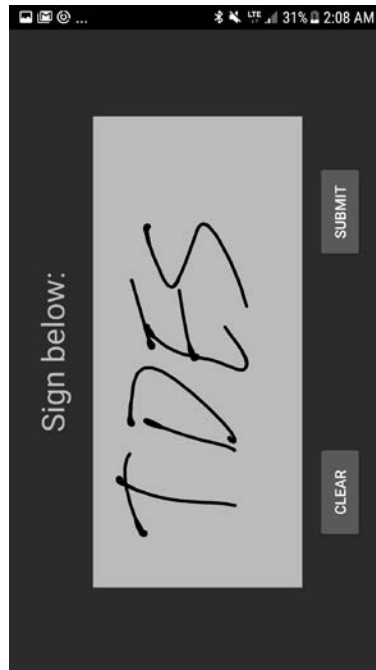


Figure 15

5. Once the signature is signed, TDES will prompt the user to allow access to the data such as contacts, phone calls, SMS messages, calendar, and photos on the Android device.

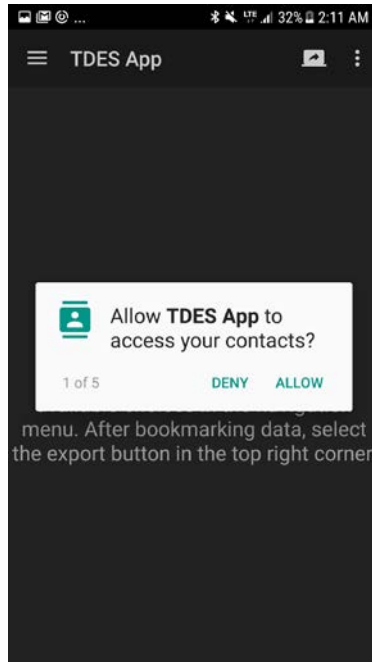


Figure 16

6. Once the user allows access to the data stored, a “Data Extraction” screen will pop up. In this screen, the user can narrow down the filtering within the consent given and furthermore, bookmark a specific data to be exported.

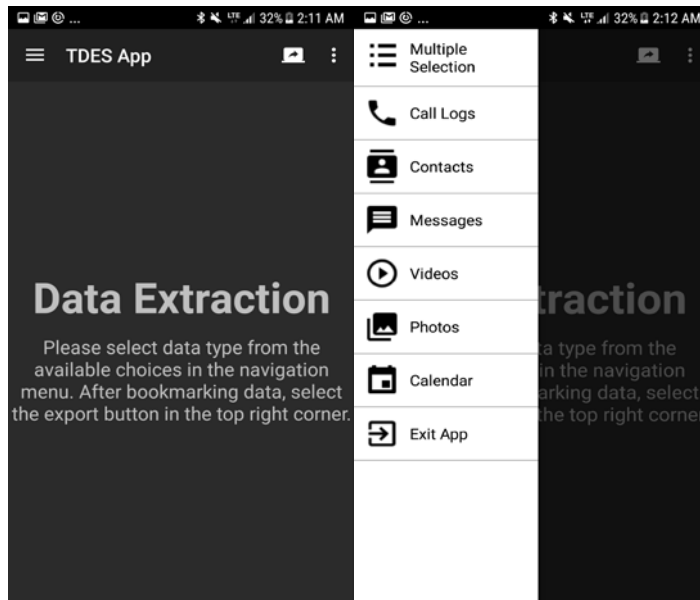


Figure 17

Figure 18

- The user can select each data type at a time to narrow down the filtering and bookmark the necessary data. However, TDES ensures that additional filtering will not go beyond the consent given by the user. For example, if the consent given allows the export of data with in the past week; Then TDES will only allow the additional filtering with in the past week such as past day. In order to bookmark a specific data, the user will need to tap on the data. Once the specific data gets highlighted in blue, user needs to tap the bookmark icon located at the upper right corner of the screen.

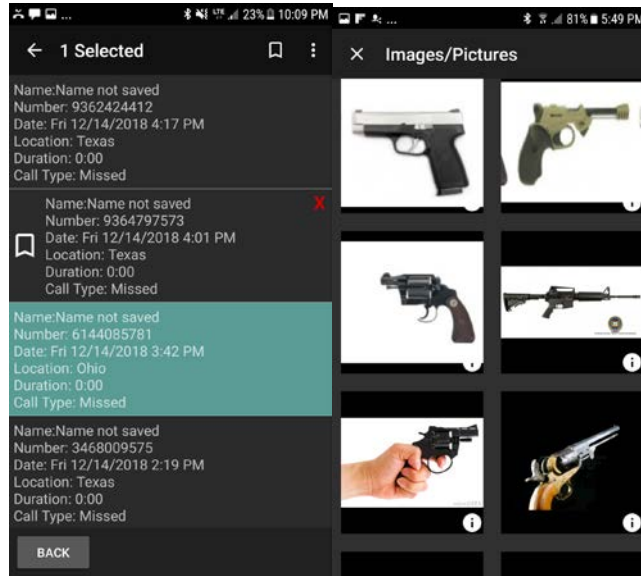

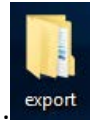


Figure 19

Figure 20

- Once all the data has been bookmarked, TDES will export the bookmarked data to the computer by tapping  located on the upper right corner on the “Data Extraction” screen.
- After a successful exportation of data, TDES on the Android device will terminate and gets uninstalled automatically.

Using the Report



1. On the Desktop, click the file named “export”.
2. On the “export” file, select the case number that the user has specified at step 2 as “Running the TDES manager” section.
3. Click on a HTML file named “Report”.
4. A report with investigator’s information, device owner’s information, bookmarked data, Case summary, and Filtering options will be displayed.(Figure 22)
5. When a Signature PNG file is clicked, the signed signature by the device owner will show.

Iteration-1	12/14/2018 8:44 PM	File folder	
ConsentFilter.json	12/14/2018 8:44 PM	JSON File	1 KB
Init.json	12/14/2018 8:44 PM	JSON File	1 KB
Report	12/14/2018 8:44 PM	HTML File	8 KB
Signature	12/14/2018 8:44 PM	PNG File	16 KB

Figure 21

TARGETED DATA REPORT	
Section-1. Case Summary	
Device Information	IMEI:356397083944942 Phone Number:19362422380
Device Owner Information	Device Owner Name:Iwal E-mail ID (mobilelabs7@gmail.com)
Investigator Information	Investigator Name:Taka
Case ID	001
Section-2.	
Contacts	Name/Number Filter Not Used
Call Logs	Name/Number Filter Not Used Date/Time Filter Used: TRUE From: Mon Nov 12 21:00:00 PST 2018 To: Fri Dec 14 20:59:59 PST 2018
Messages	Name/Number Filter Not Used Date/Time Filter Used: TRUE From: Mon Nov 12 21:00:00 PST 2018 To: Fri Dec 14 20:59:59 PST 2018
Photos	Date/Time Filter Used: TRUE Photos Consent Information From: Mon Nov 12 21:00:00 PST 2018 To: Fri Dec 14 20:59:59 PST 2018 Location Filter Not Used Machine Learning Filter Used: TRUE

Figure 22

TDES Manager 3.0 (iOS app)

This guide was written to aid in the use of the TDES Manager. This is an alpha version, which means that minor bugs may occur, requiring the user to perform certain tasks to reset the application.

Initial setup

The very first step will be to boot the desired computer from the solid state drive. To do this it is usually required to press a key when booting the computer, this key may vary depending on the vendor of the device. For example, there are laptops that will require to press <Enter> and then <F12>, some others just require <F12>. Once the computer boots from the SSD a Windows 10 login screen will be prompted. The user is "investigator" and the password to login is "investigator."

Once we login we will see several icons in the desktop. We briefly explain what they do (order is from left to right)

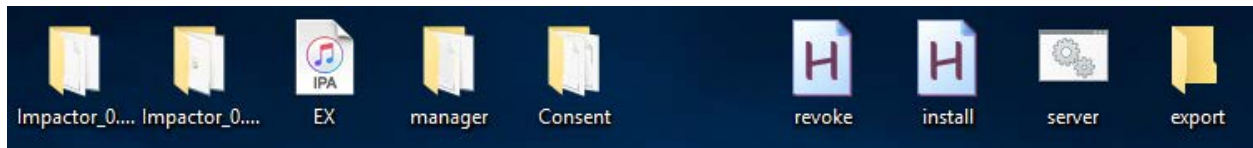


Figure 1

Files you won't interact with:

- Impactor_0.9.43: A tool used for installing the application.
- Impactor_0.9.44: A tool used for installing the application.
- EX: The application that will be installed into the device.
- manager: This folder contains the application that runs in the computer.
- Consent: This folder holds the physical consent form.

Files you will interact with:

- revoke: This file will make your device to be ready for the app install.
- install: This file will install the application in the device
- server: Once the app is installed, you will use this to run the desktop application.
- export: In this folder you will see the evidence that has been extracted.

The "Files you will interact with" displays in which order we will use the icons. We will explain in detail what happens in each stage.

Connecting the devices to a network

Once your computer is ready you will need to connect to an access point with internet connectivity. If you have a mobile hotspot then connect both devices to your hotspot.

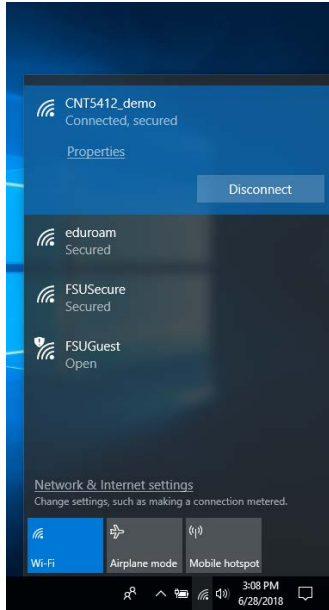


Figure 2

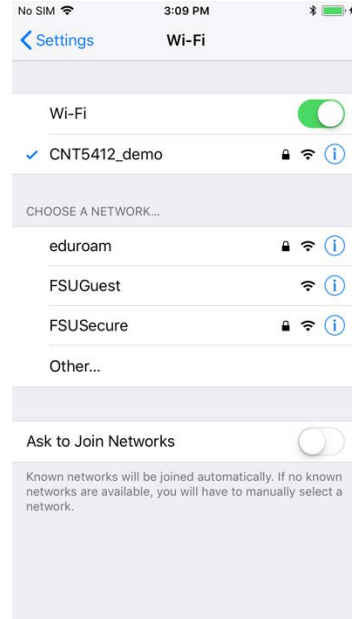


Figure 3

Trusting the device

Connect the USB cable to the iPhone and to the computer. You will see that the iPhone will prompt a window which asks you to "Trust" or "Don't Trust" the device, we will tap on "Trust".

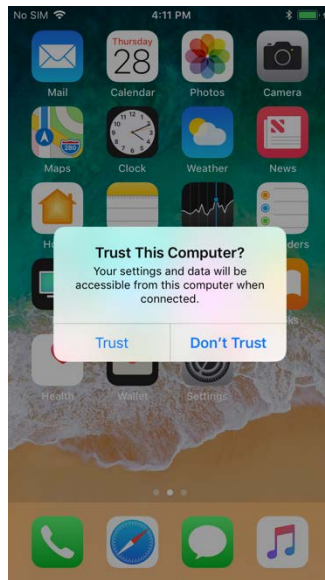


Figure 4

After you trust the device then it will ask you for the device's password

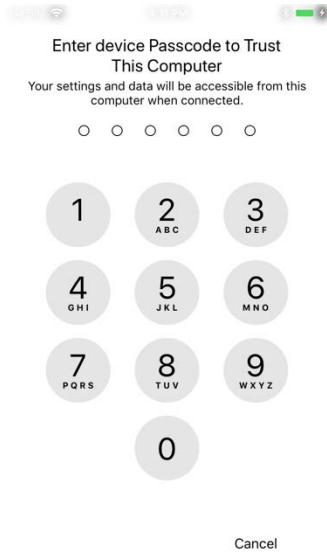


Figure 5

Now you are set. Probably an iTunes windows will come up on the computer, you can minimize it or close it.

Installing the application

Now you are ready to install the application. In order to do so you will double click the icon that says "revoke"



Figure 6

You will let this program run, it is making sure your phone can run the application. You will know that it is done after it displays a message like the following:

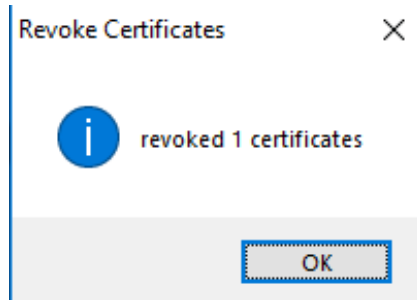


Figure 7

Now you can close the windows named "Cydia Impactor". Next, you need to double click the "install" icon.

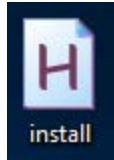
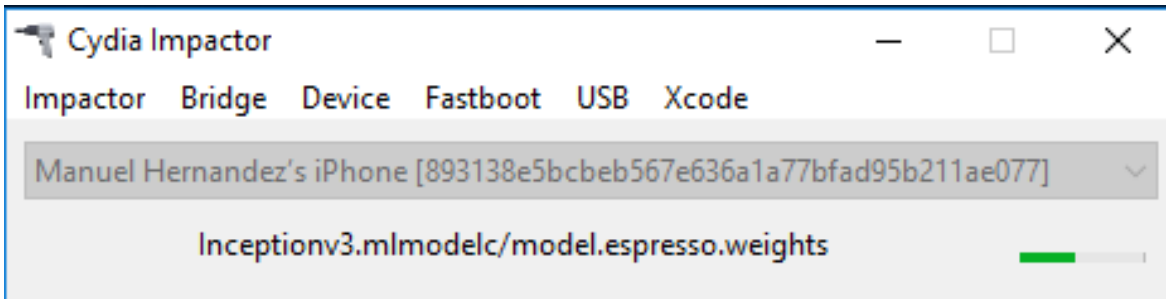


Figure 8

The procedure will be similar to the previous step, we will know that it is done when there are no more progress bars at the bottom left corner of the application



Once this is done you can close again this window.

Running the TDES Manager

Our iOS app requires the TDES Manager to be able to transfer the data from the device. After you completed the previous steps you will start the server by double-clicking the "server" icon

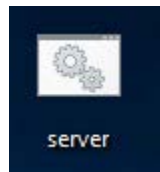


Figure 9

The following screen will be shown

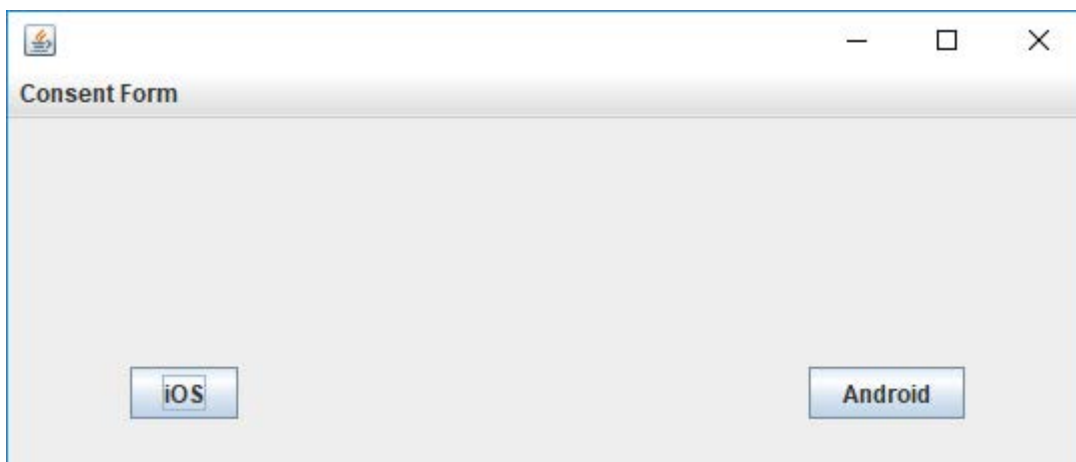
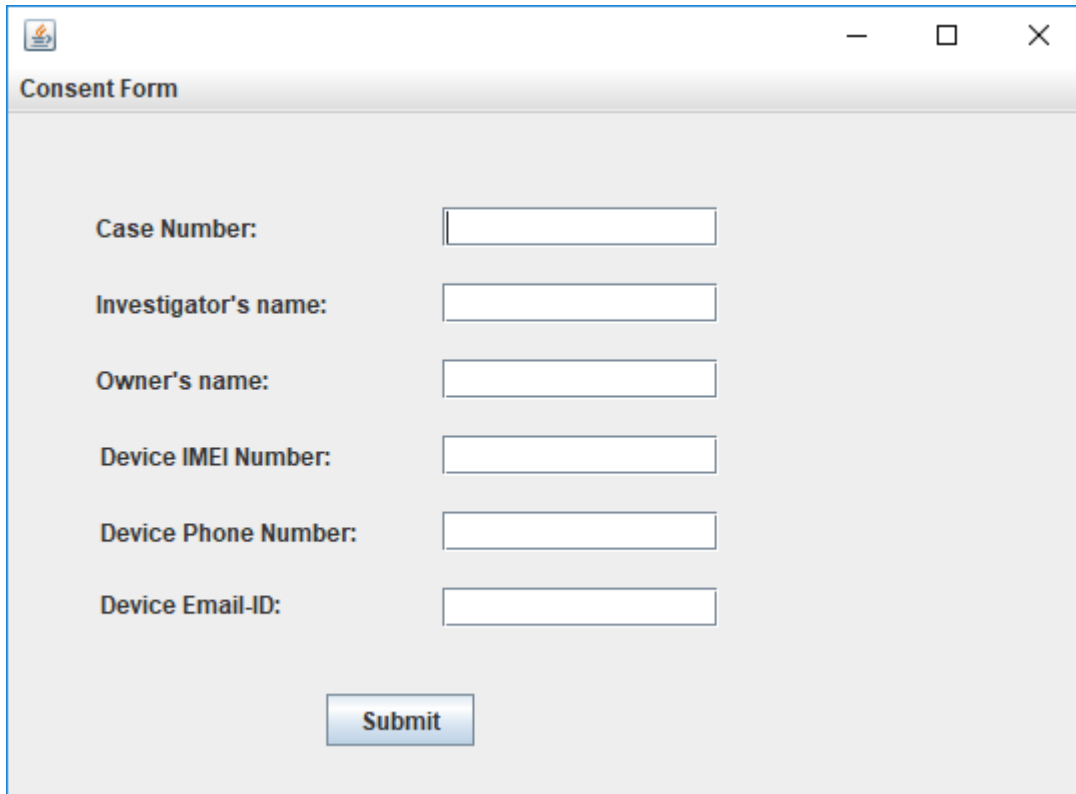


Figure 10

In this screen you have the option to choose the type of device you will be using; this guide is focused on the iOS platform, so we will select it. After that a form will be displayed, this form has fields related to the case information. All the fields are optional **except for "Case Number", which requires a unique name**. To verify you entered a valid name you can explore the folder "export" in the desktop and check that you selected a unique name.



The screenshot shows a window titled "Consent Form" with a standard Windows-style title bar (minimize, maximize, close buttons). The form contains the following fields:

- Case Number: [Text Input]
- Investigator's name: [Text Input]
- Owner's name: [Text Input]
- Device IMEI Number: [Text Input]
- Device Phone Number: [Text Input]
- Device Email-ID: [Text Input]

At the bottom center of the form is a "Submit" button.

Figure 11

Once you are done filling the form you will click Submit. A message will be displayed asking if you require to extract messages or call logs.

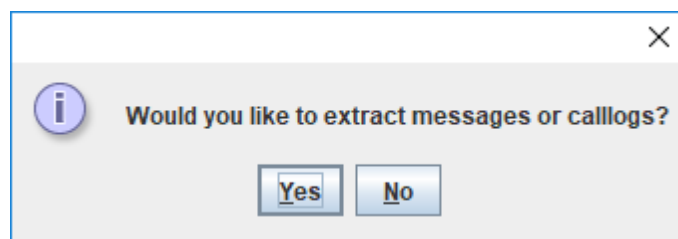


Figure 12

Clicking "Yes" will allow you to extract these resources, however beware that this will create a temporary backup of the entire device. This temporary backup is deleted after the specified elements are extracted. If "No" is selected, then you won't be able to extract messages or call logs. The "No" extraction option will allow you to work faster since there is no backup creation.

Depending on what you chose you will see different messages in the console. If you created the backup, then you will see all the files been transferred. The console will tell you that you need to connect to some IP, now you will proceed on the iOS side.

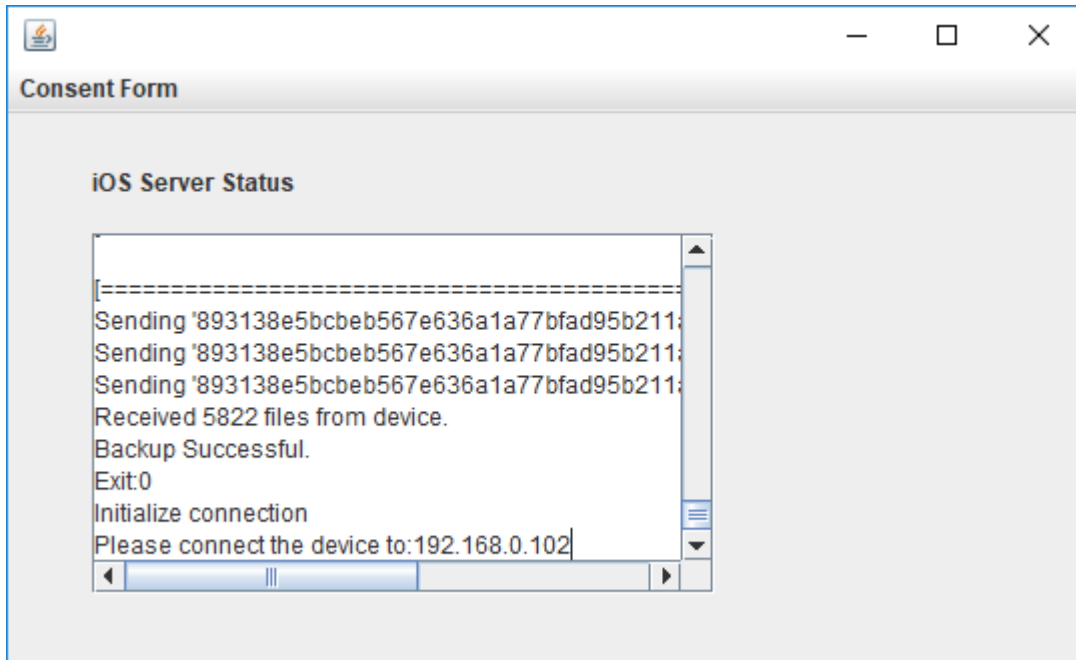


Figure 13

Using the iOS App

On the iOS device we will see that an app "EX" was installed. In order to use the installed app, you need to verify the developer which installed the application. To do so go to Settings -> General -> Device Management. From there you will select seminolermobile@gmail.com and then select the option "Trust seminolermobile@gmail.com"

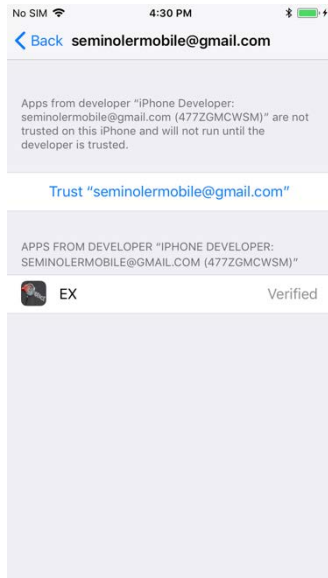


Figure 14

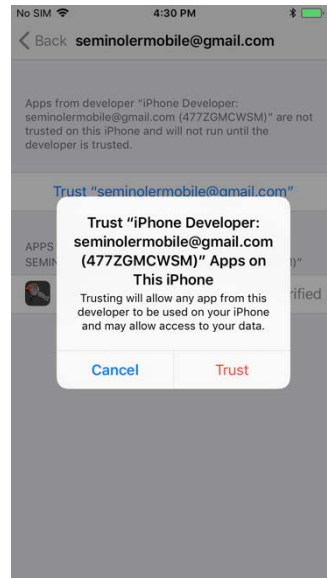


Figure 15

Now press the home button and tap on the app's icon

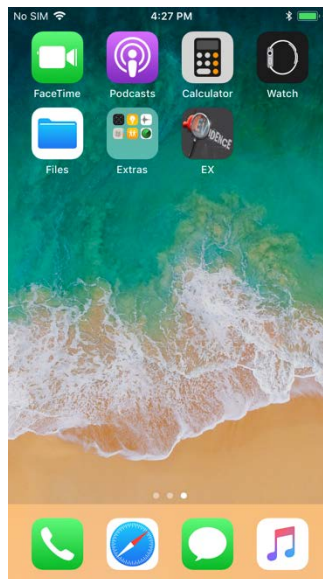


Figure 16

There will be several prompts asking for access to the device's content. For the app to work you **must** always select "OK"



Figure 17

Now there is a screen that asks for an IP address, this is the IP address that the TDES Manager tells you to connect to, after typing the IP address properly tap on the "Connect" button

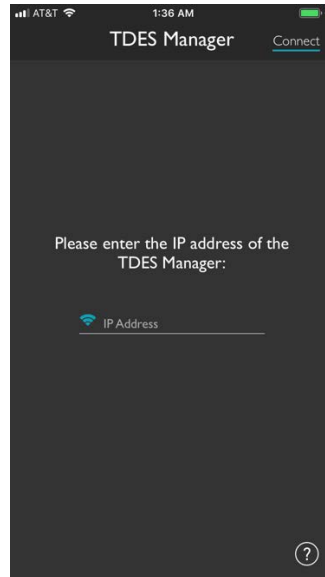


Figure 18

NOTE: If you didn't type the IP address properly or there is a network error the application won't let you get to the subsequent screens. Please make sure that the devices are connected within the same network and you typed the IP correctly. If by mistake you mistyped the IP, you need to restart the app. You can restart the app by double clicking the home button and then sliding the app up.

Using the filters in the app

When Filter Screen

From this point on we will be selecting the filters for the data. Our first screen is "When" which allows us to specify the time frame that we will be looking into. We have quick switches that allows us to search within "Today", "Last Week" or "Last Month". We can also specify an exact date range.

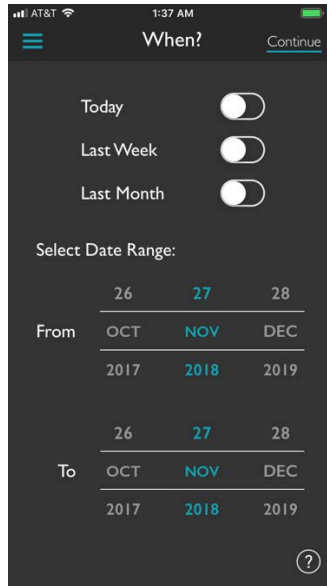


Figure 19

NOTE: "From" will start at 00:00 and "To" at 23:59.

Where Filter Screen

In the next filter we are allowed to filter based on location. This filter will just work on photos and videos if the phone has location services enabled. Also, there will be a prompt asking for permission to use location, we will say "Always allow".

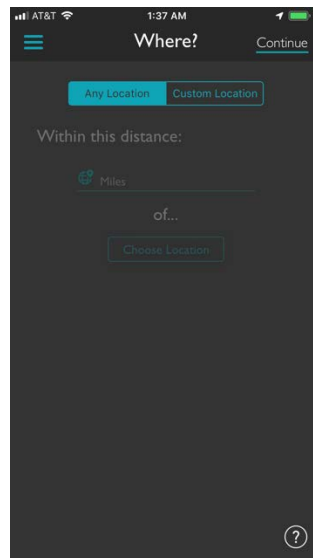


Figure 20

What Filter Screen

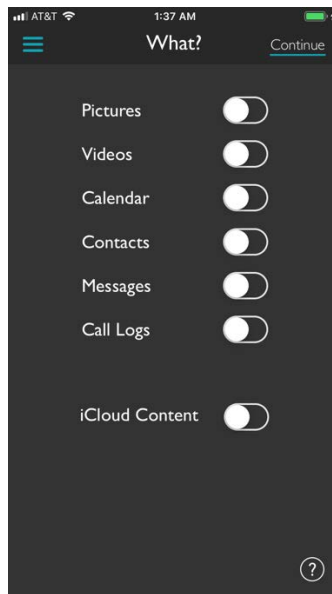


Figure 21

This screen will allow us to select the kind of evidence we are trying to extract. Currently we support the following:

- Pictures.
- Videos.
- Calendar.
- Contacts.
- Messages (Requires backup enabled).
- Call Logs (Requires backup enabled).

If we try to enable messages or call logs but we didn't select them in the TDES Manager, then it won't let us extract these resources.

Machine Learning Image Filtering

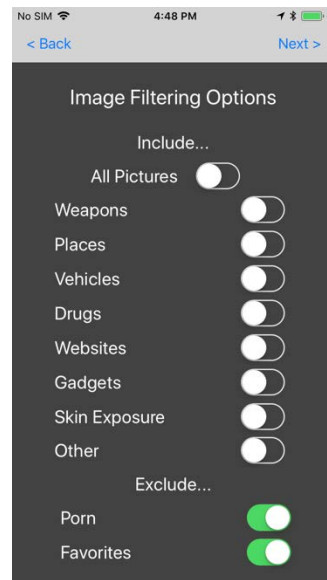


Figure 22

These filters will allow us to perform filtering using machine learning models embedded within the application. At the top part we have a switch that allows us to include all the images "All pictures", if we don't want to use image filters we can enable this switch. If we want to retrieve specific images we can enable the switches. The following categories are currently supported:

- Weapons
- Places
- Vehicles
- Drugs
- Websites
- Gadgets
- Skin Exposure
- Other

At the very bottom we see that we can exclude Pornography, if we want to do so we have to enable the switch. "Favorites" exclude button is a feature that will be added in a future version of the app.

NOTE: Other category will retrieve items outside the categories previously described. This category is experimental.

Who Filter Screen

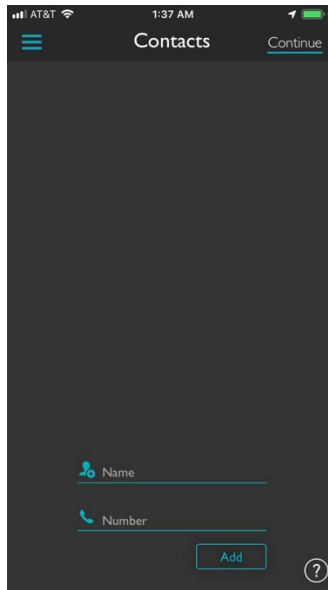


Figure 23

Here we can filter for a specific name or phone number. If these fields are left blank, then all the contacts will be retrieved. We can also decide to exclude that contact information, then the information that matches the fields will be excluded.

Consent Form

We will allow the application to use the front camera, this is for taking a picture of the person that allowed the consent form creation

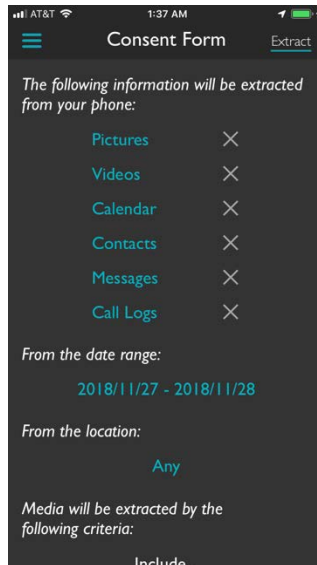


Figure 21

We can scroll down in this screen and see what the user allowed us to look at and extract

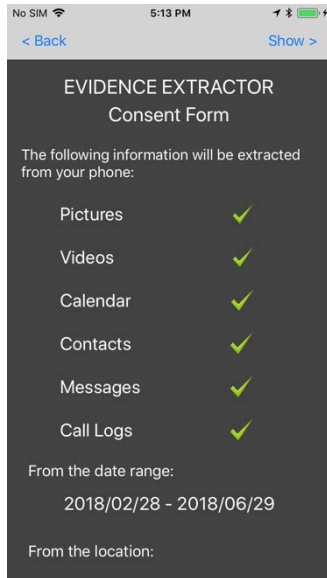


Figure 22

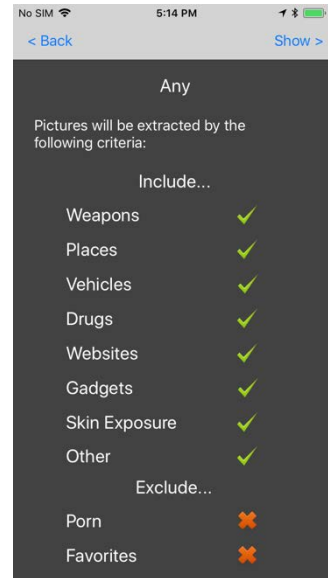


Figure 23

We will see two light blue buttons, one says, "User signature" and the other one "Investigator signature", these buttons allow us to sign our electronic consent form

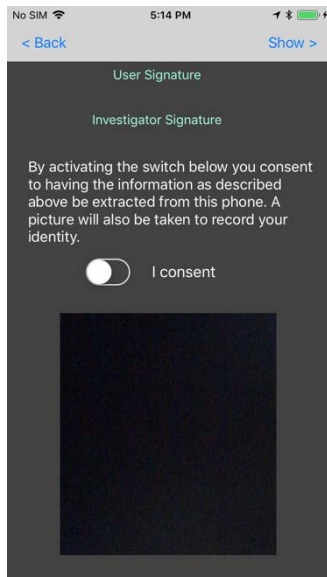


Figure 24



Figure 25

After we sign we toggle the switch "I consent". At this point a picture will be taken. The electronic consent form can be retrieved as a pdf later on.

Viewing and selecting evidence

Now we get to see the evidence that was filtered. In these set of views, we can select which evidence we want to export by bookmarking it. Bookmarking just requires that the item is tapped. If some items

in a subcategory are bookmarked, then just those items from that category will be exported. On the other hand, if there are no items bookmarked then **all items** will be exported. We can navigate through the different categories by pressing the bar buttons at the bottom of the screen. It is important to navigate to the other tabs to grant privileges to the app, once these privileges are given the prompts won't show up anymore.

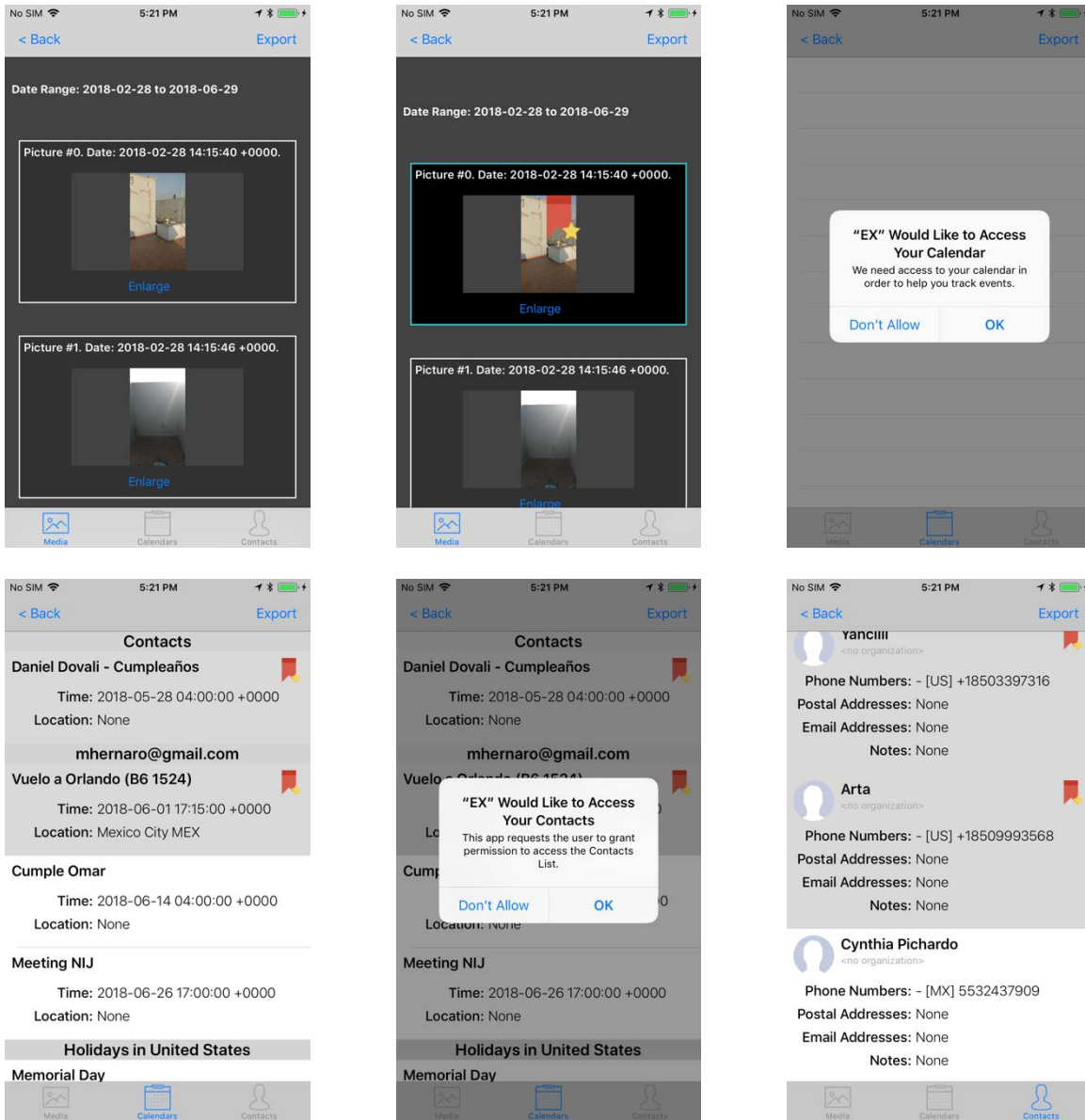


Figure 29

Exporting evidence

After selecting the evidence, we will export it with the "Export" button. We will see a new screen with a message that displays "Uploading".

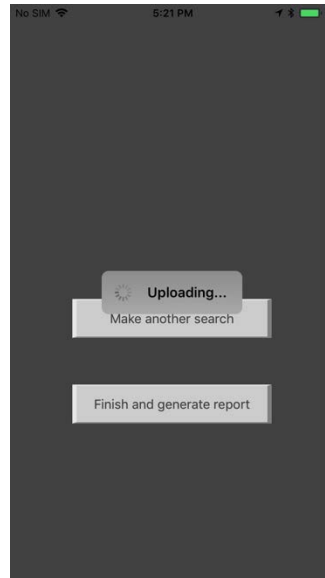


Figure 26

When data is successfully sent we will have the following screen

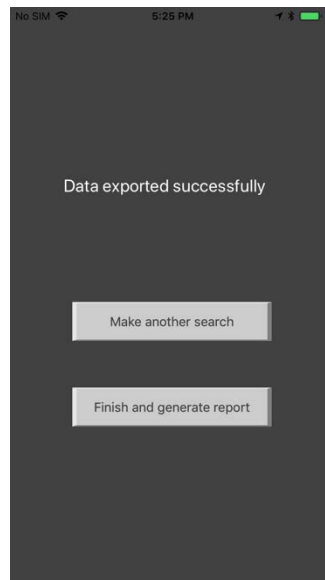


Figure 27

From this point on we can "Make another search", this will take us to the "When" screen, allowing us to perform another filtering. If we are done, we can tap on "Finish and generate report". A final screen will be displayed, asking us to exit and uninstall the app.

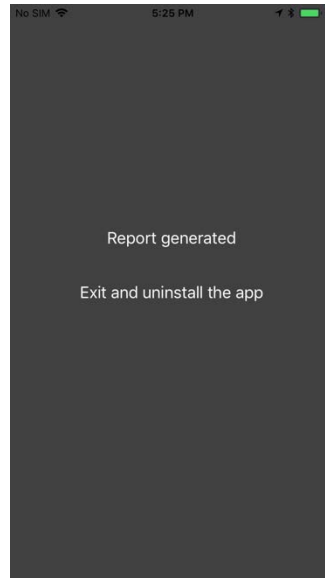


Figure 28

Going back to the TDES Manager

After the export is done we will receive a message on the TDES manager telling us that the report was successfully generated

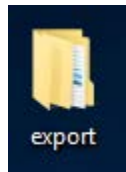


Figure 29

We can go back to the desktop in the computer and then double click the "export" folder

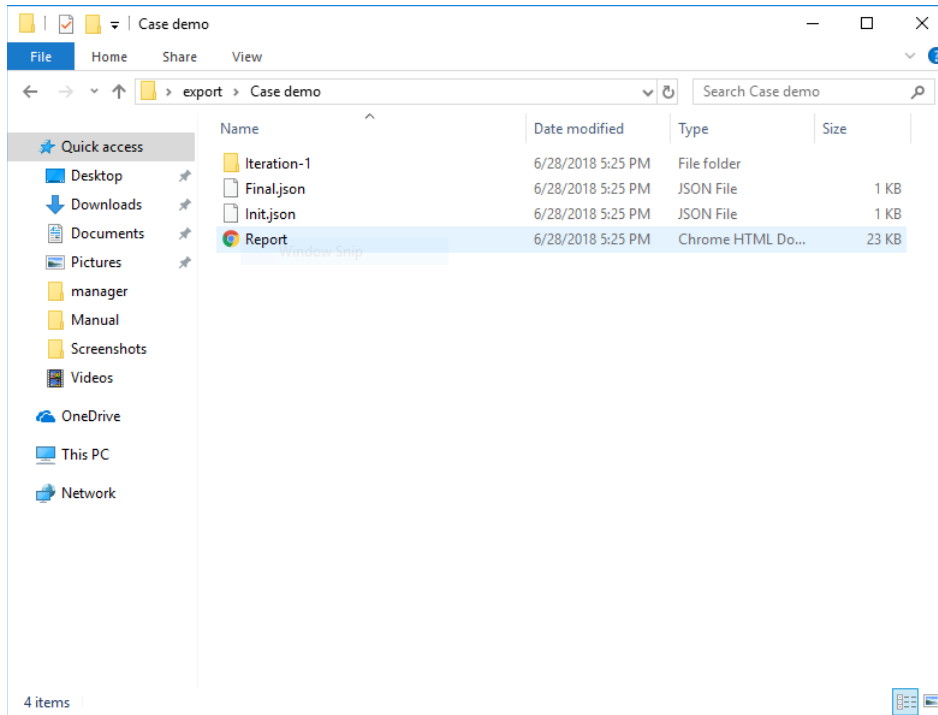


Figure 30

As we mentioned before, in here we will have the collection of all our cases. By navigating to one of them we can see that there are some json Files and a Report.html file. This file has all the information and evidence related to the case. Within Iteration-n (where n is each iteration) we have access to the evidence acquired per iteration. Similarly, as the per case basis we have an ltn_Report.html with holds the information related to this particular evidence. Inside the folder "ConsentForm" we have access to the user and investigator signatures, as well to the digital consent form generated by the App.

TDES-MI

Deployment and User Manuals

E-CIT Lab

Florida State University

Table of Contents

TDES-MI	0
TDES CLOUD BACKEND SERVER.....	2
INTRODUCTION	2
INITIAL SETUP	2
COPY OR DOWNLOAD THE PROJECT	2
COMPILE THE PROJECT	2
INSTALL THE DEPENDENCIES.....	2
INSTALL THE PROJECT (APACHE AND NGINX).....	2
TEST THE PROJECT	2
TDES PANEL (DEPLOYMENT)	3
INTRODUCTION	3
INITIAL SETUP	3
COPY OR DOWNLOAD THE PROJECT	3
COMPILE THE PROJECT	3
INSTALL THE DEPENDENCIES.....	3
INSTALL THE PROJECT (APACHE AND NGINX).....	3
TEST THE PROJECT	3
ENTRY POINT	4
MAIN VIEW	4
CASE LIST	4
CREATE/EDIT CASE.....	4
CASE DETAILS.....	5
TDES-MI ANDROID APP.....	6
INTRODUCTION	6
INITIAL SETUP	6
<i>Enable Installing "Unknown Apps" on Android Nougat</i>	<i>6</i>
<i>Enable Installing "Unknown Apps" on Android Oreo.....</i>	<i>6</i>
ENABLING USB DEBUGGING MODE ON ANDROID	7
CONNECTING THE DEVICE	8
INSTALLING THE APP.....	8
USING THE APP	10

TDES Cloud Backend Server

Introduction

This guide explains how the TDES Panel is to be setup in order to allow investigators work with cases and the evidence the witnesses upload through the TDES MI app.

Initial setup

- `npm`: The `npm` command has to be installed in order to compile the project. It can be downloaded from <http://npmjs.org>.
- HTTP server: In order for the web application to work, an HTTP server has to be installed. It can be either Apache (<https://httpd.apache.org/>) or NGINX (<https://www.nginx.com/>).

Copy or download the project

The project is located in the following path of the flash drive: `TDES mi/projects/TDES Panel/software`, or it can be downloaded from <https://juanpabloconde@bitbucket.org/juanpabloconde/tdes-panel.git>

Compile the project

Install the dependencies

Run the following command in order to install the required dependencies:

```
npm install
```

Once the project was downloaded, by using the command line, locate in the root directory of the project and run the following command:

```
npm run build
```

This will generate a `dist` folder, located under the root directory of the project.

Install the project (Apache and NGINX)

Replace the contents of the `/var/www` folder with the contents of the `dist` folder, generated in the previous step.

Test the project

By accessing <http://localhost> from the server where the project was installed, the panel should appear on the browser.

TDES Panel (deployment)

Introduction

This guide explains how the TDES Panel is to be setup in order to allow investigators work with cases and the evidence the witnesses upload through the TDES MI app.

Initial setup

- `npm`: The `npm` command has to be installed in order to compile the project. It can be downloaded from <http://npmjs.org>.
- HTTP server: In order for the web application to work, an HTTP server has to be installed. It can be either Apache (<https://httpd.apache.org/>) or NGINX (<https://www.nginx.com/>).

Copy or download the project

The project is located in the following path of the flash drive: `TDES mi/projects/TDES Panel/software`, or it can be downloaded from <https://juanpabloconde@bitbucket.org/juanpabloconde/tdes-panel.git>

Compile the project

Install the dependencies

Run the following command in order to install the required dependencies:

```
npm install
```

Once the project was downloaded, by using the command line, locate in the root directory of the project and run the following command:

```
npm run build
```

This will generate a `dist` folder, located under the root directory of the project.

Install the project (Apache and NGINX)

Replace the contents of the `/var/www` folder with the contents of the `dist` folder, generated in the previous step.

Test the project

By accessing <http://localhost> from the server where the project was installed, the panel should appear on the browser.

Entry Point

In order to access the investigator panel, open a web browser and type the following address:
<http://dna.ecit.fsu.edu/>.

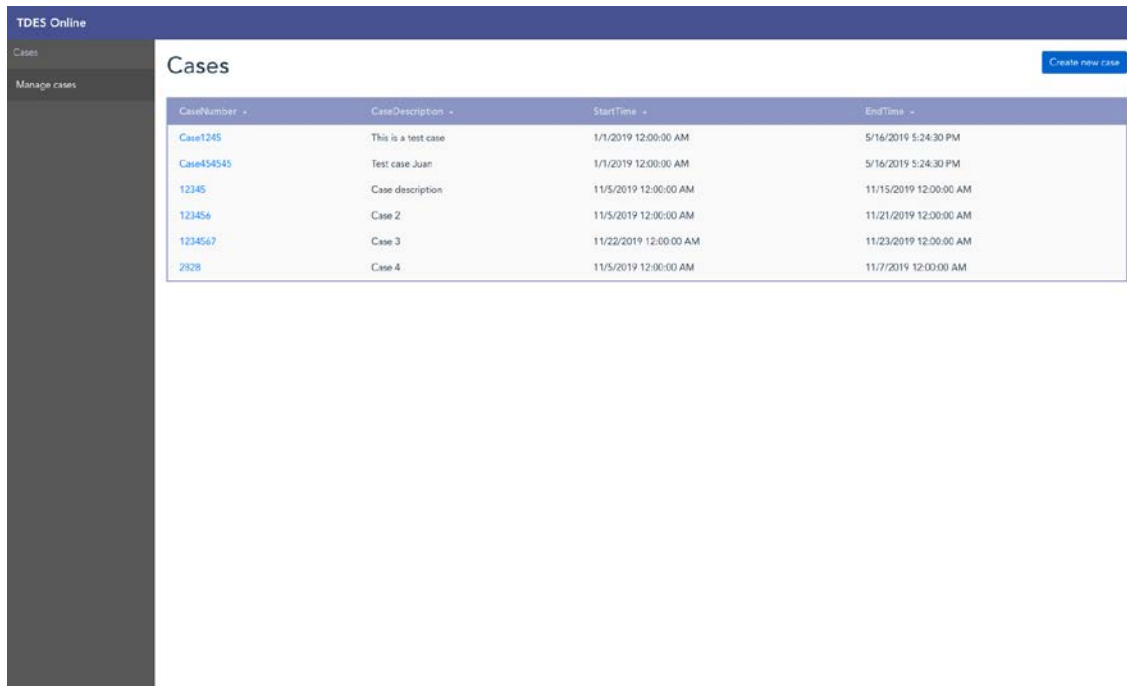
Note: If the system is installed in another server, just use the proper address.

Main view

The main view contains only the toolbar with only one option (manage cases).

Case list

Once the "manage cases" option is selected, the system will show the list of the cases available in the system.

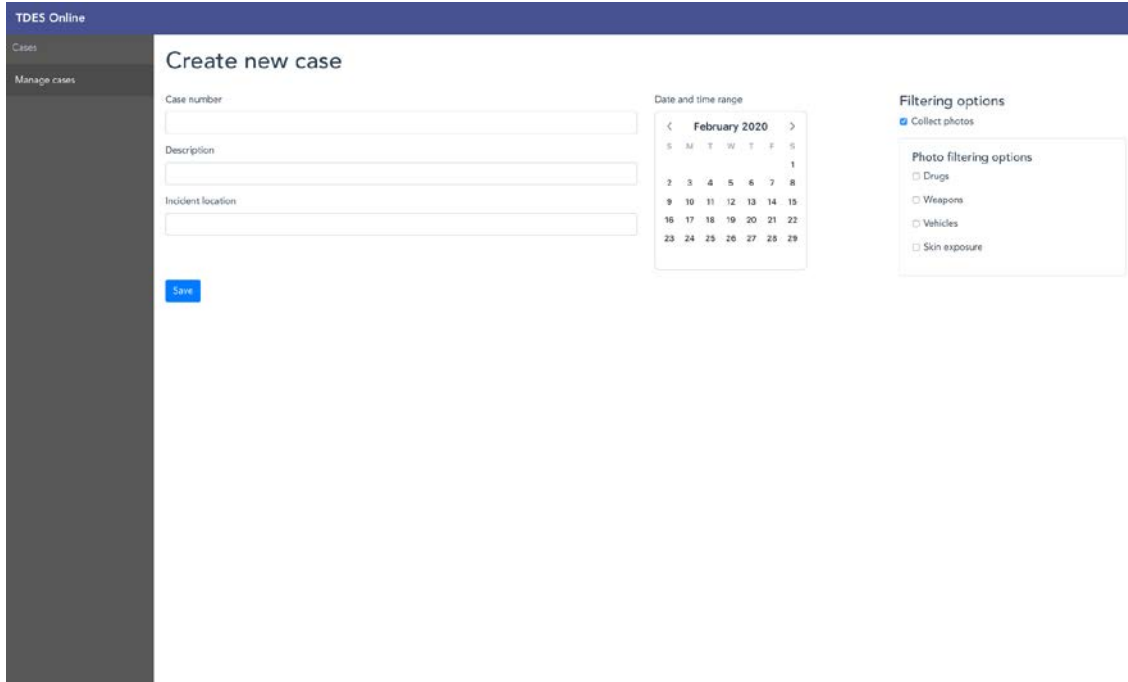


CaseNumber	CaseDescription	StartTime	EndTime
Case1245	This is a test case	1/1/2019 12:00:00 AM	5/16/2019 5:24:30 PM
Case454545	Test case Juan	1/1/2019 12:00:00 AM	5/16/2019 5:24:30 PM
12345	Case description	11/5/2019 12:00:00 AM	11/15/2019 12:00:00 AM
123456	Case 2	11/5/2019 12:00:00 AM	11/21/2019 12:00:00 AM
1234567	Case 3	11/22/2019 12:00:00 AM	11/23/2019 12:00:00 AM
2328	Case 4	11/5/2019 12:00:00 AM	11/7/2019 12:00:00 AM

The user can create new cases by clicking on the button "Create new case", or select a case from the list, which will lead to the case detail view.

Create/Edit case

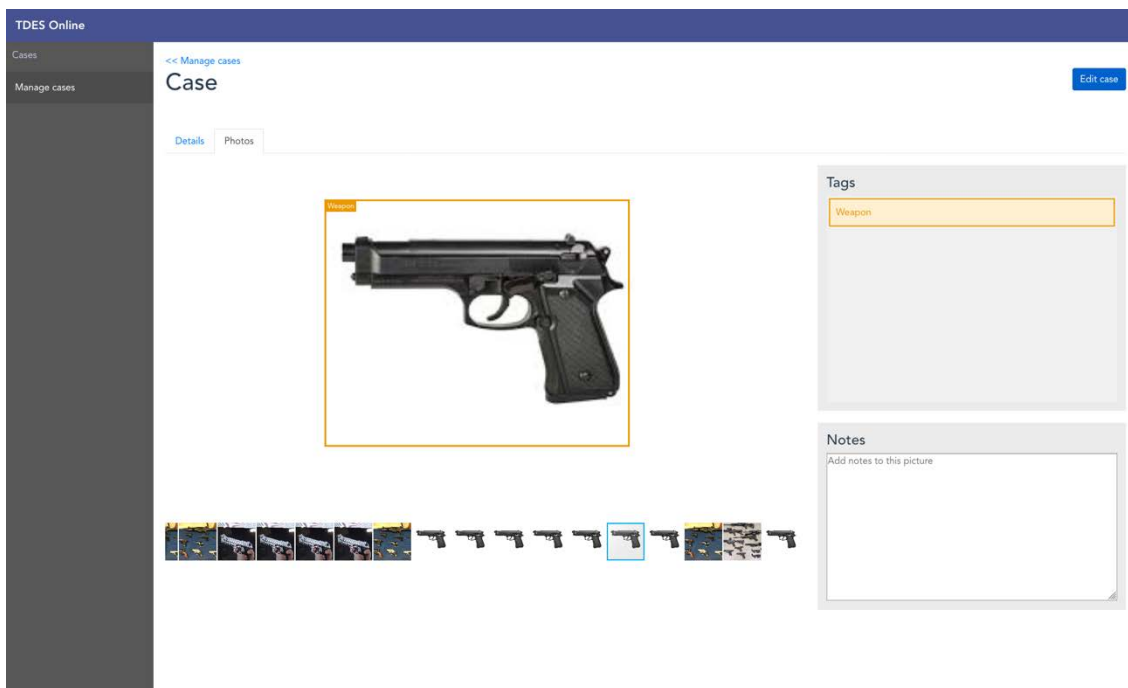
This view allows the user to create a case on the system or modify an existing one. Case-related attributes are to be set, along with the filters that the investigator wants to apply to the data the witnesses are going to upload.



Case details

The case detail view will show the information the witnesses have uploaded to the system. Currently the system supports only photo uploading.

User will also visualize the automatic image tagging on photos in which elements that were set in the case filter were detected on the pictures. The user can navigate through all the photos and visualize the different tags associated with them.



TDES-MI Android App

Introduction

This manual is written to instruct users to properly use the TDES MI app on Android devices. This is an alpha version, which means that minor bugs may occur, requiring the user to perform certain tasks to reset the application.

Initial setup

The TDESMI.apk file to be installed on the Android phone will be provided. Since Android considers any application outside of Google Play store an "unknown app", you need to tell your phone to allow you to install unknown apps.

NOTE: In case the address of the backend server changes, the new address will have to be set on the project and a new .apk file will have to be generated. This is done in file `com/android/dataextraction/util/network/NetworkConstants.java`, by changing the `BASE_URL` constant accordingly.

Enable Installing "Unknown Apps" on Android Nougat

1. Go into **Settings**
2. Tap **Security** (or **Lock Screen and Security**).
3. Scroll down to the **Device Administration** section and enable **Unknown Sources**.

Enable Installing "Unknown Apps" on Android Oreo

1. Go to **Settings**
2. Then **Apps and Notifications**

3. Select **Install Unknown Apps** (or **Install Other Apps**)

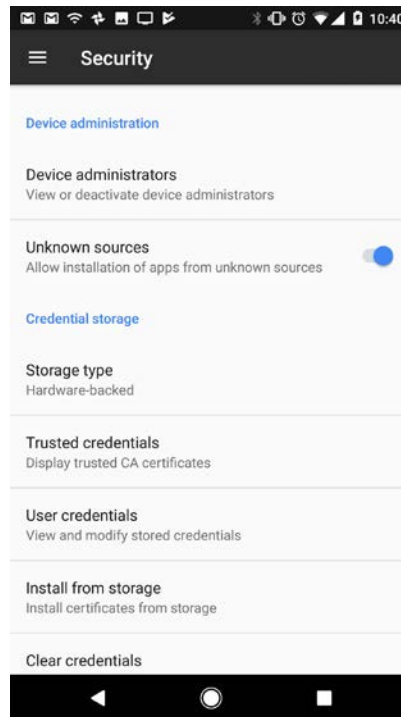


Figure 1

Enabling USB Debugging Mode on Android

Prior to connecting the Android device to the computer, Android device is required to be in the USB debugging mode ADB. The ways to enable USB Debugging mode varies from one Android version to another; here we describe the general procedures.

Navigate to **Settings > Developer** options configuration screen on the Android device

1. If Developer options is not enabled, navigate to **Settings > About** device and tap on the **Build number** seven times. Return to previous screen, **Developer options** should be visible now.

2. Click to open the **Developer options** screen, select “ON” mode and enable the USB debugging.

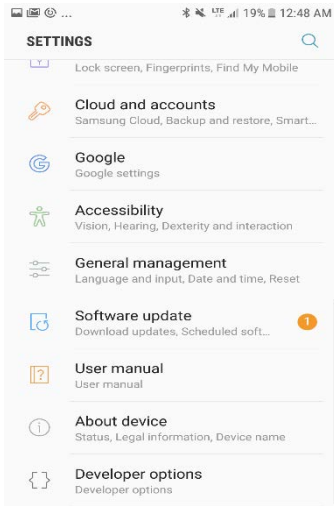


Figure 2

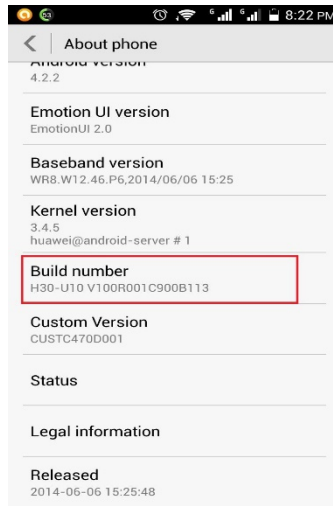


Figure 3

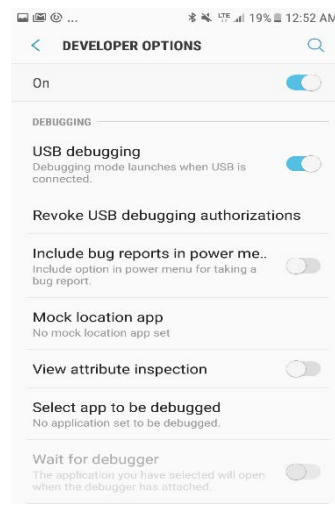


Figure 4

Connecting the device

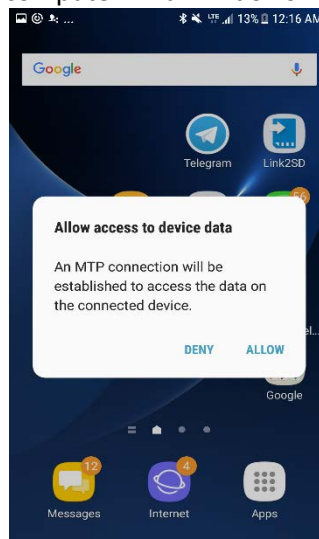
Connect the Android device to the computer with a USB cable. The phone will prompt to allow access to device data, select <ALLOW>.

Figure 5

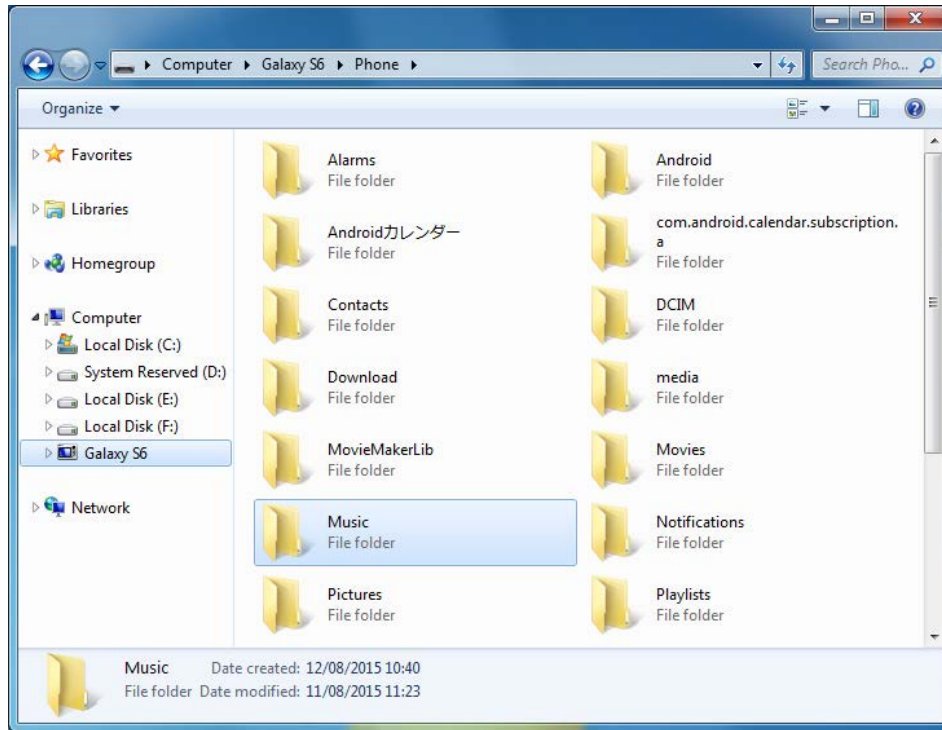
Installing the App

Once the device has been setup and connected to the computer, it is now time to install the app. In order to do that, the device has to be connected as a "media device".

Locate the folder of the Android device on the computer. In a Windows PC it should be located under



My Computer or Computer. Once located, use the APK file provided and drop that file inside the device's folder (or a folder of your choice inside it).



Now that the APK file has been copied to the Android device, locate it by browsing on it, then tap on it. The system will ask to install the app contained in the APK file. Select "install".

Using the App

1. After the app is opened, the app will prompt the user to grant consent and provide to access specified data.

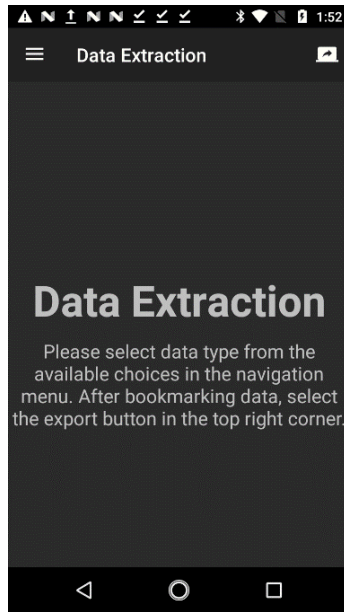


Figure 6

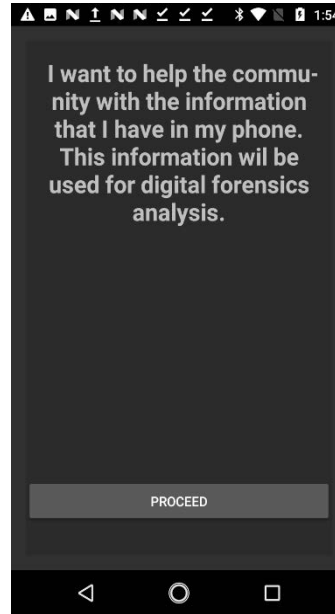


Figure 7

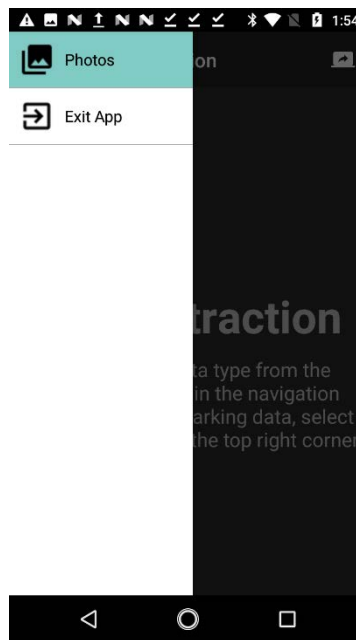


Figure 8

2. After the data types are selected from the drop-down menu on the left-hand side, App will display the data types. For this initial version, photos is the only type of data enabled.

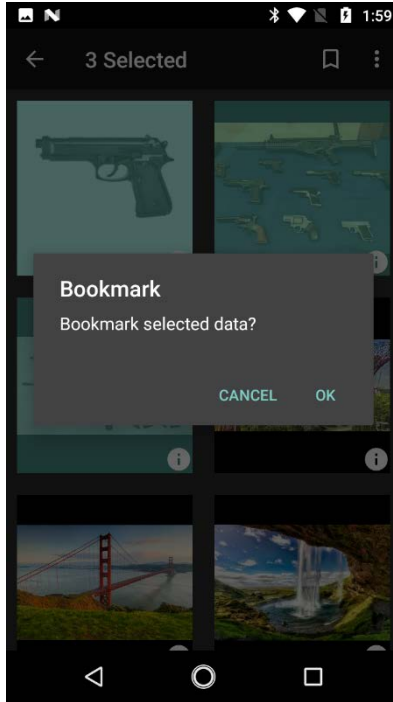


Figure 9

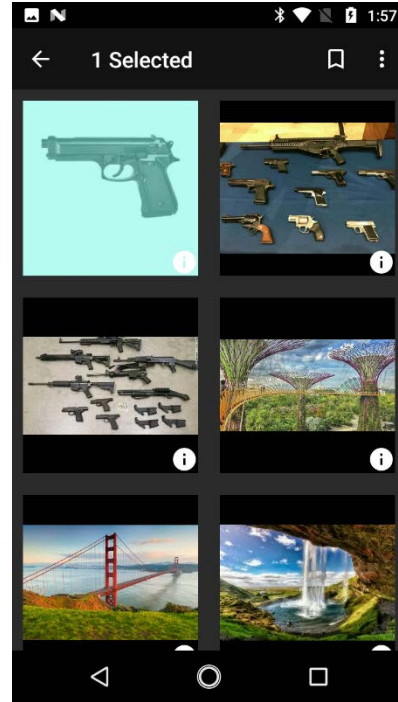


Figure 10

3. Once the data type option is selected the app will display all the filtered data type. In this screen, the user can narrow down the filtering by bookmarking a specific data to be exported.
4. Once all the data has been bookmarked, the user should tap the "Export" button on the "Data Extraction" screen. This will upload all the selected data to the server and relate it to the corresponding case (created by the investigator on the panel).

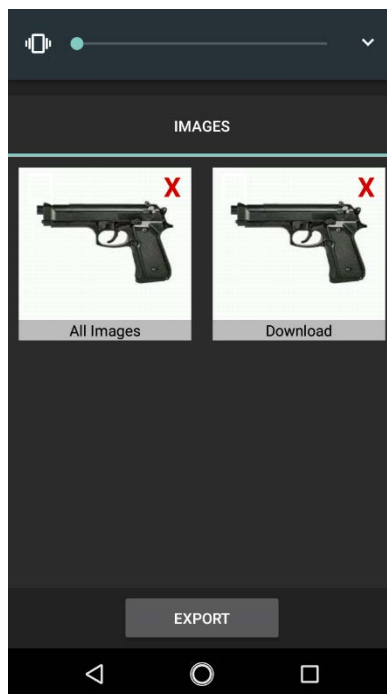
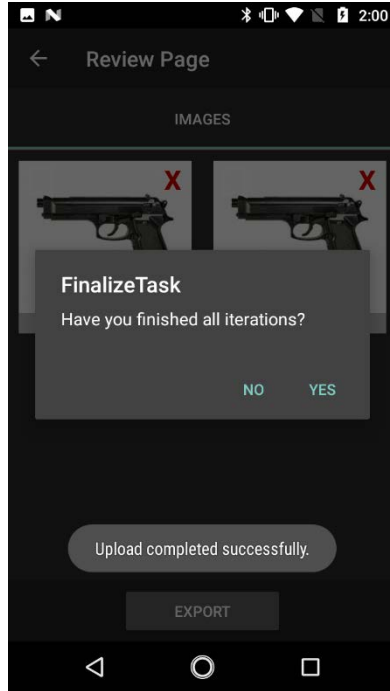


Figure 11

5. Once export button is selected the user will get notifications such as “Upload completed successfully” and “Export Successful” to confirm the data has been uploaded successfully. The



user can quit the app.

Figure 12

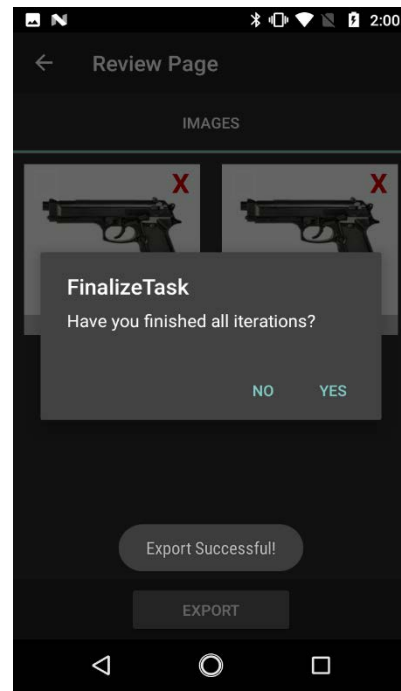


Figure 13