A picture containing chart

Description automatically generated

|  |  |
| --- | --- |
|  | Community Shield Project – Final Documentation |
|  | By: Leope, Neo (Mr) (s220605645) |
|  | Lecturer: Professor B. Haskins  Module: IT Project 3 – ITPV302  Faculty: Faculty of Engineering, the Built Environment, and Information Technology  Department: School of Information Technology  University: Nelson Mandela University  11/21/21 |

Table of Contents

[Introduction 3](#_Toc88410042)

[1.1 “Public” Protection 3](#_Toc88410043)

[1.2 Project objectives 3](#_Toc88410044)

[1.3 Problem background 3](#_Toc88410045)

[1.4 Related systems analysis 4](#_Toc88410046)

[1.4.1 My SAPS 4](#_Toc88410047)

[1.4.2 Community Police Forum 5](#_Toc88410048)

[1.4.3 Safe Community 6](#_Toc88410049)

[1.5 The project plan 8](#_Toc88410050)

[1.6 Risk Analysis 10](#_Toc88410051)

[1.6.1 Risk Handling 11](#_Toc88410052)

[1.7 References 12](#_Toc88410053)

[2.1 Plan to Address Objectives 13](#_Toc88410054)

[2.2 Project scope 14](#_Toc88410055)

[2.2.1 Information scope 14](#_Toc88410056)

[2.2.2 Functional scope 14](#_Toc88410057)

[2.2.3 Communication scope 14](#_Toc88410058)

[2.3 Business requirements 15](#_Toc88410059)

[2.4 Hardware and Software Requirements 15](#_Toc88410060)

[2.4.1 Software Requirements 15](#_Toc88410061)

[2.4.2 Hardware Requirements 15](#_Toc88410062)

[2.5 Design constraints 15](#_Toc88410063)

[2.5.1 Security constraints 15](#_Toc88410064)

[2.5.2 Interface constraints 16](#_Toc88410065)

[2.5.3 Performance constraints 16](#_Toc88410066)

[2.6 High-level use case diagram 16](#_Toc88410067)

[2.7 Entity Relational Database Model Diagram 17](#_Toc88410068)

[2.8 User Interface Design 18](#_Toc88410069)

[2.9 References 29](#_Toc88410070)

[3.1 Choice of Tools 30](#_Toc88410071)

[3.1.1 Android Studio 2020.3.1 (Arctic Fox)/28 July 2021 30](#_Toc88410072)

[3.1.2 Firebase (Updated October 29, 2021) 30](#_Toc88410073)

[3.1.3 Kotlin (Updated September 20, 2021) version 1.5.31 30](#_Toc88410074)

[3.2 Extracts of complex code 31](#_Toc88410075)

[3.2.1 getData() 31](#_Toc88410076)

[3.2.2 Rounded Card Corners 32](#_Toc88410077)

[3.3.3 GPS isLocationEnabled Function 32](#_Toc88410078)

[3.3.4 Fetching Locations 32](#_Toc88410079)

[3.3.5 Add data to the Anonymous and Non-anonymous databases 33](#_Toc88410080)

[3.3 Source code references 35](#_Toc88410081)

[3.3.1 GridLayout with CardView 35](#_Toc88410082)

[Description 35](#_Toc88410083)

[Usage 35](#_Toc88410084)

[Reference 35](#_Toc88410085)

[3.3.2 Splash Screen 35](#_Toc88410086)

[Description 35](#_Toc88410087)

[Usage 35](#_Toc88410088)

[Reference 35](#_Toc88410089)

[3.4 Problems encountered 36](#_Toc88410090)

[3.4.1 Creating a Navbar 36](#_Toc88410091)

[3.4.2 Fragment Utilization 36](#_Toc88410092)

[3.4.3 Attaching GPS Co-ordinates to SMS message 36](#_Toc88410093)

[3.5 References 37](#_Toc88410094)

[5.1 Study conclusion 37](#_Toc88410095)

Business Case

# Introduction

This document is part of an ITPV302 final year project. It provides a description of the approach followed by Mr. NR. Leope when making important application design and implementation decisions. It is intended to inform those with a general interest and those who wish to understand better the underlying “Community Shield” project development process. It also provides the project objectives and risk analysis. This document provides a clear explanation of the: Project objectives; The problem; Problem background; Related System analysis; Project plan; and Risk Analysis.

# 1.1 “Public” Protection

The **problem of** having no immediate source of refuge to turn to when in danger or in need, but instead having to wait minutes or hours for police to come to rescue, if they are not too overwhelmed to even show up, **affects** most communities and citizens in their homes, at work, at a function/event and other areas. **The impact** of this lengthy response time can be the difference between your safety and experiencing a preventable crime, saving your life or preventing threats to whistle blowers who witness a crime. A successful **solution would** provide an easy and simple way to attain immediate refuge from relevant authorities.

# 1.2 Project objectives

Real-world problem issues that Community Shield software intervention will address:

* Replace the longer phone call to police with a button click (panic button).
* Reduce threat caused to crime witness whistle-blowers by ensuring they remain anonymous.
* Ensure that individuals in need of immediate refuge can achieve such assistance as soon as possible without having to wait the average police response time of 19 minutes (Smith, n.d.).
* Provide a central means by which citizens can keep up with crime events and their own safety.

# 1.3 Problem background

2019, Bellville, Cape Town, on a Thursday morning. Concerned colleagues go over to the apartment where Lynette Volschenk, 32, lives to check in on her. Upon their arrival they discover a man wearing blue gloves in the apartment and after alerting local authorities the local authority(police) discover multiple black bags containing pieces of Lynette’s body. The Wednesday night before neighbours heard noises coming from the apartment (Etheridge, 2019).

Babita Deokaran, 53, a Gauteng senior health official from Johannesburg South was gunned down outside her home on Monday, 23rd August 2021. Babita was a witness to personal protective equipment tender fraud scandal worth over R300 million. When news broke it was unclear whether her death was personal or because of her being a whistle-blower to the crime, as witnesses were ‘undisclosed’. Since then multiple suspects have been arrested and it has been linked to her role as a whistle-blower against the unethical fraudulent activity (Bhengu, 2021).

July 13, 2021. South Africa is in distress and malls, shopping centres, local shops and small businesses are being ransacked. Upon hearing that their mall was in danger of being a victim, of the current state of affairs in the country, Soweto community show up on a Tuesday winter night to protect Maponya Mall (Head, 2021).

From the above a few problems and solutions may have been noticed:

1. Lynette Volschenk could have been saved by her neighbours had they known the noises were her being murdered in cold blood.
2. Babita Deokaran could have remained truly anonymous if no one knew her name, saw her reporting the crime and all she had to do was give an anonymous tip off via a faceless platform. And thus, would still be a present mother in her 16-year-old daughters’ life and could have gone about her life as usual.
3. The power of what can be achieved when a community and private security come together to aid in protecting that community and its assets.
4. Given the police response time it can also be seen that the only true **immediate** refuge can only be found from the surrounding community and patrolling private security

# 1.4 Related systems analysis

## 1.4.1 My SAPS

Mobile Application. “My SAPS” is a free application developed discreetly by the South African Police Services (SAPS). The current main function of the application is to allow users to send crime tip-offs (named or anonymous) to the crime stop centre. Besides sending crime tip offs, the application can also be used to find police stations nearby, identifying persons on the wanted list or even persons missing and even view crime statistics. (Vodacom, 2021)

The main features that stand out from the application that would be incorporated into the “Community Shield” application being built is the ability to locate the nearest police station and to send anonymous tip offs as seen in figure 2. The feature which will be not avoided but enhanced upon is the “Call Crime Stop” feature, this feature will be replaced with a panic button instead of a call. Community Shiel will also have a Menu screen to navigate through the application as seen in figure 1.

Graphical user interface, text, application, email

Description automatically generated

A screenshot of a cell phone

Description automatically generated with medium confidence

Figure 1 MY SAPS Home Screen

Figure 2 Submit Tip-Off First Screen

## 1.4.2 Community Police Forum

Mobile Application. “Community Police Forum” is an application which allows individuals in community to be the eyes and ears that safeguard their local community. The application allows Neighbourhood Watch, Local police, private security, armed response, etc. to communicate with each other in order to try and effectively safeguard community from crime. The application allows users to stay updated on recent criminal activities or suspicious activity that has been happening in their local community/neighbourhood.

Some features that allow for the application to meet its use cases are crime/suspicious activity reporting, GPS Based Locations, Submitting Photos, Submitting detailed descriptions, etc. (Community, Community Police Forum, 2021)

Of the many features that “Community Police Forum” has on offer having a splash screen as seen in figure 3 and User Permissions for location us as seen in figure 4 would be considered for use in “Community Shield”. GPS Based locations in order to be able to send the users GPS co-ordinates to relevant authorities.

However, some features will not be suitable to “Community Shield”. Features such as: Downloading training docs, calling crime stop and posting to forum.

Graphical user interface, application

Description automatically generatedA picture containing logo

Description automatically generated

Figure 3 Community Police Forum Opening Screen

Figure 4 Community Police Forum User Information Capturing Screen

## 1.4.3 Safe Community

Mobile Application. “Safe Community” is an application used to build a network of individuals, local Community Police Forums, The South African Police Services, Community/Neighbourhood Watches, etc. and by creating this network the application allows these various individuals and institutions to come together in protecting and battling crime in the area. (Community, 2021)

The application allows users to get incident alerts and notifications of crimes and suspicious activity in the area, view real time images from traffic cameras on major roadways, ability to contact a superhero (individual anywhere you are in an emergency), GPS locations and addresses, etc.

“Safe Community” has some amazing features to consider incorporating into “Community Shield”, the ones that stand out are the ability to navigate to make alerts as seen in figure 6, ability to contact a Superhero and report crime activity whether emergency or non-emergency. Another feature as seen in figure 3 is the splash screen as seen in figure 5.

A feature that should be avoided when creating “Community Shield” is the real time traffic camera images as those would be completely redundant to the purpose of the application.

Logo, company name

Description automatically generated

Figure 5 Safe Community Opening Screen

Graphical user interface, application

Description automatically generated

Figure 6 Image with Safe Community Application features by (Community, Safe Community - Crime Prevention In Your Area, 2021)

# 1.5 The project plan

Table 1 gives information on who is a part of the project, who the project client is and the general time frame of the project. Table 2 is a representation of the project schedule and deadlines for each project milestone.

Table Table of project team and the project client

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PROJECT DELIVERABLE | Community Shield Application | | | | |
| CLIENT | South African Community Police Association (SACPA) | | | | |
| PROJECT MANAGER | Mr N.R. Leope | | | | |
| SCOPE STATEMENT |  | | | | |
| START DATE | 29/08/2021 | END DATE | 15/11/2021 | OVERALL PROGRESS | 100% |

Table 2 Table of project milestones, timeframe to complete milestone and deadlines of milestones

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TASK NAME** | **ASSIGNED TO** | **START  DATE** | **END  DATE** | **DURATION**  in days | **% COMPLETE** | **STATUS** | **NOTES** |
| Proposal Document | Mr N.R. Leope | 20/08/2021 | 29/08/2021 | 9 | 100 | Complete |  |
| Draft Business Case Document | Mr N.R. Leope | 29/08/2021 | 19/09/2021 | 21 | 100 | Complete |  |
| Draft System Requirements, Specifications and Technical Design Document | Mr N.R. Leope | 20/09/2021 | 17/10/2021 | 27 | 100 | Complete |  |
| Pre-Design | Mr N.R. Leope | 20/09/2021 | 23/09/2021 | 3 | 100 | Complete |  |
| Design (Draft Application) | Mr N.R. Leope | 24/00/2021 | 29/09/2021 | 5 | 100 | Complete |  |
| Development | Mr N.R. Leope | 30/09/2021 | 05/11/2021 | 39 | 100 | Complete |  |
| Draft Implementation Document | Mr N.R. Leope | 18/10/2021 | 07/11/2021 | 24 | 100 | Complete |  |
| Narrated System Overview Video &  Final Document | Mr N.R. Leope | 08/11/2021 | 14/11/2021 | 6 | 100 | Complete |  |
| System Presentation | Mr N.R. Leope | 15/11/2021 | 19/11/2021 | 4 | 100 | Complete |  |
| Final System Submission | Mr N.R. Leope | 20/11/2021 | 21/11/2021 | 2 | 100 | Complete |  |

# 1.6 Risk Analysis

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RISK** | **ID** | **ASSIGNED TO** | **LIKELIHOOD** | **IMPACT** | **SEVERITY** | **MITIGATION**  **NOTES** | **CONTINGENCY**  **NOTES** | **STATUS** | **RISK HANDLING**  **STATUS** |
| Unexpected/Unplanned work that needs to be completed | 1 | Mr N.R. Leope | Low | High | Medium | Assessment of work that done from similar project to be done in order to draw correct work scheduling assumptions | Draw up current project plan to conclude a week before submission in order to be able to restructure project plan if needs must. | Mitigated | Acceptable |
| Theft of materials, intellectual property or equipment | 2 | Mr N.R. Leope | Medium | High | High | Backup Application and all project deliverables onto external hard disk drive and OneDrive | Notify police and consult a data recovery specialist. | Mitigated | Minimizable |
| Complete or partial rewrite of application due to change in policy, business case or project scope | 3 | Mr N.R. Leope | Low | High | High | Review business case early on before beginning application development | No current contingency plan | Mitigated | Avoidable |
| Lack of security user data expertise and education | 4 | Mr N.R. Leope | Medium | Medium | Medium | Test application for any user data security risks |  | Mitigated | Transferable |

This section outlines Table 3 which gives information about the possible risks, who the risk is assigned to, the likelihood that the risk will occur, Impact of the risk, severity of the risk and plans of how to stop/limit the risk and the current status of the risk(has it been mitigated or is it still threat prone)

Table 3 Risk Analysis Information

## 1.6.1 Risk Handling

**Acceptable risks:** We decide that the risk is very unlikely, or possibly, that the costs if that event happens, are lower than the expenses which are necessary to prevent the event from taking place.

**Avoidable risks:** We take action which ensures that we prevent a certain event or condition. We might create backups of all the code and store it in two separate locations. Then we avoid the risk of losing anything,

**Minimizable risks:** We take steps to ensure that the likelihood of something happening is less. We can create a bonus structure to make sure that people do not leave.

**Transferable risks:** This means that we transfer the risk to someone else. For example, we might send part of the work to another company. Then it is the other company, which needs to assume the risk. Once we have categorised all the risks, we need to decide exactly what needs to be done, if something should be done. We need to prepare ourselves as best we can so that any potential events have as little impact on the project as possible. We can, of course, have free fruit available at work and encourage everyone to exercise regularly. But someone will get sick in any case and then it is great to know how we handle that. Who is going to take over the sick person's work? When is it necessary for someone to step in? Should the person step in be someone in the team or someone from outside the team? In the case of the latter, where should the person be brought in from? How much will this cost?

# 1.7 References

Bhengu, L. (2021). Babita Deokaran murder: 2 more people, found in possession of large sums of cash, arrested. *News24*, https://www.news24.com/news24/southafrica/news/babita-deokaran-murder-two-more-arrested-caught-in-possession-of-large-sums-of-cash-20210902.

Community, S. (2021). Community Police Forum. *Google Play Store*, https://play.google.com/store/apps/details?id=za.co.sacctn.sc.cpf&hl=en\_ZA&gl=US.

Community, S. (2021). Safe Community - Crime Prevention In Your Area. *Google Play Store*, https://play.google.com/store/apps/details?id=za.co.safecommunity.main&hl=en\_ZA&gl=US.

Etheridge, J. (2019). Cape Town woman's body found chopped up at apartment building. *News24*, https://www.news24.com/news24/southafrica/news/cape-town-womans-body-found-chopped-up-at-apartment-building-20190822.

Head, T. (2021). Heroes emerge, as Soweto residents defend Maponya Mall from looters. *thesouthafrican*, https://www.thesouthafrican.com/news/maponya-mall-defended-community-soweto-who-is-protecting/.

Vodacom, P. L. (2021). MY SAPS. *Google Play Store*, https://play.google.com/store/apps/details?id=co.za.vodacom.boxfusion.saps&hl=en\_ZA&gl=US.

System Requirements

Specifications and Technical Design

This Section of the document will cover systems Requirement, specifications and technical design. The section will give detail information on the plan too address the objectives set out in the business case, give a through project scope with information, functional and communication scopes, dive into business, hardware and system requirements, design constraints, ui design and system diagrams.

# 2.1 Plan to Address Objectives

In the business case a few objectives were specified. These are real-world issues that the software intervention will address. In this section stipulates what will be done within the software to address these objectives. The following objectives were stated in the business case:

* Replace the longer phone call to police with a button click (panic button).
* Reduce threat caused to crime witness whistle-blowers by ensuring they remain anonymous.
* Ensure that individuals in need of immediate refuge can achieve such assistance as soon as possible without having to wait the average police response time of 19 minutes (Smith, n.d.).
* Provide a central means by which citizens can keep up with crime events and their own safety

These objectives will now be discussed as follows:

In order to the replace the longer phone call to police with a button click (panic button) system used by neighborhoods and citizens of South Africa, an Android-based mobile application will be developed. Android was chosen as it is the most widely used mobile operating system under the target group of not only South Africa but the African continent, accounting for 83.86% of mobile operating system market share (O'Dea, 2021). A mobile application will ensure that the South African neighborhoods and citizens have access to the system wherever they may be on any given day.

Because of the mobile access penetration rate which shows 95% of South Africans have access to a mobile device (of which 91% of mobile devices are smartphones), it is only appropriate to leverage access to this technology in order to improve the current systems and processes (McCrocklin, 2021). A mobile application will be able to address this issue by allowing users alert local authorities, through a ‘panic’ button, without having to make a phone call. As users may be stuck in scenarios whereby making a call will alarm the perpetrator and aggravate the perpetrator\s, it is essential to ensure the system is a quick process and will allow users discretion and efficiency when reporting an active crime.

Currently, if a citizen needs to report a crime, active or non-active, they need to go through a phone call which could be congested by other citizens calling. Because helplines can be congested it means if an individual is in a life-threatening situation, they may not even get the chance to be put through to report the crime, not only that but if they do get through, they have to answer a series of key questions to ensure police can locate them. A mobile application will allow the local authorities in and around the individual to get the key information and locate the individual within a few clicks that could take less time than trying to reach through a helpline. The mobile application will store key information the local authorities will need in the event of an emergency and send it through when the user accesses the application to report a crime, provided they are not just trying to give an anonymous tip-off.

# 2.2 Project scope

The project scope provides a high level view of data requirements for “Community Shield”. This includes a discussion on the information, functional and communication scope of the intended system.

## 2.2.1 Information scope

The Community Shield project involves collecting user information such as name, surname, address and cell phone. Other information that the application will require is news alerts and key information about recent dangers in and around the area the user may access, this may be gathered either by storing this key information in a various firebase databases (News Database, Wanted List Database, User Database, etc.).

## 2.2.2 Functional scope

In the application the main concept/key feature is the panic button on the “report” page, this button will allow a user to essentially send an SOS notification containing their key information (such as location) to relevant authorities.

## 2.2.3 Communication scope

In the Community Shield system an SOS notification will be sent out to relative parties and the system will work over both mobile connections and Wi-Fi. As the application requires users access to SMS and GPS services, communication will need to be made with user in order to obtain permissions access these functions

# 2.3 Business requirements

End users (such as citizens) require a system that is dependable and always available. Relevant authorities will also require the system to notify them of incidents. It should have an easy-to-use interface for the end-user to navigate as quickly in time constraint life-threatening scenarios. The system should also allow end-users to keep track of crime events and therefore should stay up to date with what is happening in various areas.

Security organizations or individuals should receive some sort of notification in order to let them know of an emergency. This notification should also make it easy for the emergency response personal to locate an individual who needs assistance.

# 2.4 Hardware and Software Requirements

## 2.4.1 Software Requirements

* The mobile application will be developed using Android Studio.
* The mobile application will use Firebase as its local storage medium and user authentication.
* Unsplash.com will be used to attain images or video necessary for UI/UX design.
* The mobile application requires a phone running (at least) Android 5.1 (Lollipop).

## 2.4.2 Hardware Requirements

* The mobile application requires a mobile phone with at least 1Gb of RAM, capable of running Android 5.1
* Application can only be installed on mobile devices with sufficient storage (more or less 500mb).
* Mobile application requires a device to have location services and a connection to the internet.

# 2.5 Design constraints

This section, discusses any constraints that may exist in the process of developing the “Community Shield” application. This includes things such as security, interface and performance constraints.

## 2.5.1 Security constraints

User information (Name, Surname, Cell nr., Physical Address) are of a sensitive nature. These alongside the other bits of information sent with the SOS notification should not get into the wrong hands. Data integrity and confidentiality should be ensured by using email and password to authenticate a user before login onto system. Thereby ensuring only authorized personnel have access to this information.

## 2.5.2 Interface constraints

The mobile interface needs to be user-friendly (especially as user may be of any age and thus whether a user is technically savvy or not, young or old) any user must be able to navigate through the application.

## 2.5.3 Performance constraints

The system requires a user to have a mobile device with an internet connection and have enabled location in settings in order to send location in SMS and also requires user to grant permission to use SMS and GPS Location services.

# 2.6 High-level use case diagram

Figure 7 presents a high-level use case diagram of the Community Shield system and its users.

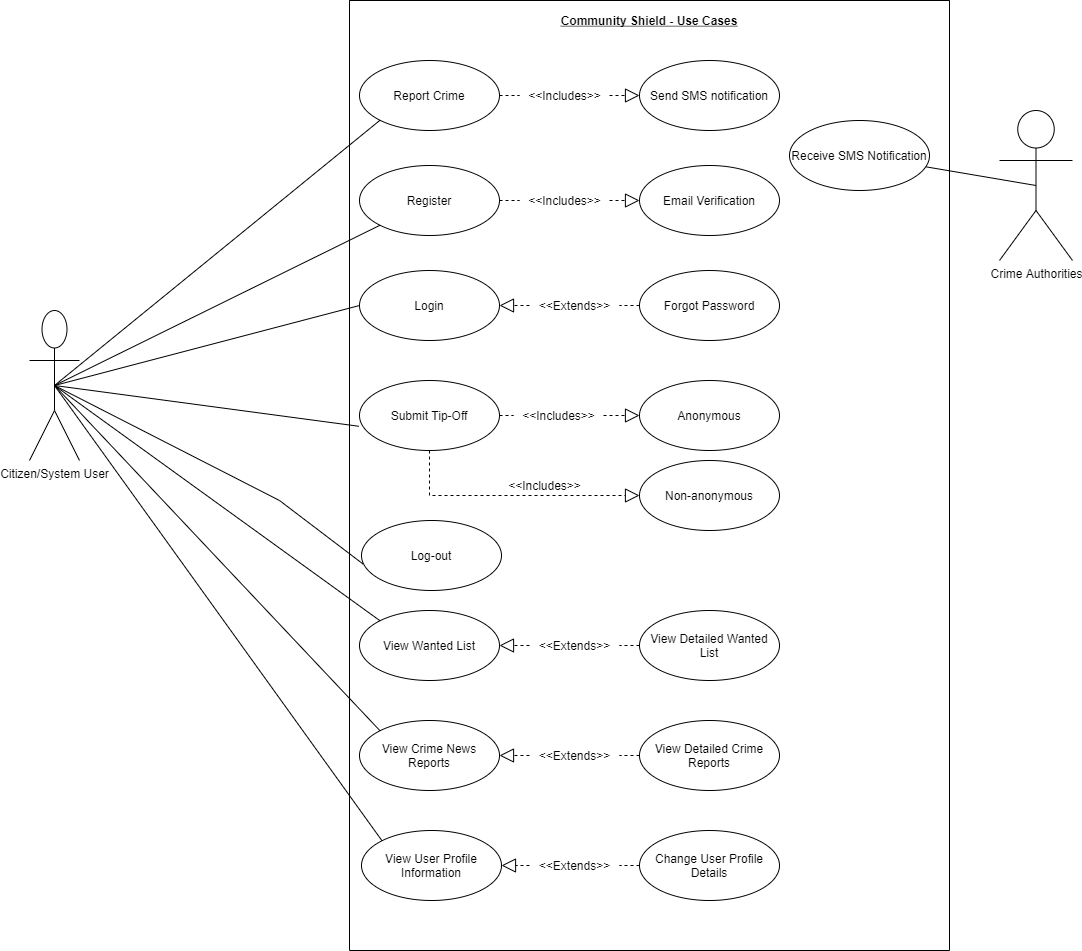


Figure 7 High Level Use Case Diagram for Community Shield

# 2.7 Entity Relational Database Model Diagram

Figure 9 represents the entity relationship diagram of the YOURPLANNEDSYSTEMNAME system.

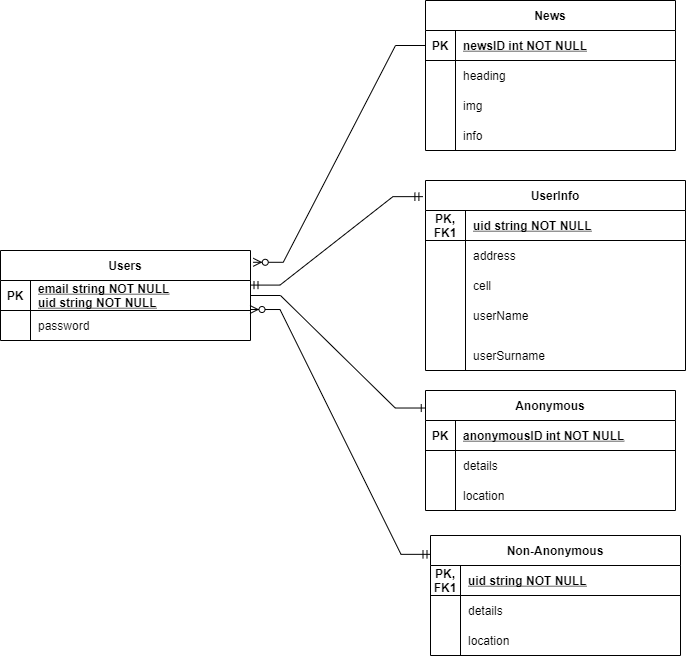


Figure 8 Community Shield Entity Relationship Diagram

# 2.8 User Interface Design

Figure 10 represents the Splash screen of the Community Shield system. This screen is the first screen when the application is opened. It is displayed for 3 seconds before navigating to the application login screen seen in figure 11.



Figure 9 Splash Screen

Figure 11 represents the login screen of the Community Shield system. The screen requires that a user enter their email address (as a username) and a password. Upon successful login, the user will be redirected to the system’s landing screen(Fig 1.5). The Screen Also Provides the User with option to Register as a new user.

Graphical user interface, website

Description automatically generatedFigure 12 represents the register screen of the Community Shield system. This screen allows a new user to register using their email and password

Figure 10 Login Screen

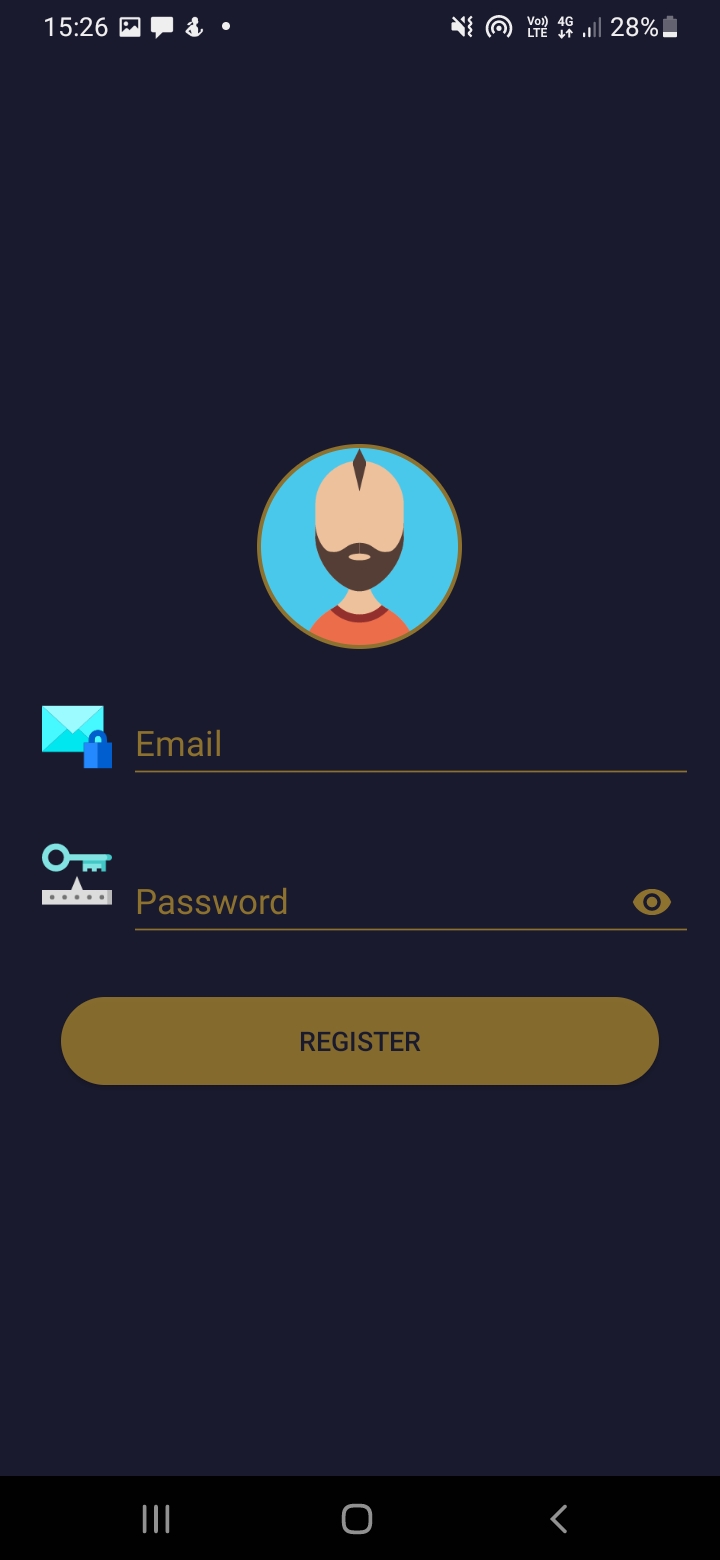


Figure 11 Register Screen

Figure 13 represents the forgot password screen of the Community Shield system. This screen allows the user to enter their valid email and a link will be sent to this email which will allow user to change password.

A picture containing text, toiletry, cosmetic

Description automatically generated

Figure 12 Forgot Password Screen

Figure 14 represents the menu screen of the Community Shield system. This screen allows the user to navigate through the various screens and various activities that can be carried out using the application.

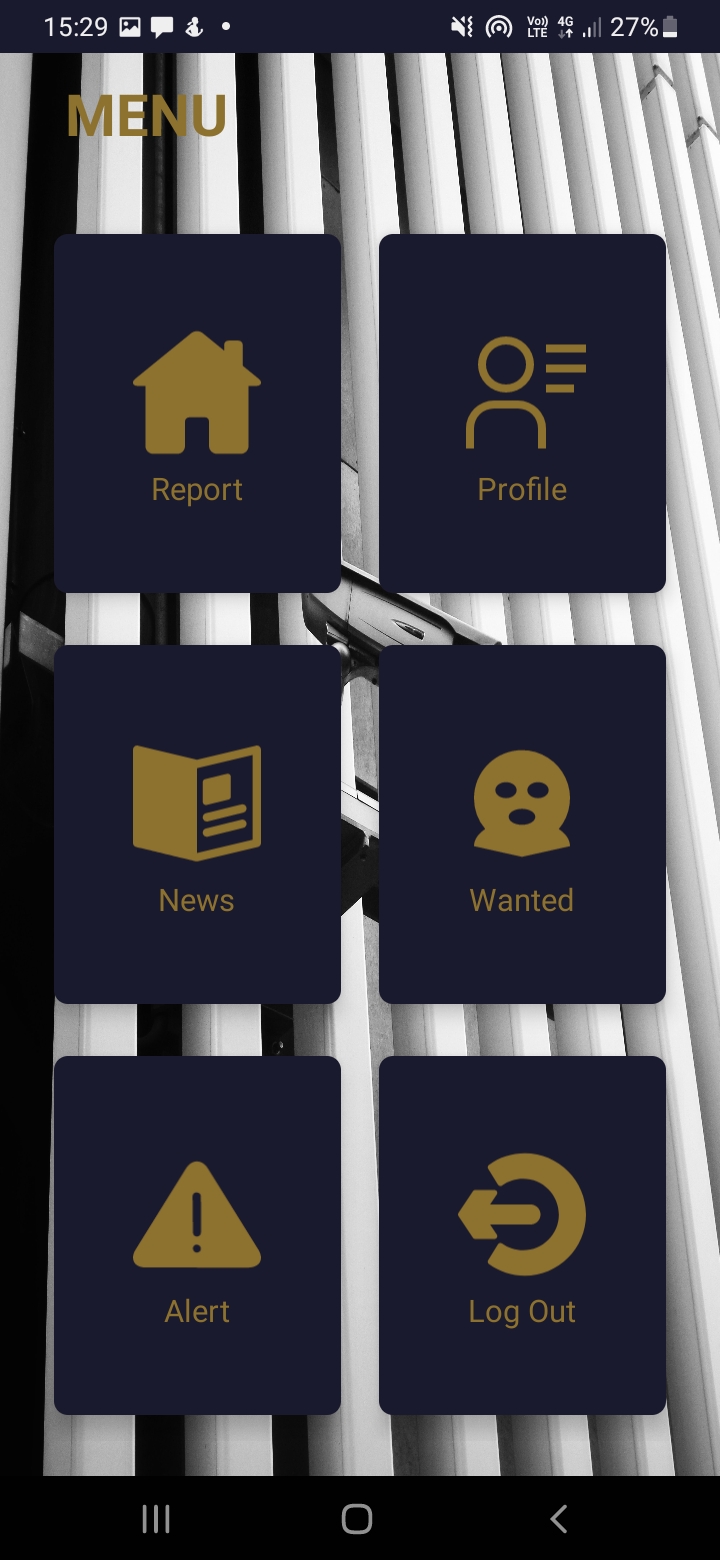


Figure 13 Menu Screen

Figure 15 represents the report screen of the Community Shield system. This screen allows the user to request emergency or non-emergency crime assistance from relevant authorities and will send them an SMS with these request details.

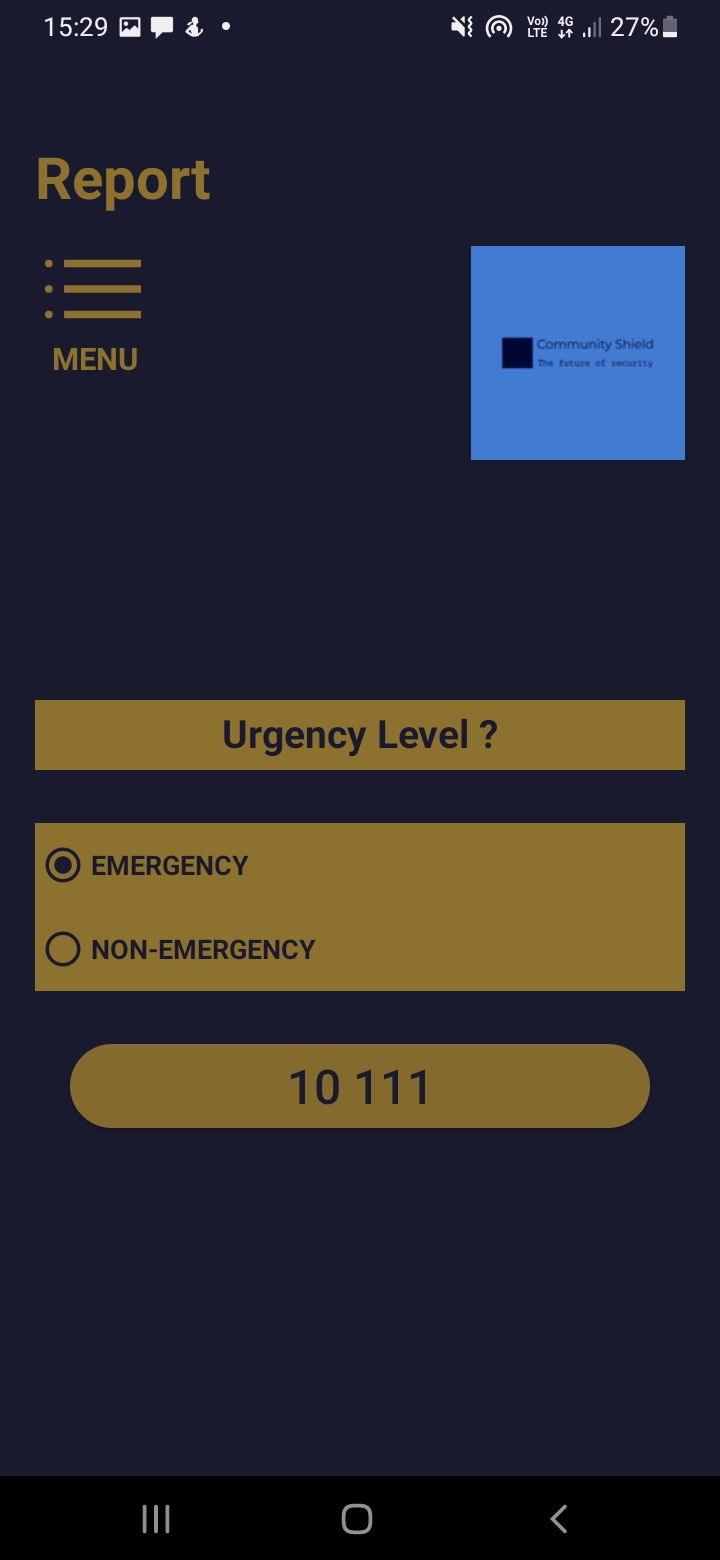


Figure 14 Report Screen

Figure 16 represents the profile screen of the Community Shield system. This screen allows the user to view the details of their user profile and to navigate either to the Edit Profile Screen (Figure 17) or Go back to menu (Figure 14).

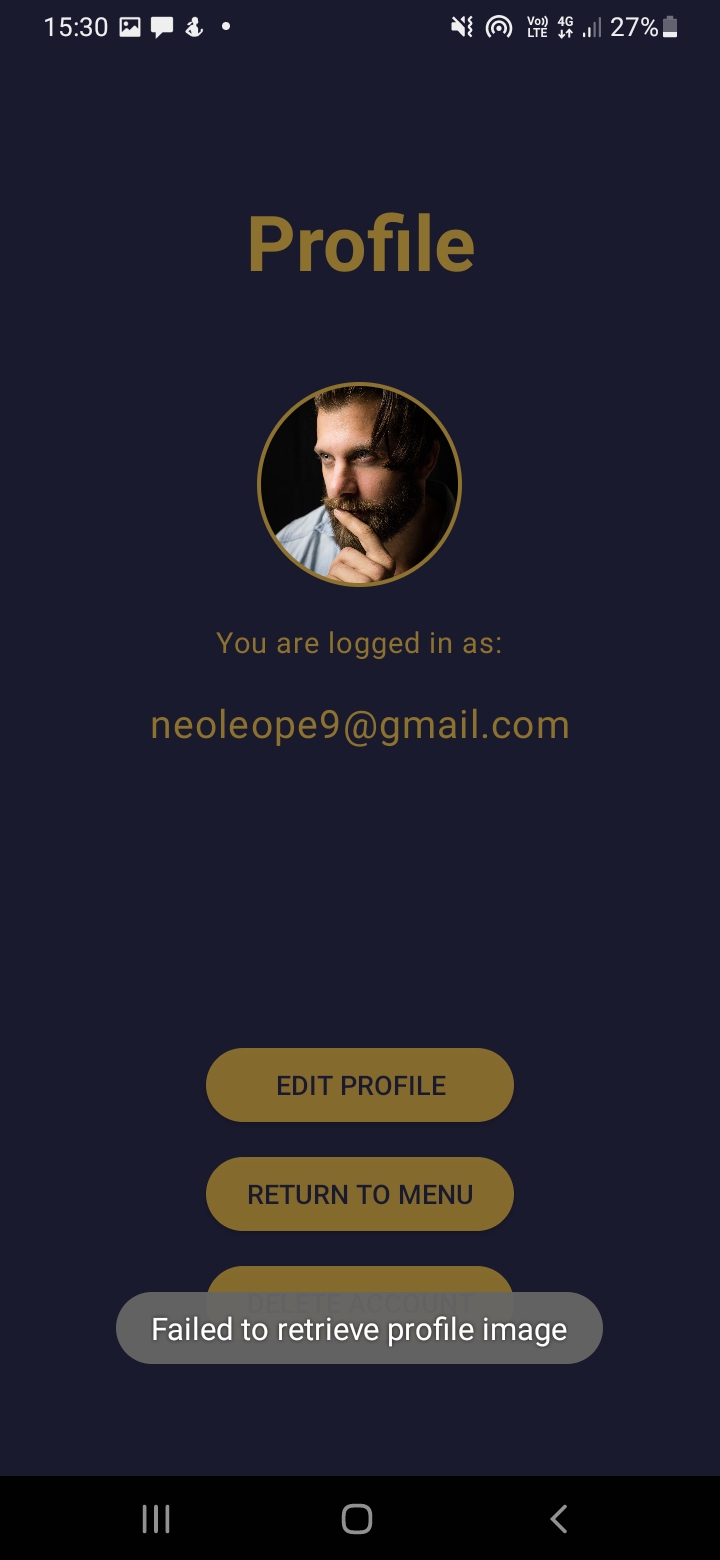


Figure 15 Profile Screen

Figure 17 represents the edit profile screen of the Community Shield system. This screen allows the user to enter their user information.

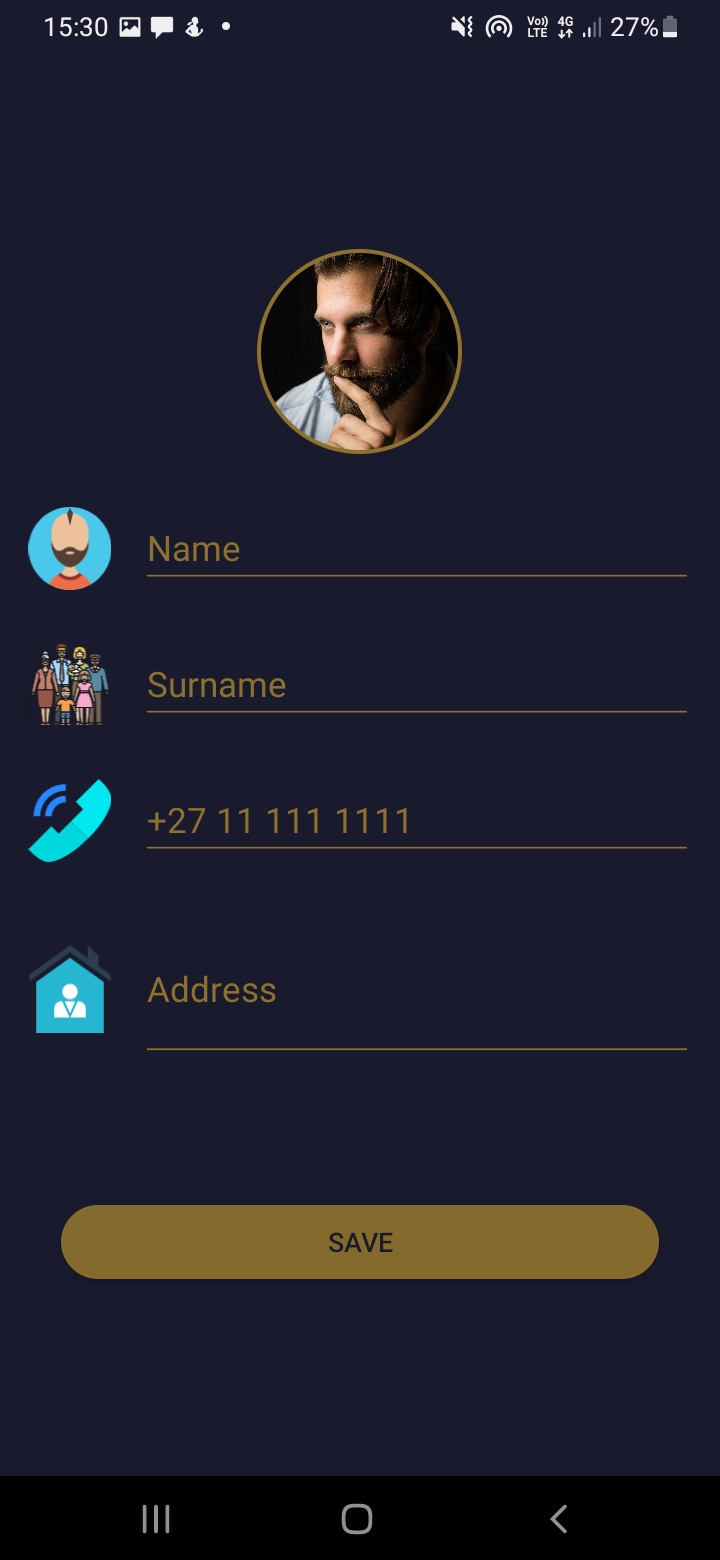


Figure 16 Edit Profile Screen

Figure 18 represents the news screen of the Community Shield system. This screen allows the user view headline crime news and click on them to read the crime story.



Figure 17 News Screen

Figure 19 represents the wanted screen of the Community Shield system. This screen allows the user to view list criminals currently being sought by authorities and click on them to read the crime story.

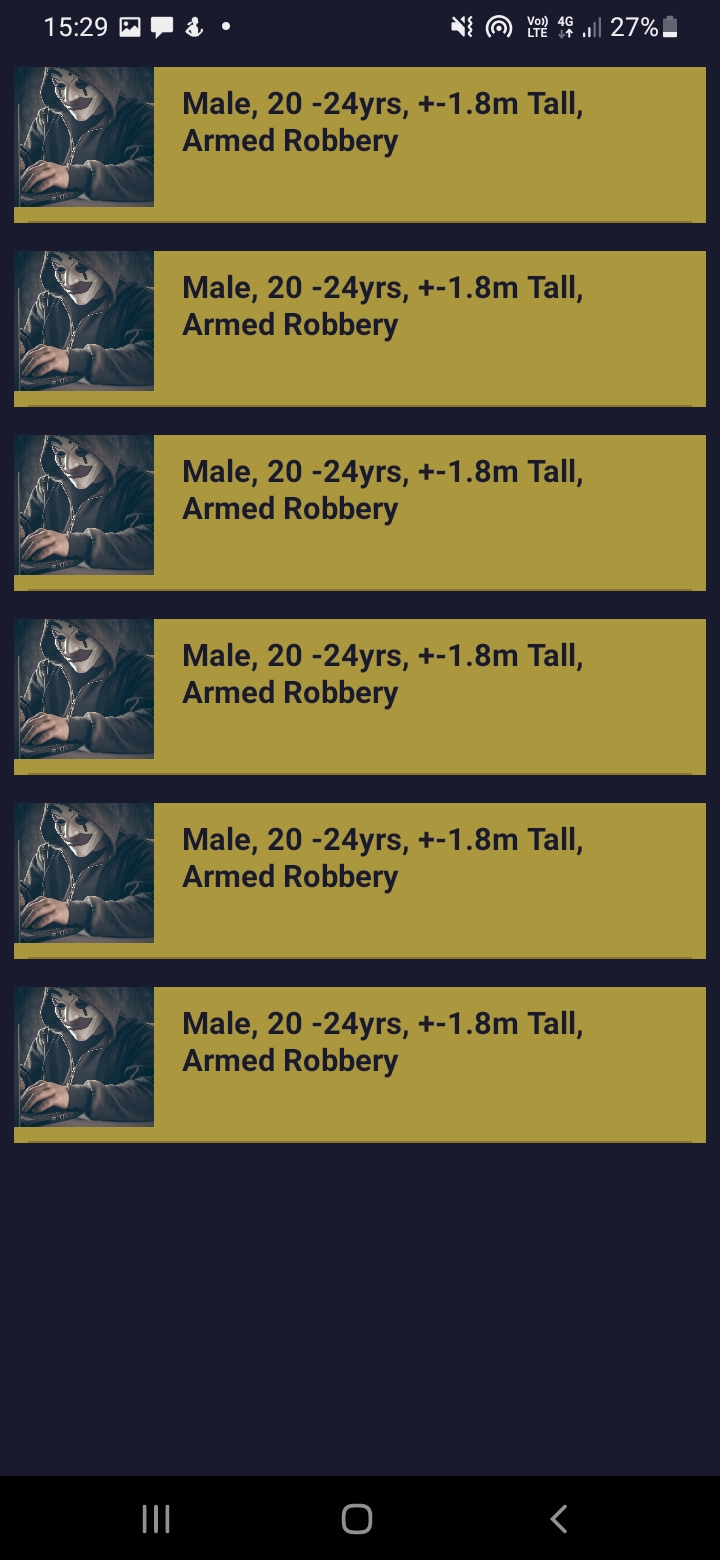


Figure 18 Wanted Screen

Figure 20 represents the alert authorities screen of the Community Shield system. This screen allows the users to send tip-offs which can be anonymous or non-anonymous with location and details of tip-off.

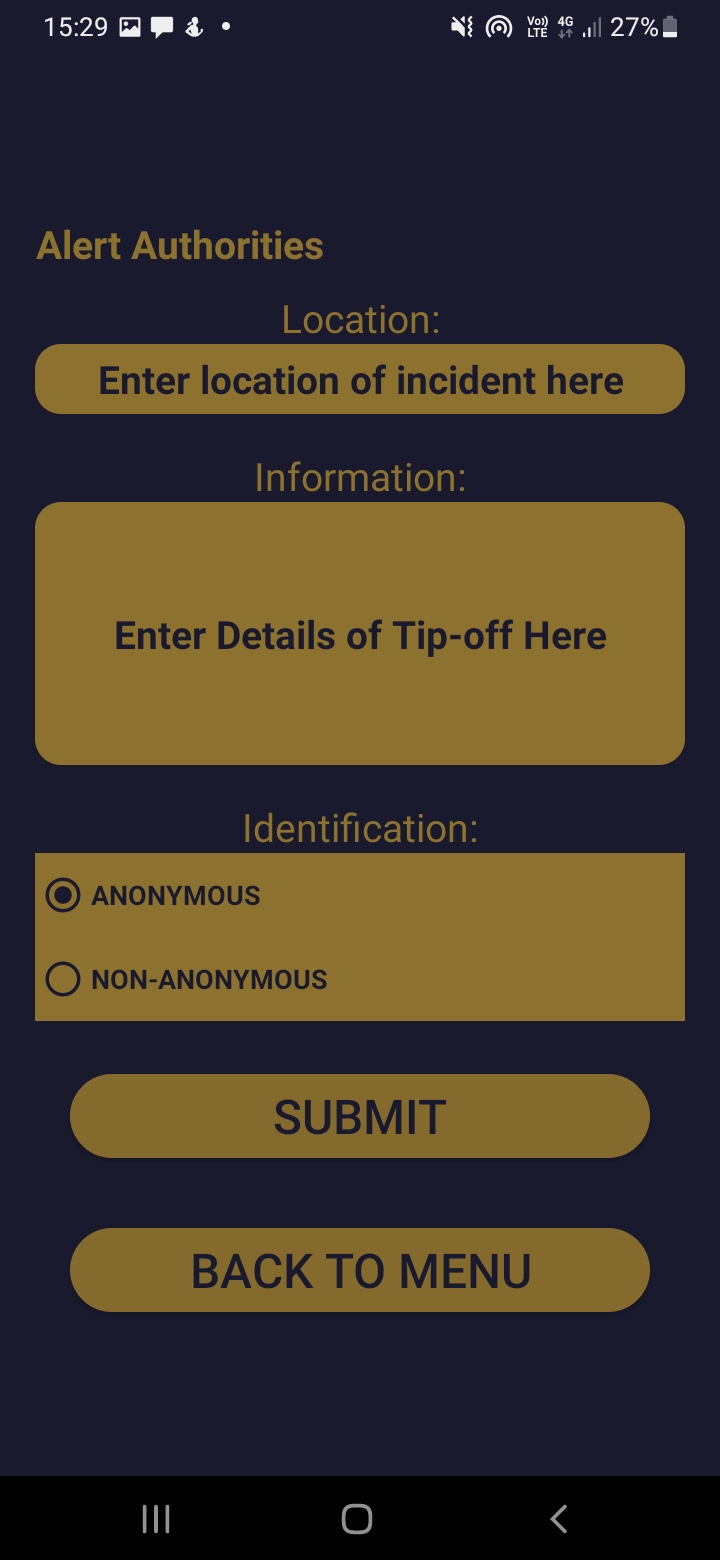


Figure 19 Alert Authorities Screen

# 2.9 References

O'Dea, S. (2021). Market share of mobile operating systems in South Africa from January 2018 to June 2021. *statista.com*, https://www.statista.com/statistics/1063937/market-share-held-by-mobile-operating-systems-in-south-africa/.

Implementation

The implementation part of this document will give an insight into the design choice of tools, difficulties encountered, external code/xml used in the application and complex code/UI that was used.

# 3.1 Choice of Tools

This section shows which technology has been used in the development of the “Community Shield” application. It also discusses why the technology was used and the perceived benefits of using the technology.

## 3.1.1 Android Studio 2020.3.1 (Arctic Fox)/28 July 2021

Android has many of the global devices market share, with over 80% of devices using an android operating system (Zitbek, 2019). As the application is about giving is of access to the large majority, it is very important to develop the application with Android Studio which is a IDE by IntelliJ for developing android operating system applications. Android Studio also provides firebase support with its firebase assistant feature (Ghanchi, 2021), this is an important feature as the application uses firebase for user login/authentication.

## 3.1.2 Firebase (Updated October 29, 2021)

One of the many benefits of using firebase is that it offers synchronizing data with real-time database (Contributor, 2019). This feature plays a key role in the ability to update the news in the database and have them reflect in real-time in the application across devices. Firebase also offers an authentication service which is key to user login in the application.

## 3.1.3 Kotlin (Updated September 20, 2021) version 1.5.31

Lastly the programming language choices were Kotlin and Java. For this application Kotlin was chosen over java as Kotlin is easier to maintain, Kotlin code is safer and should there ever be a need to migrate a legacy system(that uses java) to work with the application then its simple as Kotlin is interoperable with Java (Ababei, 2018)

# 3.2 Extracts of complex code

This section contains snippets of code that were difficult to write. A descriptive summary of what happens in the piece of code is provided for each code extract.

## 3.2.1 getData()

class WantedActivity : AppCompatActivity() {  
  
 private lateinit var newRecyclerView: RecyclerView  
 private lateinit var newArrayList: ArrayList<Wanted>  
 lateinit var suspectImageId : Array<Int>  
 lateinit var crimeInfo : Array<String>  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_wanted*)  
  
 suspectImageId = *arrayOf*(  
 R.drawable.*criminal*,  
 R.drawable.*criminal*,  
 R.drawable.*criminal*,  
 R.drawable.*criminal*,  
 R.drawable.*criminal*,  
 R.drawable.*criminal* )  
  
 crimeInfo = *arrayOf*(  
 "",  
 "",  
 "",  
 "",  
 "",  
 ""  
 )  
  
 newRecyclerView = findViewById(R.id.*wantedRecyclerView*)  
 newRecyclerView.*layoutManager* = LinearLayoutManager(this)  
 newRecyclerView.setHasFixedSize(true)  
  
 newArrayList = *arrayListOf*<Wanted>()  
 getData()  
 }  
  
 private fun getData() {  
 for (i in suspectImageId.*indices*)  
 {  
 val wanted = Wanted(suspectImageId[i],crimeInfo[i])  
 newArrayList.add(wanted)  
 }  
  
 newRecyclerView.*adapter* = WantedAdapter(newArrayList)  
 }  
}

Figure 20 Class WantedActivity (gives more detail into getData() method)

The getData() method (seen in figure 20) is used to produce data for the WantedAdapter. It makes use of a for loop to scan through the suspectImageId.indices and use the current suspectImageId to create wanted value which carries data of type “Wanted” data class. The method then passes this “wanted” value to newArraylist. Once the loop is completed, the contents created are then passed to the WantedAdapter as newArraylist. This method is one cog in the wheel which creates the Wanted section of the application.

## 3.2.2 Rounded Card Corners

<?xml version="1.0" encoding="utf-8"?>  
<shape  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shape="rectangle">  
  
 <solid android:color="@color/mirage"/>  
  
 <corners  
 android:topLeftRadius="400dp"  
 android:topRightRadius="0dp"/>  
  
</shape>

Figure 21 XML showing rounded card corners design

Figure 21 shows XML code that creates more curved/rounded corners which can be seen in the Front End of the Menu in the “Community Shield” application. It helps with the aesthetic nature of the application design.

## 3.3.3 GPS isLocationEnabled Function

Figure 22 demonstrates a function called isLocationEnabled, this function checks if the users location is enabled when requesting to send a report and if not, users will be directed to location settings to enable GPS location.

fun isLocationEnabled(): Boolean {  
 val locationManager =  
 *applicationContext*.getSystemService(Context.*LOCATION\_SERVICE*) as LocationManager  
 return locationManager.isProviderEnabled(LocationManager.*GPS\_PROVIDER*) || locationManager.isProviderEnabled(  
 LocationManager.*NETWORK\_PROVIDER* )  
}

Figure 22 isLocationEnabled function

## 3.3.4 Fetching Locations

Figure 23 gets users current location and if the location permissions are not enabled, the code extract will ask user to give permission to use location services.

private fun fetchLocation() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
  
 if (ActivityCompat.checkSelfPermission(  
 this,  
 android.Manifest.permission.*ACCESS\_FINE\_LOCATION* ) != PackageManager.*PERMISSION\_GRANTED* &&  
 ActivityCompat.checkSelfPermission(this,  
 android.Manifest.permission.*RECEIVE\_SMS*)  
 != PackageManager.*PERMISSION\_GRANTED* ) {  
 ActivityCompat.requestPermissions(  
 this,  
 *arrayOf*(android.Manifest.permission.*ACCESS\_FINE\_LOCATION*,android.Manifest.permission.*RECEIVE\_SMS*, android.Manifest.permission.*SEND\_SMS*),  
 200  
 )  
 return  
 }else{  
 requestLocation()  
 }  
  
  
 }  
}

Figure 23 fetchLocation function

## 3.3.5 Add data to the Anonymous and Non-anonymous databases

Figure 23 shows how data is added to the anonymous and non-anonymous databases.

First check if user selected Anonymous or non-anonymous and then adds data to respective databases, provided that the user entered into both fields.

//add values to databaseAnonymousReference  
if (alertType.*text* == "ANONYMOUS") {  
 if (location.*isNotEmpty*() && details.*isNotEmpty*()) {  
 val alert = AlertInfo(location, details)  
 databaseAnonymousReference.addListenerForSingleValueEvent(object : ValueEventListener {  
 override fun onDataChange(snapshot: DataSnapshot) {  
 val count = snapshot.*childrenCount* + 1  
 databaseAnonymousReference.child(count.toString()).setValue(alert)  
 .addOnSuccessListener **{** binding.editAlertLocationTextMultiLine.*text*.clear()  
 binding.editAlertTextMultiLine.*text*.clear()  
 Toast.makeText(  
 this@AlertActivity,  
 "Successfully sent alert",  
 Toast.*LENGTH\_SHORT* ).show()  
  
 **}**.addOnFailureListener **{** Toast.makeText(  
 this@AlertActivity,  
 "Alert sending - Failed",  
 Toast.*LENGTH\_SHORT* ).show()  
  
 **}** }  
  
 override fun onCancelled(error: DatabaseError) {  
 Toast.makeText(this@AlertActivity, error.*message*, Toast.*LENGTH\_SHORT*)  
 .show()  
 }  
  
 })  
  
 }  
 else  
 {  
 Toast.makeText(this@AlertActivity,"Both Location and Details fields MUST be completed",Toast.*LENGTH\_LONG*).show()  
 }  
  
}  
else  
{  
 if (location.*isNotEmpty*() && details.*isNotEmpty*()) {  
 val alert = AlertInfo(location, details)  
 databaseUidReference.addListenerForSingleValueEvent(object : ValueEventListener {  
 override fun onDataChange(snapshot: DataSnapshot) {  
 val count = snapshot.*childrenCount* + 1  
 databaseUidReference.child(count.toString()).setValue(alert)  
 .addOnSuccessListener **{** binding.editAlertLocationTextMultiLine.*text*.clear()  
 binding.editAlertTextMultiLine.*text*.clear()  
 Toast.makeText(  
 this@AlertActivity,  
 "Successfully sent alert",  
 Toast.*LENGTH\_SHORT* ).show()  
  
 **}**.addOnFailureListener **{** Toast.makeText(  
 this@AlertActivity,  
 "Alert sending - Failed",  
 Toast.*LENGTH\_SHORT* ).show()  
  
 **}** }  
  
 override fun onCancelled(error: DatabaseError) {  
 Toast.makeText(this@AlertActivity, error.*message*, Toast.*LENGTH\_SHORT*)  
 .show()  
 }  
  
 })  
 }  
 else  
 {  
 Toast.makeText(this@AlertActivity,"Both Location and Details fields MUST be completed",Toast.*LENGTH\_LONG*).show()  
 }  
  
}

Figure 23 Add to Anonymous and Non-anonymous databases

# 3.3 Source code references

This section covers the use of code from external sources, personalized and impersonalized, in the application design. The Name of what was used is given with a description, where it was used and reference to where it was adapted from.

## 3.3.1 GridLayout with CardView

### Description

XML which creates a Grid Layout of 2 Columns and 3 rows of Card Views.

### Usage

Menu Activity.

### Reference

By (SoftCoding, 2020). Full reference in reference table.

## 3.3.2 Splash Screen

### Description

a graphical control element that consists of a window with a logo and the current software version. While the software is starting up, the splash screen appears.

### Usage

A splash page is a page that serves as an introduction to the "Community Shield Application."

### Reference

By (Palace, 2020). Full reference in reference table.

# 3.4 Problems encountered

This section covers problems in application development which were encountered. There is a heading to title what was being developed, underneath which there is a description of the problem, how a solution was sought and how the problem was eventually solved in the end.

## 3.4.1 Creating a Navbar

Created the Navbar UI successfully, upon adding the Kotlin code (so the navbar could navigate to various fragments) it became apparent that the application would crash. Attempted to trouble shott and even research what the problem could possibly be, because the code was all fine.

Exhausted all possible solutions I could find or think of and realized it would take up another week trying to figure this out, therefore I decided to move application navigation to the MenuActivity.

## 3.4.2 Fragment Utilization

Lacked the knowledge that was necessary to pass data from fragment to an activity and as a result struggled working with the back end of fragments. After having to scratch the Navbar due to a different issue, it became unnecessary to use fragments and as a result the working with fragments got scrapped from the application.

## 3.4.3 Attaching GPS Co-ordinates to SMS message

Struggled attaching GPS co-ordinates to SMS message as GPS co-ordinates were not of the same type “text”. Did same trial and error troubleshooting and after multiple trials to reach a solution, about 6 different iterations, I found a solution that works by placing the declaring the GPS co-ordinates value and SMS variable in the same function (“fun onLocationResult…”).

# 3.5 References

Ababei, A. (2018, July 24). *8 Reasons to Use Kotlin Over Java for Android Development - DZone Java*. Retrieved from dzone.com: https://dzone.com/articles/what-are-the-biggest-advantages-of-kotlin-over-jav

Contributor, G. (2019, October 2019). *4 key benefits of using Firebase for mobile app development*. Retrieved from Packt Hub: https://hub.packtpub.com/4-key-benefits-of-using-firebase-for-mobile-app-development/

Ghanchi, J. (2021, May 10). *The Major Advantages of Android Studio App Development*. Retrieved from Blog - IndianAppDevelopers - iPhone, Android, Blockchain, AR, VR: https://www.indianappdevelopers.com/blog/advantages-of-android-studio-app-development/

Palace, C. (2020, April 3). *How to create a Splash Screen in Android Studio (Kotlin 2020)*. Retrieved from youtube.com: https://youtu.be/WZWr0Abomfw

SoftCoding. (2020, September 17). *Modern Dashboard UI Design Android Studio Tutorial | Cardview Android Studio | Grid layout Android*. Retrieved from YouTube.com: https://youtu.be/ixRXEoGAEZM

Zitbek. (2019). *What are the main advantages of the Android application development?| Zibtek Resource*. Retrieved from Zitbek: https://www.zibtek.com/Advantages-of-native-android-application-development

Conclusion

# 5.1 Study conclusion

At the beginning of the project a project proposal was set out to design an application that would help users get a hold of urgent security emergency services/assistant without having to make a call, just by clicking a button and the emergency services will be alerted to come their way.

This was achieved along side some other objectives, however there were other improves that could have been made to improve the application. A search bar could have been added to both the News and Wanted Screens to allow filtering to find a specific news report or wanted criminal. A navbar would be more efficient to use than move navigation to an activity diagram. Chat functionality and ability to create online group chats would add greatly to the community aspect of the app. Finally, instead of GPS co-ordinates, the application to decode the exact co-ordinates in street, house/apartment number, city, and province.

Another key future that can be added is two other systems, a system for security/local authorities to use and a system for crime news reporters to use. The security/local authorities would use their system to receive notifications, but also to update the criminal wanted database. Criminal news reporters would use their system to be able to create, read, update or delete crime news reports for the users.

All in all, this project achieved its main objective, allowing a user to submit a “call” for security emergency and also brought a better understanding of what it truly means to develop a system of this type and scale. The “Community Shield” application also provided value in other aspects such as submitting tips offs and keeping the user informed about crime. In the future the app only gets better.