

# FINAL REPORT

---

## eirguard

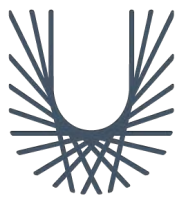
Cybersecurity governance and compliance

---

Peter O'Hare

Project Supervisor

Dr. Christopher Staff



**SE  
TU**

Ollscoil  
Teicneolaíochta  
an Oirdheiscirt

South East  
Technological  
University



# Table of Contents

<b><i>Acknowledgements</i></b> .....	<b>2</b>
<b><i>Abstract</i></b> .....	<b>3</b>
<b><i>1. Introduction</i></b> .....	<b>4</b>
<b><i>2. The Process</i></b> .....	<b>5</b>
2.1 Project Scope .....	5
2.2 Planning.....	6
2.3 Desired Functionality .....	7
<b><i>3. Technical Development</i></b> .....	<b>8</b>
<b><i>4. How Eirguard Works</i></b> .....	<b>10</b>
4.1 A brief walkthrough .....	10
4.2 Testing .....	14
<b><i>5. Development Process and Technical Issues</i></b> .....	<b>16</b>
5.1 The Application .....	16
5.2 The Back End.....	18
5.3 Scope Creep .....	20
<b><i>6. Personal Reflection</i></b> .....	<b>21</b>
6.1 Challenges.....	21
6.2 Lessons Learned .....	23
6.3 Skill Development .....	24
<b><i>7. Conclusion</i></b> .....	<b>25</b>
<b><i>8. References</i></b> .....	<b>26</b>

# Acknowledgements

Sincere appreciation and thanks are due to my project supervisor Dr. Christopher Staff, for his tireless and relentless belief that I could actually produce something! His unwavering belief in me, which at times was much stronger than my own, was instrumental in getting me over the line. His boundless empathy and compassion, throughout what turned out to be a very difficult year, cannot be understated. It was a pleasure to work with Dr. Staff, and I am truly grateful for his guidance, advice, support and understanding throughout the entire process.

Thanks Chris! Couldn't have done it without you!

# Abstract

This report details the design and development Eirguard, an iPad iOS app whose aim is to assist SMEs in the development of solid cybersecurity governance principles. Research indicated the vast majority of SMEs are aware of the importance of cybersecurity in their organisations, most having implemented certain safeguards. However, the importance of having solid cybersecurity governance principles in place was often overlooked. In the era of the new EU NIS2 [1] directive and the requirements it places on organisations, it behoves organisations to adopt strong policies and procedures in cybersecurity. Eirguard attempts to make the governance process as easy and streamlined as possible.

# 1. Introduction

Threats against cybersecurity are ever-present and show no sign of abating. Threat actors continuously seek new ways to infiltrate and compromise systems for either financial gain, intelligence gathering, or in this era of cyber warfare, inflicting as much damage on an adversary as possible.

To help combat this and ensure that member states have the strongest possible defences in place, the EU implemented the NIS2 Directive[1] which set in law definitive cybersecurity requirements for organisations operating in defined industries throughout the bloc. It was hoped that by ensuring a high common level of cybersecurity across the European Union, the level of cybersecurity attacks would be more effectively controlled and managed.

Eirguard's aim is to help with the creation and implementation of cybersecurity governance policies and procedures. It's important to understand that Eirguard does not assist with the *implementation* of the policies, just the creation and maintenance of same.

While there are a plethora of third party vendors that provide audit and implementation services for NIS2, these vendors are typically looking to upsell a continuing subscription to their monitoring service once the implementation phase has been completed. Stand-alone applications like Eirguard are notable by their absence.

In this document I'll try to give as clear a picture as I can, of my approach to the design and implementation of Eirguard. Throughout each section, I've tried to include my thought process at the time, and I also do my best to highlight my failures, as well as my successes.

In section 2 I outline the design process, planning and end goals for Eirguard. In section 3, I discuss the technical aspects and approaches that I took. Section 4 is a walkthrough on how Eirguard works, and showcases some of the features it possesses. Section 5 goes into the development process itself, and some of the issues faced during that period. Finally, in section 6, I discuss what I learned during the entire process, both personally – and of course the technical stuff!

## 2. The Process

### 2.1 Project Scope

Initially Eirguard was envisioned to be an iPad app with a complementary website that would act as an administration portal. This portal would give the “organiser”, administrator or project lead, full administration privileges. This approach was considered to allow the admin to create a timeline, goals, and user accounts for contributors. The portal dashboard (Fig. 1) would also allow the admin to view progress, create reports and share documents with management.

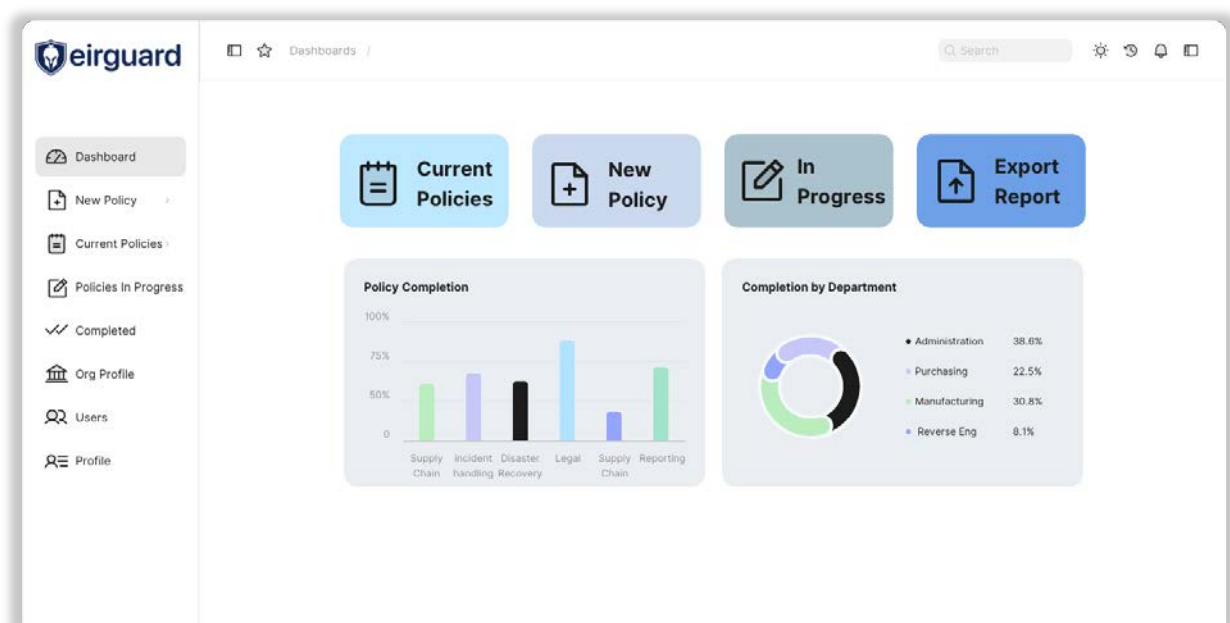


Figure 1: Admin portal design mockup

Research into technologies for the web portal led to the decision to use a Django backend for its stability and robust security, as well as being able to leverage its REST framework. The combination of the web portal and the iPad app working in tandem provided a whole stand-alone system that could be expanded and built upon in future revisions.

While the original plan offered end-users greater flexibility of use, especially in larger SMEs, in terms of moving the project forward, in discussion with my supervisor, it was decided to focus entirely on the iPad app. This decision was made for several reasons, time being the major factor. We felt it was better to concentrate on the core idea and deliver as complete a

product as possible, rather than spreading time developing over two different areas and technologies, and possibly diluting the final outcome of both.

The web based admin portal is a component that could certainly be added at a later stage should the project be developed further.

## 2.2 Planning

There were two definite visions for Eirguard from the very beginning. The first being to bring an effective governance structure to small and medium enterprises. In that context, as mentioned in the introduction, Eirguard does not assist in the actual implementation of the policies, rather only in the development of the policies themselves.

To facilitate the development of policies and procedures, Eirguard needed an established framework on which to base that development. Several frameworks were looked at including COBIT 2019[2], ISO27001[3] and NIST's Cyber Security Framework (CSF)[4]. Ultimately CSF was chosen for a number of reasons. It's a well-established and trusted framework, the financial cost to entry was minimal, but the most important aspect was the development of CSF 2.0 [5] which would update the framework to incorporate changing needs and trends.

The biggest change to CSF 2.0 was the introduction of a section on governance, and governance best-practice. This was a huge asset to the development Eirguard as it gave me a firm foothold on where to pitch the baseline implementation of governance within the app. That is, what level of policy and procedure would be required in order for an organisation to ensure a robust cybersecurity governance structure.

The second vision was inspired by the introduction of the NIS2 regulations by the EU. These regulations introduced new obligations on industries across the EU to harmonise and create a common cybersecurity standard in the bloc.

Organisations are categorised by industry, size and turnover into one of two categories – essential or important. Entities can also be designated as critical in terms of Directive (EU) 2022/2557[6]. Those falling into the essential category now have regulatory obligations to

adhere to, and also need to produce documentary proof of adherence to those obligations at any time when requested by the new supervisory body.

Eirguard’s vision is to make it easy for an organisation to quickly establish if they fall under the requirements of the critical category, and assist with the creation and storing of documents. By using CSF 2.0 as the baseline for all users, it made adding the regulatory documentation relatively straightforward.

### 2.3 Desired Functionality

Function	Implementation
<b>Capture orgs current profile</b>	Questionnaire
<b>Gap analysis</b>	Devise a way that makes sense
<b>Progress Tracking</b>	Something like Reminders perhaps?
<b>Report Generation</b>	Templates for pdf report completion
<b>NIS2 check</b>	Simple sector, size, and revenue check
<b>Actions</b>	Concrete implementation suggestions

*Table 1: Desired Functionality*

Table 1 outlines the main goals I had in mind for the final version of Eirguard. Containing the focus entirely on governance only, allowed me to avoid potential scope creep. Or at least keep the creep to a bare minimum. Section 5.3 addresses this further.



### 3. Technical Development

Developing a mobile application requires significant time and research into platforms, technologies and development strategies. I opted to develop Eirguard for iOS for a number of reasons: security, market penetration and ease of use being the main ones. I go into much greater detail behind the reasoning of the platform choices, and how the decision was made in both my project specification and research documents. However, after recent EU decisions around the introduction of the 2024 Digital Markets Act (DMA) [7], I feel further discussion is required here, as this issue was not considered in the original reasoning.

The DMA requires all “gatekeepers” (large digital services providers) allow third parties to offer their products on other stores or from their own store.

*3. The gatekeeper shall not prevent business users from offering the same products or services to end users through third-party online intermediation services or through their own direct online sales channel at prices or conditions that are different from those offered through the online intermediation services of the gatekeeper.*

Before the introduction of the DMA, the only way to download apps for your iOS device was through Apple’s App Store. This allowed Apple great power in deciding what applications were allowed onto its platform. Apple thoroughly vetted every app submitted for inclusion on the store, and rejected those that did not conform with their stringent guidelines.

Due to the implementation of the DMA Apple now have to allow third party app stores to enter their ecosystem to offer end users a choice in where they obtain their apps. For the moment, this requirement is only confined to countries within the EU, but there is a possibility that other territories may be looking at similar legislation.

While the DMA could be touted as a win for consumers who were routinely hit with the “Apple Tax”- their 30% commission on every sale or service offered through the App Store - it calls into question the security of the proposed third party stores.

Who will vet these stores? Will the apps in these stores be as vigorously tested as Apple’s App Store before being offered to consumers? One only has to look at the issues Google had with

the Play Store and the number of rogue applications available with malware, to see areas for concern. Again, I go over this in greater detail in my spec document.

Security was one of the main tenets upon which I based my decision to go with iOS over Android. For now, until we see how these third party app stores develop, as well as their approaches to app security and customer safeguarding, the decision to remain with the official Apple App Store is something I'd feel comfortable doing.

## 4. How Eirguard Works

### 4.1 A brief walkthrough

After first signup has been completed, Eirguard prompts the user to complete a questionnaire (Figure 2) to get a snapshot of their current cybersecurity stance. The 31 questions posed are taken from the CSF 2.0 framework's new section on governance.

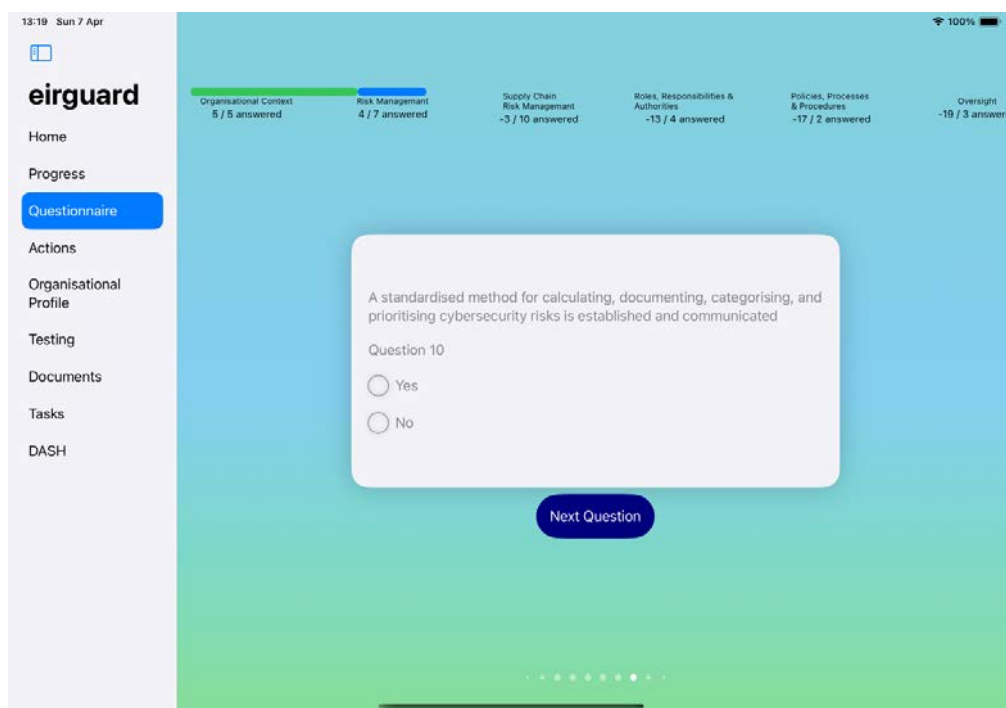


Figure 2: Eirguard's onboarding questionnaire

The questions have a binary yes/no answer to simplify the process, and only to determine if the procedure in the question has been implemented by the organisation. Once all questions have been answered, Eirguard returns an Action List of recommendations on questions the organisation answered 'no' to, that they can complete to satisfy each requirement (Figure 6).

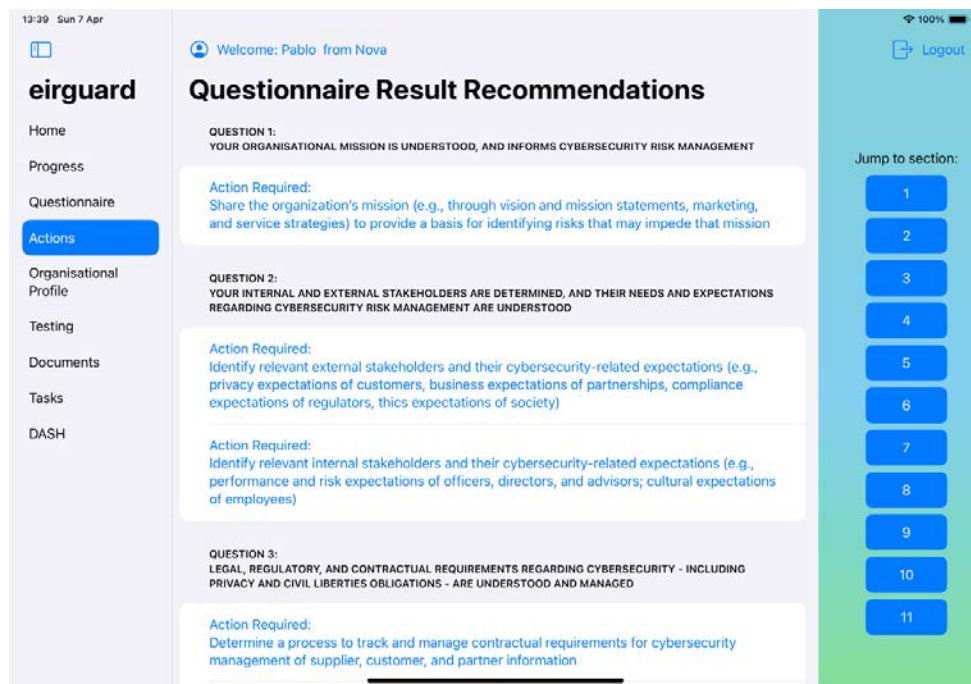


Figure 3: Recommendations list returned after questionnaire

Tapping on each action brings up a modal window that allows the user to add that action to the Task List. The Task List page allows the user to monitor what recommendations have been implemented, and what is yet to be done. When a recommendation is added to the Task List a new Task Group is created based on the original question. Some questions have more than one recommendation associated with them, so this grouping helps keep the tasks organised.

Organisational Profiles were introduced by CSF 2.0 to help organisations analyse the differences between their current stance, and the target stance of where they want to be. The final official release of CSF 2.0 at the end of February 2024, offered a downloadable Excel spreadsheet that organisations could fill out, that allowed side-by-side comparison of their Current and Target Profiles to help identify and analyse gaps.

I decided to incorporate the Organisational Profile into Eirguard because I felt it was important to keep all information relevant to the governance process in the same convenient location. Not only that, I felt that using the profile in conjunction with the questionnaire would help give the user a better understanding of where any cybersecurity gaps are in the organisation, and how best to deal with those.

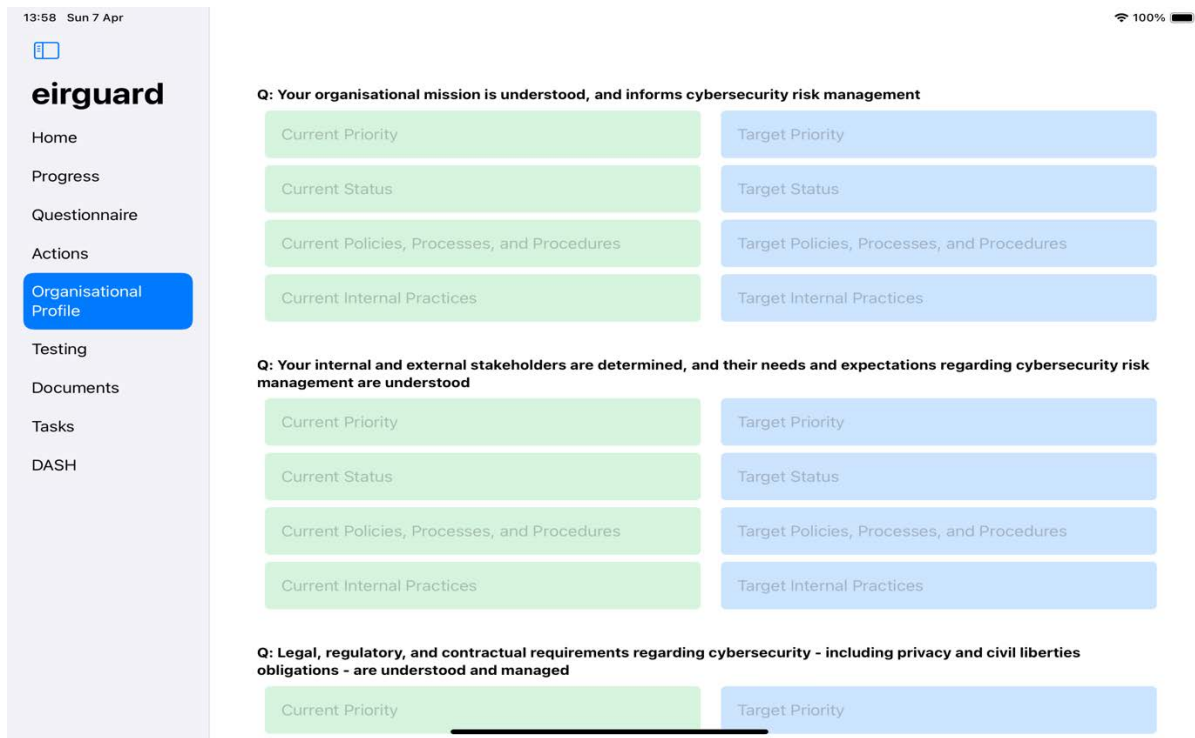


Figure 4: Organisational profile page

The organisational profile form in Eirguard can be filled in, edited, saved or exported as a pdf, where it can be printed, or shared with others in the organisation. It was originally envisaged that Eirguard would also help create documentation for the user to record ongoing progress and eventual completion of new governance policies. While this would have been a great addition, unfortunately time constraints did not allow for it.

However, I felt that documentation played a vital role in the process. After incorporating the organisational profile spreadsheet into Eirguard, I wanted to ensure that the user was able to continue to have a seamless experience. Therefore, to maintain a store of all documents relating to governance, I implemented a document scanning function. This allows the user to scan single or multi-page documents using the camera on the device, and save them as pdf's. While custom document creation in-app may have to wait for the next iteration, for the moment, users can scan printed documents related to the project, and have those at their fingertips whenever they are required.

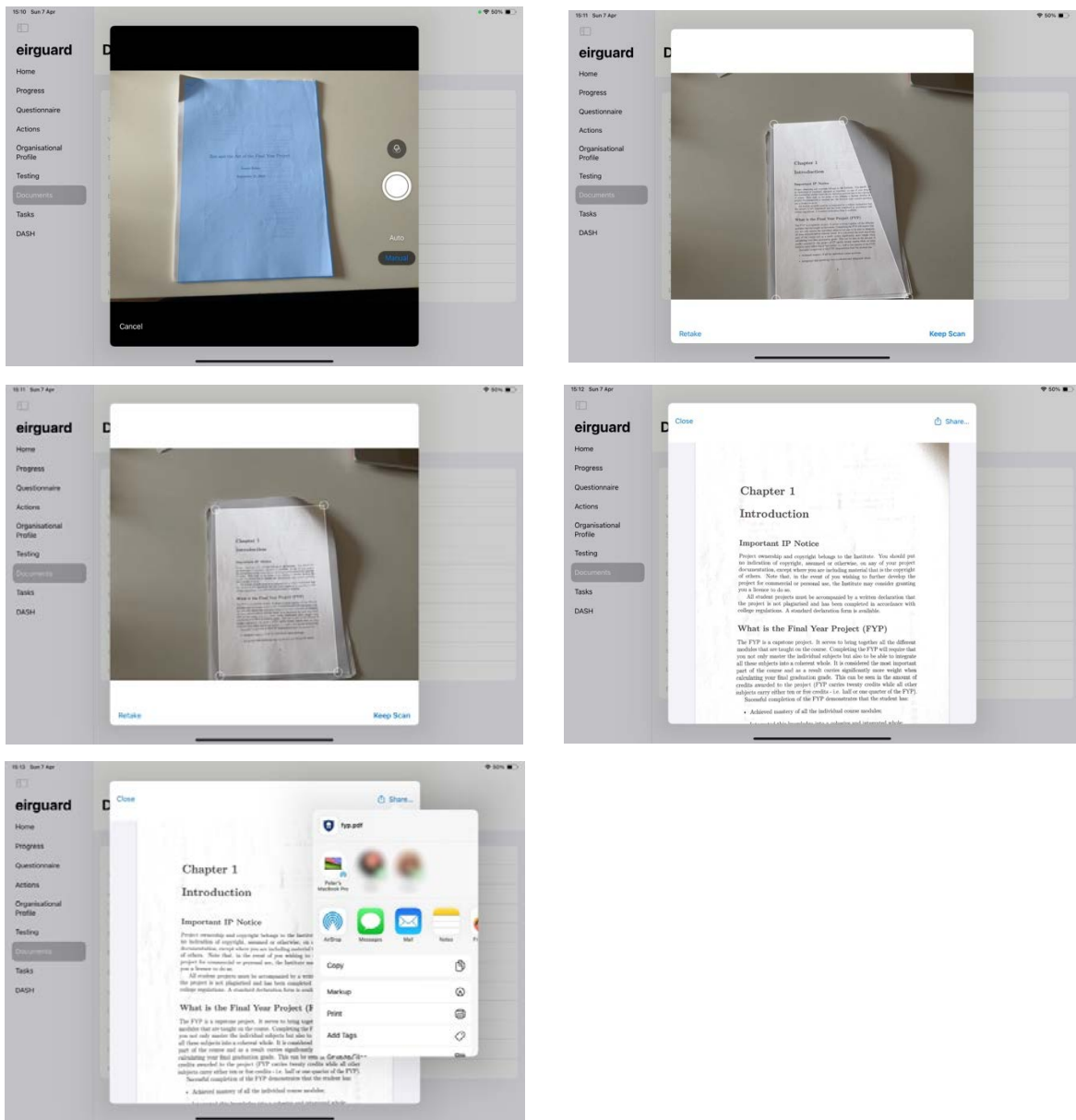


Figure 5: Process of scanning documents

To check their overall progress at a glance the user can navigate to the Progress page. Here they can view progress to date in each of the sections represented by circular gauges that display how far along each section has progressed. This view also presents the user with a Spider Chart, for a more detailed breakdown.



Figure 6: Progress page

## 4.2 Testing

As part of my research I asked several SME's to complete a small survey (full details contained in the research document). I had also asked if they would be willing to provide feedback on the application once it was ready for testing. Two of the SME's agreed to provide feedback, which I was happy with.

To beta test an iOS application, one first needs a paid Apple Developer account. Only then can you submit your app for testing. Apple has a stand-alone testing service called Testflight [8], that the app can be pushed to. Potential testers are then sent email invite links to their AppleID's from your developer account dashboard, to sign up and test the app, which will then become visible to them in Testflight.

However, this can only be done if the application passes Apple's stringent checks on first submission to the App Store. These are notoriously strict checks, in place to ensure all code aligns with best practice regarding formatting, security, and efficiency. The app will be rejected until all issues have been rectified. As this can sometimes be a lengthy process, the results of any testing carried out may not be known by the publication of this report.

Any feedback gained from testing (time permitting) takes the form of a loosely structured survey that asks the user about ease of use, quality of the functionality, quality of the user interface and overall impressions. Specific questions on whether it met their expectations, and what they felt it missed out on are also included. General feedback on the flow of the app and the overall useability is also requested. Finally, the user is asked to detail anything they felt was missing, that could improve how the app worked that they felt should be included.



## 5. Development Process and Technical Issues

### 5.1 The Application

When approaching the design of the application I first had to decide what it was I wanted the app to do, and then investigate how these could be implemented. The former was quite straightforward – pen, paper and a limitless imagination was all that was required here. The latter however poured a cold splash of reality into the equation, and helped inform the general direction of development.

Applications for Apple’s ecosystem are developed using Swift, a C based language. While I had some experience with C++, I had not used Swift extensively before – I had dabbled with it some years prior - so there was a steep learning curve involved in getting to grips with the language, as well as the implementation that took me some time to figure out.

In order to develop for the Apple ecosystem you need an Apple Developer account. No issue there, as there’s a free option available. The issues arose when trying to provision the signing and capabilities section. Xcode (Apple’s Integrated Development Environment)[9] kept throwing an error when I tried to run the project, pointing to the signing as an issue. It took longer than necessary to figure out what the issue was, toggling checkboxes, entering settings – with no definite answer forthcoming from Xcode. After much frustration I finally discovered that Apple had changed the signing and provisioning procedure for the newest version of Xcode and iOS 17. Once I created a fake Team, and assigned that to my AppleID in Xcode – I was good to go.

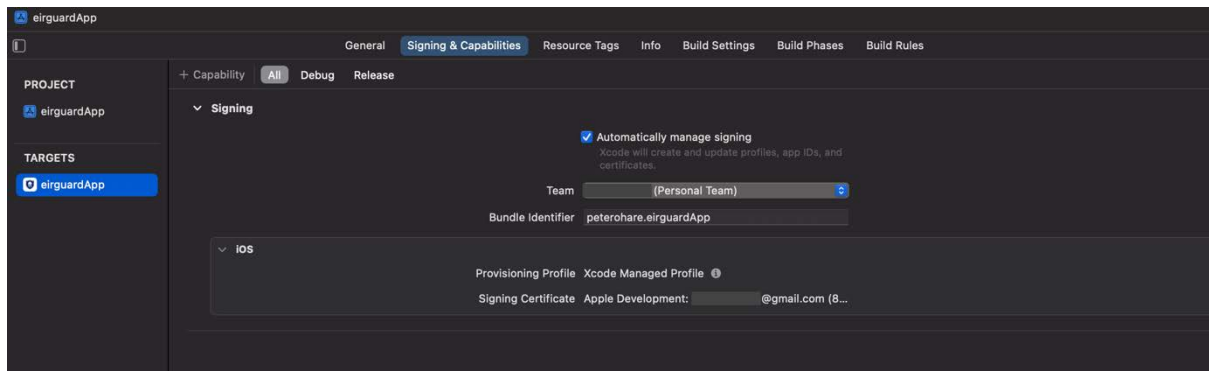


Figure 7: Signing and Capabilities in Xcode. Not as easy as it looks...

Despite the growing pains with Xcode, I found Swift to be fairly straightforward once I'd figured out the nuances and differences of the language. However, this wasn't helped, when I discovered that I'd inadvertently started with an MVC (Model, View, Controller) approach after following some older guides, instead of the newest MVVM (Model, View, View Model) approach that Apple introduced with iOS15. Thankfully, I was still early in the development process, so the impact wasn't as severe as it could have been. Apple tend to change things very quickly, and introduce new ways of doing things, while quietly removing the old ways that one might be more familiar with. Cutting edge – maybe. Annoying – most certainly.

I was also able to take an iterative and modular approach to the implementation of the application. I'd focus on getting one particular aspect of the design functioning correctly, and then move on to another area. I was then able to integrate those sections that needed to be linked together and tackle any issues that arose. As annoying as it was having to switch to the MVVM structure after starting – I found that it offered a very flexible way of working, and allowed for the modular approach, which suited my style very well. Perhaps I'll forgive them a little for that.

While Apple's documentation is very comprehensive and verbose, it can be overwhelming at times. To help get around any roadblocks I encountered, I sought out alternative sources of information. I had StackOverflow on speed-dial, and utilised ChatGPT or Google's Gemini to explain concepts that I didn't fully understand.

One of those concepts was the use of Apple's older user interface framework - UIKit [10]. Apple moved from UIKit to SwiftUI[11] when they migrated all platform development over

from Objective C to Swift. However, not all the functionality of UIKit was ported over to SwiftUI, which left me trying to figure out some Objective C and UIKit to achieve the result I wanted. The biggest example of this was utilising the camera as well as the app file storage to create the document scanning functionality.

I discovered that SwiftUI wouldn't allow me the granular approach I needed to access the deeper functionality of the system. I needed to access the system's *FileManager* so I could save the scanned document as a pdf. Unfortunately SwiftUI does not allow for such a granular approach, so I had to figure out how to pull UIKit functionality into SwiftUI, and work from there. This resulted in some very frustrating moments and inevitable loss of time while the issue was investigated and solved.

## 5.2 The Back End

I had initially decided to go with Google's Firebase[12] for the backend functionality – because it seemed to be the industry standard for app deployment. However, during the research phase I discovered Supabase[13], an open source alternative to Firebase. Being a fan of open source – I opted to move forward with Supabase.

Supabase uses PostgreSQL for their database, and while it is similar to MySQL – there are a few fundamental differences between the two. Most of the differences relate to syntax in how queries are written. I had many database errors turn out to be my using MySQL syntax in the query instead of Postgres syntax.

PostgreSQL allows for the creation of functions on the server to help simplify and automate tasks. This functionality came into its own when I was dealing with users on the database. Supabase abstracts users and their data into their own table that is segregated from the rest of the applications tables. The user table is encrypted and can only be referenced as part of a login event, and cannot be accessed by any other query for any other reason. This presents a problem when you want to store more user information beyond their email and login details. However, functions can help create a solution (Fig. 2).

```

6 CREATE OR REPLACE FUNCTION handleNewuser()
7 RETURNS trigger AS $$
8 DECLARE
9     firstname TEXT;
10    lastname TEXT;
11    bcompanyname TEXT;
12    new_org_id INT;
13
14 BEGIN
15     -- Extract user data safely with default values
16
17     IF new.raw_user_meta_data --> 'firstname' IS NOT NULL THEN
18         firstname := new.raw_user_meta_data --> 'firstname';
19     ELSE
20         firstname := '';
21     END IF;
22
23     IF new.raw_user_meta_data --> 'lastname' IS NOT NULL THEN
24         lastname := new.raw_user_meta_data --> 'lastname';
25     ELSE
26         lastname := '';
27     END IF;
28
29     IF new.raw_user_meta_data --> 'companyname' IS NOT NULL THEN
30         bcompanyname := new.raw_user_meta_data --> 'companyname';
31     ELSE
32         bcompanyname := '';
33     END IF;
34
35     -- -- Validate company name existence and handle non-uniqueness
36 BEGIN
37     -- Check if company name exists
38     IF EXISTS (
39         SELECT 1 FROM public.orgname WHERE bcompanyname = companyname LIMIT 1
40     ) THEN
41         RAISE EXCEPTION 'Company name already exists';
42     END IF;
43
44     INSERT INTO public.orgname(companyname)
45     VALUES(bcompanyname)
46     RETURNING id INTO new_org_id;
47
48 END;
49
50
51 -- Insert into profile with proper error handling
52 BEGIN
53     INSERT INTO public.profiles(id, firstname, lastname, companyname, orgid)
54     VALUES(new.id, firstname, lastname, bcompanyname, new_org_id);
55     EXCEPTION
56     WHEN others THEN
57         RAISE EXCEPTION 'Error occurred while inserting into profiles';
58 END;
59
60 RETURN new;
61 EXCEPTION
62 WHEN others THEN
63     RAISE EXCEPTION 'Error occurred while adding new user';
64 END;

```

Figure 8: PostgreSQL function to create a 'User Profile' table entry when a new user signs up

Supabase user auth allows for the addition of extra user information through the *userMetaData* parameter. Figure 9 depicts the Swift signup function:

```

113
114 // user signup
115 func signup() async throws {
116     // the data parameter for adding user metadata expects JSON dictionary
117     // here userMetaData creates the String:AnyJSON dictionary, with the user input, that the 'data' parameter expects
118     let userMetaData: [String: AnyJSON] = [
119         "firstname": .string(firstName),
120         "lastname": .string(lastName),
121         "companyname": .string(companyName)
122     ]
123
124     let response = try await supabase.auth.signup(
125         email:email,
126         password:password,
127         data: userMetaData // <-- this data is for adding extra parameters into user metadata in the auth table
128         //redirectTo: URL(string: "eirguardsec://Testing")
129     )
130
131     let checkAuth = try await supabase.auth.session
132     print(checkAuth.user.aud)
133     if checkAuth.user.aud == "authenticated"{
134
135     }
136 }

```

Figure 9: Swift user signup function

At the beginning of the function, the extra user information we want to include with the sign up is defined in the *userMetaData* as a JSON array. This array is then added as an optional parameter to the main signup query - that I mysteriously named 'response' for some strange reason.

On the Supabase end, as soon as a new user signs up, it triggers the function in *Figure 8* to execute and pull the information from *userMetaData*. This is then inserted into the user profile table allowing the additional information to be accessible. All information on the database is also encrypted to increase security and protect users data.

Learning PostgreSQL, while at times frustrating, has added a new tool to my skillset and, allowed me to streamline the main Eirguard codebase by utilising the strengths of Postgres to abstract some of the heavy lifting away. I have used triggers and functions extensively on Supabase to achieve exactly what I required. The organisation of the questions for the onboarding questionnaire, for example, as well as including their sections and total number per section, then grouping suggestions based on the answers submitted by the user was all carried out using triggers and functions to return exactly the data I needed into the application. It made for a very straightforward implementation – once I'd ironed out all the user error issues, that is.

Considering I'd never heard of Supabase before researching for this project, I'm genuinely surprised that more people haven't migrated away from Firebase. The positives of using Supabase and PostgreSQL most definitely outweighed any negatives, and I'm very satisfied with Supabase and its integration into Eirguard.

### 5.3 Scope Creep

Let's face it, there was *always* going to be a degree of scope creep, no matter how hard I tried to avoid it! The allure of adding something really cool can sometimes be too overwhelming to ignore. For Eirguard, the creep actually helped improve the functionality in my opinion. The addition of the document scanner was not something that was in the original design. The idea to add it came later during discussions with my supervisor. Once I realised that the timeline wouldn't allow for full document template creation, the scanner filled that gap nicely.

## 6. Personal Reflection

The old adage “if I only knew then, what I know now...” certainly jumps to the forefront of my mind when I look back on the last year. The FYP is something that essentially develops a life of its own, almost taking on a persona to either encourage you or berate you in equal measure, as you tackle the challenges that it presents.

How one approaches it will certainly say a lot about your character, commitment and work ethic. It will challenge your patience, your sanity, your belief in yourself, but above all it will force you take charge and push through. The FYP does not take prisoners, it does not make things easy, it engages parts of you that you may not even have known you had. Ultimately, it’s something for you to tackle head on, and strive to emerge at the other end victorious.

Of course each person will approach that cause differently, and with differing amounts of energy. Victory is unique to each person and depends on their character, their makeup and the amount of effort they are willing to expend. I’d like to outline some of my personal experiences, observations and milestones with the FYP here.

### 6.1 Challenges

*Life is never fair, and perhaps it is a good thing for most of us that it is not.*

— Oscar Wilde

Challenges are obstacles to achieving what it is we want in life. Some challenges are small in the grand scheme of things – like achieving a high score in a game, or finally being able to play a much loved song on the piano. Without challenges in life there would be no sense of achievement. No feeling of success for the little wins. Nothing to force us to get better, improve and overcome. The FYP was no exception.

There were many challenges faced along the way. The first being figuring out what my project would be in the first place. Devising a plan for how it could be achieved and then implementing that plan. During the course of my journey I had to pivot a few times, make tough decisions

and embark on a few things that I really didn't want to do. But the project needed those things to reach its ultimate destination.

Time management is always going to be something I underestimate I fear. The amount of times I found myself needing to crunch to get something done was far greater than I'd like to admit. Now, don't get me wrong – I did improve as I progressed, but now I do pine for those precious hours that were wasted, so I can reinvest them in the project and achieve something more akin to my original idea. My key takeaway here is that no matter how long I think something is going to take – double it, and add 20% - just to be sure.

Learning an entirely new language to develop this project was something that I actually wanted to take on. I wanted to push myself outside of my comfort zone and see if I could deliver something tangible and useful at the end of it all.

There were indeed times that my sanity was strained, as I faced what at the time seemed like insurmountable issues. Panic was never far away as the enormity of the situation would take hold and scare the wits out of me. However, I discovered that in spite of the uncertainty and fear, I am also a tenacious individual who won't rest until a solution is found. Ok, so it might have taken a little longer while I waited for the panic to recede, but I found that I wasn't about to let anything get the better of me. Least of all a programming language.

Another area of difficulty was encountered by my decision to base Eirguard on the new CSF 2.0 framework. When I began the project it was still undergoing a complete review, and no definitive final version had been released. I made the decision to move forward with it because I felt basing the application on the then current CSF framework wouldn't make sense with its replacement so close to release.

Information was thin on the ground, and no supplementary materials were yet available. At times I found myself floundering, wondering what direction I should take. There were a few false starts, and moves in the wrong direction, which were annoying when I had to roll back and start again. However, my supervisor was immensely helpful and resourceful as we navigated the uncertainty. He steadied me and helped guide me through what was a very spartan landscape, and how to figure out what were the areas of importance. Never

underestimate the power of impartial and detached eyes, to calmly spot the copious amount of wood, when you are manically running around in circles because all you can see are trees!

By forcing myself into areas where I felt overwhelmed at times, and fearful at others, it made me let go of some of the preconceived notions that I had about myself. Too old, too slow, not good enough. These ideas had to be firmly put by the wayside to plough though. Belief in oneself is the most powerful ally one has.

However all those challenges listed previously pale in comparison to the biggest challenge I faced during the project. The week before we were due to return to campus in September to begin the new 2023/24 academic year, my mother passed away suddenly. It was a shock to us all, as she was always the on top of life. Losing her so quickly was a body blow, and difficult to deal with, so I threw myself into my studies thinking everything would be ok. Grief had other ideas, and those first few months were some of the most difficult I've ever experienced.

They say that what doesn't kill you makes you stronger. Thankfully I had family, good friends and an understanding supervisor to help me through. That experience has taught me to never take anything in life for granted. You never know what's around the next corner. Don't just exist – live life, push yourself and be thankful for what you have.

## 6.2 Lessons Learned

Looking back over the whole process there are certainly some things I'd approach differently. Research and planning is something I would certainly devote more time to. It's not a matter of if something *can* be done. You also need to look at *how* it can be done, and then look at ways on how it can be done *better*.

I have a habit of getting excited about something and diving in to start without looking at the overall big picture (hark back to the MVC v MVVM scenario from earlier). While enthusiasm is to be applauded, half-cocked enthusiasm is something to be avoided if possible.

My half-cocked enthusiasm probably cost me a lot of time as well. Time lost to half researching something. Then diving in to get started, hitting a wall after some time, and then discovering



that if I'd researched just a *little* bit harder, I would have discovered the exact same issue I was now facing, as well as a better way to achieve the same end result. Time invested in proper research is not time wasted. It can save so many headaches down the line.

Would I still go with an iOS application if I could do it all over again? Strangely enough, despite all of the frustrations and issues I faced – I probably would. There is a profound sense of accomplishment achieved when I can pick up an iPad and point to an application that I created, that fills a need for some people. I can hold that up high and say I didn't let it beat me!

### 6.3 Skill Development

It turns out you can teach an old dog new tricks! iOS development is something I'd like to keep my finger on moving forward. Not that I plan on becoming a full time developer, quite the opposite in fact, but having that extra skill, and knowledge of the ecosystem helps me add a new badge to my toolbelt. Taking the plunge and not being afraid to dive into new things, while frustrating at times, ultimately turned out to be quite rewarding.

However, where there is positive, there must be negative – and that is also true of myself. At the same time I discovered that project management is something I need to improve upon. It's all well and good keeping the whole process in my head, but moving forward, that way of working just won't cut it. Especially on something as large as this project was. I need to develop processes and procedures to structure tasks more efficiently, and learn to create, stick to, and hit definite deadlines. The irony isn't lost on me that my application is completely geared toward developing processes and procedures.

All told, I found the process to be a rewarding one. Yes, there are things I'd definitely do differently should I have to do it all again. But, that's all part of the learning process. I learned some things about myself that I was quite happy with, and I definitely took away things that I now know I could improve on. I think that's a very positive place to finish the project on, and moving forward I can see the benefits of the process.

## 7. Conclusion

Did I achieve everything I wanted to with Eirguard? No.

Am I happy with the end result? Yes.

While the final outcome is not exactly how I envisioned Eirguard to be when I sat down to plan it in the beginning, it has evolved and grown in its own way. It has taken on a life of its own. It has morphed from being just a simple idea, into something that can ultimately prove useful. Don't get me wrong, there is still plenty that can be improved upon, and I'm very aware of that. But as a proof of concept, I think it strikes all the right notes. I would certainly love to be able to take it further, just to see what can be achieved.

The entire process of the Final Year Project has been a roller-coaster of emotion - failure, success, highs, lows and doubts. Despite all of that, the one thing I could never describe it as - is boring!

## 8. References

- [1] 'Everything you need to know about the NIS2 Directive | Enhanced Reader'. Accessed: Nov. 23, 2023. [Online]. Available: <https://www.eversheds-sutherland.com/lists/static/uploads/nis2-whitepaper.pdf>
- [2] 'COBIT | Control Objectives for Information Technologies | ISACA'. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.isaca.org/resources/cobit>
- [3] 'ISO/IEC 27001:2022 - Information security, cybersecurity and privacy protection — Information security management systems — Requirements'. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.iso.org/standard/27001>
- [4] 'NIST CSF Compliance Automation Software | ControlMap'. Accessed: Nov. 23, 2023. [Online]. Available: <https://www.controlmap.io/nist-csf-compliance-automation-software>
- [5] 'Updating the NIST Cybersecurity Framework – Journey To CSF 2.0 | NIST'. Accessed: Dec. 05, 2023. [Online]. Available: <https://www.nist.gov/cyberframework/updating-nist-cybersecurity-framework-journey-csf-20>
- [6] 'Directive - 2022/2557 - EN - CER - EUR-Lex'. Accessed: Apr. 19, 2024. [Online]. Available: <https://eur-lex.europa.eu/eli/dir/2022/2557/oj>
- [7] 'Digital Markets Act'. Accessed: Apr. 19, 2024. [Online]. Available: [https://digital-markets-act.ec.europa.eu/index\\_en](https://digital-markets-act.ec.europa.eu/index_en)
- [8] 'TestFlight - Apple Developer'. Accessed: Apr. 19, 2024. [Online]. Available: <https://developer.apple.com/testflight/>
- [9] 'Xcode 15 - Apple Developer'. Accessed: Apr. 19, 2024. [Online]. Available: <https://developer.apple.com/xcode/>
- [10] 'UIKit | Apple Developer Documentation'. Accessed: Apr. 19, 2024. [Online]. Available: <https://developer.apple.com/documentation/uikit>
- [11] 'SwiftUI | Apple Developer Documentation'. Accessed: Apr. 19, 2024. [Online]. Available: <https://developer.apple.com/documentation/swiftui/>
- [12] 'Firebase | Google's Mobile and Web App Development Platform'. Accessed: Apr. 19, 2024. [Online]. Available: <https://firebase.google.com/>
- [13] 'Supabase | The Open Source Firebase Alternative'. Accessed: Apr. 19, 2024. [Online]. Available: <https://supabase.com/>

