An Electronic Reminder for Taking Medications

Ben Gallard-Bridger
GAL19693078

19693078@students.lincoln.ac.uk

School of Computer Science

College of Science

University of Lincoln

Submitted in partial fulfilment of the requirements for the
Degree of MComp Computer Science

Supervisor Dr. Mark Doughty

May 2022

Abstract

In healthcare, there are over 3 million medications dispensed daily (NHSBSA, 2021). With people taking up to eight different medicines each day, patients can become unlikely to adhere to their medication regimes. Decreasing the likelihood of this occurring is the foundation for this project.

This project document provides insight into the successful design, development, and testing of an android application used to provide notifications to a user reminding them about when to take medication(s). This document also shows an analysis of current literature and solutions which provided insight into the development of the application. Black-box testing, in addition to beta testing, was conducted to analyse the application. The study involved four participants who provided quantitative data that suggested that receiving an electronic notification improved their adherence to their medication regime.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Scope and Rationale

One of the main problems when taking medications is remembering to take them. Up to 50% of people are not adherent to long-term medications (Marcum et al., 2013). Forgetting to take medication can lead to a decline in their well-being or lead to the patient requiring treatment for longer. In addition, people may take their medication later than they are meant to, meaning the medicines can be less effective.

The development of a mobile application was performed as part of the project to solve this issue. This application allows the user to receive reminders about the specific medications they take. It further allowed the patient to track what medicines they take, need to take at that time, and have taken in the past. With the intended aim that while using the application, a person would be more likely to take their medications as intended. With time, this would incur an overall increase in health for the user, along with decreasing costs for the NHS to assist with future complications (Mashta, 2010).

This document shows a review of current literature alongside a review of the methodology of the distinct aspects used throughout the project. In addition to this, this paper describes the artefact's design, development, and evaluation. Furthermore, the document details an analysis of data collected, showing the effectiveness of the application. Finally, this document concludes the project and analyzes how the project could have been better in hindsight.

1.2 Aims and Objectives

The aim of the project was to create a piece of software that allows the user to track the information about their medication, allowing them to receive reminders about their medicines and allowing the users to track what medications they have taken in the past. The objectives of this project were:

1. To design the User Interface using Balsamiq wireframes, showing how the application will be used on all prospective pages. This objective changed during the design, using a different piece of software (Microsoft Office PowerPoint, detailed in section 3.3), which performed similarly with extra functionality.

2. To develop the application for an Android device, using Android Studio (Detailed in 3.3). The application will allow for the user to track and get notified of current medications needing to be taken.

3. To evaluate the application on a variety of different devices, with different people. To receive 5-10 reviews of the application and calculate its effect on medication non-adherence.

Chapter 2

Literature Review

Polypharmacy is where a person takes multiple medications, whether that be for multi-morbidity or single ailments. As people age, the likelihood of having polypharmacy increases along with the number of medications. With about 40% of people over the age of fifty taking more than two medications and which increases to over 90% of those over 90 (Duerden and Payne, 2014; Department of Health & Social Care, 2021; Charlesworth et al., 2015). As the number of medications, a person takes in-creases, the likelihood that they would be non-adherent to a medication regime also in-creases (Marcum and Gellad, 2012).

Non-adherence to a medication regime can be from a variety of reasons. There are diverse types of non-adherence to a medication regime, such as: nonfulfillment, nonpersistence or nonconforming (Marcum and Gellad, 2012). Nonfulfillment is where the patient does not 'fill' their prescription. "Nonpersistence is when a patient stops taking medication after starting the medication, without being told to do so by a medical professional" (Marcum and Gellad, 2012). Nonconforming would be where the patient does not take the medication as prescribed, such as: taking incorrect doses, skipping doses, or taking doses at incorrect times. Non-adherence can be caused by a variety of different issues, some examples could be: a patient forgetting to take their medication (Nonpersistence/Nonconforming), forgetting to pick up medications (Nonfulfillment), forgetting how to take their medications (Nonconforming), not wanting to take/fill medication (Nonpersistence/nonfulfillment), feeling better so they don't continue to take medications (Nonpersistence) or cannot afford the medication (Nonfulfillment). (Marcum and Gellad, 2012)

Around 50% of people are non-adherent to their medication regime (Brown and Bussell, 2011; Marcum et al., 2013; Sahm et al., 2009). This can lead to a myriad of issues for a patient (National Health Service UK, 2021). For patients with more severe issues, taking medication can be the difference between life and death. For those with less severe ailments it could mean staying on medication for longer or experiencing worse symptoms (National Health Service UK, 2021; Brown and Bussell, 2011). This could lead to financial issues from having to miss work for longer and paying for more medication. Furthermore, this extra care needed for patients can lead to more resources being needed in the NHS such as funding and staffing. It is estimated that around £200m of prescribed medications which are given to patients are unused, either left in their homes or returned to pharmacies (Mashta, 2010). Including the other costs related to non-adherence, such as staffing, it is estimated that over £500m per annum could be saved if all people were adherent to medication regimes (Trueman et al., 2010). Furthermore, in America these preventable costs are about $100 Billion (Marcum et al., 2013).

There are a multitude of ways that someone can receive help to take their medications. This can be from active help from someone to devices such as dosette boxes or dispenser machines (Specialist Pharmacy Service, 2020; National Health Service UK, 2021).

Help from someone would require someone, usually a nurse, to provide reminders and in some cases ‘dispense’ the medication to a patient. This can be costly, due to the amount of time a nurse would need to be around someone. In contrast, it can be a more effective way of ensuring that someone was receiving the correct treatment for their ailments. Other forms of help from people would include having people ringing them to remind the patient to take their medications and what medications to take. One issue with have people help is human error, there is always a possibility that the person helping could forget what medications are needed or when medications are needed, which would lead to forgetting to inform the patient to take their medications.

Dosette boxes are the use of organized boxes with sections for each time of day for a week. In each section is the prescribed tablets for that day of the week and time of day. A pharmacist fills these individually per person. This ensures that a person knows what time of day that they should take their tablets, what tablet to take and how many. However, this method only allows for helping with tablets and do not provide an active reminder for these people to take their medications.

Dispenser machines are also another option for those who struggle to take medication. These machines give an auditory notification to notify the patient to take their medication(s). An auditory notification allows for someone to passively recognize that they need to take their medication(s). Another advantage for this type of intervention is that it can track when and what medications have been taken. However, this type of intervention can only be used with tablet medications and can be very costly for the user, being anywhere between £100 and £3000.

From looking at these options we can see that there are a few issues with the current solutions for medication non-adherence. These being, not having active reminders to incur adherence, being costly and only being able to be used with tablets.

To make an improvement on current solutions, we need to create something which allows to have active reminders, low-cost to the users and are general purpose for all medications. One idea to solve this would be a reminder application. A specialized reminder application for medication could allow for specific doses to be attached to each reminder, allowing for all types of medication rather than just tablets. An application also could be distributed through an app store for free along with the ability to use the devices' notification system to provide reminders. This alone would solve the fundamental issues of the previous solutions to the problem.

Electronic reminders, through texts and notifications on a mobile phone can improve the likelihood that a task is completed or initiated. For short term, patient adherence using medication electronic reminders have increased, in long term this may also be possible (Vervloet et al., 2012). A medication reminder application has been seen to allow for 92.5% of elderly people to stay adherent to their medications over a two-week period (Shade et al., 2019). In addition to this, 60% of people are willing to receive a message prompt to remember medications (Sahm et al., 2009).

Reminders can be given to a person using multiple senses at the same time (McGee-Lennon, 2012). Out of the five senses, a mobile phone can stimulate three senses: visual, auditory, and tactile. The visual stimulation can come from a use of an object on the users' screen, an auditory stimulation can come from the use of a tone through the device's speakers, along with the tactile stimulation using the device's vibrator. Using multiple senses to provide a notification or reminder to the user, allows for a higher response from the user, with an added benefit of allowing people who are deprived from using all their senses to respond to the notification.

The main purpose of my project is to help solve these issues, using a mobile application, with the aim of improving the general health of the population. The application will help the users take their medication through active reminders, using the notification system on their device. With it also being a mobile application rather than a device, it means that the app could be made free for anyone to download, meaning the cost would only be to have a working mobile phone.

There are other features which the application could employ for ease of use. One feature could be the ability to look through the medication which need to be taken for that day, so that if a person is needing to take medication with them, they can. Furthermore, medications can have long and complex names, so the application will employ a scanning facility, using either a 2d data matrix or barcode to recognize which medication is needed to be added. This reduces the likelihood that someone will misspell or enter the erroneous information about their medication, leading to a reduction in wrong medications taken.

Although the main purpose of my project is to help reduce medication non-adherence, there can be other benefits to using an application rather than current, more common interventions. One example of a benefit is that it would reduce the amount of plastic ending up in landfill (Department of Health & Social Care, 2021). Some blister packs which are decanted into dosette boxes or used in medication dispenser machines, use PVC as the plastic base. PVC is a non-recyclable plastic and therefore must end up in landfill rather than being used.

To conclude, polypharmacy is common in healthcare, which can cause medication non-adherence. Medication regime non-adherence can lead to issues for the patient, which in turn can cause more stress for healthcare systems, such as the NHS. There are current systems in place to help with polypharmacy and allow for a higher adherence to a medication regime, however these systems can be expensive or less effective for reminding those who forget medication. An improvement on these systems can allow for patients to have a general increase in health, leading to less of a strain on healthcare systems.

Chapter 3

Methodology

3.1 Project Management

In this project there were three main stages:

1. Design
2. Development
3. Testing

3.1.1 Project Planning

The project was planned through the waterfall method, it was chosen as it allows the tasks which need to be completed during the project to be clearly defined and broken up to allow for an accurate timeframe. Although few tasks needed to be performed synchronously, especially during the development phase, this method encouraged a focus on a specific task for a specific timeframe, giving a deadline for each individual task. In addition to this, the waterfall method also supports the use of a simple graphic known as a Gantt chart (as shown in Table 3.1), to plan, track and visualize the project.

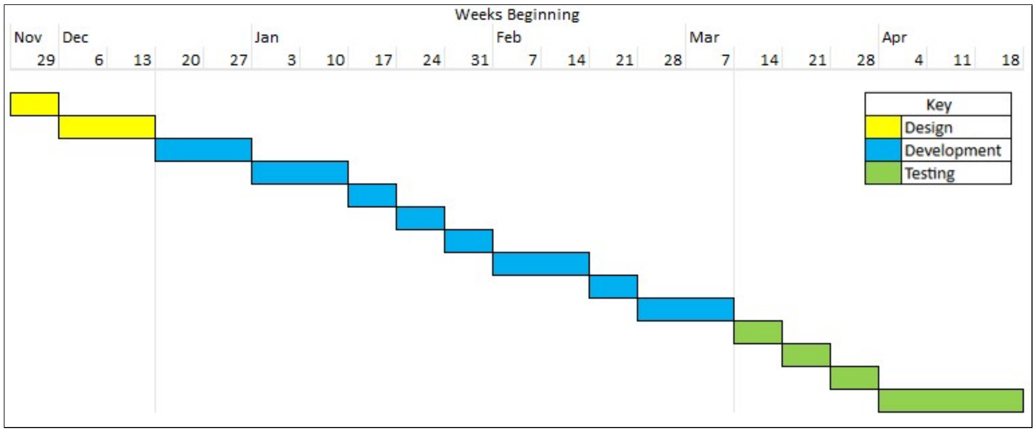


Figure 3.1: Gantt Chart of Project Stages

Table 3.1: Table of Project Tasks

Task	Duration	Start Date	End Date
Design			
User Interface	1 Week	29/11/2021	06/12/2021
Class Inheritance	2 Weeks	06/12/2021	20/12/2021
Development			
User interface	2 Weeks	20/12/2021	03/01/2022
Scanner	2 Weeks	03/01/2022	17/01/2022
Data Retrieval From API	1 Week	17/01/2022	24/01/2022
Data Storage	1 Week	24/01/2022	31/01/2022
App Security	1 Week	31/01/2022	07/02/2022
Encryption	2 Weeks	07/02/2022	21/02/2022
Reminder System	1 Week	21/02/2022	28/02/2022
Tracking System	2 Weeks	28/02/2022	14/03/2022
Testing			
Black Box Testing	1 Week	14/03/2022	21/03/2022
Create Survey	1 Week	21/03/2022	28/03/2022
Gather Participants	1 Week	28/03/2022	04/04/2022
Perform Survey	3 Weeks	04/04/2022	25/04/2022

As shown in the Gantt chart and table above, each task was planned and separated by the three phases. Performing such planning allowed for a decomposition of the phases, and more accurate time management for each phase. Although the tasks were planned for specific dates, if a task were completed earlier, the following task would be initiated earlier, subject to possibility. This gave some flexibility to spend longer on some tasks when needed, without compromising the overall timeframe of the project.

3.1.2 Risk Assessment

As part of the planning for the project, a risk assessment was performed to allow for mitigation of events which would be detrimental to the outcome of the project.

Table 3.2: Risk Assessment Table

Risk	**Likelihood**	**Impact**	**Management**
Not Enough Participants	Low	High	Recruit participants early and recruit participants from groups
COVID-19 Lockdown	Low	Low	Use online forms for testing and use an online method of allowing participants to download the application.
Medication mistake - too many / not enough / missed dosage	Low	High	Use global times to check when medication should be administered, checking if a previous dose has been logged.

Completing the risk assessment made it evident that the likelihood of an event occurring was low. Although it was unlikely that an event would occur, it was still beneficial to adjust the plan to further reduce the likelihood. During the evaluation of the project, there was a lack of participants in the study, however, making an adjustment to the representation of results mitigated the issue. This is detailed in Section 3.4.

3.2 Software Development

There are multiple methodologies used when performing software development. For this specific project, the artefact was developed using an agile methodology. Agile development allows for the reiteration of sections, if newer and better ideas arise at the time, rather than locking in features and tasks at the beginning of development. Agile is an overarching term used for many iterative methodologies in software development, only giving specific ideas rather than set rules, for this reason specifically the agile framework scrum was used.

Scrum uses sprints to allow for multiple iterations of the same features. A sprint is a time boxed event, “almost never longer than thirty days”, consisting of work performed in iterations (Andrei et al., 2019). Sprints are focused on implementing a single feature, without interruption or intervention.

Another methodology or framework which could have been used is waterfall. The waterfall methodology focuses on fixed phases, with each phase being completed in a specific order. These phases are set before the project starts and assumes that no changes will need to be made in the future. When talking about software development, this is rarely the case (Andrei et al., 2019).

Generally, in terms of software development, features to be developed, may require the feature before to be developed, but more commonly, features can be developed in any order. Specifically in this project, the features to be developed, have no need for a specific order or time frame, but more require the flexibility of scrum. For this project, each task in the implementation stage, as shown in the Gantt chart (Figure 3.1), is its own sprint. The tasks for the software implementation are not required to be in a specific order, so the sections in that phase are not constricted to the times set, and therefore can be swapped, the main constricting items in that section are the starting and ending dates for the development.

3.3 Toolsets and Machine Environments

This section will detail the specific toolsets and machine environments which were used to produce the project.

3.3.1 Android Version 8.0+

For the operating system that the application was developed on, an android device of version 8.0 or higher was selected.

The first reason for an Android device is that in 2021, over a billion Android phones were shipped out to users, with an estimated three billion active users. Three billion people are a sizable portion of the global population, meaning that more people would be able to access the application globally. The reason to ensure that the application is compatible with android version 8.0 and above is that most current android devices run android version 9.0 or above (Rinaldi, 2021). Ensuring that the application uses functionality available on android 8.0 allows a broader community of people to access the application.

The opposing option for the development of the application would have been to develop on IOS. Choosing against IOS for the project was due to the amount of people using an IOS device when compared to an android device, along with the current average price point of the differing devices. For IOS there are only one billion users actively, which is a third of the current active users for android devices (Kastrenakes, 2021). In addition to this, an IOS device is on average 2.7 times more expensive than the average android device, making an android device accessible to a wider portion of the population (Chowdhry, 2015). With the application being a proof of concept rather than an end product, the accessibility of the application is not a factor as the only people using the application will be from the study.

3.3.2 Android Studio

With the application being developed on an android device, there are many options which can facilitate development. The de facto standard of android development as seen by many is Android Studio (David, 2019).

Furthermore, being the de facto standard for development on the specific operating system, it means that there are many tutorials and documentation about specific features which are required for the development of the application.

An alternative for android development is visual studio, by Microsoft. Android development in visual studio is provided through C++ and further allows for cross-platform development. This would allow for the application to be implemented on both Android and IOS, consequentially allowing for a larger amount of people to gain access to the application. However, to load applications onto an IOS phone, there is a need to use an application with the device connected, or be part of the Apple Developer Program (Apple, 2022). Loading the applications in-person would spoil the anonymity of the study and the Apple Developer Program incurs a financial charge. In comparison, an android application can be installed through downloading a file. These reasons, along with Android Studio being specialized for android development, lead to the conclusion that for this project, that Android Studio should be used for the development.

3.3.3 GitHub

GitHub is a tool used by many programmers, to implement a form of version control, with the ability to store the development code online. This software was used to implement version control, and subsequently allowed for the storage of the application files in the cloud, providing the ability for the access of these files from any device with internet access.

To perform the version control, the program, along with all the code files, were stored on an online repository which was accessed through the GitHub desktop application. Each change in the files was able to be uploaded to the online repository, showing the differences between each time the code was submitter or ‘committed’ to the repository. Using GitHub enabled visibility on each change, allowing for the ability to revert changes which were made.

An alternative for the storage of the code, could have been to use a cloud-based storage solution such as Google Drive or Microsoft OneDrive. These options allow for a continuous synchronisation of the data and would also support accessing the files from any device. However, these options do not implement version control, meaning that if an error was made and saved, it could not be reverted without having to manually go through each file and reverting the change. These reasons made it clear that GitHub was a more suitable solution for the storing of the application code.

Table 3.3: Feature Comparison Matrix for Storage Methods

	Storage Methods			
Features	**Google Drive**	**OneDrive**	**GitHub**	**Local Storage**
Cloud Storage	Y	Y	Y	N
Version Control	N	N	Y	N
Free Max Size	15GB	5GB	100GB	N/a
Paid Maximum Size	100GB	1TB	100GB	N/a

3.3.4 Google Forms and Google Sheets

For the testing, both beta and study, Google Forms was used. This is a cloud-based form service, with fully customizable forms. This customizability allows the data to be collected without having any contact with the participant, and without collecting any data relating to the participant. This was beneficial due to the sensitivity of the data collected, meaning that the data in no way can be linked back to the participant.

Having a purely online service such as Google Forms, allowed for the anonymous collection of data, along with the service of automatically entering this into a spreadsheet, hosted by Google Sheets. Google Sheets is an online spreadsheet, which is stored in Google Drive. When accessing files from Google Drive, they are encrypted during transfer. This allows for the data to be private to external sources (Google, 2022).

Furthermore, as Google Sheets are stored online, they can be accessed from any device. The servers which Google Drive files are stored upon have a redundancy system implemented known as RAID (Hildebrand and Serenyi, 2021). This means that the data stored on Google Drive is highly unlikely to be able to be lost, as the data is split in such a way across multiple drives that if one drive 'goes down' for any reason, the data can be restored from the files on the other drives.

Another option instead of Google Forms and Google Sheets, would be to use Microsoft Forms and Microsoft Excel. These tools both provide the same functionality as one another, with some minor differences between the two platforms. One of the main reasons for choosing the Google platform over the Microsoft platform was due to the lack of options in types of questions (Ask Any Difference, undated).

3.3.5 Microsoft Office PowerPoint

PowerPoint is a piece of software primarily used to present 'slides' however, it has the capability to drag and drop shapes and pictures, which in turn can function as a 'wireframe' or basic design of the application. To further show how the application would work, PowerPoint also has support for shapes being a hyperlink, which can link to different slides. This can give the effect of a working application design, showing the interaction between pages.

The original option to perform the user interface design of the software, was to use Balsamiq Wireframes. Balsamiq Wireframes is a piece of software which is specialised for the creation of wireframes for different applications. PowerPoint was used in the project instead of Balsamiq Wireframes as there are many free options to view a PowerPoint presentation which would allow more people to view and interact with the design, when compared to a Balsamiq wireframe.

3.3.6 DM+D Data API by Chris Wilson

As the application will provide scanning functionality to retrieve GTIN codes for identification of medications, an API was needed to search by GTIN to find medications. This API returns information about the medication searched, along with informing if the data is not in the database. Through some research, it seemed that this API currently was the only available API for this specific use case (Wilson, 2021).

3.4 Research Methods

To perform the research, quantitative data is being collected for analysis. Performing the analysis of quantitative data, allows for an elicitation directly into whether the application made a positive effect on a persons' adherence to a medication regime. Qualitative data would not allow for this because qualitative data is based upon a person's opinion rather than objective data, which can be used for an unbiased evaluation.

The effectiveness of the application on medication non-adherence can be measured quantitatively, by having the participant report the exact number of times which they were non-adherent to their regiment before and during the use of the application. These two values, once collected, can be compared to see if there is a positive or negative trend, which can further lead to a conclusion on whether the application made a positive impact on the participants' adherence.

The type of data being collected in the survey, is ratio data. Ratio data can be categorised and ranked, whilst also being able to infer equal intervals between neighbouring data points, along with a true zero point (De LAND and Chase, 1990). As the data is coming from two separate questions it allows it to be categorised, whilst it being numerical it can be ranked. With the questionnaire asking the participant to report the exact number of times they were non-adherent to their medication regime, the difference between neighbouring data points is the same. Finally, if the participant provided a zero on either of the questions, it would signify that they were fully adherent to their medication regime, and in turn never forgot to take medication properly during the respective period.

The original intentions to represent the results of the study, was to use a histogram. With the two categories of data, before and during (using the application), being of the same type and having a limited range, they can be placed on the same graphs with translucent bins and differing colours. This would allow for a direct contrast between the data retrieved from the two questions. If the histogram containing the data about whether the participant was adherent to their regime during the use of the application has a lower average than its counterpart, thus implying that the application had a positive effect on the participants medication adherence. However, using a histogram to represent the data was not the right choice for the project, due to a limited number of participants. As a result, an arithmetic mean of the two categories of data was calculated to find the average difference between the results from both questions, which is slightly indicative of the original plan.

Chapter 4

Design, Development and Evaluation

4.1 Software Development Projects

4.1.1 Requirements Elicitation, Collection and Analysis

To elicit the requirements of the application, research was performed into similar solutions to the same problem. Some examples include the Pill Reminder and Med Tracker app by Medisafe® and the MyTherapy Pill Reminder by MyTherapy®. In addition to researching current solutions to the problem, researching current literature around the problem topic was performed and provided some insight into how the application should work.

Looking at examples and current literature surrounding the topic, common themes were evident, some of which would be key requirements/features for the application:

1. Gives active notifications to the user
2. Allows the user to see all their medications
3. Allows the user to see when they last took a medication
4. Allows the user to enter a medication

There were also secondary requirements which although were not required to create the application, would have provided better functionality. Such as:

5. Scanning medication packets
6. Showing medication to be taken in the future

4.1.2 Design

There were two sections to complete the design of the application. These were: The user interface (UI) and the class structure.

User Interface

The design of the user interface was created in Microsoft Office PowerPoint. This was done through creating 'slides' for each page, using hyperlinks on 'shapes' to navigate back and forth through the slides as though it was a working application. This provided an overall design of the application and how the navigation through the application would work. This allowed for a form of testing for the pages and allowed for multiple iterations of the pages and the flow between them, overall giving a better design in the end.

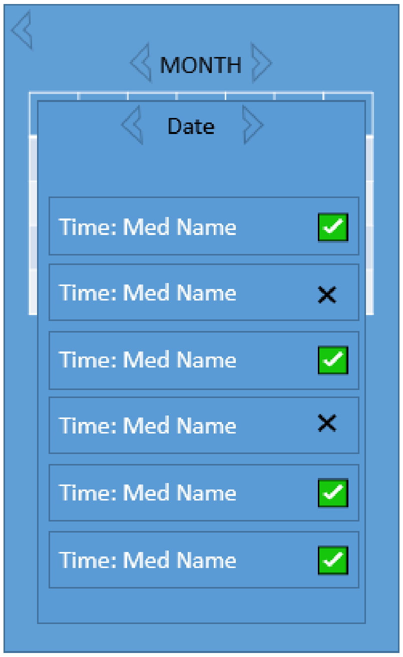


Figure 4.1: Example of a page design on PowerPoint

As shown in the example in the above figure (Figure 4.1), it can be clear to see what the page will look like when it is fully developed. Seeing the design before it is developed allows for quick alterations to be made, in case of the design being inefficient or unnatural for the user. Changing the interface in the design rather than during development, also decreases on development time. The design of all the pages can be seen in Appendix B.

Class Structure

To design the class structure, research was first performed to elicit what information could be stored about medication. Finding this information, an inheritance hierarchy was created, which showed how different classes would inherit each other along with the values to be stored by each class. This allowed for a direct visual interpretation of what was needed for the differing classes during development.

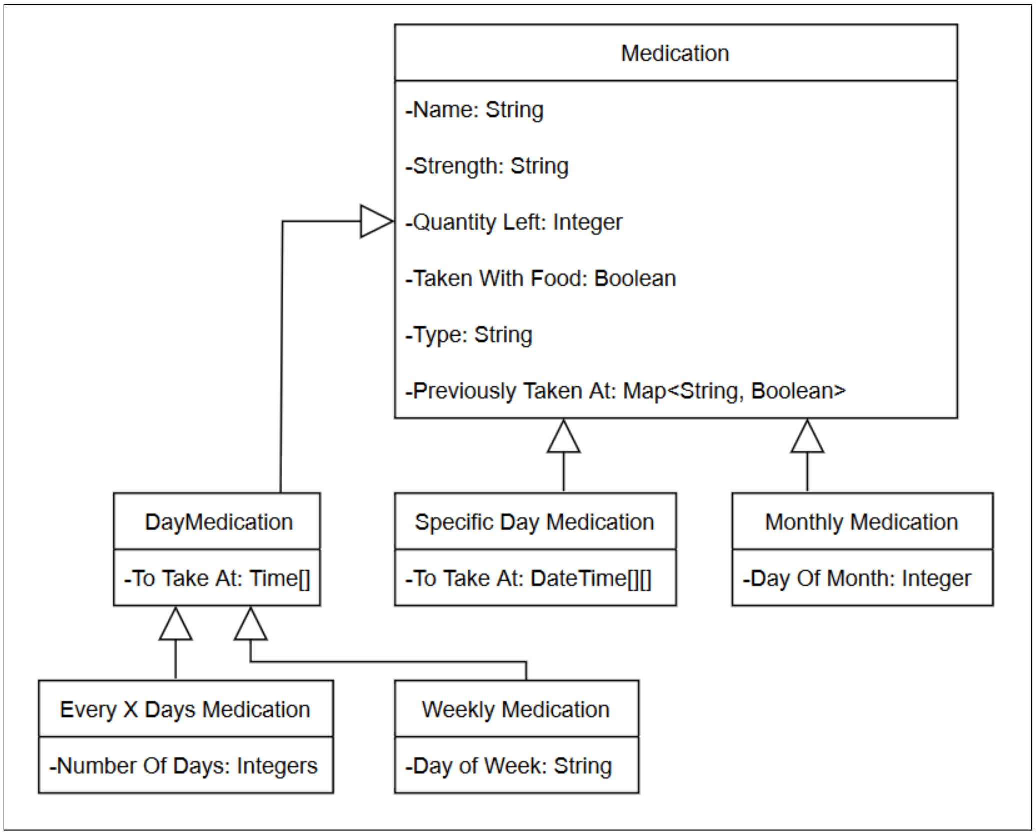


Figure 4.2: UML Class Diagram

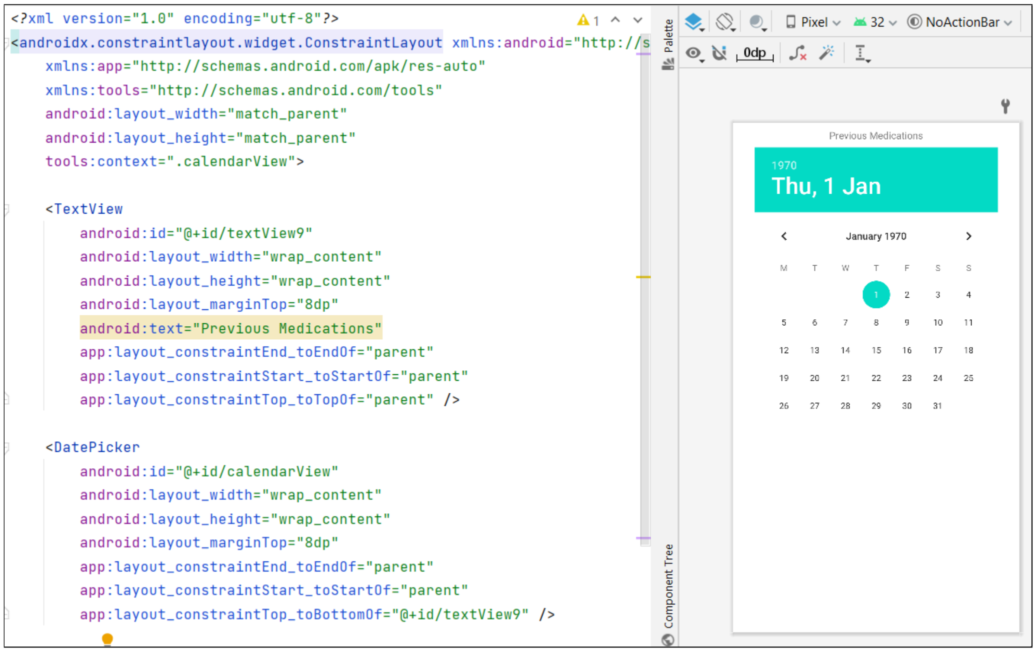
From the diagram shown above (Figure 4.2), the inheritance of the classes from the one base class ‘Medication’. Out of the six classes in the diagram, only the classes without children are being used for the implementation of the medication. Having these classes inherit from the base ‘Medication’ class, allows for all medications, regardless of the type, to be stored in the same array.

4.1.3 Implementation

Each feature to be implemented in the application was developed in the same order as shown in the task table above.

User Interface

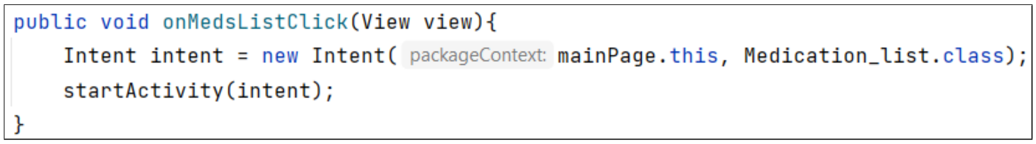
The user interface was developed using Android Studio's design view. The design view uses pre-built modules which can be 'dragged and dropped' onto what is known as a Layout Files. In Android Studio for each item to shown directly to the user, a layout file needs to be defined. A layout file is defined using the XML file format, and as such contains XML code which defines the layout for the screen. This was completed for each page to be shown to the user, along with separate layouts which are used by specific modules which were implemented at a further stage in development.



```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://s
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".calendarView">

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:text="Previous Medications"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <DatePicker
        android:id="@+id/calendarView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView9" />
```

Figure 4.3: Layout File Example, Showing Code and Visual Representation

The implementation of the navigation between pages was with buttons which were assigned events, in turn triggering functions in Java. One such function can be seen in figure 5 and uses the standard practice of using invoking a new activity through an intent (Android Developers, 2021d).



```
public void onMedsListClick(View view){
    Intent intent = new Intent( packageContext: mainPage.this, Medication_list.class);
    startActivity(intent);
}
```

Figure 4.4: Example of Navigation Function for Android Application in Java

Each page of the application shown graphically can be seen in Appendix C. Once the pages were designed and the interactivity developed, some testing was performed to ensure that the different pages of the application were able to be navigated to/from one another.

Scanning Functionality

To implement the scanning functionality, the use of an external module was used. The library used was the code scanner module by Yuriy Budiyev (Budiyev, 2022). This library implemented a fully functioning scanning activity which allowed for the control of the device's camera, along with the flash functionality for low light situations.

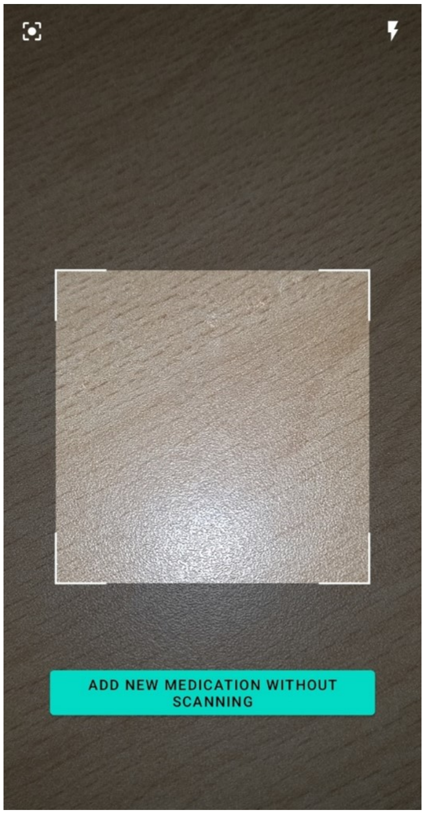
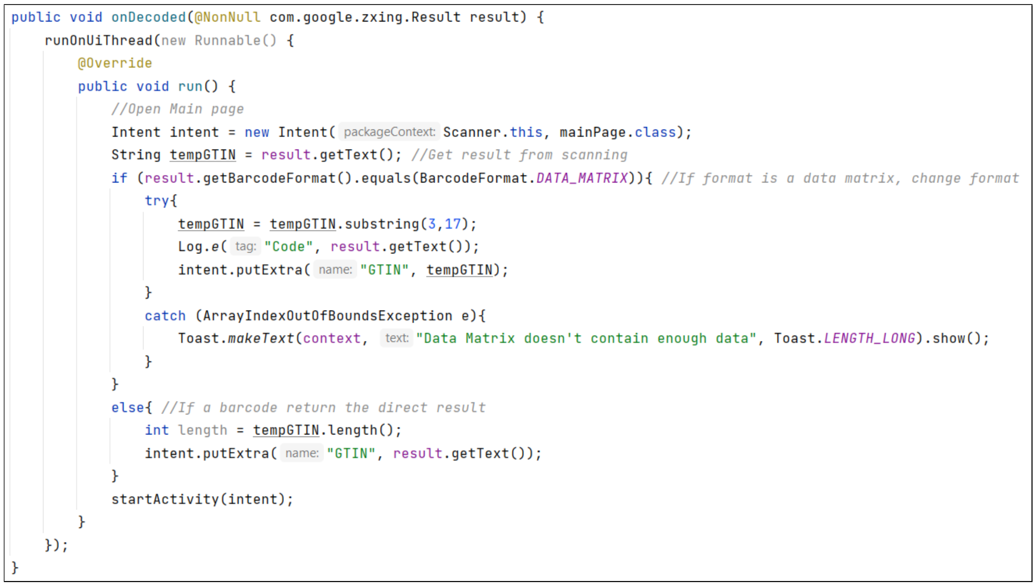


Figure 4.5: Example of Scanning Page Working

The library further provides a function to retrieve the data from the scanned image with a function. This function accepted a special built class as a parameter to contain information relating to the scanned code, such as the type of code scanned. This was especially useful as the differing types of code, return different amounts of data, and as such the read data needs to be parsed in diverse ways to extract the GTIN. The function definition for the program is below in figure 4.6.



```
public void onDecoded(@NonNull com.google.zxing.Result result) {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            //Open Main page
            Intent intent = new Intent( packageContext: Scanner.this, mainPage.class);
            String tempGTIN = result.getText(); //Get result from scanning
            if (result.getBarcodeFormat().equals(BarcodeFormat.DATA_MATRIX)){ //If format is a data matrix, change format
                try{
                    tempGTIN = tempGTIN.substring(3,17);
                    Log.e( tag: "Code", result.getText());
                    intent.putExtra( name: "GTIN", tempGTIN);
                }
                catch (ArrayIndexOutOfBoundsException e){
                    Toast.makeText(context,  text: "Data Matrix doesn't contain enough data", Toast.LENGTH_LONG).show();
                }
            }
            else{ //If a barcode return the direct result
                int length = tempGTIN.length();
                intent.putExtra( name: "GTIN", result.getText());
            }
            startActivity(intent);
        }
    });
}
```

Figure 4.6: Function Definition for Interpreting the Scanned Code Data

In the definition of the function to interpret the scanned code data, figure 4.6, the text of the result is taken and stored as a string. After this, the format of the code scanned is then checked to allow for the correct parsing of the string. There are two forms of code which are on UK medications, a data matrix, and a barcode. A data matrix contains not only the GTIN number for the medication, but also the expiry date for the specific batch, and the batch number, (Mediclabel, undated). In contrast, the barcode only contains the GTIN code for the medication. Because of this, the returned string needs to be parsed to retrieve only the GTIN number. Once the GTIN number is extracted from the string, it is sent to the main page, using the putExtra method on the intent. Furthermore, if the data matrix scanned contains too little data, it is likely that it is the matrix is not for medication, and this is relayed to the user using a ‘Toast’. A Toast is a popup notification, which displays to the user for a specific length of time and contains a string. The notification is in the middle of the page to alert the user to the issue.

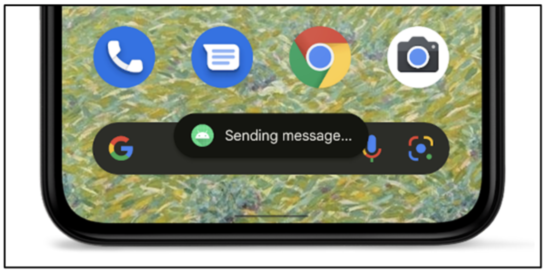
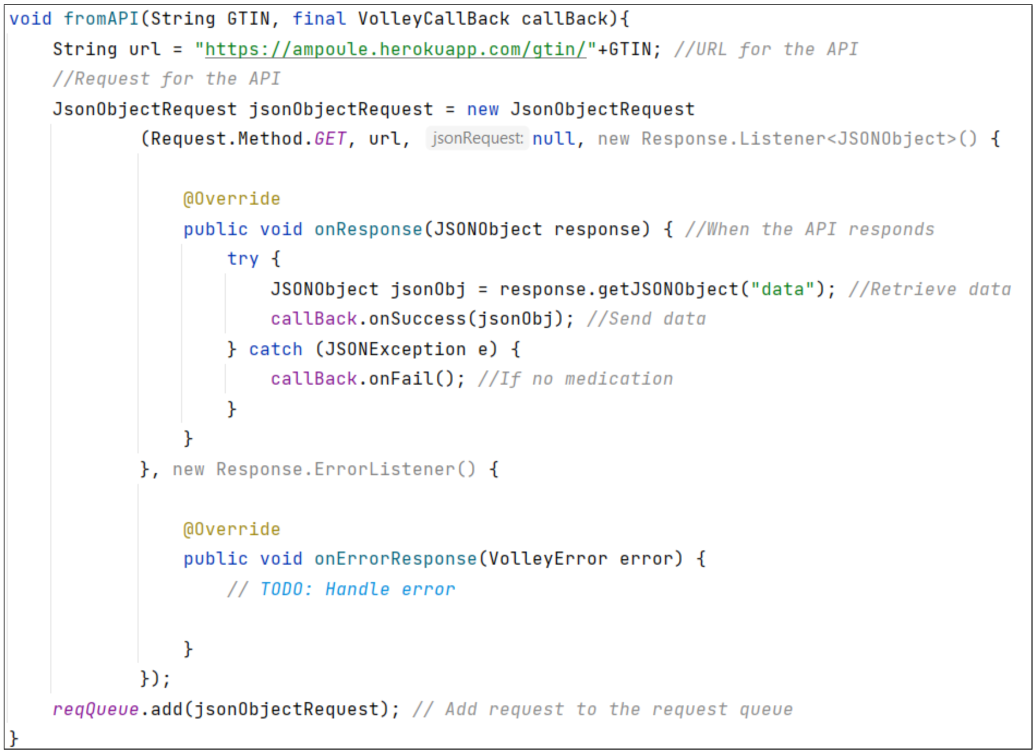


Figure 4.7: Example of a Toast Notification (Android Developers, 2021e)

Data Retrieval From API Functionality

To implement the data retrieval from the API, when the main page is reached, a check to see if there is an Extra containing a string exists. An Extra is a 'Bundle' of additional information which can be used to provide extended information to a component, in this case, an intent (Android Developers, 2021b). Once it is known for an Extra to exist, the data is then sent to a subroutine inside the singleton instance of the data controller.



```
void fromAPI(String GTIN, final VolleyCallBack callBack){
    String url = "https://ampoule.herokuapp.com/gtin/"+GTIN; //URL for the API
    //Request for the API
    JsonObjectRequest jsonObjectRequest = new JsonObjectRequest
            (Request.Method.GET, url, jsonRequest: null, new Response.Listener<JSONObject>() {

                @Override
                public void onResponse(JSONObject response) { //When the API responds
                    try {
                        JSONObject jsonObj = response.getJSONObject("data"); //Retrieve data
                        callBack.onSuccess(jsonObj); //Send data
                    } catch (JSONException e) {
                        callBack.onFail(); //If no medication
                    }
                }
            }, new Response.ErrorListener() {

                @Override
                public void onErrorResponse(VolleyError error) {
                    // TODO: Handle error

                }
            });
    reqQueue.add(jsonObjectRequest); // Add request to the request queue
}
```

Figure 4.8: Subroutine to Retrieve Data From API

Inside the data controller class, a subroutine was created to allow for the retrieval of data from the API. A concatenation of the string containing the link to the API and the GTIN was performed, this web address, when accessed, will provide a JSON object containing information about the medication searched for. In contrast, if the number provided is not of GTIN format or incorrect a JSON object will still be provided but containing a failed status. Examples of JSONs of correct and incorrect GTINS can be seen in appendix H. Once the JSON is received, if correct it will make a callback to the on-success function defined in the VolleyCallback. This function calls another function which opens a popup with the ability to add medication.

As the details of the medication have been retrieved from the API, this popup 'locks' the fields which it is able to fill for the user, meaning that no accidental changes are able to be made.

Data Storage (& Management)

The data was stored using a singleton class in Android (Geeks for Geeks, 2021). A singleton class allows for a single instance to be created across the application, which can be accessed throughout multiple activities without the need to reload the files and without the data being dumped from memory during garbage collection. This singleton class stores an array of a custom class type 'Medication', which contains all the medications entered into the system.

The Data Controller class was implemented to control the importing, storing and sorting of the data, along with some other general-purpose methods. This class includes a total of six properties and fields, used to store data to be used in activities. Along with properties and fields, fourteen subroutines were developed, three of which were: a private constructor and two overloaded instance retrieval subroutines.

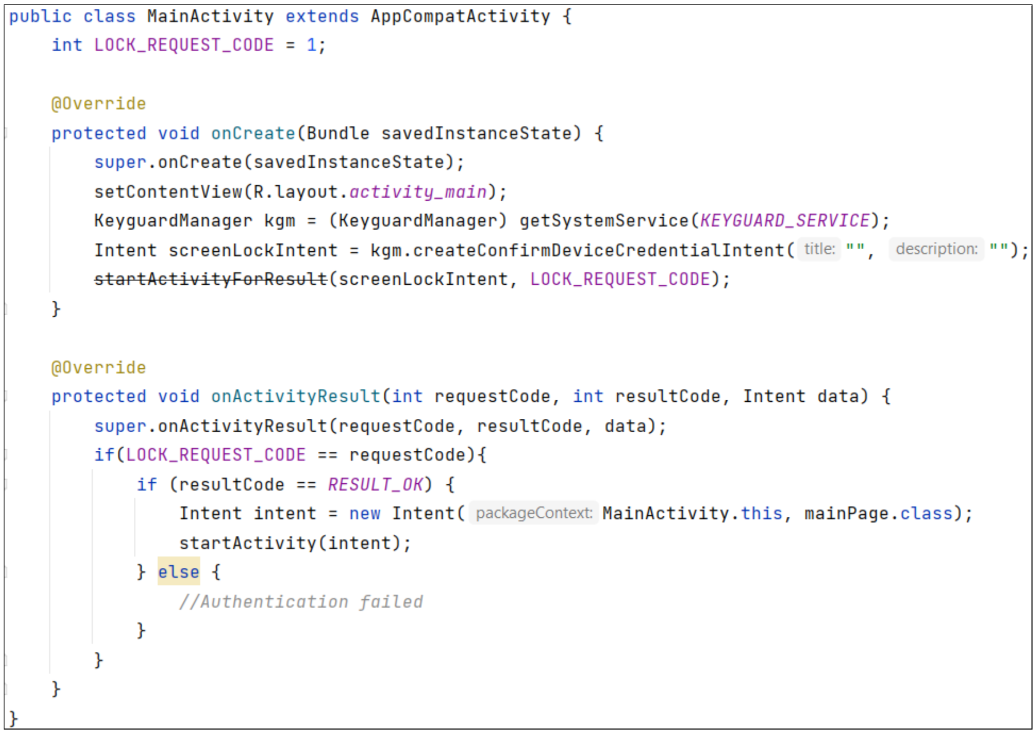
When the application is first loaded, the 'DataController' method 'GetInstance' with two parameters, is called. This method calls the private constructor, which sets the two parameters, along with building the data from the files if they exist. The function also returns an instance of 'DataController'. If a secondary class needs access to the 'DataController' the other 'GetInstance' subroutine is called, which also returns an instance of ‘DataController’. As this method requires no parameters to be passed, it can be called from any activity.

Along with these subroutines, there are two subroutines which deal with the storage, encryption, and decryption of the medications, these subroutines will be detailed below in the encryption section. In addition to these, four subroutines were implemented which retrieve the list of medications and information relating to the next medications to be taken. One subroutine also calculates which medications have been previously taken on a specific day.

The code for this singleton class, can be seen in appendix D.

App Security

When the user opens the application, they are presented with a page that asks for the primary method of security for their device. This method is usually a PIN code or a biometric system such as fingerprint scanning or facial scanning (Android Developers, 2021c). Once the correct information has been entered into the system, the ‘mainPage’ activity can be accessed, which will allow for the access of the data. If incorrect information has been entered, the user is informed of this through some text appearing on the screen, and the application will not be accessible.



```java
public class MainActivity extends AppCompatActivity {
    int LOCK_REQUEST_CODE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        KeyguardManager kgm = (KeyguardManager) getSystemService(KEYGUARD_SERVICE);
        Intent screenLockIntent = kgm.createConfirmDeviceCredentialIntent( title: "",  description: "");
        ~~startActivityForResult~~(screenLockIntent, LOCK_REQUEST_CODE);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(LOCK_REQUEST_CODE == requestCode){
            if (resultCode == RESULT_OK) {
                Intent intent = new Intent( packageContext: MainActivity.this, mainPage.class);
                startActivity(intent);
            } else {
                //Authentication failed
            }
        }
    }
}
```

Figure 4.9: Code to Cause Device Authentication Popup

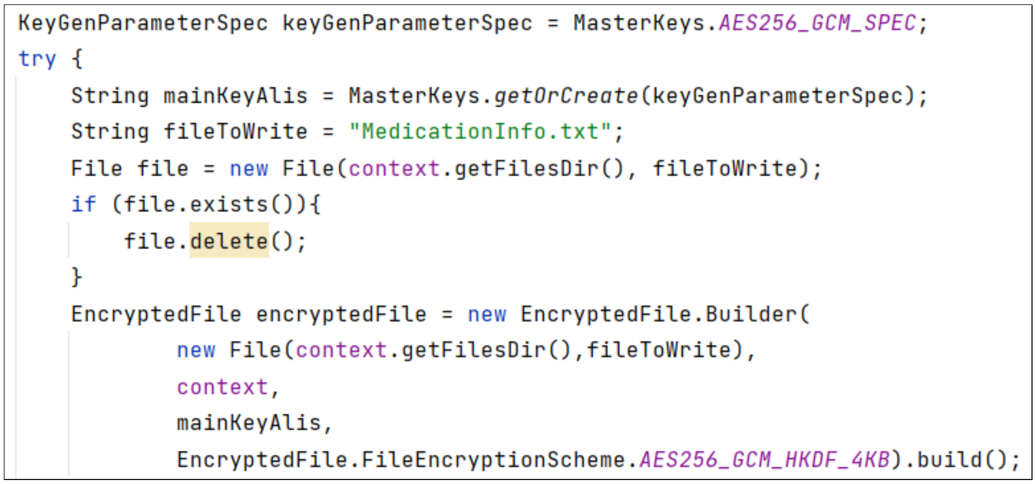
This was developed using the code as shown in figure 4.9. To start, an instance of the ‘KeyguardManager’ class is retrieved by the ‘getSystemService’ method. This instance can then be used to create a ‘ConfirmDeviceCredential’ intent, which can be used to retrieve a result using ‘startActivityForResult’. To allow navigation to the ‘mainPage’ activity, the ‘onActivityResult’ method was overridden. Overriding this method allowed for a custom action to be enacted when the result was returned from the activity. The request code in this function allows for the system to check that the correct request is being checked, along with the ‘resultCode’ containing a value pertaining to the result of the credential check. If this code returns the correct code, the ‘mainPage’ can then be opened, essentially giving access to the application.

Encryption

The encryption of the application is developed as part of the storage system. All medication information, as stored by the system on the device, is stored in a secure folder and encrypted using the AES 256-bit standard. This standard with a quantum computer could take up to 2.29×10^{32} years to crack, making it essentially impossible to do so (Tobias, 2022).

For an Android device, encryption can be performed through the use of the 'androidx.security.crypto' module. This module is part of the standard android library and allows for the creation or collection of the encrypted key, encryption of a file and decryption of the same file.

In an android device, security keys are kept inside what is known as a KeyStore. Some KeyStores are hardware based, using a truly random hardware system to generate encryption keys, however, most currently use a software based KeyStore (Android Developers, 2021a). The KeyStore will only give access to the key designated to the specific app, making it impossible for another application to use the key to potentially decrypt the files stored by the application.

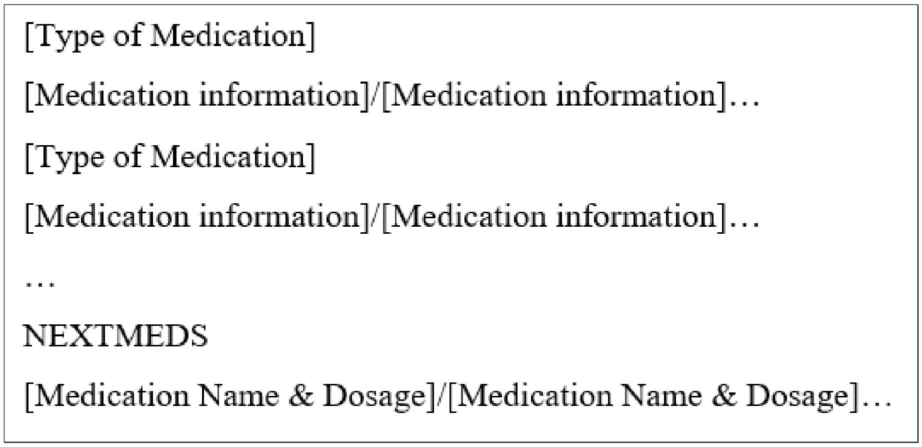


```java
KeyGenParameterSpec keyGenParameterSpec = MasterKeys.AES256_GCM_SPEC;
try {
    String mainKeyAlis = MasterKeys.getOrCreate(keyGenParameterSpec);
    String fileToWrite = "MedicationInfo.txt";
    File file = new File(context.getFilesDir(), fileToWrite);
    if (file.exists()){
        file.delete();
    }
    EncryptedFile encryptedFile = new EncryptedFile.Builder(
            new File(context.getFilesDir(),fileToWrite),
            context,
            mainKeyAlis,
            EncryptedFile.FileEncryptionScheme.AES256_GCM_HKDF_4KB).build();
```

Figure 4.10: Code Example for Opening/Creating an Encrypted File

In Appendix D, you can see the code produced to retrieve the key, and encrypt/decrypt the file, along with adding the information back to the DataController. For creating/writing to the file, a file is first opened, and checked if it already exists, at which point it would be deleted. This would be to allow for a brand-new file to be created, as any piece of data could have been changed in the file, which as it was encrypted, would need to be decrypted, edited, and then re-encrypted for storage. As the data would have already been loaded into the system, and this would have been needed for the first saving of the data, it seemed logical to overwrite the file each time.

Once the old file, if it existed, was deleted, the system creates a new encrypted file; when written to, an encrypted file automatically encrypts the byte array using the defined encryption method, as discussed previously, the AES 256-bit standard. To retrieve a byte array from the data stored in the system, one can be converted from a String. With most datatypes having a function which returns the string version of the data, retrieving a string of all the data was possible by concatenating a list of strings. The format that this final string would be in is an important aspect, as it would allow the system to discern what information is currently being represented. This, along with the medication classes inheriting from the same base class, enables the classes to represent similar data, made it clear that a method to distinguish which type of medication is being defined is necessary. A layout for the file was made and iterated over to figure out the easiest way to allow the system to discern between the types of medication.



```
[Type of Medication]
[Medication information]/[Medication information]…
[Type of Medication]
[Medication information]/[Medication information]…
…
NEXTMEDS
[Medication Name & Dosage]/[Medication Name & Dosage]…
```

Figure 4.11: Layout of the File for Storage

As shown by the figure above, the file will have two lines of data for each medication. The first line of each medication will inform the system what type of medication is stored on the next line. For the second line of each medication, each item of data will be separated by a '/'. This symbol was chosen as it was a symbol not used by the data types in the medication classes.

Now that a format for the file has been decided, it was possible to formulate the string which would be encrypted. To formulate the string, the list of medications was first looped through, using an if statement to check what type the medication is. The type of medication is then added to the string, along with each piece of information dedicated to the medication separated by the '/' character. Once all the medications have been added, the system then adds the names of the medications which need to be taken next.

Once the string has been formatted, it is converted into a byte array using the '.getBytes()' object method on the string variable. To finally submit the file into the encrypted file, the output stream for the encrypted file is opened and passed the byte array, storing the information in the file.

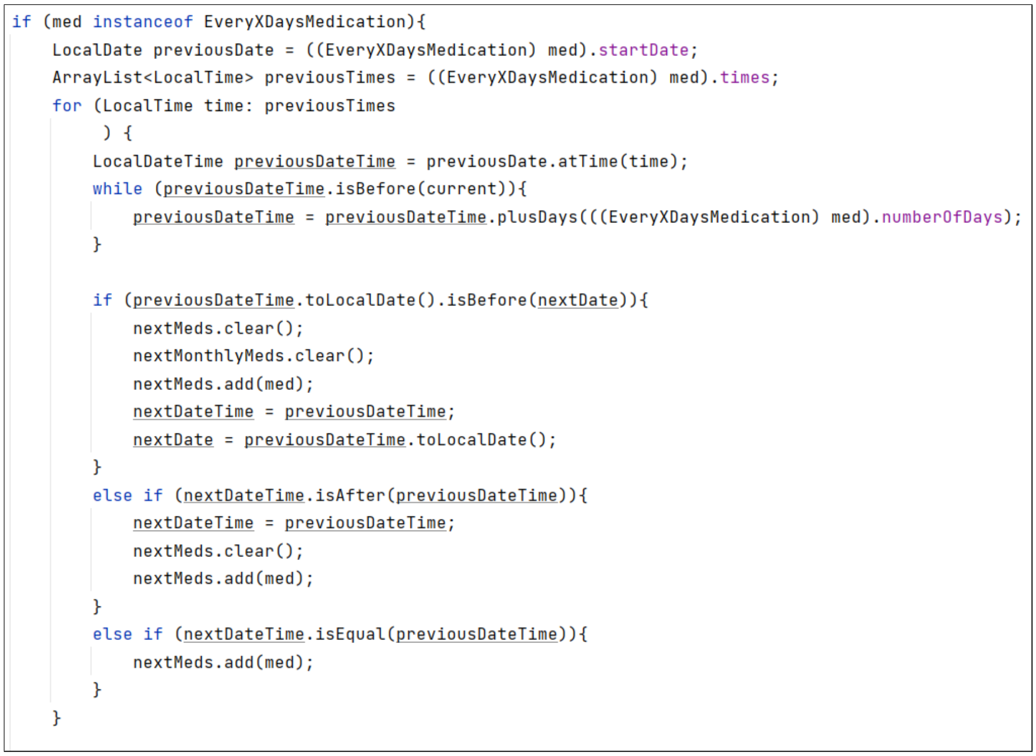
In addition, the decryption is performed with similar steps. To start, the encrypted file is opened, using the same method as with the encryption, using the EncyptedFile type. The input stream was opened from the file, and each byte was read from the file, and then converted into a string. This string is then split into an array, where each element is a line from the document. The array containing each line in the file is then iterated through, checking which lines will represent which types of medication. The 'data' line for the medication is then further split into its individual parts, being parsed into their original data types and then being added back into the data controller class.

Reminder System

The reminder system is based on the use of a notification system to alert the user that they need to take medications, which if clicked on, presents the user with the ability to see the medication(s) they need to take along with the ability to mark that they have taken the medication(s).

To start, a method to calculate when the next medication was needed. This was implemented inside the DataController class and is recalculated every time a new medication is added, or the application is opened.

To calculate the time when the next medications are to be taken, the program first loops through each medication in the system. It checks what type of medication it is, which allows for the use of type specific functions to access the information about the medication. Each medication has a different algorithm used to check all their information.



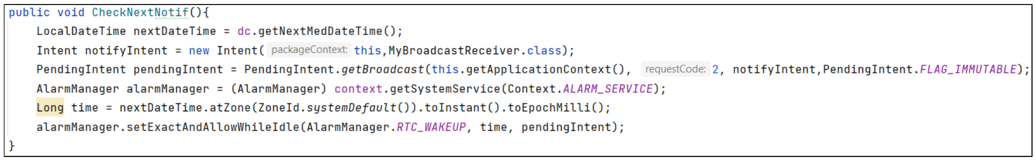
```
if (med instanceof EveryXDaysMedication){
    LocalDate previousDate = ((EveryXDaysMedication) med).startDate;
    ArrayList<LocalTime> previousTimes = ((EveryXDaysMedication) med).times;
    for (LocalTime time: previousTimes
         ) {
        LocalDateTime previousDateTime = previousDate.atTime(time);
        while (previousDateTime.isBefore(current)){
            previousDateTime = previousDateTime.plusDays(((EveryXDaysMedication) med).numberOfDays);
        }

        if (previousDateTime.toLocalDate().isBefore(nextDate)){
            nextMeds.clear();
            nextMonthlyMeds.clear();
            nextMeds.add(med);
            nextDateTime = previousDateTime;
            nextDate = previousDateTime.toLocalDate();
        }
        else if (nextDateTime.isAfter(previousDateTime)){
            nextDateTime = previousDateTime;
            nextMeds.clear();
            nextMeds.add(med);
        }
        else if (nextDateTime.isEqual(previousDateTime)){
            nextMeds.add(med);
        }
    }
```

Figure 4.12: Code for Checking Next ‘EveryXDays’ Medication

For ‘Every ‘x’ Days’ frequency of medication, the application takes the original date and loops through the times. It uses this information to combine them into mixed date/times and continually adds the number of days between doses until it is after the current date/time. Once this happens, the system checks if this date time is before the next medication time, if so it reassigns the next medication time to that time and clears the list of next medications. If the date/times are the same, it appends the current medication to the next medication list. As monthly medications can be taken at any time during a day, it also checks if the date is before the next medication time, as monthly medication would not be taken on those days.

For the other frequencies of medication, the checks to assign the medication to the current next medication time and arrays are the same, however finding the dates after the current date time is tailored to each frequency of medication. The code for this can be seen in Appendix D.

Now that the next medications have been set and the time has been figured out, the notification needs to be set. To create a notification in an android application, it needs to be performed through the alarmManager class. This class allows for an activity to be started at a prespecified time. To cause this activity to make a notification, the notification is built inside the activity using the standard method of creating a notification as specified in the Android Developers documentation. This notification is set to open the separate login page for the notification, 'loginNotification'.

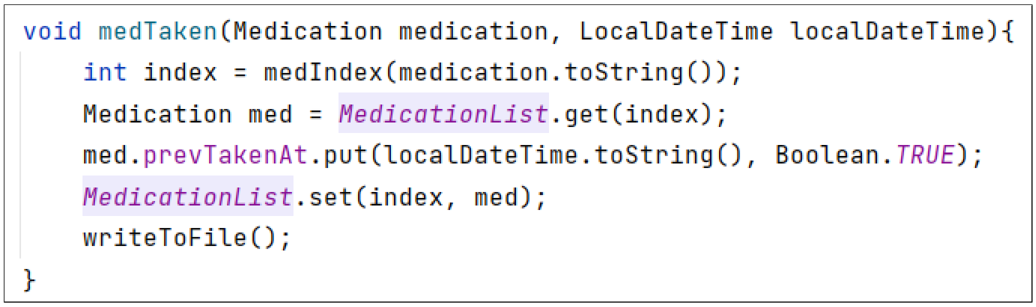


```java
public void CheckNextNotif(){
    LocalDateTime nextDateTime = dc.getNextMedDateTime();
    Intent notifyIntent = new Intent( packageContext: this,MyBroadcastReceiver.class);
    PendingIntent pendingIntent = PendingIntent.getBroadcast(this.getApplicationContext(),  requestCode: 2, notifyIntent,PendingIntent.FLAG_IMMUTABLE);
    AlarmManager alarmManager = (AlarmManager) context.getSystemService(Context.ALARM_SERVICE);
    Long time = nextDateTime.atZone(ZoneId.systemDefault()).toInstant().toEpochMilli();
    alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP, time, pendingIntent);
}
```

Figure 4.13: Code to set a Notification at a Specific Time

Tracking System

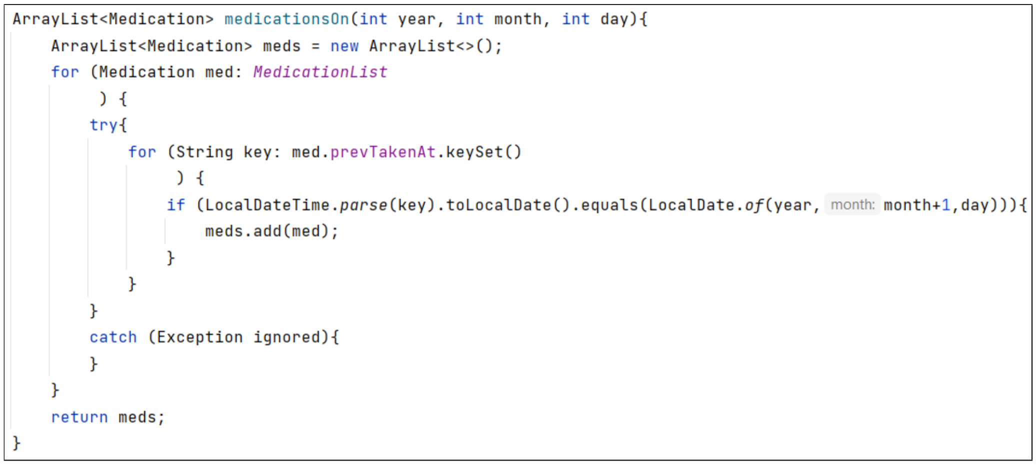
Once the user logs in through the notification, they are met with tickboxes to allow for them to mark that they have taken the medication. Once this tickbox has been checked, it is unable to be unchecked by the user, and the medication is marked as taken on that specific day. To do this, the system searches through the medications in the system to find the index of the medication with the same name/dosage combination as the one checked, appending the current date/time to the end of the taken array inside of the medication. Once it is written, it gets written to the file to ensure no data-loss.



```java
void medTaken(Medication medication, LocalDateTime localDateTime){
    int index = medIndex(medication.toString());
    Medication med = MedicationList.get(index);
    med.prevTakenAt.put(localDateTime.toString(), Boolean.TRUE);
    MedicationList.set(index, med);
    writeToFile();
}
```

Figure 4.14: Code to Track When a Medication was Taken

In addition to this, the users can access when they took medication through the calendar on the calendar page. This page displays a popup whenever the user clicks on a date, and then searches through all the medications, checking if the dates are equal. It adds these medications to an array and returns it to the main function, where these are applied to a recycler, which displays the name of all the medications.



```
ArrayList<Medication> medicationsOn(int year, int month, int day){
    ArrayList<Medication> meds = new ArrayList<>();
    for (Medication med: MedicationList
         ) {
        try{
            for (String key: med.prevTakenAt.keySet()
                 ) {
                if (LocalDateTime.parse(key).toLocalDate().equals(LocalDate.of(year, month: month+1,day))){
                    meds.add(med);
                }
            }
        }
        catch (Exception ignored){
        }
    }
    return meds;
}
```

Figure 4.15: Code to Calculate Medications Taken On A Specific Date

4.1.4 Testing

In the project, there were two forms of testing: black-box and beta testing.

Black-Box Testing

The first stage of testing was to perform black-box testing. This is where the developer uses the finished product and checks the functionality of specific features on the software. This form of testing can be performed quicker than other options, due to its lack of needing to evaluate each subroutine during programming. This was especially useful due to the time constraints for the project.

To begin black box testing, a list was made which contained all the implemented features. Using this list, each feature was assigned several tests, along with a predicted outcome for each. If each test passed, the application would be proven to be completely working and therefore ready for the next stage of testing.

As shown by the results of the testing, appendix A, there was only one test which did not perform as expected, which was on the “Scanner Page” on test six. The test showed that through checking a variety of different data matrices, the matrices which stored a lower number of characters caused the application to crash. This was quickly rectified by checking the length of the data returned by scanning the data matrix and therefore informing the user that their data matrix is likely not to be from a medication.

With all the features checked and the application working as expected, the second stage of testing was able to begin.

Beta Testing

The second stage of testing was through beta testing. Beta testing usually involves an external party or parties evaluating the application through normal use, the recruited beta testers were also used as participants in the study, detailed in sections 3.4 and 4.2. To receive the results of the beta testing, any issues the participants have will be able to be sent through an online form. Issues which were raised during beta testing are:

1. When the medications are to be taken, if the screen orientation flips, it makes the checkboxes available to be clicked again
2. Notifications would override each other if medications were to be taken too close together
3. Sometimes the camera wouldn't focus well on the barcodes
4. Some placeholder features were not obvious they were placeholders
   (a) Upcoming Medication
   (b) Edit Medication List
5. Some of the text is cut off on the screen in popups
6. Requirements for each text box should be clearer/Some text is default
7. Issue where some notifications wouldn't appear

As shown by the above list, seven issues were found during testing. Most of these issues were minute, and therefore did not pose a significant issue to the workings of the application and did not require immediate rectification. In regard to the notification issue, more testing is needed as this issue was only found by one participant, suggesting that the outcome was erroneous, this implies that it may be down to user error or that there were issues with some hardware.

In addition to asking the beta testers about any issues with the application, the form also included a section asking if there were any additional features which would have been useful for future development opportunities. Such features suggested are:

1. Add Medications taken outside of the notification page
2. App should automatically close
3. Features for future development suggested in previous sections:
   (a) View medications to be taken over a period of time
   (b) Edit Medications
   (c) Amount of medications left and sends notification when running low

4.1.5 Operation

This project provides a solution to the problem and has potential users. However, the purpose of the project was about finding if a solution like the one produced, would provide effective assistance to those who can be non-adherent to their medication regiment. Therefore, the application developed, is a proof of concept and does not provide the full functionality which would be required if the application was to be deployed publicly. Instead of deploying the application publicly, the application will not be deployed through 'normal' means such as Google Play (The software designed to download applications easily for android devices).

Therefore, the application was only downloadable through a link, provided after the acceptance of participation in the study, for use during the beta testing/study. This link gave access to a repository which contained instructions for installation along with the 'APK' file, which is the installation file format for android.

The application provides no incentive for the user nor instruction for the user, in regards of application use. For future development opportunities, an instructional tutorial within the application would be highly beneficial for traversal within the application, potentially partnered with an instructional manual to allow for a permanent reference to the tutorial.

4.2 Research

4.2.1 Participant Recruitment

Participants for the study were recruited through an anonymous basis, by sending an anonymous online form (appendix E) into a group of current students at the University of Lincoln. This inherently will apply some bias to the results, as students are typically younger, with more experience with technology. However, the functionality of the application is not the topic of the study, but rather the use of the notifications and their effect on a persons' medication non-adherence.

In addition to this, there were two requirements for the participants: that they had access to an Android device and they had internet access to download the application and provide feedback through the forms.

4.2.2 Evidence That Ethical Procedures Have Been Followed

Ethical approval was given to the project, in which the study performed as described in the forms submitted. In accordance with the ethical form submitted, the sign-up form was accessible at any time and contained all the information a project information document would have contained. This allowed for the participants to have adequate time before signing up to the study and allowed for the participants to give informed consent. Once they had agreed to partake in the survey, they were presented with a page to allow them to download and install the application along with the ability to access the results form. The participants were instructed after a week to fill in the second form provided to them.

4.2.3 Study Design

The research question for the study is “Can an electronic reminder improve medication regime adherence?”. The plan was to get anonymous participants to download the application and after a brief period, provide information on the applications effect on their medication adherence, asking for the frequencies in which they would forget medication before and during the use of the application. As the data being collected is numerical data, the data can be analysed. Before the data is analysed, the data would be checked for erroneous values, removing these values from the dataset. The non-erroneous values can then be averaged to find an average difference between the two values. These averages can then be compared to each other, to see if there is a positive or negative trend when using the application.

The hypothesis for the study is that if people use electronic reminders, it can improve their adherence to a medication regime.

4.2.4 Procedure

The procedure of the study from a participant point of view, would start with receiving a message in an online group asking if anyone would like to participate in my study, if so to click on the link. Once the participant clicked on the link, they would be taken to a Google Forms page, which showed the information about the study, such as: its purpose, contact details and right to withdraw. At the bottom of this page, the participant is asked to tick a checkbox, indicating they agree to partake in the study and have read the information displayed.

If the participant submits the form and has ticked the checkbox, they are greeted with a page showing two links, one to direct them to download the application and the other to direct them to the results form. The participant is asked to download the application first and once they click on the corresponding link, is met with a GitHub repository page. This page shows them an instructional page on how to download the application through two methods along with the file for downloading.

After a week, the participant was asked to access the second link from the form.

This would take them to a Google Form, which asked four questions. Two questions related to Beta testing and two questions related to the study, asking for them to input numbers relating to the number of times medications were missed. Once this form was submitted, that would be the last interaction that the participant would have with the study.

4.2.5 Results and Analysis

The specific results from the forms can be seen in appendix G. With issues reported by one participant, it seemed they were unable to effectively use the application, potentially giving erroneous information. Due to this we end up with 3 participants' data, out of the 4 participants.

Analysing the results gives us an average of $2.\dot{6}$ for the number of times medications were not taken before. There is an average of $0.\dot{6}$ for the number of times medications were not taken during the use of the application. This would imply that the use of such a notification, along with the use of informing the participants of which medications are due, would allow for people to become more adherent to their medication regime. To ensure these results are non-anomalous, the study could be repeated with a larger participant group to find more averages, along with a longer period to see if the effects last.

Chapter 5

Conclusions

To conclude, the results from the study identify that it is possible to use an electronic reminder, such as the one provided by the application developed for this project, to increase a persons' adherence to a medication regime. It is unclear at this stage whether the effect will last for a longer period or whether this effect only lasts a short amount of time, such as the week used for the study.

In terms of the aims and objectives laid out in the introduction, the project fulfilled the aim, along with the objectives. It managed to supply an application which enabled the users to track information about their medication(s), receive reminders about their medication(s) along with tracking which medication(s) they have taken in the past. This application was developed for an Android device and tested with 4 different people. The user interface was designed using Microsoft Office PowerPoint, showing the interactions between different pages in the application.

There were some issues during testing, although only one person out of the whole study had a negative experience using the application. For future release, more development is needed to ensure that no issues are experienced during the use of the application, along with enabling all the features that would be needed on a day-to-day basis for all users.

The application supplied could be more efficient in solving the research question. This is due to the application having some issues, and therefore needed more insight from the users into how the application was intended to be used. This can make it harder for general users, and therefore also for the people who were testing the application.

Chapter 6

Reflective Analysis

The application was developed in a timely manner, and the overall solution and results of the study, gave an answer to the question asked. This application was successful in its purpose, allowing many participants in the study to increase their medication adherence levels over the course of a week. This was due to the time management techniques which were laid out in the beginning of the project, giving clear perspective on the tasks ahead and the tasks completed.

However, there were some issues with the application during the study/beta testing. In retrospect, having the beta testing incur multiple phases separate to the study phase would allow for all issues with the application be sorted before the study, which would decrease the likelihood of people having issues with the application itself.

Secondly, if possible, more active recruiting of participants would have potentially allowed for more people to partake in the study. This could have given more accurate results and made the results of the study more reliable.

Finally, the solution lacks a degree of elegance which other current solutions provide. This application developed was not an artefact designed for release, but in contrast, more of a proof of concept, meaning the polish it could have had was not needed. Even though the complexity was not needed, it would have allowed for the testing to go through in a smoother manner, allowing for more accurate results.

References

Andrei, B.A., Casu-Pop, A.C., Gheorghe, S.C. and Boiangiu, C.A. (2019). *A Study On Using Waterfall And Agile Methods In Software Project Management. Journal Of Information Systems & Operations Management,* 13(1) 125-135. Available from: `http://www.rebe.rau.ro/RePEc/rau/jisomg/SU19/JISOM-SU19-A12.pdf` [accessed on 27/10/2021].

Android Developers (2021a). *Android keystore.* Android developers' documentation. Available from: `https://developer.android.com/training/articles/keystore` [accessed on 15/03/2022].

– (2021b). *Intent.* Android developers' documentation. Available from: `https://developer.android.com/reference/android/content/Intent` [accessed on 25/01/2022].

– (2021c). *Show a biometric authentication dialog.* Android developers' documentation. Available from: `https://developer.android.com/training/sign-in/biometric-auth` [accessed on 15/02/2022].

– (2021d). *Start another activity.* Android Developers' documentation. Available from: `https://developer.android.com/training/basics/firstapp/starting-activity` [accessed on 20/12/2021].

– (2021e). *Toasts Overview.* Android Developers' documentation. Available from: `https://developer.android.com/guide/topics/ui/notifiers/toasts` [accessed on 03/01/2022].

Apple (2022). *Distributing Your App for Beta Testing and Releases.* Apple Developer Documentation. Available from: `https://developer.apple.com/documentation/xcode/distributing-your-app-for-beta-testing-and-releases` [accessed on 21/03/2022].

Ask Any Difference (undated). *Difference Between Microsoft Forms and Google Forms (With Table).* Ask Any Difference. Available from: `https://askanydifference.com/difference-between-microsoft-forms-and-google-forms-with-table/` [accessed on 12/02/2022].

Brown, M.T. and Bussell, J.K. (2011). *Medication Adherence: WHO Cares?* Mayo Clinic Proceedings 86(4), 304-314. Available from: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3068890/` [accessed on 18/01/2022].

Budiyev, Y. (2022). *Code scanner library for Android, based on ZXing*. GitHub. Available from: https://github.com/yuriy-budiyev/code-scanner/ [accessed on 02/01/2022].

Charlesworth, C.J., Smit, E., Lee, D.S., Alramadhan, F. and Odden, M.C. (2015). *Polypharmacy among adults aged 65 years and older in the United States*. Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences, 70(8), 989–995. Available from: https://pubmed.ncbi.nlm.nih.gov/25733718/ [accessed on 10/01/2022].

Chowdhry, A. (2015). *Average iPhone Price Increases To $687 And Android Decreases To $254, Says Report*. United States of America: Forbes. Available from: https://www.forbes.com/sites/amitchowdhry/2015/02/03/average-iphone-price-increases-to-687-and-android-decreases-to-254-says-report/?sh=4f5bb141539e [accessed on 04/04/2022].

David, M. (2019). *Pros and cons of Android app development tools*. Tech Target. Available from: https://www.techtarget.com/searchsoftwarequality/feature/Native-Android-development-What-are-the-best-tools [accessed on 21/11/2021].

De LAND, P.N. and Chase, W.W. (1990). *Statistics notebook: Entry I.J, Types of Data: Nominal, Ordinal, Interval, and Ratio Scales. Optometry and Vision Science*. 67(2) 155. Available from: https://journals.lww.com/optvissci/Citation/1990/02000/Statistics_Notebook__Entry_I_J,_Types_of.21.aspx [accessed on 03/04/2022].

Department of Health & Social Care (2021). *Good for you, good for us, good for everybody. A plan to reduce overprescribing to make patient care better and safe, support the NHS, and reduce carbon emissions*. Available from: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1019475/good-for-you-good-for-us-good-for-everybody.pdf [accessed on 15/01/2022].

Duerden, M. and Payne, R. (2014). *Polypharmacy – What is it and how common is it*. Prescriber. 25(21), 44-47. Available from: https://wchh.onlinelibrary.wiley.com/doi/epdf/10.1002/psb.1274 [accessed on 13/01/2022].

Geeks for Geeks (2021). *Singleton Class in Java*. Geeks for Geeks. Available from: https://www.geeksforgeeks.org/singleton-class-java/ [accessed on 22/01/2022].

Google (2022). *Is Google Drive secure?* Google. Available from: https://support.google.com/drive/answer/141702?hl=en-GB&co=GENIE.Platform%3DDesktop [accessed on 05/04/2022].

Hildebrand, D. and Serenyi, D. (2021). *Colossus under the hood: a peek into Google's scalable storage system*. Google Cloud Blog. Available from: https://cloud.

google.com/blog/products/storage-data-transfer/a-peek-behind-colossus-googles-file-system [accessed on 04/04/2022].

Kastrenakes, J (2021). *Apple says there are now over 1 billion active iPhones.* The Verge. Available from: https://www.theverge.com/2021/1/27/22253162/iphone-users-total-number-billion-apple-tim-cook-q1-2021/ [accessed on 18/11/2021].

Marcum, Z.A. and Gellad, W.F. (2012). *Medication Adherence to Multi-Drug Regimens.* Clinics in Geriatric Medicine. 28(2), 287-300. Available from: https://pubmed.ncbi.nlm.nih.gov/22500544/ [accessed on 21/01/2022].

Marcum, Z.A., Sevick, M.A. and Handler, S.M. (2013). *Medication Nonadherence: A Diagnosable and Treatable Medical Condition.* JAMA 309(20) 2105-2106. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3976600 [accessed on 06/11/2021].

Mashta, O. (2010). *English NHS wastes £300m of drugs every year, study finds.* British Medical Journal. Available from: https://www-bmj-com.proxy.library.lincoln.ac.uk/content/341/bmj.c6799 [accessed on 19/01/2022].

McGee-Lennon, M.R. (2012). *Reminders that make sense: Designing multisensory notifications for the home.* Journal of Assistive Technologies. 6(2), 93-104. Available from: https://eds-p-ebscohost-com.proxy.library.lincoln.ac.uk/eds/detail/detail?vid=0&sid=387bea81-cc85-4e5e-931f-7edf1b5daf53%40redis&bdata=JnNpdGU9ZWRzLWxpdmUmc2NvcGU9c2l0ZQ%3d%3d#AN=edsemr.10.1108.17549451211234957&db=edsemr [accessed on 24/01/2022].

Mediclabel (undated). *Datamatrix.* Mediclabel. Available from: https://www.mediclabel.nl/en/datamatrix/ [accessed on 20/01/2022].

National Health Service UK (2021). *Medicines: tips for carers.* National Health Service UK. Available from: https://www.nhs.uk/conditions/social-care-and-support-guide/practical-tips-if-you-care-for-someone/medicines-tips-for-carers/ [accessed on 06/11/2021].

NHSBSA (2021). *Prescription Cost Analysis - England 2020/21.* NHSBSA. Available from: https://www.nhsbsa.nhs.uk/statistical-collections/prescription-cost-analysis-england/prescription-cost-analysis-england-202021 [accessed on 25/10/2021].

Rinaldi, C. (2021). *The most popular Android version in 2021 is ridiculously old.* NextPit. Available from: https://www.nextpit.com/most-popular-android-version-2021-old [accessed on 04/04/2022].

Sahm, L., MacCurtain, A., Hayden, J., Roche, C. and Richards, H.L. (2009). *Electronic reminders to improve medication adherence – are they acceptable to the patient?* Pharmacy World & Science 31(6) 627-629. Available from: https://proxy.library.lincoln.ac.uk/login?url=https://search.ebscohost.com/

login.aspx?direct=true&site=eds-live&db=edsovi&AN=edsovi.00008547.200912000.00007 [accessed on 01/11/2021].

Shade, M., Kupzyk, K., Pullen, C., Boron, J. and Manley, N. (2019). *Ease of Use and Usefulness of Medication Reminder Apps among Rural Aging Adults.* Journal of Community Health Nursing. 36(3), 105-114. Available from: https://pubmed.ncbi.nlm.nih.gov/31291770/ [accessed on 11/01/2022].

Specialist Pharmacy Service (2020). *What products or interventions are available to aid medication adherence?* Available from: https://www.sps.nhs.uk/wp-content/uploads/2020/05/SPS_ProductsInterventionsToAidMedicationAdherence_May20_FINAL.pdf [accessed on 21/01/2022].

Tobias, E. (2022). *128 or 256 bit Encryption: Which Should I Use?* Ubiq. Available from: https://www.ubiqsecurity.com/128bit-or-256bit-encryption-which-to-use/ [accessed on 23/04/2022].

Trueman, P., Taylor, D.G., Lowson, K., Bligh, A., Meszaros, A., Wright, D., Glanville, J., Newbould, J., Bury, M., Barber, N. and Jani, Y.H. (2010). *Evaluation of the scale, causes and costs of waste medicines.* Available from: https://discovery.ucl.ac.uk/id/eprint/1350234/ [accessed on 21/01/2022].

Vervloet, M., Linn, A.J., van Weert, J.C.M., de Bakker, D.H., Bouvy, M.L. and van Dijk, L. (2012). *The effectiveness of interventions using electronic reminders to improve adherence to chronic medication: a systematic review of the literature.* Journal of the American Medical Informatics Association 19(5) 696-704. Available from: https://proxy.library.lincoln.ac.uk/login?url=https://search.ebscohost.com/login.aspx?direct=true&site=eds-live&db=edsovi&AN=edsovi.00042637.201209000.00004 [accessed on 06/11/2021].

Wilson, C. (2021). *DM+D Data API.* GitHub. Available from: https://github.com/chriswilson1982/dmd-data-api [accessed on 12/01/2022].

Word Count 12459

Appendix A
Black Box Testing Tables

This appendix shows the tables of the testing results for the blackbox testing.

A.1 Opening Page

Table A.1: Table of Tests for the Opening Page

Test No	Test	Expected Result	Actual Result
1	Open Application	Mobile lock is required to open the application	As Expected
2	Use correct authentication method	Allows access to the application, opening the main page	As Expected
3	Use false authentication method	Does not allow access to the application, staying on the authentication page	As Expected

A.2 Main Page

Table A.2: Table of Tests for the Main Page

Test No	Test	Expected Result	Actual Result
1	Click on Calendar Button	Opens the calendar view activity	As Expected
2	Click on Med List Button	Opens the Medication list activity	As Expected
3	Click on New Med Button	Opens the scanner activity, asking for camera permissions if not previously allowed.	As Expected

A.3 Calendar Page

Table A.3: Table of Tests for the Calendar Page

Test No	Test	Expected Result	Actual Result
1	Click on date in calendar	Popup showing medication taken on specific day	As Expected
2	Change month/year on calendar	Shows the selected month and year	As Expected
3	Select date in future	Nothing	As Expected

A.4 Medication List Page

Table A.4: Table of Tests for the Medication List Page

Test No	Test	Expected Result	Actual Result
1	Pre-Requisite: Added Medication(s) Check Medications are visible	Medications are visible	As Expected

A.5 Scanner Page

Table A.5: Table of Tests for the Scanner Page

Test No	Test	Expected Result	Actual Result
1	Open Page	Camera is visible	As Expected
2	Click on lightning symbol	Torch is turned on and off	As Expected
3	Scan barcode – Medication	Main page is re-opened with data corresponding to medication scanned	As Expected
4	Scan barcode – Non-Medication	Main page is reopened, the addMedication popup is empty	As Expected
5	Scan 2D-Data matrix - Medication	Main page is reopened with data corresponding to medication scanned	As Expected
6	Scan 2D-Data matrix – Non-Medication	Main page is reopened, the addMedication popup is empty	As expected – doesn’t work for small data matrix
7	Press "add medication without scanning" button	Main page is reopened, the addMedication popup is empty	As Expected

A.6 Add Medication Popup

Table A.6: Table of Tests for the Add Medication Popup

Test No	Test	Expected Result	Actual Result
1	Add Every X days Medication	Medication is added and shown in the Med List Page	As Expected
2	Add Specific Day Medication	Medication is added and shown in the Med List Page	As Expected
3	Add Weekly Medication	Medication is added and shown in the Med List Page	As Expected
4	Add Monthly Medication	Medication is added and shown in the Med List Page	As Expected
5	Try to add Medication without filling all the entries	Brings up a 'Toast' message to inform the user that some items are not filled in	As Expected
6	Click off the popup to stop adding medication	Goes back to the mainPage Activity	As Expected

A.7 Notifications

Table A.7: Table of Tests for the Notifications

Test No	Test	Expected Result	Actual Result
1	Set notification to appear while app is on	Notification appears	As Expected
2	Set notification to appear while app is shut down	Notification appears	As Expected
3	Click on notification	Takes the user to the notification login page	As Expected

A.8 Notification Login Page

Table A.8: Table of Tests for the Notification Login Page

Test No	Test	Expected Result	Actual Result
1	Open Page	Authentication appears	As Expected
2	Correct authentication method	Takes the user to the list of medication to be taken at that time	As Expected
3	Incorrect authentication method	Informs the user that the authentication method is incorrect.	As Expected

A.9 Medications to Take Page

Table A.9: Table of Tests for the Medications to Take Page

Test No	Test	Expected Result	Actual Result
1	Open Page	Medications to take are visible with the tick box visible	As Expected
2	Marked as taken – Checkbox ticked	Tick box goes grey, unable to be changed	As Expected

A.10 File Reading/Writing

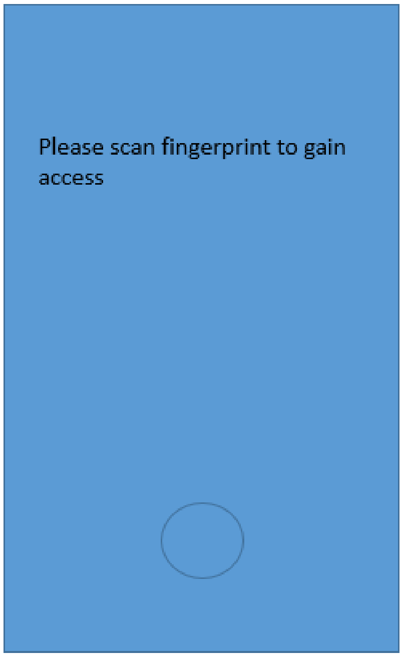
Table A.10: Table of Tests for File Reading and Writing

Test No	Test	Expected Result	Actual Result
1	Medications are visible after shutting application down	Medications are shown to the user through the Med List	As Expected

Appendix B
Application Design

This appendix shows images of the design for the user interface.

B.1 Fingerprint Scanning Page



Please scan fingerprint to gain access

Figure B.1: Design of the Fingerprint Scanning Page

B.2 Main Page

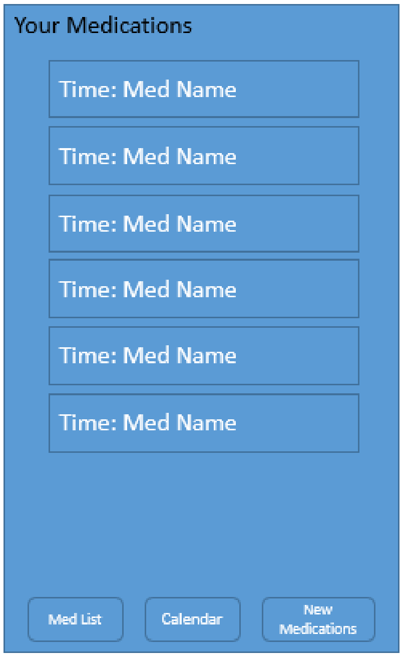


Figure B.2: Design of the Main Page

B.3 Medication Scanning Page

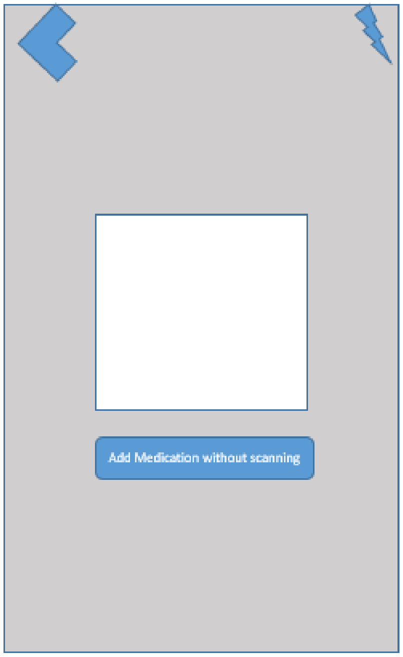


Figure B.3: Design of the Scanning Page

B.4 Medication Add Popup

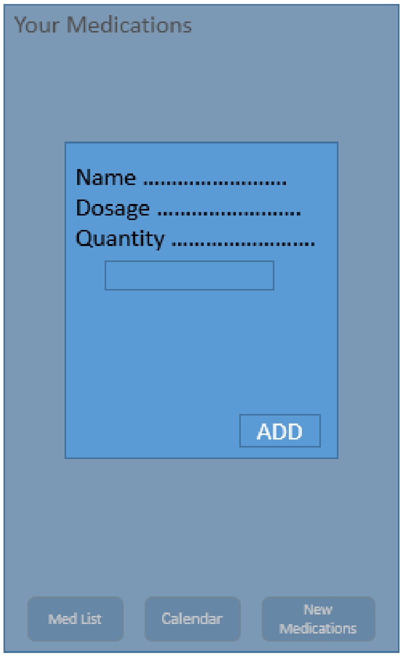

Your Medications
Name ....
Dosage ....
Quantity ....
ADD
Med List
Calendar
New Medications

Figure B.4: Design of the Medication Add Popup

B.5 Medication List Page

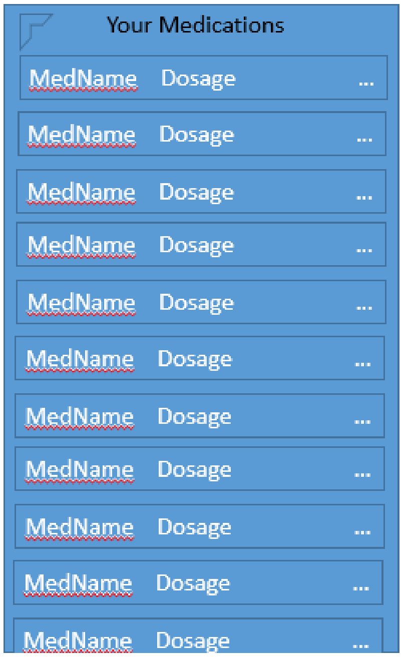


Figure B.5: Design of the Medication List Page

B.6 Medication List Option Popup

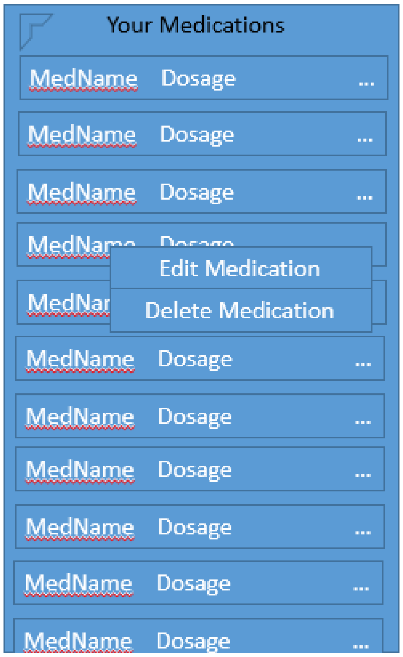


Figure B.6: Design of the Medication List Option Popup

B.7 Medication Edit Page

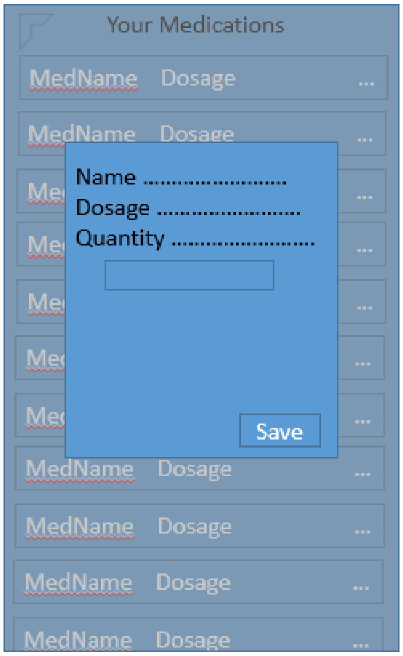


Figure B.7: Design of the Medication Edit Page

B.8 Calendar Page

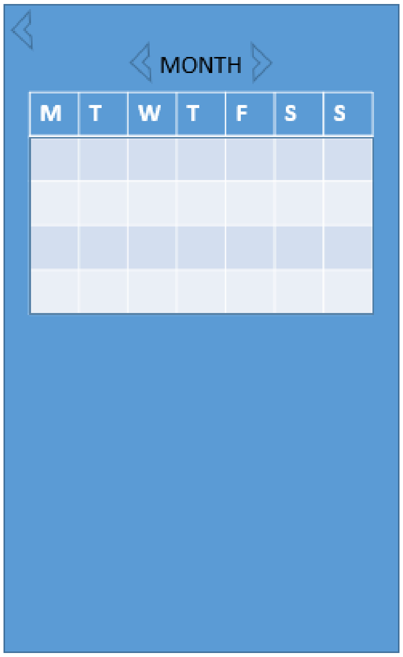


Figure B.8: Design of the Calendar Page

B.9 Medication Taken Popup

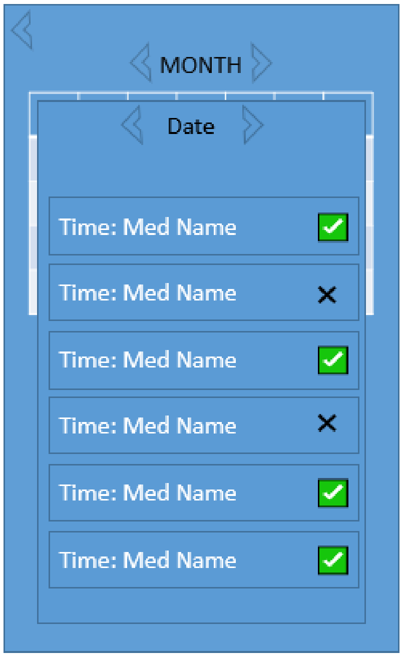
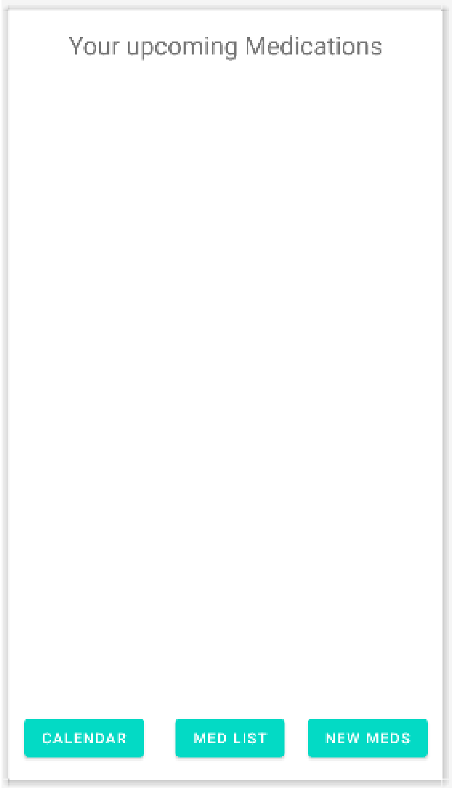


Figure B.9: Design of the Medication Taken Popup

Appendix C
Application User Interface

This appendix shows images of the user interface once developed for the application.

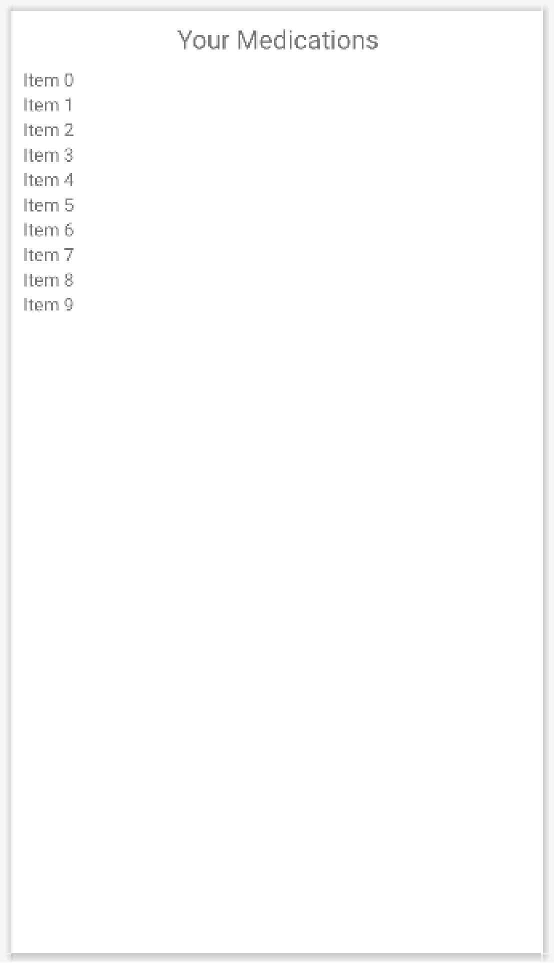
C.1 Main Page



Your upcoming Medications

Figure C.1: Developed UI of the Main Page

C.2 Medication List Page

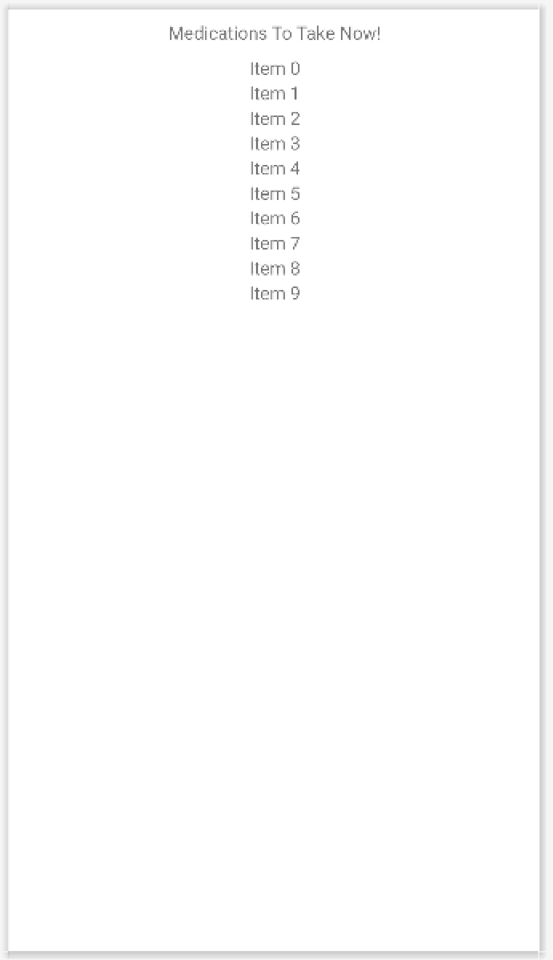


Your Medications

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

Figure C.2: Developed UI of the Medication List Page

C.3 Medication to Take



Medications To Take Now!

Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9

Figure C.3: Developed UI of the Medication to Take Page

C.4 Scan Medication

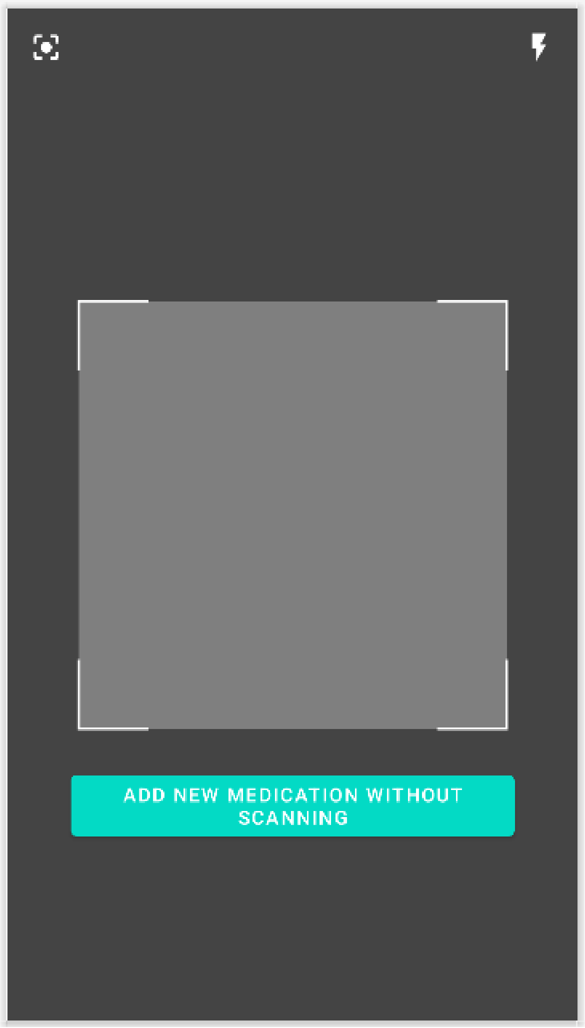
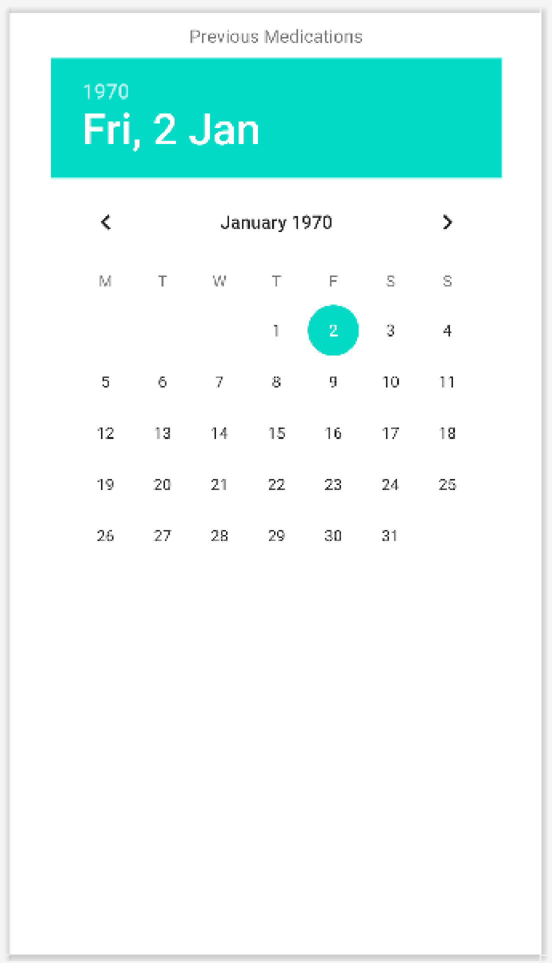


Figure C.4: Developed UI of the Scanner Page

C.5 Calendar Page



Previous Medications

1970

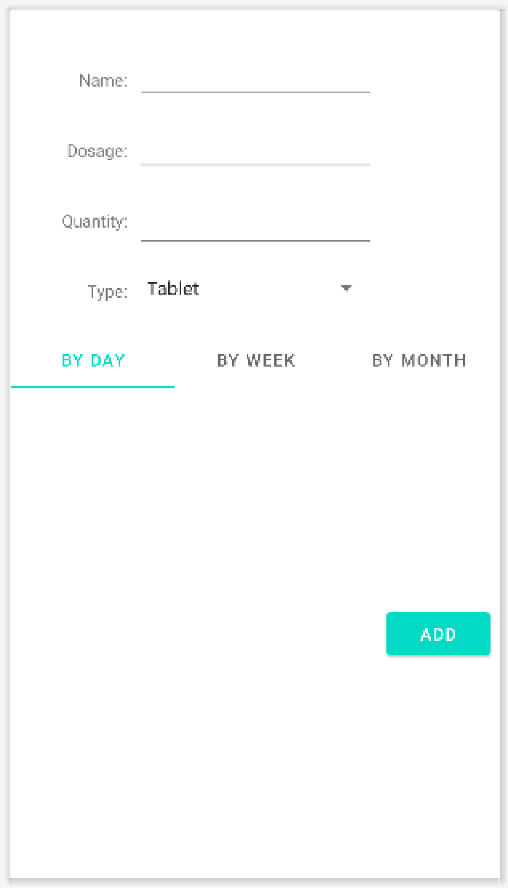
Fri, 2 Jan

January 1970

M	T	W	T	F	S	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Figure C.5: Developed UI of the Calendar Page

C.6 Add Medication Popup



Type: Tablet

Figure C.6: Developed UI of the add Medication Popup

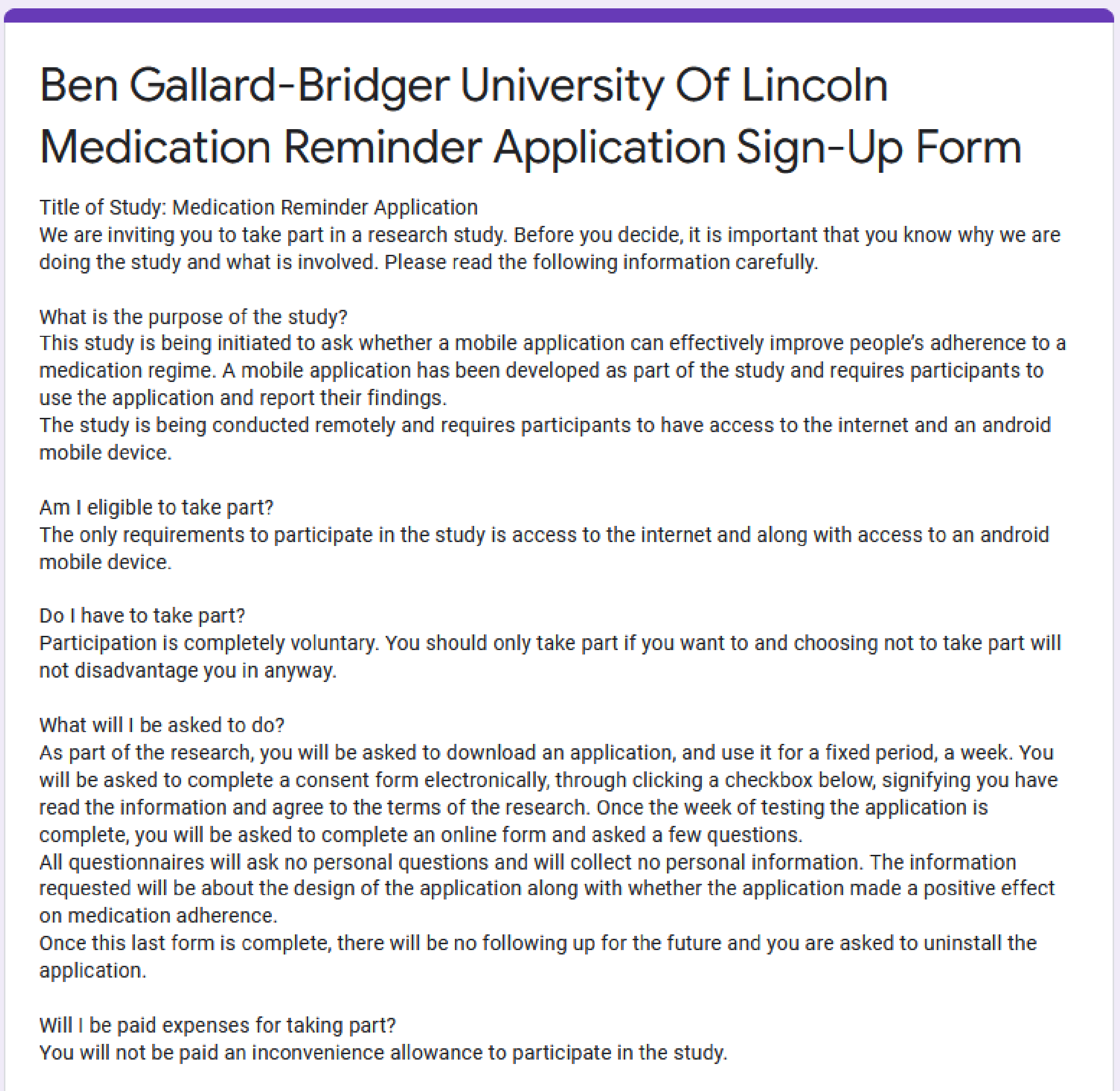
Appendix D
Application Code

Due to the size of the project, the code is hosted on GitHub and can be found in the repository at: `https://github.com/BenGallardBridger/Project`

Appendix E

Sign Up Form

This appendix contains the images showing the sign up form for the study.



Ben Gallard-Bridger University Of Lincoln Medication Reminder Application Sign-Up Form

Title of Study: Medication Reminder Application
We are inviting you to take part in a research study. Before you decide, it is important that you know why we are doing the study and what is involved. Please read the following information carefully.

What is the purpose of the study?
This study is being initiated to ask whether a mobile application can effectively improve people's adherence to a medication regime. A mobile application has been developed as part of the study and requires participants to use the application and report their findings.
The study is being conducted remotely and requires participants to have access to the internet and an android mobile device.

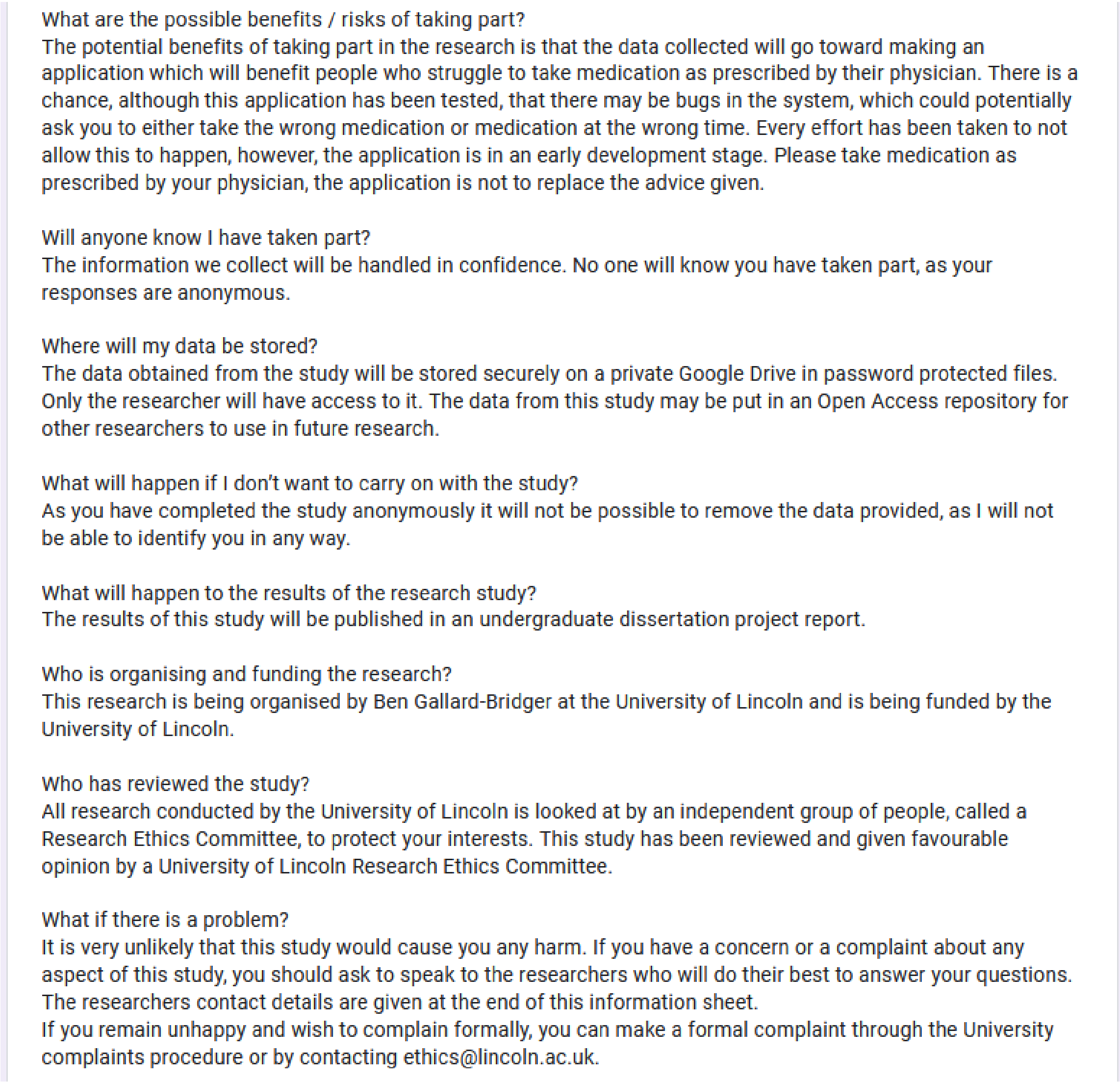
Am I eligible to take part?
The only requirements to participate in the study is access to the internet and along with access to an android mobile device.

Do I have to take part?
Participation is completely voluntary. You should only take part if you want to and choosing not to take part will not disadvantage you in anyway.

What will I be asked to do?
As part of the research, you will be asked to download an application, and use it for a fixed period, a week. You will be asked to complete a consent form electronically, through clicking a checkbox below, signifying you have read the information and agree to the terms of the research. Once the week of testing the application is complete, you will be asked to complete an online form and asked a few questions.
All questionnaires will ask no personal questions and will collect no personal information. The information requested will be about the design of the application along with whether the application made a positive effect on medication adherence.
Once this last form is complete, there will be no following up for the future and you are asked to uninstall the application.

Will I be paid expenses for taking part?
You will not be paid an inconvenience allowance to participate in the study.

Figure E.1: Section one of the Sign Up Form



What are the possible benefits / risks of taking part?
The potential benefits of taking part in the research is that the data collected will go toward making an application which will benefit people who struggle to take medication as prescribed by their physician. There is a chance, although this application has been tested, that there may be bugs in the system, which could potentially ask you to either take the wrong medication or medication at the wrong time. Every effort has been taken to not allow this to happen, however, the application is in an early development stage. Please take medication as prescribed by your physician, the application is not to replace the advice given.

Will anyone know I have taken part?
The information we collect will be handled in confidence. No one will know you have taken part, as your responses are anonymous.

Where will my data be stored?
The data obtained from the study will be stored securely on a private Google Drive in password protected files. Only the researcher will have access to it. The data from this study may be put in an Open Access repository for other researchers to use in future research.

What will happen if I don't want to carry on with the study?
As you have completed the study anonymously it will not be possible to remove the data provided, as I will not be able to identify you in any way.

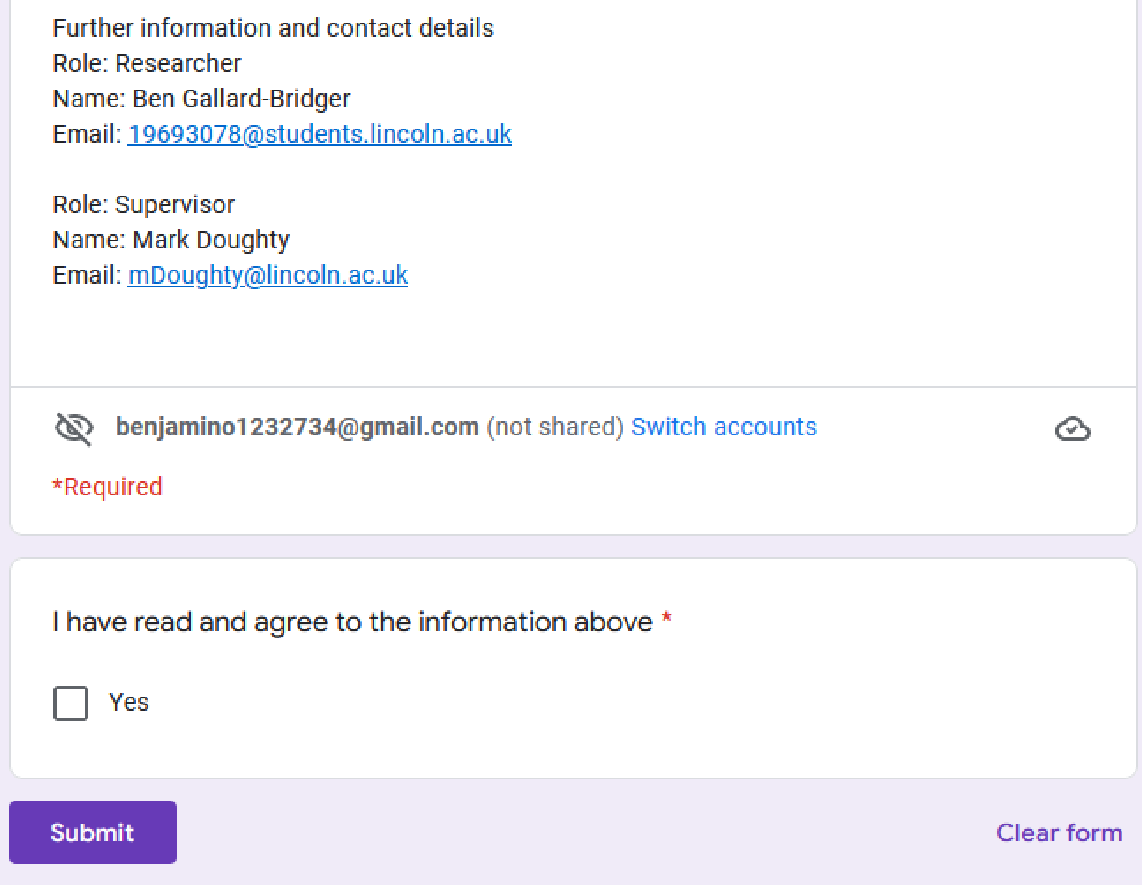
What will happen to the results of the research study?
The results of this study will be published in an undergraduate dissertation project report.

Who is organising and funding the research?
This research is being organised by Ben Gallard-Bridger at the University of Lincoln and is being funded by the University of Lincoln.

Who has reviewed the study?
All research conducted by the University of Lincoln is looked at by an independent group of people, called a Research Ethics Committee, to protect your interests. This study has been reviewed and given favourable opinion by a University of Lincoln Research Ethics Committee.

What if there is a problem?
It is very unlikely that this study would cause you any harm. If you have a concern or a complaint about any aspect of this study, you should ask to speak to the researchers who will do their best to answer your questions. The researchers contact details are given at the end of this information sheet.
If you remain unhappy and wish to complain formally, you can make a formal complaint through the University complaints procedure or by contacting ethics@lincoln.ac.uk.

Figure E.2: Section two of the Sign Up Form



Further information and contact details
Role: Researcher
Name: Ben Gallard-Bridger
Email: 19693078@students.lincoln.ac.uk

Role: Supervisor
Name: Mark Doughty
Email: mDoughty@lincoln.ac.uk

benjamino1232734@gmail.com (not shared) Switch accounts

*Required

I have read and agree to the information above *

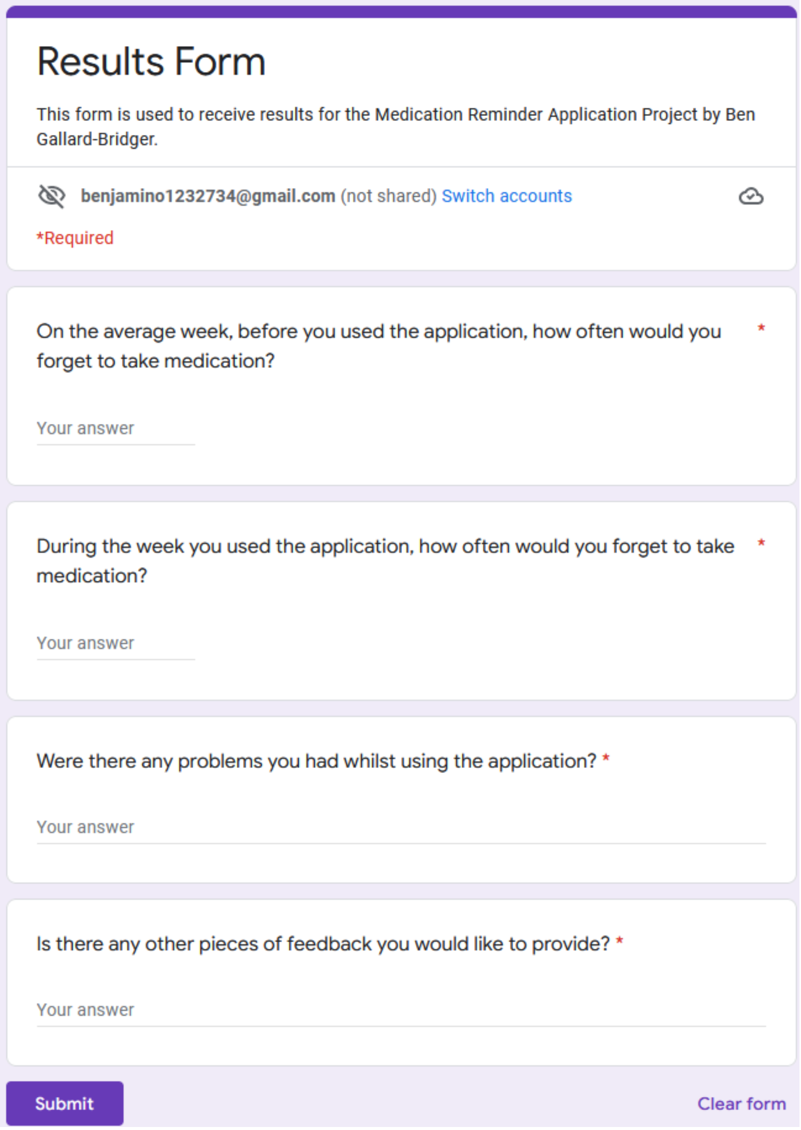
□ Yes

Figure E.3: Section three of the Sign Up Form

Appendix F

Results Form

This appendix contains the images showing the result form for the study.



Results Form

This form is used to receive results for the Medication Reminder Application Project by Ben Gallard-Bridger.

benjamino1232734@gmail.com (not shared) Switch accounts

*Required

On the average week, before you used the application, how often would you forget to take medication? *

Your answer

During the week you used the application, how often would you forget to take medication? *

Your answer

Were there any problems you had whilst using the application? *

Your answer

Is there any other pieces of feedback you would like to provide? *

Your answer

Figure F.1: Image of the Results Form

Appendix G
Results

G.1 Question Number Assignment

For the ease of the document, Each question will be assigned a number which is shown below.

Question 1: On the average week, before you used the application, how often would you forget to take medication?

Question 2: During the week you used the application, how often would you forget to take medication?

Question 3: Were there any problems you had whilst using the application?

Question 4:Is there any other pieces of feedback you would like to provide?

G.2 Participant 1

G.2.1 Question 1

3

G.2.2 Question 2

1

G.2.3 Question 3

The application was simple and nice to use.

G.2.4 Question 4

i like that the app uses fingerprint to protect the users data

G.3 Participant 2

G.3.1 Question 1

3

G.3.2 Question 2

1

G.3.3 Question 3

I had a few issues with the reminders however this was due to user error and not the developed application.

G.3.4 Question 4

The use of the secure login was a really nice additional feature as it felt safer to use. However the overall design could have been slightly clearer as with some of the txt it wasn't very clearly displayed.

G.4 Participant 3

G.4.1 Question 1

2

G.4.2 Question 2

0

G.4.3 Question 3

Yes, I wasn't able to edit or delete any medication that I had already added. So if I made a mistake when entering details I had to add another medication. Also, when notifications went off for multiple medications, they'd all override each other so I could only mark 1 medication as taken and the other notifications would disappear. I sometimes had a problem when scanning the barcode for medication, the camera wasn't able to focus each time and I had to close the app and reopen for it to finally work. I also wasn't able to view my medication on the upcoming medication page, so I wasn't able to prepare for what medication I needed with me each day.

G.4.4 Question 4

Apart from removing the issues above, I'd find it really useful to have a feature that allows you to enter the amount of meds left and sends notifications when running low to get a new prescription.

G.5 Participant 4

G.5.1 Question 1

0

G.5.2 Question 2

19

G.5.3 Question 3

- Wasn't able to edit or remove any medication
- Only appeared once where I could tick that I had taken the medication (however when the orientation changed on the page the tick box unticked, I reticked it another 2 times, so said that I had taken the medication 3 times that day in the calendar
- If I wasn't able to tick that I had taken the medication nothing in the app would allow me to find that page and add it onto the calendar or confirm I had taken it
- notifications were inconsistent and only received about 2 a week when it was set to receive 3 a day
- camera capture wasn't working for any of the medication that I tried
- didn't ask for passcode each time the app was opened so could still be accessed by others if the app is open in the background
- add new medication screen doesn't all fit, writing is cut off
- add new medication dosage/quantity wasn't very clear what was required regarding input for each dose or pack of tablets
- every x days asks for number of days but doesn't specify if it means every 2 days or receive notifications for only 2 days
- no notifications if no time is inputted for new tablet but no warning message to say you need to input a time
- by the month has a title of 'textview' rather than an actual title
- no notification for when to buy more medication when low

G.5.4 Question 4

No

Appendix H

Examples of JSON Files

```
{"data":
    {
        "name": "Mirtazapine",
        "strength": "30",
        "units": "mg",
        "type": "tablet",
        "quantity": "28"
    },
    "message": "Success",
    "status": "success",
    "error": false
}
```

Listing 1: JSON Success example

```
{
    "data": null,
    "message": "Error: No drug data found for that GTIN",
    "status": "fail",
    "error": true
}
```

Listing 2: JSON Fail example