Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Engineering

Building a Registry of CCTV Notices and their Privacy Practices in Dublin

Abhinav Sinha

Supervisor: Prof. Dave Lewis

April 17,2023

A Final Year Project submitted in partial fulfilment
of the requirements for the degree of
BAI (Computer Engineering)

Declaration

I hereby declare that this Final Year Project is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signed: Date:

Acknowledgement

Words cannot begin to express the immense gratitude to my supervisor Dr.Dave Lewis. His constant support, guidance and invaluable feedback were instrumental in the research and making of this project.

I would also like to thank the school of engineering, Trinity College Dublin for arranging a final year project which allowed me to learn, develop, understand and implement research methods and work on breaking new ground.

Lastly, I would like to thank my family, whose constant moral support motivated me in the development of this project.

Abstract

The increasing use of CCTV surveillance devices and practices like facial recognition have become common in public spaces. These practices raise pertinent questions regarding an individual's right to privacy.

Accordingly, an individual must be able to discern the kind of data being collected, processed and the purpose for which the collected data is being used for. As per the General Data Protection Regulation 2018 (GDPR), there must be transparency with the individual whose data is being collected and processed. Furthermore, the individual must have the right to access a copy of their personal data (Article 15 GDPR), thereby making it the organisation's responsibility to make this data available.

In this project, we take a look at the current legislation requirements, practices followed by organisations collecting CCTV surveillance data and improve data transparency with the intended user.

We also take a look at the current approaches taken by organisations to make their collected data more accessible. Current solutions are used as a basis for the project development and implementation while also taking a look at the shortcomings of current state of the art solutions.

This project finally develops a full stack system which consists of a publicly available registry consisting of the various locations of CCTV cameras, their privacy notices and the contact information of the organisation which collects surveillance data. This registry is used to present the user with a mobile application dashboard which allows the user to view these on a map. The dashboard allows the user to view the cross referenced CCTV location of cameras belonging to a variety of organisations and presents them with the contact information and associated privacy notices of the CCTV devices.

Contents

List of Abbreviations

List of Figures

List of Tables

1 Introduction

> "even if you're not doing anything wrong, you're being watched and recorded."
>
> — Edward Snowden
>
> (The Guardian (2013), Interview On NSA Whistleblowing)

With the increase in surveillance devices and data capturing, an individual's right to privacy has become a major concern. Technology to process data, Artificial Intelligence (AI) and Facial Recognition Technology (FRT) has improved exponentially.

Legislation like GDPR and Proposed Artificial Intelligence Act 2021 (AI Act) have various obligations which need to be followed by organisations who process the data collected from Closed-circuit television (CCTV) cameras or develop and use AI-based surveillance applications for biometric identification of individuals. In particular, we look at the privacy policies of CCTV installers and controllers[1].

These CCTV cameras are often discreetly installed. They may also perform real time facial recognition on the footage. Furthermore, the privacy policies are not easily associated with a given CCTV device, which makes it difficult for an individual to ascertain the extent of the data being collected and processed by the data controllers.

1.1 Motivation

In recent times, there has been a rapid increase in surveillance devices. From parks and malls to corners of the street. This added with the advanced FRT and processing of data being performed has led to a potent combination. Authorities assert that this is being done in order to protect citizens and prevent unauthorised/ criminal activities [1], however, questions arise pertaining to invasion of privacy, invasive surveillance and violation of an individual's fundamental right to privacy.

[1] Refer 2.1

1.1.1 Incidents pertaining to public surveillance

The AI, Algorithmic and Automation Incidents and Controversies (AIAAIC) is an independent, non-partisan, public interest initiative [2] founded in 2019. The AIAAIC keeps a track of AI incidents and reports around the world in the form of a public repository. At the time of writing, this repository consisted of reports ranging from invasive surveillance being carried out by domestic amazon ring [3] to wrongful arrests caused by inaccurate facial recognition [4].

Wrongful arrest leading to death

Labourer Mohammed Khadeer, 35, was misidentified by surveillance FRT leading to his wrongful arrest. As a suspect, he was detained and tortured in the Medak police station in Telangana state, India, for five days following which he succumbed to his injuries.

When it comes to CCTV surveillance, especially surveillance accompanied by FRT, biometric identification can be ambiguous sometimes. If individuals are more aware of these identification practices, it might lead to increased self awareness and lead the individual into making an informed decision (Refer 2.1).

Jaywalking incident in China

In the city of Shenzhen, China, local authorities have launched a new surveillance system which uses FRT to crack down on jaywalking as well as other crimes [5].

The authorities cross referenced the individuals from their database and displayed the information of such individuals on boards for public display. Not only were the individuals faces shown, but also sensitive information like their social identification numbers were also displayed.

This kind of public identification is extremely harmful and may not be in the best interest of the general population.

Automatic Number Plate Recognition (ANPR)

In Ireland, the national police and security service use ANPR. This technology automatically reads the number plates and speed of vehicles. This system is also capable of recording other offences such as dangerous driving and breaking red light [6].

This system allows for better monitoring and will help the police in identifying vehicles which are stolen, cited or have been used to break a traffic law.

Conversely, this system showcases how easily identification of a vehicle can lead to the automatic ticket being generated for a violation due to active surveillance.

1.1.2 Potential use of CCTV surveillance data

Currently, FRT and surveillance equipment is not being used to an extent which may cause any issues to people in the short term. However, in the long term, this data can be used for other purposes.

Social Credit System (SCS) in China is being applied in an increasing number of areas of everyday life [7]. The system uses rewards and punishment to specific human behaviour in order to calculate their social credit.

Considering the above mentioned credit system and taking it a bit further, mass surveillance and constant monitoring of a person's behaviour and social misdemeanours being used for alternate purposes. The system could be used to determine a person's insurance premiums and calculate their credit scores.

Extending this even further, Deoxyribonucleic acid (DNA) data banks are currently used for the identification in criminal investigations [8]. However, this data can be used to link future generations of family members to crimes committed by their ancestors. Considering an individual's right to privacy, they may not want such links to be stored permanently (in some cases extending to a couple of generations). An individual must have transparency when it comes to their collection and processing of their personal data and at any point of time, there should be an option for consent withdrawal.

1.1.3 Public concern regarding surveillance

Due to the nature of surveillance data, there has been an uproar when it comes to the kind of collection and processing which is performed on surveillance data.

Data Protection Commission (DPC) Blogs

The DPC has a dedicated public blogs section which showcases various viewpoints of the general public. In one instance, a person did not get the CCTV recording for an incident where the individual was identifiable. This particular incident raised concern with respect to discovery and the right to access one's own personal data. [9]

Access Now

In an open letter published by **Access Now**, there was a call for the global ban on biometric recognition technologies that enable mass and discriminatory surveillance [10]. As stated in the open letter "the use of such technologies in publicly accessible spaces is incompatible with our human rights and civil liberties and must be banned outright for good". [10]

University College Dublin (UCD) Open Letter to Irish Times

Another open letter addressed the issue of mass surveillance leading to the identification and tracking of individuals without warranted suspicion. FRT at this scale will lead to powerful inferences amongst which a large number of individuals have no interest whatsoever. Inevitably, it

referred to a chilling effect, altering how people use public and online spaces [11].The open letter consists of 53 signatories, each being distinguished academic professionals in the privacy domain.

1.2 Research Objectives

This project aims to develop a user-friendly application dashboard which enables the user to detect and locate the CCTV cameras installed on a map. Additionally, the application also displays the contact information of controllers of such devices, thereby, increasing user transparency as per GDPR. The project objectives are as follows:

1. To identify the types of data collection in CCTV privacy notices.
2. To collect real life examples of such CCTVs and build a publicly available registry.
3. To create an easy to use and open source dashboard which provides the CCTV information and its usage.
4. To develop a crowd sourcing solution for the gathering of CCTV devices and their notices.
5. To design, implement and test a full stack application, learn about REST API backend development and cross platform mobile application development.

1.3 Report Outline

Chapter 1 The reader is greeted with an introduction to the research project. It flourishes the reader with the motivation behind the research project, the research objectives and overall layout of the report.

Chapter 2 This chapter explains the various terminologies and provides a summary of the reviewed literature, and the state of the art in this research space.

Chapter 3 Elaborates on the design process, requirements and considerations made for the implementation of this project and the overall system architecture.

Chapter 4 This chapter explains the details of the implementation of this project. It includes dependencies and their installation, source code snippets and artefacts.

Chapter 5 Evaluates the overall project, testing methodologies and overall outcomes of the project.

Chapter 6 Addresses some of the other aspects of the project like ethical considerations and avenues for future work.

Chapter 7 Presents the reader with the overall report conclusion and discusses the various explored aspects throughout the course of this project.

2 Background and related work

2.1 Related concepts and definitions

Before going further, it is imperative to understand some aspects of the entities involved when it comes to data handling.

Personal Data Any kind of information regarding a living person (such person is identified as data subject) which can be used for identification based on physical, physiological, genetic or other such identifiable traits [12].

Data Controllers A "data controller" refers to a person, company, or other body which decides the purposes and methods of processing personal data [12].

Data Processor A "data processor" refers to a person, company, or other body which processes personal data on behalf of a data controller [12].

Consent Personal data can undergo different levels of processing based on the individual's consent. Under the GDPR, consent to processing must be freely given, specific, and informed.

An individual cannot be coerced to give consent and explicit details regarding the use of data must be provided. Only after lawful consent, can processing on personal data be performed [12].

Data Protection Officers (DPO) The GDPR requires data controllers and data processors to appoint a DPO in certain circumstances [12]. The DPO is the responsible authority for monitoring GDPR compliance, cooperation with authorities and record keeping[1].

Informed Choices Since processing of personal data requires consent, this raises the concept of informed choices. When an individual is accommodated with all information, which includes the kind of data being used to process, the level of processing being performed, then the choice made by the individual is known as an informed choice.

[1] Other responsibilities : `https://www.usoft.com/blog/five-main-tasks-of-the-data-protection-officer`

2.2 Related Research

2.2.1 Related Legislation

GDPR 2018

The legislation requires increased transparency (Article 12 GDPR) and ensures access (Article 15 GDPR) of collected data to the data subject. In terms of surveillance, this gives the data subject the right to request information about the surveillance footage, the retention period of collected surveillance footage and the kind of processing that is being performed on this surveillance footage.

Proposed AI Act 2021

Modern surveillance devices and data processing go a step beyond. FRT and feature detection may be performed by the data controller. Depending on the associated risk level, the proposed AI Act ensures that a system does not violate an individual's fundamental rights and the open availability of such information be available for public scrutiny [13].

2.2.2 Literature Review

Dark Patterns After GDPR

Since the inception of GDPR there have been several Consent Management Platforms (CMP) which have been introduced to the web to conform with the European Union (EU)'s GDPR. This study refers to the consent when companies collect and process user data. The study scraped the designs of the 5 most popular CMP on the top 10,000 websites in the UK. The dark patterns and implied consent were ubiquitous. Only 11.8% met the minimum requirements based on European Law. The study also discussed the effect of changing the styles of consent forms on the website [14].

Drawing a parallel with surveillance, an individual walking by surveillance equipment is unclear of the data collection practices in place. The user must be provided with adequate information for them to make an informed choice regarding the processing of their data.

Irish Document on community CCTV

With the miniaturisation and development of technology [1], surveillance is only going to get more invasive. A decrease in crime rate is expected as the level of surveillance increases, however, several areas in the UK and Ireland have shown that with increasing surveillance, it does not necessarily imply a reduction in crime rate (in some cases, it was found to increase). Regardless of the use of collected surveillance data in prosecuting crime, it is a fact that the community is subject to extensive and invasive surveillance without the burden of proof that it reduces crime rate.

With time, increased sophistication in surveillance techniques will only result in greater invasion of an individual's privacy.

2.2.3 Related Work

After the inception and adoption of the GDPR, there have been great steps taken to ensure transparency for the data subject[2]. This has been followed by some excellent research being carried in the privacy domain.

Data Protection Vocabulary (DPV)

The DPV enables expressing machine-readable metadata about the use and processing of personal data based on legislative requirements such as the GDPR [15]. Essentially, it provides a structure for storing the why/who/where with respect to an individual's personal data. The DPV standard could help in the context of privacy notices. It uses a combination of concepts and relations to store the privacy notice data.

Basically, it helps in the structured storage of permissions requested by a service/authority in their privacy policy document.

Consent Right

Developed in a similar space, the Consent Right project [16], aims to improve consent management on the internet. The project developed a browser extension which summarises the various cookies and permission requested by a website and assesses the compliance standards of websites. It also has a feature which allows the user to toggle and disable the website cookies.

This project increases the user transparency with respect to privacy cookies and also alerts the user of websites that do not conform to the regulations laid out in the GDPR.

2.2.4 State of the art

Hidden Camera Detector Applications

Currently, there are various camera detector applications available in the market namely, Hidden Camera Finder(IOS Platform), Hidden Camera Detector(Android Platform). These applications are based on :

1. Detection based on electromagnetic field
2. Detecting light reflection from lens
3. Detecting InfraRed equipped cameras
4. Detecting bluetooth camera devices
5. Detection based on Wireless Fidelity (WiFi) camera on network

These applications rely on the sensor of the device to capture any kind of traffic/signal disturbance [17]. These applications have proved their efficacy by detecting "Spy" Cameras which are well hidden and extremely difficult to notice.

[2]Individual whose personal data (Refer 2.1) is collected

Internet of Things (IoT) based camera detection

Another approach along similar lines, this paper elaborates on detecting a hidden camera which may be recording and transmitting the recording over the network. It proposes the use of an arduino to capture the signal and jam the signal, thereby preventing transmission [18]. This not only detects the hidden camera, but also prevents forwarding of unauthorised recording[3].

Publicly available datasets

Currently, websites like GeoHive [4] and SMART Dublin [5] publish the location of CCTV cameras. These datasets are available in multiple formats and can be viewed in the inbuilt browser.

Despite this availability, they have the following drawbacks:

1. The datasets consist only of CCTV locations installed by certain authorities like the Dublin City Council (DCC).
2. They lack additional information like associated privacy notices, contact information and other such details
3. These datasets do not contain any information of domestically installed surveillance devices.

There is a lack of a user oriented approach which allows the quick detection of surveillance. Finding the location, privacy notice and contact information of CCTV surveillance devices would promote user transparency. This will also enable the user to carry out data access requests more proactively, should they wish to do so.

[3] recording without consent
[4] GeoHive Link: `https://www.geohive.ie/`
[5] SMART Dublin Link: `https://smartdublin.ie/`

3 Design and methodology

3.1 Design Thinking

In order to design this project, the design thinking methodology [19] was utilised. Design thinking process describes a structure which comprises of 5 phases namely **empathise, define, ideate, prototype and test**.

3.1.1 Empathise and define

The intended user, who walks in public streets is unaware of the various surveillance devices which are surveilling and recording the user. The application should provide a convenient and easy to use interface for the user to be able to locate these surveillance devices.

3.1.2 Ideate

The application should consist of the location of installed CCTV devices and present them to the user. Since location data is being dealt with, the easiest way to represent the same would be on a map.

3.1.3 Prototype

The application requires structure and needs a User Experience (UX)/User Interface (UI) design which shows the presentation and arrangement of data in the application for the actual implementation of the application.

3.1.4 Test

The application will be tested based on the handling of CCTV locations. As the application will be a full stack application, the components will require unit testing and overall integration testing.

3.2 Requirements

3.2.1 Functional Requirements

CCTV Location Pins The application must show the CCTV location marker. From the user's perspective, it will be easiest to view these markers on a map embedded in the application.

Information box The application should generate an information box based on the selected marker. This would include contact information like email address and phone number of the CCTV controller.

Redirect Privacy Notice The information box should contain a button that can redirect the user to the relevant privacy notice for further deep dive.

Crowdsourcing Since the data points are not available at any one place, for improved data management, we can harness the power of crowdsourcing. This should include the uploading of photos of the specific CCTV devices.

3.2.2 Non-functional Requirements

Consistent and reliable Data should be consistent and the system must be as reliable as possible.

Ease of use The application interface must be easy to navigate for the intended users.

Performance The application should be light and responsive.

Open source As the application works in the privacy domain and aims to improve transparency, the project will be developed in an open source environment.

3.3 Use Case

Use case Diagram

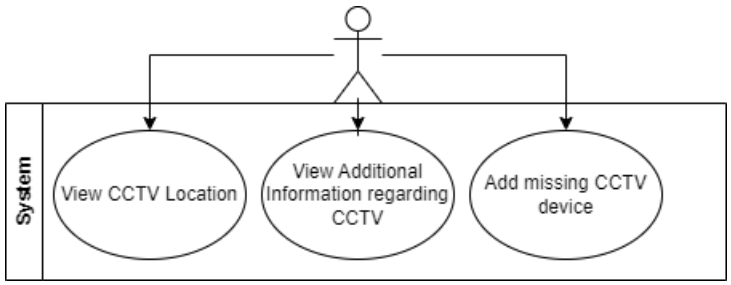


Figure 3.1: Use case diagram for the System

The above use case, generated from the list of functional requirements, showcases the various aspects and defines the use case of the overall system.

Primary Actor An individual itinerant from place to place.

System CCTV surveillance detection application.

Stakeholders User, CCTV controllers, data processor.

Use Case Goal For the user to detect and locate surveillance equipment in their vicinity.

Precondition The user must be present in an area where surveillance is being carried out.

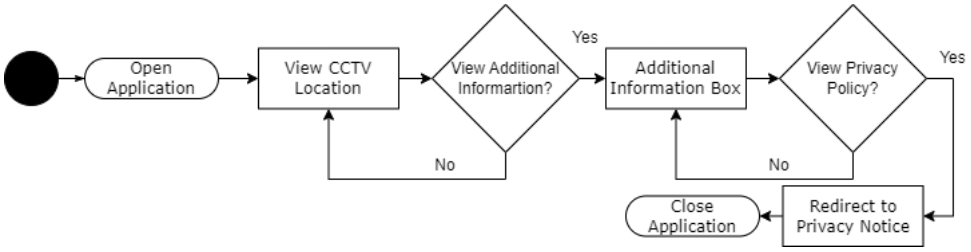


Figure 3.2: Activity Diagram for Flow 1

Flow 1 A user walking in a public space can use the application to detect nearby surveillance equipment. Upon opening the application, the user can:

1. View pin locations which show the exact locations of surveillance devices.
2. Upon clicking on a particular pin, the user must be able to view additional information about the device, mainly the contact information regarding the surveillance device.
3. If the user wishes to explore further about the kind of data processing, collection and retention, there is a button which takes the user to the respective privacy notice.

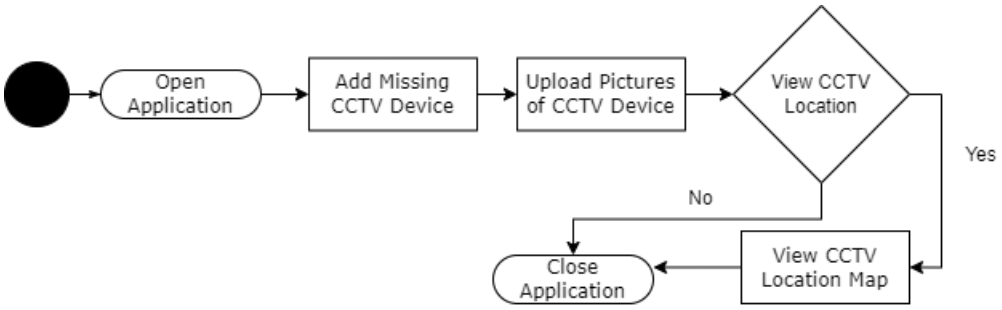


Figure 3.3: Activity Diagram Flow 2

Flow 2 In case the user decides to upload a missing CCTV device, the following flow will apply

1. A user finds a CCTV surveillance device not listed in the application.
2. The user uploads the data point along with pictures of the CCTV device to be added to the database.
3. The user can return to viewing the CCTV locations or close the application.

3.4 Technical Overview

In this section, we discuss the outline of the various tools and subject knowledge that was required to develop this project. The application was designed to be a full stack application. It consisted of a backend server which would handle all the requests and a frontend dashboard for user interaction.

As the application was location based and needed nearby CCTV devices, it made sense to proceed with a mobile application for the front end. Although a website could have been used to view the CCTV location markers, the interaction and location data would have been better in a mobile experience.

The development of the prototype was performed on **Figma**. This was crucial as it allowed for design changes which could have been performed without the creation of the entire application.

As for the data cleaning and manipulation, **Microsoft Excel** was used to make changes to CSV files.

In order to create the full stack application, the development process required a good understanding of **Javascript**, **JavaScript Object Notation (JSON) Objects**, **REST APIs**.

In order to create the front end of the application, a working knowledge of **Dart**, **flutter**, **Object Oriented Programming (OOP)** concepts and **cross platform** application development were required. The flutter documentation[1] proved to be an excellent resource for the mobile development aspect.

In order to assist the development process and make the source code open source, other Version Control System (VCS)/house keeping tools such as **git** and **github** were used.

Visual Studio Code was chosen as the preferred development environment for this project as it contained a plethora of plugins and packages which assisted and streamlined the overall project development.

Lastly, for testing, **postman**, **ngrok** and additional flutter packages were used.

[1] Refer A1.4.1

3.5 Implementation Considerations

To address the functional requirements, several key decisions had to be made. This section describes these key decisions which were made and alternatives to achieve the requirements.

3.5.1 Data

In order to show the locations of CCTV devices on a map, the latitudes and longitudes of the devices have to be known. Fortunately, the following public resources for available for this purpose:

GeoHive The GeoHive website consisted of various datasets. Unfortunately, they were not suitable for the purposes of the application

SMART Dublin The SMART Dublin website had open repositories which consisted of CCTV location datasets. These datasets mainly comprised of CCTV devices installed by the DCC

After analysing both resources, the SMART Dublin website was chosen. The datasets were better formatted and suitable for the application development process.

Cleaning Data

The initial dataset[2] consisted of CCTV ID, SiteID, CCTV name, latitude and longitude of the installed locations. This was an excellent starting point for showcasing these points on a map. The dataset consisted of 825 entries. These 825 entries were converted to JSON format using an online conversion tool[3] and a data_original.json file was created.

_id	SiteID	Site_Description_Cap	Lat	Long
1	1	ABBEY ST @ MARLBOROUGH ST (LUAS)	53.348754	-6.257607
2	2	AMIENS ST @ SEVILLE PL	53.354711	-6.246679
3	3	AMIENS ST @ TALBOT ST	53.351374	-6.250073
4	4	ANNESLEY PL @ POPLAR ROW	53.361145	-6.240513
5	5	NCR @ AUGHRIM ST	53.355797	-6.292406
6	6	BAGGOT ST BRIDGE	53.334055	-6.245199
7	7	BAGGOT ST @ FITZWILLIAM ST	53.336815	-6.249116
8	8	BAGGOT ST @ WATERLOO RD	53.333171	-6.243319

Figure 3.4: Original CSV source file

In the above figure, only 8 entries have been shown to illustrate the fields in the CSV fields.

[2]Link to Original Dataset : `https://data.smartdublin.ie/dataset/traffic-signals-and-scats-sites-locations-dcc/resource/400bb920-0263-4573-b920-0d0be005eab1?view_id=a92e7ad9-e408-4f42-8344-e26b9da7f916`

[3]Link to Conversion Tool : `https://csvjson.com/csv2json`

cctv_id	cctc_road	lat	long	policy_url	Phone	Email
39	Harcourt St	53.333965	-6.263233	https://www.cctvireland.ie/pub	01 6663805	Null
52	Parnell St	53.350357	-6.266422	https://www.odce.ie/Portals/0/	Null	info@odce.ie
203	Richmond St South	53.330227	-6.264374	https://www.richmond.gov.uk/(	020 8831 6001	
204	Ranelagh Rd	53.330338	-6.259884	https://www.ranelagh.bonitas.(	0303 123 1113	
218	Charlemont Place	53.330822	-6.258903	https://www.claytonhotelcharle	+353 (0)1 960 6700	res.charlemont@claytonho
237	Merrion Sq North / I	53.341104	-6.250811	https://www.nationalgallery.ie/	+ 353 1 661 5133	
276	Stephen's Green	53.337704	-6.262714	https://stephensgreen.com/priv	+353 (01) 4780888	info@stephensgreen.com
280	St. Stephen's Green	53.33989	-6.260723	https://stephensgreen.com/priv	+353 (01) 4780888	info@stephensgreen.com
282	Dawson Street	53.341327	-6.258309	https://www.ria.ie/sites/defaul	00 353 1 6762570	dataprotection@ria.ie
286	College Green	53.344643	-6.259576	https://www.tcd.ie/about/policies/cctv-policy.php#s		dataprotection@tcd.ie

Figure 3.5: CSV file after adding requisite fields

Adding Fields

In order to fulfil the project objectives, the following additional fields were required:

Contact Information The application had to display the contact information of the installed CCTV device. This required additional fields like email address and phone number of the CCTV installer/controller.

Privacy Notice In order to understand the kind of data being collected and processed, the privacy notice associated with a particular device had to be examined. This privacy notice should be available to the application user via a button click. In order to achieve this, the URL of the privacy notice was added as a field to the CSV file.

Since it was not possible to manually search and add the data in the related fields for all the 825 entries, 10 entries were shortlisted and a data_modified.json file was created

3.5.2 Schema Modelling

For the storage of the above dataset in the database, we needed a schema.The schema was to be designed based on the CCTV entity. Accordingly, based on the problem, the following class diagram for the schema was created:

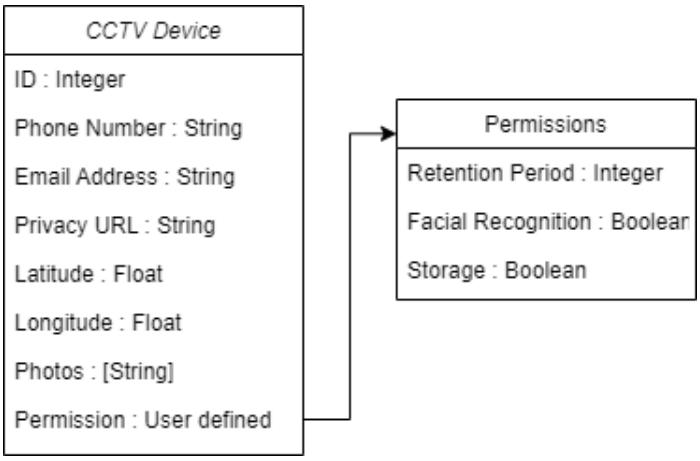


Figure 3.6: Schema model represented in class Diagram

3.5.3 Analysis of Privacy Policies

An important consideration was the permissions field. Not all privacy notices were created equally. Some were available as a Portable Document Format (PDF) document, whereas others were websites. Each of these privacy notices had to be examined in order to extract the requisite information. This included the retention period, processing of data, handlers of data and other such vital pieces of information. For this, we look at some of the parsing strategies that could be used.

Machine Learning (ML) Parser An ML parser could be executed on the privacy policies which would go through the privacy policies and extract key pieces of information. Since there is no common structure to Privacy policies, the execution of such parsers would require extensive corner case handling and exception management.

DPV compliant Policies DPV provides a proper structure for the meta data in privacy notices. Instead of websites and PDFs, if the notices were to use the DPV methodology, the program could comb through these DPV compliant notices and extract the relevant clauses for the application.

An example of a privacy policy[4].

3.5.4 Crowdsourcing

Many tasks in data management cannot be machine automated and manual execution of said tasks is not feasible. Crowdsourcing technique is an effective method of harnessing the human cognitive ability [20] and creating an effective database based on crowd input.
The crowd sourcing approach was mainly used to solve the following challenges:

Privacy Policy The search and association of every single privacy policy is troublesome. In case of crowdsourcing, the end users can upload a new CCTV datapoint and also upload the associated privacy policy.

Photographs It is difficult to locate a CCTV camera solely based on location markers. While adding a datapoint, the users can click photographs of the CCTV camera and upload it. The photographs make it easier for the end user to see where the camera is installed.

Contact Information The contact information of a data controller can be crowdsourced in a similar manner.

In order to resolve the data harnessing problem, crowdsourcing was used as a solution, however, this came with its own set of challenges.

Suppose, a number of users have uploaded a CCTV camera picture. Each user has taken a picture from a different angle, different distance and using a different camera. These location points form clusters around a CCTV camera which can be visualised as clusters on a map.

[4]**St Stephens Privacy Policy :** `https://stephensgreen.com/privacy-policy/`

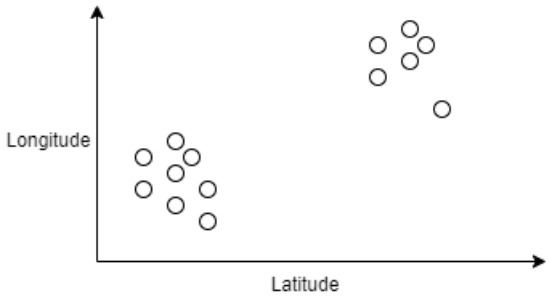


Figure 3.7: Raw received unclustered input

Based on this user input, the inputs need to be resolved into two data points and added to the database. For this to work, the latitudes and longitudes of a cluster need to be resolved into two centres (for the 2 data points).

Clustering[5]

Fortunately, geo clustering is a common issue and there are a couple of algorithms that can be used to resolve these geo spatial clusters. The following two algorithms can be utilized:

K-means Algorithm The k-means clustering [6] algorithm is traditionally used in the context of Machine learning. Specifically, it is used in the context of unlabeled supervised, unsupervised learning [21]. Despite its application in machine learning, it is useful in clustering of geo spatial data. As the data points consist of latitude and longitude, we can apply k-means to cluster and find centroids of our crowdsourced data.

DBScan Algorithm Similar to k-means clustering algorithm, DBScan clustering algorithm is a machine learning clustering algorithm. The k-means clustering algorithm has a couple of disadvantages and despite good performance, it is not very suitable for geo spatial data. DBScan is a density based clustering algorithm that has shown to work well in geo spatial clustering.

After the application of above algorithms, we get two clusters each with a centre. After the resolution of data points, the **top-k**[7] [20] highest rated data points can be added to the primary backend.

[5] **Note:** Due to the extensive scope of clustering and time constraints for this project, this aspect could not be implemented in due time.

[6] K-means Algorithm: `https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67`

[7] top-k is a crowdsourcing selection technique wherein data points with highest votes are inserted into the database

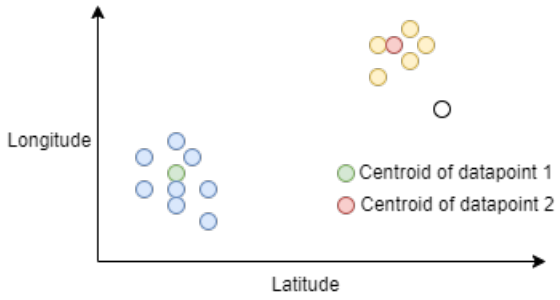


Figure 3.8: Clustered Data after processing

Despite the resolution of new data points, these algorithms do not produce perfect output. They are suitable for well defined data points, however, in case of crowdsourcing, the data can be quite random.

Crowdsourcing alternatives

Crowdsourcing is an excellent technique when it comes to data management, however it has certain drawbacks [20]:

Error Prone Even the best crowdsourced databases are prone to incorrect information as it is based on user input. This results in a noisy collection which may not be the most reliable

Diverse Since the harnessed data is collected from various sources, there is a certain level of discrepancy to be expected, especially when it comes to spatial data.

In order to ensure that we have more reliable information on hand, we can allow the CCTV controller to upload their data.

Alternate Approach

This approach is implemented by simply generating an authorised API key that can be used by the CCTV installation authorities or the CCTV controllers and upload the CCTV device data point. This has the following benefits:

1. The datapoint will not consist of any incomplete attributes as the CCTV controller has uploaded the datapoint.
2. The datapoint is accurate and will not require any kind of crowdsourced pre processing for it to be added to the database
3. As the data has been collected from the point of source, there will be no discrepancy with respect to the data.

3.6 Technical Considerations

3.6.1 Database design

As this project is in its infancy, a rigid schema would have increased development and testing time. Future changes to the project would also have become difficult. In some cases, like the permissions attribute, it would be useful to have a key-value association.

In order to alleviate some of these concerns, NoSQL databases were the ideal choice for this project. Amongst the available NoSQL databases, MongoDB was chosen as the NoSQL database.

MongoDB provided valuable development resources, community and provided free cloud hosting for the database. The mongoDB cloud hosting service, Mongo Atlas allowed the project database to be hosted without any setup in the local environment.

3.6.2 Backend Design

The back end for this project was to develop a REST API server. A REST API server sends data simply in the form of JSON. This would enable other developers to use the backend as a public registry and incorporate it in their own application.

For the purposes of quick development and deployment, Javascript and consequently, the **node.js** and **express.js** technology stack was used.

The backend design was challenging. Essentially, there were two operations that needed to be performed, the reading of CCTV devices and the handling of crowdsourced input. There were concerns regarding:

Data Integrity The crowdsourced data had to be verified before it would be used in the application.

High number of operations A single backend would have to handle a high number of read/write operations

In order to alleviate these concerns, a 2 backend approach was used. It has the following advantages:

1. The primary backend acts as a single source of truth.
2. All the processing takes place on the crowdsourced database which can be assigned a more powerful system.
3. Separation of backends reduces the write load on the primary backend thereby improving overall system performance.
4. A periodic write from the crowdsourced backend to the primary backend can be performed after datapoint verification.

3.6.3 Front End Design

The choice of creating a mobile application for the dashboard of this project is as follows:

1. Due to the penetration and wide use of smartphones, most users will more likely use a mobile application[8].
2. A web page may not have been suitable considering that the user's current location was required.

For the development of a mobile application, there were several choices. Flutter and react native were the most common tools for cross platform development. Eventually it was decided that the project will be implemented in flutter because of its performance and the author's prior experience with Flutter. Flutter also had a large developer community along with several tutorials and resources which would have aided the development process in case of any setbacks.

As for the Frontend structure of this project, the project utilises the Model View Controller (MVC) design framework. In this framework:

Model The model folder consists of the class definitions of the CCTV entities along with associated getters, setters and transformation functions.

View The view folder consists of all the presentation logic. All the pages and various views are included in the view folder.

Controller The controller folder consists of all the object controller and business logic [9].

[8] IOS and Android constitute 99.37% of the total smartphone ecosystem as per: `https://www.bankmycell.com/blog/android-vs-apple-market-share/`

[9] In this project, the controller folder was not required as the class definitions were sufficient

3.6.4 Overall System Architecture

Overall, after the various considerations and requirement fulfilment, the following system architecture was devised:

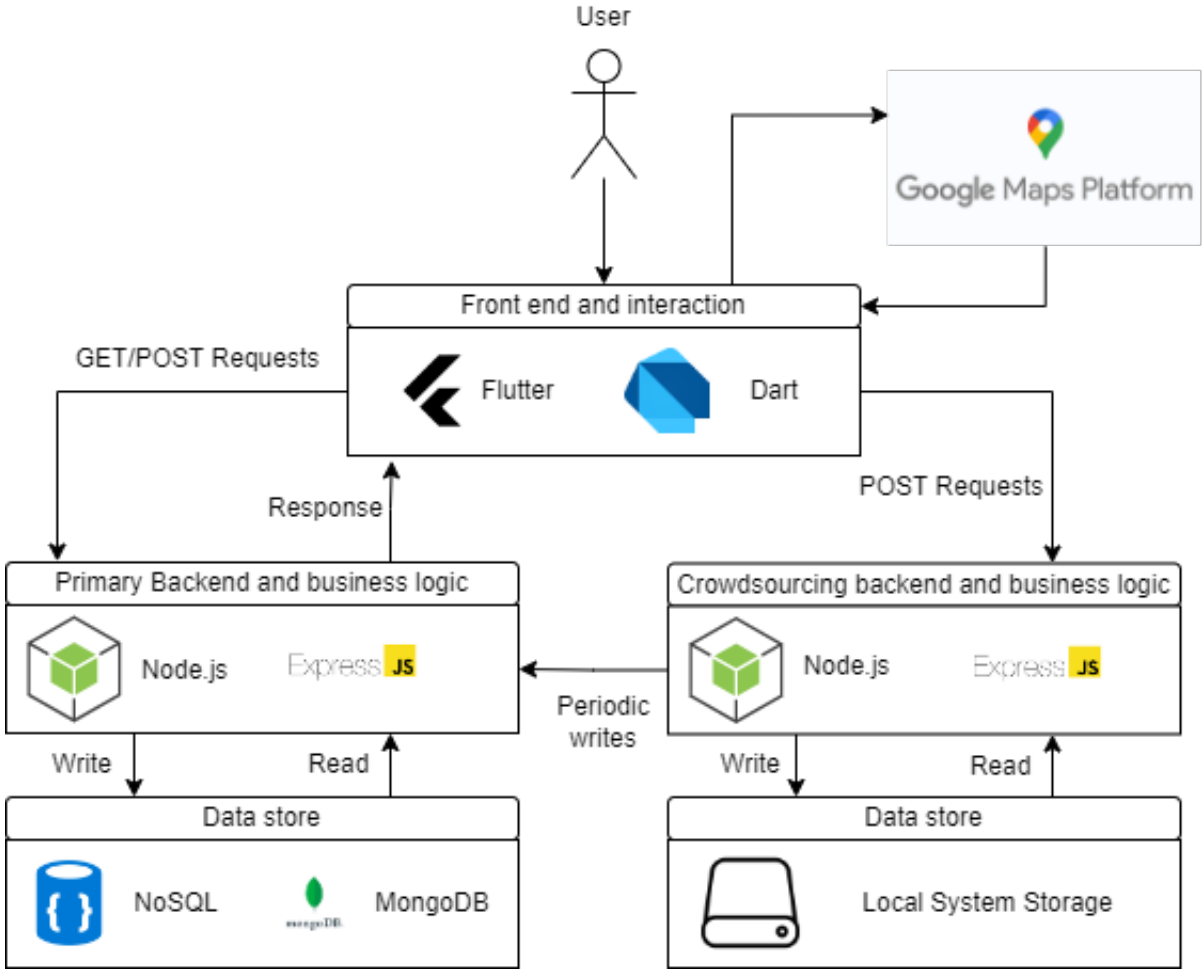
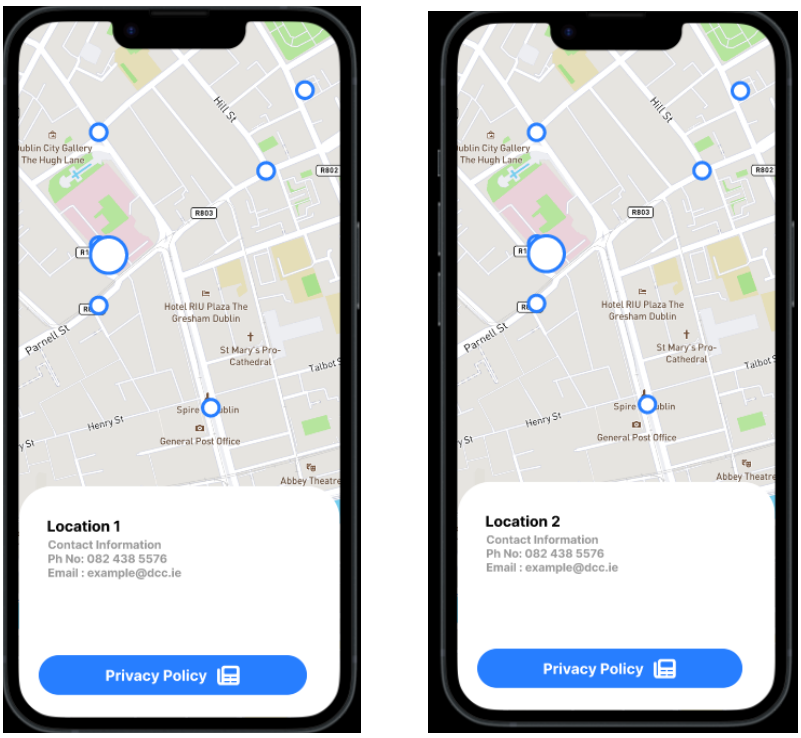


Figure 3.9: Overall System Design Architecture

3.7 Application prototyping

In designing an application, application prototyping provides a playground environment to design the UI/UX of the application and make modifications. As shown in figure 3.10, application design can be reviewed and altered quickly without actual code development.



(a) Information box with Location 1 Artefact

(b) Information Box with Location 2 Artefact

Figure 3.10: Figma Prototype Artefact

4 Implementation

4.1 Installation and dependencies

In order to aid the development process, several 3rd party libraries and dependencies were used to expedite the development process.

Backend The back end comprised of the following libraries for development[1]:

Node.js v18.13.0 A server environment which allows the execution of javascript files and test server operations on localhost.

Express.js v4.18.2 It is a back end web application framework which allows quick generation of method routes for web servers and backend development.

Nodemon v2.0.20 This library is extremely useful during development. Nodemon builds and executes the server every time there are any changes to the server files. This helps save time in rebuilding and checking server output.

Mongoose v6.9.1 The mongoose library aids the process of schema creation and Database (DB) interaction.

dotenv v16.0.3 The dotenv library allows the storage of environment variables and private variables like API keys in hidden files. These variables are only used by the developer.

object-sizeof v2.6.1 This package allows the developer to find out the size of a JSON object in bytes.

generate-api-key v1.0.2 This package is used to create the API keys for the authorised addition of data.

[1] **v** corresponds the version of package used at the time of development

Frontend There were various packages that were installed throughout the development process of the front end mobile application. The dependencies are available at pub.dev[2] and are added to the project simply by running the following command in the terminal of project directory:

```
flutter pub add <Dependency name>
```

The following dependencies were used during the course of the development of mobile application

google maps flutter v2.2.4 Flutter package which allows developers to incorporate google maps into their application

geolocator v9.0.2 This package is used to interact with the location feature and getting the current location of the user

geocoding v2.0.5 This package allows the transformation of address to coordinates and vice versa

url launcher v6.1.10 The privacy policy statements consisted of hyperlinks which were to be opened in the mobile web browser, this package allows opening of such Uniform Resource Locator (URL)s

path provider v2.0.13 The crowdsourcing aspect of this application required photos to be clicked and uploaded to the crowdsourcing database. The package was used to navigate the photo path in the mobile device

image picker v0.8.6 For uploading the photograph, the application needs access to the camera and the gallery. This package facilitated the integration for photo handling.

http v0.13.5 API requests were to be carried out by the mobile application in order to get data from the backend and also upload data to the backend. The package assists in web applications and client server communication.

test v1.22.0 Flutter provides this package to developers to assist with unit, widget and integration tests for the testing of the developed application.

[2]pub.dev Link : `https://pub.dev/`

4.2 Back End

The following sections show the development of the primary server:

4.2.1 Setting up Primary Backend:

The first step was to set up the server. A backend server was initialised with the npm packages. The following code snippet created an express application and used that application to initiate a server and listen for incoming traffic on port 3000.

```
const express = require('express');
const app = express();
app.get('/',(req,res)=>{
    res.send('We are at home');
})
app.listen(3000);
```

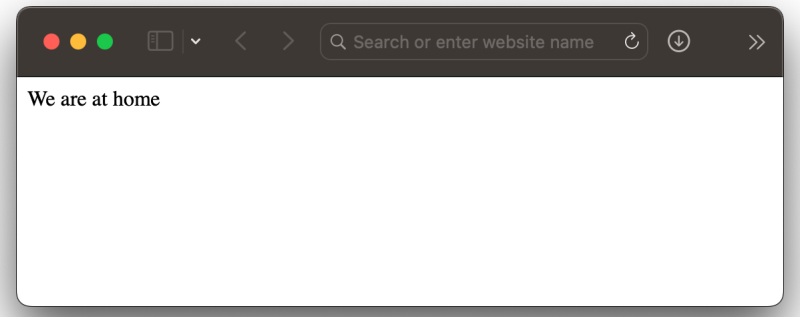


Figure 4.1: Setting up server

4.2.2 Connecting To database:

Following the creation of the backend server, the next step was to connect to the database. This was done by using mongoose. The URL for the database was copied from the mongoDB atlas dashboard supplied with the developer's credentials for access.

```
const express = require('express');
const mongoose = require('mongoose');
const {Schema} =mongoose;
const config = require('dotenv').config();
const app = express();

const PASSWORD = process.env.PASSWORD

mongoose.set('strictQuery',false);
mongoose.connect(`mongodb+srv://admin:`+PASSWORD+`@cluster0.tzrbc9a.mongodb.net/
.then(()=>app.listen(3000))
```

```
    .then(()=>console.log("Connected to Database"))
    .catch((err)=>console.log(err));
```

Console Output:

```
abhinavsinha@Abhinavs-MacBook-Air backend % npm start

> backend@1.0.0 start
> nodemon app.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node app.js`
Connected to Database
```

4.2.3 Defining Schema:

In order to perform the Create, Retrieve, Update, Destroy (CRUD) operations, a database schema was to be defined. Using mongoose, a schema declaration was added, following which a collection object was created. This collection object was used to perform the database related queries and operations.

```
    const cctvOrignialSchema = new Schema({
        "_id":Number,
        "SiteID": Number,
        "Site_Description_Cap": String,
        "Lat": Number,
        "Long": Number
    })

const cctv_orginal = mongoose.model('cctv_original',cctvOrignialSchema);
```

4.2.4 Setting up GET Route:

For a REST API, we need to set up routes for the client traffic. These routes receive a request from the client and a response is generated. In the following snippet, the server sends all available CCTV locations to the client's request.

```
    app.get('/allCCTVOriginal',(req,res)=>{
    cctv.find().then((result)=>{
        res.send(result)
    }).catch((err)=>{
        console.log(err)
    })
})
```

Client Side response output:

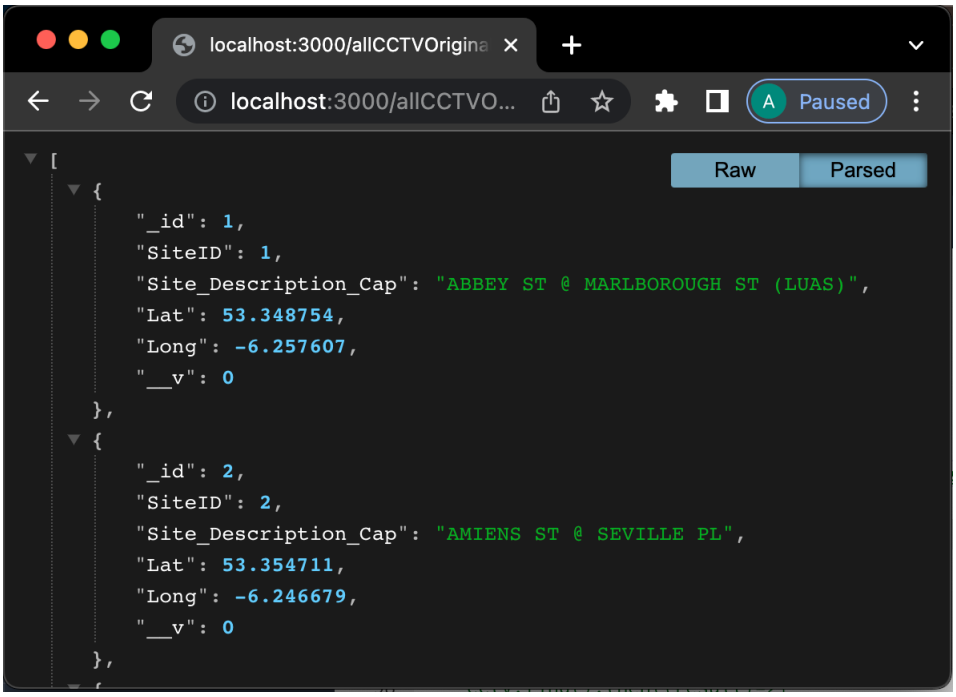


Figure 4.2: REST API Response output

4.2.5 Optimization:

A user does not require all the CCTV camera locations. For optimization of output and performance, only the CCTV devices within a certain radius of the user are sent. The CCTV location in the user's vicinity is calculated using the haversine formula[3]. The user's location is extracted from the request and accordingly the list of devices is generated.

```
function calcDistance(lat1,lat2,long1,long2){
    long1 =  long1 * Math.PI / 180;
    long2 = long2 * Math.PI / 180;
    lat1 = lat1 * Math.PI / 180;
    lat2 = lat2 * Math.PI / 180;

        // Haversine formula
        let dlon = long2 - long1;
        let dlat = lat2 - lat1;
        let a = Math.pow(Math.sin(dlat / 2), 2)
                + Math.cos(lat1) * Math.cos(lat2)
                * Math.pow(Math.sin(dlon / 2),2);

        let c = 2 * Math.asin(Math.sqrt(a));

        // Radius of earth in kilometers. Use 3956
        // for miles
        let r = 6371;
```

[3]A formula which can be used to calculate the distance from latitude and longitude

```
        // calculate the result
        return(c * r);
}

    app.get('/getnearbyOriginal',(req,res)=>{
    finalResult=[];
    // currLat = req.query.lat;
    // currlong = req.query.long;
    cctv_orginal.find().then((result)=>{
        for(let index in result){
            lat1 = result[index]['Lat'];
            long1 = result[index]['Long'];
            distance = calcDistance(lat1,currLat,long1,currLong);
            if(distance<=1){
                finalResult.push(result[index]);
            }
        }
        res.send(finalResult);
    }).catch((err)=>{
        console.log(err);
    });
})
```

Performance Improvement:

Without Optimization:

```
Elapsed Time: 0 millisecond
100307 Bytes
```

With Optimization:

```
Elapsed Time: 1 millisecond
6135 Bytes
```

Hence, this optimization done by using the haversine formula reduces the response size. This comes at the cost of computing nearby CCTV camera locations. Since this is being done on a list and the list is being traversed once, overall the computation has O(n) time complexity[4] and O(n) space complexity.

[4] Linear Complexity

4.2.6 API Key generation

As discussed in section 3.5.4, the server would require the generation of an API key for the CCTV installers/controller to upload the data. The following code snippet shows the generation and response of a API key generate request:

```
const { generateApiKey } = require('generate-api-key');
app.get('/requestAPIKey',(req,res)=>{
const key = generateApiKey();
res.send(key);
})
```

Browser Output

```
API Key: yn9W~SA8b7dGT4iwJ8XCrb5QAR
```

The generated API key can now be used by the CCTV installers[5].

4.3 Crowdsourcing Backend:

The crowdsourcing backend was generated in a manner similar to the primary backend. For the crowdsourcing backend, a POST route was created to accept the images from the client. There is no limit as to the size of the image being uploaded, however, this can be changed by adding a parameter to the fileUpload object. This POST route saved the incoming images in the local development machine's storage. The following code snippet shows the POST route definition. Figure 4.7 shows the execution and saving of photos captured by mobile devices on the server.

```
const express = require("express");
const fileUpload = require("express-fileupload");
const app = express();
app.use(fileUpload());
app.use(express.static('public'));
app.post('/upload',(req,res)=>{
    const {image} = req.files;
    if(!image) return res.sendStatus(400);
    image.mv(__dirname+'/upload/'+image.name);
    res.sendStatus(200);
});
app.get('/',(req,res)=>{
    res.send("We are at home");
})
app.listen(3000);
```

[5]Complete implementation of this aspect could not be developed in time

4.4 Front End

The flutter application consists of a stack structure where each of the components have been layered on top of each other. Only high level code is illustrated below, the configuration and other source file are open source and available in section A1.1:

4.4.1 Integrating Google Maps

```
import 'package:google_maps_flutter/google_maps_flutter.dart';
...
GoogleMap(
            initialCameraPosition: CameraPosition(
              target: LatLng(53.350357, -6.266422),
              zoom: 18,
            ),
            onMapCreated: (controller) {
              mapController = controller;
              addMarker();
            },
            myLocationEnabled: true,
            myLocationButtonEnabled: true,
            markers: _markers.values.toSet(),
          ),
          ...
```

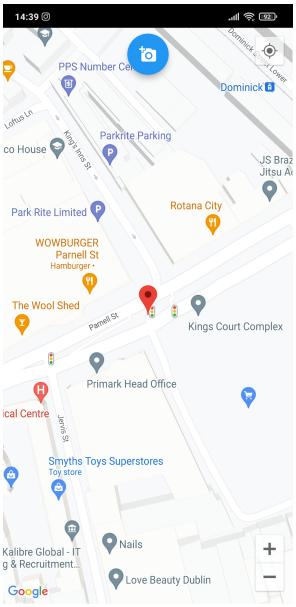
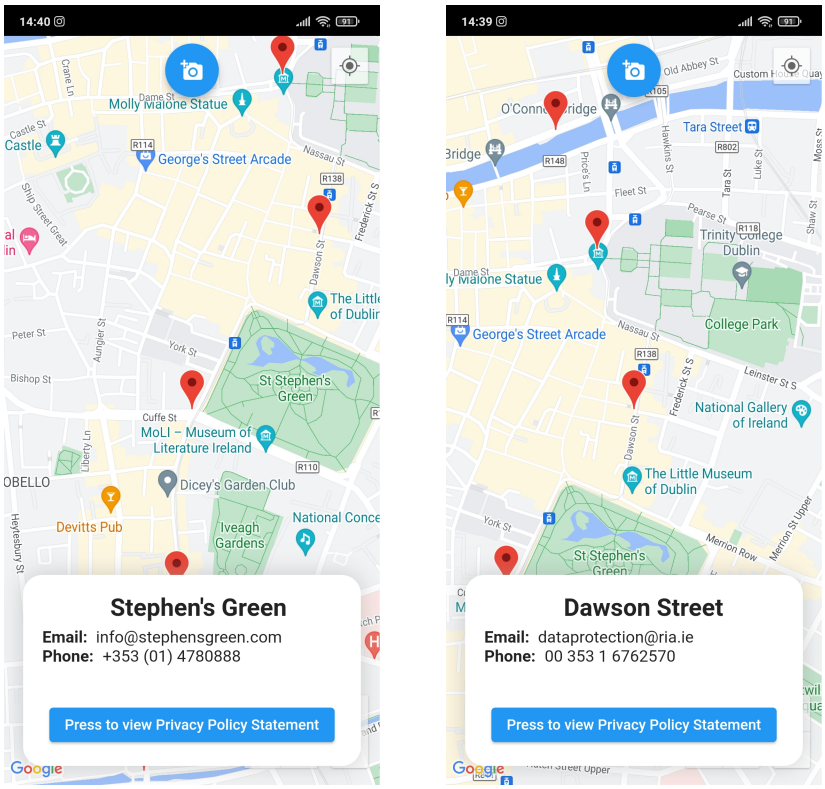


Figure 4.3: Google Map Integration Artefact Screenshot

4.4.2 Information Window

The information window allows the user to view additional associated information of the CCTV device. It shows the contact information for further queries and a redirect button which when pressed, opens the privacy notice for the user's perusal. The code for this implementation is quite extensive[6].



(a) Location 1 (b) Location 2

Figure 4.4: Information Window Artefact Screenshot

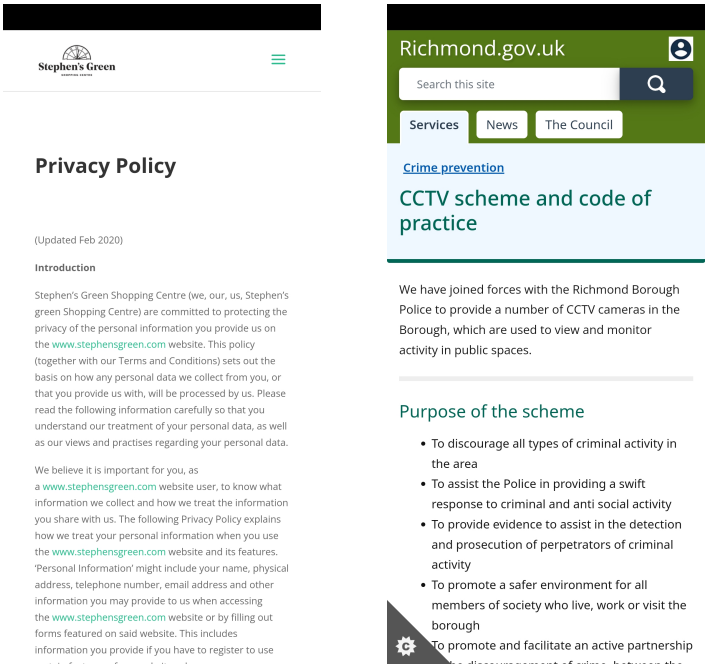
4.4.3 Privacy Notice Redirect

The privacy notice redirect button allows the user to view the privacy notice, should they wish to explore it. The following code shows the definition of the URL launcher function and the button which resides in the information box.

```
import 'package:url_launcher/url_launcher_string.dart';
...
mylaunchURL(String s) async {
    if (await canLaunchUrlString(s)) {
        await launchUrlString(s);
    } else {
        throw 'Could not launch $s';
    }
}
...
```

[6]The complete source code can be found in section A1.1

```
ElevatedButton(
    onPressed: () {
        mylaunchURL(_currentCCTVData.policyUrl!);
        },
        child: Text(
            "Press to view Privacy Policy Statement",
                style: TextStyle(fontSize: 18),
                ),
            ),
...
```



(a) Location 1 (b) Location 2

Figure 4.5: Privacy Notice Redirect Artefact Screenshot

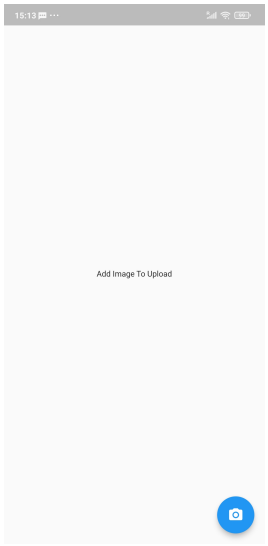
4.4.4 Uploading Photograph for crowdsourcing

In order to support the upload of crowdsourced images, the mobile application had to capture the images, and upload it on the defined POST route. The following code snippet[7] shows the getting of image from device and upload function to support the transfer of image to the server.

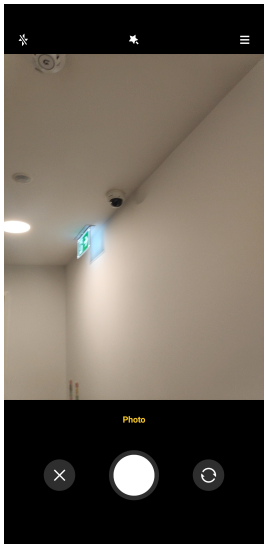
```
...
Future getImage() async {
        final image = await imagePicker.
        pickImage(source: ImageSource.camera);
        setState(() {
        _image = File(image!.path);
        });
    }
```

[7]... shows the presence of other code. See section A1.1 for complete source code

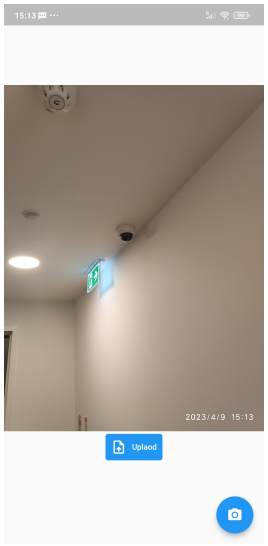
```
...
...
    uploadImage() async {
        final request = http.
        MultipartRequest("POST", Uri.parse(URL + "/upload"));
        final headers = {"Content-type": "multipart/form-data"};
        request.files.add(http.MultipartFile('image',
        _image.readAsBytes().asStream(),
        _image.lengthSync(),
            filename: _image.path.split("/").last));
        request.headers.addAll(headers);
        final response = await request.send();
        http.Response res = await http.Response.fromStream(response);
        final resJson = jsonDecode(res.body);
        message = resJson['message'];
        print(message);
        setState(() {});
    }
...
```



(a) Add Image Screen



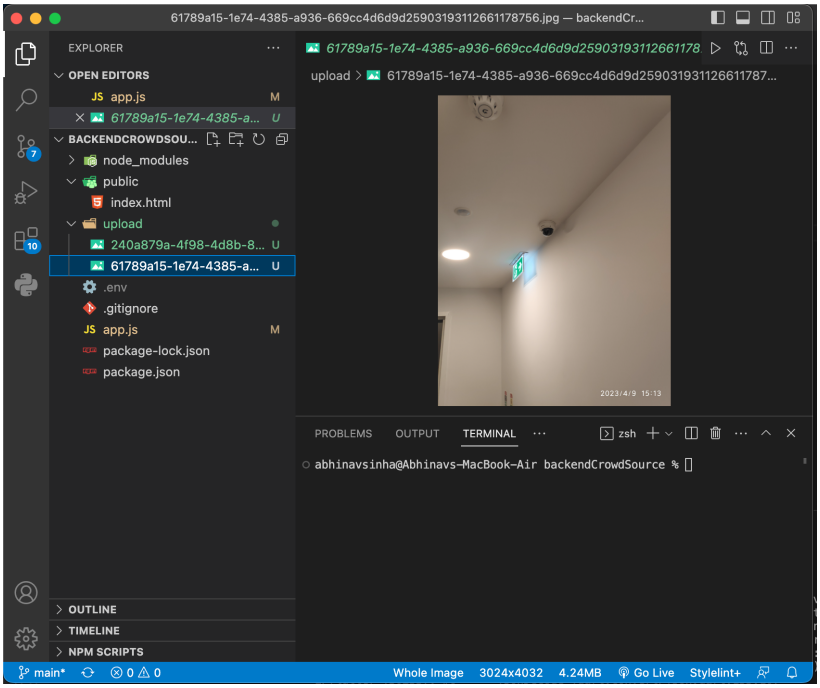
(b) Click Image using Camera



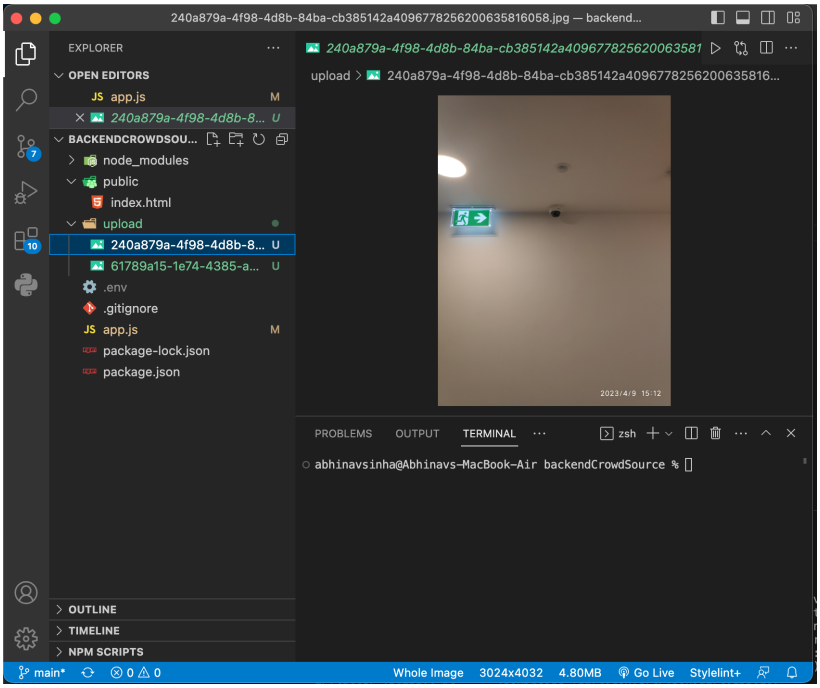
(c) Upload Image

Figure 4.6: Photo Upload Artefact Screenshot

At Server:



(a) Image 1



(b) Image

Figure 4.7: Server side Image Upload Artefact Screenshot

5 Evaluation

5.1 Testing

5.1.1 Primary Backend - Manual Testing

Postman is a developer tool which is often used to test backend REST APIs. This application mainly used 2 REST Methods:

GET Methods:

The GET methods were tested by making a request to the URL and getting a status 200 response from the server. Since the list of routes wasn't extensive, manual testing was the choice of testing strategy

POST Methods:

In a similar fashion, postman was used to test the POST method routes in the server. The expected response was status 200 and the database was manually checked to see whether the correct data has been uploaded.

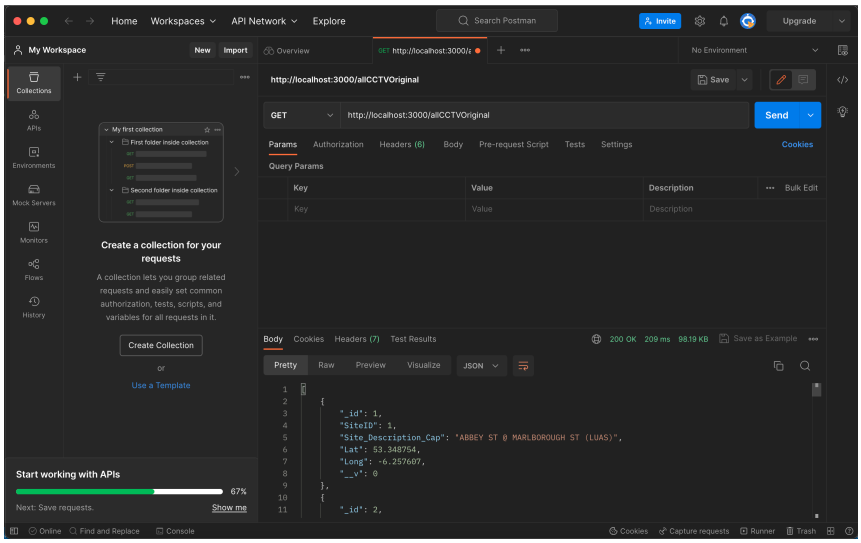


Figure 5.1: Postman test Artefact Screenshot

Load Testing

In order to test the response serving capability of the primary backend, the data_original.json dataset was uploaded and requested from the server. The json consisted of over 800 CCTV locations and responded in less than a millisecond (See section 4.2.5). For a higher order load test, a larger dataset file would have to be used.

5.1.2 Crowdsourcing Backend- Manual Testing

The testing process for the crowdsourcing backend was performed with the help of a tool called ngrok. ngrok is a Command Line Interface (CLI) tool which allows the developer to host the localhost server and maps it to a testing domain. This testing domain was embedded as a URL in the frontend of the application and the server was manually looked up to check if the appropriate POST method route is working. See section 4.7 for the upload artefact on the server.

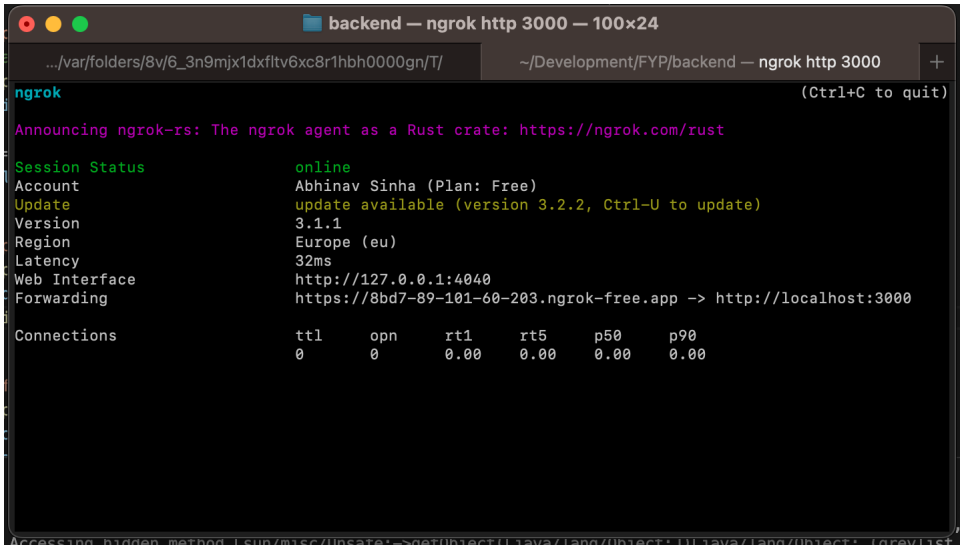


Figure 5.2: ngrok providing localhost proxy

5.1.3 Frontend Testing

The application was deployed on a physical device, relevant device information is as follows:

```
Platform: Android
Device: Xiaomi Poco F1
Android Version: 10
MIUI Version: 12.0.3
```

In order to test the flutter application, the test dependency had to be added. This was done using the following command in the terminal:

```
flutter pub add test --dev
```

After the addition of the dependency, the functionality and behaviour of the application was tested. Flutter supports 3 kinds of testing:

1. **Unit Testing** is performed on individual functions, method or class.
2. **Widget Testing** is performed on individual widgets in the present in the flutter application.
3. **Integration Testing** is used to test the entire application.

Unit Testing

In our flutter application, most of the functions return an unexpected Future Type which is dependent on user activity. These functions do not necessarily have a return value. Hence, even if a test were to be created, because of the absence of an expected value, the test would not render anything useful. Therefore, the functions were manually tested as a part of the application.

Widget Testing

For widget testing, the testing was done on the widgets that are appearing on the screen. If the widgets were being constructed and displayed on the screen, then this test would be successful. The following code showcases the testing done on the outermost Scaffold widget:

```
    import 'package:flutter/material.dart';
    import 'package:flutter_test/flutter_test.dart';
    import 'package:frontend/pages/HomePage.dart';
    import 'package:google_maps_flutter/google_maps_flutter.dart';
    void main() {
        testWidgets('Application generation test', (WidgetTester tester)
        async {
        // Build our app and trigger a frame.
        var scaffold = find.byType(Scaffold);
        expect(scaffold, findsOneWidget);
    });
}
```

Console Output

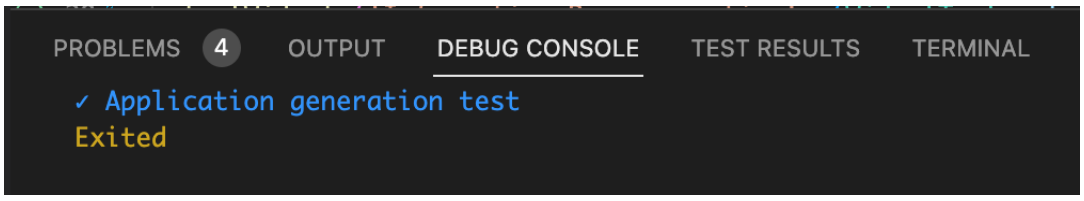


Figure 5.3: Console output artefact screenshot

Integration Testing

As the application consisted of an embedded google maps widget, the integration testing aspect was challenging. Despite extensive research and looking into testing packages like mockito, at the time of project implementation, there was not a reliable way to create integration tests for a google map embedded application. Hence, the application was deployed on the device and manually tested. See section 4.3 for application artefact.

5.2 Requirements Evaluation

Functional Requirements	**Assessment**
CCTV Location Pins	Success: The CCTV location are marked on the embedded Google Map within the application
Information Box	Success: The information box window shows the requisite information and presents it to the user
Redirect Privacy Notice	Success: The user has been provided with a button that redirects the user to the privacy policy of the CCTV controller.
Crowd Sourcing	Success: The crowdsourcing aspect allows the user to upload photographs of the CCTV Device.
Non-Functional Requirements	**Assessment**
Consistent and Reliable	The database has 2 redundancy clusters which store the data for reliability. 825 data points were retrieved a number of times providing consistent results. However, consistency and reliability of the server can only be ascertained by performing a battery of tests such as load (See section 5.1.1), performance, redundancy and security tests.
Easy to use	The locations of CCTV devices are easier to grasp when viewed on a map. Hence, the application was designed around a google map. This makes it easier for the visualisation of the exact location of the CCTV device. The information box shows the requisite information. Various considerations and highlighting techniques were made to ensure clear communication of information. Due to the project timeline, the UI did not undergo user testing. In order to ascertain if the UI/UX is easy to use, the application will have to undergo usability testing and gain user feedback.
Performance	Success: An optimization in the server backend allowed the resultant response size to be reduced greatly (refer 4.2.5).
Open Source	Success: The project source code is publicly available on GitHub for the perusal of other developers (See section A1.1).

Table 5.1: Assessment of functional and non-functional requirements

5.3 Evaluation of Objectives

5.3.1 Research Objectives

1. A great deal of understanding regarding the kind of information collected, stored and processed by CCTV controllers.
2. Explored various real life examples of CCTV cameras, their controllers and the legislation around surveillance. Resources like SMART Dublin were extensively inspected and a publicly available registry was built.
3. Carried out the design, implementation and testing of a dashboard using a cross platform mobile development framework.
4. Extensive research and solution development of a crowdsourcing solution.

5.4 Limitations

Despite the successful implementation of the project, there are areas of improvement in the implementation of this project:

Crowdsourcing A theoretical solution and plan has been explored in this project, however, due to the timeline of implementation, this aspect could not be implemented in due time.

Deployment Currently, the project is executed only in a local development environment. In order to launch it for public use, the application will need to be further tested and deployed on a cloud service.

Insufficient Data points At the time of development, a dataset from SMART Dublin was used to create the public registry. This needs to be extended to include more datasets from various sources. Crowdsourcing will also help in the creation of data points in the public registry.

Summary of data collection The application does not summarise the kind of access/data collection that is being performed by the CCTV data controller due to the dissimilar structure of various privacy notices.

Mobile CCTV The application does not cater to CCTV devices which are present in vehicles. This applies to public transport vehicles such as buses where public surveillance is carried out.

Underlying Algorithm The kind of FRT algorithm being used by the CCTV device is not shown. Certain algorithms have bias and the performance varies for different FRT algorithms (See section 6.2)

6 Discussion

6.1 Ethical Considerations

The application has been designed to aid the detection and view location of surveillance devices. A user may choose to follow a path which does not consist of a lot of surveillance devices in order to protect themselves and their privacy.

On the contrary, a user may use this kind of application for escaping surveillance after committing a misdemeanour/crime.

Under DPC guidelines, certain exceptions can be made. In cases of crime and policing, the authorities can run facial recognition in surveillance footage in order to aid the investigation.

The application has been developed as a tool for protecting an individual's privacy. However, these considerations need to be made before public deployment for an application with such implications.

6.2 Gender and racial bias in facial recognition

ML and FRT are not perfect. Oftentimes, there is a disparity in match rate with respect to the particular characteristics of the person. This may vary based on the individual's gender and race [22].

Since these recognition technologies are the basis of CCTV surveillance, it is important to note that there is a certain bias towards an individual's recognition.

With this project, the aim is to increase user transparency. The privacy notice must include the underlying algorithms being used along with the error rates and bias that is associated with them. A user must be privy to this kind of information as it would affect the consent choices made by an individual of a particular gender/race.

6.3 Avenues for Future Work

As this project is in its infancy, there are several avenues where future work in this space can be carried out.

User Alert The application could include a feature which alerts the user via a notification when they are under surveillance.

CCTV Device Specification Exploration of the specific device specification used in the field of surveillance. The kind of recognition strategy, the input required and various other aspects of the physical CCTV device itself.

Domestic Surveillance With increasing use of devices like the amazon ring, there is widespread domestic surveillance being carried out. Domestic surveillance is not subject to the same level of scrutiny as public surveillance, however, it is highly dependent on the range of view and visibility of the camera.

Range of View A camera has a limited range of view. This aspect needs to be accounted for in the detection of the surveillance. The application can include the direction and area of surveillance and present a more detailed view to the intended user.

Automated consent An automated consent withdrawal feature which sends an email to the respective authority about withdrawing consent from further processing, based on the user's preferences.

6.4 Open source

In the space of privacy and security, increased transparency is paramount. This project has been made open source for development and contributions. This project is available on github [1]. The community can view the source code, send in developer feedback or make pull requests for contribution.

6.5 Application Data Collection

The developed application is an information service application and does not require any kind of registration by the user. The only data which is collected by the application is the current location data of the user. The location data is used to view the CCTV devices in the vicinity of the user. The location data is sent only after the user's explicit consent.

In case of crowdsourcing of photographs, there is a chance that personal data may be recorded. If the user is uploading a photograph and in an happenstance another individual's facial data may be recorded. The crowdsourced photographs do not undergo any processing, however this is a point of concern in case of future extension of the project.

[1] See Section A1.1 for source code

7 Conclusion

Increasing surveillance and improved data processing, this project report explores the various aspects of an individual's fundamental right to privacy. The reader is presented with the motivation behind the project, including several instances of misconduct where extensive surveillance and improper FRT led to unfortunate incidents. These incidents reinforced the need to improve user transparency as suggested by the GDPR. The report delves into the state of the art, reviews the approaches of other projects and examines the shortcomings which need to be addressed.

With the current scenario in mind, the report follows up on developing a solution which is more user-friendly and addresses the points of concern. Creating a list of functional requirements, the overall application flow and use cases are presented. Subsequently, the application design is developed along with several considerations. The project considerations constituted an important facet. The design choices and techniques in order to implement crowdsourcing imparted great knowledge.

With these design choices and the architecture set, the project was developed with constant back and forth between writing new code and debugging. With the application in place, testing and assessment of functionality using postman, ngrok and flutter widget testing inspired confidence in the application. The extensive nature of this project has led to some interesting avenues for future work which could build upon this research.

As a sidebar, the discussions section brings some other aspects of the underlying technologies which are equally important. The developed application is a tool, its use depends on the individual. Ethically, it should be used to safeguard one's right to privacy.

Reflecting on the work done in this project, it has been a success not only objectively, but also from a learning point of view. The copious amount of research and in-depth technical know-how developed during the course of this project has been astounding. In conclusion, while we are progressing towards a world with no boundaries, where extensive surveillance is violating the right to privacy, this project works towards giving users the control over their personal data and helping them make informed choices.

Bibliography

[1] R. Buckley, "Data privacy and community cctv schemes," data.oireachtas.ie, 01 2019. [Online]. Available: https://data.oireachtas.ie/ie/oireachtas/libraryResearch/2019/2019-01-14_data-privacy-and-community-cctv-schemes_en.pdf

[2] AIAAIC, "Aiaaic - about aiaaic," www.aiaaic.org. [Online]. Available: https://www.aiaaic.org/about-aiaaic

[3] AIAAIC , "Aiaaic - amazon ring video doorbell neighbour privacy invasion," www.aiaaic.org, 10 2021. [Online]. Available: https://www.aiaaic.org/aiaaic-repository/ai-and-algorithmic-incidents-and-controversies/amazon-ring-video-doorbell-neighbour-privacy-invasion

[4] AIAAIC, "Aiaaic - mohammed khadeer facial recognition wrongful arrest, death," www.aiaaic.org, 02 2023. [Online]. Available: https://www.aiaaic.org/aiaaic-repository/ai-and-algorithmic-incidents-and-controversies/mohammed-khadeer-facial-recognition-wrongful-arrest-death

[5] V. X. Xu and B. Xiao, "Chinese authorities use facial recognition, public shaming to crack down on crime," ABC News, 03 2018. [Online]. Available: https://www.abc.net.au/news/2018-03-20/china-deploys-ai-cameras-to-tackle-jaywalkers-in-shenzhen/9567430

[6] An Garda Síochána, "Automatic number plate recognition," Garda, 2023. [Online]. Available: https://www.garda.ie/en/roads-policing/road-safety/automatic-number-plate-recognition.html

[7] S. Domezet, M. Lubura, I. SusakLozanovska, and N. Ilik, "Chinese social credit system: New challenges for the right to privacy?" *Journal of Liberty and International Affairs (JLIA)*, vol. 7, pp. 136–148, 01 2021. [Online]. Available: https://elib.tcd.ie/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edshol&AN=edshol.hein.journals.jlia7.61

[8] M. A. Rothstein and S. Carnahan, "Legal and policy issues in expanding the scope of law enforcement dna data banks," *Brooklyn Law Review*, vol. 67, p. 127, 2001. [Online]. Available: https://heinonline.org/HOL/LandingPage?handle=hein.journals/brklr67&div=14&id=&page=

[9] Data Protection Commission, "Data protection commission - cctv, discovery and access requests," Data Protection Commission, 02 2021. [Online]. Available: https://www.dataprotection.ie/en/dpc-guidance/blogs/cctv-discovery-and-access-requests

[10] Access Now, "Ban biometric surveillance," Access Now, 06 2021. [Online]. Available: https://www.accessnow.org/campaign/ban-biometric-surveillance/

[11] J. Kalathas, "Open letter to the irish times: Experts' red line on policing facial recognition technologies," UCD Centre for Digital Policy, 06 2022. [Online]. Available: https://digitalpolicy.ie/ireland-experts-red-line-on-garda-facial-recognition-tech/

[12] Data Protection Commission, "Definition of key terms," Data Protection Commission, 01 2021. [Online]. Available: https://www.dataprotection.ie/en/individuals/data-protection-basics/definition-key-terms

[13] B. Mueller, "The artificial intelligence act: A quick explainer," Center for Data Innovation, 05 2021. [Online]. Available: https://datainnovation.org/2021/05/the-artificial-intelligence-act-a-quick-explainer/

[14] M. Nouwens, I. Liccardi, M. Veale, D. R. Karger, and L. Kagal, "Dark patterns after the gdpr: Scraping consent pop-ups and demonstrating their influence," *CoRR*, vol. abs/2001.02479, 2020. [Online]. Available: http://arxiv.org/abs/2001.02479

[15] H. J. Pandit, A. Polleres, B. Bos, R. Brennan, B. Bruegger, F. Ekaputra, J. Fernandez, R. Hamed, E. Kiesling, M. Lizar, E. Schlehahn, S. Steyskal, and R. Wenning, "Creating a vocabulary for data privacy," doi.org, 10 2019. [Online]. Available: https://harshp.com/research/publications/032-creating-vocabulary-data-privacy

[16] R. Taneja, "Consent right - a browser extension to provide granular element removal in cookie consent banner," 04 2021.

[17] C. Cawley, "5 ways to find hidden cameras using your mobile phone," MUO, 12 2018. [Online]. Available: https://www.makeuseof.com/tag/use-smartphone-detect-hidden-surveillance-cameras/

[18] A. Pravin, J. T. Prem, P. K. Mohana, T. Judgi, and N. Srinivasan, "Efficient framework for hidden camera detection and jamming using iot," pp. 634–637, 2022.

[19] C. Keogh, "Design thinking," tcd.ie, 2021. [Online]. Available: https://www.tcd.ie/trinity-electives/electives/design-thinking/

[20] C. Chai, J. Fan, G. Li, J. Wang, and Y. Zheng, "Crowdsourcing database systems: Overview and challenges," dbgroup.cs.tsinghua.edu.c. [Online]. Available: https://dbgroup.cs.tsinghua.edu.cn/ligl/papers/icde19-tutorial.pdf

[21] D. T. B. Kurniawan, "The clustering algorithm with geolocation data," medium.com, 04 2021. [Online]. Available: https://medium.com/thelorry-product-tech-data/the-clustering-algorithm-with-geolocation-data-d6dd07ed36a

[22] J. Buolamwini, T. Gebru, S. Friedler, and C. Wilson, "Gender shades: Intersectional accuracy disparities in commercial gender classification *," *Proceedings of Machine Learning Research*, vol. 81, pp. 1–15, 2018. [Online]. Available: https://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf

A1 Appendix

A1.1 Github Repository

The source code for this project is available at:`https://github.com/abhisinha2001/FYP`

A1.2 Figma Prototype:

The Figma prototype developed is available at : `https://www.figma.com/file/gTTTIgAi2TEskuYzK9GiCK/CCTV-SafeView?node-id=0%3A1&t=5PZ7jEliHxBl0x0p-1`

A1.3 Security and Privacy Considerations:

With respect to the development and implementation of the application the following considerations were made:

A1.3.1 Database:

1. The database is hosted on MongoDB atlas, a cloud solution running MongoDB. The console follows the principle of least privilege. This is implemented by the access
2. Additionally, the database has been configured to only allow access from incoming traffic from the primary development laptop's IP Address. This further increases any kind of risk/ unauthorised access to the database. being granted to a single user, in this case the developer of the application.
3. Furthermore, the data stored in the database is only released by API endpoints defined in the backend
4. There is a well defined schema for the data being stored, so as to ensure data integrity and validation.

A1.3.2 Backend:

As for the backend, the following measures were taken:

1. The server consists of various credentials that need to be stored in order to connect to the database.The dotenv package has been used. The dotenv package allows the developer to

store the variables in a hidden file that are to be used in the application. This provides us with a layer of security as the credentials are neither stored nor used in plain text.

2. As the project is open source, the dotenv configuration file is added to .gitignore, so as to prevent the upload of the configuration file.
3. For the integrity of the data being stored, an API key is generated and provided to authorised data controllers who wish to add a location of a CCTV device to the database
4. As with any server, this backend is susceptible to DDOS attacks. If it were to be deployed for public use, mitigation strategies such as load balancers and firewalls would have to be incorporated.
5. Exception management has been implemented to handle any unforeseen or incompatible requests.
6. The server only accepts traffic on port 3000 for local development and testing.

A1.3.3 FrontEnd:

The front end of the application uses a variety of services and external packages. For some of these services, an API key is required. For instance, the google maps API requires the developers API key for being used.

1. Similar to the backend, the API key is stored in a hidden configuration file and is never used directly.
2. As the application is intended for public use and data viewing, user credentials are not created. Similar to the TFI leap card application, there are no login credentials required to use the app.
3. The http dart package has been used for client server communication. This package uses https by default for any kind of client server communication.

For this particular project, having login credentials would have hampered application adoption. However, if the application did require login credentials, external OAuth[1] services. would have been employed.

A1.3.4 Credentials:

As for the creation of passwords itself, care was taken to not use a pre-existing password. Furthermore, an appropriate password length was used. The password comprised of a random combination and included special characters so as to protect it from any brute force attack

[1] External 3rd party authentication service

A1.4 Documentations

A1.4.1 Flutter Documentation

The flutter documentation is a available at : https://docs.flutter.dev/

A1.4.2 Mongoose Documentation

The mongoose documentation is available at: https://mongoosejs.com/docs/

A1.5 Articles on clustering

1. K-means : https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
2. Clustering: https://ieeexplore.ieee.org/document/9862035/figures#figures
3. Geo- spatial Clustering: https://medium.com/thelorry-product-tech-data/the-clustering-algorithm-with-geolocation-data-d6dd07ed36a

A1.6 App Development Resources

1. Figma Design: https://dribbble.com/shots/18909504-Map-Navigation-App-UI-Kit
2. Google Map Tutorial: https://www.youtube.com/watch?v=EYcslTjRqCY&ab_channel=CodeX
3. Getting current Location: https://medium.com/@fernnandoptr/how-to-get-users-current-location-address-in-flutter-geolocator-geocoding-be563ad6f66a#~:text=Get%20user's%20latitude%20and%20longitude,-Now%20it's%20time&text=To%20query%20the%20current%20location,%3BPosition%20position%20%3D%20await%20Geolocator
4. Inserting into MongoDB: https://www.geeksforgeeks.org/mongoose-insertmany-function/
5. Flutter Packages and their use: https://pub.dev/