Novel Strategies to Fight Child Sexual Exploitation and Human Trafficking Crimes and Protect their Victims

H2020 – 101021801

www.heroes-fct.eu

D5.1 Mobile and web tool to identify, geolocate, and report possible victims

Authors: Budi Arief (UNIKENT)

Contributors: Allan McLeod, Virginia Franqueira, Julio Hernandez-Castro (UNIKENT)

Deliverable nature	Demonstrator (DEM)
Dissemination level	Public (PU)
Version	1.0
Date	30/11/2023

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021801

Document Information

Project Acronym	HEROES
Project Title	Novel Strategies to Fight Child Sexual Exploitation and Human Trafficking Crimes and Protect their Victims
Grant Agreement No.	101021801
Project URL	www.heroes-fct.eu
EU Project Officer	Elina Manova

Deliverable	**Number**	D5.1	**Title**	Mobile and web tool to identify, geolocate, and report possible victims		
Work Package	**Number**	WP5	**Title**	Multi-Sectoral and Multi-Disciplinary Strategies to Improve and Reinforce Prevention Programs		
Date of Delivery	**Contractual**		30/11/2023		**Actual**	12/01/2024
Status	Version 1.0				Final	
Nature	DEM		**Dissemination Level**		PU	

Responsible partner	**Name**	Budi Arief	**E-mail**	b.arief@kent.ac.uk
	Partner	UNIKENT	**Phone**	+44 (0)1227 816797
Contributing partners	Allan McLeod, Virginia Franqueira, Julio Hernandez-Castro (UNIKENT)			
Reviewers	Jesús Angel Alonso-López (UPM), Pablo Gallegos (IDENER)			
Security Approval	Julio Hernandez-Castro (UNIKENT)			

Abstract (for dissemination)

D5.1 outlines the development of an Android mobile application that will allow citizens to report incidents regarding potential Child Sexual Abuse (CSA) / Child Sexual Exploitation (CSE) and Trafficking of Human Beings (THB) to the relevant Law Enforcement Agency (LEA).

Keywords	Citizen reporting tool, mobile app, CSA, CSE, THB, LEA

Disclaimer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101021801. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the funding body.

Version History

Version	Date	Change Editor	Changes
0.1	12/11/2023	Budi Arief (UNIKENT)	Initial Draft report template
0.2	4/12/2023	Budi Arief (UNIKENT)	Integrating the functional and non-functional requirements
0.3	13/12/2023	Budi Arief (UNIKENT)	Updating Sections 1.3, 2, and 4.1; completing Sections 4 and 5
0.4	18/12/2023	Virginia Franqueira (UNIKENT)	Completing Sections 2.1 and 2.3
0.5	22/12/2023	Budi Arief (UNIKENT)	Completing the draft for internal review
0.6	26/12/2023	Jesús Alonso López (UCM)	Internal review
0.7	2/1/2024	Pablo Gallegos (IDENER RD)	Internal review
0.8	14/1/2024	Julio Hernandez-Castro (UNIKENT)	Security review
1.0	15/1/2024	Ana Lucila Sandoval Orozco (UCM)	Final version for submission

Table of Contents

List of Figures

List of Abbreviations

API	Application Programming Interface
APK	Android Package Kit
AWS	Amazon Web Services
CR	Citizen Reporting
CSA	Child Sexual Abuse
CSE	Child Sexual Exploitation
EC2	Elastic Compute Cloud
JAR	Java ARchive
JDK	Java Development Kit
LEA	Law Enforcement Agency
ORM	Object Relational Mapping
OS	Operating System
S3	Simple Storage Service
RDS	Relational Database Service
REST	REpresentational State Transfer
THB	Trafficking of Human Beings
UI	User Interface

Executive Summary

Deliverable D5.1 “Mobile and web tool to identify, geolocate, and report possible victims” presents the activities carried out within – and the outcomes achieved from – Task 5.1 “Development of a mobile app to identify, geolocate, and report possible THB and CSA/CSE victims”.

The aim of Task 5.1 is to develop a mobile application (smartphone app) that will allow end users (in this case, concerned citizens) to report to the Law Enforcement Agency (LEA) any incidents regarding potential cases of Child Sexual Abuse (CSA), Child Sexual Exploitation (CSE), or Trafficking of Human Beings (THB) that they might have witnessed. This app is complemented with a web front-end system that can be used by LEAs to view and respond to the incidents reported by citizens.

To achieve this aim, a complete set of software development life cycle – from design and implementation to testing, deployment and documentation – has been conducted. The resulting Citizen Reporting (CR) tool consists of two main components:

1. a mobile (smartphone) application that can be used by citizens to report suspected CSA/CSE and THB incidents to LEA, and
2. a web server application for hosting the database that is used for storing the reports, as well as for providing the web front-end (UI) for LEA to view and respond to the submitted citizen reports.

This deliverable provides an overview of the context of the CR tool within the HEROES project (including its objectives, motivation and other relevant background material); the functional and non-functional requirements of the CR tool; a detailed description of the tool itself (including its hardware and software requirements, system architecture and current deployment set-up, as well as the rationale behind design decisions and the key technologies chosen); along with the accompanying manuals (installation and user’s manuals).

In conclusion, Task 5.1 explores the feasibility of an implementation of a software solution that can be used by citizens to report CSA/CSE or THB incidents to LEA. This has been demonstrated by the construction of an easy to use proof-of-concept tool called “Citizen Reporting” tool, which is described in detail in this D5.1 deliverable.

1. Introduction

This document represents the deliverable coming out from Task 5.1 “Development of a mobile app to identify, geolocate, and report possible THB and CSA/CSE victims”, as part of WP5 “Multi-sectoral and multi-disciplinary strategies to improve and reinforce prevention programs” of the HEROES project [1].

The nature of D5.1 is a Demonstrator (DEM) deliverable, as it involves the creation of a software tool that can be used to showcase the practical and real-world improvements that come out of the HEROES project. The rest of this section provides an overview of this software tool, including its objective, motivation and the tool’s context in relation to the rest of the HEROES project.

1.1. Objective: What the tool does

The aim of Task 5.1 is to develop a mobile application (smartphone app) that will allow end users (in this case, concerned citizens) to report to the LEA any incidents regarding potential CSA/CSE and THB cases that they might have witnessed. This app is complemented with a web front-end system that can be used by LEAs to view and respond to the incidents reported by citizens.

End users (citizens) will be able to log in to the application using their credentials (email address and password)[1]. Once an end user has logged in, they will be able to create a new report and provide all the necessary information to LEA regarding the respective incident.

The application is designed to be very easy to use, with clear and simple navigation features, accompanied by suitable explanatory textual description about the actions that they can perform at each stage of the navigation. In addition, another aim of Task 5.1 involves the provision of first-hand information to LEAs, in order to proceed as soon as possible with their investigation of particular cases.

The application collects relevant information from the end user, including the nature of the incident (either CSA/CSE or THB), a textual description of the incident, the date and time of the incident, the type of the incident (either physical or online), the location of the incident, as well as a photo evidence (optional). After the end user confirms that they would like to submit the report, all of these pieces of information will be sent to a web server, from which a web front-end is available for the LEA to view and manage the submitted reports.

The application should support more than one language, at least the languages of the countries involved in the HEROES project (English, Spanish, Greek, Bulgarian). For a proof-of-concept demonstration, the application currently only supports English language. But the application has been designed with “a modular approach” in mind, which will allow for various language packages to be added easily and seamlessly later on.

1.2. Motivation: Why it is necessary

The motivation behind this Citizen Reporting app is to win precious time in the processing and investigation of potential CSA/CSE and THB cases. Users are citizens who witnessed or heard about possible CSA/CSE and/or THB incidents, who then try to help by reporting as much information about the incident as possible to the appropriate LEA.

The reporting is done via this app, by answering specific questions such as the date, time, location, description of crime scene, what happened, etc. (to prevent the need to take a photo, which might be difficult or dangerous to do in certain circumstances). The report is then sent via a dedicated web server to their local LEA, who will assess the submitted information and perform a triage to decide on the best action to follow.

On the LEA side, there is a complementary web-based front-end that will allow officers to access the stored information provided by the citizens (i.e. the reports), as well as their contact details in case more information

[1] As a proof-of-concept, the registration of end users is handled manually and separately for now – the main focus here is to show the features that allow for the end users to report any potential CSA/CSE and THB incidents, and to provide any pertinent information, such as the time of the incident, the location, as well as – where possible – any evidence such as photographs of the incident.

or a follow up is needed.

1.3. The tool in the context of the Project

The Citizen Reporting tool consists of two front-ends: (i) a mobile (smartphone) app for citizens, and (ii) a web front-end (UI) for LEAs.

The CR app will be made freely available for the general public to download and use, to make it easier for citizens to report possible CSA/CSE or THB cases they might have witnessed. In particular, this app will be valuable for those who – due to the nature of their work – might be in contact with many potentially vulnerable people, such as care workers, hospital staff, or border control officers. The app will provide a simple and interactive explanation that will introduce the user to report to and collaborate with the LEA in identifying possible cases of child sexual exploitation and/or trafficking.

Through the web front-end of this CR tool, police officers will have first-hand information, and all the data generated will allow statistical analysis to locate hot spots (based on the CR app's geolocation functionality), which are linked to these incident locations. The information about geolocation (as well as timeframe of the reports) can assist LEAs when planning operations to help victims. For instance, these pieces of information will help LEAs to identify incident hot spots, so that actions to combat crime situations can be better planned and the response of the involved entities will be more effective in terms of solutions and speed of identifying and attending the victims.

2. Background

In the fight against crime, it is important to provide suitable support for crime reporting, in order to alert the LEA – who may not have the view of everything happening in their jurisdiction area – regarding the possibility of incidents taking place or have taken place. This will help LEA to respond to incidents quicker, and to manage their resources accordingly, based on the trends and indications provided in these crime reports.

In the past, crime reporting would be done in-person (i.e. citizens would need to come to their local police station to file a report), or via the phone. There are many barriers and challenges with having to report in-person. Firstly, the reporter might be reluctant to come to the police station (e.g., their background might make them at odds with the police). Secondly, this would require a lot of time and effort in the part of the reporter to make. Furthermore, the reports were likely to be stored in a "pen-and-paper" means, which made them cumbersome to process and manage efficiently.

The use of phone line to report crime (e.g., through a dedicated phone number such as 101, 112 or even 999) removed some of the barriers present in the in-person reporting. However, some citizens may still be reluctant to speak to the police, or some of the information might not be easy to convey (such as photo evidence of the incident).

Advances in the Internet technologies, along with the popularity of smartphones changed all that. Smartphones, with their built-in cameras and GPS, can make the process of reporting crime a lot quicker, easier, and enriched with suitable evidence and pertinent data. Furthermore, it is possible to make the process more anonymous (by not imparting or at least minimising the amount of information that can be used to identify the reporter), which potentially can address some of the reluctance that citizens may have in dealing with the police.

2.1. Brief state of the art

"Crime reporting apps" is one of six categories of crime prevention applications, according to the typology by Wood et al. [2]. Others are more related to personal safety (e.g., panic features) and informing communities of crimes and dangers in a local area. It is part of a broader trend of "community policing" [3].

Allowing ordinary citizens to report witnessed incidents to the police digitally has been recognised as a more convenient, effective and efficient way of reporting, compared to other forms such as in-person or phone calls [3, 4]. On the other hand, it allows a quicker response and potentially enhances productivity by the police in tackling crimes [5]. It also serves the important purpose of collecting evidence from eyewitnesses close to an incident, therefore, increasing the chances of accurate reporting before memory fades [6]. However, empirical evidence has indicated that citizens engagement in such reporting is proportional to the level of trust that the public holds in law enforcement of their country [7].

Digital crime reporting directly to law enforcement seems to be a trend. For example, Tip Submit[2] is used in New Orleans/US, AlertCops[3] is used in Spain, Hawk Eye[4], and Reporty[5] has been rolled out in Nice/France. A number of publications proposed crime reporting apps for different regions such as Saudi Arabia [8], India [9], and across Africa [4, 10, 11].

Although most crime reporting apps are used for several types of crime, there are some designed for specific ones. Roshan et al. [12] proposed an app for Android to allow reporting of suspected human trafficking incidents to authorities in countries with high rates of such crime, such as Brazil, Bangladesh, Haiti, Pakistan, India, Sri Lanka, Nepal, Uganda, and Ghana. In this case, the user can select to stay anonymous. The authors plan to incorporate "a filter... for screening of junk information" before submission, but no details are given. Chinoko et al. (2021) [13] proposed an Android app for reporting child sexual abuse. However, they provided very few details to allow any meaningful evaluation and comparison with the HEROES app.

[2] https://new.tipsubmit.com/en/help/about
[3] https://alertcops.ses.mir.es/publico/alertcops/en/
[4] https://www.thecable.ng/meet-hawk-eye-new-app-helping-nigerians-report-crime
[5] https://www.straitstimes.com/world/europe/french-city-rolls-out-app-for-reporting-crime

Some apps and online crime reporting resources allow the alternative of submitting a basic level of incident information anonymously (e.g., Tip Submit). However, especially in the case of reporting child sexual abuse, this may give the opportunity for misleading/false details or even framing of an individual. Charitou et al. [3] discussed the aspect of anonymity in ICT-based community policing applications stating that revelation of personal data should be justified. The HEROES app leaves it up to citizens the decision about level of evidence provided (such as using the phone's gallery or not). In terms of personal data collected from the citizen reporting an incident, it complies with the precedent of the AlertCop app in use in Spain.

2.2. Key technologies

There are three key generic technologies being used for this deliverable:

- The Internet: this provides the communication backbone for the whole Citizen Reporting (CR) tool.
- Smartphones (mobile devices), along with the application (or "app" in short) ecosystem that are closely associated with smartphones. In this project, Android platform has been chosen for the proof-of-concept, due to the more widespread use of Android phones (compared to the iPhone), as well as the more open development environment, which allows for quicker experimentation and testing of the developed app.
- Web services: they serve as the back-end infrastructure for receiving and storing the reports submitted from the smartphones, as well as providing the front-end for the LEA to view and manage the reports.

More specifically, two groups of software technologies are used:

1. *Key mobile application technologies*: The Android application is a Xojo mobile framework[6] based application specifically targeted for Android phones. The application is packaged as a single Android Package Kit (APK) for deployment.
2. *Key server-side technologies*: The server-side application is Java Development Kit (JDK) 17 Spring Boot based and uses the following key components:
   - Liquibase[7] for schema management
   - Hibernate Object Relational Mapping (ORM)[8] for mapped PostgreSQL access
   - Logback[9] for application logging
   - Spring Security[10] for user-based management
   - Embedded Tomcat[11]

   The web UI is a Xojo web framework based application and packaged as a ZIP for server-side deployment. The server-side application is packaged as a single Java ARchive (JAR) file for deployment.

2.3. Comparison with other proposals

The following projects have been identified as related to this deliverable of HEROES.

The broader scope of the CITYCoP project (Citizen Interaction Technologies Yield Community Policing)[12], similar to this deliverable of the HEROES project, is "community policing". CITYCoP has developed a mobile application (SecureU[13]), and corresponding back-end, to streamline communication between citizens (representatives of a community) and law enforcement agencies to report risks and receive risk alerts. Although the

[6] https://www.xojo.com/products/mobile.php
[7] https://www.liquibase.com/
[8] https://hibernate.org/orm/
[9] https://logback.qos.ch/
[10] https://spring.io/projects/spring-security/
[11] https://tomcat.apache.org/
[12] https://cordis.europa.eu/project/id/653811
[13] https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bc25edc9&appId=PPGMS

technological meanings are the same, the CITYCoP app relates to the concept of "risk" rather than "crime". The former can include, e.g., anti-social behaviour (per-se not a criminal activity) or safety-related risks such as a fire, while the latter aims to report two specific types of criminal activities: child sexual abuse and human trafficking.

The TRILLION Project (TRusted, CItizen - LEA coILaboratIon over sOcial Networks)[14] also relates to "community policing". TRILLION has developed a mobile application[15] to allow *real-time, bi-directional collaboration* between law enforcement agencies, first responders and citizens, and to allow reporting of risks such as those related to accidents, disasters, crime and incidents. Its focus, therefore, is also not the same as this deliverable for HEROES although the technological means are the same.

The INSPEC2T (Inspiring CitizeNS Participation for Enhanced Community PoliCing AcTions)[16] is another project related to "community policing". INSPEC2T has developed two mobile applications: one for law enforcement and one for citizens. The former supported officers in community policing operations and in decision making. The latter had three main functionalities with bi-directional communication: support to community engagement (e.g., private and public), support to incident reporting and management (e.g., noise pollution, fire), and support security awareness (e.g., notification of incidents, safe map). The scope of the apps developed under this project are broader and allows real-time interaction, unlike this deliverable of the HEROES project.

All projects allow inclusion of evidence in citizens reporting to law enforcement agencies.

[14] https://cordis.europa.eu/project/id/653256
[15] https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bd51e7b9&appId=PPGMS
[16] https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b6b3efee&appId=PPGMS

3. Tool Description

The CR tool consists of two main components: (i) a mobile (smartphone) application that can be used by citizens to report CSA/CSE and THB incidents, and (ii) a web server application for hosting the database that is used for storing the reports, as well as for providing the web front-end for LEAs.

3.1. Hardware and Software requirements

3.1.1. Mobile application requirements

The mobile application is built to work on Android phones with the Operating System (OS) versions listed below:

- Android 14 (Release October 2023)
- Android 13
- Android 12
- Android 11
- Android 10 (potentially supported but not tested)

The baseline Android phone used for testing is a Google Pixel 6a running Android 14.

There are no specific hardware requirements other than the phones support the versions of the Android OS listed above.

3.1.2. Server application requirements

The server application and web UI server-side components are supported on these Linux versions:

- Amazon Linux AL2023
- Ubuntu Linux 20.04.6 LTS
- Ubuntu Linux 22.04.3 LTS

The server is currently deployed to Amazon Amazon Web Services (AWS) on three server instances, one Elastic Compute Cloud (EC2) instance each for the main server application and the web User Interface (UI) server-side components deployment, with an Relational Database Service (RDS) instance being used for the PostgreSQL database.

The EC2 instances are “t2.micro” instances with the following specification:

> 1 vCPU, 1.0 GiB RAM with 6 CPU credits per hour

The RDS instance is a “db.t3.micro” with the following specification:

> 1 core, 2 vCPUs, 1 GiB RAM and 12 CPU credits per hour

Although the server application and components are currently deployed to two instances, it is possible to host them on a single instance if required. It would also be possible to host all of the server-side elements, including the database, on a single physical server instead of AWS instances if needed.

Java 17 or higher is required on each EC2 instance and PostgreSQL 15.3 or better on the RDS instance.

3.2. System architecture

The main back-end system is a layered Java Spring Boot based application with three main layers providing the REpresentational State Transfer (REST) Application Programming Interface (API), Services and Repositories

(see Figure 1).

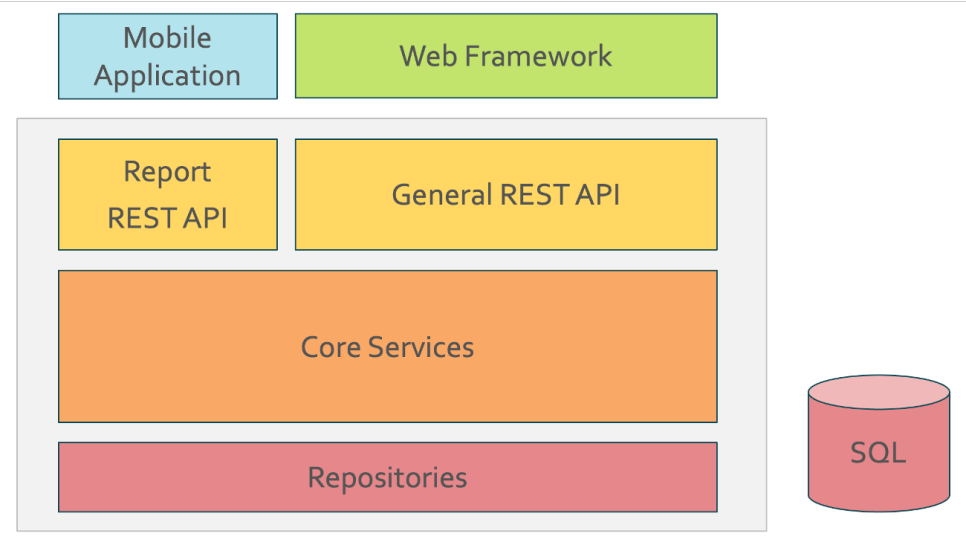

Figure 1: The architecture of the CR tool

The REST API is provided through Spring Controllers that delegate to business logic within the service layer, provided by Spring Services. All interactions with external storage are handled by the Repositories with only one currently required for PostgreSQL. An additional repository may be added later if it becomes necessary to store images as objects instead of as encoded data within the PostgreSQL database.

The Report REST API provides the mechanism for reporting from the mobile application and would be capable of being deployed separately from the General REST API should the need arise. This would allow better separation and therefore provide the ability to lock the General REST API for internal use by the web UI only. The Report REST API could still then be used more broadly but opportunity for attack would be reduced. This approach would also potentially allow for separate scalability.

The Web Framework shown in the diagram above provides a convenient mechanism for the pilot, as well as for demonstration and testing. It is built using the Xojo web framework and deployed separately to the main back-end application and communicates with it via the General REST API.

The Mobile Application is a Xojo mobile framework based. Internally, the application uses a workflow for the two types of report, with one for exploitation and one for trafficking. Both workflows are currently identical, but it would be possible to have them present different report screens to the user. Data for each workflow is already separate, with it currently being possible to have two reports open and in progress at the same time.

3.3. Details of the tools

3.3.1. Deployment

The CR tool is currently deployed using Amazon AWS on three server instances, one EC2 instance each for the main server application and the web UI server-side components deployment, with an RDS instance being used for the PostgreSQL database. The EC2 instances are currently running Amazon Linux AL2023 but deployment to Ubuntu 20.04.6 LTS or Ubuntu 22.04.3 LTS are also possible.

Although the server application and components are currently deployed to two instances, it is possible to host them on a single instance. It would also be possible to host all of the server-side elements, including the database, on a single physical server instead of AWS instances if needed.

3.3.2. Technology Choices

The main application is Java Spring Boot based, and this was chosen for speed of development, speed of execution and the ability to predictably scale services as required.

The following Spring Boot components are currently being used:

- Liquibase for Schema Management
- Hibernate ORM for mapped PostgreSQL Access
- Logback for System Logging
- Embedded Tomcat

Spring Security is also being used to secure the application, but this is not yet fully configured to support user-based security.

PostgreSQL is being used for data persistence, with the possibility of using Amazon Simple Storage Service (S3) like object stores for image storage. It has not been found necessary to separately store images, with image data currently being stored as encoded data within the main database.

As this is a Spring Boot application, much of the configuration for the application is exposed through the application properties and therefore easily configurable without the rebuilding of the system. Some configurable elements of the implementation have also been exposed through the application properties too.

The mobile application (for citizens to use) is a Xojo mobile framework based application that is specifically targeted to Android mobile phones. It would be possible to use portions of this project to also build for iOS devices, but the main focus has currently been for Android only. Some very limited tested has been performed on iOS for demonstration purposes.

A web-based front-end (for LEA to use) is currently being provided using the Xojo web framework, but this is very much being used for convenience and is not viewed as being the most appropriate method for providing a web UI.

4. Manuals

As a quick reminder, the Citizen Reporting tool consists of two front-ends: (i) a mobile (smartphone) app for citizens, and (ii) a web-based front-end for LEAs. This section provides two sets of manuals (installation manual and user's manual) for each of these two front-ends.

4.1. Installation manual

4.1.1. Installation of the mobile application for citizens

For now, the CR application is provided as an ".apk" (Android Package Kit) file, which needs to be copied to the Android device run as a side-loaded app.

In the future, it is envisaged that the CR app will be made available via standard app stores (including Google Play), with a lot more straightforward and usual installation process.

4.1.2. Installation of the server-side environment for LEA

4.1.2.1 Server application installation

Create the application directory structure in the "/opt" directory:

```
> cd /opt
> mkdir titan
> mkdir titan/bin
> mkdir titan/config
> mkdir titan/keys
> mkdir titan/log
```

Copy the appropriate version of the application package to the "./bin" directory and add a symbolic link:

```
> cp ~/titan-x.x.x-RELEASE.jar ./bin/
> ln -s ./bin/titan-x.x.x-RELEASE.jar ./bin/titan.jar
```

Note: replace the "x.x.x" with the appropriate version number.

Create a script to run the application in the "./bin" directory with the following content:

```
#!/bin/bash
java -jar ./bin/titan.jar
```

Create an "application.properties" configuration file in the "config" directory with the following content:

```
# Datasource configuration
spring.datasource.url=jdbc:postgresql://localhost:5432/titan
spring.datasource.username=titan
spring.datasource.password=[password]
spring.datasource.driverClassName=org.postgresql.Driver
spring.jpa.show-sql=false

# Schema versioning configuration
spring.liquibase.change-log=classpath:/db/changelog/db.changelog-master.xml

# Logging configuration
logging.file.path=/opt/titan/log/

# SSL configuration
# server.ssl.key-store-type=PKCS12
```

```
# server.ssl.key-store=/opt/titan/keys/keystore.p12
# server.ssl.key-store-password=[password]
# server.ssl.key-alias=[alias]

# Server configuration
server.port=8080

# Security configuration
spring.security.user.name=titan
spring.security.user.password=[password]
```

4.1.2.2 PostgreSQL database

Create a new database user role for the HEROES project:

```
> sudo su postgres
> createuser titan -P --interactive
```

You will be prompted to enter and confirm a password for the new role:

```
Enter password for new role:
Enter it again:
```

And then prompted to decide if the new user should be a superuser:

```
Shall the new role be a superuser? (y/n) y
```

Then use the PostgreSQL CLI to create a database for the project:

```
> psql
# CREATE DATABASE titan;
# quit
```

4.1.2.3 Database connection

Configure the database connection by modifying the "Datasource configuration" section in "application.properties". The resulting connection section should look as below but with the appropriate details:

```
# Datasource configuration
spring.datasource.url=jdbc:postgresql://localhost:5432/titan
spring.datasource.username=titan
spring.datasource.password=[password]
spring.datasource.driverClassName=org.postgresql.Driver
spring.jpa.show-sql=false
```

Replace the password shown above in square brackets with the one entered when creating the database.

4.1.2.4 SSL configuration

A certificate and private key will be required in order to configure the application to run with secure communication. Once you have a certificate, create the keystore as follows:

```
> openssl pkcs12 -export -in [certificate file] -inkey [private key file]
    -out ./keys/keystore.p12
```

Replace the certificate and private key filenames in the above as appropriate.

Once a keystore has been created, it can be configured for use by modifying the "SSL configuration" section in "application.properties" as follows:


```
# SSL configuration
server.ssl.key-store-type=PKCS12
server.ssl.key-store=/opt/titan/keys/keystore.p12
server.ssl.key-store-password=[password]
server.ssl.key-alias=[alias]

# Server configuration
server.port=8443
```

Replace the password and alias with the those used when creating the keystore.

4.1.2.5 Web UI installation

Create the application directory structure in the "/opt" directory:

```
> cd /opt
> mkdir rhea
> mkdir rhea/bin
> mkdir rhea/log
```

Copy the appropriate version of the application package to the "./bin" directory and unzip it:

```
> cp ~/HeroesWeb-x64-x.x.x.zip ./bin/
> unzip HeroesWeb-x64-x.x.x.zip
```

Replace the "x.x.x" with the appropriate version number.

Create a script to run the application in the "./bin" directory with the following content:

```
#! /bin/bash
./bin/HeroesWeb --port=8080 --Logging=/opt/rhea/log/ 2>&1 | ts | tee -a
    /opt/rhea/log/rhea.log
```

4.2. User's manual

4.2.1. User's manual of the CR app for citizens

4.2.1.1 Running the Application for the First Time

When the application is run for the first time, the user will be presented with a welcome screen (which displays the HEROES project logo), asking them to follow the instructions on the next screens to setup the application (see Figure 2, left). The application version number is displayed at the bottom of this screen.

After that, the user will be presented with an opportunity to choose the language they prefer to use for the application (see Figure 2, middle). Please note that it is always possible to change the language choice later on, via the "Settings" menu (see Section 4.2.1.9).

Finally, the user will be requested to enter the username and password that have been supplied to them, and press "Login" to apply the changes (see Figure 2, right) to login properly to the application, before being able to use the rest of the application's features.

4.2.1.2 Home Screen

Upon a successful login, the user will be presented with the "Home" screen (see Figure 3).

This is the key starting point of the Citizen Reporting app. From this screen, users can start the reporting process for a crime or incident. Specifically, they can choose from one of the two types of incident: CSA/CSE

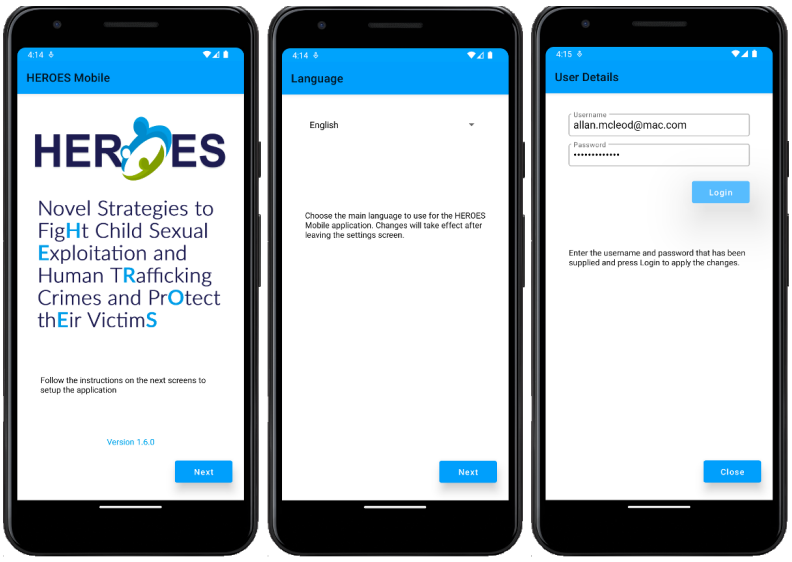

Figure 2: First Run: Title/Welcome Screen (left); Language Setup Screen (middle); User Setup (Login) Screen (right)

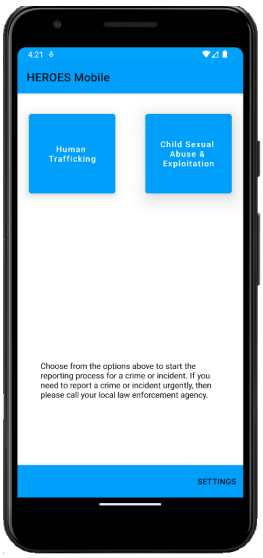

Figure 3: Home Screen

or THB. Please note that if a user needs to report a crime or incident urgently, they should call their local police number instead of using this app.

Furthermore, from this “Home” screen, users can adjust the settings of their application (please see Section 4.2.1.9 for further detail regarding this feature).

4.2.1.3 Reporting: Description Screen

Once the user has selected the type of incident (either CSA/CSE or THB), the app will present the user with a text box, within which they can enter a text description of the incident (see Figure 4, left). The button below this text box will only allow the user to move to the next screen once a textual description has been entered (see Figure 4, right).

4.2.1.4 Reporting: Incident Date and Time Screen

The next step in the incident reporting process is to choose the date and time of the incident using the top two buttons shown in Figure 5 (left). The preset buttons in the middle of Figure 5 (left) allow convenient dates to be set easily (Now, Yesterday, Last Week or Last Month).

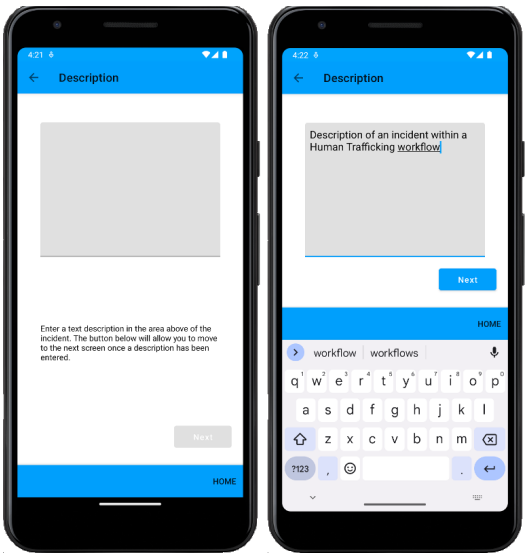

Figure 4: Entering the description of the incident

The user will be able to move to the next step using the “Next” button at the bottom of the screen, once both the date (see Figure 5, middle) and time (see Figure 5, right) have been chosen.

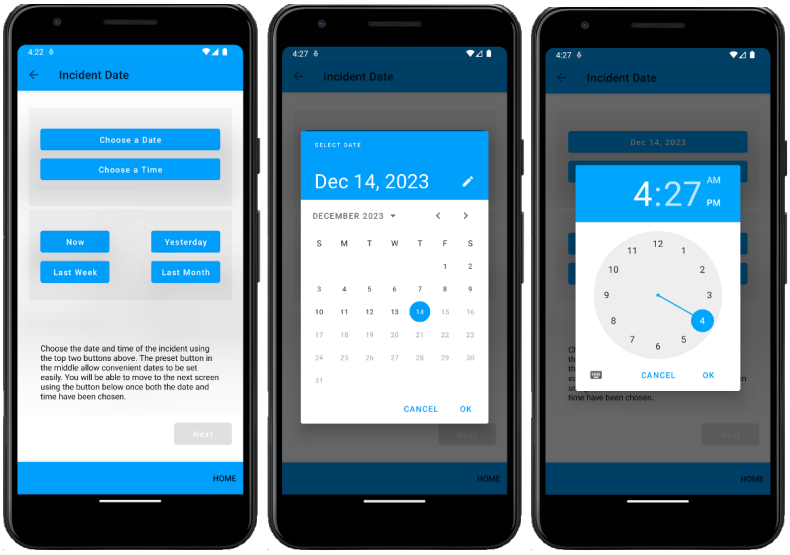

Figure 5: Entering the date and time of the incident

4.2.1.5 Reporting: Report Type Screen

The user can then choose if this report is about a physical incident that the user have witnessed (see Figure 6, left) or if this relates to an incident or material seen on a website (see Figure 6, right). The button at the bottom of the screen will allow the user to move to the next screen once a selection has been made.

4.2.1.6 Reporting: Incident Location Screen

In accordance to the report type, the user can then enter the location of the incident. This can be an address or a description of the location for a physical incident (see Figure 7, left) or a website/URL address (see Figure 7, right). The text field for the physical incident location – as well as the field below the website/URL address – can be used by the user to add further notes (such as why they thought they should report what they saw as an incident). The “Next” button will allow the user to move to the next screen once location information (either physical or online) has been entered. The user may choose to leave the notes area blank.

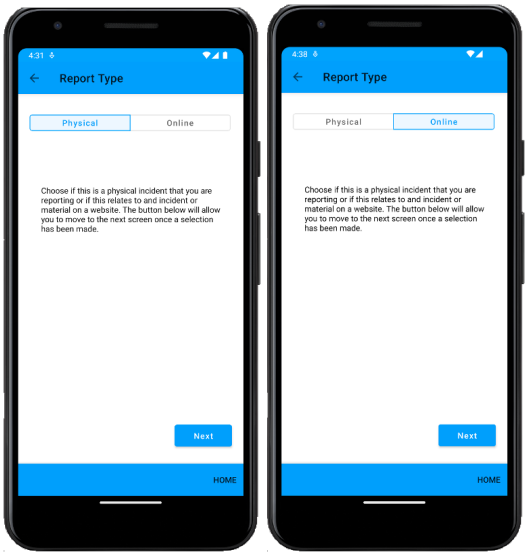

Figure 6: Choosing the report type – either physical (left) or online (right)

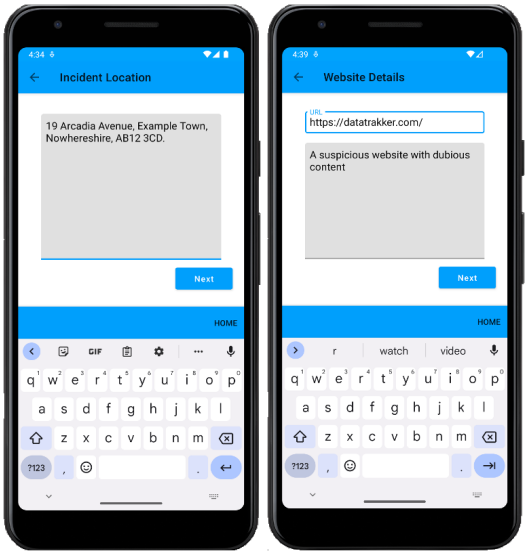

Figure 7: Entering information regarding the incident's location – either a physical location (left) or a website/URL address (right)

4.2.1.7 Reporting: Photo Evidence Screen

The user can then upload a photo evidence to support their incident report. They can choose an image from the gallery (see Figure 8, left), or take a picture with the phone's camera (see Figure 8, right). Adding a photo evidence is optional and the user may continue without doing this.

4.2.1.8 Reporting: Confirming and Submitting

Finally, the user can confirm and send the report by pressing the "Confirm Report" button shown in Figure 9 (left). This button will be enabled once the user's current location has been obtained (see Figure 9, middle). Once the report has been submitted, the reporting process is complete, and the user will be presented with an acknowledgement message, and they can then go back to the Home Screen (see Figure 9, right).

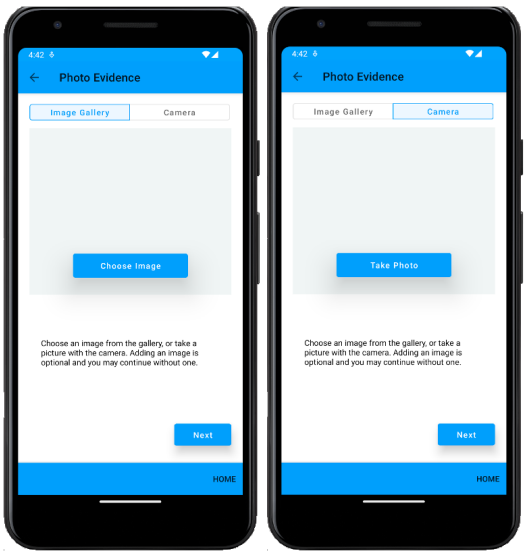

Figure 8: Adding photo evidence, either an existing image from the gallery (left), or taking a new photo using the phone's camera (right)

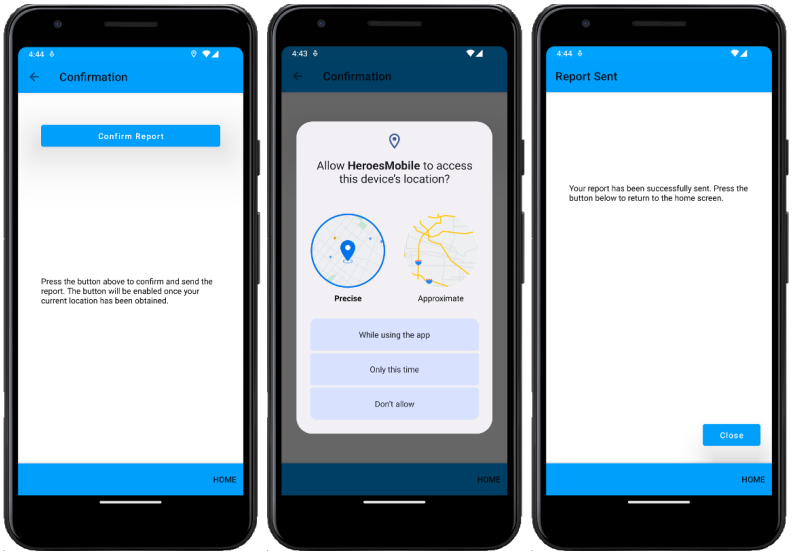

Figure 9: Confirming and submitting the incident report

4.2.1.9 Settings Screens

From the "Home" screen (see Section 4.2.1.2) users can find the "Settings" button at the bottom right hand corner (see Figure 3). Clicking this button will present the user with three tabs to allow them to change the settings related to the user entity, the language, and some special service features (see Figure 10, left, middle, and right, respectively):

- User: Enter the username and password that has been supplied and press Login to apply the changes.
- Language: Choose the main language to use for the HEROES Mobile application. Changes will take effect after leaving the settings screen.
- Service: Choose the service provider, enable demonstration mode and test the connection. Demonstration mode enables the application to behave as if a report has been submitted without sending it to the cloud service.

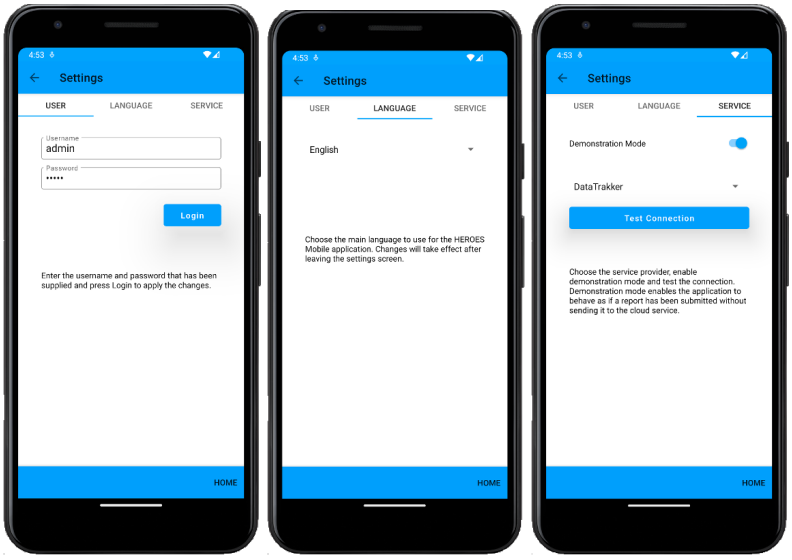

Figure 10: Settings Screens: user (left); language (middle); service (right)

4.2.2. User's manual of the web front-end for the LEA

LEA officer will be able to use the web front-end part of the CR tool to view the reports submitted by citizens, and to add notes as part of their task in responding to these reports.

The current implementation of this web front-end is pretty basic, mainly to serve as a proof-of-concept at this stage. Further refinement and improvement to the web front-end will be made after the first pilot has been completed (end of January 2024), leveraging the comments and feedback that will be received from the first pilot.

4.2.2.1 Viewing Citizen Reports

For now, the main feature that the LEA's web-based front-end provides is the ability to view CSA/CSE and THB reports submitted by citizens.

In Figure 11 (left), the LEA is presented with a list of all reports of CSA/CSE and THB incidents that have been submitted by concerned citizens.

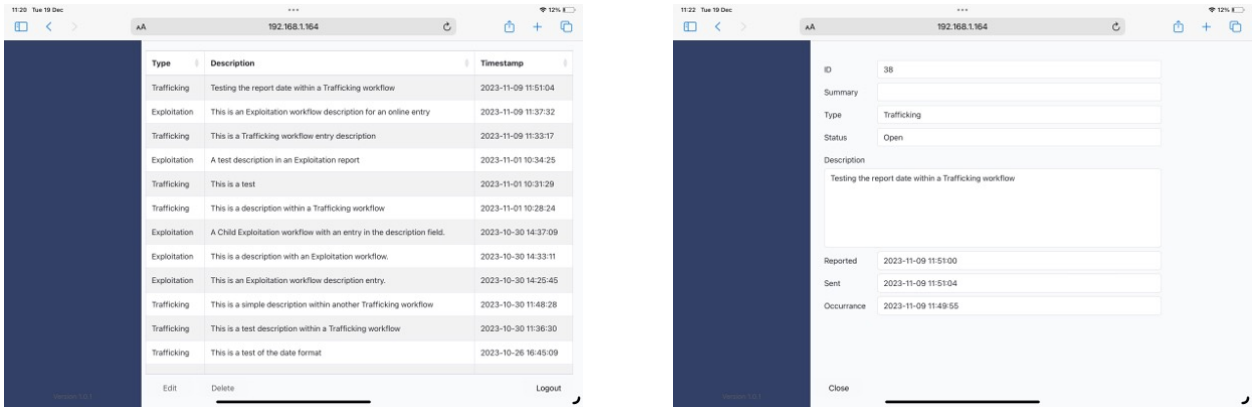

Figure 11: Viewing citizen reports: a list of all reported incidents (left); and a detailed view of one incident report (right)

Clicking one of the reports in the list will open up a detailed view of that report (see Figure 11, right). Information relevant to each reported incident (such as its description, the date/time, as well as location information and photo evidence, if any) is presented in this page. This page also provides an editable text field (called "Summary"), in which a LEA officer can add their notes or comments regarding this particular reported incident (e.g., if there is a priority to be made regarding this report).

4.2.2.2 Server-Side Logging

Finally, it is possible to show the raw data of the server activities, in order to view actions that have been performed on the LEA web-based front-end (for instance, to do an audit). An example of this server-side log information can be seen in Figure 12.

```
    : Started TitanApplication in 3.242 seconds (process running for 3.757)
2023-10-23T10:48:43.953+01:00  INFO 9538 --- [nio-8081-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
    : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-10-23T10:48:43.953+01:00  INFO 9538 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet
    : Initializing Servlet 'dispatcherServlet'
2023-10-23T10:48:43.954+01:00  INFO 9538 --- [nio-8081-exec-1] o.s.web.servlet.DispatcherServlet
    : Completed initialization in 1 ms
2023-10-23T10:48:44.022+01:00  INFO 9538 --- [nio-8081-exec-1] c.p.t.controller.CrimeReportControlle
r   : Submit a new crime report
Hibernate: insert into crime_report (description,timestamp) values (?,?)
2023-10-23T11:29:10.053+01:00  WARN 9538 --- [l-1 housekeeper] com.zaxxer.hikari.pool.HikariPool
    : HikariPool-1 - Retrograde clock change detected (housekeeper delta=29s727ms), soft-evicting co
nnections from pool.
2023-10-23T11:36:09.293+01:00  WARN 9538 --- [l-1 housekeeper] com.zaxxer.hikari.pool.HikariPool
    : HikariPool-1 - Thread starvation or clock leap detected (housekeeper delta=6m59s240ms).
2023-10-23T11:37:44.771+01:00  WARN 9538 --- [l-1 housekeeper] com.zaxxer.hikari.pool.HikariPool
    : HikariPool-1 - Thread starvation or clock leap detected (housekeeper delta=1m5s338ms).
2023-10-23T11:41:20.287+01:00  INFO 9538 --- [nio-8081-exec-3] c.p.t.controller.CrimeReportControlle
r   : List all crime reports
Hibernate: select c1_0.id,c1_0.description,c1_0.timestamp from crime_report c1_0 order by c1_0.times
tamp desc
2023-10-23T11:41:25.585+01:00  INFO 9538 --- [nio-8081-exec-4] c.p.t.controller.CrimeReportControlle
r   : List all crime reports
Hibernate: select c1_0.id,c1_0.description,c1_0.timestamp from crime_report c1_0 order by c1_0.times
tamp desc
2023-10-23T12:00:48.385+01:00  INFO 9538 --- [nio-8081-exec-6] c.p.t.controller.CrimeReportControlle
r   : Submit a new crime report
Hibernate: insert into crime_report (description,timestamp) values (?,?)
2023-10-23T13:22:48.591+01:00  INFO 9538 --- [nio-8081-exec-8] c.p.t.controller.CrimeReportControlle
r   : Submit a new crime report
Hibernate: insert into crime_report (description,timestamp) values (?,?)
2023-10-23T13:23:32.510+01:00  INFO 9538 --- [io-8081-exec-10] c.p.t.controller.CrimeReportControlle
r   : Submit a new crime report
Hibernate: insert into crime_report (description,timestamp) values (?,?)
2023-10-23T13:41:39.967+01:00  INFO 9538 --- [nio-8081-exec-2] c.p.t.controller.CrimeReportControlle
r   : Submit a new crime report
```

Figure 12: An example of the server-side log information

5. Conclusions

The main aim of the system described in this deliverable (D5.1) is to provide a mechanism for the reporting of (potential) CSA/CSE and THB by concerned citizens (who might have witnessed an incident), to their relevant/local LEA.

This is achieved through the use of a mobile application for Android phones that sends data to a back-end server-side system through a REST API. That same REST API also provides the ability to view and interact with the reports, ultimately through a web-based UI (front-end) that LEA officers can use.

A proof-of-concept Citizen Reporting (CR) tool – comprising the two key components: Android app and server-side system – has been designed, implemented, tested and deployed to demonstrate the feasibility of the solution.

Further refinement and improvement of the CR tool will be made in due course, taking into account the relevant feedback and comments from potential end users.

References

[1] "Novel Strategies to FigHt Child Sexual Exploitation and Human TRafficking Crimes and PrOtect thEir VictimS," http://www.heroes-fct.eu, accessed: 4 January 2024.

[2] M. A. Wood, S. Ross, and D. Johns, "Primary Crime Prevention Apps: A Typology and Scoping Review," *Trauma, Violence, & Abuse*, vol. 23, no. 4, pp. 1093–1110, 2022. [Online]. Available: https://doi.org/10.1177/1524838020985560

[3] C. Charitou, D. G. Kogias, S. E. Polykalas, C. Z. Patrikakis, and I. C. Cotoi, "Use of apps for crime reporting and the eu general data protection regulation," *Societal implications of community-oriented policing and technology*, pp. 55–61, 2018.

[4] M. Mwiya, J. Phiri, and G. Lyoko, "Public Crime Reporting and Monitoring System Model Using GSM and GIS Technologies: A Case of Zambia Police Service," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, vol. 4, pp. 207–226, 2015. [Online]. Available: https://templaterepublic.com/wp-content/uploads/Crime-Report-10.pdf

[5] K. N. Jayasinghe and M. P. L. Perera, "Impact of Crime Reporting System to Enhance Effectiveness of Police Service," *International Journal of Computer Trends and Technology*, vol. 69, no. 5, pp. 1–5, 2021. [Online]. Available: https://doi.org/10.14445/22312803/IJCTT-V69I5P101

[6] H. M. Paterson, C. van Golde, C. Devery, N. Cowdery, and R. Kemp, "iWitnessed: Capturing Contemporaneous Accounts to Enhance Witness Evidence," *Current Issues in Criminal Justice*, vol. 29, no. 3, pp. 273–281, 2018. [Online]. Available: https://doi.org/10.1080/10345329.2018.12036102

[7] F. D. Boateng, "Crime Reporting Behavior: Do Attitudes Toward the Police Matter?" *Journal of Interpersonal Violence*, vol. 33, no. 18, pp. 2891–2916, 2018. [Online]. Available: https://doi.org/10.1177/0886260516632356

[8] K. Tabassum, H. Shaiba, S. Shamrani, and S. Otaibi, "e-Cops: An Online Crime Reporting and Management System for Riyadh City," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*, 2018, pp. 1–8. [Online]. Available: https://doi.org/10.1109/CAIS.2018.8441987

[9] D. Lal, A. Abidin, N. Garg, and V. Deep, "Advanced Immediate Crime Reporting to Police in India," *Procedia Computer Science*, vol. 85, pp. 543–549, 2016.

[10] C. Oduor, F. Acosta, and E. Makhanu, "The Adoption of Mobile Technology as a Tool for Situational Crime Prevention in Kenya," in *2014 IST-Africa Conference*, 2014. [Online]. Available: https://doi.org/10.1109/ISTAFRICA.2014.6880669

[11] J. Akinyede, C. Olebu, A. Ponnle, F. Akinluyi, A. Thompson, O. Dahunsi, B. Alese, and M. Oyinloye, "Development of a Real-Time Crime Management System in Southwestern Nigeria: The Mobile Application," *European Journal of Science, Innovation and Technology*, vol. 2, pp. 68–82, 2022. [Online]. Available: https://ejsit-journal.com/index.php/ejsit/article/view/138/124

[12] S. Roshan, S. V. Kumar, and M. Kumar, "Project spear: Reporting human trafficking using crowdsourcing," in *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*, 2017, pp. 295–299.

[13] V. E. Chinoko, R. Kalimuthu, and P. Macheso, "A cloud based android system for reporting crimes against child sexual abuse," *International Journal Of Computer Communication And Informatics*, vol. 3, no. 2, pp. 84–93, 2021. [Online]. Available: https://doi.org/10.34256/ijcci2128

HEROES

Consortium

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101021801