

CITY UNIVERSITY OF LONDON

(BSC) COMPUTER SCIENCE

MediPill

A healthcare app to aid care home workers with in-patient records and regular pill reminders

2022 -
2023

Ayesha Kayani

AYESHA.KAYANI@CITY.AC.UK

Table of Contents

Abstract	5
1 Introduction	6
1.1 Problem to Solve.....	6
1.2 Objectives.....	6
1.2.1 Primary Objectives.....	6
1.2.2 Secondary Objectives	6
1.3 Project Beneficiaries	7
1.4 Project Limitations	7
1.5 Risks.....	7
1.5.1 Risks to my Project	7
1.5.2 Risks that my project poses to others.....	8
1.6 Risk Register	9
2 Output summary.....	10
2.1 iOS Mobile Application Source Code	10
2.2 READ ME – Main Submission	10
2.3 READ ME	10
2.4 Instruction Manual.....	10
2.5 Participant Information sheet.....	10
3 Literature Review.....	11
3.1 Analysis of existing Software	11
3.2 Tools and libraries for development	12
3.2.1 Swift.....	12
3.2.2 Firebase	12
3.2.3 Figma	12
3.3 Third-party resources.....	13
3.3.1 Notion.....	13
3.3.2 Canva	13
3.3.3 YouTube.....	13
3.3.4 Qualtrics XM	13
3.3.5 Pexels.....	13
4 Method.....	14
4.1 Software Development Methodology	14
4.2 System Requirements	14
4.2.1 System Requirements Gathering	14
4.2.2 Initial Domain Model - CRC Cards	14
4.2.3 Use case Index.....	14
4.2.4 Usage Model - Use case Specification.	14
4.2.5 Entity relationship diagram	15
4.3 Design Documentation	15
4.3.1 App Logo.....	15
4.3.2 User Interface Model - User interface & User Experience (UI & UX).....	15
4.3.3 Usability.....	15
4.3.4 Accessibility	15
4.3.5 IOS Design Guidelines.....	15

4.3.6	Use of IOS Native Components – Navigation Bar	16
4.4	Prototype UI Development.....	16
4.4.1	Hand–drawn UI prototype.	16
4.4.2	UI Mock-ups	16
4.4.3	UX - Interactive Prototype.....	17
4.5	Implementation	17
4.5.1	Development Environment	17
4.5.2	Programming Languages.....	17
4.5.3	UI Design - Version Control	17
4.5.4	Xcode Set Up Process.....	17
4.6	Database Implementation	18
4.6.1	Firebase	18
4.6.2	Sign Up Feature	18
4.6.3	Login Feature.....	19
4.6.4	Notification Implementation.....	19
4.7	Testing.....	19
5	Results.....	20
5.1	Software Development Methodology	20
5.1.1	Agile Development	20
5.1.2	Steps Taken	20
5.2	System Requirements	20
5.2.1	System Requirements Gathering – APPENDIX C	20
5.2.2	Initial Domain Model - CRC Cards – APPENDIX D	20
5.2.3	Use case Index – Iteration - APPENDIX E	20
5.2.4	Usage Model - Use case Specification – APPENDIX F	20
5.2.5	Entity relationship diagram – APPENDIX G	20
5.3	Design Documentation	20
5.3.1	App Logo.....	20
	Refer to the App Logo Survey Results – Appendix H	20
5.3.2	User Interface Model - User interface & User Experience (UI & UX).....	20
	Refer to the UI Development Survey Results – Appendix H	20
5.3.3	Usability.....	22
5.3.4	Accessibility	22
5.3.5	IOS Design Guidelines.....	22
5.3.6	Use of IOS Native Components – Navigation Bar	23
5.4	Prototype Development.....	23
5.4.1	Hand–drawn prototype.....	23
5.4.2	Mock-ups.....	24
5.4.3	Interactive Prototype	24
5.5	Implementation	25
5.5.1	Development Environment	25
5.5.2	Programming Languages.....	25
5.5.3	UI Design - Version Control	25
5.5.4	Xcode Set Up Process.....	25
5.6	Database Implementation	26
5.6.1	Firebase	26
5.6.2	Firebase Authentication	26
5.6.3	Sign Up Feature	27
5.6.4	Login Feature.....	27
5.6.5	Set Up Patient Profile Feature.....	27
5.6.6	View Patient Profile Feature	28
5.6.7	Add / Edit / Delete Patient Medication Feature	28

5.6.8	Add / View Appointment Feature	29
5.6.9	Notification Feature	29
5.7	Non-Database Implementation	30
5.7.1	Launch Screen	30
5.7.2	Onboarding Screen.....	30
5.7.3	Log in/ Sign up Screen	30
5.7.4	Search Patient /Medicine Feature	30
5.7.5	Patient List Screen	32
5.7.6	Interactive Calendar Feature.....	32
5.8	Testing.....	32
	Refer to the App Testing Survey Results – Appendix H	32
6	Conclusions.....	33
6.1	Project Objectives Review	33
6.2	Conclusions drawn from the project.	33
6.3	Implications for future work.....	33
6.4	Personal Progress.....	33
7	References.....	35
8	Appendix	37
	(APPENDIX A) – PDD.....	37
	(APPENDIX B) – Code Reuse (Adapted Code).....	46
	(APPENDIX C) – System Requirements Gathering	51
	(APPENDIX D) – Initial Domain Model - CRC Cards.....	55
	(APPENDIX E) – Use case Index - Iterations.....	55
	(APPENDIX F) – Usage Model - Use case Specification.	57
	(APPENDIX G) – Entity relationship diagram.....	65
	(APPENDIX H) – Qualtrics Survey Reports.....	66
	App Logo Survey	66
	MediPill UI Development Survey.....	70
	MediPill App Testing Survey	75
	(APPENDIX I) – User interface & User Experience (UI & UX)	78
	(APPENDIX J) – Final Product	80
	Notifications.....	80
	Onboarding Screens	81
	Login / Sign Up Screens	82
	Patient List Screen	83
	Set Up Patient Profile screen.....	84
	Patient Profile Screen	85
	Edit/Delete Medication Screens.....	86
	Add Appointment Screen	87
	Calendar Screen.....	88

Abstract

My proposed iOS healthcare app is a prototype designed to improve the quality of care provided to care home residents. It facilitates medication management, appointment scheduling, and patient record keeping.

The app will allow staff to easily schedule and send medication and appointment reminders to residents, ensuring that medications are taken on time and appointments are not missed. In addition, the app enables staff members to conveniently access and modify patient records, providing a comprehensive overview of each resident's medical history and current health status.

As a university student, I aim to utilize my proficiency in iOS development and user-centric design to develop an app that is user-friendly and customized for the requirements of care home residents and staff. In the end, the app would serve to enhance the standard of care given to residents of care homes and facilitate and enhance the job of care home employees.

1 Introduction

1.1 Problem to Solve

Initially, I intended to create a pill reminder app as this was something I could relate to. Living with elderly members I would see how much medication they must regularly take, but often forget to or couldn't remember if they already took their pills that day or not.

After doing some research online and from the knowledge I gained from my internship, I was made aware of the insufficient use of technology within care homes. Having previously worked on a project for a well-known UK Healthcare Provider, my main role was to help with the integrations between their Hospitals and Care Homes. Although the company is a global competitor, you'd assume that they would have these basic integrations already in place.

The problem I am aiming to address is the insufficient use of technology within care homes, resulting in difficulty in managing patient medication reminders, appointment schedules, and medical records. The solution I am proposing is a single app for Care Home workers incorporating these features to streamline and improve their business operations.

1.2 Objectives

Having primary and secondary objectives helps to clarify the project's purpose and priorities.

- Primary objectives - Main goals
- Secondary objectives – The supporting goals that help to achieve the primary objective.

1.2.1 Primary Objectives

1. **Develop an easy-to-use iOS healthcare app** that incorporates pill reminders, patient appointment reminders, and patient records for care home residents used by staff.
2. **Streamline the medication management process** for care home staff, reducing the risk of medication errors and improving patient outcomes.
3. Enhance patient care by **improving appointment scheduling** to reduce missed appointments.
4. Allow care-home staff, to **access patient records in real-time** to make informed decisions.

1.2.2 Secondary Objectives

1. Allow for **easy data entry and record keeping**.
2. Provide **customizable reminders and notifications** to ensure that residents take their medication on time and attend appointments.

1.3 Project Beneficiaries

My prototype has no current beneficiaries, however, if it were deployed in a real-life setting, **the solution's proposed beneficiaries would care home residents and care home staff.**

To assist the residents in maintaining their health and well-being, the app would remind the care home staff about any upcoming medications or appointments for their patients. Also, it would reduce medication errors and enhance efficiency among Care Home staff in managing medication schedules, appointments, and patient records.

1.4 Project Limitations

As a prototype developed by a university student without any real beneficiaries, there are some limitations to the proposed project.

Some potential limitations include:

1. **Limited Resources:** As a student, there may be limited resources, including time, funding, and access to healthcare professionals along with the technology resources I may need to further develop my app. This may impact the overall quality of my app.
2. **Limited Feedback:** Without real beneficiaries, it may be challenging to gather comprehensive feedback from the target users, which could limit the app's ability to meet the unique needs and requirements of care home staff and residents.
3. **Limited Real-World Testing:** Without real-world testing with actual beneficiaries, it may be challenging to determine the app's effectiveness and identify any areas that require improvement.

1.5 Risks

1.5.1 Risks to my Project

Below is a list of potential risks relating specifically to my project:

1. **User acceptance:** From my research earlier, I gathered that there may be resistance from care home staff to adopt new technology, affecting the success of the app. **To mitigate this, I will regularly gather feedback from those around me to understand their experiences with the application and identify areas for improvement.**
2. **Data privacy and security:** The app will be handling sensitive patient information, and it's important to ensure the protection of this data. **To mitigate this, I will be using a dummy dataset with fake patient details.**
3. **Integration with existing systems:** The app may need to integrate with existing systems used in care homes, leading to potential compatibility issues. **Since this is a student project, I will only be building a prototype so this isn't an issue, I will face. However, if this application was to be deployed for a customer to mitigate this risk, I would conduct a compatibility test of the new application with existing systems before deployment to identify and address any potential issues.**

1.5.2 Risks that my project poses to others.

Below is a list of potential risks for my project that may harm others and how to reduce them to acceptable levels:

- **Data privacy and security:** The app will handle sensitive patient information and it's important to ensure that this data is protected from unauthorized access or misuse. **I will be using a dummy dataset, however, if I were to deploy this project in a real-life setting, I would be using real patient records. To reduce any risks associated with this I will put in place appropriate security measures such as encryption, secure data storage, and access controls would be implemented and regularly tested.**
- **User error:** Care home staff may use the app incorrectly, leading to incorrect or inconsistent patient information. **This solution is only a prototype, so I will not face this risk; however, to reduce the risk of user error in a real-life setting, I will suggest my client provide comprehensive user training and support for the care home staff. Additionally, a user-friendly interface should be implemented in the app to help reduce this risk.**

(Kshitij, 2022)

1.6 Risk Register

A risk register is a document that helps to identify, assess, and manage potential risks that may impact a project. It is essentially a tool for recording and monitoring risks throughout the project lifecycle. A risk register typically contains a list of identified risks, along with their probability of occurrence the potential impact on the project, and any planned or implemented mitigation measures. Creating a risk register will allow me to manage possible risks in a proactive manner rather than just responding to them when they happen.

Risk	Likelihood	Impact	Severity	Mitigation
Technical Issues	High	High	9/10	Conduct regular code reviews and tests, identify and address issues early, and have a backup plan in place in case of major technical issues.
Inadequate User Testing	Medium	Medium	5/10	Conduct extensive user testing. Gather feedback and iterate on the app design and functionality and involve end-users in the development process.
Delayed Development	High	Medium	6/10	Set realistic timelines and milestones, prioritize tasks, and allocate resources effectively.
Incomplete Features	Medium	High	6/10	Prioritize essential features and functionalities, conduct regular progress reviews, and adapt to changing requirements. Have a contingency plan in case of incomplete features.
Physical health risk	Low	Medium	6/10	Balance work efficiently and screen time to ensure I am not overworked.

2 Output summary

2.1 iOS Mobile Application Source Code

Description	iOS Mobile Application Source Code refers to the software code that constitutes the programming instructions and logic behind a mobile application designed for Apple's iOS operating system.
Output Type	The output type of iOS Mobile Application Source Code is code I have written in Swift on XCode.
Link	Download the Zip file

2.2 READ ME – Main Submission

Description	This file includes where I have uploaded everything for the examiner to view
Output Type	Rich Text Format
Link	On Moodle - Project Main Submission Area (non- Confidential)

2.3 READ ME

Description	This is a text file for the examiner attached with my code. It states how to extract and run the code
Output Type	Rich Text Format
Appendix Link	On Moodle – Product Package

2.4 Instruction Manual

Description	Here I have written a step-by-step guide on how to run and use the app for the examiner.
Output Type	Rich Text Format
Appendix Link	On Moodle – Product Package

2.5 Participant Information sheet

Description	<p>The participant information template provides a clear and detailed explanation of the purpose of a healthcare app research study, and what participation entails.</p> <p>It covers aspects such as voluntary participation, confidentiality, data privacy, and the potential risks and benefits of participating. The template aims to obtain informed consent from potential participants.</p>
Output Type	Word Document
Appendix Link	On Moodle – Project Main Submission Area (Confidential)

3 Literature Review

3.1 Analysis of existing Software

The table below compares three popular healthcare apps; PillPack (by Amazon Pharmacy), Medisafe and Doctor on Demand, against my own. I have chosen to analyse these three apps in specific as they are widely used in the healthcare industry.

	PillPack	Medisafe	Doctor on Demand	MediPill (My Solution)
Websites	PillPack	Medisafe	Doctor on Demand	
Medication Reminders	Yes - Pre-sorted and packaged doses delivered to your door.	Yes - Reminders, refill tracking and interaction checking	No	Yes - Users will receive notifications on their work phones.
Virtual Doctor Visits	No	No	Yes	No – N/A Users will receive Doctor Appointment Reminders for their patients.
User-friendly Interface	Yes - Easy to use, with 24/7 support	Yes – Simple with customization options.	Yes - User-friendly app and website	Yes - User-friendly app
Prescription delivery	Yes	No	No	No – N/A
Medication tracking	Yes - PillPack tracks your prescriptions and refills	Yes - Tracks medication and sends reports to healthcare providers.	No	No.
Integration with healthcare providers	Yes - Collaborates with healthcare providers	Yes - Can be integrated with healthcare providers	Yes - Integrates with some healthcare providers.	Upcoming - Since I am only building a prototype this is not relevant yet.
Price	Varies by insurance and medications	Free with basic features. A premium version is available.	Varies by service	Since I'm building a prototype and have no stakeholders or beneficiaries involved, Pricing is not relevant yet.

Conclusion

In conclusion, the table above provides a helpful comparison of the apps. While PillPack and MediPill focus on medication delivery and tracking, Medisafe and Doctor on Demand offer virtual doctor visits. Which app to use will ultimately depend on personal needs and tastes.

3.2 Tools and libraries for development

A list of the tools and libraries I used to create my healthcare app is provided below.

3.2.1 Swift



Since Swift was created by Apple exclusively for their platforms, I opted to use it in Xcode over other programming languages mostly because doing so made sense when creating an iOS app. I also found that Swift is easy to use which in turn reduces my development time. There are also many videos online to aid me in my development.

3.2.2 Firebase



Initially, I intended to use MySQL, however, I ended up opting for firebase due to there, being an abundance of use for YouTube videos online to support me. Firebase provides a comprehensive and scalable back end solution to integrate seamlessly with an iOS app. It has an intuitive user interface, is simple to set up, and is simple to integrate with other technologies.

3.2.3 Figma



I chose to use Figma for my app UI development due to its many collaborative features, cloud-based storage and accessibility from any device, vector-based design tools, and the ability to create prototypes and animations. Figma also has an extensive library of UI components and plugins for me to work with.

3.3 Third-party resources

3.3.1 Notion



I used to notion keep myself organised during the development of my app. Notion is a great tool to easily organise information and embed links to external resources, such as YouTube videos that I could refer to add a later date or the different version controls of the UI I built on Figma. In general, Notion helped streamline my workflow, making it easier to keep track of progress and identify what jobs still needed to be done with their built-in Task Manager view.

3.3.2 Canva



I chose to use Cava to develop my logo over other tools, as I am already quite proficient in Canva. Canva offers a user-friendly interface and has a wide range of customizable design templates and elements for me to use. The pre-designed templates can also make it simpler for me, a non-designer, to swiftly and easily develop a logo that looks professional.

3.3.3 YouTube



Using YouTube as a resource for developing my app was incredibly helpful. There were plenty of tutorials and step-by-step guides that helped me through the app development process.

3.3.4 Qualtrics XM



Using Qualtrics for creating surveys during the app development process can be highly beneficial. The platform allowed me to design and distribute surveys to gather feedback from my participants. It also allows me to analyse the data

collected from these surveys so I can make informed decisions about my app's features, design, and functionality.

3.3.5 Pexels



As an excellent resource for finding high-quality images for my project, I utilized Pexels. For my app, I required a variety of visuals to improve the user experience and communicate information in an attractive manner. I was able to effortlessly search and download images that met my requirements on Pexels, without any concerns about licensing or copyright problems. In

general, Pexels saved me time and guaranteed a professional appearance and atmosphere for my app.

4 Method

4.1 Software Development Methodology

I have opted to use an agile software development technique while creating my app. In Agile software development, teamwork and continuous progress take centre stage with a fluid and adaptable methodology. The fast reaction to changing requirements and user feedback makes it perfect for developing healthcare apps in the real world. This is crucial in the healthcare sector since rules and patient needs are subject to quick change.

Since I am only building a prototype and have no stakeholders, most of the steps here are unnecessary, however, according to the agile development technique I would collaborate closely with the stakeholders, including care home staff in an Agile development approach, to collect requirements, prioritise features, and create a minimal viable product (MVP). I will constantly demonstrate the app during the development process to get input from individuals around me, enabling further product enhancement.

4.2 System Requirements

4.2.1 System Requirements Gathering

Even though the solution I'm developing is simply a prototype, obtaining requirements is still a crucial step. It entails determining the precise features and functionalities required in the app to achieve the project's core and secondary goals.

Additionally, it would entail ranking these requirements in order of relevance and development resources available. To make sure that they direct the design and development of the app, I described the requirements using a VOLERE template.

4.2.2 Initial Domain Model - CRC Cards

CRC card is a tool used in Agile to help design the responsibilities and collaborations between different classes of objects in a system. Using CRC Cards can aid in the comprehension of system functionality. I created the CRC cards on visual paradigm using my knowledge from 1st year.

4.2.3 Use case Index.

I created a Use Case Index to have a comprehensive list of all the functionalities that MediPill can perform. It assists me in recognizing the desired functionality and operations of the application. Having a use case index can help me track the progress of my development and prioritize which features should be implemented first through iteration; which is an Agile approach.

4.2.4 Usage Model - Use case Specification.

I developed a Use Case Specification for my iOS app to specify and record the functional needs of the system from the viewpoint of the user. The Use Case Specification aims to provide a detailed explanation of how the system should react to certain scenarios. This will ease my job of validating the software's intended operation and meeting user needs.

4.2.5 Entity relationship diagram

An Entity-Relationship (ER) diagram shows the information and connections between various entities within a system. To better understand the data that must be kept and how it is related, I designed an ER diagram for my iOS health app.

For instance, there will be connections between patients, prescriptions, appointments, and healthcare providers in my app. I may plan how the data will be arranged in the database of my app and view these relationships with the aid of an ER diagram.

4.3 Design Documentation

4.3.1 App Logo

To make the logo for my healthcare app, I used an online graphic design tool called Canva. First, I chose a template that matched my vision for the logo. Then, I modified it by changing the colours, fonts, and shapes until I got the right look. I decided to use a logo template that had a heart on it to symbolise the healthcare industry. After that, I downloaded the logo in different file formats so that it could work on different devices and platforms.

4.3.2 User Interface Model - User interface & User Experience (UI & UX)

I was able to quickly experiment with various design layouts, colours, and other design components to give my app a distinctive and captivating appeal by utilising Figma to construct the UI.

4.3.3 Usability

One of the key factors that I considered when developing my app was the thumb-friendly heat map analysis as shown below.



This analysis divides the screen into three main areas - Natural, Stretch, and Hard. The natural green area is the easiest to reach. The stretch area requires a little more effort and includes the upper half of the screen. The hard area is the most challenging to access and includes the corners of the screen.

I carefully placed the most important features of my app in the natural / stretch areas for easier access. I also

tried to minimize the use of hard areas. By considering the above map, I was able to create an app that was comfortable for the care home staff who will be using iPhones.

(Dholakia, n.d.)

4.3.4 Accessibility

Before I began to develop my app, it was essential for me to consider the guidelines set by the international standards from the WTC web accessibility initiative. International standards for web accessibility guide my app's development process to create inclusivity for all users.

4.3.5 IOS Design Guidelines

My app's UI was designed to match IOS's aesthetic by following Apple's IOS Design Guidelines. Typography, colour schemes, layout, and more were among the design's various elements.

The IOS Design Guidelines recommend using consistent visual language across the app. The app should feel and look consistent in its entirety. To accomplish this, I ensured consistency in the interface elements and made them complement the app's overall appearance. I also used a consistent colour scheme on all the screens of my app.

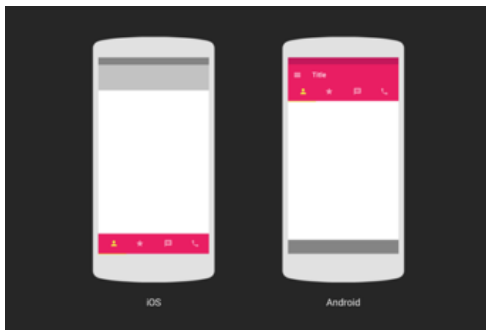
IOS Design Guidelines require the use of clear and concise language. The clear and concise text was a must in my app's design. I ensured the use of simple language across the app.

(Ideas, n.d.)

(Anon, 2020)

4.3.6 Use of IOS Native Components – Navigation Bar

IOS native refers to the set of design guidelines and features that are unique to the iOS platform. These features include the navigation bar, tab bar and other interface elements that are specific to iOS.



The navigation bar's location at the bottom of the screen is a significant characteristic of iOS native. The location of its positioning is not like those of Android apps, where the navigation bar is placed at the screen's top. Having the navigation bar at the bottom of the screen provides practicality and comfort to iPhone users accessing it with their thumbs. The app's user experience and usability can be enhanced simply by careful consideration of feature placement.

(Ideas, n.d.)

4.4 Prototype UI Development

4.4.1 Hand-drawn UI prototype.

When I began designing my app, I started off by drawing rough sketches on paper to get a clearer idea of its layout and possible functionality. This process helped me identify essential features and potential roadblocks.

As I refined the hand-drawn prototype, I was able to think about alternative user pathways and identify areas where the app might need additional functionality or clarification. This helped me create a truly user-centric app and avoid potential problems later. Overall, the hand-drawn prototype was an important tool in developing my app.

4.4.2 UI Mock-ups

I chose to then create a digital mock-up of my app UI Design using Figma. I was able to test out several colour schemes, typographies, and layout possibilities, allowing me to refine the design of the app and ensure that it was both visually appealing and easy to use.

I also used Figma to create banners to help promote the features of the app. To highlight essential aspects of the app, these banners can be placed on marketing materials. In addition to strengthening its brand identity and promoting its distinctive features, this can aid in increasing the app's visibility and drawing in new users.

4.4.3 UX - Interactive Prototype

Using the basic Figma UI design that I had just created I then went on to create an interactive prototype. The process involves creating clickable components and linking them together to simulate the user flow. By doing so, I can identify any usability issues or inconsistencies in the design and make necessary changes before development.

4.5 Implementation

4.5.1 Development Environment

The project outcome can be heavily influenced by the development environment and its tools. Xcode was utilized to create the app's front end. XCode's user-friendly interface makes code creation, testing, and debugging easy for developers. Xcode is equipped with Interface Builder and an integrated debugger which makes app creation possible for multiple platforms like iOS, macOS, and watchOS.

The Firebase real-time database was used to manage data storage and retrieval for the app. With the Firebase real-time database, the app effortlessly stored and retrieved patient data and files in a secure way while also enabling real-time data access. Firebase authentication integration facilitated user registration, login, and account management in the project.

4.5.2 Programming Languages

Swift and UIKit, were my top choices for primary programming languages. These languages in combination are excellent for developing an interactive UI. Swift's modern and strong features let me create concise and neat code that was easy to read and maintain. Moreover, I had access to a broad selection of pre-made instruments and assets within UIKit that simplified the creation of complex interfaces for me.

I used Storyboard to improve my app's user experience by visually organising its interfaces like buttons, text boxes and labels. Storyboard streamlined the task of improving the user experience by presenting a clear visual of the app's functionality.

4.5.3 UI Design - Version Control

Notion was essential during the design process for managing versions of the UI design. By creating a dedicated page for my UI development, I can quickly review and compare different iterations of the design, since I can also embed the Figma link into the dedicated Notion page.

4.5.4 Xcode Set Up Process

Setting up Xcode was the first step I took. I found the process to be straightforward and intuitive. I had already had Xcode downloaded on my Mac. I simply opened Xcode and created a new project.

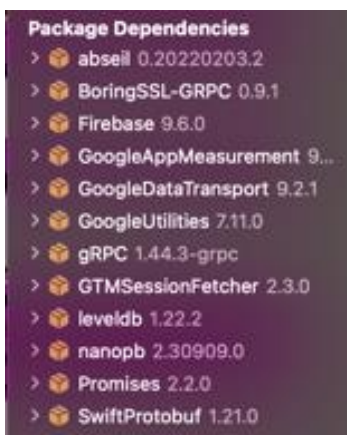
I created a new project to indicate the type of app I was developing and select a template for either iOS or macOS. I picked the iOS App template, which equipped me with a basic structure for my app. The view controller and storyboard template provided within enabled me to quickly create the user interface. I streamlined my workflow by adding the colour scheme and intended app images into the assets folder.

I then set up the app's user interface using Storyboard. Buttons, text fields, and labels were among the assortment of components added to the interface; all of which were connected to the view controller. Create visually appealing and interactive user interfaces with ease using Storyboard.

Finally, I set up the project's build settings and run configurations. This included selecting the simulator device, setting up build options, and configuring the debugger. Xcode offered me every necessary tool to develop a thriving application in its comprehensive development environment.

4.6 Database Implementation

4.6.1 Firebase



The choice for my app's database backend system was Firebase. Firebase was the ideal choice for my app's database backend due to its cloud-hosted NoSQL database, which permitted real-time data storage and synchronization.

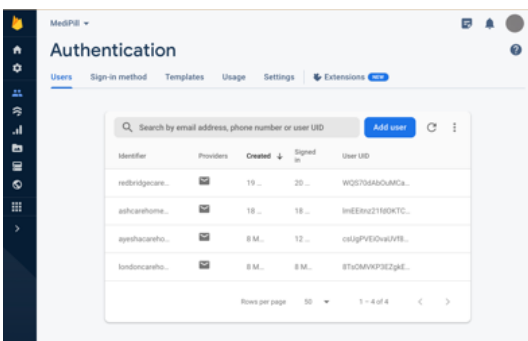
The database implementation process was commenced by opening a Firebase account and setting up a new project. I directly accessed the database from my app by installing the Firebase SDK into my Xcode project after the setup.

I developed a database architecture that matched my app's requirements. With Firebase's web console, I was able to generate a database and configure the right data framework. I created collections and documents to hold my app's data that includes user information, preferences, and settings.

Once the database structure was in place, I then implemented the Firebase SDK within my app's code. Firebase's APIs facilitated the reading and writing of data to the database. Firebase's API is well-documented and simple to use, making database interactions a cinch.

At last, I implemented Firebase authentication into my app, which provided users with secure account creation and login. By implementing rules, the data was secure from unauthorized access or modification, exclusively available to authorized users.

4.6.2 Sign Up Feature



I utilized Firebase to add a registration feature, which was one of the most notable additions I made. Upon completing the integration of Firebase Authentication into my application and setting up an associated Firebase project, I constructed a page for user registration with fields requesting the name of the Care home, work email address, password, and name of Care home Administrator.

I then employed Firebase's API to craft a fresh user account for Firebase Authentication system. Using Firebase API enabled storing user credentials securely in Firebase Authentication, creating a reliable authentication system for the app.

The Firebase Authentication system was modified to include rules that restricted access to the app's features exclusively to authorized users. I established restrictions for accessing specific features and data to ensure user privacy and security.

Overall, I found the Firebase sign-up feature implementation process to be straightforward and efficient. I utilized Firebase Authentication to construct a dependable and secure authentication mechanism that satisfied my application's requirements. Firebase's API made integrating email verification and password reset features seamless.

4.6.3 Login Feature

After setting up the signup feature, it only made sense to then work on the login feature for my app. The process seemed relatively straightforward again.

There is now a login button situated on my app's home screen. After users press the login button, they are authorized to sign in utilizing their email and password on the Firebase authentication screen.

4.6.4 Notification Implementation

The last feature using Firebase that I implemented was the notification feature for my app. I think that Firebase is an excellent choice for adding push notifications because it offers a sturdy and straightforward platform. Integrating Firebase Cloud Messaging (FCM) into my app was the initial step I took.

Dispatching notifications to my app's users was achieved through message crafting and delivery using the Firebase console. Creating serverless functions with Firebase Cloud Functions allowed me to trigger them by events such as user sign-ins or database changes. I was able to generate more intricate notifications that could activate based on user behaviours.

4.7 Testing

The development process includes testing, which is a crucial part. To ensure that my app was functioning as intended and meeting user needs, I made sure to incorporate regular testing. I depended on those around me to test my app and give feedback to conduct testing. Friends, family, and colleagues who were willing to use the app and provide honest feedback were included.

Using surveys, I gathered user feedback on their app experience, including any issues or suggestions for improvement, as well. After administering physical assessments and closely scrutinizing the actions of app users, I gained a deeper understanding of their needs and choices. Through consistent user testing and soliciting for feedback, I managed to enhance and fine-tune my application with the purpose of satisfactorily meeting the demands of my intended consumer demographic.

5 Results

5.1 Software Development Methodology

5.1.1 Agile Development

My choice for project methodology was Agile development due to its adaptability and flexibility. The Iterative approach by Agile allowed for quick feedback, which was important in delivering a product meeting user requirement. Agile development proved a highly efficient approach for this project, and I would consider implementing it in future development processes.

5.1.2 Steps Taken

Smaller, more manageable tasks were used to break down the project into specific sprints as part of implementing the Agile methodology. After that, I drafted a record of assignments that necessitated attention, coupled with estimated time frames for each. The project was effectively managed, and a high-quality prototype was delivered by utilizing Agile development.

5.2 System Requirements

You can find the results for the below System requirements in the appendix.

5.2.1 System Requirements Gathering – **APPENDIX C**

5.2.2 Initial Domain Model - CRC Cards – **APPENDIX D**

5.2.3 Use case Index – Iteration - **APPENDIX E**

5.2.4 Usage Model - Use case Specification – **APPENDIX F**

5.2.5 Entity relationship diagram – **APPENDIX G**

5.3 Design Documentation

5.3.1 App Logo

Refer to the App Logo Survey Results – **Appendix H**

To design an effective logo for my healthcare app, I conducted a survey for my participants to gather feedback on the logo's design elements. The survey results showed that respondents found the green and blue logo to be modern, up to date, and credible, and that the heart shape in the logo could be improved to better reflect the purpose of the app. Additionally, the logo was deemed gender-neutral and memorable.

Based on the survey results, I used Canva to create and edit the logo. Incorporating the survey feedback was essential in refining the logo design to accurately convey the purpose of the app and create a professional, trustworthy image that aligned with the healthcare industry.

5.3.2 User Interface Model - User interface & User Experience (UI & UX)

Refer to the UI Development Survey Results – **Appendix H**

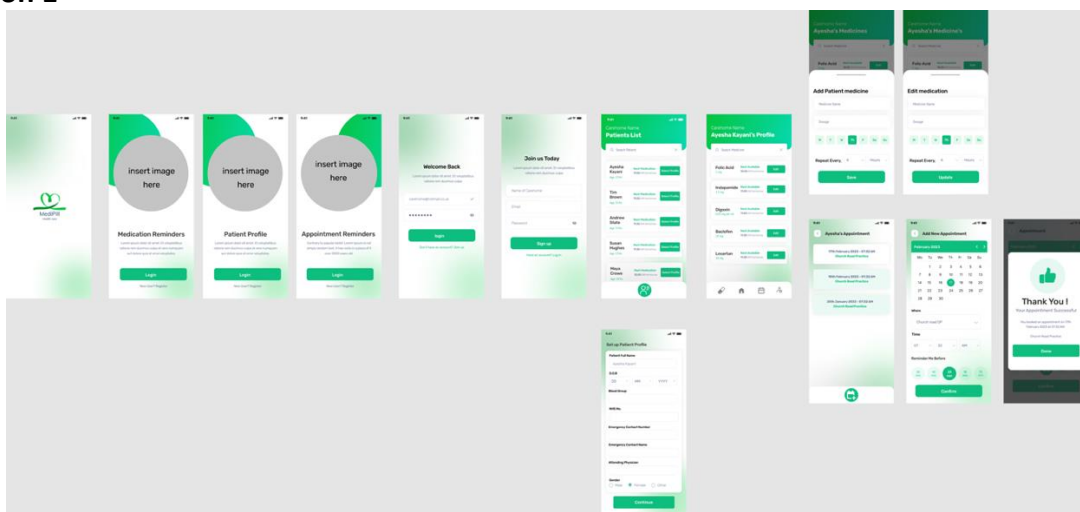
I utilized Figma for creating and modifying the user interface (UI) and user experience (UX) design during the development and design of the MediPill application. Users expressed their appreciation for the simplicity of use and the availability of buttons and icons, according to survey results.

I incorporated some changes to the UI design, such as adding a notification page and including patient details on the patient profile page, based on the feedback received. I removed unnecessary screens, such as the thank you screen that appears after adding an appointment, as well. On the screen that is designated for logging in or signing up, I have recently adjusted allow users to opt between an option of a login button and a sign-up button. Additionally, I included a delete medication button to make it easier for users to manage their medication.

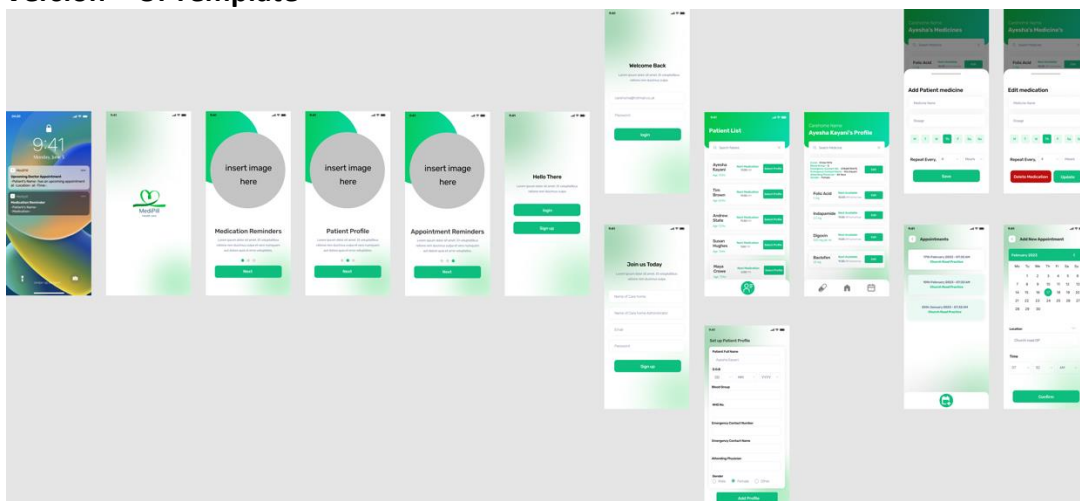
While some users suggested adding more colours, most of them appreciated the consistent colour scheme and clean layout. Overall, the changes made to the UI/UX design improved the user experience and made the app more accessible and easier to use.

Below you can see the changes I have made to the UI/UX design using the feedback from the surveys.

Version 1

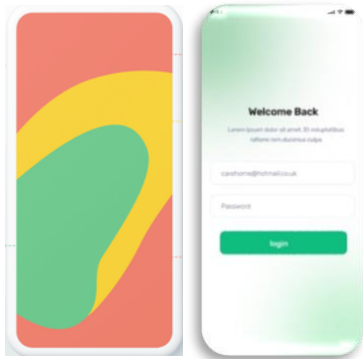


Final Version – UI Template



5.3.3 Usability

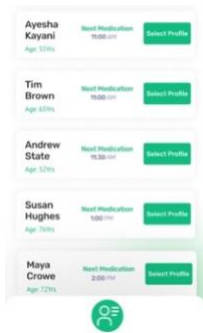
I focused on creating a mobile app with good usability. I took into consideration the thumb-friendly heat map analysis of an iPhone screen, which divides the screen into three areas based on ease of access - Natural, Stretch, and Hard.



I placed the most important features of the app in the Natural area to ensure they are easily accessible. I also minimized the use of Stretch and Hard areas, which can be uncomfortable for users to reach with their thumbs.

In the left image, comfort for the user was considered with important features carefully situated in the yellow and green zone. My iPhone app is both comfortable and easy to use, thanks to the thumb-friendly heat map analysis it underwent.

5.3.4 Accessibility



I emphasized the critical importance of accessibility when developing my IOS app. Following international standards like the WTC web accessibility initiative is vital to make the app accessible to users with disabilities.

You can observe an illustration of the bold text and large buttons/labels used in my app on the left. Through the examination of these attributes, I can broaden my app's audience and enhance its accessibility and usability. The application can appeal to a wider group of users and enhance the overall user experience with this approach.

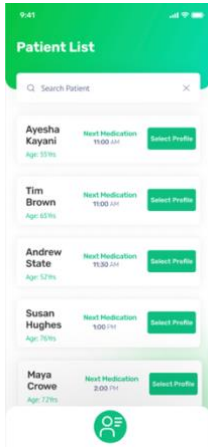
5.3.5 IOS Design Guidelines

I closely followed the IOS Design Guidelines from Apple when building my app. Typography, colour schemes, and layout are among the design aspects suggested in these guidelines to ensure the app complements the IOS operating system's aesthetic.

The IOS Design Guidelines emphasize a coherent visual language throughout the app. The app's consistency in colour and font contributed to the seamless integration of every interface element.

Following IOS Design Guidelines closely yields an app that boasts visual excellence, user-friendliness, and consistency to the IOS operating system. The guidelines played a crucial role in creating a visually pleasing and user-friendly app.

5.3.6 Use of IOS Native Components – Navigation Bar



I optimized user experience and usability by using the navigation bar and other iOS-specific interface elements in accordance with the iOS design guidelines.

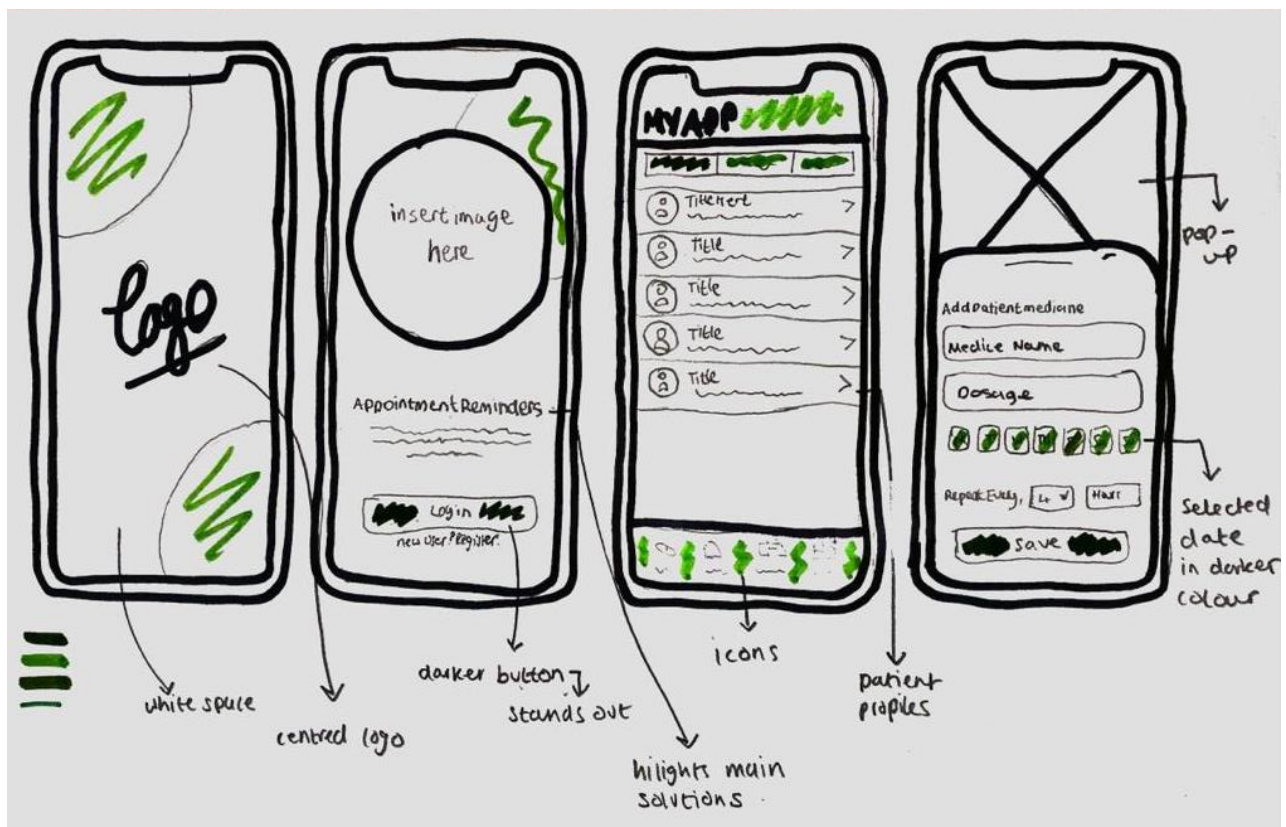
The bottom placement of the navigation bar is a defining characteristic of iOS native. Observe the “add patient button” on the left which directs users to the “create patient profile” page at the bottom of the screen.

Incorporating iOS native features in my app enabled it to be user-friendly and meet the expectations of iOS users. Among the various aspects I considered to improve the app's usability, positioning the navigation bar at the bottom was one of them.

5.4 Prototype Development

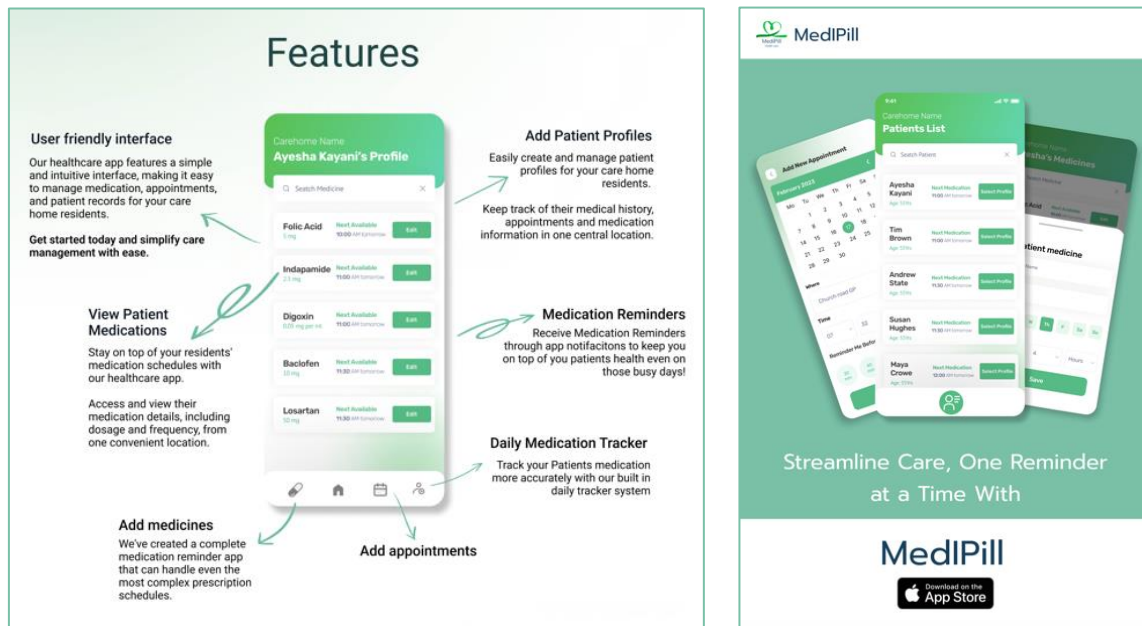
5.4.1 Hand-drawn prototype.

Below, you can see the very first draft of the UI development. As explained in the methodology, I focused on the main concepts that I wanted within my app and made sure to highlight them so I can focus on them further in the development process.



5.4.2 Mock-ups

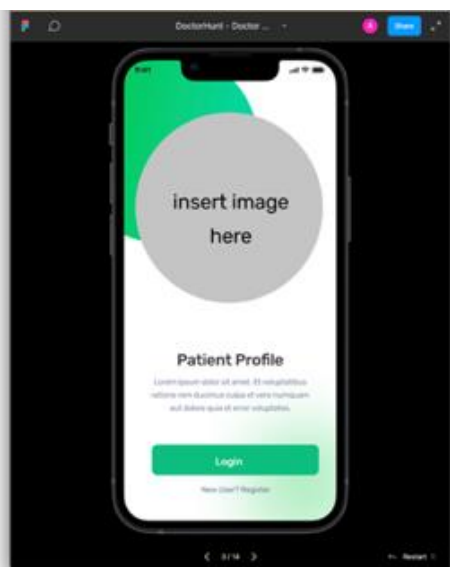
I opted to switch to Figma to create a polished and professional design for my app, departing from the hand-drawn prototype. Sharing my mock-ups and getting feedback for improvements was made possible with Figma.



Promoting the app's unique features was done through visually effective banners which I created in Figma alongside mock-ups for UI design. You could utilize these banners on promotional materials to boost exposure and engage fresh audiences. Figma enabled me to produce visually stunning banners that reinforced the app's branding.

Overall, creating mock-ups using Figma was a crucial step in the app development process. Refining the design through Figma made sure every feature and screen had a consistent look. I created a polished and professional design by testing various design concepts and collecting feedback from others.

5.4.3 Interactive Prototype



After designing the basic UI of my app using Figma, I moved on to creating an interactive prototype. Ensuring a user-friendly design necessitated testing my ideas during this step. The process involved creating clickable components and linking them together to simulate the user flow. This allowed me to identify any usability issues and inconsistencies in the design and make necessary changes.

Creating an interactive prototype helped refine and polish the design while saving valuable time and resources.

5.5 Implementation

5.5.1 Development Environment

Xcode was my chosen tool for creating the front-end systems. Designing and developing my app's interface was made easy with this user-friendly software. With XCode's integrated development environment, I was able to construct, experiment, and perfect my app's functionality minus the need for any additional tools.

To handle the app's backend systems, I chose to use Firebase. Opting for Firebase means benefiting from its user authentication, real-time database, and cloud storage features that are exceptional. Firebase integration made managing patient data, storing files securely, and accessing data in real-time possible within my app. Firebase's comprehensive analytics and crash reporting tools help identify and resolve any potential issues effectively.

I began by incorporating the Firebase SDK into my Xcode project. The Firebase SDK was integrated into my Xcode project, and the authentication system was established to enable user registration, login, and account management. The real-time database was then built to manage data storage and retrieval. The Firebase console made it effortless to configure my database.

My use of Xcode and Firebase enabled me to produce reliable software with an efficient interface and potent backend capabilities. The integration of Xcode and Firebase was essential in helping me design and launch my app seamlessly.

5.5.2 Programming Languages

In developing interactive user interfaces, my study identified Swift and UIKit as the primary programming languages. The modern and powerful features of Swift allowed me to create concise and maintainable code. Also, utilizing UIKit's various pre-existing elements and assets allowed me to swiftly develop complex interfaces.

To visually organize the UI elements such as buttons, text boxes, and labels in my app, I employed Storyboard. I could make quick UX improvements by previewing my app's appearance and function with Storyboard.

5.5.3 UI Design - Version Control

I discovered that using Notion to store different versions of the UI design was an effective method for maintaining version control and monitoring the project's progress.

Creating a unique Notion page for UI development helped me compare design versions. Additionally, embedding the Figma link into the dedicated Notion page made it easier for me to access and modify the design as needed. Overall, this technique facilitated my organization and adherence to the project's goals.

5.5.4 Xcode Set Up Process

Setting up Xcode was an easy process since I already had Xcode downloaded on my laptop; all I had to do was create a new project using an iOS template. The iOS App template provided a basic structure for my app, and the view controller and storyboard template enabled me to quickly create the user interface. I streamlined my workflow by adding the colour scheme and intended app images into the assets folder.

Then, I utilized Storyboard to establish the app's UI that contained text fields, buttons, and labels all linked with the view controller. Storyboard made it easy to create visually appealing and interactive user interfaces.

Afterward, I established the project's build settings and run configurations, chose the simulator device, and tailored the debugging options. I found that Xcode offered all the necessary tools to develop a thriving application in its comprehensive development environment. XCode's setup process proved to be critical in developing my app.

5.6 Database Implementation

5.6.1 Firebase

Firebase was my preferred option for my app's database backend system due to its cloud-hosted NoSQL database, enabling real-time data storage and syncing. The creation of a Firebase account and setting up a new project initiated the database implementation process. The Firebase SDK was installed into my Xcode project to enable me to access the database directly.

To meet my app's requirements, I developed a database architecture that matched it. Using Firebase's web console, I generated a database and configured the right data framework. I created collections and documents to hold my app's data that included user information, preferences, and settings.

After setting up the database structure, I implemented the Firebase SDK within my app's code. The data in the database was both read and written through Firebase's APIs. Firebase's API documentation was thorough and easy to navigate, which made working with the database smooth sailing.

Firebase authentication was successfully incorporated into my app to guarantee secure account creation and login. To ensure that only authorized users could access or modify the data, I established regulations. Overall, Firebase provided an excellent database solution for my app, and the implementation process was straightforward.

5.6.2 Firebase Authentication

In my app, I utilized Firebase Authentication as the primary mechanism for securing user accounts. I found the process of integrating Firebase Authentication to be simple and efficient. Firebase Authentication provided my app with reliable security features. Firebase Authentication provided an excellent tool for secure and reliable authentication in my app. I enforced regulations that limited access to authorized users only to boost the authentication system's security. I also implemented email verification and password reset features through Firebase's API, which further strengthened the app's security. The Firebase Authentication tool is one I found to be excellent for implementing secure and reliable authentication in my app.

5.6.3 Sign Up Feature

The registration feature in my app was efficiently implemented through Firebase. Creating an associated Firebase project and integrating Firebase Authentication was my foremost step. A registration page was crafted with slots dedicated to the care home name, password, work email address, and the care home administrator's name.

Then, I employed Firebase's API to establish a fresh user account for Firebase Authentication. Storing user credentials securely in Firebase Authentication provided a reliable authentication system for the app. By incorporating rules that limit access to authorized users only, I adjusted the Firebase Authentication to safeguard user privacy and security.

The Firebase sign-up feature was implemented with ease and efficiency. The Firebase API made integrating email verification and password reset features a breeze. The Firebase Authentication mechanism I developed was dependable and secure, fulfilling my application's requirements.

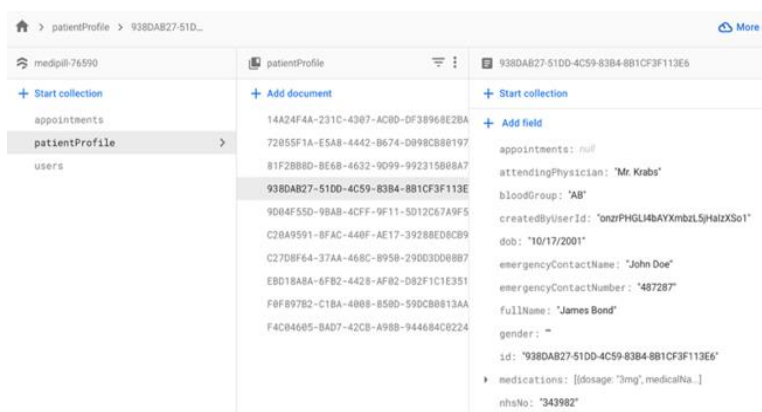
5.6.4 Login Feature

My attention shifted to the login feature once I had set up the signup on my app. The process seemed relatively straightforward as I utilized Firebase Authentication, which I had previously integrated into my app.

The ability for users to log in was facilitated by the login button located on the home screen of my app. Users are led to a Firebase authentication page for credentials verification once they've clicked the login button. This was made possible through Firebase's API, which enabled the verification of user credentials stored securely in Firebase Authentication.

Overall, the implementation of the login feature was a seamless process, thanks to the Firebase Authentication system. My app's authentication mechanism was made secure and accessible to only authorized user, all thanks to Firebase's API.

5.6.5 Set Up Patient Profile Feature



The next feature I implemented for my app using firebase as the backend was the set-up patient profile feature. By entering personal information, medical history, and other relevant details, care home professionals can create patient profiles.

I began by making a "patientProfiles" Firebase

collection to save patient profiles. The Firebase platform generates a distinct ID for each profile. New profiles require patients to input their name, date of birth, and basic information, per app prompt. This data is then stored in the "patientProfiles" collection, associated with the user's unique ID.

Throughout the process, I tested the setup patient profile screen thoroughly to ensure that it functioned correctly and efficiently. After completing the tutorial, I made further adjustments to the patient profile screen to suit the specific needs of my app.

I also added additional fields to capture more detailed medical information, such as Blood Group. Ensuring confidentiality, Firebase implements secure storage and encryption for all patient medical and personal information.

Using Firebase allows for real-time updates and syncing across all devices with the Set-Up Patient Profile feature. A seamless user experience is achieved as any profile information update made by a staff member on one device is immediately reflected on all other devices.

Overall, I am proud of the implementation of the Set-Up Patient Profile feature using Firebase. Patients can confidently manage their medical information with this secure and efficient profile creation tool.

5.6.6 View Patient Profile Feature

I then implemented a View Patient Profile feature for my app, which allows caregivers to view and access patient profiles that were previously created. A caregiver can effortlessly retrieve the patient's medication and relevant information using Firebase once they have set up the patient profile.

A patient can be found by caregivers via a search function using their name. The caregiver can access the patient's medical information after viewing their profile.

Firebase enables secure and dependable access to patient details for caregivers via the View Patient Profile feature without compromising confidentiality of personal and medical information. Furthermore, Firebase's real-time notifications guarantee that medical professionals can retrieve current patient data, a vital component during emergency situations.

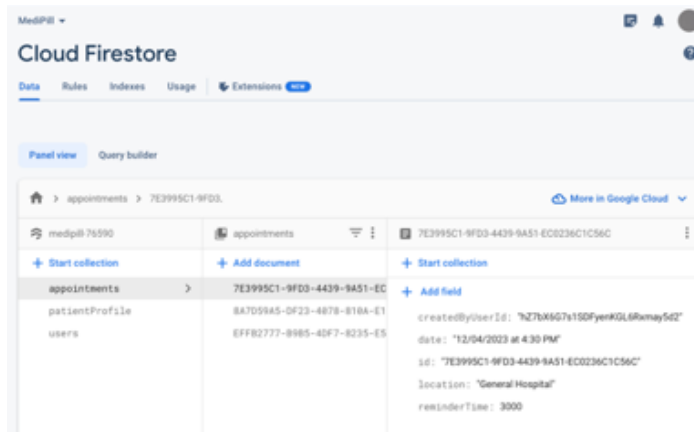
Overall, I am pleased with the implementation of the View Patient Profile feature using Firebase.

5.6.7 Add / Edit / Delete Patient Medication Feature

To create the add/edit/delete patient medication feature on Xcode, I followed a systematic approach. Firstly, I designed the user interface for the medication feature and created the necessary view controllers to manage user input. Afterwards, I implemented Firebase's real-time database to store and retrieve medication information for each patient. To ensure that only authorized users can access the medication feature, I used Firebase Authentication.

Finally, I coded the functionality to add, edit, and delete medication information, making sure to update the Firebase database accurately. My testing procedures were thorough to guarantee the feature functioned accurately and efficiently throughout the process.

5.6.8 Add / View Appointment Feature



Another feature I implemented using Firebase as the backend was the Add Appointment feature. This feature enables caregivers to schedule appointments for patients, guaranteeing that patients get timely care.

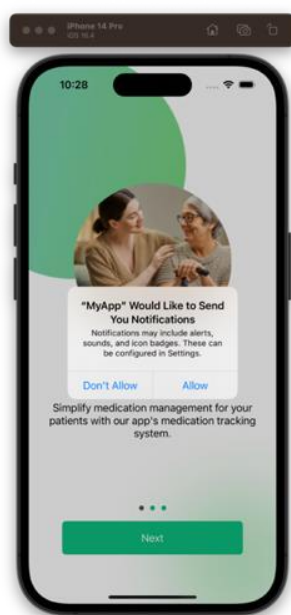
I created a new Firebase collection named “appointments” specifically for storing appointment information. Each appointment is assigned a

unique ID generated by Firebase. Caregivers must enter appointment details like time, date, and hospital location when scheduling a new appointment. The “appointments” collection stores this data and links it to the appointment's distinctive ID.

Firebase provides secure and encrypted appointment data storage to guarantee patient medical information privacy and security. Firebase's real-time updates provide healthcare providers with critical appointment information updates which can prove crucial in emergencies. To enable caregivers to view appointment information, I created an appointment list page in the app that displays all upcoming appointments.

Overall, I am pleased with the implementation of the Add / View Appointment feature using Firebase. Patient privacy is protected while healthcare providers improve their care through the secure and efficient scheduling system provided. Access and management of appointment information are easier with real-time updates and improved search functions, improving healthcare providers' user experience.

5.6.9 Notification Feature



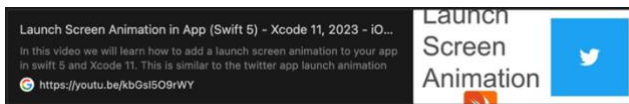
For the easy integration and guaranteed reliability it provides, I selected Firebase Cloud Messaging (FCM) to add push notifications to my app. After integrating FCM, my app became capable of dispatching push notifications to its users.

Using the Firebase console, I created notifications by crafting messages and delivering them to users. I was able to create more complex notifications triggered by events such as user sign-ins or database changes by leveraging Firebase Cloud Functions to create serverless functions.

Overall, I found that implementing the notification feature using Firebase was a straightforward process. Crafting and delivering notifications to my app's users was made easy with Firebase's robust tools, and the ability to trigger notifications based on user behavior was especially helpful.

5.7 Non-Database Implementation

5.7.1 Launch Screen

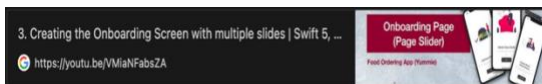


To create the launch screen on Xcode, I followed a tutorial on YouTube by iOS Academy. The tutorial detailed how to

create a launch screen using the storyboard in Xcode.

The project's launch screen was established by creating and selecting a new file. Then, I included a picture element in the start screen and modified its dimensions and location. I also added constraints to ensure the image view would display correctly on different screen sizes. To give a professional touch, the launch screen was customized by inserting the app logo at the centre. The launch screen was tested by me post-tutorial to ensure proper display. (Academy, 2020)

5.7.2 Onboarding Screen

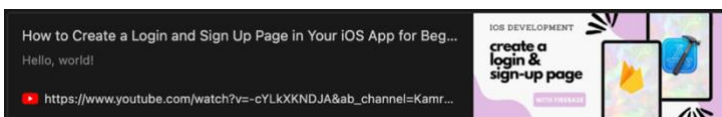


The onboarding screens for my app were made using the YouTube tutorial by Emmanuel Okwara.

By using bold graphics and straightforward language, I aimed to make the app's onboarding procedure enticing and engaging. I also made sure to consider the overall user flow and ensure that the onboarding screens seamlessly transitioned into the app's main interface. The onboarding screens were crucial in enhancing my app's user experience by successfully introducing its features to new users.

(Okwara, 2021)

5.7.3 Log in/ Sign up Screen



To create the login/signup screens for my app, I started by researching best practices and common design patterns for authentication

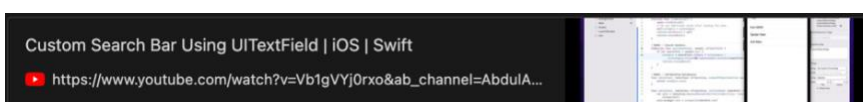
experiences. The YouTube video below was my guide in creating the login/signup screens in Xcode. The tutorial explained how to create screens, which involves UI design, user input validation, and Firebase Authentication for user authentication and account creation.

The seamless and user-friendly authentication experience was achieved by designing an interface that was visually appealing and easy to navigate. User input validation was integrated to guarantee a smooth login/signup process.

In addition, I employed Firebase Authentication to handle secure and reliable user authentication and account creation. My app's login/signup screens were pivotal in maintaining both user account security and ease of use.

(Ohly, 2022)

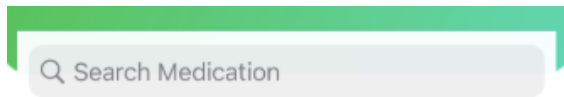
5.7.4 Search Patient /Medicine Feature



To create the search patient/medicine feature for my app I used the

YouTube tutorial by Abdul Azeem to guide me through the process of implementing the

feature in Xcode. The video tackled the necessary steps in creating a search bar interface, connecting it to the database, and filtering the outcomes as per user input. I centred my efforts on designing a user-friendly interface that allowed for simple input and fast retrieval of search results.
(Azeem, 2022)



To begin with, I created a UISearchBar and set its delegate to my PatListViewController instance. Whenever the user enters text in the search bar,

the search bar (_:textDidChange:) method is called automatically, and the patient data is filtered and updated accordingly.

The first step I took to handle the search bar functionality was to include an extension for my PatListViewController class that handles the search bar functionality in my app. I implemented the UISearchBarDelegate protocol, which provides methods for handling events related to a search bar.

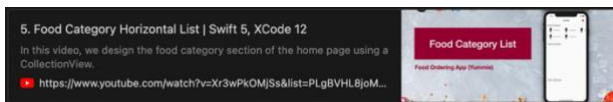
The search bar (...textDidChange...) method is called whenever the text in the search bar changes. Next, I used a guard statement to check if the text is empty. If the search bar is empty, I set the isSearching flag to false, reset the filteredPatients array to an empty array, and reload the patientListCollectionView with the original patient data.

If the search bar is not empty, I set the isSearching flag to true and used the filter() method to populate the filteredPatients array with patient data that contains the search text. The closure used inside the filter() method checks if the lowercase name property of each patient object contains the lowercase search text. Finally, I reload the patientListCollectionView with the filtered patient data.

```
// I used this link to help me write the code for the search bar below
// https://www.youtube.com/watch?v=Vb1gVYj0rxo&ab_channel=AbdulAzeem
extension PatientViewController: UISearchBarDelegate {
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        let text = searchText.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !text.isEmpty else {
            // If the search bar is empty, reset the filtered data and reload the collection view
            isSearching = false
            filteredMeds = []
            medicinesCollectionView.reloadData()
            return
        }
        // Filter the patient data by name and reload the collection view with the filtered data
    }
}
```

Overall, I am very happy with the Search Patient/Medicine feature implementation. It has improved the usability of the app significantly, allowing users to quickly find the information they need. I believe that this feature will be particularly useful for healthcare professionals who need to quickly access patient or medicine data in a busy clinical environment.

5.7.5 Patient List Screen



The patient list screen for my app was designed by first identifying the crucial features and functionalities needed to present the list of patients. Using the video provided as a guide, I leveraged Xcode and Swift to build the necessary views, controllers, and models required for the patient list screen.

Firestore was also utilized to ensure patient data storage and list accuracy. I utilized the UICollectionView to generate a patient roster that allows for swift searching, sorting, and selection by the user.
(Okwara, 2021)

5.7.6 Interactive Calendar Feature



I also implemented an Interactive Calendar feature for my app using the provided YouTube tutorial. This feature was designed to allow users to easily schedule and manage appointments or events within the app.

Overall, I am very pleased with the Interactive Calendar feature implementation. It has significantly improved the functionality of the app, allowing users to easily schedule and manage appointments or events. I believe that this feature will be useful for care home staff as they need to constantly keep track of multiple medications and appointments within a given period.
(Academy, 2022)

5.8 Testing

Refer to the App Testing Survey Results – Appendix H

In the process of software development, testing plays a pivotal role that cannot be underestimated. I implemented a regular testing schedule to ensure that my app was functioning correctly and meeting the needs of users.

To solicit sincere feedback on the application, I relied upon individuals within my social network such as friends and co-workers. To extract opinions from users regarding their interaction with the application, comprising of any complications they faced and recommendations for enhancement, a supplement to my approach included surveys. Furthermore, to acquire more insights into potential areas that require development or modifications within the app's functional architecture, user feedback channels become indispensable components of this process.

One downfall I faced during the testing stage was that I sent out three surveys to my participants and only received feedback on the first two. I was reluctant enough to receive a solid number of responses for the first two, but the number of participants for the third Survey drastically dropped. Ultimately, I successfully enhanced the application and implemented essential modifications to fulfil the requirements of my targeted audience using the feedback from the surveys.

6 Conclusions

6.1 Project Objectives Review

The totality of the primary and secondary aims for this project were effectively accomplished by me, warranting contentment regarding significant performance. To improve patient outcomes and streamline the medication management process for care home staff, I developed a successful iOS healthcare app that included patient records, pill and appointment reminders, and appointment scheduling.

The app also permitted staff to retrieve patient records instantly, empowering them to make informed decisions regarding patient care. To guarantee that residents took their medication and attended appointments on time, the app also offered customizable reminders and notifications. Ensuring that the app allows for easy data entry and record-keeping was crucial for efficient care management, and I was able to achieve this. The imminent possibility of this application revolutionizing patient care in residential healthcare facilities is a source of immense excitement for me. Moreover, I am exceedingly gratified to have successfully achieved all the predetermined goals established by this undertaking.

6.2 Conclusions drawn from the project.

In conclusion, I gained a lot of knowledge from this project in various ways. Not only did I successfully create the app, but I also gained valuable experience in coding, UI/UX design, and project management.

I realized the significance of collecting user feedback and testing throughout the development process, as this input was crucial in refining the app to meet user needs. Developing my skills in Firebase and coding in Swift will be valuable for future projects. Additionally, it will be beneficial. This project emphasized the importance of efficient project management, attention to detail, and dedication to the development process. I am confident that the skills and experience gained through this project will serve me well in future software development endeavours.

6.3 Implications for future work

There are several implications for future work based on my experience developing this app. One major implication is the importance of user feedback and testing throughout the development process. Going forward, I will place an even greater emphasis on gathering feedback from users to ensure that my apps meet their needs and are user-friendly.

Also, one of my objectives is to advance further in coding and UI/UX designing skills so that subsequently I can create enhanced and more pragmatic applications, ultimately realizing the significance of being up to date with emerging technologies and changes in industries. Therefore, to keep my work relevant and effective, I will consciously try towards staying informed on such developments, and developing this app has taught me valuable lessons that will assist me with future software development.

6.4 Personal Progress

Concerning my own development and progress during the assigned time frame I feel satisfied with what was achieved; however, success came my way when I developed the app effectively while grappling with using Firebase and coding in Swift at once.

Additionally, I have improved my UI/UX capabilities that are going to assist me in future work, but despite this, I accept that there are regularly spaces for improvement. In future assignments, I aspire to produce better designs for user-friendly interfaces and improve upon productivity by increasing coding efficiency. Overall, I feel pleased with the progress that has been made thus far and look forward to developing more of my abilities in future projects.

7 References

Kshitij (2022). *Case study: A Medication Reminder App*. [online] Medium. Available at: <https://uxplanet.org/remedi-a-medication-reminder-app-ui-ux-case-study-8de1fe494f27> [Accessed 2 Apr. 2023].

Dribbble. (n.d.). *Thumb-Friendly Heat Map Analysis*. [online] Available at: <https://dribbble.com/shots/9756551-Thumb-Friendly-Heat-Map-Analysis> [Accessed 2 Apr. 2023].

Ideas. (n.d.). *iOS App Design Guidelines & Inspiration | Adobe XD Ideas*. [online] Available at: <https://xd.adobe.com/ideas/principles/app-design/ios-app-design/#:~:text=Design%20principles> [Accessed 2 Apr. 2023].

Ivo Mynttinen / User Interface Designer. (n.d.). *The iOS Design Guidelines*. [online] Available at: <https://ivomynttinen.com/blog/ios-design-guidelines>.

Ideas. (n.d.). *Essential Patterns of Mobile Navigation | Adobe XD Ideas*. [online] Available at: <https://xd.adobe.com/ideas/principles/app-design/essential-patterns-mobile-navigation/>.

www.youtube.com. (n.d.). *How to Create a Login and Sign Up Page in Your iOS App for Beginners*. [online] Available at: https://www.youtube.com/watch?v=-cYLkXKNDJA&ab_channel=KamrynOhly [Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *Launch Screen Animation in App (Swift 5) - Xcode 11, 2023 - iOS Development*. [online] Available at: <https://youtu.be/kbGsl5O9rWY> [Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *3. Creating the Onboarding Screen with multiple slides | Swift 5, XCode 12*. [online] Available at: https://www.youtube.com/watch?v=VMiaNFabsZA&feature=youtu.be&ab_channel=EmmanuelOkwara [Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *5. Food Category Horizontal List | Swift 5, XCode 12*. [online] Available at: https://www.youtube.com/watch?v=Xr3wPkOMjSs&list=PLgBVHL8joMCthoqCwsWAUWIiyJaPrqPA4&index=5&ab_channel=EmmanuelOkwara [Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *Custom Search Bar Using UITextField | iOS | Swift*. [online] Available at:
https://www.youtube.com/watch?v=Vb1gVYj0rxo&ab_channel=AbdulAzeem
[Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *UICollectionView Introduction (Xcode 14, iOS 16) - WWDC 2022*. [online] Available at:
https://www.youtube.com/watch?v=B_VFHeg2LH4&ab_channel=iOSAcademy
[Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *How To Pass Data through Segues in Xcode with Swift*. [online] Available at:
https://www.youtube.com/watch?v=LOnajTZ03dM&ab_channel=KamrynOhly
[Accessed 2 Apr. 2023].

www.youtube.com. (n.d.). *Swift 5: Firebase Database in App - Setup/Read/Write Data (Xcode 11) - Beginners*. [online] Available at:
https://www.youtube.com/watch?v=tpsffoRh9u0&ab_channel=iOSAcademy
[Accessed 2 Apr. 2023].

8 Appendix

(APPENDIX A) – PDD

Cover sheet

The title of your degree programme:

BSc Computer Science

A short, informative project title:

MediPill – An Application for Care home workers assisting with patient records and Pill/
Appointment reminders

Your name and email address:

Ayesha Kayani - Ayesha.Kayani@city.ac.uk

The name of your consultant and name(s) for any client(s):

Consultant - Christina Gaek

Who proposed the project (you, your client or someone else):

I proposed the project

A brief outline of all arrangements for proprietary interests in your project work and/or outside help. Attach documentation of agreements. If there are no such arrangements, this must be clearly stated:

No Agreements

Any other promises you are making to secure acceptance of the project (e.g., by your consultant or project team member):

No Promises

Project Proposal

Problem to be solved.

Initially, I intended to create a pill reminder app as this was something I could relate to. Living with elderly members I would see how much medication they must regularly take, but often forget to or couldn't remember if they already took their pills that day or not.

After doing some research online and from the knowledge I gained from my internship, I was made aware of the insufficient use of technology within care homes. Having previously worked on a project for a well-known UK Healthcare Provider, my main role was to help with the integrations between their Hospitals and Care Homes. Although the company is a global competitor, you'd assume that they would have these basic integrations already in place. Seeing this was not the case I decided to incorporate this element into my app.

The problem I am aiming to address is the insufficient use of technology within care homes, resulting in difficulty in managing patient medication reminders, appointment schedules, and medical records. The solution I am proposing is a single app for Care Home workers incorporating these features to streamline and improve their business operations.

Although I am unable to share the work that I have previously worked on concerning this topic due to confidentiality, I can provide the details of my manager who will be more than happy to provide you with a basic report of the project.

Project Objectives (and possibly research questions)

This project shall:

- 1. Incorporate pill reminders, patient appointment reminders, and patient records into a single app.**
 - a. This will potentially enhance the technology used within care homes to improve the ease and efficiency of patient care.

Project Beneficiaries

This product is a prototype; however, if it was to be deployed for a customer the beneficiaries would likely be:

- 1. Care home residents.**
- 2. Care home staff, including care workers and medical professionals.**
- 3. Care home managers and administrators.**

These beneficiaries would benefit from the improved technology and increased efficiency in managing patient care, leading to improved patient outcomes and a higher quality of life for care home residents.

Work Plan

Gathering Requirements :

- Conduct research and gather requirements for care home staff from the internet; to maintain a comprehensive list of requirements for the app.
- **Resources:** Internet

Design :

- Design the UI of the app user interface design tools such as Figma to meet the requirements gathered.
- Show the proposed UI to students/healthcare professionals to retain feedback and make any changes if necessary.
- **Resources:** Figma, Google Forms

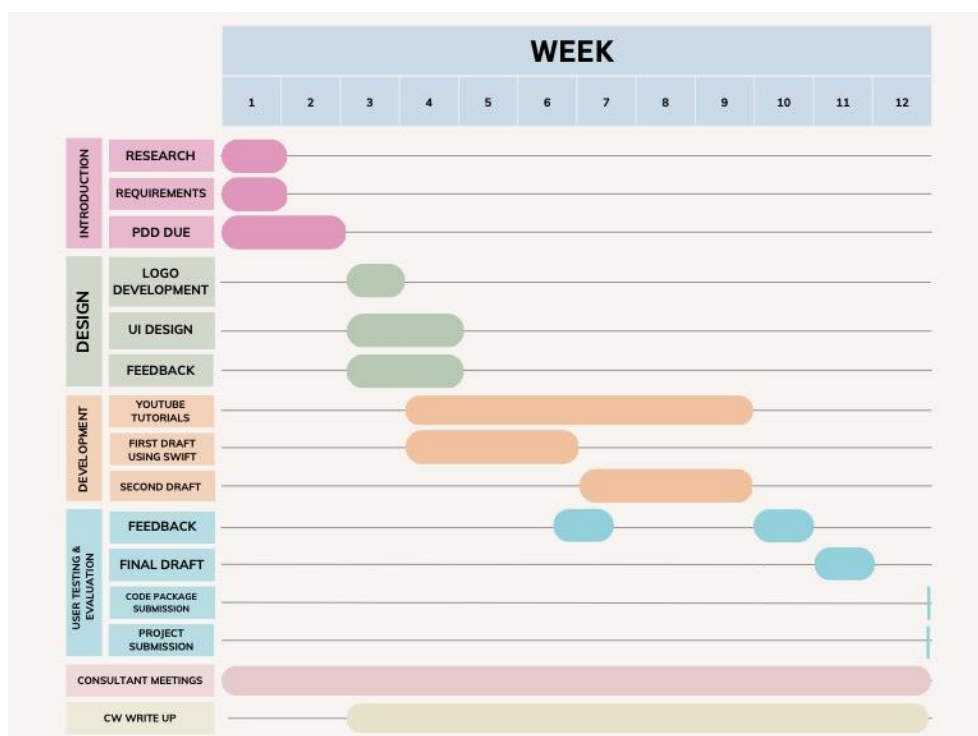
Development :

- Develop a working prototype of the app in Swift using XCode to meet the requirements gathered.
- **Resources:** XCode, YouTube tutorials and development tools such as 'Playgrounds'.

User Testing and Evaluation :

There are various methods and techniques for testing an application. Ideally, testing should be an ongoing process throughout the development of my application to ensure that any problems or issues are identified and addressed promptly.

- Obtain feedback from students/healthcare professionals to improve the app. This will ensure a refined version of the app with improvements based on user feedback.
- Evaluate the app to identify areas for improvement and provide ongoing maintenance to ensure an up-to-date and well-functioning app.
- **Resources:** User feedback forms (Google Forms)



Risks – (1) Risks to your project

Below is a list of potential risks relating specifically to my project:

4. **User acceptance:** From my research earlier, I gathered that there may be resistance from care home staff to adopt new technology, affecting the success of the app. **To mitigate this, I will regularly gather feedback from those around me to understand their experiences with the application and identify areas for improvement.**
5. **Data privacy and security:** The app will be handling sensitive patient information, and it's important to ensure the protection of this data. **To mitigate this, I will be using a dummy dataset with fake patient details.**
6. **Integration with existing systems:** The app may need to integrate with existing systems used in care homes, leading to potential compatibility issues. **Since this is a student project, I will only be building a prototype so this isn't an issue, I will face. However, if this application was to be deployed for a customer to mitigate this risk, I would conduct a compatibility test of the new application with existing systems before deployment to identify and address any potential issues.**

Risks – (2) Risks that your project poses to others

Below is a list of potential risks for my project that may harm others and how to reduce them to acceptable levels:

- **Data privacy and security:** The app will handle sensitive patient information and it's important to ensure that this data is protected from unauthorized access or misuse. **As this is a student project, I will be using dummy data, however, if I were to deploy this project on a wider scale and use real patient records to reduce this risk, appropriate security measures such as encryption, secure data storage, and access controls would be implemented and regularly tested.**
- **User error:** Care home staff may use the app incorrectly, leading to incorrect or inconsistent patient information. **This product is only a prototype, so I will not face this risk; however, to reduce such risk, comprehensive user training and support should be provided, and user-friendly interfaces should be implemented in the app.**

References

(Kshitij, 2022)

Ethics Checklist

Research Ethics Review Form: BSc, MSc, and MA Projects

Computer Science Research Ethics Committee (CSREC)

<http://www.city.ac.uk/departments-computer-science/research-ethics>

Undergraduate and postgraduate students undertaking their final project in the Department of Computer Science are required to consider the ethics of their project work and to ensure that it complies with research ethics guidelines. In some cases, a project will need approval from an ethics committee before it can proceed. Usually, but not always, this will be because the student is involving other people ("participants") in the project.

To ensure that appropriate consideration is given to ethical issues, all students must complete this form and attach it to their project proposal document. There are two parts:

PART A: Ethics Checklist. All students must complete this part.

The checklist identifies whether the project requires ethical approval and, if so, where to apply for approval.

PART B: Ethics Proportionate Review Form. Students who have answered "no" to all questions in A1, A2 and A3 and "yes" to question 4 in A4 in the ethics checklist must complete this part. The project supervisor has delegated authority to provide approval in such cases that are considered to involve MINIMAL risk.

The approval may be **provisional** – identifying the planned research as likely to involve MINIMAL RISK.

In such cases, you must additionally seek **full approval** from the supervisor as the project progresses and details are established. **Full approval** must be acquired in writing, before beginning the planned research.

A.1 If you answer YES to any of the questions in this block, you must apply to an appropriate external ethics committee for approval and log this approval as an External Application through Research Ethics Online - https://ethics.city.ac.uk/		<i>Delete as appropriate</i>
1.1	Does your research require approval from the National Research Ethics Service (NRES)? <i>e.g., because you are recruiting current NHS patients or staff?</i> <i>If you are unsure, try - https://www.hra.nhs.uk/approvals-amendments/what-approvals-do-i-need/</i>	NO
1.2	Will you recruit participants who fall under the auspices of the Mental Capacity Act? <i>Such research needs to be approved by an external ethics committee such as NRES or the Social Care Research Ethics Committee - http://www.scie.org.uk/research/ethics-committee/</i>	NO
1.3	Will you recruit any participants who are currently under the auspices of the Criminal Justice System, for example, but not limited to, people on remand, prisoners, and those on probation? <i>Such research needs to be authorised by the ethics approval system of the National Offender Management Service.</i>	NO
A.2 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee, you must apply for approval from the Senate Research Ethics Committee (SREC) through Research Ethics Online - https://ethics.city.ac.uk/		<i>Delete as appropriate</i>
2.1	Does your research involve participants who are unable to give informed consent? <i>For example, but not limited to, people who may have a degree of learning disability or mental health problem, which means they are unable to make an informed decision on their behalf.</i>	NO

2.2	Is there a risk that your research might lead to disclosures from participants concerning their involvement in illegal activities?	NO
2.3	Is there a risk that obscene and or illegal material may need to be accessed for your research study (including online content and other material)?	NO
2.4	Does your project involve participants disclosing information about a special category or sensitive subjects? <i>For example, but not limited to racial or ethnic origin; political opinions; religious beliefs; trade union membership; physical or mental health; sexual life; criminal offences and proceedings</i>	NO
2.5	Does your research involve you travelling to another country outside of the UK, where the Foreign & Commonwealth Office has issued a travel warning that affects the area in which you will study? <i>Please check the latest guidance from the FCO - http://www.fco.gov.uk/en/</i>	NO
2.6	Does your research involve invasive or intrusive procedures? <i>These may include but are not limited to, electrical stimulation, heat, cold or bruising.</i>	NO
2.7	Does your research involve animals?	NO
2.8	Does your research involve the administration of drugs, placebos, or other substances to study participants?	NO
A.3 If you answer YES to any of the questions in this block, then unless you are applying to an external ethics committee or the SREC, you must apply for approval from the Computer Science Research Ethics Committee (CSREC) through Research Ethics Online - https://ethics.city.ac.uk/ Depending on the level of risk associated with your application, it may be referred to the Senate Research Ethics Committee.		<i>Delete as appropriate</i>
3.1	Does your research involve participants who are under the age of 18?	NO
3.2	Does your research involve adults who are vulnerable because of their social, psychological, or medical circumstances (vulnerable adults)? <i>This includes adults with cognitive and/or learning disabilities, adults with physical disabilities and older people.</i>	NO
3.3	Are participants recruited because they are staff or students of City, University of London? <i>For example, students studying a particular course or module.</i> <i>If yes, then approval is also required from the Head of the Department or Programme Director.</i>	NO
3.4	Does your research involve intentional deception of participants?	NO
3.5	Does your research involve participants taking part without their informed consent?	NO
3.5	Is the risk posed to participants greater than that in normal working life?	NO
3.7	Is the risk posed to you, the researcher(s), greater than that in normal working life?	NO

<p>A.4 If you answer YES to the following question and your answers to all other questions in sections A1, A2 and A3 are NO, then your project is deemed to be of MINIMAL RISK.</p> <p>If this is the case, then you can apply for approval through your supervisor under PROPORTIONATE REVIEW. You do so by completing PART B of this form.</p> <p>If you have answered NO to all questions on this form, then your project does not require ethical approval. You should submit and retain this form as evidence of this.</p>		<i>Delete as appropriate</i>
4	<p>Does your project involve human participants or their identifiable personal data?</p> <p><i>For example, as interviewees, respondents to a survey or participants in testing.</i></p>	YES

PART B : Ethics Proportionate Review Form

If you answered YES to question 4 and NO to all other questions in sections A1, A2 and A3 in PART A of this form, then you may use PART B of this form to apply for a proportionate ethics review of your project. Your project supervisor has delegated authority to review and approve this application under proportionate review. You must receive final approval from your supervisor in writing before beginning the planned research.

However, if you cannot provide all the required attachments (see B.3) with your project proposal (e.g., because you have not yet written the consent forms, interview schedules etc), the approval from your supervisor will be **provisional**. You **must** submit the missing items to your supervisor for approval prior to commencing these parts of your project. Once again, you must receive written confirmation from your supervisor that any provisional approval has been superseded by with **full approval** of the planned activity as detailed in the full documents. **Failure to follow this procedure and demonstrate that final approval has been achieved may result in you failing the project module.**

Your supervisor may ask you to submit a full ethics application through Research Ethics Online, for instance if they are unable to approve your application, if the level of risks associated with your project change, or if you need an approval letter from the CSREC for an external organisation.

B.1 The following questions must be answered fully. All grey instructions must be removed.		Delete as appropriate
1.1.	Will you ensure that participants taking part in your project are fully informed about the purpose of the research?	YES
1.2	Will you ensure that participants taking part in your project are fully informed about the procedures affecting them or affecting any information collected about them, including information about how the data will be used, to whom it will be disclosed, and how long it will be kept?	YES
1.3	When people agree to participate in your project, will it be made clear to them that they may withdraw (i.e., not participate) at any time without any penalty?	YES
1.4	Will consent be obtained from the participants in your project? Consent from participants will be necessary if you plan to involve them in your project or if you plan to use identifiable personal data from existing records. "Identifiable personal data" means data relating to a living person who might be identifiable if the record includes their name, username, student id, DNA, fingerprint, address, etc.	YES
1.5	Have you planned to ensure that material and/or private information obtained from or about the participating individuals will remain confidential?	YES

B.2 If the answer to the following question (B2) is YES, you must provide details		Delete as appropriate
2	Will the research be conducted in the participant's home or other non-University location? <i>If YES, you must provide details of how your safety will be ensured.</i>	NO

B.3 Attachments			
ALL the following documents MUST be provided to supervisors if applicable.	YES	NO	Not Applicable

All must be considered prior to final approval by supervisors. A written record of final approval must be provided and retained.			
Details on how safety will be assured in any non-University location, including risk assessment if required (see B2)			✓
Details of arrangements to ensure that material and/or private information obtained from or about the participating individuals will remain confidential (see B1.5) <i>Any personal data must be acquired, stored, and made accessible in ways that are GDPR compliant.</i>			✓
Full protocol for any workshops or interviews**			✓
Participant information sheet(s)**	To be provided		
Consent form(s)**	To be provided		
Questionnaire(s)** <i>sharing a Qualtrics survey with your supervisor is recommended.</i>	To be provided		
Topic guide(s) for interviews and focus groups**			✓
Permission from external organisations or Head of Department** <i>e.g., for recruitment of participants</i>			✓

****If these items are not available at the time of submitting your project proposal, then *provisional approval* can still be given, under the condition that you must submit the final versions of all items to your supervisor for approval later. *All such items must be seen and approved by your supervisor before the activity for which they are needed begins. Written evidence of **final approval** of your planned activity must be acquired from your supervisor before you commence.***

Changes

If your plans change and any aspects of your research that are documented in the approval process change therefore, then any approval acquired is invalid. If issues addressed in Part A (the checklist) are affected, then you must complete the approval process again and establish the kind of approval that is required. If issues addressed in Part B are affected, then you must forward updated documentation to your supervisor and have received written confirmation of approval of the revised activity before proceeding.

Templates for Consent and Information

You must use the templates provided by the University as the basis for your participant information sheets and consent forms. You **must** adapt them according to the needs of your project before you submit them for consideration.

Participant Information Sheets, Consent Forms and Protocols must be consistent. Please ensure that this is the case prior to seeking approval. Failure to do so will slow down the approval process.

We strongly recommend using Qualtrics to produce digital information sheets and consent forms.

Further Information

<http://www.city.ac.uk/departments-computer-science/research-ethics>

<https://www.city.ac.uk/research/ethics/how-to-apply/participant-recruitment>

<https://www.city.ac.uk/research/ethics>

(APPENDIX B) – Code Reuse (Adapted Code)

I used this link to help me write the code below for the onboarding screens
<https://www.youtube.com/watch?v=VMiaNFabsZA>

```
import UIKit

class OnboardingViewController: UIViewController {

    //IBOutlets

    @IBOutlet weak var collectionView: UICollectionView!
    @IBOutlet weak var nextButton: UIButton!
    @IBOutlet weak var pageControl: UIPageControl!

    //Properties
    var slides: [OnboardingSlide] = []

    var currentPage = 0 {
        didSet {
            // Update the page control and next button's title based on the current page index
            pageControl.currentPage = currentPage
            if currentPage == slides.count - 1 {
                nextButton.setTitle("Get Started", for: .normal)
            } else {
                nextButton.setTitle("Next", for: .normal)
            }
        }
    }

    // Lifecycle Methods
    override func viewDidLoad() {
        super.viewDidLoad()

        // Create onboarding slides and set up the collection view
```

I used this link to help me write the code for the login and sign-up page using Firebase.

https://www.youtube.com/watch?v=-cYLkXKNDJA&ab_channel=KamrynOhly

```
import UIKit
import Firebase

class LoginViewController: UIViewController {
    @IBOutlet weak var emailTextField: UITextField!
    @IBOutlet weak var passwordTextField: UITextField!

    private let firebaseService = FirebaseService()

    override func viewDidLoad() {
        super.viewDidLoad()
    }

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        view.endEditing(true)
    }

    // Triggered when user clicks on Login button
    @IBAction func loginClicked(_ sender: UIButton) {
        // Check if email and password are not empty
        guard let email = emailTextField.text else {return}
        guard let password = passwordTextField.text else {return}
        view.showActivityIndicator()
        // Call FirebaseService to sign in user
        FirebaseService().signIn(email: email, password: password) { result in
```

I used this link to help me write the code for the search bar below.
https://www.youtube.com/watch?v=Vb1gVYj0rxo&ab_channel=AbdulAzeem

```
extension PatientViewController: UISearchBarDelegate {

    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        let text = searchText.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !text.isEmpty else {
            // If the search bar is empty, reset the filtered data and reload the collection view
            isSearching = false
            filteredMeds = []
            medicinesCollectionView.reloadData()
            return
        }
        // Filter the patient data by name and reload the collection view with the filtered data
        isSearching = true
        filteredMeds =
        patient?.medications?.filter({$0.medicalName.lowercased().contains(text.lowercased())}) ?? []
    }
}
```

I used this link as an overview help me write the code below
<https://developer.clevertap.com/docs/push-notifications-ios>

```
// If a date is provided, create a calendar-based trigger for the notification
if let date {
    let fireDate = Calendar.current.dateComponents([.hour, .minute, .second], from: date)
    trigger = UNCalendarNotificationTrigger(dateMatching: fireDate, repeats: repeats)
}
// If a trigger time is provided, create a time interval-based trigger for the notification
else if let triggerTime {
    guard triggerTime >= 60 else { return }
    trigger = UNTimeIntervalNotificationTrigger(timeInterval: TimeInterval(triggerTime),
```


I used this link to help me write the code to navigate to another view controller
<https://stackoverflow.com/questions/27374759/programmatically-navigate-to-another-view-controller-scene>

```
private func presentViewController(patient: PatientProfile) {  
    let storyboard: UIStoryboard = UIStoryboard(name: "Main", bundle: nil)  
    let patientViewController = storyboard.instantiateViewController(withIdentifier:  
"PatientViewController") as! PatientViewController  
    patientViewController.patient = patient  
    patientViewController.delegate = self  
    self.present(patientViewController, animated: true)
```

I used this link to help me write the code to get the search bar text in Swift
<https://stackoverflow.com/questions/30058311/get-searchbar-text-in-swift>

```
// Extension to handle search bar functionality  
extension PatListViewController: UISearchBarDelegate {  
    // This method is called whenever the search bar text changes  
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {  
        let text = searchText.trimmingCharacters(in: .whitespacesAndNewlines)  
        guard !text.isEmpty else {  
            // If the search bar is empty, reset the filtered data and reload the collection view  
            isSearching = false  
            filteredPatients = []  
            patientListCollectionView.reloadData()  
            return  
        }  
        // Filter the patient data by name and reload the collection view with the filtered data  
        isSearching = true  
        filteredPatients = patients.filter({$0.fullName!.lowercased().contains(text.lowercased())})
```

I used this link to help me write the code below

<https://stackoverflow.com/questions/35586512/uicollectionview-numberofitemsinsection-being-called-twice>

```
// This method returns the number of cells to display in the collection view

func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section:
Int) -> Int {

    print(patients.count)

    return isSearching ? filteredPatients.count : patients.count
```

I used this link to help me write the code below

<https://www.hackingwithswift.com/example-code/uikit/how-to-register-a-cell-for-uicollectionview-reuse>

```
// This method dequeues and configures each cell in the collection view

func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath)
-> UICollectionViewCell {

    let cell = collectionView

        .dequeueReusableCell(withReuseIdentifier: PatientListCollectionViewCell.identifier, for:
indexPath) as! PatientListCollectionViewCell

    cell.patientDelegate = self

    cell.setup(category: isSearching ? filteredPatients[indexPath.item] :
```

I used this link to help me write the code below

<https://stackoverflow.com/questions/40004856/collectionviewdelegate-method-doesnt-get-called-after-swift-3-migration>

```
func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath:
IndexPath) {

    let profile = patients[indexPath.row]

    presentViewController(patient: profile)
```

(APPENDIX C) – System Requirements Gathering

I went through a rigorous process to determine the requirements for my system and to commence work on this project I assessed its current system status while also determining its objectives and identifying every stakeholder that shall play a vital role.

Considering various restrictions such as limited budget or technical feasibility I prioritized requirements depending on how crucial they are for the accomplishment of the project.

Furthermore, I implemented diverse methodologies including conducting surveys, interviews and organizing focus group discussions among other things, to elicit valuable feedback from relevant stakeholders thus ascertaining that requirements duly capture their needs. Lastly, the set of chosen requirements outlines a specific and comprehensive plan for constructing the system in line with project objectives.

Requirement ID: MP001	Requirement Type: FR	Event/Use Case #2
Description: The system shall allow users to create an account by providing a unique username, a secure password, and an email address.		
Rationale: Users need to have a unique identity within the system to be able to securely store and manage their personal information.		
Source: n/a		
Fit Criteria: The system shall verify that the provided email address is valid and not already associated with an existing account. Upon successful sign up, the user shall be redirected to the login page.		
Customer Satisfaction: 4	Customer Dissatisfaction: 5	
Priority: High	Conflicts: None	
Supporting Material: User authentication wireframe		Volere Source: Atlantic Systems Guild
History:		

Requirement ID: MP002	Requirement Type: FR	Event/Use Case #6
Description: The system shall allow Caregivers to set up and view Patient profiles.		
Rationale: Caregivers need to be able to add and have access to patient information within the system to provide appropriate care.		
Source: Feedback from Caregivers		
Fit Criteria: The system shall allow Caregivers to set up a Patient profile. The system shall display the Patient's profile information, including medication lists, appointment schedules, and any other relevant details provided by the Caregiver. Caregivers shall be able to edit and update Patient profile information as needed.		
Customer Satisfaction: 5	Customer Dissatisfaction: 3	

Priority: High	Conflicts: None
Supporting Material: Patient profile wireframe	Volere Source: Atlantic Systems Guild
History:	

Requirement ID: MP003-001	Requirement Type: FR	Event/Use Case #7
Description: The system shall allow Caregivers to add medications for a patient.		
Rationale: Caregivers need to keep track of the medications given to the patient.		
Source: Feedback from Caregivers		
Fit Criteria: The system shall allow Caregivers to add medication name, dosage, and frequency for a patient.		
Customer Satisfaction: 5	Customer Dissatisfaction: 3	
Priority: Essential	Conflicts: None	
Supporting Material: Add Medication wireframe	Volere Source: Atlantic Systems Guild	
History:		

Requirement ID: MP003-002	Requirement Type: FR	Event/Use Case #8
Description: The system shall allow Caregivers to edit medications for a patient.		
Rationale: Caregivers need to modify medication details when there are changes in the patient's treatment plan.		
Source: Feedback from Caregivers		
Fit Criteria: The system shall allow Caregivers to edit medication details for a patient.		
Customer Satisfaction: 4	Customer Dissatisfaction: 3	
Priority: Essential	Conflicts: None	
Supporting Material: Edit Medication wireframe	Volere Source: Atlantic Systems Guild	
History:		
Requirement ID: MP003-003	Requirement Type: FR	Event/Use Case #9

Description: The system shall allow Caregivers to delete medications for a patient.		
Rationale: Caregivers need to remove medications that are no longer required for the patient.		
Source: Feedback from Caregivers		
Fit Criteria: The system shall allow Caregivers to delete medication details for a patient.		
Customer Satisfaction: 3	Customer Dissatisfaction: 2	
Priority: Essential	Conflicts: None	
Supporting Material: Delete Medication wireframe		Volere Source: Atlantic Systems Guild
History:		

Requirement ID: MP004-001	Requirement Type: FR	Event/Use Case #10
Description: The app shall allow Caregivers to add appointments for patients.		
Rationale: Caregivers need to keep track of Patients' appointments to ensure proper care is provided.		
Source: Feedback from Caregivers		
Fit Criteria: Caregivers can successfully add a new appointment for a Patient and specify the appointment details.		
Customer Satisfaction: 4	Customer Dissatisfaction: 2	
Priority: Medium	Conflicts: None	
Supporting Material: Add Appointment wireframe		Volere Source: Atlantic Systems Guild
History:		

Requirement ID: MP004-002	Requirement Type: FR	Event/Use Case #12
Description: The app shall allow Caregivers to view appointments for patients.		
Rationale: Caregivers need to know about upcoming appointments to properly care for Patients.		
Source: Feedback from Caregivers		
Fit Criteria: Caregivers can successfully view a list of upcoming appointments for a selected Patient, including details such as the date, time and location.		
Customer Satisfaction: 4	Customer Dissatisfaction: 2	

Priority: Low	Conflicts: None
Supporting Material: View the Appointment wireframe	Volere Source: Atlantic Systems Guild
History:	

Requirement ID: MP005	Requirement Type: FR	Event/Use Case #11
Description: The system shall provide pill reminders for Caregivers to ensure that Patients take their medication as prescribed.		
Rationale: Caregivers need to keep track of medication schedules to ensure the health and well-being of patients.		
Source: Feedback from Caregivers and industry best practices.		
Fit Criteria: The system shall send notifications to Caregivers when it is time for a Patient to take their medication.		
Customer Satisfaction: 5	Customer Dissatisfaction: 3	
Priority: High	Conflicts: None	
Supporting Material: Notification Page wireframe	Volere Source: Atlantic Systems Guild	
History:		

Requirement ID: MP006	Requirement Type: FR	Event/Use Case #11
Description: The system shall provide appointment reminders for Caregivers.		
Rationale: Caregivers need to be reminded of upcoming appointments to ensure they are prepared and can plan if necessary.		
Source: Feedback from Caregivers		
Fit Criteria: Caregivers will receive a notification of upcoming appointments.		
Customer Satisfaction: 4	Customer Dissatisfaction: 3	
Priority: High	Conflicts: None	
Supporting Material: Notification Page wireframe	Volere Source: Atlantic Systems Guild	
History:		

(APPENDIX D) – Initial Domain Model - CRC Cards

Visual Paradigm Professional (vshakoon@City University London)

User		CareGiver	
Super Classes: User		Sub Classes:	
Sub Classes: Caregiver		Description: A caregiver is a user who provides care to one or more patients	
Description: The user is the main entity of the app which includes caregiver.		Attributes:	
Name	Description	Name	Description
Username	The username for the user	Name	The name of the caregiver
Password	The password for the user	Patients	A list of patients the caregiver is responsible for
Email	The email of the user	Responsibilities:	
Name	Collaborator	Name	Collaborator
Sign up	Create a new user account	Set up patient profile	Create a new patient profile and it to the list of patient in that case home.
Login	Log into the app using user credentials	View patient profile	The information about a patient they are responsible for
View profile	View patient profiles	Add patient medicine	Add a new medication for a patient, they are responsible for
		Edit patient medicine	Modify an existing medication for a patient they are responsible for
		Delete patient medicine	Delete an existing medication for a patient they are responsible for
		Add patient appointment	Add a new appointment for a patient they are responsible for
		View patient appointments	View information about an appointment for a patient they are responsible for
		Delete patient appointment	Delete any appointments that a patient is no longer attending

(APPENDIX E) – Use case Index - Iterations

In my use case index below, I made sure to prioritize the items on the table according to logic. Multiple factors make this useful, recognizing pivotal use cases or requirements can help streamline the system development process. Prioritizing the crucial features based on my comprehensive knowledge allows me to allocate suitable resources and time for their timely completion.

Second, ordering the table by priority can help to ensure that the dependencies between different use cases are properly considered. For example, if the system allows staff members to add medication for patients, patient profiles must be set up first to avoid errors or inconsistencies. Prioritizing features based on their dependencies can help to ensure that the development process is efficient and effective.

Logical Order	Use Case	Priority Level	Justification
1	Login	High	Required for access to the system.
2	Sign Up	High	Important for new users to access the system.
3	View Patient List	High	Necessary so staff can view their patient profiles.

4	Set up Patient Profile	High	Necessary for adding new patients to the system.
5	View Patient Profile / Medicine	High	Necessary as this page is central that allows users to make changes to the patient details.
6	Add Patient Medicine	High	Critical for managing patient health and safety.
7	Edit Patient Medication	High	Critical for managing patient health and safety.
8	Delete Patient Medication	High	Critical for managing patient health and safety.
9	Add Patient Doctor Appt	High	Critical for managing patient healthcare appointments.
10	Appointment and Pill System Notifications	Medium	Useful for reminding staff of upcoming appointments and medication.
11	View Patient Doctor Appt.	Low	Useful but not critical for staff to view patient appointments.

(APPENDIX F) – Usage Model - Use case Specification.

Use case: Login	#1
Brief description: The User logs into the app using their registered email address and password.	
Primary actors: User (Care Giver) or Care Home Administrator	
Secondary actors: None	
Preconditions: The user is already registered for an account.	
Flow of events: <ol style="list-style-type: none"> 1. User opens the MediPill app. 2. The system displays 3 onboarding screens that the User can scroll through. 3. The User is then prompted to a “Welcome Back Screen”. 4. The user enters their registered email address and password. 5. The user clicks on the "login" button. 6. The system validates the user's login information. 7. If the login information is correct, the system logs the user into their account and displays the Care homes profile page. 	
Postconditions: The user is logged in to their account and can access their profile and other features of the app.	
Alternative flow: <ol style="list-style-type: none"> 1. The user enters incorrect login information. <p>If the user enters incorrect login information, the system will not let them press the Login button to access the rest of the system.</p>	

Use case: Sign Up	#2
Brief description: The User registers for an account for the individual Care Home providing necessary information such as Care Home Name, email address and password.	
Primary actors: Care Home Administrator	
Secondary actors: None	
Preconditions: None	
Flow of events: <ol style="list-style-type: none"> 1. Care Home Administrator launches MediPill app. 2. The system displays 3 onboarding screens that the User can scroll through. 3. The Care Home Administrator is then prompted to a “Welcome Back Screen”. 4. The user should then click the “Don’t have an account? Join us” text button. 	

<p>5. On the Join us Today Screen the User can fill out the 4 boxes; “Name of Care home”, “Name of Care home Administrator”, “Email” and “Password”.</p> <p>6. The user should then select the “Sign up” Button underneath.</p>
<p>Postconditions: The user is registered for an account and can now log in to the app using their credentials.</p>
<p>Alternative flow:</p> <p>1. The existing user attempts to register again.</p> <p>If a user attempts to register with an email address that is already associated with an existing account, the system will display an error message and prompt the user to log in instead.</p> <p>2. The user enters invalid information.</p> <p>If the user enters invalid or incomplete information, the system will display an error message and prompt the user to correct the information and resubmit the form.</p>

Use case: View Patent List	#3
<p>Brief description: A staff member logs in to their care home account, where they can view their list of assigned patients.</p>	
<p>Primary actors: User (Care Giver)</p>	
<p>Secondary actors: None</p>	
<p>Preconditions: The Care home has already registered for an account.</p>	
<p>Flow of events:</p> <ol style="list-style-type: none"> 1. Care Giver logs in to their care home account using their email and password. 2. System displays list of assigned patients, including their names, age, and time of next medication. 3. Care Giver can click on a patient's name to view their patient profile page. 	
<p>Postconditions: Staff member can view their individual profile and the list of patients assigned to them.</p>	
<p>Alternative flow:</p> <p>1. Care home has not yet assigned any patients.</p> <p>If the care home has not assigned any patients, the system will display a blank screen as there are no profiles to add.</p>	

Use case: Set up Patient Profile	#4
<p>Brief description: A Care Giver sets up a new patient profile on the system, adding relevant patient information. This profile can then be accessed by the staff member</p>	

on their "Patient List" page.
Primary actors: User (Care Giver)
Secondary actors: None
Preconditions: Staff member has access to the system and the necessary permissions to add new patient profiles.
Flow of events: <ol style="list-style-type: none"> 1. A Care Giver logs in to the system using their email and password. 2. Once logged in, the Care Giver is directed to the "Patients List" page. 3. Here the Care Giver can click the "Profile button" at the bottom of the screen where they will then be directed to a "Set up patient profile" Page. 4. The System displays a form for adding new patient information, including name, D.O.B, Blood Group, Emergency Contact Number, Emergency Contact Name, Attending Physician and Gender. 5. The Staff member fills out the form with the new patient's information and submits the form by clicking the "Set Up" button at the bottom. 6. The System creates a new patient profile and adds it to the "staff patient list" page. 7. Staff members can access the patient's profile by navigating to the "staff patients list page" and selecting the patient from the list.
Postconditions: New patient profile is created on the system and can be accessed by a staff member on their "Patients List" page.
Alternative flow: <p>1.Care Giver wants to edit a patient profile.</p> <p>If a staff member wants to update a patient's profile, they can navigate to the patient's profile and make changes.</p>

Use case: View Patient Profile / Medicine	#5
Brief description: Staff member views a patient's profile on the system, which contains information such as name, age, what medication they take and any other relevant details.	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Staff member has access to the system and the necessary permissions to view patient profiles.	
Flow of events: <ol style="list-style-type: none"> 1. Staff member logs in to the system using their email and password. 2. Once logged in, Staff member is directed to the "Patient List" page. 	

<ol style="list-style-type: none"> Here they can select the patient whose profile they want to view by clicking on the "Select Profile" button. The System displays the patient's profile, which contains information such as name, D.O.B, Blood Group, Emergency Contact Number, Emergency Contact Name, Attending Physician and Gender. It also shows the staff member what medication the patient takes along with the dosage and time for their next dose. Staff member can view the patient's profile information.
Postconditions: Staff member can view the patient's profile on the system.
Alternative flow: 1. Staff member wants to edit a patient profile. If a staff member wants to update a patient's profile, they can navigate to the patient's profile and make changes by clicking the "edit" button.

Use case: Add Patient Medicine	#6
Brief description: Staff member adds medication information for a patient, including the name of the medication, dosage, and frequency. The system generates notifications on the Care Givers phone at the designated times for taking the medication.	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Care Giver has access to the system and the necessary permissions to add medication information for patients.	
Flow of events: <ol style="list-style-type: none"> Care Giver selects the patient they want to add medication information for. Care Giver selects the Pill Icon button at the bottom of the screen on the "patient's profile" page. System displays the "Add Medication" form, where the Care Giver can enter medication information, including the name of the medication, dosage, and frequency. Care Giver enters the medication information and selects the "Save" button. System validates the entered information and confirms that the medication has been added to the patient's profile. System generates notifications on the Care Giver's phone at the designated times for taking the medication. Care Giver can view the added medication information on the patient's medication page. 	
Postconditions: Medication information is added to the patient's profile and the system generates notifications on the Care Givers phone at the designated times for taking the medication.	
Alternative flow:	

1.1. Care Giver wants to edit medication information.

If a Care Giver wants to edit medication information for a patient, they can navigate to the patient's medication section and make changes by selecting the "Edit" button.

1.2. Care Giver wants to delete medication information.

If a Care Giver wants to delete medication information for a patient, they can navigate to the patient's medication section and click on the "Edit" button for the respective medication which then navigates them to a screen where they can then select the "Delete Medication" button.

Use case: Edit Patient Medication	#7
Brief description: Staff member edits medication information for a patient, including the name of the medication, dosage, and frequency. The system adjusts the notifications on the Care Givers phone at the designated times for taking the medication accordingly.	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Care Giver has access to the system and the necessary permissions to edit medication information for patients.	
Flow of events: <ol style="list-style-type: none"> 1. Care Giver navigates to the specific patient's Profile they want to make the changes in. 2. The Care Giver then select the medication they want to edit. 3. The system displays the current medication information for the selected medication. 4. Care Giver updates the medication information, including the name of the medication, dosage, and frequency. 5. The system updates the medication information for the patient in the system. 6. The system adjusts the notifications on the Care Givers phone at the designated times for taking the medication accordingly. 	
Postconditions: The medication information for the patient is updated in the system, and the notifications on the Care Givers phone are adjusted accordingly.	
Alternative flow: None	

Use case: Delete Patient Medication	#8
Brief description: Staff member deletes medication information for a patient. The system removes the notifications on the Care Givers phone for the deleted	

medication.
Primary actors: User (Care Giver)
Secondary actors: None
Preconditions: Care Giver has access to the system and the necessary permissions to delete medication information for patients.
Flow of events: <ol style="list-style-type: none"> 1. Care Giver navigates to the specific patient's Profile they want to make the changes in. 2. The Care Giver then select the medication they want to delete by clicking the "Edit" Button. 3. The system displays the current medication information for the selected medication. 4. Care Giver select the "Delete Medication" Button at the bottom of the screen. 5. The system removes the medication information for the patient in the system. 6. The system removes the notifications on the Care Givers phone for the deleted medication.
Postconditions: The medication information for the patient is deleted in the system, and the notifications on the Care Givers phone for the deleted medication are removed.
Alternative flow: None

Use case: Add Patient's Doctor Appt	#9
Brief description: Staff member adds an appointment for a patient, including the date, location, time, and set a 20-40 reminder before the appointment.	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Care Giver has access to the system and the necessary permissions to add appointments for patients.	
Flow of events: <ol style="list-style-type: none"> 1. Care Giver navigates to the patient's profile page. 2. Care Giver selects the "calendar Icon" at the bottom of the screen allowing for them to view the patients' appointments. 3. The caregiver can then select the "calendar icon with a plus" at the bottom of the screen to add a new appointment. 4. The system prompts the Care Giver to enter the appointment details, including date, location, time and can set a 20-40 reminder before the appointment. 5. Care Giver enters the appointment details and selects the "Confirm" button. 	

6. The system saves the appointment details and generates a reminder notification on the Care Givers phone 20-40 minutes before the appointment time. 7. Care Giver receives the reminder notification on their phone. 8. Care Giver can select the notification to open the app and view the appointment details.
Postconditions: The appointment information is added to the patient's profile and a reminder notification is generated on the Care Givers phone 20-40 minutes before the appointment time depending on what they selected.
Alternative flow: None

Use case: Appointment and Pill System Notifications	#10
Brief description:	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Care Giver has access to the system and has added an appointment or a medication for a patient.	
Flow of events: <ol style="list-style-type: none"> Caregiver logs in to MediPill app and views the list of patients. Caregiver selects a patient and views their medication schedule for the day. MediPill app sends a push notification to the Caregiver reminding them to administer the medication to the patient. Caregiver administers the medication to the patient. Caregiver logs the administration of the medication in the app by marking it as taken. Caregiver adds a doctor's appointment to the patient's profile in the app. MediPill app sends a push notification to the Caregiver reminding them of the patient's upcoming appointment. Caregiver ensures the patient is ready for the appointment and accompanies them to the appointment if necessary. 	
Postconditions: The appointment information is saved to the Patient's profile and can be viewed by the caregiver.	
Alternative flow: None	

Use case: View Patient's Doctor Appt	#11
Brief description:	
Primary actors: User (Care Giver)	
Secondary actors: None	
Preconditions: Care Giver has access to the system and the necessary permissions to	

view patient appointments.

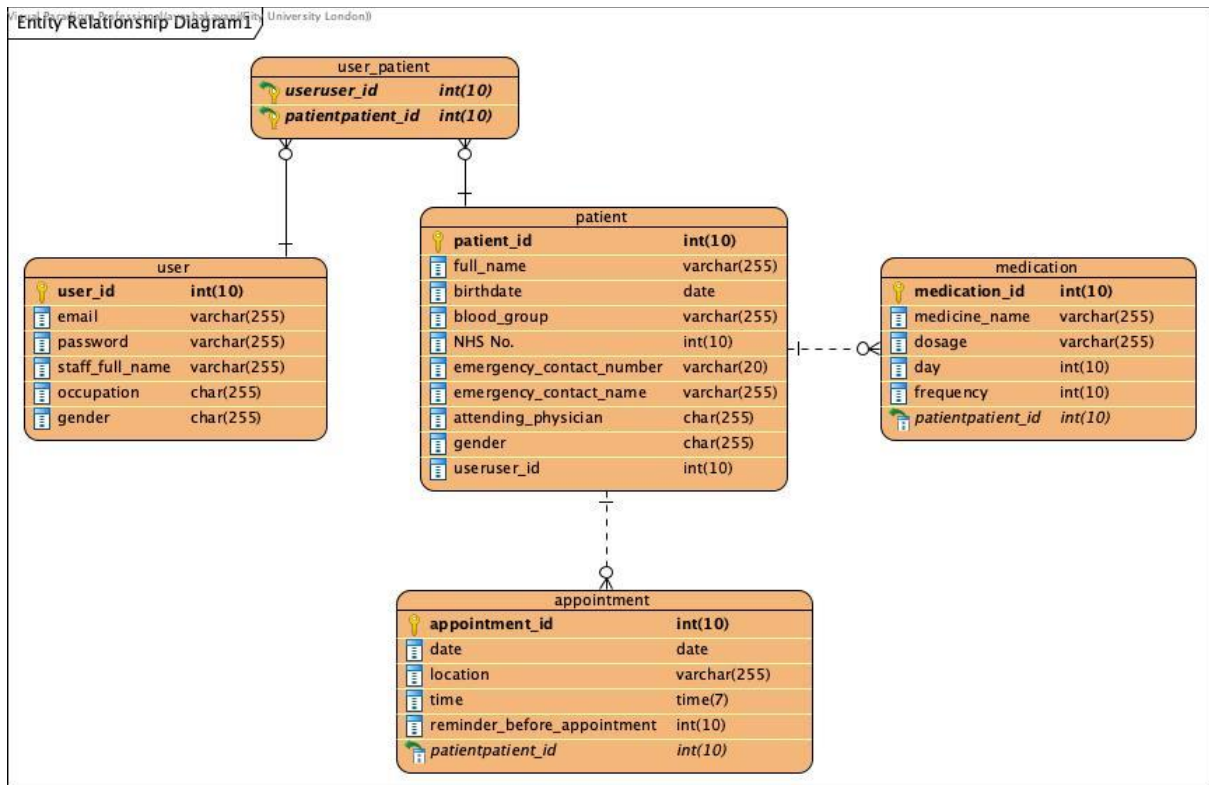
Flow of events:

1. Care Giver logs into the system.
2. Care Giver navigates to the patient's profile page.
3. Care Giver selects the "calendar Icon" at the bottom of the screen to view the patient's appointments.
4. The system displays a list of all appointments for the selected patient, including the appointment date, time, and location.

Postconditions: Care Giver views the patient's appointment information.

Alternative flow: None

(APPENDIX G) – Entity relationship diagram



Entities:

- User (which includes both Care Givers and Care home administrator)
- Patient
- Medication
- Appointment

Relationships between these entities are:

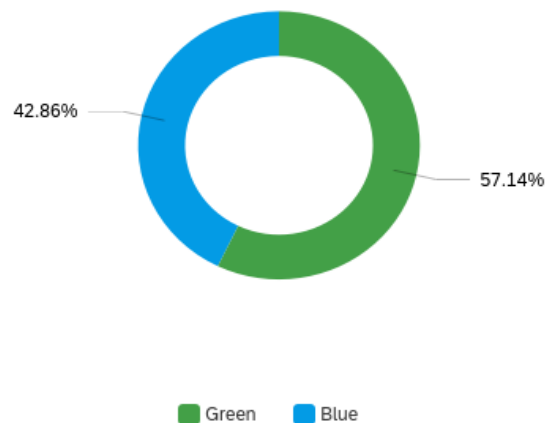
- A User can have multiple Patients (**many-to-many**) – Many staff can share patients.
- A Patient can have multiple Medications (**one-to-many**)
- A Patient can have multiple Appointments (**one-to-many**)

Default Report

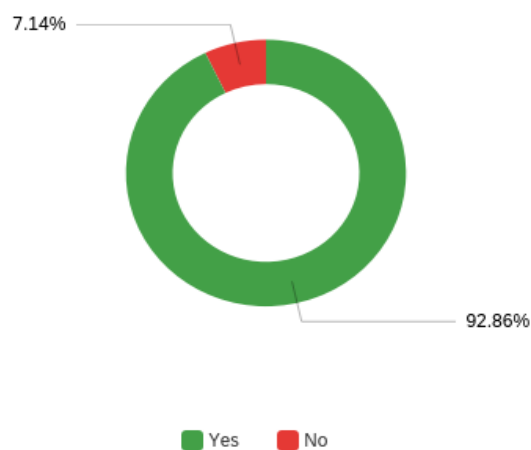
MediPill Logo

May 4th 2023, 2:52 pm BST

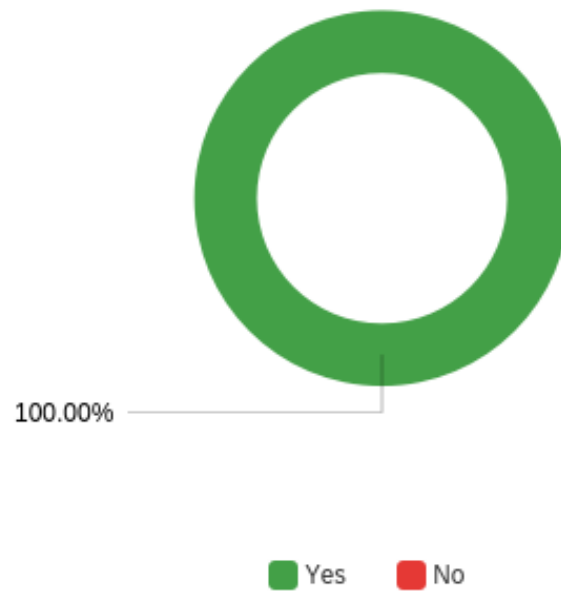
Q1 - Does the green / blue logo resonate with the healthcare industry?



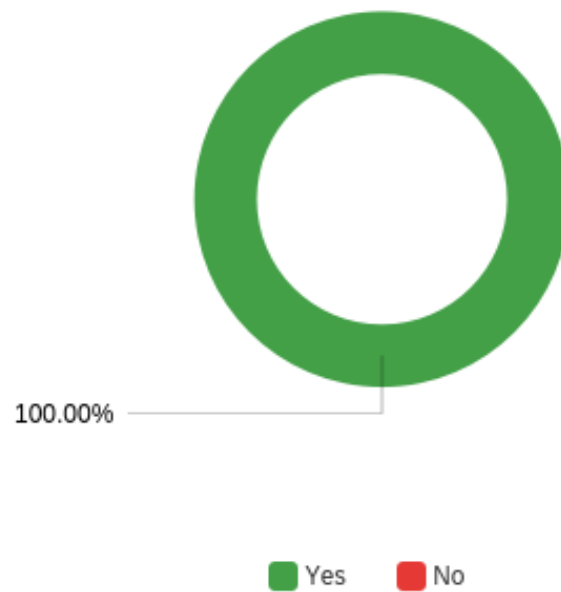
Q2 - Does the app name accurately convey the purpose of the app?



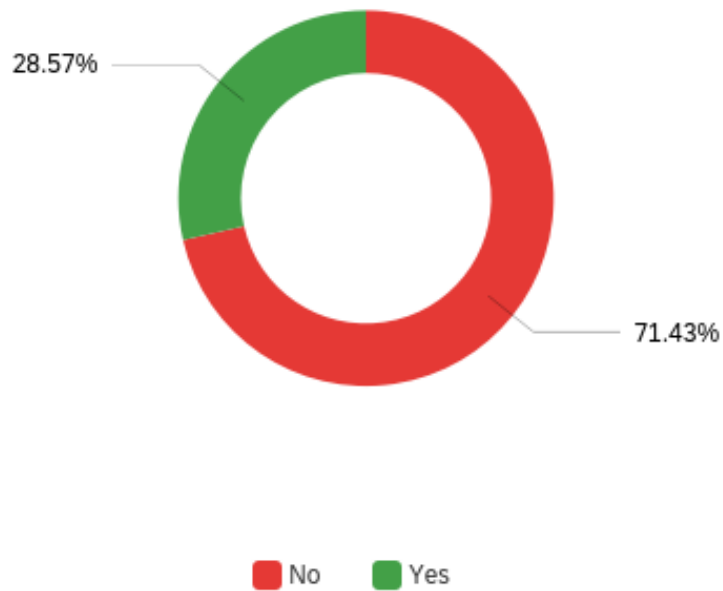
Q3 - Does the app logo feel modern and up to date?



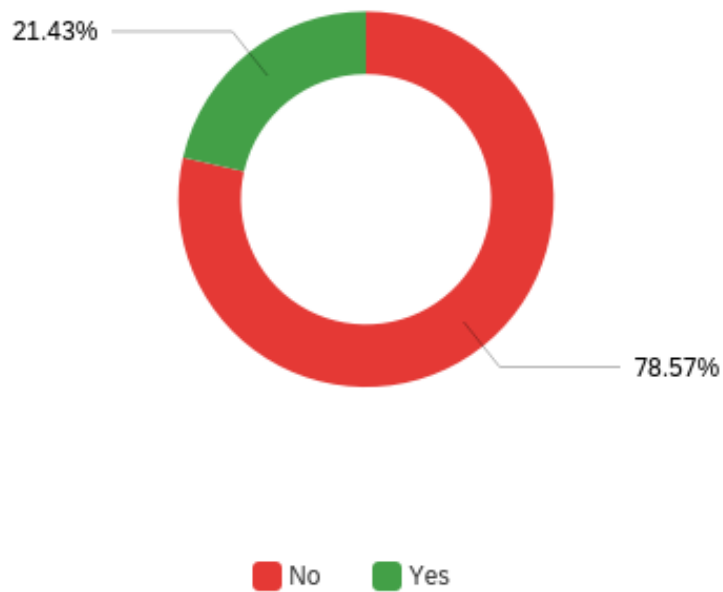
Q4 - Does the app logo feel trustworthy and credible?



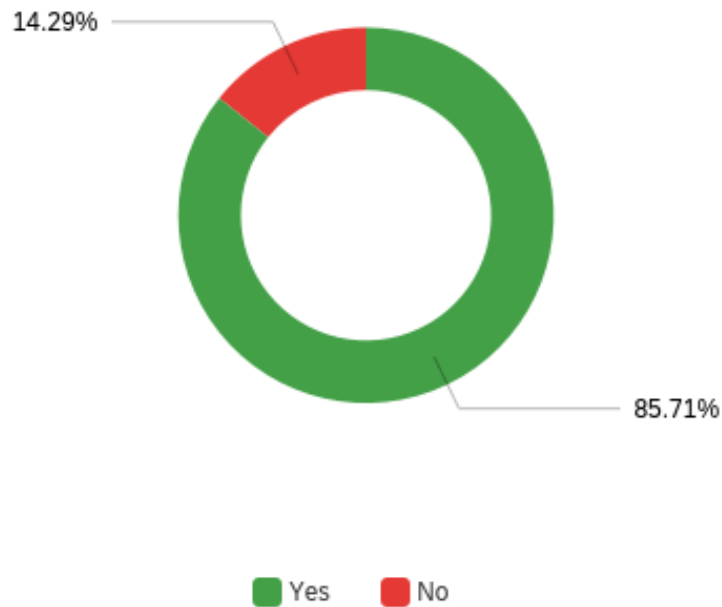
Q5 - Does the app logo feel gender-neutral?



Q6 - Is the app logo memorable?



Q7 - Is the app logo easy to recognize and distinguish from other logos in the healthcare industry?



Q8 - Is there anything about the healthcare app logo that you would change or improve?

Is there anything about the healthcare app logo that you would change or improve?

Joining the words "Health" and "care" together to state "Healthcare" makes more sense perhaps

maybe add pills?

Maybe the heart shape needs to change a little

It looks modern

No

Can use without Wi-Fi

N/A

Default Report

MediPill UI Development

May 4th 2023, 3:17 pm BST

Q1 - What do you like the most about the current user interface of the MediPill app?

What do you like the most about the current user interface of the MediPill app?

its simple and straightforward

Ease of use

Consistent colour scheme, clean layout

There is one colour scheme being used throughout

Good colour scheme and large buttons

Good

It's software

It is simple and not overwhelming

It is very appealing and efficient in containing information which is necessary

Able to find what you're looking for. Very clear.

Ease of access

Easy to use

colour

Q2 - What do you dislike the most about the current user interface of the MediPill app?

What do you dislike the most about the current user interface of the MediPill app?

think dark mode will be better suited

Nothing

The large grey circles at the start

Nothing

A bit too much green. Looks like an Aero bar

Nothing

Nothing

Nothing

Nothing

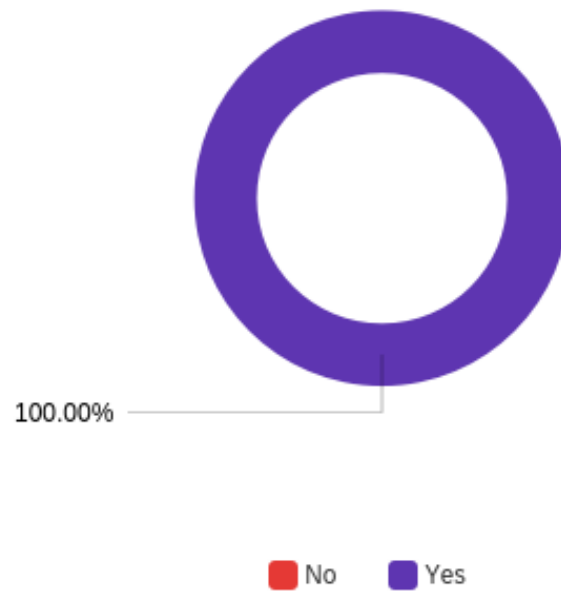
Nothing. It looks very good and professional

Not good with new tech, slowly learning how to use it

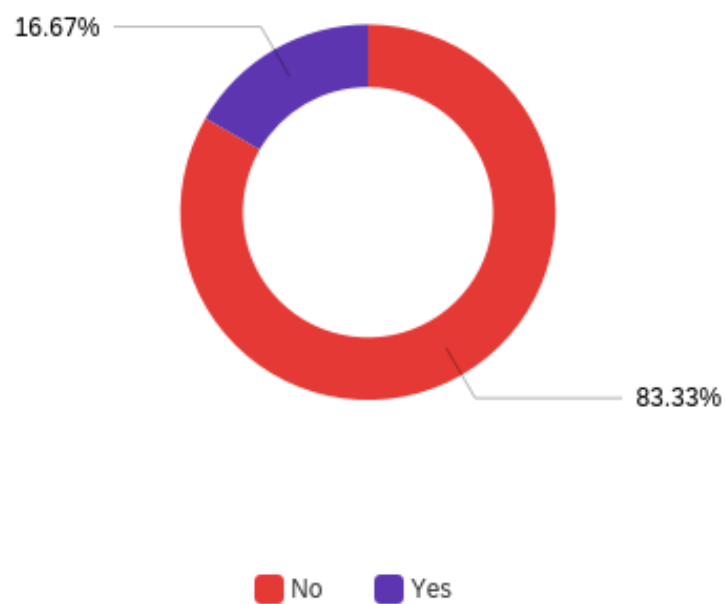
Not much

nothing

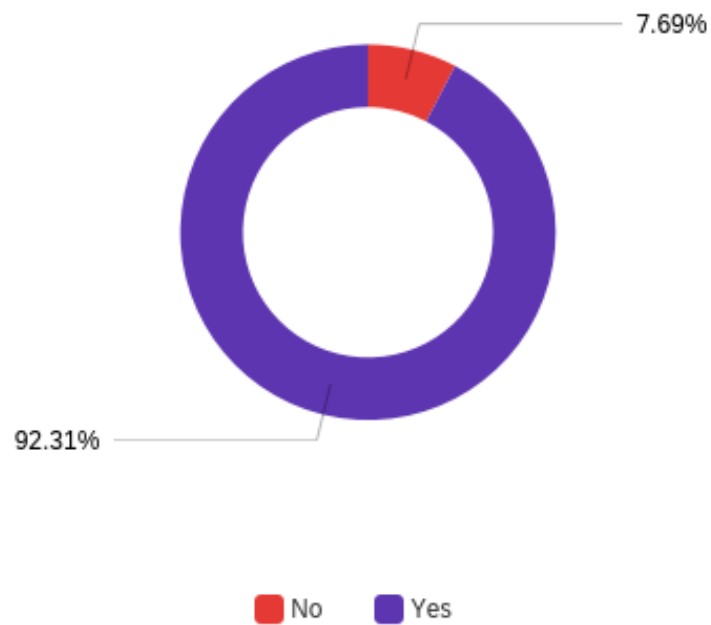
Q3 - Are the buttons and icons on the app easy to understand and navigate?



Q4 - Are there any features that you find difficult to locate on the app?



Q5 - Is the color scheme of the app aesthetically pleasing?



Q6 - How would you rate the overall user experience of the MediPill app on a scale of 1-10? With 10 being Excellent.

#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	User Experience	7.00	10.00	8.85	0.86	0.75	13

Q7 - Is there anything else you would like to add or suggest to improve the user interface of the MediPill app?

Is there anything else you would like to add or suggest to improve the user interface of the MediPill app?

No

No

I suggest no changes other than adding 1 more colour

No

No

No

N/A

Nothing

N/A

Nope

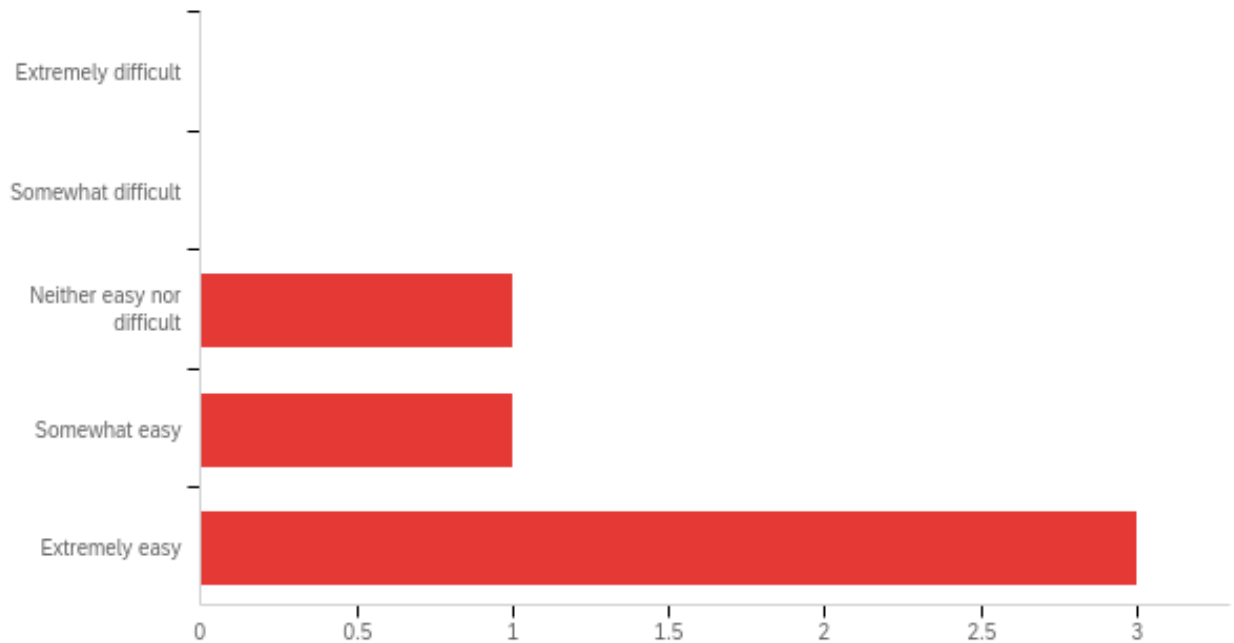
no

Default Report

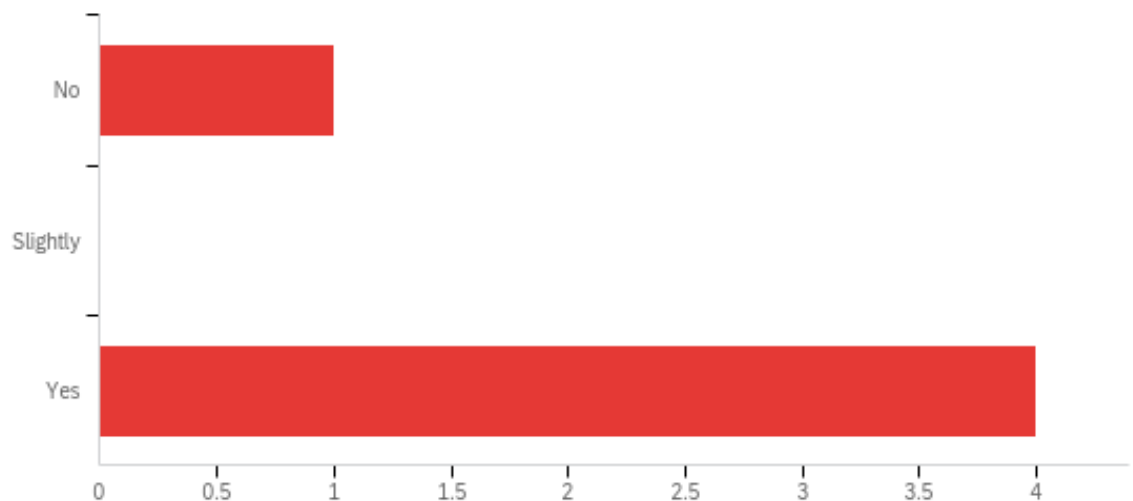
MediPill App Testing

May 4th 2023, 3:21 pm BST

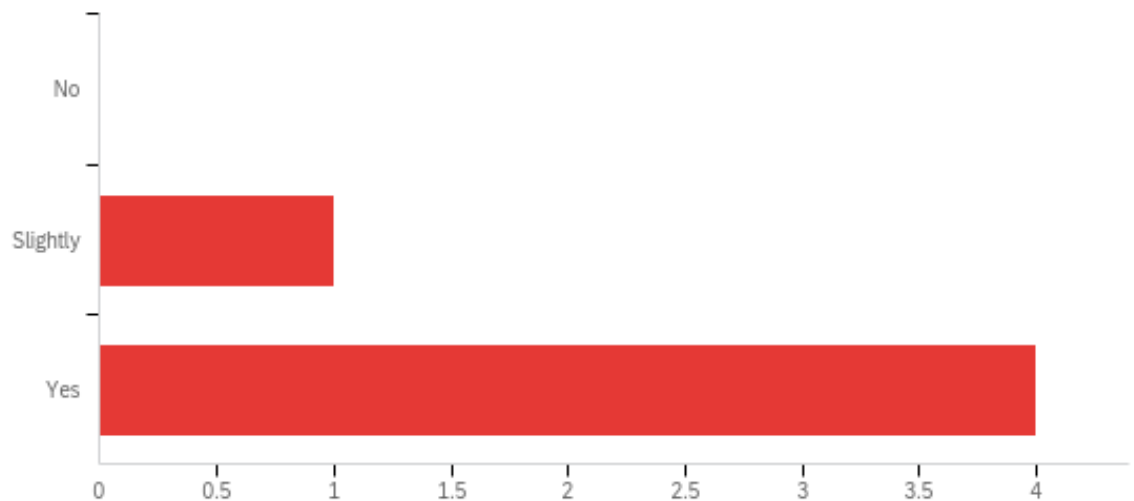
Q1 - How easy was it to sign up and log in to the app?



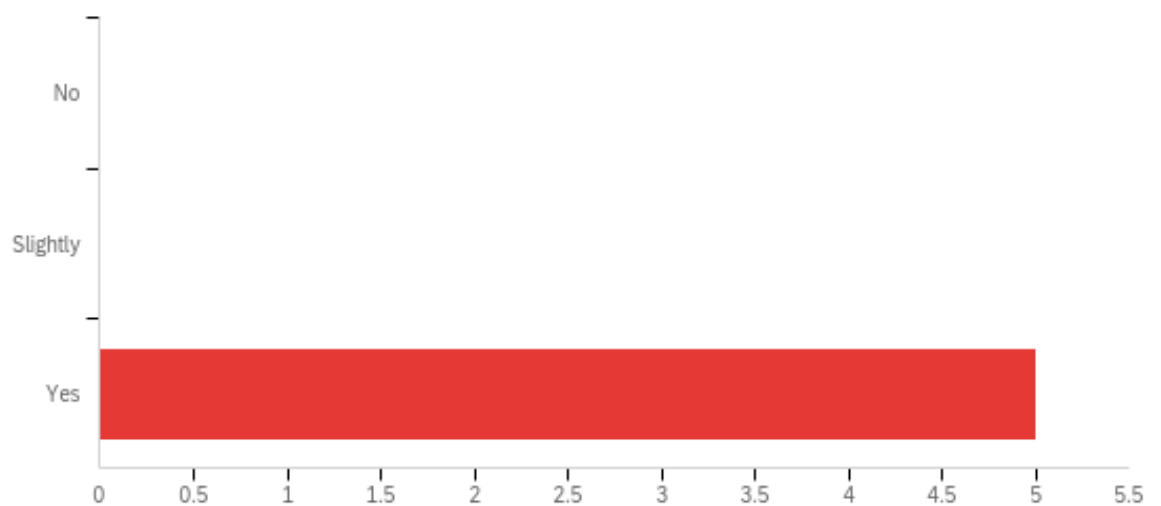
Q2 - Did you find the user interface easy to use?



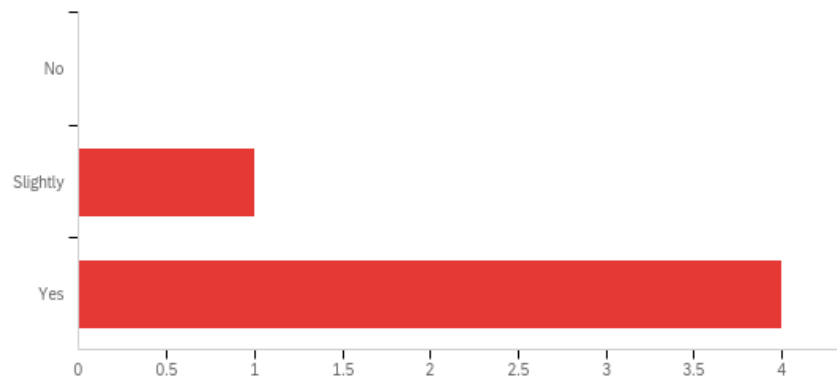
Q3 - Were the features of the app easy to locate and use?



Q4 - Did you find the app helpful in managing medication reminders for patients?



Q5 - Did you find the patient profiles easy to set up and manage?



Q6 - Were there any features missing that you would like to see added to the app?

Were there any features missing that you would like to see added to the app?

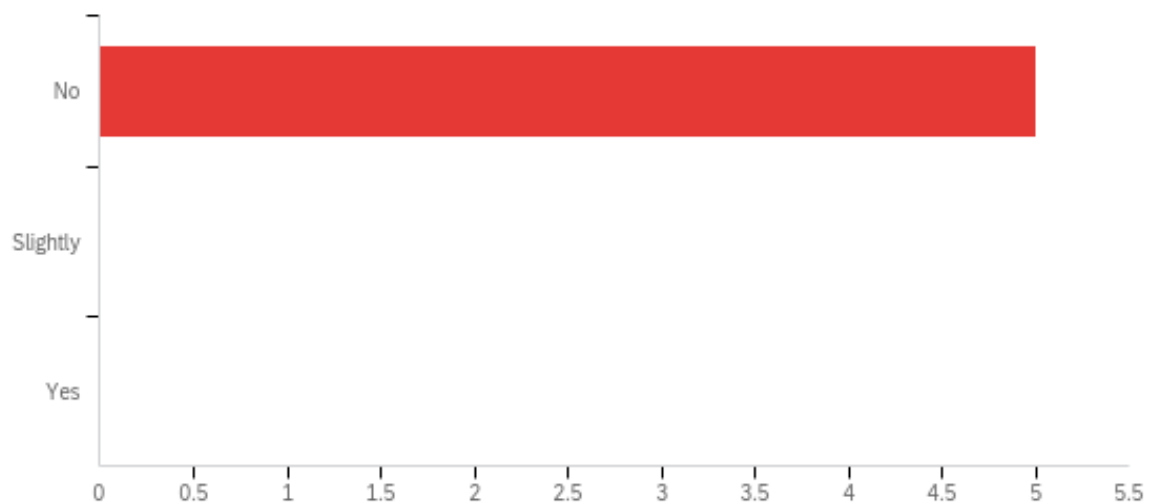
N/A

No, I think the app had all the necessary features and was not overwhelming

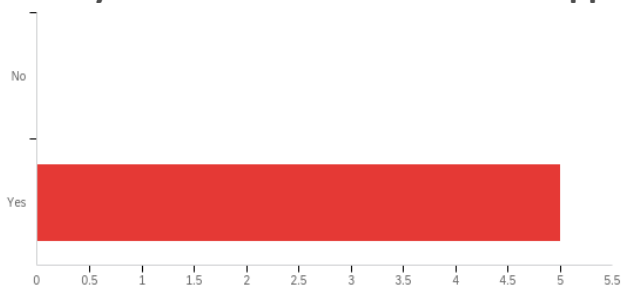
Not that I was aware of

No

Q7 - Did you experience any issues or bugs while using the app?



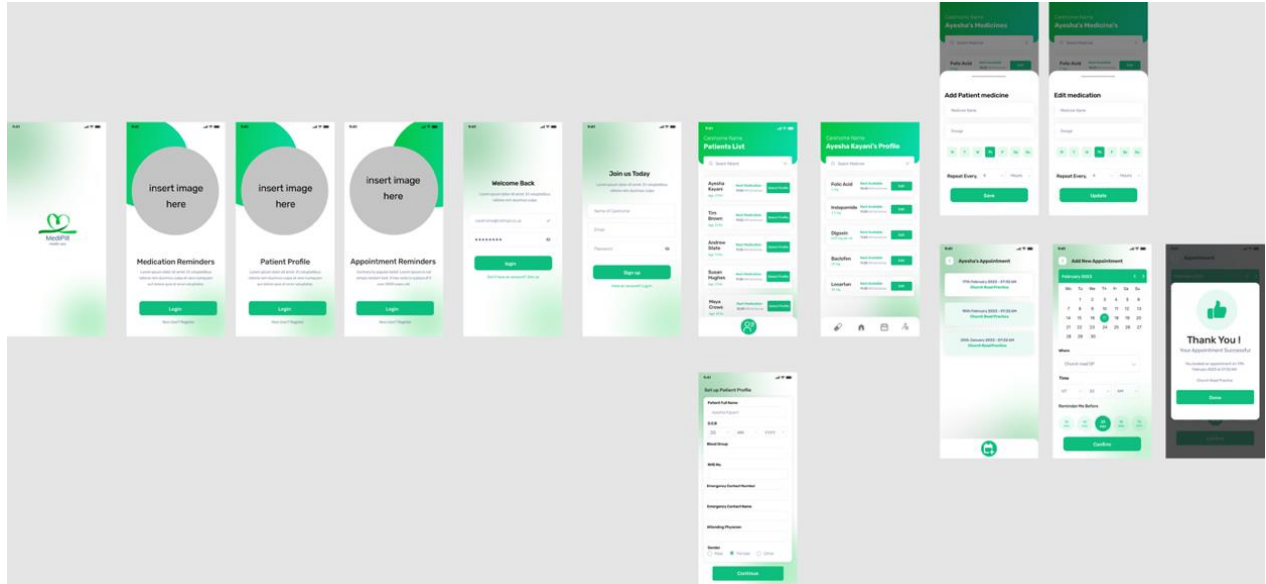
Q8 - Would you recommend the MediPill app to others?



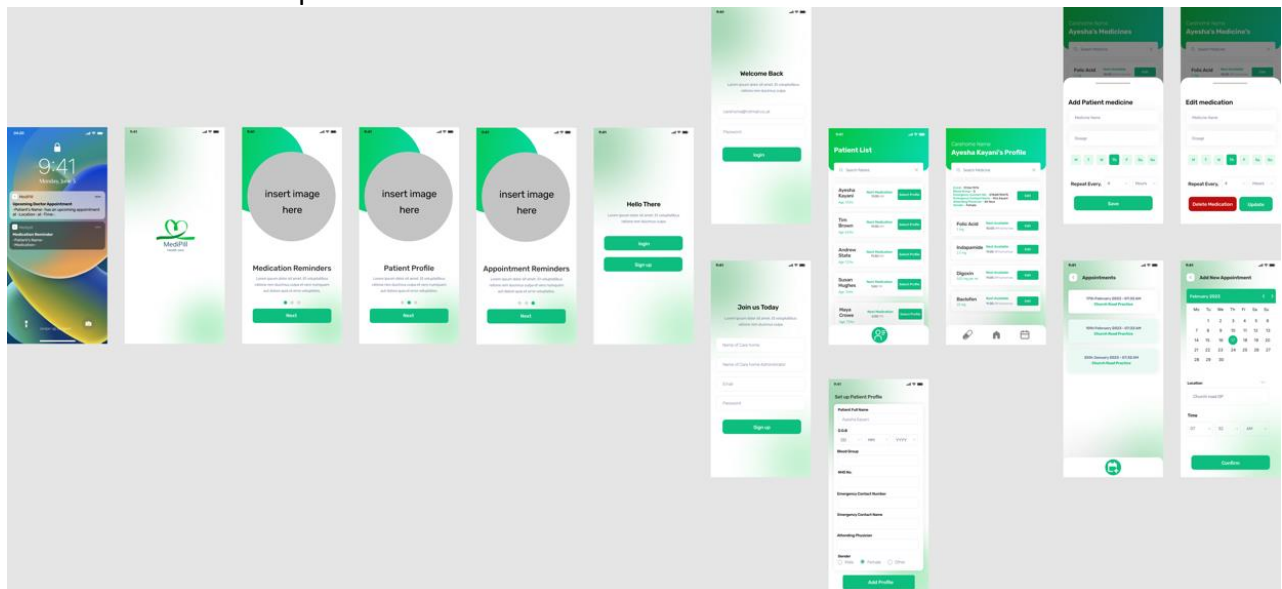
(APPENDIX I) – User interface & User Experience (UI & UX)

The feedback I gathered from the surveys led to alterations being made on the UI design with a view of enhancing user experience. For example, I restructured the navigation menu with descriptive labels that make it easier to use. Some additional features were added to increase functionality and streamline the user's workflow.

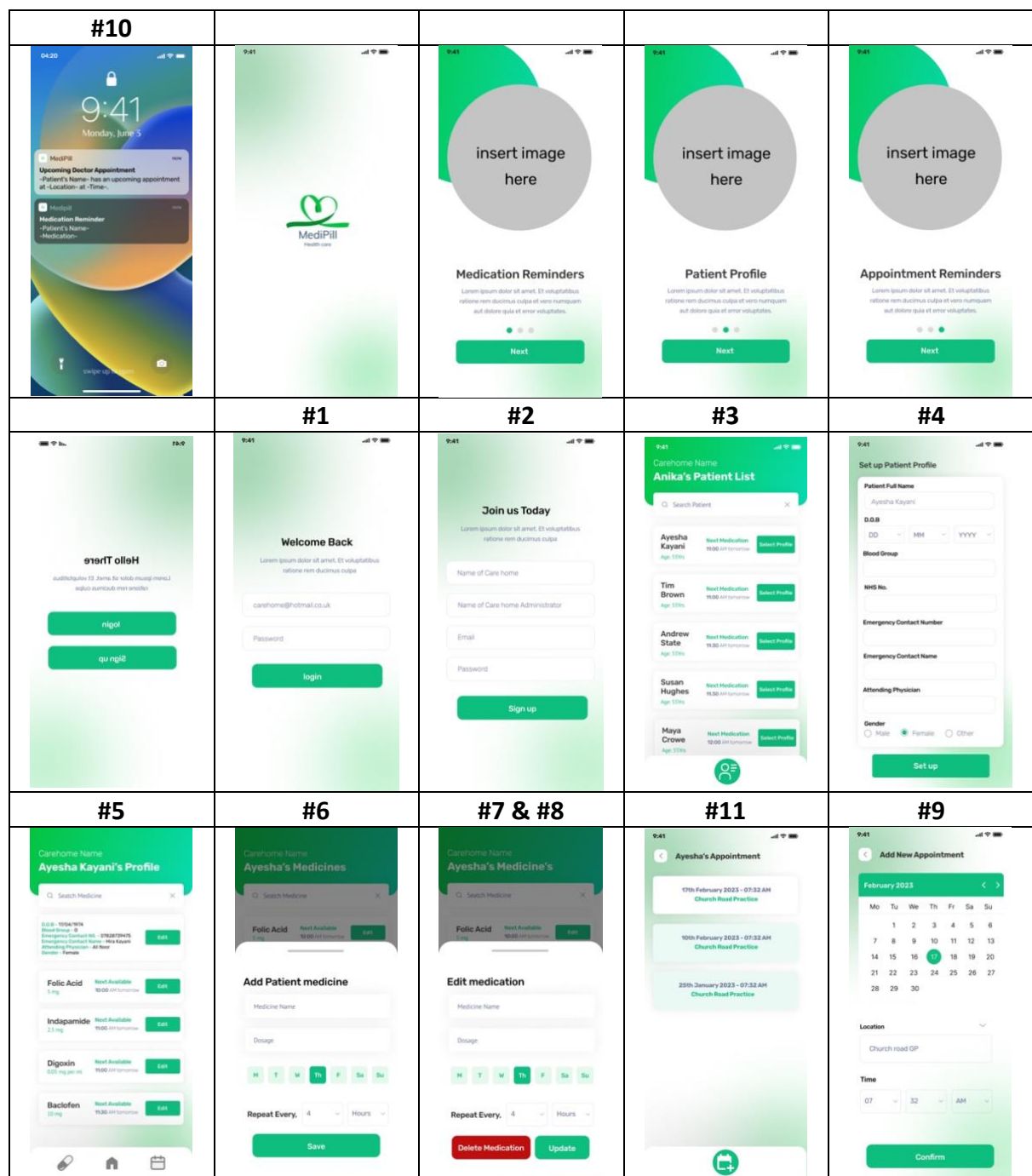
Version 1



Final Version – UI Template

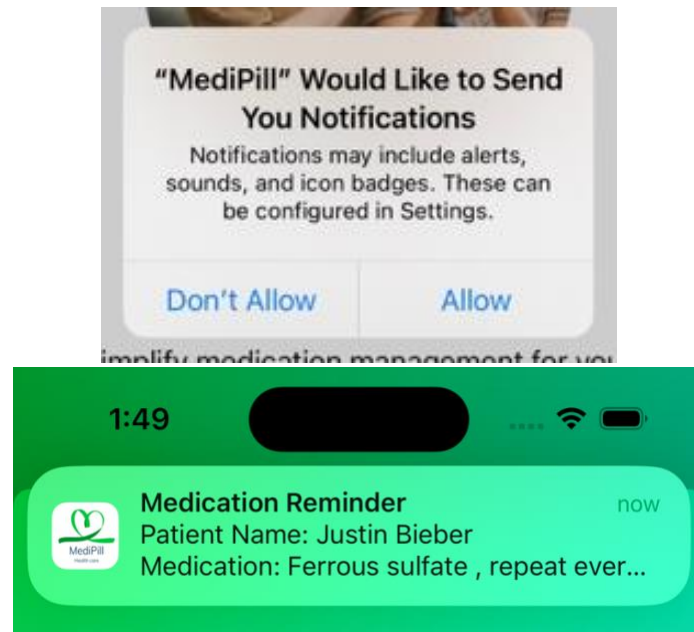


Final Version - UI Template with Use Case Numbering



(APPENDIX J) – Final Product

Notifications

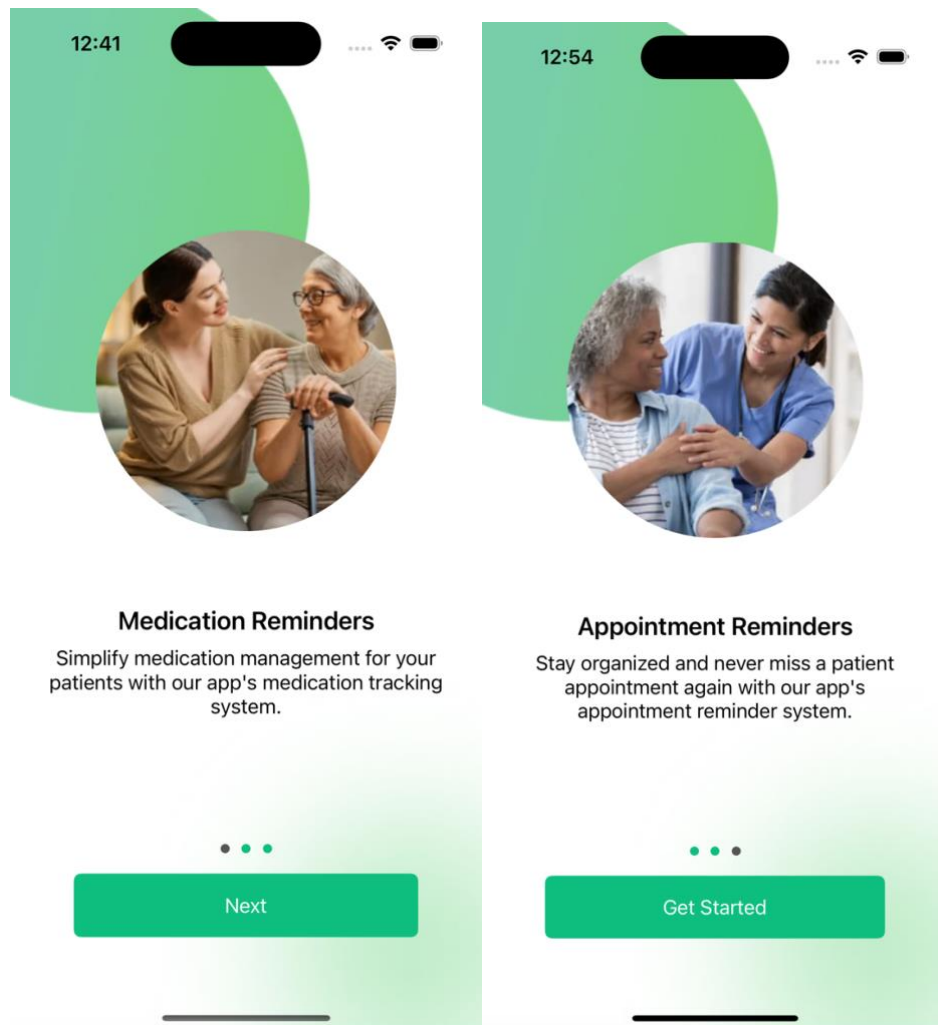


```
// Check if notifications are allowed for the app
func isNotificationsAllowed() -> Bool {
    let center = UNUserNotificationCenter.current()
    var isAllowed = false
    // Get the notification settings for the app
    center.getNotificationSettings { (settings) in

        if(settings.authorizationStatus == .authorized) {
            print("Push notification is enabled")
            isAllowed = true
        } else {
            print("Push notification is not enabled")
            isAllowed = false
        }
    }
}

return isAllowed
```

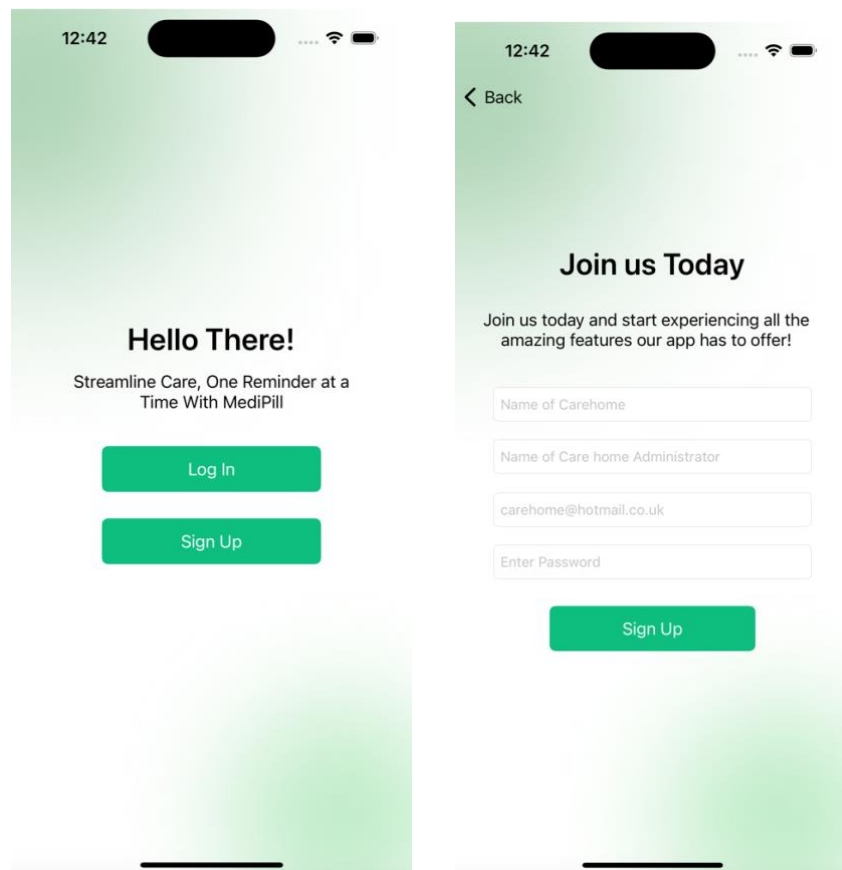

Onboarding Screens



```
// Create onboarding slides and set up the collection view
```

```
slides = [  
    OnboardingSlide(title: "Medication Reminders", description: "Simplify medication  
management for your patients with our app's medication tracking system.", image:  
imageLiteral(resourceName: "1.png")),  
    OnboardingSlide(title: "Patient Profile", description: "Empower your healthcare  
practice with our app's patient profile management system. Easily add and store  
comprehensive information for all your patients.", image: imageLiteral(resourceName:  
"2")),  
    OnboardingSlide(title: "Appointment Reminders", description: "Stay organized  
and never miss a patient appointment again with our app's appointment reminder
```

Login / Sign Up Screens



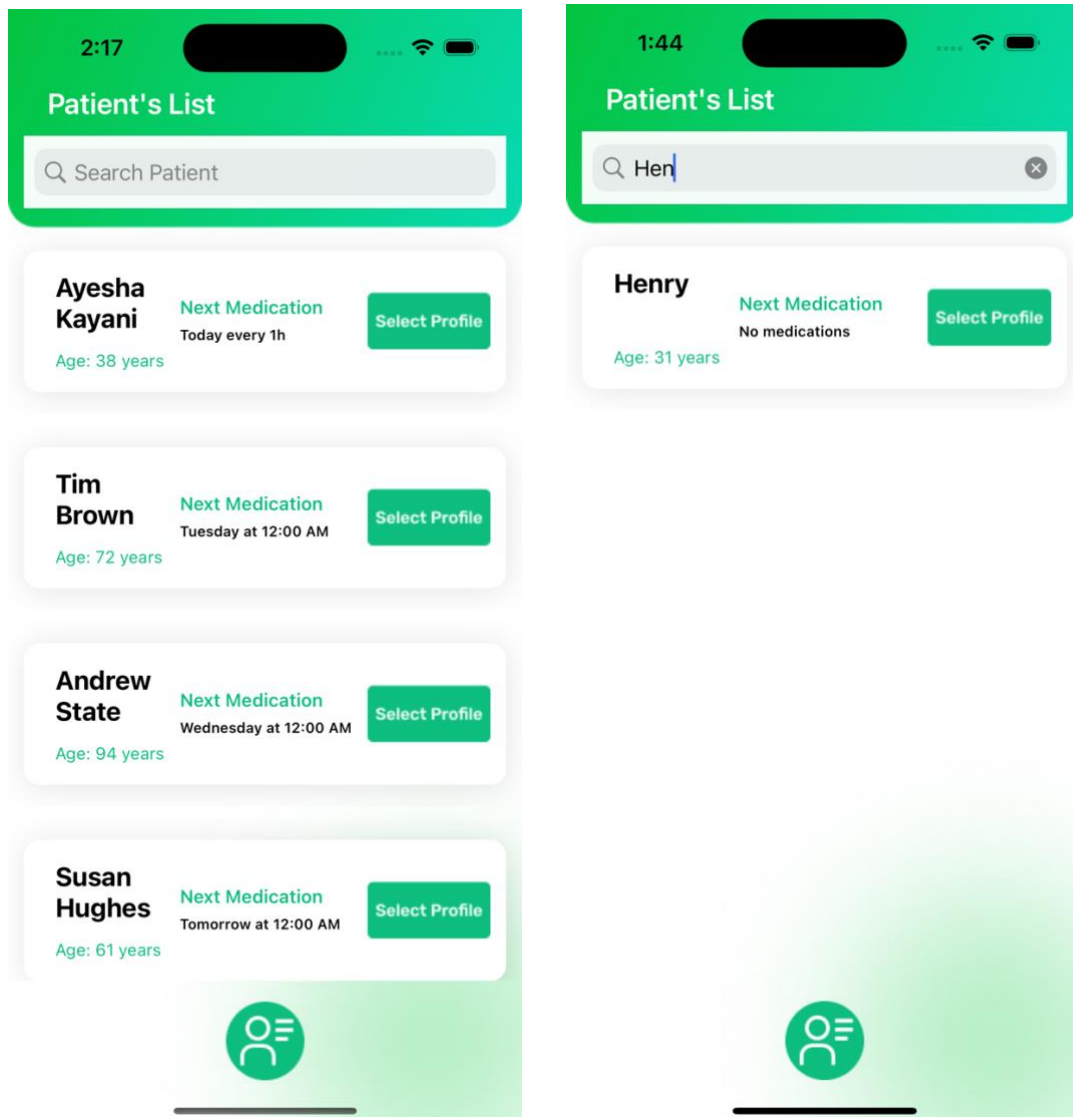
```
// Called when signup button is clicked
@IBAction func signupClicked(_ sender: UIButton) {

    // Get email and password from text fields
    guard let email = emailTextField.text else {return}
    guard let password = passwordTextField.text else {return}

    // Get optional values from text fields
    let adminField = adminTextField.text ?? ""
    let nameField = nameTextField.text ?? ""
    view.showActivityIndicator()

    // Use FirebaseService to create a new user
    FirebaseService().createUser(email: email, password: password, careHomeName:
nameField, adminName: adminField) { success in
        self.view.hideActivityIndicator()
    }
```

Patient List Screen



```
// Extension to handle search bar functionality
extension PatListViewController: UISearchBarDelegate {
    // This method is called whenever the search bar text changes
    func searchBar(_ searchBar: UISearchBar, textDidChange searchText: String) {
        let text = searchText.trimmingCharacters(in: .whitespacesAndNewlines)
        guard !text.isEmpty else {
            // If the search bar is empty, reset the filtered data and reload the collection view
            isSearching = false
            filteredPatients = []
            patientListCollectionView.reloadData()
        }
    }
}
```

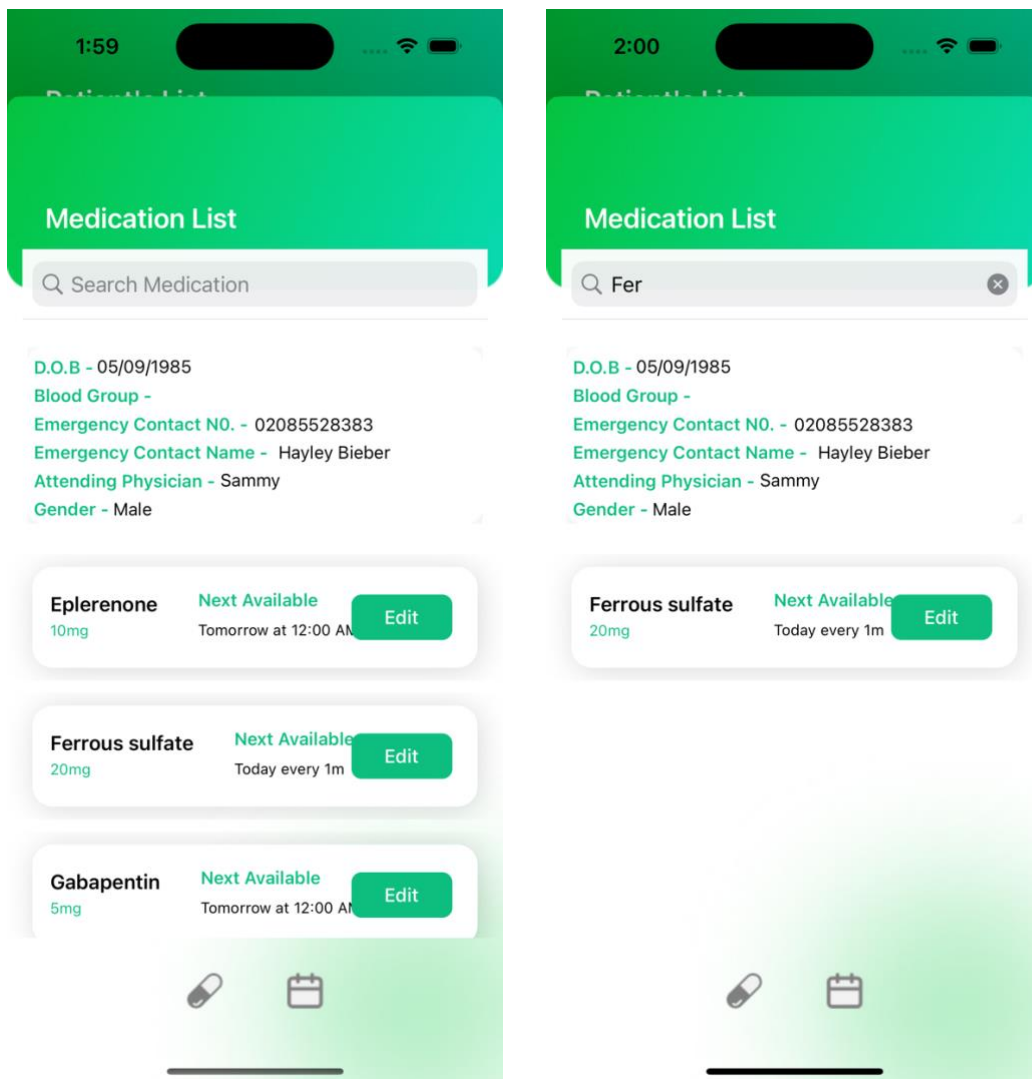
Set Up Patient Profile screen.

The image displays two mobile app screens for setting up a patient profile. Both screens have a green header with a back arrow and the title 'Set up Patient Profile'. The left screen shows a form with the following fields: 'Patient Full Name' (Justin Bieber), 'Gender' (Male/Female), 'D.O.B' (May 14, 1985) with a date picker open showing months from February to August and years from 1982 to 1988, and 'Attending Physician' (Hayley). The right screen shows the same form with different data: 'Patient Full Name' (Ayesha), 'Gender' (Male/Female), 'D.O.B' (May 14, 2023), 'Blood Group' (AB), 'NHS No.' (21412353454213), 'Emergency Contact Name' (Henry), 'Emergency Contact Number' (Sam), and 'Attending Physician' (Fiona). Both screens feature a green 'Add Profile' button at the bottom.

```
// IBOutlets
@IBOutlet weak var DOB: UILabel!
@IBOutlet weak var BG: UILabel!
@IBOutlet weak var ECTNum: UILabel!
@IBOutlet weak var ECName: UILabel!
@IBOutlet weak var ApName: UILabel!
@IBOutlet weak var Gender: UILabel!
@IBOutlet weak var View: UIView!
...

// Public Method
// Configure the cell with the given `Patient` object.
func setup(category: PatientProfile) {
```

Patient Profile Screen

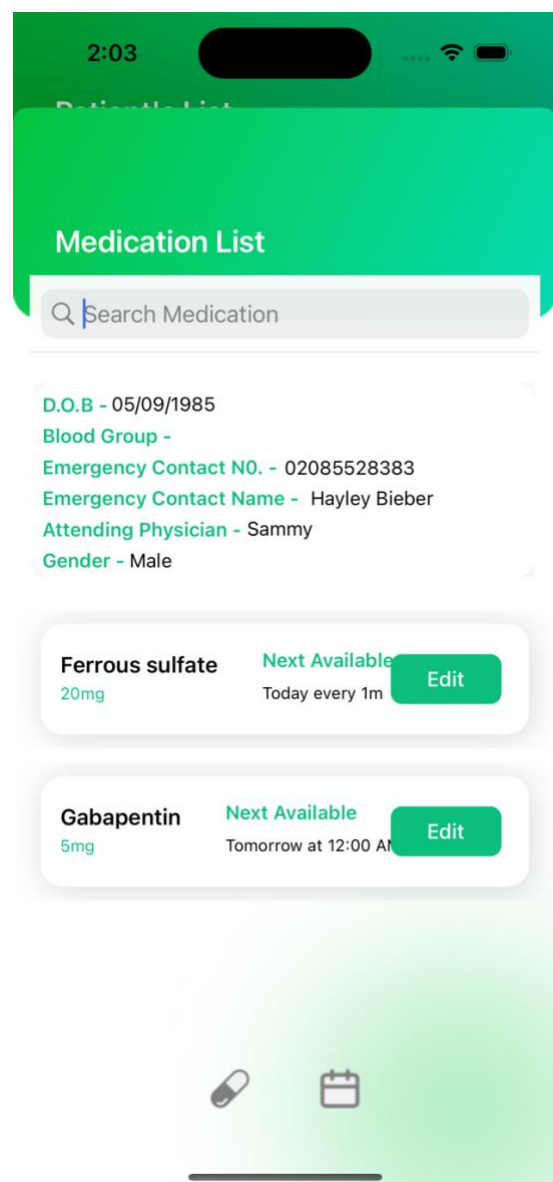
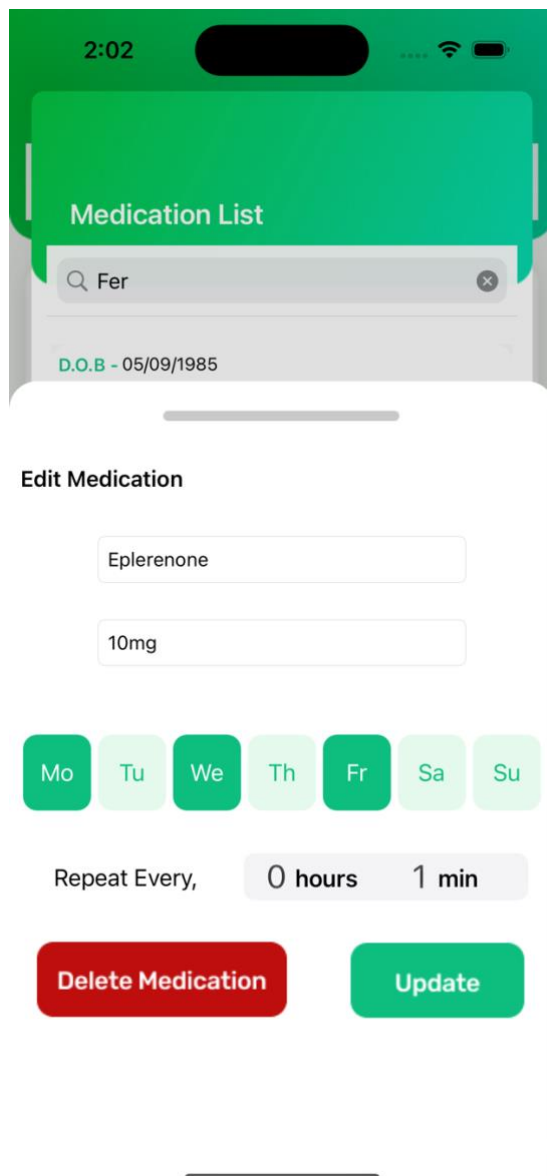


```
@IBAction func saveButton(_ sender: Any) {
    guard let patientProfile else { return }

    let medicalName = nameField.text ?? ""
    let dosage = dosageField.text ?? ""

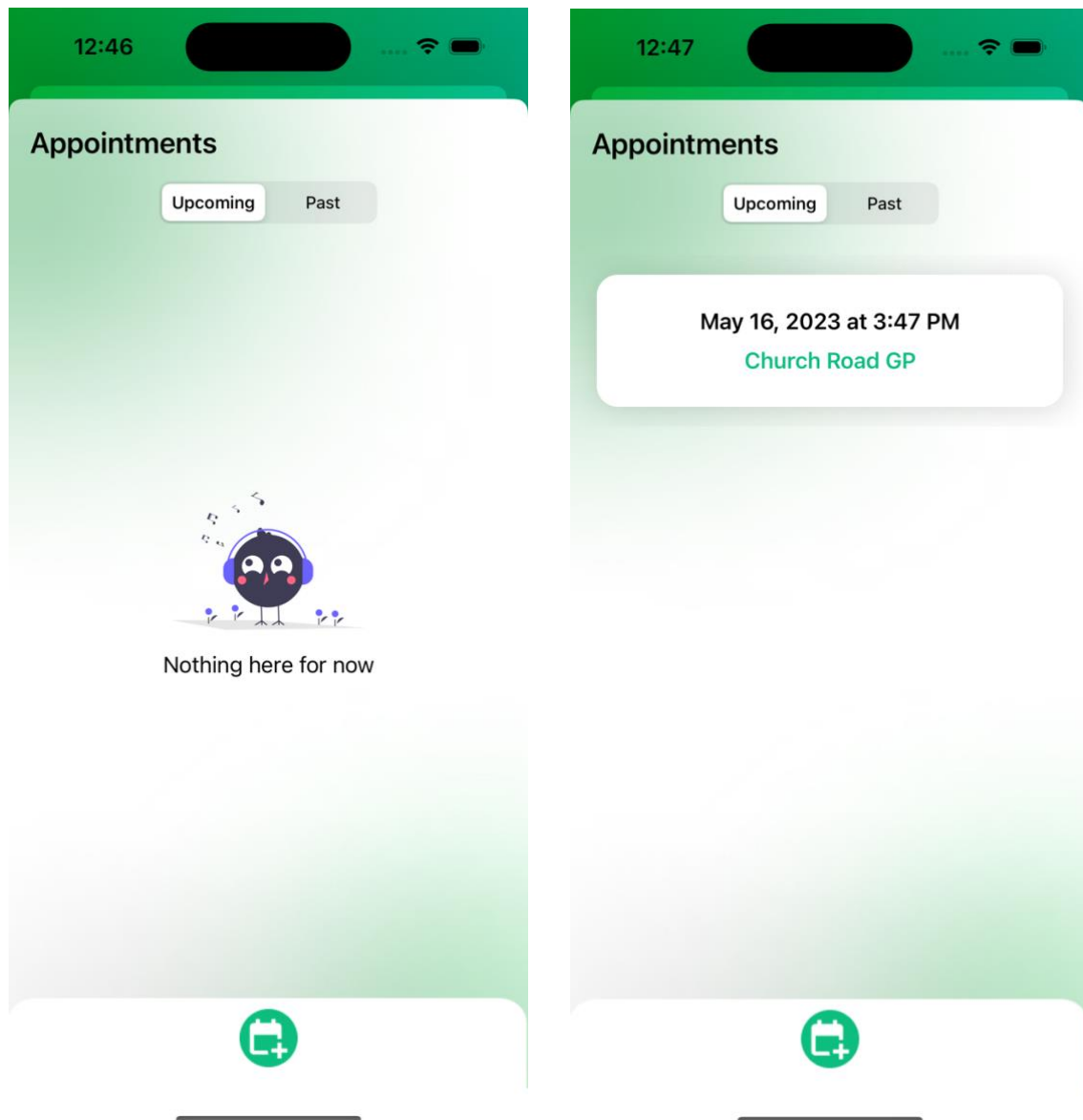
    let medication = Medication(id: UUID().uuidString, medicalName: medicalName, dosage:
dosage, weekdays: getSelectedWeekdays(), intervalTimer:
Int(repeatDatePicker.countDownDuration))
    guard !medication.weekdays.isEmpty else {
        showAlert(message: "Please select weekdays!")
    }
}
```

Edit/Delete Medication Screens



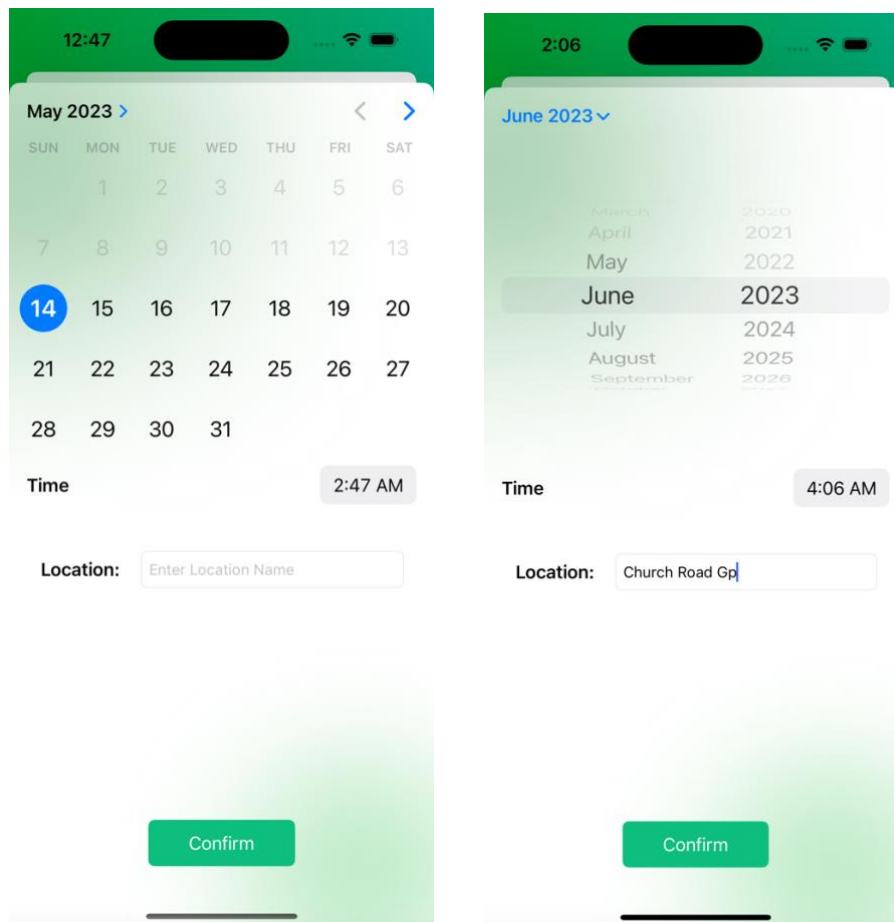
```
// Calls this if you want to remove a medication from a profile
private fun removeMedication(patientId: String, medication: Medication) {
    FirebaseService().removeMedication(toPatient: patientId, medication: medication) {
        success in
            if success {
                // do something after success
                self.delegate?.medicationDeleted(id: medication.id)
                self.dismiss(animated: true)
            }
    }
}
```

Add Appointment Screen



```
extension AppointmentViewController: CalendarVCDelegate {  
    func appointmentAdded(appointment: Appointment) {  
        if patient.appointments == nil {  
            patient.appointments = [appointment]  
        } else {  
            patient.appointments?.append(appointment)  
            patient.appointments?.sort(by: {$0.timestamp < $1.timestamp})  
        }  
        configureAppointments()  
    }  
}
```

Calendar Screen



```
private func scheduleAppointmentNotification(appointment: Appointment) {
    let date = Date(timeIntervalSince1970: TimeInterval(appointment.timestamp))
    let notificationBody = "\(patient.fullName ?? "") has an upcoming appointment at
    \appointment.location) at \getTime(from: date))"
    let content = UNMutableNotificationContent()
    content.title = "Upcoming Doctor Appointment"
    content.body = notificationBody
    let appointmentInterval = TimeInterval(appointment.timestamp - 3600)
    let trigger = UNTimeIntervalNotificationTrigger(timeInterval: appointmentInterval -
    Date().timeIntervalSince1970, repeats: false)
    print("ZZ1", trigger.timeInterval)
    let request = UNNotificationRequest(identifier: appointment.id,
    content: content, trigger: trigger)
```