

FUNCTIONAL SPECIFICATION

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

CA420

Student Name: Cian McCann

ID: 11482422

DISCLAIMER

I the undersigned declare that the project material, which I now submit, is my own work. Any assistance received by way of borrowing from the work of others has been cited and acknowledged within the work. I make this declaration in the knowledge that a breach of the rules pertaining to project submission may carry serious consequences.
I am aware that the project will not be accepted unless this form has been handed in along with the project.

TABLE OF CONTENTS	PAGE
1. INTRODUCTION	2
1.1 Overview	2
1.2 Business Context	3
1.3 Glossary	3
2. GENERAL DESCRIPTION	5
2.1 System Functions	5
2.2 User Characteristics and Objectives	5
2.3 Operational Scenarios	6
2.4 Limitations	8
3. FUNCTIONAL REQUIREMENTS	8
Other Software Requirements	13
4. SYSTEM ARCHITECTURE	15
Diagram	15
5. HIGH LEVEL DESIGN	17
Context Diagram	17
Sequence Diagram	17
Dataflow Diagram	18
6. PRELIMINARY SCHEDULE	19
Gant Chart	20
7. APPENDENCIES	20

1. INTRODUCTION

1.1 OVERVIEW

This functional specification will contain a detailed overview of the Community Area Alert system. The aim of this report is to describe exactly what the Community Area Alert system is required to do and provide enough information on the software tools and packages which will be used so that the system can be designed to meet its requirements. I will provide a detailed description of each component in the system and their requirements, as well as a high level description of what the system should do.

The Community Area Alert system will be made up of a mobile application with a supporting desktop application. It will be developed to meet the current requirements of the community alert scheme as well as adding increased functionality so it has the potential to replace the current scheme. The mobile application will be developed using Adobe PhoneGap Build, this is a cloud service which will enable me to develop the mobile application using HTML5, CSS and JavaScript. By developing using PhoneGap build I can make the application available to Android, IOS and Windows smartphone users. The mobile application will work along with the desktop application to meet the requirements of the current system. This desktop application will be developed using Java Swing. Both the mobile and desktop application will connect to a MongoDB database hosted on MongoLabs.

Overview of the System

The current community area alert scheme operates in rural areas in Ireland. According to the schemes Director of Services which I interviewed it aims to “reduce the amount of crime and reduce the fear of crime.” Setting up a community area text alert scheme enables An Garda Siochana to send out a text alert to members of a community when they receive a report of a crime or suspicious behaviour. The desktop app will allow the Gardai to create new reports and select the area where the report came from, this will then notify everyone using the mobile app within a certain kilometre radius of the report.

Need for the System

According to a member of the Gardai I questioned, the biggest limitation of the current system is the lack of communities which have set up a text alert scheme. This is mainly due to the how complex it is to get the scheme up and running in a community. Some areas may contain up to 500 households, this requires at least 8 volunteers to call to each house with a consent form and then call to each house a week later to collect the form. It requires that one person keeps all of the consent forms and creates a full list of numbers from the area to pass on to the Gardai. The community then requires a launch night where they invite members of the Gardai, local papers and photographers as a message to criminals.

The cost of the text alert system at the moment is too high. Usually 85% of the houses in an area will sign up to the scheme. Each house will pay an average of 15 euro which will cover the cost of sending text alerts every 3-4 weeks for 2 years. However in areas when the text alert scheme is most successful it can cost up to 1200 euro a year for the Gardai to send text alerts through a service provider such as Eircom. According to James O'Neill (Director of Services) the more popular the system is getting the less sustainable it is becoming.

My system will aim to reduce the costs by sending mobile app notifications rather than text messages. It will also reduce the complexity by using GPS locations rather than setting up communities.

Main Functions

Mobile App:

- User account creation.
- Capture image of ID to confirm address.
- Retrieve local reports.
- Update report.
- Set preferences.

Desktop App:

- User Account verification.
- Create new report.
- Update report.
- Remove Report.

1.2 BUSINESS CONTEXT

This system has potential to upgrade the current Community Area Alert scheme. I had a long conversation with James O'Neill who is the Director of Services from Muintir na Tire. He worked along with the Garda Siochana to develop the current system, he gave me advice on the requirements the app will need to meet. He also said he will meet me in DCU and I can possibly present it to the Garda Community Relations office. Currently when a Community Area Alert scheme is set up it has the potential to reduce crime by up to 24% and increase drug detection by 14%. I hope my app can improve these figures by making the system more widespread and affordable for everyone.

1.3 GLOSSARY

Community Area Alert Scheme

Refers to the current implementation of the system which uses text alerts.

Community Area Alert System

The proposed system I will develop including a mobile app, desktop app and database.

Report

An alert created by the Gardai created when crime or suspicious behaviour is reported.

MongoDB

NoSql Database.

James O'Neill

Director of Services for Muintir na Tire. Helped to develop the current scheme by working alongside An Garda Siochana.

2. GENERAL DESCRIPTION

2.1 SYSTEM FUNCTIONS

- The system is made up of 3 main parts a mobile app, desktop app and a MongoDB database.
- Both the mobile and desktop apps will send information to the database.
- Both the mobile and desktop apps will retrieve information from the database.

Mobile App

- Provide User Interface.
- User Sign In.
- User Sign up.
- Capture ID image.
- Submit ID image (confirm address).
- Download current nearby reports.
- Send notification of new report.
- Increase threat level of report.
- Change order of reports.
- Add update to report.
- Send image update on report.
- Turn off notifications on report.
- Edit user account preferences.

Desktop App

- Provide User Interface.
- Admin sign in only.
- Create report.
- Update report.
- User account verification.
- Remove report.

MongoDB Database

- Store user accounts.
- Store reports.

2.2 USER CHARACTERISTICS AND OBJECTIVES

The desktop app will be developed for users with admin access only. Admin users will be made up of An Gardaí Síochána and a member of Muintir na tíre. Since I intend to provide a user interface that follows UI design principals even an admin user with basic computer skills should be able to accomplish the goal which brought them to use the desktop app.

The mobile app will be used by people of all ages who have a smartphone running Android, IOS or Windows. Community Alert aims to reduce the fear of crime in an area for everyone, according to James O'Neil "it especially helps elderly people to feel safer at night". The

mobile app will therefore have to be developed so that even people with little or no experience with mobile applications can receive important reports about a crime or suspicious behaviour.

Objectives of a user:

- Receive a reliable notification rather than a text.
- Be able to set preferences on how and when they receive notifications like other smartphone apps.
- Sign up and store a GPS location so they receive relevant updates from their area.
- An easy way to communicate with other members of the community.
- Feel safe in their home.
- Help to reduce crime in the area.

Requirements to make these objectives achievable:

- A stable database running 24/7.
- User preference settings within the mobile app.
- User account creation.
- User must have a suitable document containing their address which they can capture an image of to confirm where they live. E.g., driving licence.
- A text box under a report where they can add updates.
- An app they can rely on to notify them when An Garda Siochana issues a report of a crime or suspicious behaviour.

2.3 OPERATIONAL SCENARIOS

All bold text represents what the user of the mobile app will experience.

USE CASE 1: Registering

- **User opens the mobile application for the first time.**
- **User selects “Sign Up”.**
- **User enters information such as their name and address.**
- **User now has to confirm their address so criminals cannot abuse the system.**
- **The camera on the user’s phone opens.**
- **The user takes an image of a document with their name and address e.g. Irish Driving Licence.**
- **An message is displayed asking the user to “Please wait while an admin confirms your account”**
- **Notification is sent to the user indicating “Your address has been verified, you can now log in.”**
- **User logs in and can use the app.**

USE CASE 2: Failed Log In

- **User opens the mobile app and is not signed in.**
- **User selects the “Sign In” option.**
- **User enter a username and passwords and clicks “Confirm”**
- **One of the following is displayed:**
 - **“Username or password is incorrect”. With an option to enter their username and password again.**

- **"Your account needs to be verified by an admin before you can log in. You will be notified when this status changes."** With an option to press "Ok".
- **"The address on this account did not match the address on your registration photo. Please retake."** With an option for the user to "Retake" and send a new image.

USE CASE 3: Confirming users account

- An admin from Munitir na Tire logs into the systems email address.
- Opens a new email from a user who is attempting to confirm their address.
- The admin checks that the information on the picture of the driving licence matches the information stored on the users account on the MongoDB database.
- The admin logs into the desktop app.
- Selects "Register User" and enters the users account number from the email.
- The Admin confirms the users account.

Use Case 4: Editing a User Preferences

- **User selects "Sign In" and enters their username and password.**
- **User selects the "User Preferences" button.**
- **User selects GPS notification range (currently set to 3km).**
- **A drop box appears.**
- **User selects "2" from the dropdown box.**
- **User clicks confirm.**
- **The users account on the database is updated so they only receive notifications from reports within 2 kilometres from their house.**

USE CASE5 5: Report

- **A woman in a rural area has spotted a white van circling the road she lives on.**
- **The woman contacts An Garda Siochana and reports the suspicious behaviour.**
- The Guards feel like the people in the van could be watching a house for a chance to rob something and decide the community should be notified.
- An admin from An Garda Siochana logs into the desktop application.
- Selects "Create new report".
- The admin enters information such as a title ("Suspicious vehicle in area"), description ("White Van spotted multiple times circling."), threat level ("medium") and to allow community collaboration i.e. can they leave comments on the report and communicate through the mobile app.
- The admin puts a marker on a map to pinpoint the exact location where the woman last spotted the van and creates the report.
- **Everyone with the mobile app downloaded within the GPS radius they defined in their preferences (e.g. 2KM) will be notified.**
- John who lives down from the woman is notified on the mobile app. He is expecting a delivery and thinks it might be the courier.
- John opens the mobile app, selects the report called "Suspicious vehicle in area", he posts a comment saying **"Expecting a delivery soon, possibly a lost courier, ill check it out now."** **This comment is seen by everyone who was notified of the report originally.**
- John rings the courier service and finds out that the delivery man is lost in the area and cannot find his house.
- John rings the Guards and explains the situation.

- **An admin deletes the report which sends out a notification to users indicating the threat is over.**

2.4 LIMITATIONS

- Time.
- Cost (Adding the application to Google Play, App Store and Windows Store).
- Lack of developer experience with PhoneGap and MongoDB.
- Smartphones without GPS technology leading to users receiving non relevant reports.

3. FUNCTIONAL REQUIREMENTS

USER ACCOUNT CREATION

Description: The opening page of the mobile app will present the user with the option to sign up. Once they select this option they will be presented with a form where they enter information into the following fields:

- Name
- Password
- Confirm Password

Once they have filled out the fields correctly they can click “Confirm”.

This will bring them to the address page. They can enter their address into a form using Google Maps which will display a map of the area they entered. They are then required to place a marker in the exact location of their house. Once they finish they select “Confirm”.

The user will then have to confirm their address. The camera application on their phone will open and they will be asked to take a picture of a document which confirms their Name and Address e.g. Driving Licence. Once they select “Confirm” the image and the user’s information will be automatically emailed for confirmation.

Critically: It is essential so that:

- We can store an exact GPS location for each user so they can be notified about relevant reports in their area.
- User can edit and store their user preferences.

Technical Issues:

- Implementing Google maps.
- Assigning each user a unique ID so their account can be confirmed.
- Storing all of the information in a MongoDB database.
- Automatically generating an email and attaching an image.

Dependencies:

- Google maps API.
- GPS.
- User’s device having a camera. Otherwise they can manually create and send the email.

USER ADDRESS VERIFICATION

Description: Using information such as name and address entered by a user when they signed up for the mobile app to compare it to the image of a relevant document containing the same information. Once an admin has confirmed the information matches they can sign into desktop app. Select "User Verification", enter the users unique ID which is sent along with the email. Select "Verify User".

Critically: We can confirm they live in the area they selected to prevent a criminal registering an account in an area and receiving a notification when he is spotted committing a crime.

Technical Issues:

- Creating an email address to receive the user's information and matching image.
- Adding account information to the MongoDB database.

Dependencies:

- User Sign Up.
- Admin Sign In.

SIGN IN

Description: When a user opens the mobile or the desktop app they will be presented with the option to Sign In. The user will then be presented with a form containing 2 fields:

- Name.
- Password.

The user fills out these fields and then selects "Sign In".

If the information is correct the user is signed in and can use the application.

If the information is incorrect they will be presented the username and password field again.

They will also be presented with an option to reset their password.

Critically:

User must sign in to the mobile app:

- To allow us to notify them of relevant reports using their GPS location.
- To allow us to access their user preferences.
- To allow them to comment on reports which allow "Community Collaboration".
- Increase the threat level of a report.

Admins must sign in to the desktop app:

- To allow them to confirm a user's address which verifies their account.
- To allow them to create or view a report.
- To allow them to update or delete a report.

Technical Issues:

- Querying the MongoDB database.

Dependencies:

- Dependent on the user having already created and verified their account.

CREATE NEW REPORT

Description: An admin logs in to the desktop and selects “Create New Report”. A form is presented with the following fields.

- Title
- Description
- Threat level
- Expire After

They will then be presented with an option to allow “Community Collaboration” or not.

They will then be presented with a Google Maps form allowing them to enter the address the report came from, the admin then places a marker on the exact location where the issue was reported.

Once the admin selects “Confirm” the report will be sent out and relevant users will be notified.

Critically:

- Used to notify users of the reported crime or suspicious behaviour.
- Essential to select an accurate GPS location for a report.

Technical Issues:

- Implementing Google Maps.
- Storing a report in the MongoDB database.
- Notifying users that a new report has been created.

Dependencies:

- Admin signing in to the desktop app.
- Google maps API.

NOTIFICATION

Description: Mobile app should create a notification on a user’s phone if a report within their set GPS radius is created or updated.

Critically:

- Reduce crime by making people aware of a crime or suspicious behaviour.
- Reduce fear of crime.
- Replaces the original text alert system.

Technical Issues:

- Sending a notification to Android, IOS and Windows devices running the mobile app.

Dependencies:

- User being registered and signed in.
- User having a set GPS radius.
- User having notifications enabled for the mobile app.

VIEW REPORT

Description: When a report is created within the users GPS radius they can select the report from the mobile app. This will display information such as the:

- Time and date report was created.
- Title.
- Description.
- Threat level.
- Updates.
- Time of updates.
- Option to stop receiving notifications.
- Option to update if Community collaboration is enabled.

Critically: If a user receives a notification displaying the title of a report they can open the report to receive more information, leave an update, increase the threat level or turn off notifications from the report.

Technical Issues: Retrieving all relevant data on a report from the MongoDB database.

Dependencies:

- User must be registered and signed in.
- Report must have been created within the users GPS radius.
- The report could not have been deleted.

EDIT PREFERENCES

Description: Allow the user to set up the mobile app to suit them best. Allows the user to edit settings such as:

- GPS Radius: A user living in a town may only want notifications within 1 kilometre of their house. A user in a rural part of Ireland may want notifications up to 3 kilometres from their house.
- Time: They can select a time in which they do not wish to be notified such as between 11pm and 7 am.
- On/Off: For example if a user was going on holidays they could turn off all notifications of reports around their house.

Critically:

- Improves on the current scheme as James O'Neill identified notifying elderly people during the night as in issue.
- Increases the relevance of notifications.

- Improves user satisfaction.

Technical Issues:

- Storing user preferences in the MongoDB database.

Dependencies:

- On the user having already created and verified their account.
- On the user being signed in.

UPDATE REPORT

Description: An admin can Sign In to the desktop and update the information on a report. This can be done by selecting a particular report and adding a new comment to the description of this report using the comment box provided.

A user can Sign In to the mobile app and select a report they were notified about. If the admin allowed Community Collaboration on this report when it was created a text box will appear allowing the mobile user to type a comment a post it as an update.

Critically: Updating a report should notify all users who were notified when the report was originally created that new information has been added regarding the crime or suspicious activity.

Technical Issues:

- Updating the MongoDB database.
- Notifying relevant users of the update.

Dependencies:

- A report being previously created.
- Admin Sign in.
- User Sign in and a report having Community Collaboration enabled.

DELETE REPORT

Description: An admin can choose to delete a report using the desktop app. This will send a notification to users that the report has been removed and there is no longer a threat to the community. An admin can delete a report by signing into the desktop app, selecting a report and clicking on "Delete Report".

Critically: Some reports may be false alarms or if no new information is received for a number of hours it is essential that the report can be removed to reduce fear in the community.

Technical Issues:

- Notifying users in the area.
- Removing the information from the MongoDB database.
- If a user refreshes their list of reports the deleted report should be removed.

Dependencies:

- Admin signing in.

- A previously created report still being active.

INCREASE THREAT LEVEL

Description: Each report has a threat level. It is set by the admin through the desktop app when the report is created. When a user is notified of a report in their area they can view the report, this will give them the option to increase the reports threat level if they feel in danger.

Critically: Increases the attention of An Gardaí Síochána.

Technical Issues: Storing a threat level for each report and providing an interface option to increase it.

Dependencies:

- User must be registered and signed in.
- Report must have been created within the users GPS radius.
- The report could not have been deleted.

STOP NOTIFICATIONS

Description: Each user will have the option to enable or disable notifications on a particular report. This option will be displayed when the user views a particular report which they have been alerted on.

Critically: Some people may feel a report is not relevant to them and will not want to receive further notifications.

Technical Issues: All the reports a user unsubscribed from will need to be stored.

Dependencies:

- User must be registered and signed in.
- Report must have been created within the users GPS radius.
- The report could not have been deleted.

OTHER SOFTWARE REQUIREMENTS

Additionally to the standard functional requirements of the system I have identified other important system requirements and characteristics.

Correctness is a key characteristic of the Community Area Alert system. The mobile and desktop application will have to perform as expected consistently to gain and maintain the trust of users who have moved over from the previous text alert scheme. It is important that each user receives each relevant report in their area. This is also the reason why **reliability** is identified as a key attribute of the system.

Usability is also important so that users of the mobile and desktop app who do not have a lot of experience with technology can still achieve their goals. **Testability** of the app is very important so that even the first users who download the app can have a quality experience.

Flexibility has been given a high priority since the system will have to function the same across multiple mobile operating systems.

Since this is a stand-alone system with clear and specific goals **Reusability** and **Interoperability** will not be an issue during its development.

Efficiency is classed as a medium priority characteristic. It will be important that users receive a notification quickly after a report is created however I don't intend to spend much time on trying to reduce the standard time this takes.

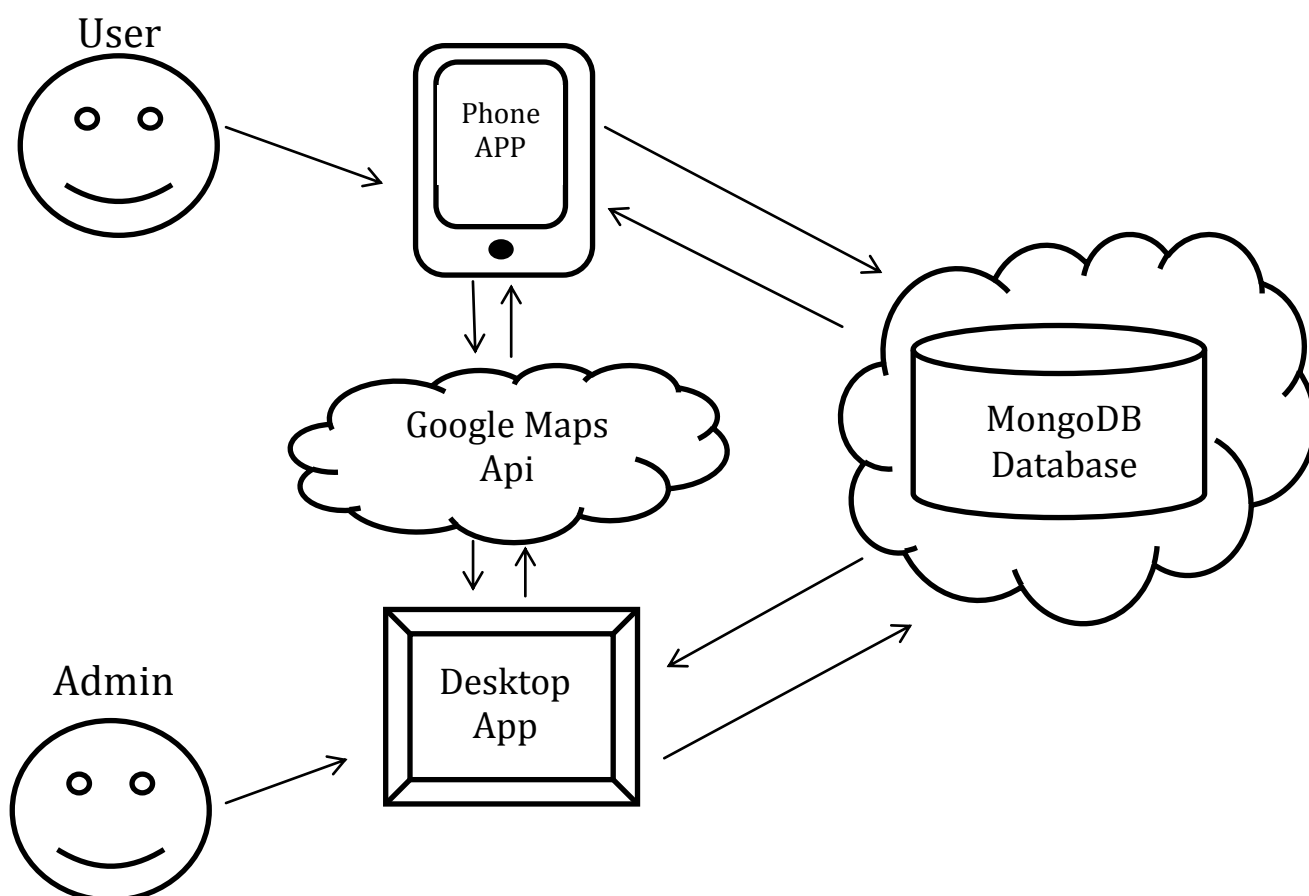
The **Availability** of the database is important. It should be available 24/7 from release to ensure users can have the best possible experience using our system. If the database crashes it should be recovered so the system can pick up from where it left off.

Security is an important characteristic of the system. The information on each account should be kept private with a password. It is also important from the point of view of confirming user addresses to reduce the chance of the system being misused by criminals.

The following table contains a full breakdown of the characteristics of the Community Area Alert system.

ID	Characteristic	H/M/L	1	2	3	4	5	6	7	8	9	10	11	12
1	Correctness	H										●		
2	Efficiency	M					●							
3	Flexibility	M								●				
4	Integrity/Security	M								●				
5	Interoperability	L	●											
6	Maintainability	M						●						
7	Portability	M								●				
8	Reliability	M										●		
9	Reusability	H	●											
10	Testability	M							●					
11	Usability	H										●		
12	Availability	M										●		

4. SYSTEM ARCHITECTURE



User: This is a member of the public who has downloaded the Community Alert Area smartphone application. They are running the app on an Android, IOS or Windows Mobile device. The Community Alert Area smartphone application will be available to download from either the Google Play store, Apples App Store or the Windows Phone store. Each user will have to create a registered account to access the features of the smartphone application.

Admin: There are 2 types of admins for the Community Alert Area system.

1. A member of Munitir na Tire who is in charge of this Community Alert Area system. They will have the ability to complete registration of new users who have emailed their information along with a verification photo.
2. A member of An Gardai Siochana. When they receive a report of a crime or suspicious behaviour which they feel the local community should be alerted on they will have the ability to log in to the desktop app of this Community Alert Area system and create a new report. Other functionality will also be provided which I mentioned above.

Desktop Application: This will be developed using java with the java Swing framework used to create the graphical user interface. It will communicate with 2 other systems:

1. **MongoDB database:** The desktop application from the Community Alert Area system will send and retrieve information to and from the database. An example is when a new report is created it will be stored on the MongoDB database, an admin can then retrieve the report at a later stage to update or delete it.
2. **Google Maps API:** When a new report is created the admin can enter the address where the report came from into a form which will use the Google Maps Api to return a map of the area. The admin can then place a marker on the exact location which will return a GPS location using the Maps Api.

Smartphone App: This part of the Community Alert Area system will be developed using Adobe PhoneGap Build so it can be made available on all major mobile operating systems. It will provide a user with an alternative option to joining a community alert scheme and paying for the text alert system while still meeting the requirements of the previous scheme.

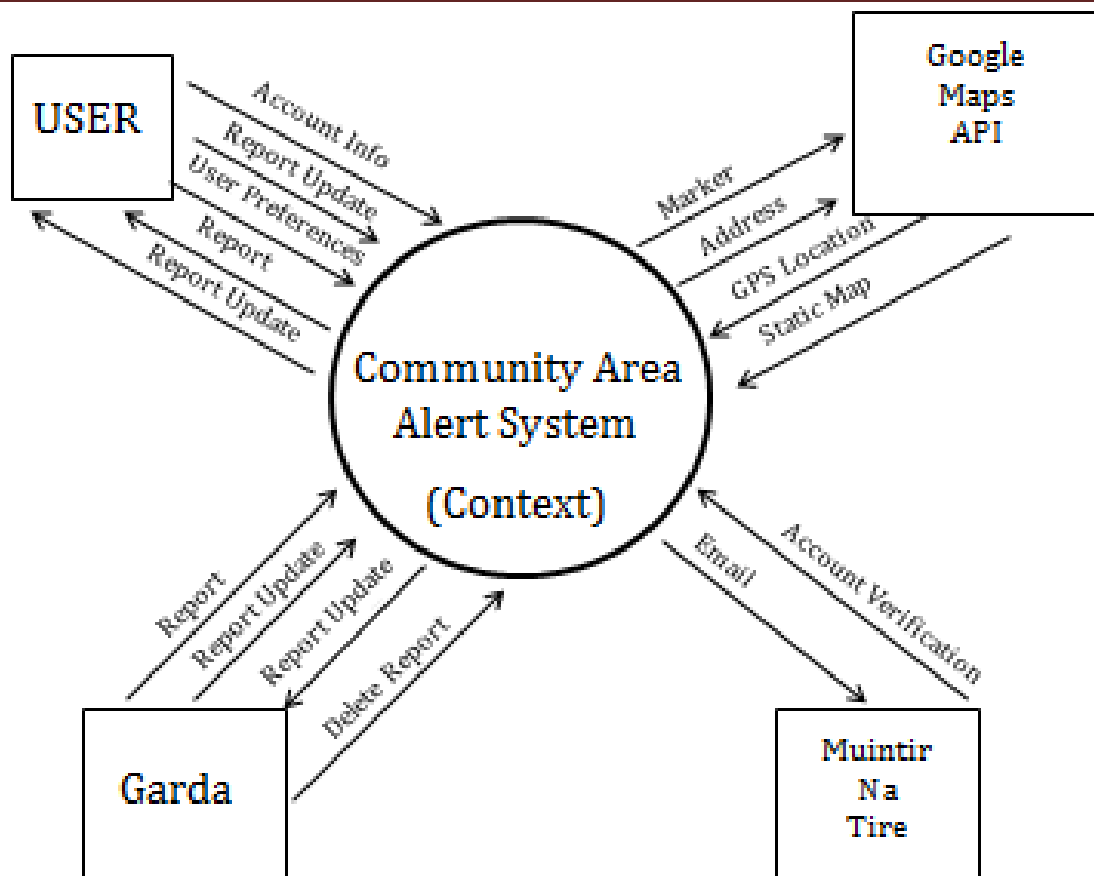
Google Maps Api: In order for this app to meet requirements it is essential to get accurate GPS coordinates. By using the Google Maps Api I can allow the user to enter an address which will return a static map of the area. It then has functionality to allow a user to place a marker on the map, by taking the GPS location of the marker a mobile user can generate an exact location of their house and an admin can generate the exact location of a reported crime or suspicious behaviour.



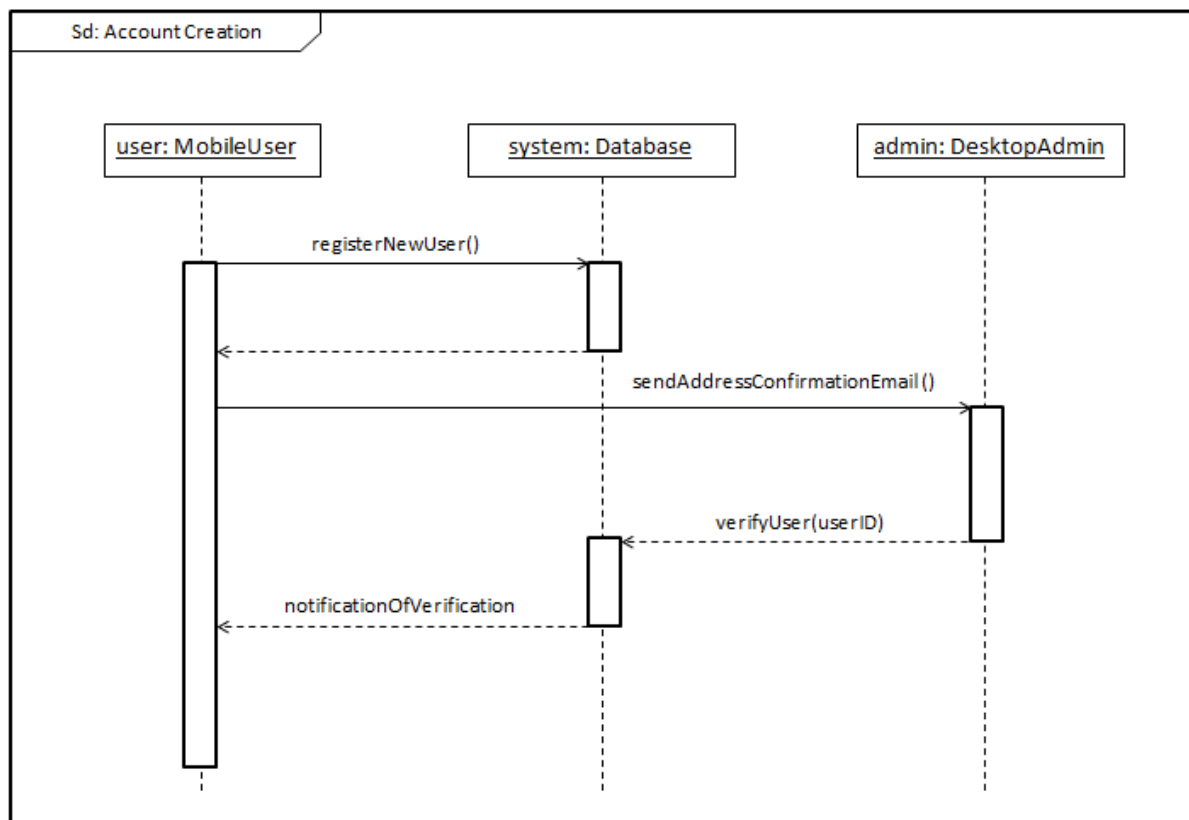
Database: In order to explore new technologies all of the data generated from the mobile and desktop apps from the Community Area Alert system will be stored in a NoSql MongoDB database. This database will send and receive information such as reports to and from the mobile and desktop apps. The database for this system will be hosted on MongoLabs

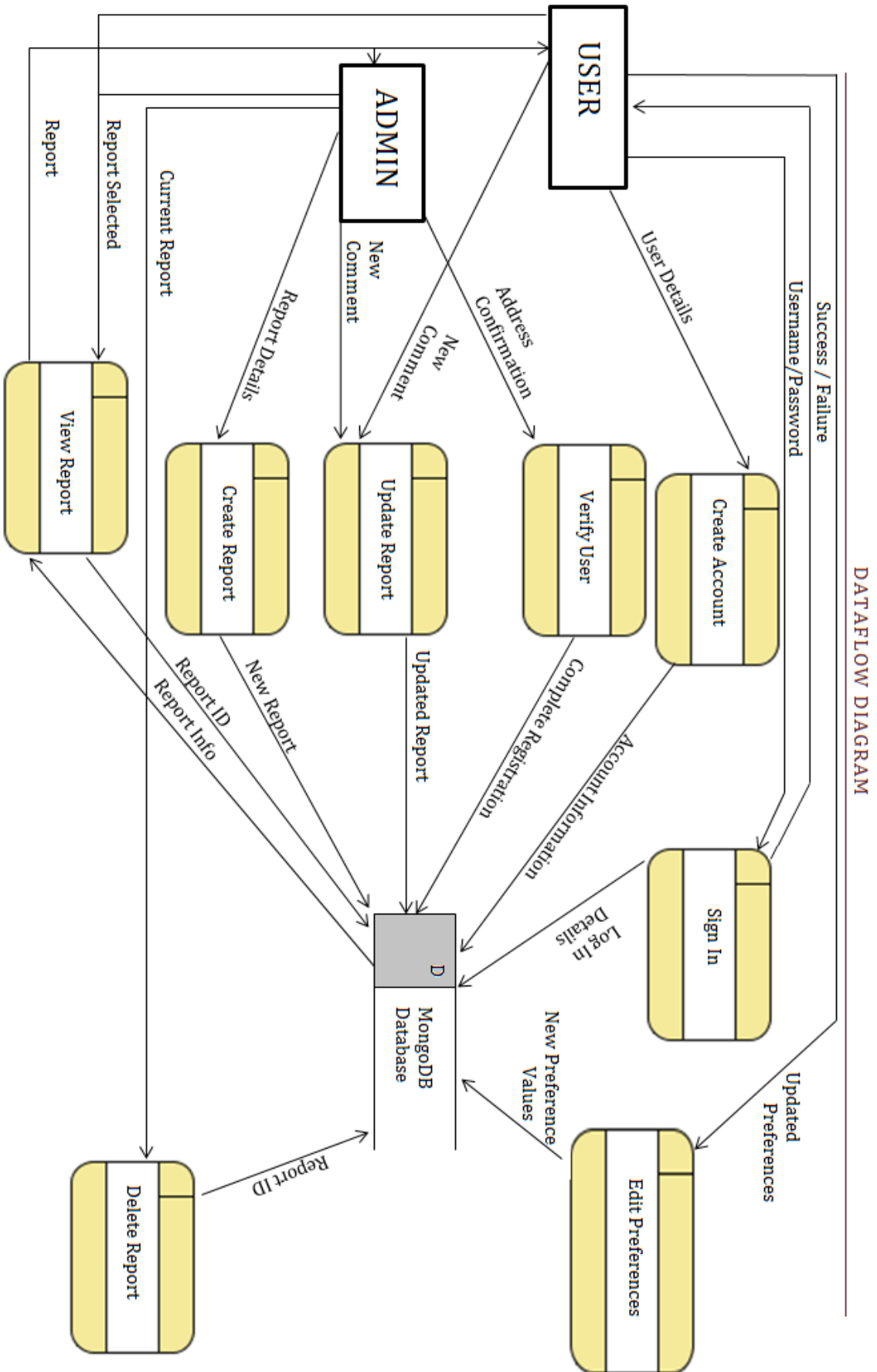
5. HIGH LEVEL DESIGN

SYSTEM CONTEXT DIAGRAM



SEQUENCE DIAGAM





6. PRELIMINARY SCHEDULE

1st – 8th of February

Implementing basic mobile app: I expect the first week of development will be slow as I will need to learn how to use PhoneGap and its deployment process. During the first week I intend to develop a basic application which can send and query data to and from a MongoDB database.

9th – 15th of February

Integrate user accounts: Add functionality to the mobile app allowing a user to create a new account and store their information to the MongoDB database. I will also add a form allowing the user to enter their name and password and confirm if the data entered is correct.

16th – 23rd of February

Verify account: This will involve allowing a user to use the built in camera on their device to take an image of a document confirming their name and address. The app should then automatically generate an email with the users account information and the captured image as an attachment. This is the first time I will be using 3rd party Cordova plugins with PhoneGap.

24th – 26th of February

Improve the interface: I will use proper GUI design techniques and standards to create and layout each page of the mobile app. This will involve retrieving a report I manually generated and displaying it correctly. It will also integrate buttons and pages for functionality which have not been added at this stage.

27th of February to 4th of March

Increasing functionality: Adding the ability for a user to update their user preferences, increase the threat level of a report, unsubscribe from a report and comment on a report.

5th – 8th of March

Finalize mobile app: Leaving some time at this stage to overcome any unresolved issues which popped up during development.

8th – 16th of March

Implementing basic desktop app: Get used to the java Swing framework. Create a basic application which can communicate with a MongoDB database.

17th – 21st of March

Add functionality to sign in: This will involve manually creating 2 admin accounts and adding a form to the desktop allowing sign in using the username and passwords of both accounts.

22nd March– 1st of April

Integrate report creation: Add the necessary user interface elements to allow an admin to create a report. This also includes integrating Google Maps. The information then needs to be stored in a way which allows it to be accurately retrieved from the MongoDB database.

2nd – 5th of April

Account Verification: Add functionality which allows an admin to complete the registration of a user's account. This will be used once an admin has checked the email to confirm the information on the image matches the information on their user account.

5th – 10th April

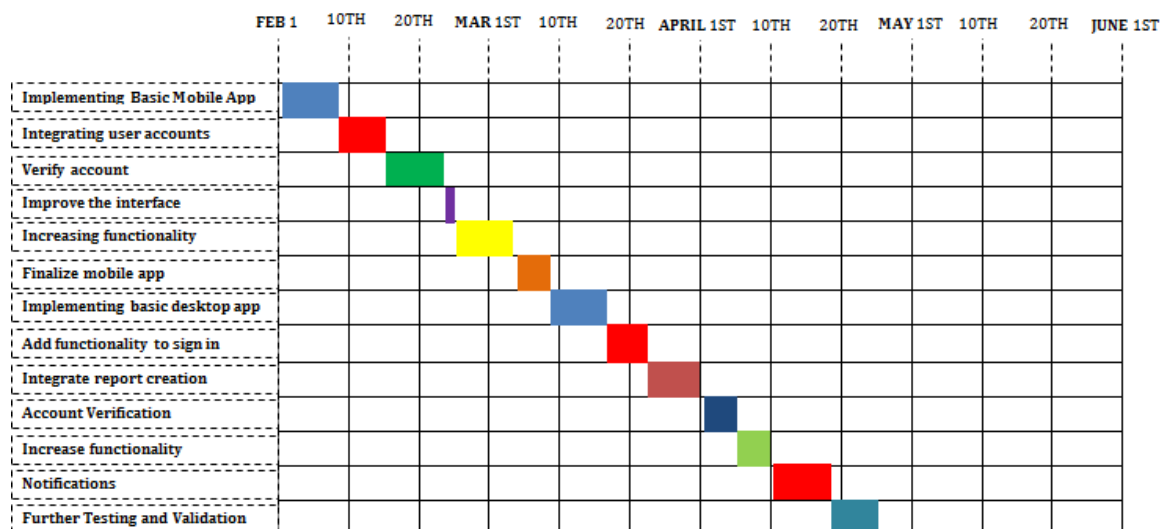
Increase functionality: Add the ability to delete or update a report. Use proper GUI design techniques and standards to create and layout each page and part of the desktop app.

11th – 17th April

Notifications: Ensure that the desktop and mobile app work together so that the mobile app receives notifications when a report is created, deleted or updated.

18th – 24th April

Further Testing and Validation



APPENDICES

It is important to note that if this system is never implemented to replace the current system, the functionality of the mobile app can still be increased to allow users to generate their own reports allowing it to function as a standalone smartphone application. However I intend to work along with James O'Neill to offer a viable solution to the current scheme which he said I can present to the Garda Community Policing board and hopefully get implemented.