



BSc Computer Science

**CM2020 Agile Software Projects
Final Report**



“DoseManager”

A user-centric mobile application that manages medications.

Team 66 members:

LEE CHANG LOONG BENJAMIN - Team Leader (UOL ID:)

GOH ZHONG DA (UOL ID:)

TAN JUN RONG (UOL ID:)

SHAMBHAVI PALANISAMY (UOL ID:)

HENRY ZHANG JIAHUA (UOL ID:)

Module Leader: Dr Sean McGrath

Table of contents

Abstract.....	4
Introduction.....	4
Planning and Research.....	5
Breakdown of Roles.....	5
Development Methodology.....	5
Project Management Tools.....	5
Gantt Chart.....	7
Resource and Time Allocation.....	8
Apache Cordova Framework.....	8
Frontend.....	8
Backend.....	10
Database Schema Diagram.....	12
Literature & Market Research.....	13
Design and Development.....	14
High-fidelity wireframe.....	14
User Testing Results.....	14
First iteration (Mid-fidelity Prototype).....	14
Conclusion.....	19
Second iteration (High-Fidelity prototype).....	20
Frontend (User Interface).....	20
Backend (User Experience).....	29
Conclusion.....	35
Final iteration (Final working prototype).....	36
Frontend (User Interface).....	36
Backend (User Experience).....	39
Conclusion.....	42
Design choices.....	43
Colour Scheme.....	43
Logo.....	44
Components of the software.....	44
Presentation Layer.....	44
Application Layer.....	44
Data Layer.....	44
Structure and Implementation.....	45
Setting up Cordova environment for development and testing.....	46
Firebase Cloud Firestore.....	48
API Calls.....	50
Libraries and plugins used.....	52
Evaluation.....	58
Frontend.....	58
Design Issues.....	58

Backend.....	61
Technical issues.....	61
Overall.....	69
Conclusion and Summary.....	70
Frontend.....	70
Backend.....	70
Overall.....	71
References.....	72
Individual reflections.....	73

Abstract

This report presents all the resources and information collected and the process and steps taken to research the critical concepts behind *DoseManager*, a medicine tracking and reminder Android application. This report will also include the documentation of our group's entire journey from start to end whilst building *DoseManager*. We will begin by talking in depth about the central concept as well as the aims and objectives, followed by the requirements, the market and literature research done, earlier wireframes and designs in the initial phases of the project, the analysis and outcome of our research, the different prototypes we built, and the feedback received from user testing, and finally an analysis of the final product itself.

Introduction

Our team embarked on this journey of building *DoseManager* as we recognised the importance of managing our health and well-being and the implications of poor medication adherence. The aim of our project has been clear from the beginning – to provide a user-friendly application to aid users in keeping track of their medications. The primary stakeholders of *DoseManager* are elderly Singaporean patients with language barriers (such as being not fluent in English) and busy professionals who consume medicine regularly. The secondary stakeholders are mainly caretakers of these elderly or close family relatives.

The team, split into the frontend and backend groups, has focused on bringing our ideas to fruition. The backend team focused on building the backbone of the application through API integration, database development, and middleware connection, while the frontend team focused on portraying information and designing the application attractively yet concisely based on different feedback gathered through user testing. We will begin this report by diving into the planning and research for building *DoseManager*.

Planning and Research

Breakdown of Roles

The team consists of 5 members named at the beginning of this report. Our team split into two smaller groups to allow for better time management and allocation of roles, and to prevent conflict between members. Although most of us are more familiar with the front end, 3 group members were more confident in their technical abilities, thus taking on the backend tasks. The other two members worked on the frontend and we worked on the report collectively. The two teams held occasional meetings within their groups to discuss and ensure that everything was functional. Our whole team also had weekly meetings over Discord to catch up on our progress and ensure we were on track to complete the project.

Development Methodology

Our team met to discuss and decide on the approach used to develop DoseManager. We chose the *Agile methodology* as it was taught in this module, and we wanted to apply what we learned to develop our application. The agile methodology focused on individual interaction, where *User-Driven Development* was employed, end-user feedback was consistently collected and reviewed, and the necessary changes were integrated into *DoseManager*. We broke down the different phases of the project into smaller tasks, which provided our team with greater visibility and understanding of the product. This helps increase efficiency as team members can better understand and avoid downtime in seeking clarifications. Incorporating regular feedback loops and embracing iterative development were pivotal aspects of our Agile approach. This allowed us to maintain continuous improvement and adaptability throughout the development process.

Project Management Tools

Our team used GitHub and Git version control tools concurrently to maintain our codebase. We had a main repository containing all of our combined codes, and we created individual branches based on the different pages we each worked on (Figure 1). This allowed us to work on different pages simultaneously without interfering with each other's work. Once we were done working on our code in our individual branches, we would submit a pull request to merge our code into the main branch. As a team, we would then review the code to see if there were any issues, whether the code was functional, and for any potential conflicts. If there were no issues, the team leader would then merge it into the local main repository before pushing a new one to the remote repository. We would also give each other feedback or debug each other's code through Discord screen share or via the VSCode Live Share extension.

main ▾ 11 branches 0 tags

Switch branches/tags

Find or create a branch...

Branches Tags

✓ main default

- Page-1---select-profiles
- Page-2--Homepage
- Page-3--Add-medication
- Page-4--Medication-image
- Page-5--Profile
- Page-6--Upcoming-appointments
- Page-7--chatbot
- Page-8--Personal-info
- Page-9---settings

[View all branches](#)



Figure 1 - GitHub repository branches

Gantt Chart

DOSEMANAGER Project Timeline

PROJECT TITLE	DoseManager
PROJECT MEMBERS	LEE CHANG LOONG BENJAMIN - Team Leader GOH ZHONG DA TAN JUN RONG SHAMBHAVI PALANISAMY HENRY ZHANG JIAHUA

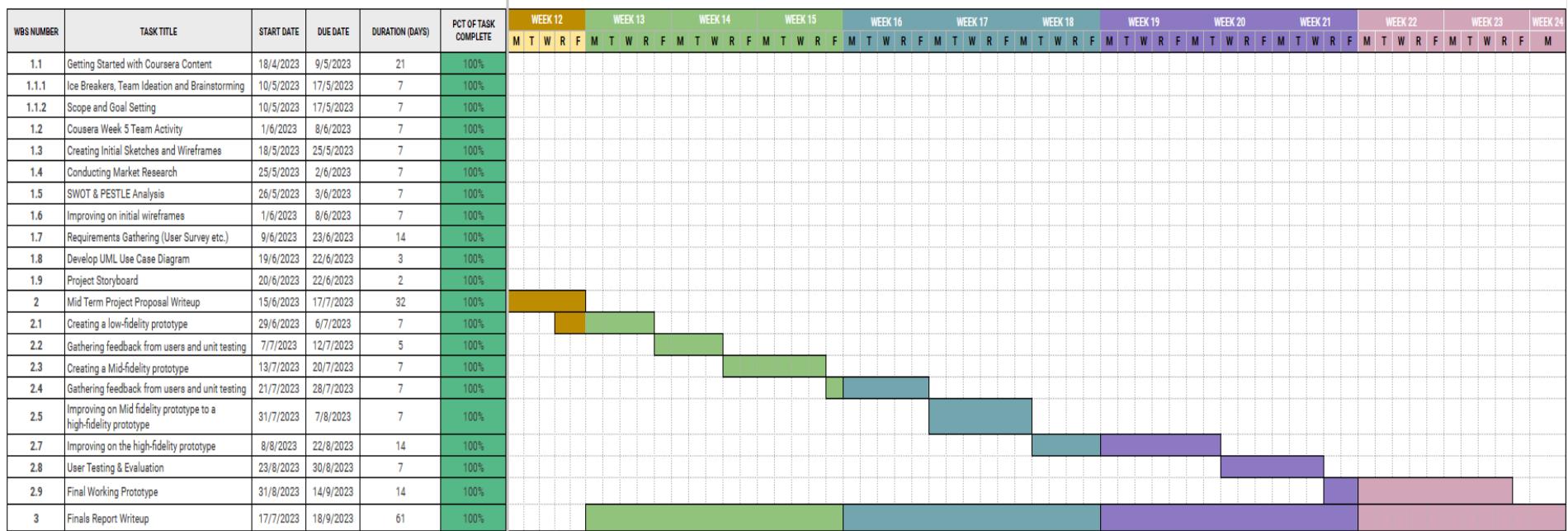


Figure 2 - Gantt Chart

The above Gantt chart offers a visual representation of our workflow progress over 22 weeks. This particular chart was generated utilising Google Sheets.

Resource and Time Allocation

Apache Cordova Framework

As highlighted in our midterm report, our team opted for Cordova as the framework for building our Android application. This choice was motivated by the need for a rapid development cycle and a minimal learning curve, given that most team members lacked proficiency in native mobile development languages like Flutter or React. Cordova facilitated our Android app development by enabling us to harness native features through JavaScript APIs, all while utilising vanilla HTML, CSS, and JavaScript. This approach allowed us to expedite the development process without requiring extensive expertise in mobile app development.

Frontend

For our front-end development, we mainly used Bootstrap 5 as our main CSS framework to design the overall layout of our application. Bootstrap provides ready-made components and styles that can be customised and combined. However, since it had been a while since we used Bootstrap, our front-end team spent the first two weeks (Week 12 and Week 13) reading up on the Bootstrap documentation to refresh and familiarise themselves with the Bootstrap framework and researching user interface principles to better design the layout for the user to have a pleasant experience while using our application.

<u>DoseManager Project “Frontend” Team</u>			
Tasks	Duration	Discussion Points	Participants
Refreshing knowledge regarding using Bootstrap	2 weeks	Reviewed Bootstrap documentation and learnt about the best practices. Brushed up on Bootstrap features, such as naming the classes in HTML, their different grid sizes and positioning	Shambhavi Henry
Refreshing knowledge on CSS	1 week	Reviewed the CSS fundamentals using the help of W3C.	Shambhavi Henry
Working on the first iteration of <i>DoseManager</i>	1 week	Defined and discussed the design requirements and worked on them as a team	Shambhavi Henry
Conducting user testing and researching methods to make our application more appealing	1 week	Gathered user feedback and researched the UI/UX improvements	Shambhavi Henry
Working on the second iteration of <i>DoseManager</i>	1 week	Implement changes based on feedback gathered	Shambhavi Henry
Conducting user testing and researching methods	1 week	Continued user testing and explored additional improvements	Shambhavi Henry

to make our application more appealing		to be made.	
Working on the final iteration of <i>DoseManager</i>	1 week	Implement changes based on feedback gathered	Shambhavi Henry

Table 1 - Frontend Tasks Allocation

The screenshot shows the official Bootstrap 5 documentation website. The top navigation bar includes links for 'Docs', 'Examples', 'Icons', 'Themes', and 'Blog'. A search bar and a 'View on GitHub' button are also present. The main content area features a large heading 'Get started with Bootstrap'. Below it, a sub-section titled 'Quick start' provides instructions for setting up Bootstrap. A code block displays the HTML code for a basic Bootstrap demo page, including the meta viewport tag and an H1 header. On the left, a sidebar contains navigation menus for 'Getting started' (with 'Introduction' selected) and 'Customize' (with 'Overview' selected). On the right, a 'On this page' sidebar lists various documentation topics like 'CDN links', 'Next steps', and 'Important globals'.

Figure 3 - Bootstrap 5

Additionally, we also used specific and custom CSS styling for the different pages of our application in order to give it a more refined and consistent look across the application. Using custom CSS styling allowed us to design the user interface (UI) to meet the needs and preferences of our target audience. It also allowed us full control and flexibility over designing our application as we were able to experiment with different layouts, colour schemes, etc. Furthermore, building custom CSS files for our application allowed us to hone and develop our skills while maintaining reduced dependencies on external frameworks and libraries.

Backend

The backend team needed to be more familiar with backend development and integration into Android. Therefore, they spent longer studying and learning the programs that would be used. The backend team focused mainly on Cordova plugins that would interact and display Android notifications for reminders and a database that could be used to retrieve and display data for *DoseManager*.

DoseManager Project “Backend” Team				
Tasks	Duration	Complete	Discussion Points	Participants
Searching for usable open-sourced APIs to integrate to the application	1 week	Found different APIs to use such as RxTerms from NLM, to autocomplete medication data	Found many different APIs however certain ones were not open-sourced and they took awhile to get back to our emails.	Benjamin Jun Rong Zhong Da
Researching, and learning about: git, JavaScript, SQLite, Cordova, Firebase.	2 weeks	Backend team members became familiar with the different languages and skills required.	Had little to no knowledge on cordova and firebase, thus spent a while getting familiar with them.	Benjamin Jun Rong Zhong Da
Working on our own pages	2 weeks	Worked on programming the backend component for our respective assigned pages	Benjamin - Profiles Jun Rong - Medication Zhong Da - Appointments	Benjamin Jun Rong Zhong Da
Exploring another way to store our input data rather than using SQLite (Cloud Firestore)	1 week	SQLite had compatibility issues with Cordova as it was mostly outdated, therefore we decided to use Cloud Firestore	There were many issues when using the cordova SQLite plugin as it was outdated thus had many compatibility issues also with the other stuffs we were using	Benjamin Jun Rong Zhong Da
Continued working on our own pages	2 weeks	Worked on changing the SQLite backend database to Cloud Firestore for our pages	Had to learn how to integrate Cloud Firestore into our code, using it as our database	Benjamin Jun Rong Zhong Da
Consolidating our pages together	1 week	Consolidated our worked on pages together and spent the week debugging errors	Had a few bugs when we consolidated our pages together; certain namings were different. We also had weren't sure how to link up our pages to each unique profile	Benjamin Jun Rong Zhong Da
Research and implementation of	1 week	Found a new Cordova Plugin Library that	Realised that Firebase Cloud Messaging required	Benjamin Jun Rong

local notification library		supported Android 12 and above and could customise actions based on our notification logic	a premium plan to use Cloud Functions for notification scheduling. Had to switch to local notifications and source for a newer library.	Zhong Da
Researched on how to link the pages up together & linking them up	1 week	Understood that we had to create sub-collections inside the main collection to link the items to each individual unique profile	Did further research on linking the different collections together.	Benjamin Jun Rong Zhong Da
Re-consolidated the pages together	1 week	Consolidated our work on pages together once again and continued debugging errors	Consolidated the pages together once again and debugged the errors together as a team.	Benjamin Jun Rong Zhong Da

Table 2 - Backend Tasks Allocation

Database Schema Diagram



Figure 4 - Database Schema Diagram

We have a main collection named **Profiles**. Each document within this collection represents an individual user or profile. The unique identifier (ID) for each document in this collection is the **profileID**, which ensures that each user has a distinct presence in the database. Each **profileID** directly corresponds to a unique user. This design ensures that every user's data is isolated and personalised, providing a customised user experience. All user-related data is nested under their respective profile.

Literature & Market Research

From our previous studies done while researching and designing our application in the initial stages, we had found that the World Health Organization (WHO) estimates that 50% of patients tend to not adhere to their medication regimens, thus not reaping the full benefits of their prescribed medications. We also found that a study done by the Duke-NUS Medical School, Singapore, discovered that 83% of Singaporean elderly who are unable to read in English consume at least one medicine while 16% consume more than 5. It also concluded there is a direct link between elderly with low English health literacy and medication non-adherence, and that medication labels ought to include "bilingual text and/or pictograms" to assist the aforementioned elderly.

The table below, which is also included in our mid-term report, summarises our market research done in the initial stages of this project, where we investigated the available medication reminder applications and evaluated their functions, advantages, and disadvantages.

Application Name	Features								
	Push notification reminder to take medication	Push notification for upcoming appointments	Multi-device sync	Multilingual support feature (supports Chinese, Malay Tamil?)	Is the app available on iOS?	Is the app available on Android?	Is the app paid?	Does the app have premium features that are available only when paid for?	App/Play store rating
Pillo	✓	✓	✓	✗	✗	✓	✗	✗	4.7
MediSafe	✓	✓	✓	✗	✓	✓	✗	✓	4.2
CutePill	✓	✗	✓	✗	✓	✓	✗	✓	4.3
MyTherapy	✓	✓	✓	✗	✓	✓	✗	✗	4.9
DoseCast	✓	✓	✓	✗	✓	✓	✗	✓	4.0

Table 3 - Market Research Summary

Therefore, these studies support our desire to build a medication tracking and reminder application with a language change feature that supports the 4 most commonly spoken languages in Singapore - English, Chinese, Malay, and Tamil.

Design and Development

High-fidelity wireframe

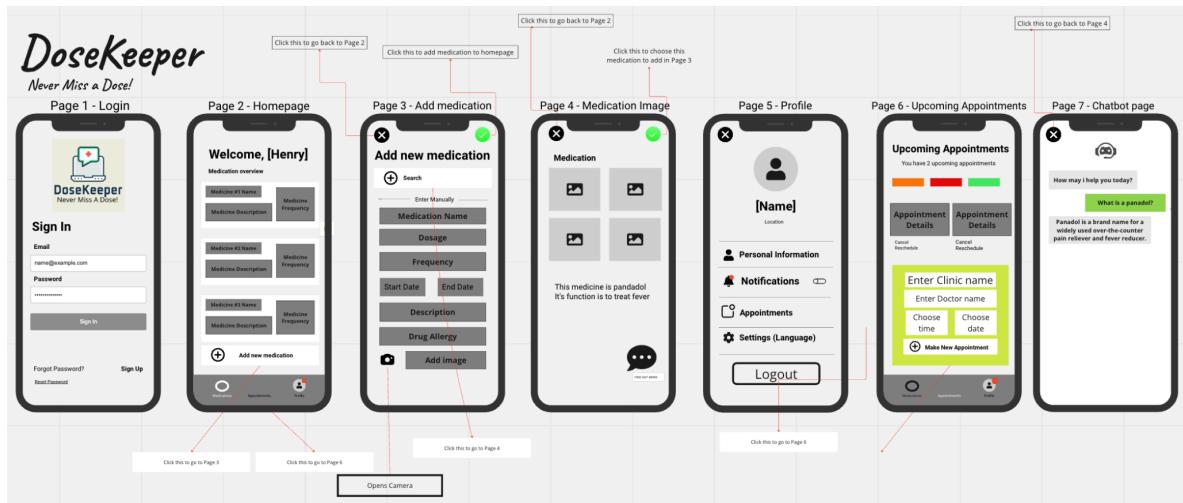


Figure 5 - High-fidelity wireframe

Figure 5 portrays the final high-fidelity wireframe of *DoseManager* we had designed as a team prior to building the app itself. The following user testing conducted and prototypes we coded and developed show our progress and changes over time, how we followed the designs, and why we deviated from them in certain circumstances.

User Testing Results

First iteration (Mid-fidelity Prototype)

We created a survey using Microsoft Forms and questioned participants on their thoughts regarding our application, providing them images of each of our pages and asking for any suggestions or improvements that could be made to our respective pages.



WELCOME

SELECT YOUR PROFILE

- [John Smith](#)
- [Sam Smith](#)
- [Ben Smith](#)

1. How do you find the layout of the Profile login page? Any improvements to be made? (Colours, layout etc.)

7 Responses

ID ↑	Name	Responses
1	anonymous	pretty good
2	anonymous	Simple and not bad.
3	anonymous	good
4	anonymous	not bad
5	anonymous	clear and simple layout
6	anonymous	simple layout
7	anonymous	easy to understand layout.

Figure 6b

Figure 6a

Create A New Profile

First Name:

Last Name:

Gender:

Date of Birth:

2. How do you find the layout of the Create Profile page? Any improvements to be made? (Colours, layout etc.)

7 Responses

ID ↑	Name	Responses
1	anonymous	i love it
2	anonymous	Similar to the first page, simple layout.
3	anonymous	good
4	anonymous	not bad
5	anonymous	simple and clear layout
6	anonymous	nice and simple layout.
7	anonymous	the layout is easy to look at and understandable.

Figure 7b

Figure 7a

As the surveyors unanimously agreed that they liked the layout of the ‘profile login page’ and ‘create new profile’ page due to their simplicity, we decided to stick to the same design, without changing much.



Figure 8a

3. How do you find the layout of the Home page? Any improvements to be made? (Colours, layout etc.)

8 Responses

1	-----	lighter tones.
3	anonymous	don't like the red circle
4	anonymous	not bad
5	anonymous	simple to understand layout, do not like the red circle.
6	anonymous	nice and simple layout.
7	anonymous	The layout is readable, but the black text on the dark red circle may not be the best for other people to see.
8	anonymous	Not a bad design, like that the red stands out, contrasting between the light colors to let the user know it is important

Figure 8b

As for the homepage, the surveyors seemed to like the design and layout itself, but not the colours we chose. Some even recommended we use lighter colours to make the text more readable. Thus, we have decided to stick to the layout but change the choice of colours moving forward.

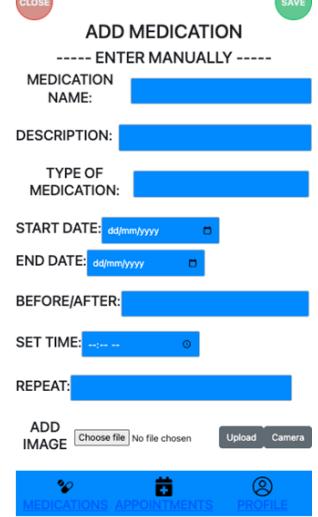


Figure 9a

4. How do you find the layout of the 'Add Medication' page? Any improvements to be made? (Colours, layouts etc.)

8 Responses

2	anonymous	It looks messy as the boxes are all of different size, the texts also do not start from the same place.
3	anonymous	looks like a mess
4	anonymous	weird alignments
5	anonymous	the names of the sections are not aligned. The boxes are of different sizes also.
6	anonymous	messy layout with weird alignment of boxes and texts.
7	anonymous	the alignment of the boxes and the different input areas are not the same.
8	anonymous	Messy

Figure 9b

The surveyors mentioned that the alignment and layout of this page were messy as there was no consistency. The team therefore agreed to design it in a simpler way with less distracting colours and a simpler yet organised layout.

Medication Name

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper, tellus id finibus luctus, erci arcu elementum nulla, et dictum odio lacus id leo. Integer ut elementum nulla. Curabitur laoreet urna quis euismod venenatis. Nullam eget luctus massa. Morbi risus quam, vehicula sed elit maximus, fermentum porttitor neque. Aliquam feugiat facilis tellus, luctus fermentum libero vestibulum vulputate. Ut non urna erat. Proin eget tempor nulla. Pellentesque accumsan varius orci nec auctor. Sed quis ipsum ac nisl sollicitudin vulputate ac aliquet ante. Quisque vitae ligula nibh.

Figure 10a

5. How do you find the layout of the 'Medication Info' page? Any improvements to be made? (Colours, layouts etc.)		
8 Responses		
1	anonymous	no improvements is needed
2	anonymous	Simple layout, not bad.
3	anonymous	good
4	anonymous	good
5	anonymous	good
6	anonymous	simple layout.
7	anonymous	simple and easy to look at layout.
8	anonymous	Not bad.

Figure 10b

Surveyors agreed that this page had a simple layout that was easy to read and not distracting to look at. However, the team decided that moving forward, we will not be including this page in our application due to its redundancy and make do with a manual input medication description to be included in the add medicine page. Additionally, we were also unable to find a medicine description API to use. More information can be found under the technical issues that we faced.

John Doe
Singapore

Personal Information

Notifications

Upcoming Appointments

Settings (Language)

Change Profile

Figure 11a

6. How do you find the layout / user interface of the Profile page? Any improvements to be made? (Colours, layout etc.)		
8 Responses		
2	anonymous	Simple layout, prefer to have some colors around or maybe have everything aligned to the left instead of aligned to the middle.
3	anonymous	not bad
4	anonymous	not bad
5	anonymous	not bad
6	anonymous	simple layout.
7	anonymous	simple and understandable layout. easy to look at.
8	anonymous	Not bad, quite nice.

Figure 11b

The surveyors mentioned that although this page had a nice and simple layout that was not too distracting, adding some colours and making it more colourful would make it attractive. The team agreed on using the base colour of the application (Hex Code: #0096ff) for this page.

Upcoming Appointments

You have XX upcoming appointments and XX missed appointments.

Upcoming Appointment	Missed Appointment
----------------------	--------------------

Appointment Name
Appointment Location & Doctor's Name goes here

Edit

Appointment Name
Appointment Location & Doctor's Name goes here

Edit

Add New Appointment

Medications
 Appointments
 Profile

7. How do you find the layout of the Upcoming Appointments page? Any improvements to be made? (Colours, layout etc.)

8 Responses

1	anonymous	all the information displayed properly
2	anonymous	Simple layout, colors may need some work.
3	anonymous	good
4	anonymous	good
5	anonymous	simple to understand layout.
6	anonymous	clear layout.
7	anonymous	Simple and easy to understand and readable layout.
8	anonymous	Quite nice.

Figure 12a

Figure 12b

The surveyors all liked the layout of this page, and how the required information was neatly displayed on the page. Hence, the team agreed to not make any frontend changes (unless required) to this page.

Create Appointment

Clinic Name:

Level Number:

Department:

Doctor's Name:

Choose Appointment Time

Choose Appointment Date

Save

Medications
 Appointments
 Profile

8. How do you find the layout of the Create Appointment page? Any improvements to be made? (Colours, layout etc.)

8 Responses

2	anonymous	Simple layout, easy to understand.
3	anonymous	good
4	anonymous	good
5	anonymous	clear and simple layout.
6	anonymous	easy to understand layout.
7	anonymous	Simple and easy to understand and readable layout. May be able to bold the input sections similar the other page to make it more understandable.
8	anonymous	Not bad

Figure 13a

Figure 13b

The surveyors liked the simple layout and design of this page as it was easy to read and understand for them, allowing us to stick to the same design, and we have decided to bold the input words in the end.

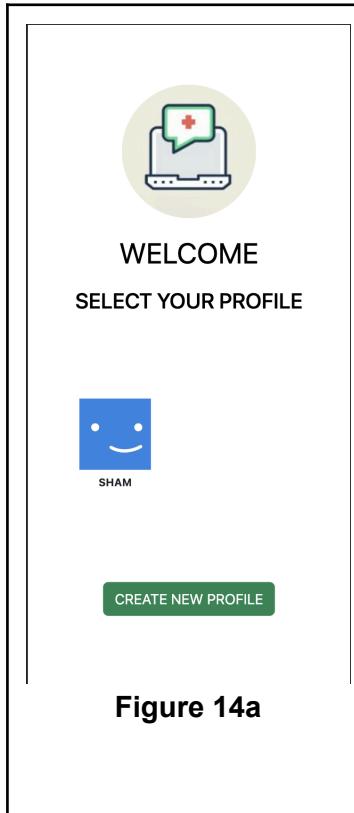
Conclusion

Based on the first round of user testing we conducted, the team agreed to make the changes according to the surveyors' feedback. We will be changing the colour scheme accordingly and will be applying it to all the pages to ensure a consistent outlook across the different pages. We will also be simplifying the layout on certain pages to make it easier on the users' eyes.

Second iteration (High-Fidelity prototype)

Frontend (User Interface)

Like we did with the first iteration, our team conducted a survey to gather feedback and improvements for our application's UI upon making improvements from the first user testing. The results and findings are presented in the following images.



Profile Login Page
Are there any improvements or feedback for the profile login page?
looks very bland. Could use more colors
background looks plain
The profile page looks simple and easy to use
different color for background
Not bad page, simple and easy to understand
looks neat and tidy
it is neat and simple, i like it, it is straightforward
-
Pretty good.
no improvements
the colour contrast is pretty good
Not a bad layout, quite simple and clean looking, with colors to differentiate the users.
Clean and nice.
quite simple
simple and plain, i like it
simple and neat

Figure 14b

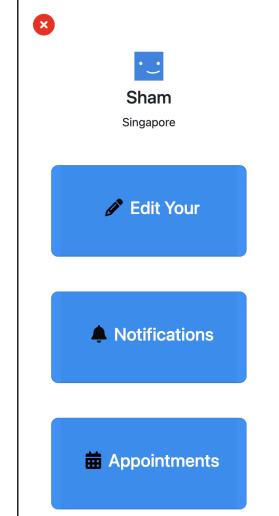


Figure 15a

Profile Page

Are there any improvements or feedback for the profile page?

Buttons are too big
font sizes don't match. buttons are too big
setting buttons looks way too big
buttons are too big and profile picture looks small
The buttons for the respective items are too big.
no improvements
buttons can be smaller
buttons sizing is not suitable, too large
Simple and clean layout, easy to understand. The buttons may need some resizing as they are too big.
different colours and make the buttons smaller
profile image cannot be displayed?
There is too much "blue space" between each button.
Clean and nice.
colour is nice but buttons are too big

good, i like
different colour buttons would make it look nicer

Figure 15b

The profile page was not well liked by our users, with most of them mentioning that the buttons were too big and that it was not suitable for general use.

Add Medication

Medicine Name:

Medicine Description:

Type of medication:

Amount:

Before/After:

Set Time:

Frequency:

Submit

Figure 16a

Add Medication Page

Are there any improvements or feedback for the add medication page?

looks simple and easy to use

Maybe have a cancel button instead of a cross

seems plain but form has all the required data i would need

This is okay, similar to the other page.

no improvements

no improvements

-

Simple and clean layout, easy to understand.

no improvements

so far no, the form is easy to fill in

Simple and minimalistic layout, easy to understand and look at.

Clean and nice.

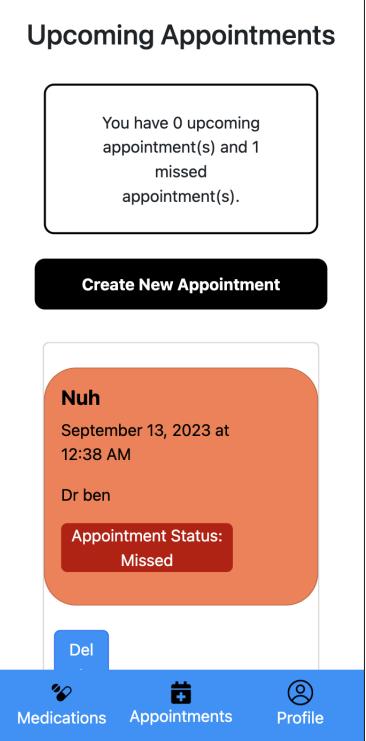
simple, easy to use

simple and good, i like it

neat and direct to the point

Figure 16b

The general consensus for our Add Medication Page was positive, with most users mentioning that our page was simple, clean, and easy to use.



Upcoming Appointments

You have 0 upcoming appointment(s) and 1 missed appointment(s).

Create New Appointment

Nuh
September 13, 2023 at 12:38 AM
Dr ben
Appointment Status: Missed

Del

Medications Appointments Profile

Figure 17a

Upcoming Appointments Page

Are there any improvements or feedback for the upcoming appointments page?

16 responses

- no improvements
- button color can be changed and delete button needs to be resized
- the rectangle box behind the appointment can be removed
- change button color and have a different color for the appointment box at the top
- rectangle outline in the background can be removed. delete button should be resized as well
- The colors need to be changed, similar to the other page. The delete button has sizing issues.
- looks simple, i like it
- delete button can be resized. create new appointment button color is weird.
- Simple and clean layout, easy to see what is going on. Similar to the other page, the button may need some change to the color, and the delete button needs to be resized.
- can use different colours
- status green, is upcoming, status red is missed, very clear
- Delete button needs to be resized as the "Delete" text is wrapped down. The black button doesn't fit well with the rest of the page. Other than that it is good.
- Clean and nice, but the delete text is too large for the button, therefore it wrapped downwards.
- i like this, simple and nice
- can make the cards smaller

Figure 17b

The Upcoming Appointments page also received positive feedback. However, some feedback also mentioned that our Add New Appointment button colour was weird and did not match with the page and that our Delete button text was wrapped downwards and did not fit into a normal-sized button. A few users also brought up the point that we could possibly change our colours for the page.

Create An Appointment

Appointment Location

Appointment Date & Time
 dd/mm/yyyy, --:-- --

Doctor Name

Set Reminder
 5 minutes before

Save **Cancel**

Figure 18a

Create Appointment Page

Are there any improvements or feedback for the create appointments page?

-
-
looks plain but other than that the form has the required details
It is okay, same thoughts as the other page, simple to understand.
no improvements
no improvements
save button and cancel button can be larger
Simple and clean layout, easy to understand what is going on.
no improvements
appointment, date time proper format
Simple and minimalistic layout, easy to understand and look at.
Clean and nice.
nice
simple but save button and cancel button can be larger
neat, no improvements

Figure 18b

Most users found this Create Appointment page simple and easy to understand even though it was plain.

<p>Language Settings</p> 	<p>Language Settings Page</p> <p>Are there any improvements or feedback for the language settings page?</p> <p>16 responses</p> <ul style="list-style-type: none"> no improvements save button color can be changed change the color of save button save button could be smaller to differentiate from the other buttons. Language button looks too big save button color does not match and looks dull The color of the save button do not match the rest of the page. maybe can change the colour of the save button to green save button color needs to be changed Simple and clean layout, easy to understand what is going on. the language change very fast There is a lot of "spacing" between the text and the box, might need some adjustments. The black button needs some recoloring also as it doesn't match with the rest of the buttons. Clean and nice. buttons too big change the colour smaller buttons
--	--

Figure 19a

Figure 19b

The Language Settings page received many comments regarding the colour of our Save button as its colour didn't fit well with the rest of the page. Some also thought the buttons were too big.

<div style="border: 1px solid #ccc; padding: 10px;"> <h3 style="text-align: center;">Create A New Profile</h3> <p>First Name: <input type="text" value="Enter your first name"/></p> <p>Last Name: <input type="text" value="Enter your last name"/></p> <p>Gender: <input type="text" value="Male"/></p> <p>Date of Birth: <input type="text" value="dd/mm/yyyy"/> <input type="button" value=""/></p> <p style="text-align: center;">Save Cancel</p> </div>	<p>Create Profile Page</p> <p>Are there any improvements or feedback for the create profile page?</p> <hr/> <p>plain background</p> <hr/> <p>looks okay</p> <hr/> <p>form looks okay but could use a different color background</p> <hr/> <p>It is okay, quite simple and minimalistic.</p> <hr/> <p>no improvements</p> <hr/> <p>maybe can use different colour background</p> <hr/> <p>okay</p> <hr/> <p>Simple and clean layout, easy to navigate.</p> <hr/> <p>no improvements</p> <hr/> <p>the form looks neat and tidy with proper alignment</p> <hr/> <p>Simple and minimalistic layout, easy to understand and look at.</p> <hr/> <p>Clean and nice.</p> <hr/> <p>simple and just noce, i like it</p> <hr/> <p>good</p> <hr/> <p>good and direct, straight to the point</p>
---	--

Figure 20a

Figure 20b

This page received positive feedback for it being simple. However, some mentioned that the background colour could be different.

Welcome, Sham

Medication overview

Add new medication

Interferon beta-1b (Injectable)
Pain
Dosage: 3 Pill Everyday

KINERET (Injectable)
medciidn
Dosage: 2 Heh

Medications **Appointments** **Profile**

Figure 21a

Home Page

Are there any improvements or feedback for the home page?

16 responses

no improvements

button color can be changed.

colors dont match. the button colors can be changed to match the theme

Add new medication button color does not match the rest of the color theme

the orange for the medicine information does not match the bar below.

The add medication button color and the rest of the colors do not match. The delete button sizing is wrong also. But layout wise is ok.

looks nice, soft colours and are not harsh on the eyes

delete button color can be changed to red. add new medication button should also be another color than black

Simple and clean layout, easy to see what is going on, the buttons need some changes to coloring and sizing.

different colour perhaps

the alignment is great, short description

Delete button needs to be resized as the space on the left is bigger than the right. The black button doesn't fit well with the rest of the page. Other than that it is good.

Clean and nice, but the delete text is off-center.

maybe can change the colour scheme

maybe can make the cards smaller

Figure 21b

While receiving mostly positive feedback on the layout, a few users also mentioned we could change our page's colours.

Overall UI Feedback

Anything else that can be improved on?

16 responses



-

Overall the whole application can use more colors

Application could use a background theme color instead of leaving it white. Button colors look very dull

better color theme. current one looks too plain

needs background color or a main color theme

It is generally okay in terms of the layout, however there are certain sizing issues and color issue that can be changed to give the app a better look.

NA

more colors are needed and the main buttons need to be of different color

Certain button sizing and coloring needs to be changed, otherwise it is good.

overall can maybe use some different colours

overall pretty good

Overall it is okay.

i like it, its simple but maybe can change some colours

better colours required coz the current one looks too plain

overall i think its quite nice, but some minor changes can be made

Figure 22 - Additional comments from user survey

The overall feedback for our application's UI was positive regarding the layout. However, most of our users prefer a more vibrant colour scheme with background colours to enhance the visual appeal.

The second round of user testing has given us a general idea on how to proceed with the next iteration of our application. We will work towards fixing the issues that our users mentioned for our next iteration. There was also positive feedback to our application, such as our layout being good and clean, and the certain colours we used, which were good. We will aim to maintain these positive responses given in our next iteration.

Backend (User Experience)

We have also surveyed our participants on our application's functionality by asking them to rate on a scale of 1-5, with one being Strongly Disagree and five being Strongly Agree, the different aspects of our application

I found the application easy to use.

16 responses

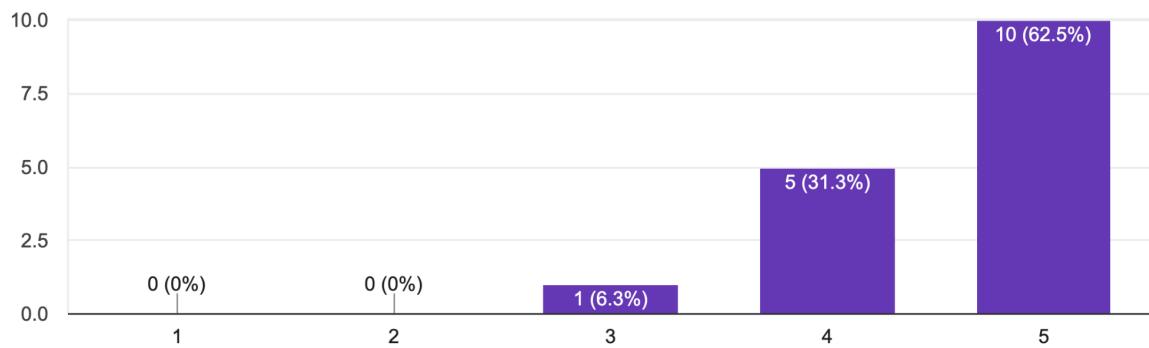


Figure 23 - Responses on ease of application

Most participants felt that *DoseManager* was generally easy and simple to use and had no issues using and running it from the APK file given to them to download.

I would need the help/support of a technical person to be able to use this application

16 responses

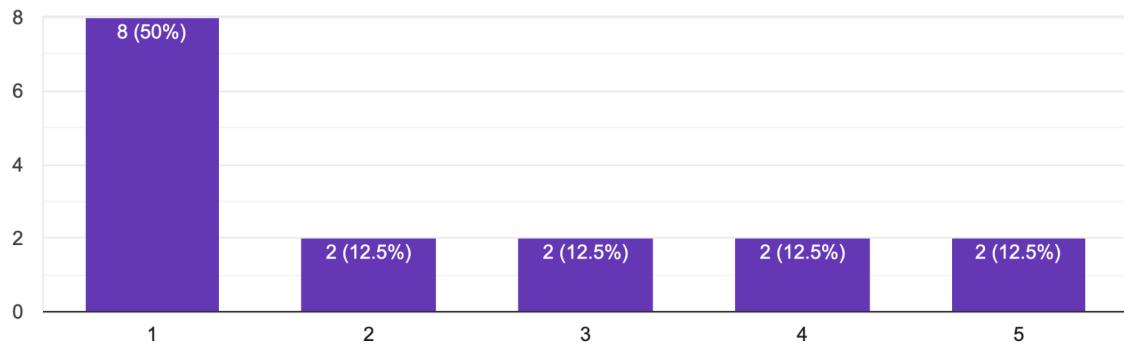


Figure 24 - Responses on requirement of assistance

Our application was simple enough to use such that most of our users did not require any technical knowledge. They could run our application and use it straight away without any instructions or help. This was a positive sign for our development as we wanted to cater to the elderly.

I thought there was too much inconsistency in this system

16 responses

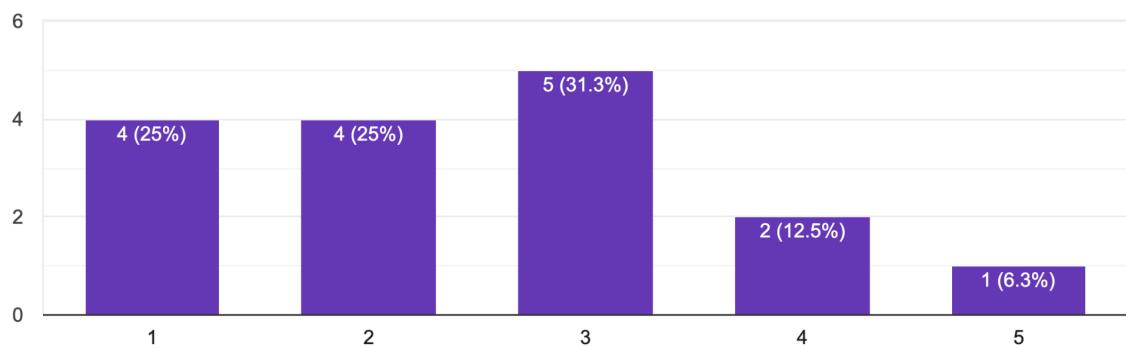


Figure 25 - Responses on inconsistency in the system

Most felt that *DoseManager* was consistent across pages.

I think the various functions in the system were well integrated.

16 responses

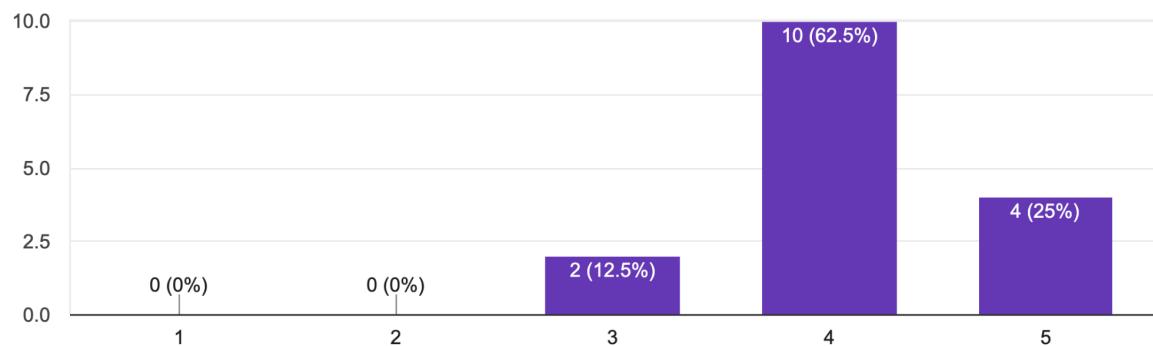


Figure 26 - Responses on integration of functions

Most of our users felt that the different functions in our application were well integrated. We wanted our users to use our application without any trouble navigating around the various buttons and functions.

I think most people would be able to learn how to use this system quickly.

16 responses

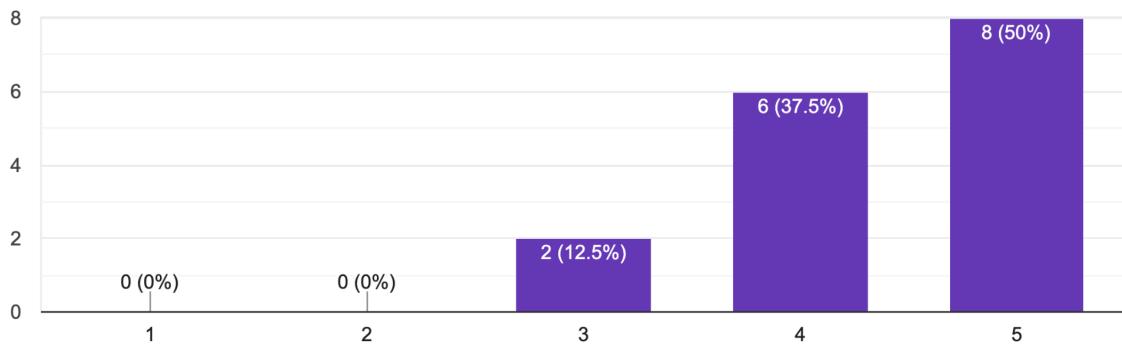


Figure 27 - Responses on whether system is easy to learn

I felt very confident using the system.

16 responses

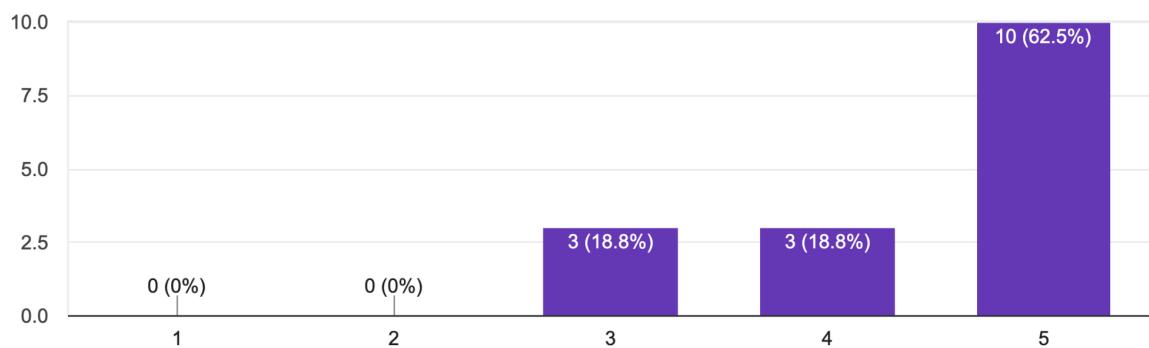


Figure 28 - Responses on confidence in using the system

I needed to learn about other things in order to use this application

[Copy](#)

16 responses

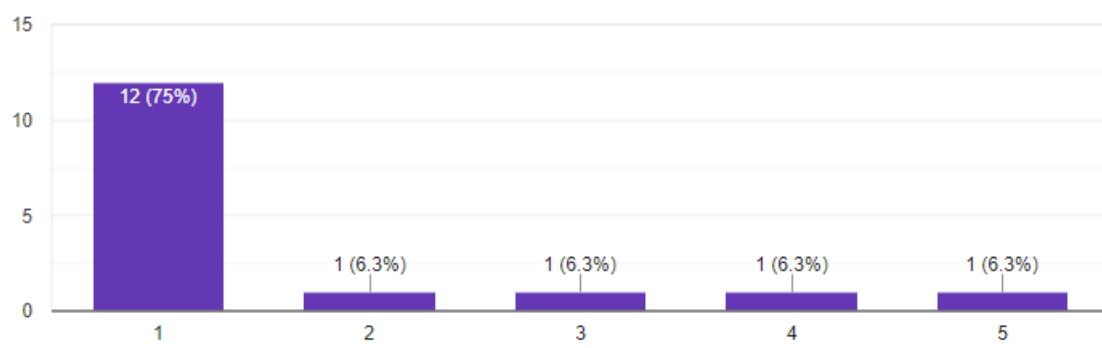


Figure 29 - Responses on the need to learn new skills

From figures 27-29, our users agreed that most people would be able to learn how to use our application quickly and that they felt confident using our application. They also disagree that they require specific knowledge or skills to use our application.

The buttons work as intended

16 responses

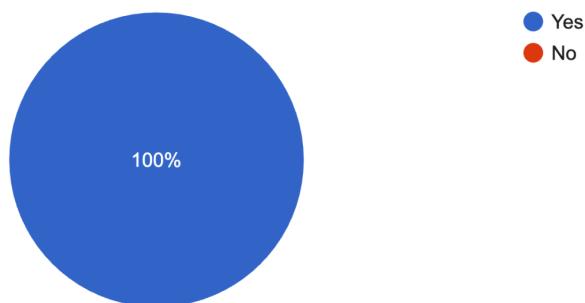


Figure 30 - Responses on whether the buttons work as intended

Figure 30 shows that our users had no issues with the buttons of our application.

I find the language change feature useful.

16 responses

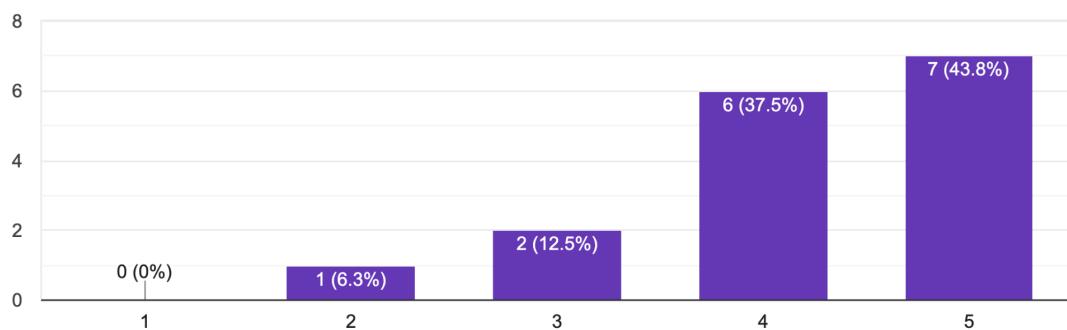


Figure 31 - Responses on whether the language change feature is useful

The majority of our surveyors agreed that the language change feature of our application was useful to them.

The whole navigation process is smooth

16 responses

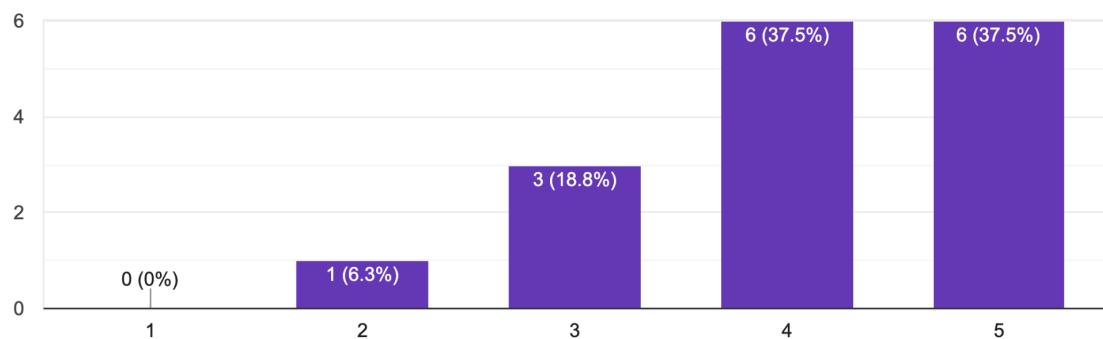


Figure 32 - Responses on the navigation process of the app

Most users also agreed that the navigation of our application was smooth.

Any other comments regarding the functional aspects (UX) of our application?

16 responses

i could submit empty inputs to the forms. maybe there should be a required valid input for them

overall functions is good and working well, i will use this reminder app to keep track of my medication in the future

could have background colour

Input validation could be worked on. i could submit an empty form

i think the app functions like it was intended to. i would use it on a daily basis

The app functions well overall. I would use the application if I require to take medications next time.

Keep up the good work Team 66!

for the most part the key requirements were working as intended

The whole application is smooth and easy to use

no comments

i could submit an empty appointment in the appointment page. maybe try to make it a requirement to have a valid input before submitting

I could leave certain input fields empty within the add medication page.

-

No comments.

I could submit an empty form.

The app works as intended.

Figure 33 - Additional comments regarding the UX of the app

The last question in our survey was to find out more from our users and gather their comments regarding feedback on our application's UX. Most users agree our application was good and functions well overall. They generally liked our application as it was simple and clean. These are the positives that we will keep and continue in our next iteration. However, some mentioned they were able to submit empty inputs into our different forms while others had issues with our navigation, mentioning that it needed to be smoother. Some users felt they needed help from tech-savvy personnel to use the application, while some felt that the application needed to be more consistent. We considered this feedback while working on our next iteration.

Conclusion

In conclusion, the feedback we've gotten on our app's UI and UX has given us some beneficial information for our next iteration. Users requested changes to button sizes and colour schemes, as well as the usage of vibrant colours and better contrast in the UI. Positive UX reviews emphasised the application's simplicity, usability, and seamless feature integration. The positive feedback from our users who felt confident using our application gave us the confidence that our application was simple enough to be used. However, some users voiced concerns about the application's consistency, problems with the navigation, and the need for assistance with certain tasks. Our upcoming development cycle was focused on addressing these issues while maintaining highly lauded features like the application's overall functionality, language change feature, and user-friendly design. This feedback will be instrumental in refining and enhancing the application for a more intuitive and satisfying user experience.

Final iteration (Final working prototype)

Our team created another survey to gather user feedback with regard to our final working product. The results of the survey are shown below.

Frontend (User Interface)

The overall look and feel of the app's interface are visually appealing.

 Copy

16 responses

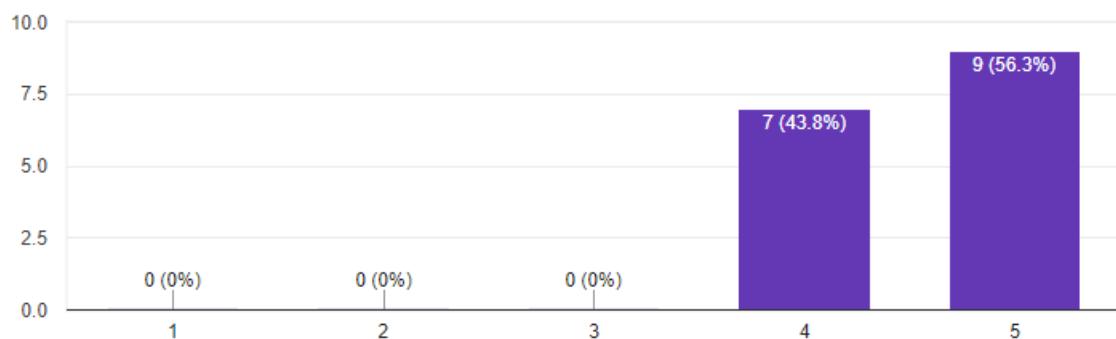


Figure 34 - Responses on the app's interface

Our users agreed that *DoseManager* was visually appealing.

The navigation within the app is intuitive and easy to understand.

 Copy

16 responses

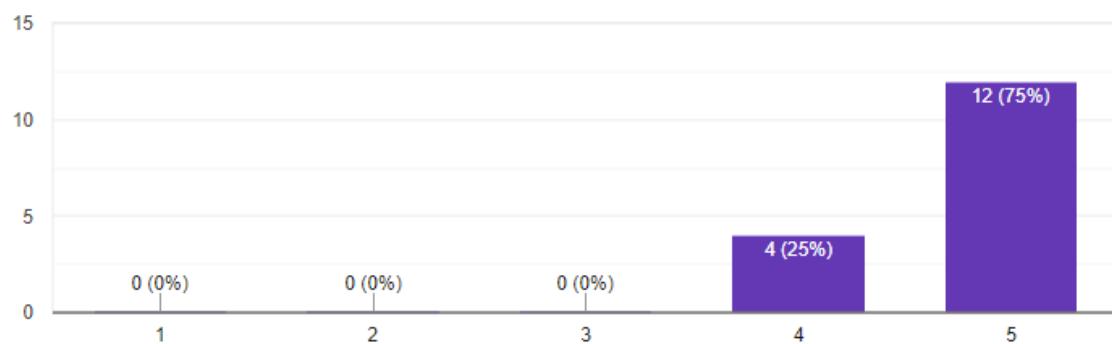


Figure 35 - Responses on the intuitiveness of the app

Our users gave positive feedback with regards to the application's navigation.

The app's layout and organization make it easy to find the features or information I need.

 Copy

16 responses

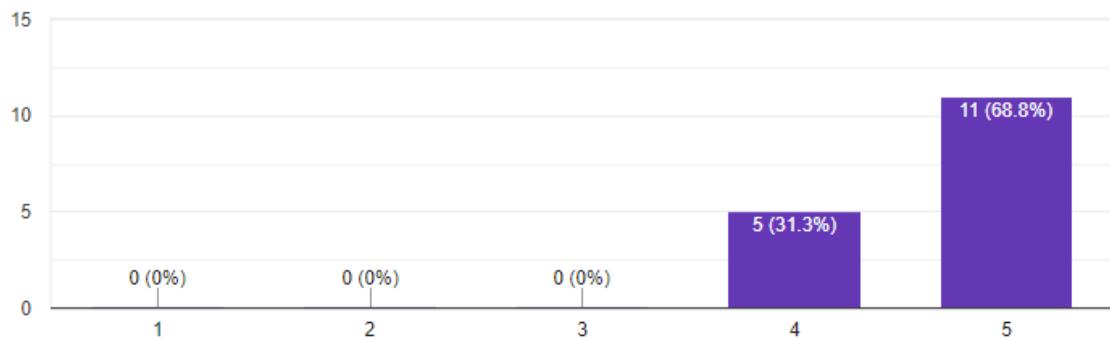


Figure 36 - Responses on the layout and organisation of the app

There was a positive consensus regarding our app's layout and organisation, making it easier for our users to find features or information they require.

The font size and readability of text in the app are comfortable for me.

 Copy

16 responses

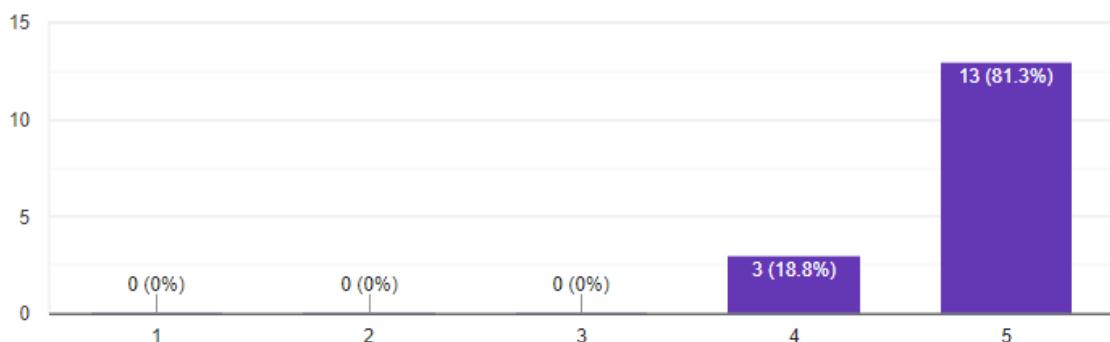


Figure 37 - Responses on the text of the app

The users agreed that the readability of the text in our app was comfortable for them.

The use of colors in the app is pleasing and enhances the user experience.

 Copy

16 responses

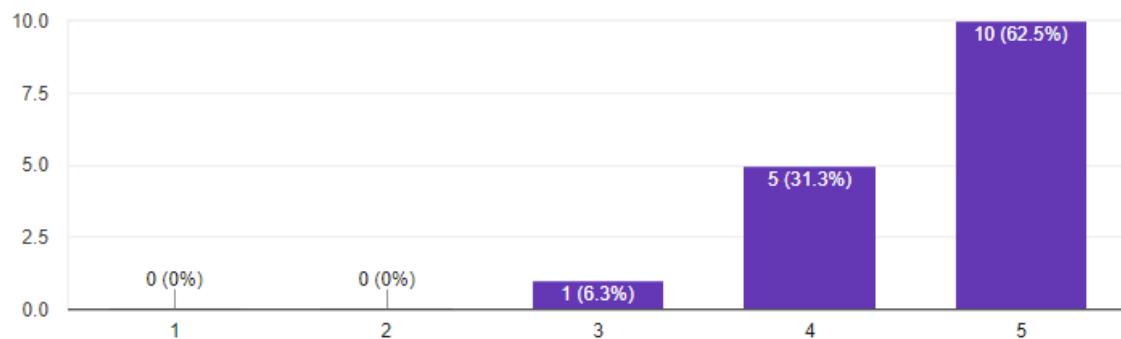


Figure 38 - Responses on the colours of the app

Most of the feedback we got agreed that the blend of colours for the application was similar to each other, and it was a pleasant experience using the application.

Backend (User Experience)

I found the application easy to use.

 Copy

16 responses

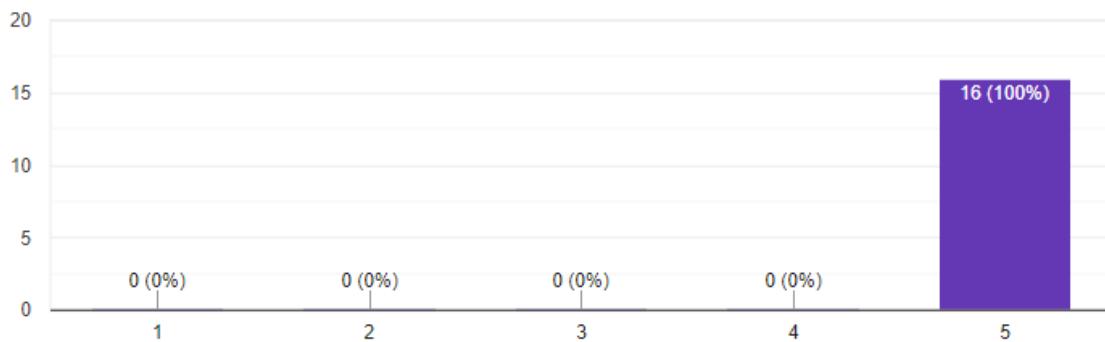


Figure 39 - Responses on whether the app was easy to use

All of our users agreed that our application was easy to use.

The app responds quickly to my actions, and there is minimal lag or delay.

 Copy

16 responses

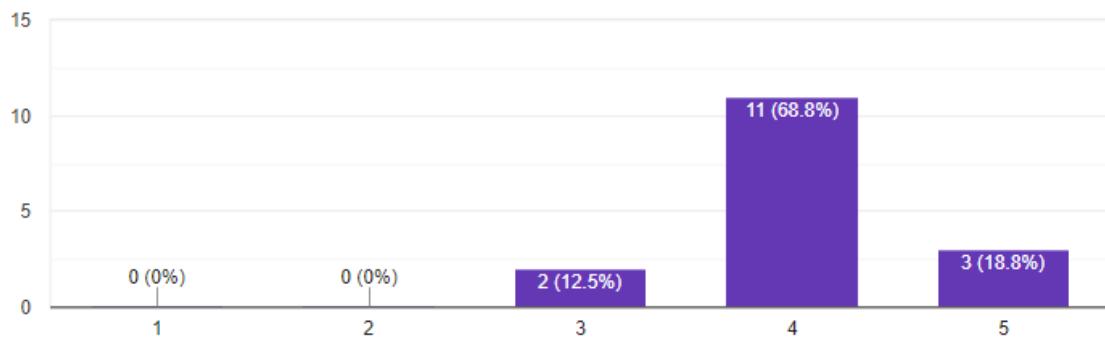


Figure 40 - Responses on the responsiveness of the app

Most of the users felt that the application's performance speed was reasonable, with a small minority saying it could be slightly better.

I think the various functions in the system were well integrated.

 Copy

16 responses

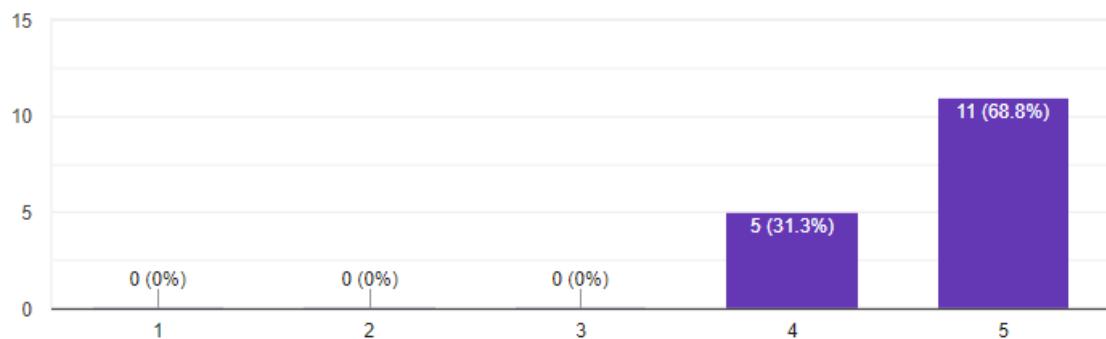


Figure 41 - Responses on the integration of functions in the app

Our users agree that our application's functions were well integrated.

The buttons work as intended.

 Copy

16 responses

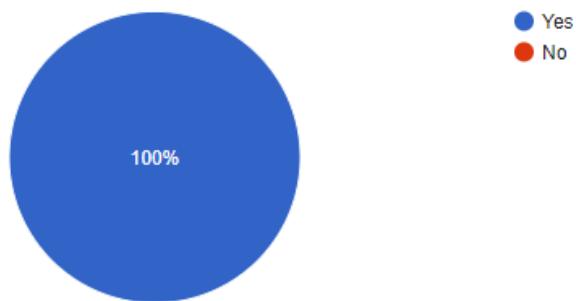


Figure 42 - Responses on whether the buttons work as intended

Users agree that all of our buttons work as intended.

The notifications work as intended.

 Copy

16 responses

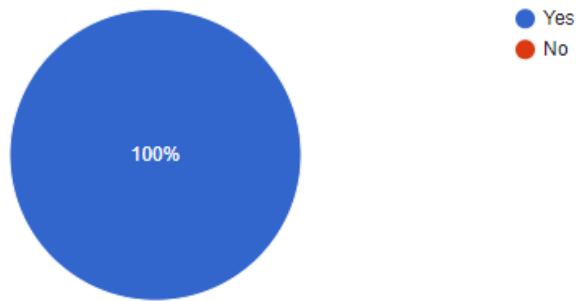


Figure 43 - Responses on whether the notifications work as intended

Users agree that the notifications work as intended.

Overall, I am satisfied with the user experience provided by this app.

 Copy

16 responses

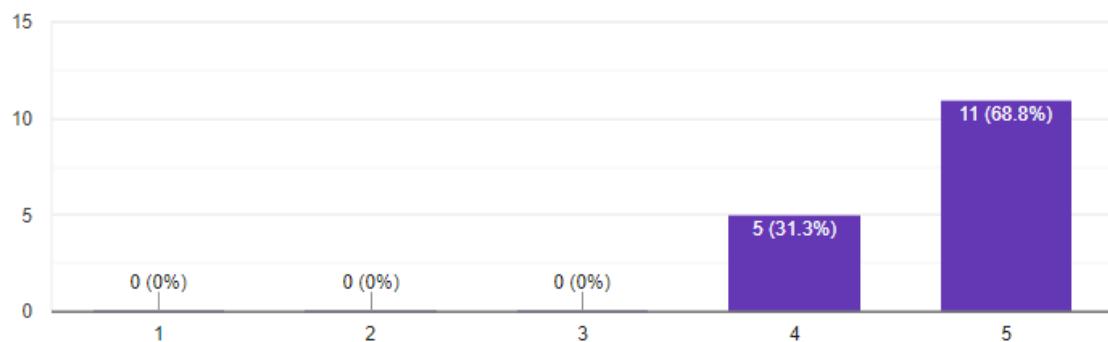


Figure 44 - Responses on the overall UX of the app

Users of our application generally felt their experience was positive.

Any other feedbacks or comments?

16 responses

when changing language there is a noticeable lag for the text translation between pages.

the colors used for the application were very neutral and calming

the aesthetic of the app was really minimalistic and the required information was displayed neatly

When the language changed the other pages took awhile to be translated, otherwise the app is good.

The lag when I am loading the other pages when the language is changed is bad, however I do not use the other language therefore it does not affect me. Great app!

the color choice was good

color theme was good. loved the minimalistic look to it

Good app. I liked how the intuitive it feels to use the app, and everything functions as intended. I would use this app if I need to in the future.

the flow and navigation of the app was good

No comments

I am able to use the app with ease.

NIL

Great app for the patients :)

Great app. Would use it in the future if I need something like this.

Figure 45 - Additional comments regarding the UX of the app

We had multiple positive comments from our users. However, a handful also mentioned that when they swapped languages, our pages took a while to be translated.

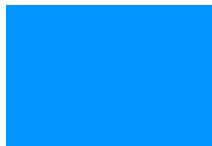
Conclusion

From our survey results, our users largely agreed that the UI of our app was good. This meant that our users widely accepted our app, and thus, we would be able to push this out for users to use. Although there were a few people who did not like the UI of our app, due to time constraints we were unable to correct them.

On the backend side, the feedback was mostly positive, with slight improvements on optimising the application to load faster, especially the language translation. We hope to be able to continue working on this project past this course and improve it to suit more of our users.

Design choices

Colour Scheme



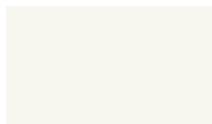
RGB Code: RGB (0, 150, 255)

We have decided to use this shade of blue for *DoseManager* as it tends to be associated with calmness. It can also create a soothing and peaceful environment, which can be beneficial when using *DoseManager*. The colour is not too intruding and has a gentle vibe associated with it. Additionally, as most of the application has a white background to it, blue compliments well with the background.



RGB Code: RGB (234,234,219)

We chose a background colour for the application's pages that closely matches our logo. This decision was made to create a seamless experience for our users. As they navigate through the application, we aim to reinforce our brand by providing a consistent visual reminder of our logo. Additionally, it complements the shade of blue used. This cohesive design approach aims to enhance user familiarity and brand recognition.



RGB Code: RGB (247,247,237)

Finally, we opted for a lighter shade within our background to provide a subtle contrast to our medicine and appointment cards. Our intention was to create a noticeable yet subtle difference for our information cards while maintaining a connection to the overall theme. We used the same colour for forms within the pages, contributing to a sense of calmness and familiarity throughout our application.

Logo



Figure 46 - DoseManager Logo

We decided to create a logo and catchphrase that is eye-catching and stands out to the user so they can have a general understanding of what our application is about just by looking at it. The red cross in the logo symbolises healthcare, which is related to medication.

We decided to name our application *DoseManager* because that is what we aim to do - manage the doses of our users. The name directly communicates the main function of the app and is straightforward, allowing our users to easily remember it. It also portrays professionalism to users as it is accompanied by a well-designed UI and efficient backend development.

Components of the software

Presentation Layer

The presentation layer of *DoseManager* has been created using basic web development tools, like HTML and CSS to create the buttons, and input boxes, and displaying all the medicine information as well as the appointment details. Additionally, we utilised the Bootstrap framework to help us with the styling of our pages.

Application Layer

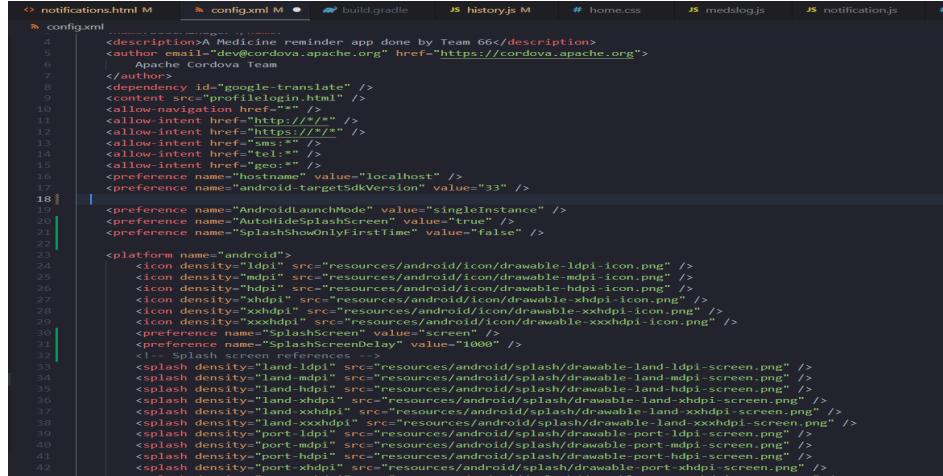
Our application contains HTML, CSS and JavaScript, connecting to certain external APIs such as the RxTerms from the National Library of Medicine, and the Google Translate API. This resulted in the application loading slowly at certain times, as we have to fetch data through the respective APIs to then perform certain functions to be displayed in our application.

Data Layer

Finally, the data layer within the application would consist of using Firebase's database service, Firestore, to store all the data collected within the application. There are a few pages within the application that contain a form to collect necessary details such as the medicine details, appointment details, and individual profile details. Data collected from these forms will then be used to carry out CRUD operations on our firestore database.

Structure and Implementation

The Cordova application has a config.xml file, as seen in **Figure 47**. It defines all the metadata, permissions and initial page bootup to display.



```
<description>A Medicine reminder app done by Team 66</description>
<author email="dev@cordova.apache.org" href="https://cordova.apache.org">
    Apache Cordova Team
</author>
<dependency id="google-translate" />
<content src="profilelogin.html" />
<allow-intent href="http:///*" />
<allow-intent href="https:///*" />
<allow-intent href="sms:*" />
<allow-intent href="tel:*" />
<allow-intent href="geo:*" />
<preference name="hostname" value="localhost" />
<preference name="android-targetSdkVersion" value="33" />
<preference name="AndroidLaunchMode" value="singleInstance" />
<preference name="AutoHideSplashScreen" value="true" />
<preference name="SplashShowOnlyFirstTime" value="false" />
<platform name="android">
    <icon density="ldpi" src="resources/android/icon/drawable-ldpi-icon.png" />
    <icon density="mdpi" src="resources/android/icon/drawable-mdpi-icon.png" />
    <icon density="hdpi" src="resources/android/icon/drawable-hdpi-icon.png" />
    <icon density="xhdpi" src="resources/android/icon/drawable-xhdpi-icon.png" />
    <icon density="xxhdpi" src="resources/android/icon/drawable-xxhdpi-icon.png" />
    <icon density="xxxhdpi" src="resources/android/icon/drawable-xxxhdpi-icon.png" />
    <preference name="SplashScreenDelay" value="screen" />
    <!-- Splash screen references
    <splash density="land-ldpi" src="resources/android/splash/drawable-land-ldpi-screen.png" />
    <splash density="land-mdpi" src="resources/android/splash/drawable-land-mdpi-screen.png" />
    <splash density="land-hdpi" src="resources/android/splash/drawable-land-hdpi-screen.png" />
    <splash density="land-xhdpi" src="resources/android/splash/drawable-land-xhdpi-screen.png" />
    <splash density="land-xxhdpi" src="resources/android/splash/drawable-land-xxhdpi-screen.png" />
    <splash density="port-ldpi" src="resources/android/splash/drawable-port-ldpi-screen.png" />
    <splash density="port-mdpi" src="resources/android/splash/drawable-port-mdpi-screen.png" />
    <splash density="port-hdpi" src="resources/android/splash/drawable-port-hdpi-screen.png" />
    <splash density="port-xhdpi" src="resources/android/splash/drawable-port-xhdpi-screen.png" />
    <!-- End of splash screen references -->
```

Figure 47 - Image of the config.xml file

We organised the different files by their file types and created the different folders to store them in. This keeps the files neat and when we reference the JavaScript files, they would all be contained in one place.

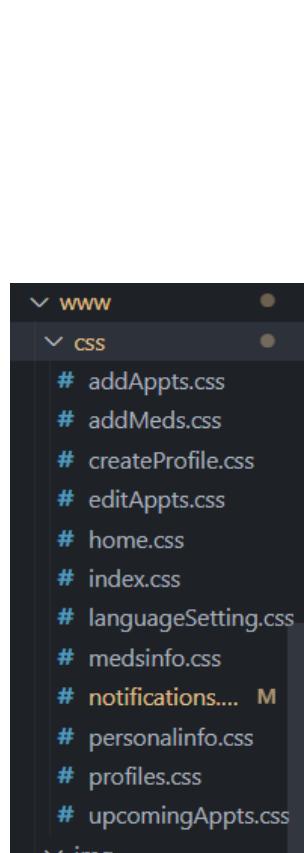


Figure 48a

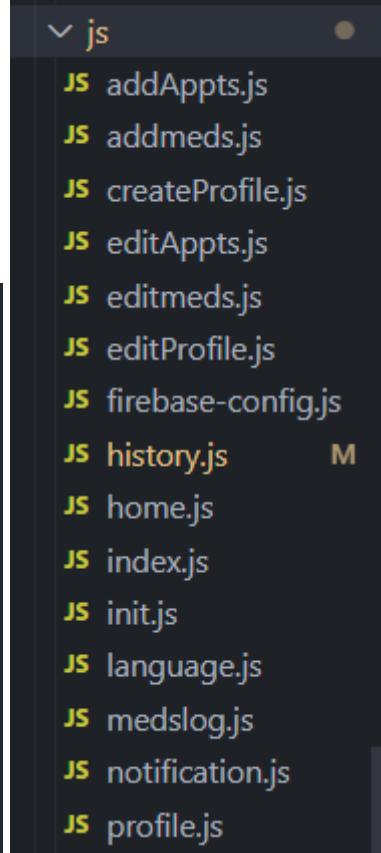


Figure 48b

Another important thing to note is since we are developing an Android application, when we run “cordova build android” to build our APK (Android Package Kit) file, Cordova looks into the “www” folder (figure 48a) for the web assets (HTML) and application logic (Javascript) to build and convert these assets and repackage it into an APK file for android use. So we needed to keep these files under the “www” folder to allow Cordova to build the APK file without any potential errors.

Setting up Cordova environment for development and testing

User variables for benja	
Variable	Value
ANDROID_HOME	C:\Users\benja\AppData\Local\Android\Sdk
ANDROID_SDK_ROOT	C:\Users\benja\AppData\Local\Android\Sdk
GameFirstUserPath	C:\ProgramData\ASUS\GameFirst
GRADLE_HOME	C:\Users\benja\Downloads\gradle-8.1.1-all\gradle-8.1.1
JAVA_HOME	C:\Program Files\Java\jdk-11

Figure 49 - Setting up Cordova Environment for *DoseManager* on Android.

Java Development Kit (JDK) Installation:

- Downloaded and installed Java Development Kit (JDK) version 11 from the official Oracle website.
- Set up the JAVA_HOME system environment variable to the path of installation

Gradle Installation:

- Downloaded Gradle 8.1.1 from the official Gradle website.
- Set up the GRADLE_HOME system environment variable to the path of installation
- Added the Gradle bin directory to the system PATH variable.

Android Studio and SDK Installation:

- Downloaded and installed Android Studio.
- During the installation, ensured that the Android SDK was installed and set its system environment variable to the path of installation
- Configured Android Studio to recognize the JDK and Gradle installations.

Cordova Installation:

- Installed Cordova using npm (Node Package Manager) with the command: **npm install -g cordova**

Setting up a New Cordova Project:

- Created a new Cordova project using the command: `cordova create DoseManager`
 - Navigated to the project directory: `cd DoseManager`

Adding the Browser and Android Platform:

- Added the Browser and Android platform to the Cordova project using the command:
cordova platform add android/browser

Google Services Integration:

- For Firebase integration and other Google services, downloaded the google-services.json file from the Firebase Console.
 - Placed the google-services.json file in the root directory of the Cordova project to ensure proper integration of Google Services.

Building and Running the App for testing and debugging:

- Used the command `cordova build android` to generate the Android APK.
 - For Android feature testing purposes, used the command `cordova run android` to deploy and run the app on a connected Android device or emulator

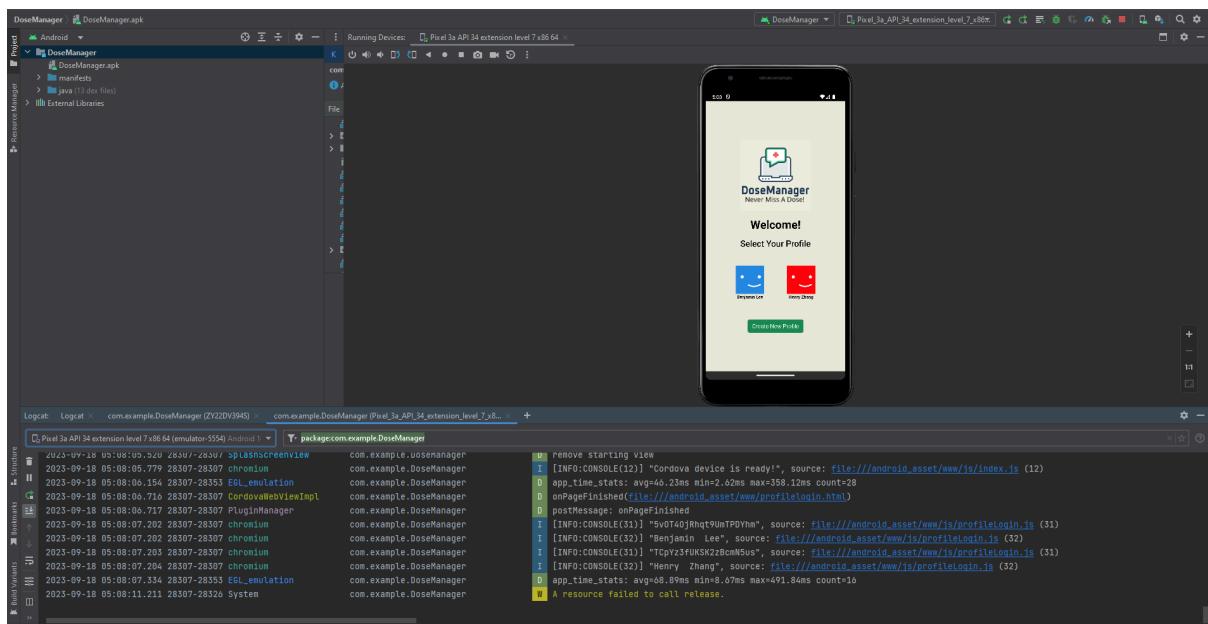


Figure 50 - Android studio's adb logcat to view Android system console logs

For Debugging on the browser, we used the command `cordova run browser` and pressed f12 to enter Chrome developer tools and simulated mobile behaviour under the device toolbar.

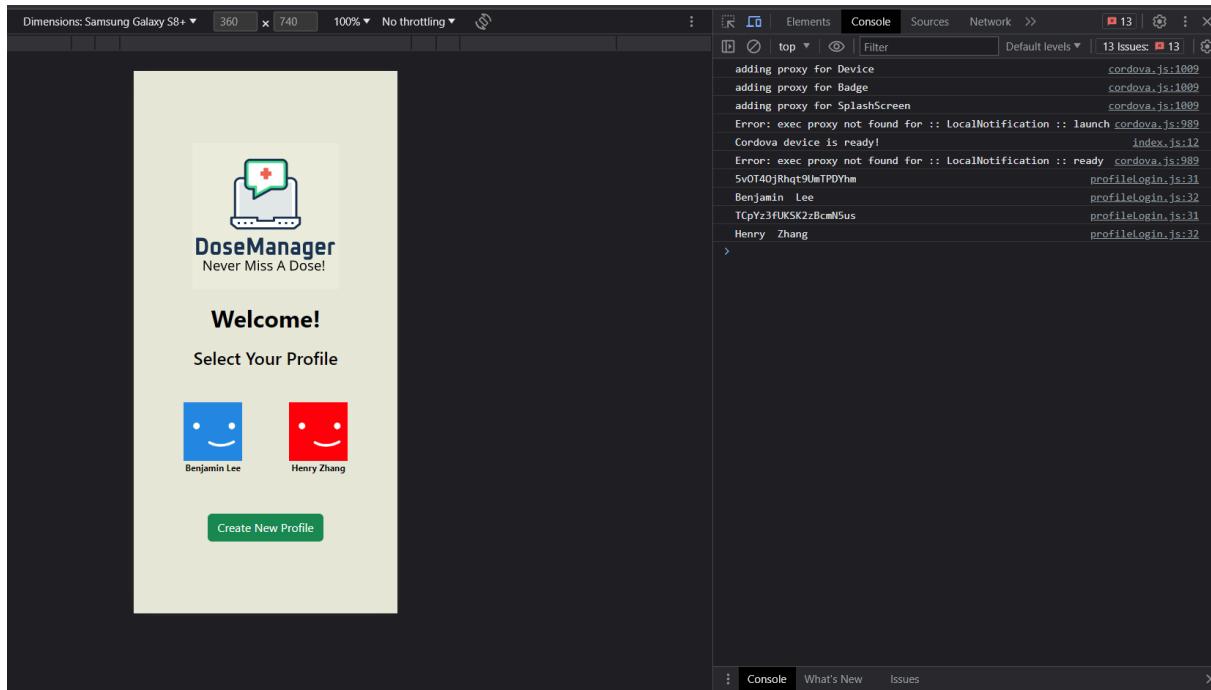


Figure 51 - Chrome's developer tools with device toolbar to simulate mobile behaviour

Firebase Cloud Firestore



Figure 52 - Cloud Firestore Logo

The backend team decided on using Firebase's Cloud Firestore due to its ease of setting up the connections and database for the application. Firestore was also chosen over other databases due to its cloud and real-time functionality across all devices, which was crucial for our application. After the connection with Firebase was established, we could interact with the database and carry out CRUD operations. (**Figure 53a & Figure 53b**)

```

www > js > JS init.js > ...
1  document.addEventListener("deviceready", function() {
2    const firebaseConfig = {
3      apiKey: "AIzaSyAt4SUUmSwvkHdas68AYQdj0e7fkfL547gQ",
4      authDomain: "dosemanager-d0236.firebaseio.com",
5      projectId: "dosemanager-d0236",
6      storageBucket: "dosemanager-d0236.appspot.com",
7      messagingSenderId: "373646054095",
8      appId: "1:373646054095:web:89660fa48e041a7d231dba",
9      measurementId: "G-XDL965JQ9H"
10     };
11
12     if (!firebase.apps.length) {
13       firebase.initializeApp(firebaseConfig);
14     }
15
16     var firestore = firebase.firestore();
17     window.firebaseio = firestore;
18   }, false);
19

```

Figure 53a – Global Firebase initialisation in init.js

```

notifications.html M config.xml M addmeds.html X build.gradle JS history.js M # home.css JS medslog.js J
www > addmeds.html > html > head > script
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Add medication</title>
8
9    <!-- Stylesheets -->
10   <link rel="stylesheet" href="css/addMeds.css">
11   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
12     integrity="sha384-rbsA2VBK0hggwzxH7pPCaAq046MgnOM80zw1Rwuh61DGLwZJEdK2Kadq2F9CU665" crossorigin="anonymous">
13
14  <!-- Firebase -->
15  <script src="https://www.gstatic.com/firebasejs/8.10.1.firebaseio.js"></script>
16  <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-firestore.js"></script>
17
18  <!-- Scripts -->
19  <script src="cordova.js"></script>
20  <script src="js/init.js"></script>
21  <script src="js/notification.js"></script>

```

Figure 53b- referencing our Firebase config throughout different parts of our app

```

notifications.html M medslog.html X JS history.js M # home.css JS medslog.js JS notification.js # notifications.css M JS upcc
www > medslog.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8" />
5    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6    <title>Medication Log Page</title>
7
8  <!--firebase-->
9  <script src="https://www.gstatic.com/firebasejs/8.10.1.firebaseio.js"></script>
10 <script src="https://www.gstatic.com/firebasejs/8.10.1/firebase-firestore.js"></script>
11

```

Figure 54 - Including Firebase components into our application

We used Firestore to store medication data in a cloud database for *DoseManager*. As Firestore is a form of noSQL database, it uses key-value pairing and instead of tables and rows, we have collections and documents, which suit our needs in managing complex medication-related information.

Firstly, we created a Firebase project, and within it, the Firestore database. A number of collections were created to store data such as medication information, appointment details and profile details.

Within each collection, documents were created for each medication, specifying the name, dose, frequency, etc. We were also able to use Firestore queries to retrieve medication data, such as all medications that need to be taken today. Likewise, appointment details were stored in a separate Firestore collection. This allowed us to provide our users with a sturdy and intuitive UI, making our app efficient.

Firebase also provides real-time updates, so our users can rest assured that their medication data is always up-to-date, as any changes or updates would be reflected immediately across all devices. This would instil trust and faith in our users during their time using *DoseManager*.

Thus, starting from the initial step in the development of our application, Firestore has allowed us to employ real-time updates and sophisticated querying capabilities, empowering our users to utilise *DoseManager* with utmost ease and confidence.

The screenshot shows the Cloud Firestore interface. On the left, the project name 'dosemanager-d0236' is visible. Under the 'Medicine' collection, there is a document named '6F0HuW1Fp3c7vrVH3lFs'. The document details are as follows:

- description:** "Ibuprofen is used to relieve pain from various conditions such as headache, dental pain, menstrual cramps, muscle aches, or arthritis. It is also used to reduce fever and to relieve minor aches and pain due to the common cold or flu."
- dosage:** "1 pill/day"
- name:** "Ibuprofen (Oral Pill)"
- timestamp:** August 30, 2023 at 10:30:26 PM UTC+8

Figure 55 - The Cloud Firestore collection layout

API Calls

National Library of Medicine API

We have decided to use an API to fetch medical data such as medicine names. Our API key will be taken from the [National Library of Medicine](#), which is a website with a collection of different API keys. This API will aid us in our “autocomplete” function, whereby when the user enters a part of the medication name, it will provide suggestions on the full medication name which will allow the user to click and fill in the name automatically as can be seen in Figure 56.

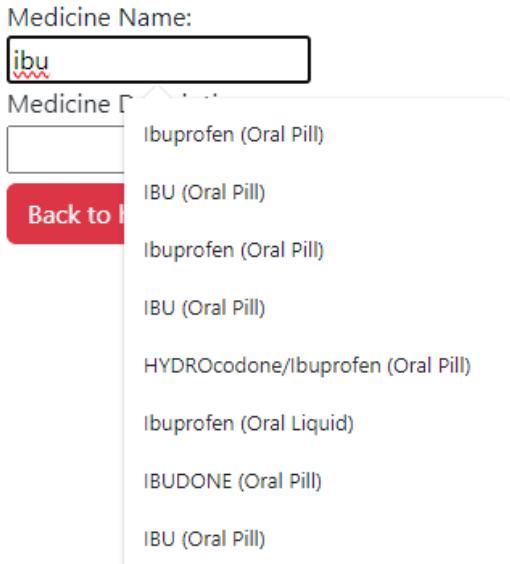


Figure 56 - Image of the “autocomplete” function working

We used the Node.js application to install our dependencies and packages that we would need. The list of the different packages that we require for this application can be seen in our package.json file in Figure 57.

```

    "dependencies": {
      "bootstrap": "^5.3.1",
      "firebase": "^8.10.1",
      "google-translate": "^3.0.0"
    }

  },
  "author": "Apache Cordova Team",
  "license": "Apache-2.0",
  "devDependencies": {
    "cordova-android": "^12.0.0",
    "cordova-browser": "^7.0.0",
    "cordova-ios": "^7.0.0",
    "cordova-plugin-firebase-analytics": "^7.0.5",
    "cordova-plugin-firebase-messaging": "^7.0.4",
    "cordova-support-android-plugin": "~2.0.4"
  },
  "cordova": {
    "plugins": {
      "cordova-plugin-firebase-messaging": {
        "ANDROID_FIREBASE_BOM_VERSION": "30.3.1",
        "IOS_FIREBASE_POD_VERSION": "9.3.0"
      },
      "cordova-plugin-firebase-analytics": {
        "ANALYTICS_COLLECTION_ENABLED": "true",
        "AUTOMATIC_SCREEN_REPORTING_ENABLED": "true",
        "ANDROID_FIREBASE_BOM_VERSION": "30.3.1",
        "IOS_FIREBASE_POD_VERSION": "9.3.0"
      }
    }
  }
}
  
```

Figure 57 - List of different packages used in our app

To interact with Firebase, we had to ensure our Firebase SDKs were all set up correctly, the configs that we declared had to match our project's API key within Firebase and to initialise the connection between our mobile application to Firebase.

Libraries and plugins used

```
PS C:\Users\benja\Desktop\SIM UOL\Year 2 Sem 2\Agile software projects\DoseManager-TO-THE-MOON\platforms\android> npm list
com.example.dosemanager@1.0.0 C:\Users\benja\Desktop\SIM UOL\Year 2 Sem 2\Agile software projects\DoseManager-TO-THE-MOON
+-- bootstrap@5.3.1
+-- cordova-android@12.0.1
+-- cordova-browser@7.0.0
+-- cordova-ios@7.0.0
+-- cordova-plugin-badge-fix@0.8.10
+-- cordova-plugin-browsersync@1.1.0
+-- cordova-plugin-device@2.1.0
+-- cordova-plugin-firebase-analytics@7.0.5
+-- cordova-plugin-local-notification-12@0.1.4
+-- cordova-plugin-splashscreen@6.0.1
+-- cordova-support-android-plugin@2.0.4
+-- cordova.plugins.diagnostic@7.1.2
+-- firebase@8.10.1
+-- google-translate@3.0.0

PS C:\Users\benja\Desktop\SIM UOL\Year 2 Sem 2\Agile software projects\DoseManager-TO-THE-MOON> cordova plugin list
cordova-plugin-badge-fix 0.8.10 "Badge"
cordova-plugin-device 2.1.0 "Device"
cordova-plugin-firebase-analytics 7.0.5 "FirebaseAnalyticsPlugin"
cordova-plugin-local-notification-12 0.1.4 "LocalNotification"
cordova-plugin-splashscreen 6.0.1 "Splashscreen"
cordova-support-android-plugin 2.0.4 "cordova-support-android-plugin"
cordova.plugins.diagnostic 7.1.2 "Diagnostic"
```

Figure 58 - list of libraries and commands

The team used a variety of libraries for our application. Namely,

- Cordova-plugin-badge-fix 0.8.10 "Badge"
- Cordova-plugin-device 2.1.0 "Device"
- Cordova-plugin-firebase-analytics 7.0.5 "FirebaseAnalyticsPlugin"
- Cordova-plugin-local-notification-12 0.1.4 "LocalNotification"
- Cordova-support-android-plugin 2.0.4 "Cordova-support-android-plugin"
- cordova.plugins.diagnostic 7.1.2 "Diagnostic"

Cordova -plugin badge

This was used to manage app notification badge numbers on android devices as a preview for users for unread notifications

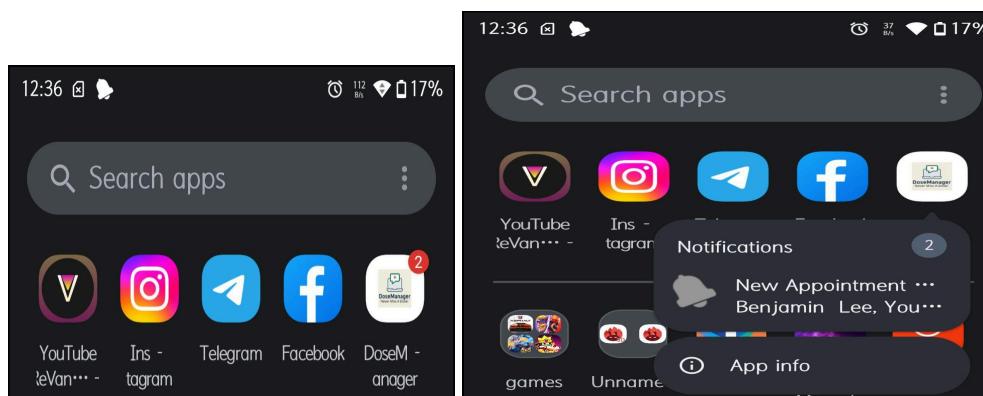


Figure 59a

Figure 59b

Cordova -plugin local-notification

This plugin enabled our app to show local notifications on the device which are triggered by our application itself. Users can then trigger specific actions based on the notification which affects the status of their reminder.

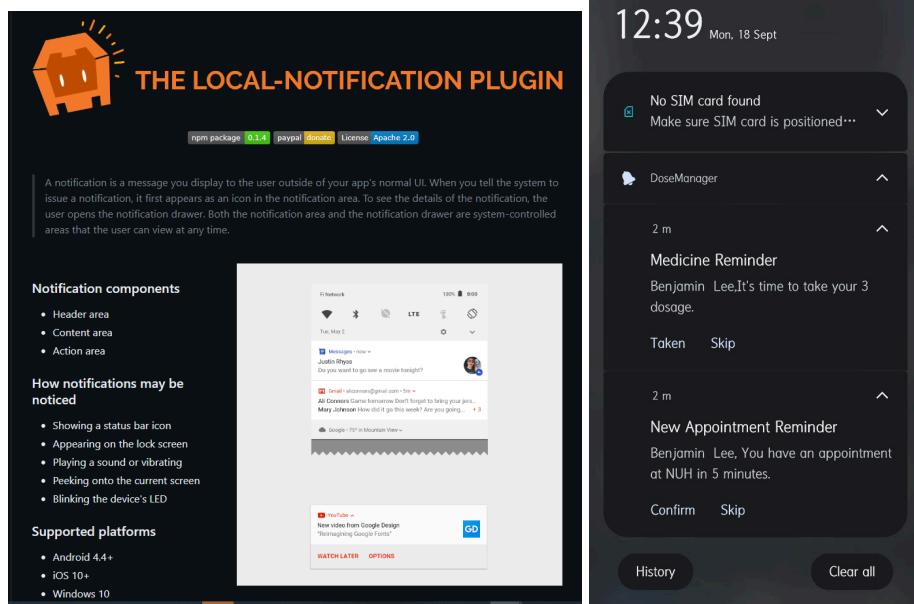


Figure 60 - Cordova Local Notifications 12 implementation

Cordova -plugin-firebase-analytics

The Firebase Analytics Dashboard Monitoring (Figure 61) gave our team an overview of the users using our application. From the images shown below, we are able to see statistics such as average engagement time of our application and the total number of users using our application. We were even able to view our users by their device model and their country of origin. This allows us to modify our application to fit our user's needs to keep them more engaged and to aid them in using *DoseManager*.

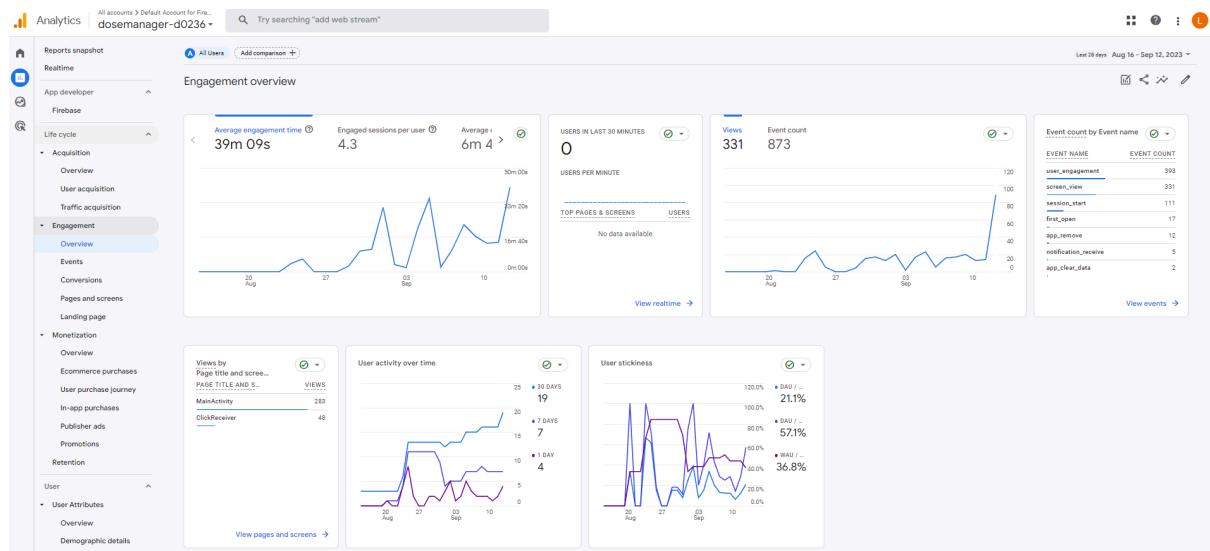


Figure 61 - Firebase Analytics Dashboard Monitoring DoseManager

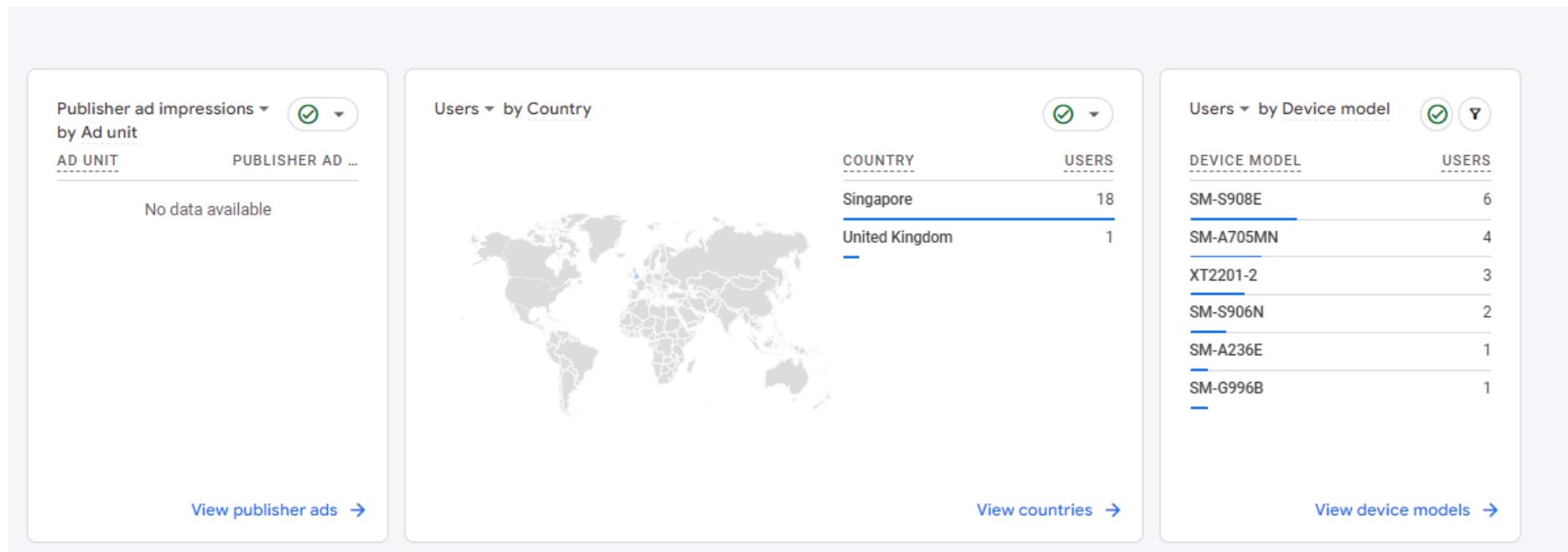


Figure 62 - DoseManager stats based on country

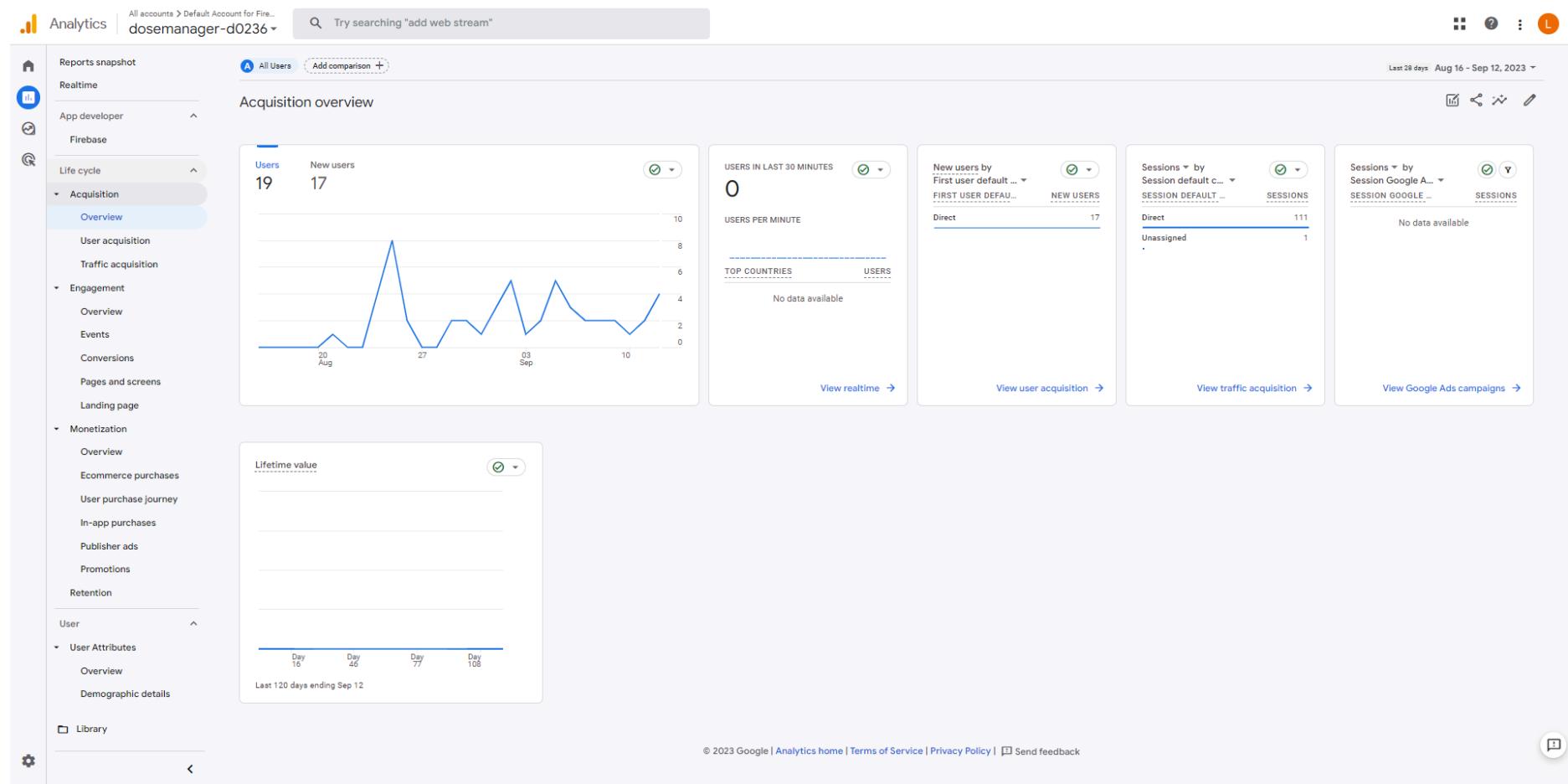


Figure 63 - Stats Overview

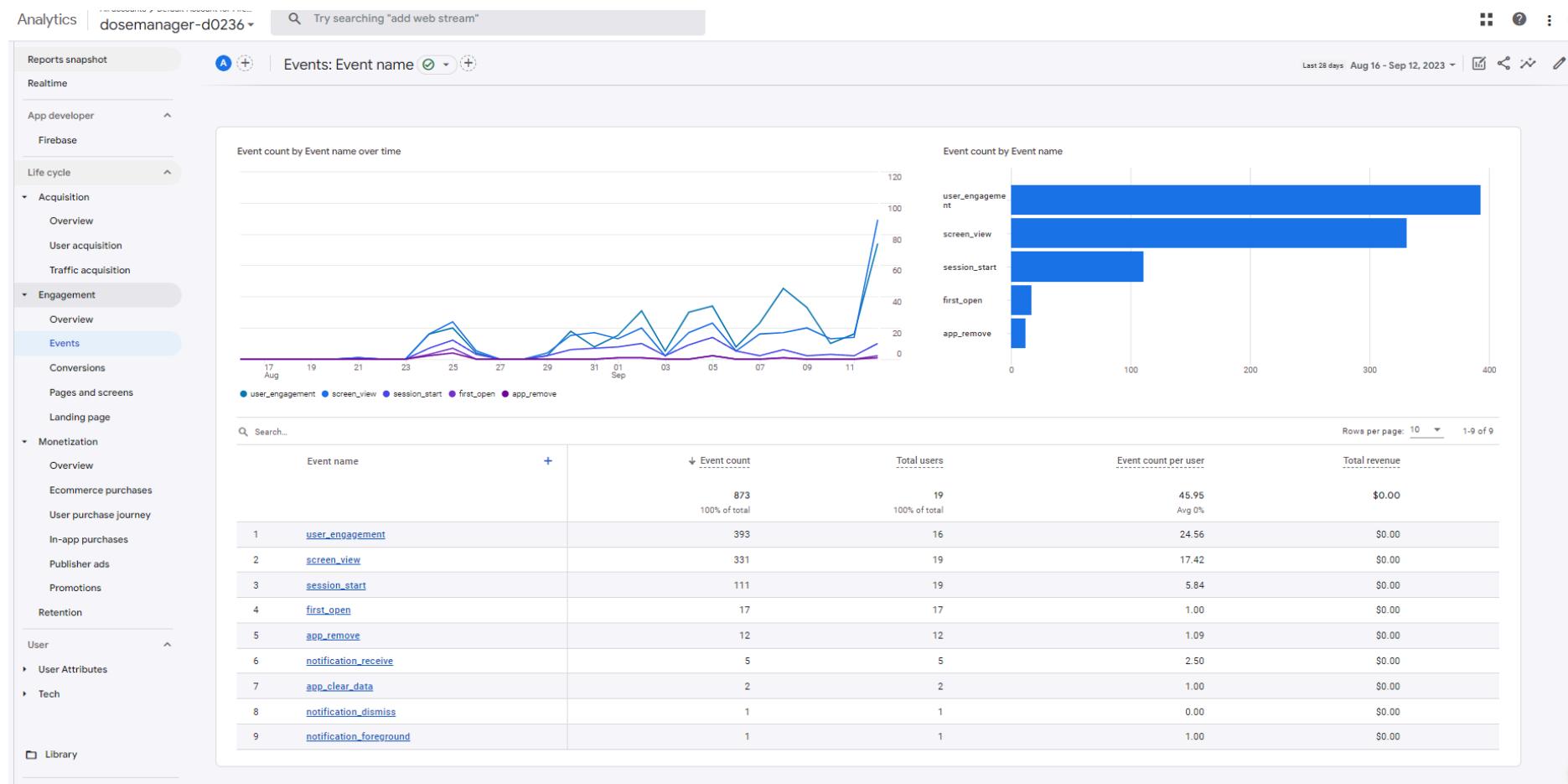


Figure 64 - Events stats

Cordova-support-android plugin

This plugin helped support the various Android functionalities within Cordova and bridges any potential compatibility issues between Cordova and Android devices.

Cordova-plugin-diagnostic

This plugin allowed the backend team to access and manage various device diagnostics. In this use case of *DoseManager*, it was used to request permission for notifications and redirect to the notification permission settings once confirmed.

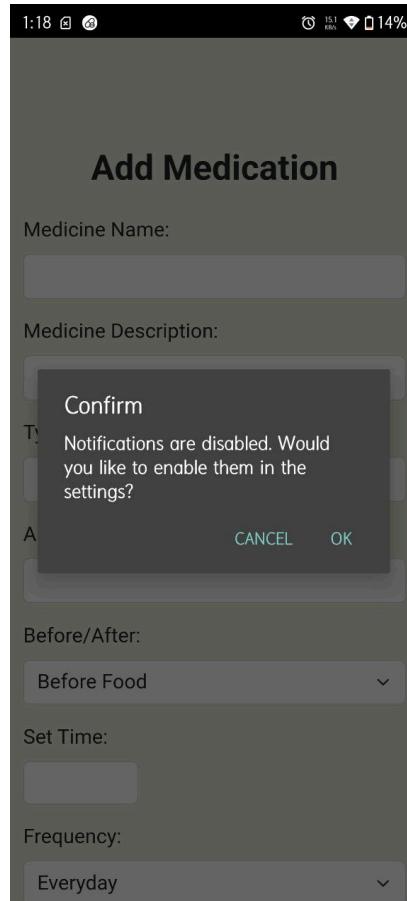


Figure 65a

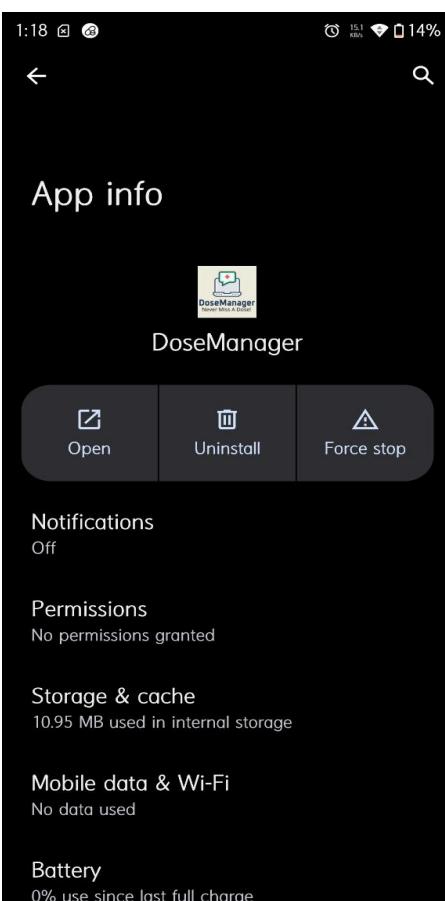


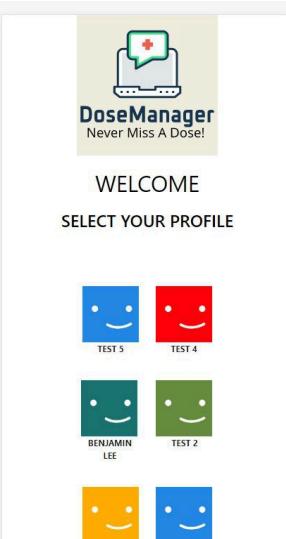
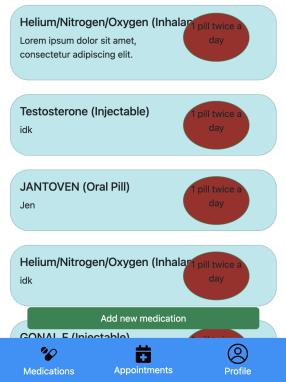
Figure 65b

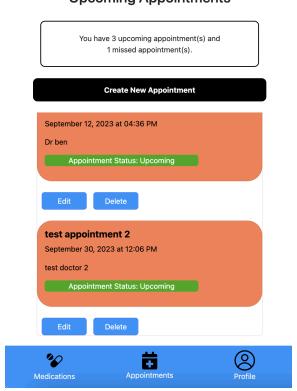
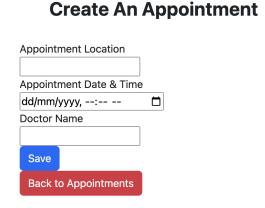
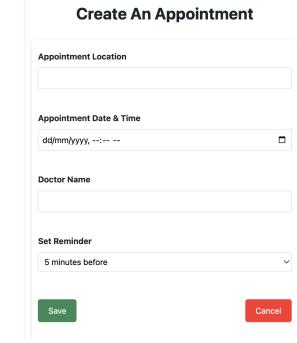
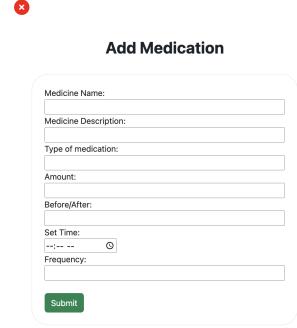
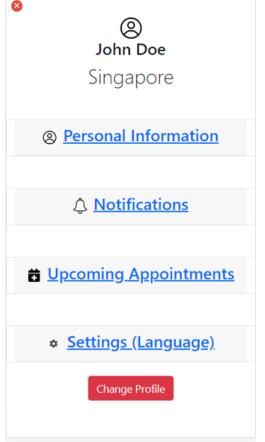
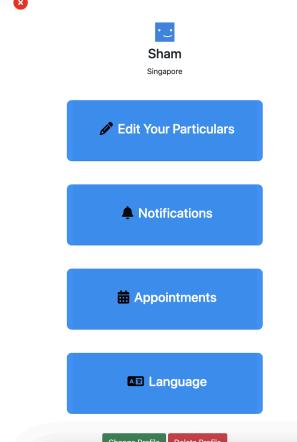
Evaluation

Frontend

There were certain design issues the frontend team faced while working on DOseManager. The table below summarises the most important issues we faces, and how we addressed them.

Design Issues

Issue Faced	Before	After	Solution and Explanation						
When multiple profiles were added to the application, the way it was presented on the profile login page looked messy.	 <p>WELCOME SELECT YOUR PROFILE</p> <ul style="list-style-type: none"> JUN RONG BENJAMIN BENJAMIN LEE 	 <p>WELCOME SELECT YOUR PROFILE</p> <table border="1"> <tbody> <tr> <td>TEST 5</td> <td>TEST 4</td> </tr> <tr> <td>BENJAMIN LEE</td> <td>TEST 2</td> </tr> <tr> <td>TEST 6</td> <td>TEST 3</td> </tr> </tbody> </table>	TEST 5	TEST 4	BENJAMIN LEE	TEST 2	TEST 6	TEST 3	<p>We used the <code>display: grid</code> property to let the list of profiles be shown as a grid container, letting us create a grid layout.</p> <p><code>grid-template-columns: repeat(2, 1fr)</code> specifies that there are going to be 2 columns of equal sizes in the grid while <code>gap:10px</code> specifies that there should be a gap of 10px between the grids</p>
TEST 5	TEST 4								
BENJAMIN LEE	TEST 2								
TEST 6	TEST 3								
When there was a big number of medications added to the list, they overflowed behind the “Add New Medication” button, hiding from sight. (A similar problem was faced with the upcoming appointments page as well)	 <p>Welcome, Sham Medication overview</p> <ul style="list-style-type: none"> Helium/Nitrogen/Oxygen (Inhalant) Lorem ipsum dolor sit amet, consectetur adipiscing elit. Testosterone (Injectable) idk JANTOVEN (Oral Pill) Jen Helium/Nitrogen/Oxygen (Inhalant) idk <p>Add new medication</p> <p>Medications Appointments Profile</p>	 <p>Welcome, Sham Medication overview</p> <p>Add new medication</p> <p>idk</p> <p>Dosage: 3 Pill Everyday</p> <p>Edit Delete</p> <p>Helium/Nitrogen/Oxygen (Inhalant) Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p> <p>Dosage: 3 Pill Every Week</p> <p>Edit Delete</p> <p>Medications Appointments Profile</p>	<p>The team added in a window which allows the user to scroll through the list of medications and moved the “Add New Medication” button to the top of the page. This ensures that the list of medications does not hide behind the button. The same solution was applied to the upcoming appointments page as well.</p>						

<p>There were many colours being used in the upcoming appointments page, causing it to look distracting and messy.</p>			<p>We simplified the layout and limited the number of colours used on this page to 2 main colours - a shade of orange and blue.</p>
<p>The page where users could create new appointments looked very simple and out of touch, and did not work with the CSS of the rest of the pages of our application.</p>			<p>We aligned the items on the page along the same margin, and used colour-coded buttons - green for 'Save' and red for 'Cancel'.</p>
<p>The add new medication page also had a very simple layout that looked out of place when compared to the rest of the pages.</p>			<p>We aligned the items on the page along the same margin and used a colour-coded button - green for 'Save' - and a button with a red cross for cancel.</p>
<p>The initial CSS for the profile page was very simple and did not look nice. It also did not follow the same stylings as the rest of the pages, and it lacked colours.</p>			<p>We aligned the items on the page along the same margin and coloured the buttons with the same shade of blue we used for the footer.</p>

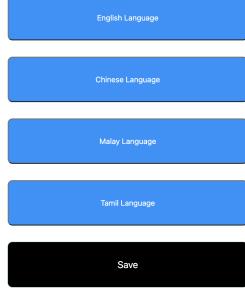
<p>The layout of the language settings page had small buttons, used different colours, and looked out of place when compared to the rest of the pages.</p>	 <p>Language settings</p> <ul style="list-style-type: none"> English language Chinese language Malay language Tamil language <p>Back to Home</p>	 <p>Language Settings</p> <ul style="list-style-type: none"> English Language Chinese Language Malay Language Tamil Language <p>Save</p>	<p>We aligned the items on the page along the same margin and coloured the buttons with the same shade of blue we used for the footer. We used the colour black for the save button to differentiate it from the languages button.</p>
--	---	---	--

Table 4 - Design Issues

Backend

Technical issues

Issues Faced:

There were many technical issues faced by the backend team.

Firstly, our initial plan with reference to the mid-term report was to use the Cordova Sqlite native plugin.

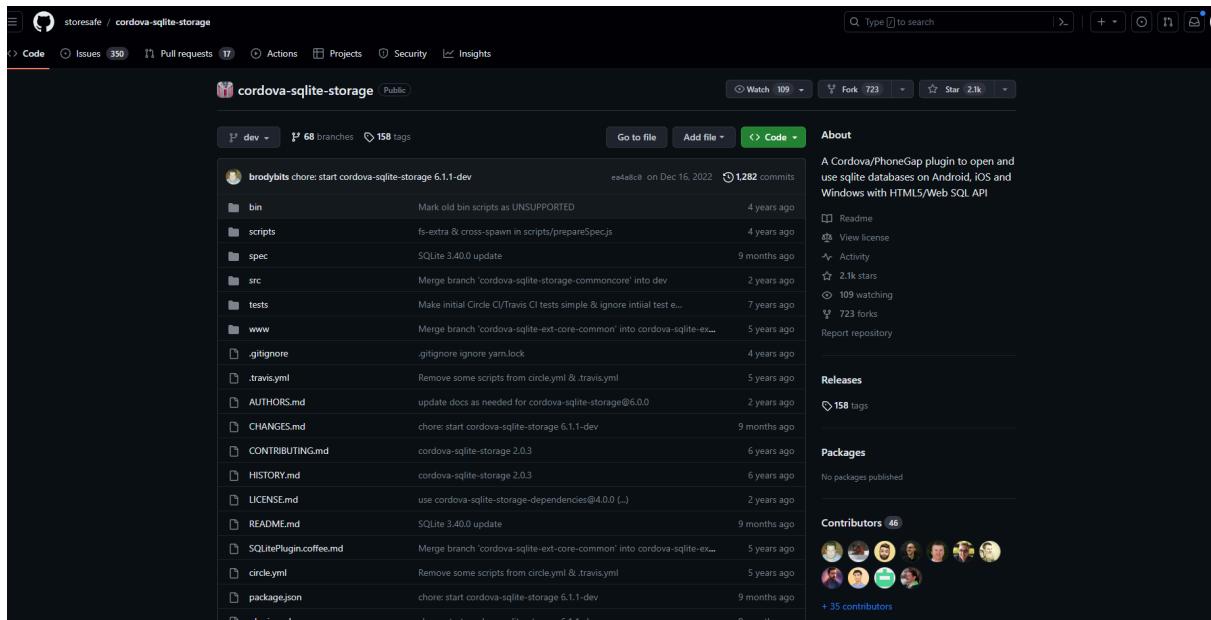


Figure 66 - cordova-sqlite-storage plugin on github

Cordova SQLite storage plugin our team planned to use. Our initial method involved installing the plugin via “cordova plugin add cordova-sqlite-storage” and initialising the database locally on the device’s storage when the app has been downloaded. We then used the ‘transaction’ method to execute SQL queries to interact with the database via SQL queries. Finally, we manipulated the DOM elements to take or retrieve data from the database.

However, with this method, we faced the issue where the data in the database was not stored throughout the session, and each time we closed the app or refreshed it, the data was gone. We could validate if the data was in the database each time it had been inserted by executing a select query in the console under the developer tools.

```
successfully
▶ DevTools failed to load source map:
Could not load content for chrome-extension://gkdkmjjodidppndkbkhknakbefl
s.map: System error:
net::ERR_BLOCKED_BY_CLIENT
> db.transaction(function(tx) {
  tx.executeSql('SELECT * FROM
ProfileTable', [], function(tx, rs) {
  console.log('Data in
ProfileTable:', rs.rows);
}, function(tx, error) {
  console.log('SELECT error: '
+ error.message);
});
< undefined
  Data in ProfileTable: VM55:3
  ▶ {Length: 0, item: f}
> |
```

Figure 67 - Executing a select query in the console

This was when we decided that we should be looking at other alternatives to hosting the backend for *DoseManager*. One feature concerning our mid-term report that we wanted to implement was a Real-time Data Synchronisation feature where individual users could have their unique set of medicine reminders across different Android devices . This was where we selected Firebase to be used, and in particular, Cloud Firestore to handle hierarchical data. It provides real-time synchronisation, which means that any changes in data are immediately propagated to all connected clients. This was a critical feature for *DoseManager*, as we wanted to ensure that users could manage their medicine reminders consistently across various Android devices

One of the issues we faced was when using the forms. When the user clicks on the edit button to edit the Appointment or Medication, it will bring them to an edit page that will display their current data and allow them to change it. However, when the user clicks the update/save button, an error will occur, and the data will not be changed.

Edit Appointment

Appointment Location
Skin clinic

Appointment Date & Time
09/05/2023 03:50 PM

Doctor Name
Dr sam

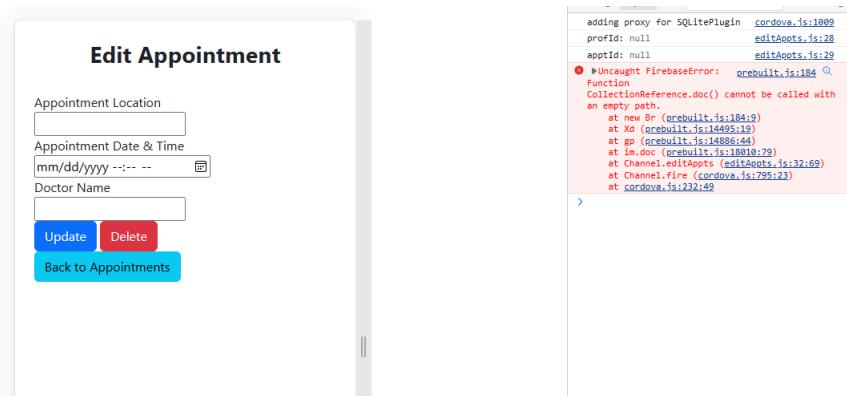


Figure 68 - Technical issues with form submissions

Our team discovered that the form is performing the default submission behaviour in our code, thus it prevents us from redirecting back to our appointments page in this case. We fixed this by adding an `event.preventDefault()` function which is inbuilt in JS, to continue propagating the event, which solved our issue.

```
form.addEventListener("submit", function (event) {  
  // Prevent the default form submission behavior  
  event.preventDefault();
```

Figure 69 - Code used to prevent the issue

Another issue that our team members faced was when displaying the data from the Collection. As we were using the input data, the data was displayed in the format YYYY-MM-DDTHH: MM as shown in the image below. Another issue with the display was that we were not able to differentiate between the missed and upcoming appointments as we were missing the functions to compare the current date time with the appointment date time.

Upcoming Appointments

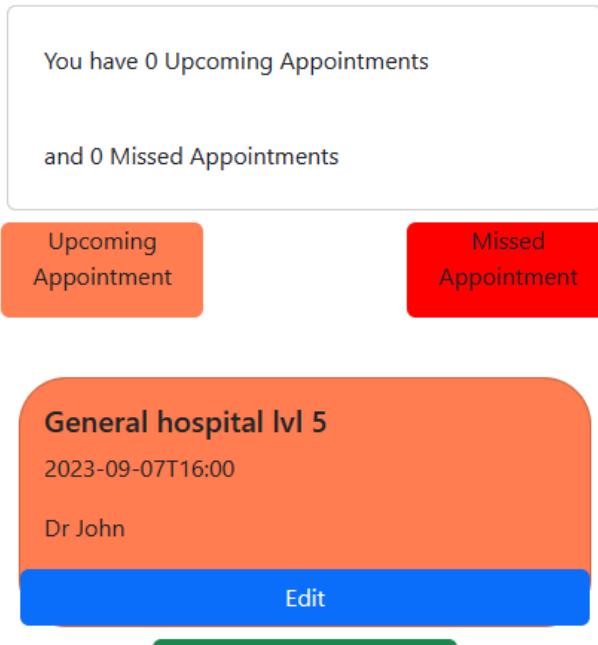


Figure 70 - Appointments page with the weird date and time format

To fix this issue, we created a separate function to separate the returned date into the format Month Date, Year at HH: MM AM/PM. We also created a function to compare the current date with the appointment date, and having 2 different variables, Upcoming and Missed appointments, which act as a counter, whereby if an appointment is missed, the Missed Appointments counter will increase by 1. These variables will then be parsed into the display data and show the correct values.

```
function formatDate(inputDate) {
  const options = { year: 'numeric', month: 'long', day: 'numeric', hour: '2-digit', minute: '2-digit' };
  return new Date(inputDate).toLocaleDateString('en-US', options);
}
```

Figure 71 - Code used to change the date and time display format

```

subCollectionRef
    .orderBy("apptDateTime")
    .get()
    .then(function (querySnapshot) {
        querySnapshot.forEach(function (subDoc) {
            var subDocData = subDoc.data();
            var apptDateTime = subDocData.apptDateTime;
            var currentDateTime = new Date().toJSON();
            var id = subDoc.id;
            var formattedDate = formatDate(apptDateTime);

            console.log(id);
            if(apptDateTime > currentDateTime){
                //populating html page with each appointment card details
                var apptCard = `

                    <div id = "upcomingAppt" class="card mt-4 rounded-5">
                        <div class="card-body">
                            <h5 class="card-title" >${subDocData.apptLocation}</h5>
                            <p id="apptDateTime" class="card-text">${formattedDate}</p>
                            <p id="docName" class="card-text">${subDocData.docName}</p>
                        </div>
                        <button class="edit-button btn btn-primary" data-item-id="${subDoc.id}">Edit</button>
                    </div>
                `;
                //Add to the upcoming appointment counter to be displayed at the top of the page
                upcomingAppts++;
            } else if (apptDateTime < currentDateTime) {
                //populating html page with each appointment card details
                var apptCard = `

                    <div id = "missedAppt" class="card mt-4 rounded-5">
                        <div class="card-body">
                            <h5 class="card-title" id="apptLocation">${subDocData.apptLocation}</h5>
                            <p id="apptDateTime" class="card-text">${formattedDate}</p>
                            <p id="docName" class="card-text">${subDocData.docName}</p>
                        </div>
                        <button class="edit-button btn btn-primary" data-item-id="${subDoc.id}">Edit</button>
                    </div>
                `;
                //Add to the missed appointment counter to be displayed at the top of the page
                missedAppts++;
            };

            apptList.innerHTML += apptCard;
            updateUpcomingApptsHtml(upcomingAppts);
            updateMissedApptsHtml(missedAppts);
        });
    });
}

```

Figure 72 - Code of individual “Collections” created

Initially, when using the Cloud Firestore, our team created individual “Collections” for each of our parts, as we split the work into us working on different parts of the application. For example, we had a “Collection” for Appointments, Medications, Profiles, and Notifications. However, as they were all “Main” collections on their own, we had trouble linking up the Profiles with the rest of the collections. For our application, we wanted each user profile to have their own collection of medications, appointments and notifications.

dosemanager-d0236	Appointments	⋮	8RnCQnlk8woRfGq8CgDt	⋮
+ Start collection	+ Add document		+ Start collection	
Appointments >	8RnCQnlk8woRfGq8CgDt >		+ Add field	
Medicine	RILTm1Nm9zUQ90lu7JEQ		description: "A NEW APPOINTMENT "	
Notifications	RVuLiDS2sz0bc3mZGpxD		name: "APPOINTMENT"	
ProfileTable	fTqZGj42Ms6tzNhHDBZ0			

Figure 73 - Layout of the Firestore Cloud having individual “Collections”

Seen by the image, the different parts had their own collection, therefore we had issues linking them up together to make them unique to each profile.

We solved that issue by having one “Main” collection, the Profiles collection, and having the Appointments and Medications collection be a “Sub” collection under the specific user profile, identifying them by their unique “document” ID.

The screenshot shows the Firestore Cloud interface. On the left, there's a sidebar with a list of collections: Appointments, Medicine, Notifications, ProfileTable, Profiles, and **ProfilesTesting**. The **ProfilesTesting** collection is highlighted with a red box. In the middle, under **ProfilesTesting**, there's a list of documents. One document, **Sg010LgvP4V6SEtFe9cX**, is highlighted with a red box. Under this document, there are two subcollections: **Appointments** and **Medicine**, which are also highlighted with red boxes. On the right, the details for the **Medicine** subcollection are shown, containing the fields **dateOfBirth: "2023-09-15"**, **gender: "male"**, and **name: "test 1"**.

Figure 74 - Firestore Cloud with both collections under an individual Main “collection”

```
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
var firestore = firebase.firestore();

function getProfileIdFromURL(){
  var urlParams = new URLSearchParams(window.location.search);
  return urlParams.get("id");
};

var profileId = getProfileIdFromURL();
console.log(profileId);
const profilesCollectionDocID = profileId;
const profilesCollectionRef = firestore.collection("ProfilesTesting").doc(profilesCollectionDocID);
const subcollectionName = "Medicine"
```

Figure 75 - Code used to create a subcollection under an individual main “Collection”

For example, from the image above, we can see that under the “Main” collection “ProfilesTesting”, the user has the unique “document” ID “Sg010LgvP4V6SEtFe9cX”, and he has his own unique collection of “Appointments” and “Medicine”. His profile also has the 3 fields: “dateOfBirth”, “gender”, and “name” to identify the user for us to display his details in the application so that the user knows that it is his profile.

Another issue we had with linking the different subcollections together with the main Profiles collection was to be able to retrieve the unique document ID of the specific user profile. To solve this issue, when the user clicks on their profile on the profiles page, it will redirect to the home page, and the URL will contain their unique document ID, which we can use to retrieve their unique ID, to be able to parse into the other pages and functions to create subcollection inside their specific document ID.

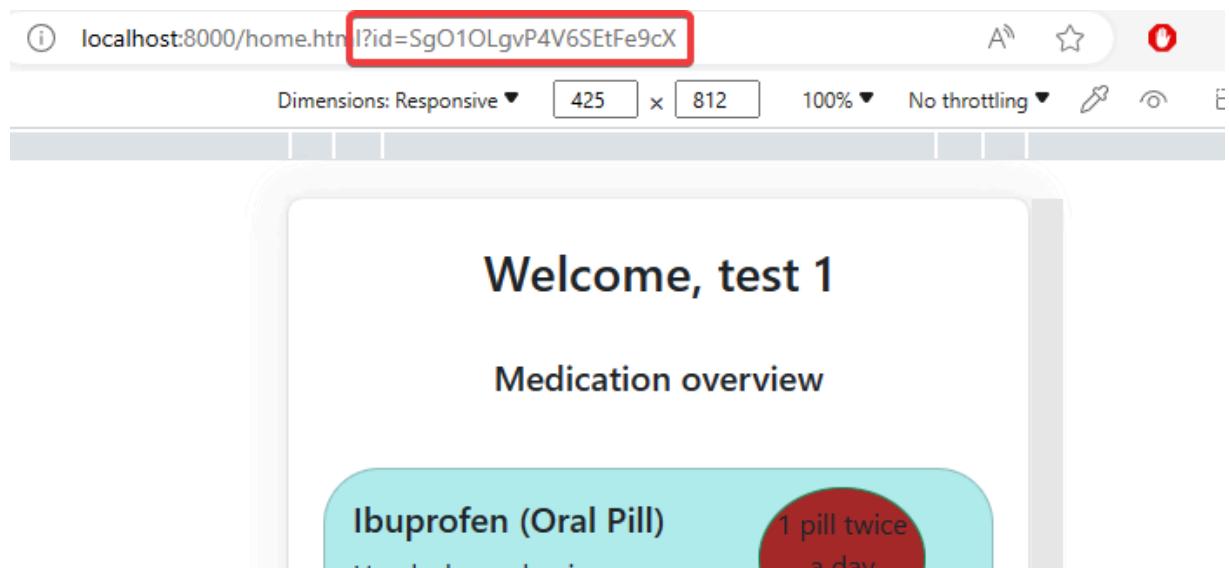


Figure 76 - Adding the profile id to the URL for easier referencing

As for the editing of the Medications and Appointments part, we did the same thing but split it up by adding an & in the URL, followed by the unique medication or appointment document ID. This will allow us to retrieve the different parameters from the URL to parse into the correct functions, fetching the correct data.

```
function getParametersFromURL() {
    var urlParams = new URLSearchParams(window.location.search);
    var profileId = urlParams.get("id");
    var apptId = urlParams.get("apptId");

    return { profileId, apptId };
}

var { profileId, apptId } = getParametersFromURL();
```

Figure 77 - Code used to fetch the id from the URL

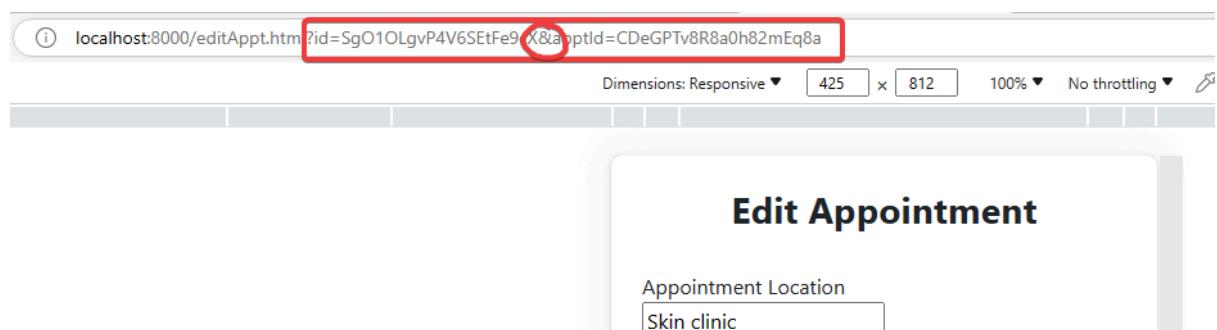


Figure 78 - Added the id of the appointments to be after the profile id

Other technical issues include the Language setting, as we were struggling to find a way to aid us in translating the language to other languages across all pages. We used the Google

Translate widget, which worked to help us translate the language on one single page. However, this means that we would have to add the widget to every page, and the user has to translate each page every time they go to another page. We then found a Google Translate Element script, which is open-sourced to our HTML, and created a function to allow us to translate the pages according to the selected language under our settings(language) page.

```
<!--google translate-->
<script src="https://translate.google.com/translate_a/element.js?cb=googleTranslateElementInit"></script>
```

Figure 79 - Google Translate script added to our HTML

```
function googleTranslateElementInit() {
  new google.translate.TranslateElement(
    { pageLanguage: "en" },
    "google_translate_element"
  );
}
```

Figure 80 - JS function to call the Google Translate

```
//referencing google widget
var languageSelect = document.getElementById("google_translate_element");

// adding event listeners to each button
englishBtn.addEventListener("click", function () {
  languageSelect.querySelector(".goog-te-combo").value = "en";
  //Trigger translation
  languageSelect
    .querySelector(".goog-te-combo")
    .dispatchEvent(new Event("change"));
});

chineseBtn.addEventListener("click", function () {
  languageSelect.querySelector(".goog-te-combo").value = "zh-CN";
  //Trigger translation
  languageSelect
    .querySelector(".goog-te-combo")
    .dispatchEvent(new Event("change"));
});

malayBtn.addEventListener("click", function () {
  languageSelect.querySelector(".goog-te-combo").value = "ms";
  //Trigger translation
  languageSelect
    .querySelector(".goog-te-combo")
    .dispatchEvent(new Event("change"));
});

tamilBtn.addEventListener("click", function () {
  languageSelect.querySelector(".goog-te-combo").value = "ta";
  //Trigger translation
  languageSelect
    .querySelector(".goog-te-combo")
    .dispatchEvent(new Event("change"));
});
```

Figure 81 - Adding the event listeners to the different language buttons

This allowed our team to create a translate function on the settings(language) page, which will translate the current language into the different selected languages across all pages. For our application, we currently have four different languages to choose from: English, Chinese, Malay, and Tamil, as these are the four main languages used in Singapore.

Overall

In conclusion, the backend team encountered several technical issues throughout the development phase. The Cordova Sqlite native plugin was initially intended to be used, but problems with data persistence surfaced. This encouraged the use of Firebase, particularly Cloud Firestore, which allowed for real-time data syncing between several devices.

Additional issues were faced in forms handling, resulting in the implementation of event.preventDefault() to resolve submission errors. Displaying data from the Collection posed format and differentiation challenges, which were addressed by creating specific functions.

Linking separate components of the application within Firestore was a significant challenge. Subcollections were used under user profiles, and the implementations were implemented to fix this. Additionally, distinct document IDs for user profiles were obtained using URL parameters.

Language translation presented an obstacle initially but was overcome by incorporating a Google Translate Element script and creating a translation function, allowing dynamic language changes across all pages. This feature was designed to cater to Singapore's diverse linguistic landscape, offering English, Chinese, Malay, and Tamil options.

Conclusion and Summary

Frontend

In summary, our team, drawing from their experience in our first year's Web Development module, successfully designed a user-friendly layout for the application's front end. While revisiting Bootstrap, which had been studied over a year ago, was necessary during the frontend development process, the team quickly regained proficiency and proceeded with efficiency.

The team knew that despite the extensive work that went into the seamless integration of the backend, the appearance of the application was what would make people want to use the app. Therefore, the main idea we had in mind while designing the app was two things - the appeal and the simplicity. Our biggest issue while working on the backend was conveying our ideas to one another, as the team members had different outlooks on creativity. However, with ample discussions and presentations of each others' ideas, we were able to come to an agreement on what the application should look like and were able to pull it off with our communication skills and teamwork.

Given the time and opportunity, the frontend team would like to make our application even more appealing to users, perhaps by adding animations and more functions to the different pages. We would also like to make the application's transitions and navigation process faster and smoother.

Backend

To summarise the backend for the application, the team managed to set up and establish a connection with Firebase. The Firestore database was set up with all the necessary collections and documents to store and manipulate all the required data. The team was able to collect data from the user for medications and appointments and store them in the database for CRUD operations. These data could then be fetched and displayed to the respective pages. The push notification was also set up using Cordova's local plugin notification 12, with the given time set by the user.

As a result, our backend team managed to produce a functional application for our user's needs. The main pitfall of the back-end implementation was that Cordova was used in conjunction with other libraries, such as React. Thus, there were more avenues to aid in the easier creation of different features. Our team, however, were using Cordova with vanilla HTML, CSS and JS. Thus, we had many issues with implementing our different features. Given the time, our team would have liked to pick up another way to develop our application, possibly using Flutter as it is another widely used software.

Due to our time constraints, the back-end team was not able to implement the ChatBot page. If we had more time, we would have implemented the ChatBot using Firebase as well. We would need to set up a database for the chat history and logs, so we will probably be looking at Firebase in real-time instead of Firestore for its fast data synchronisation. We intend to use chatGPT's API to process the logic behind the chatbot and answer the user's queries.

Overall

As an overall summary of this project, our team has embarked on a harrowing journey to create a mobile application. It has successfully created a working product for our target audience. Our front-end team, given that they had experience working with CSS and Bootstrap, managed to deliver on that end with the positive feedback that we got from our users.

As for the back-end team, we managed to implement it to a working state, given the limited amount of knowledge and time that we had regarding back-end and mobile app development. Our team members have learnt a lot during this period of adversity and gained a better insight into developing software for users.

Through this project, our team members also gained experience in different ways of communication, ensuring that time and resources are managed well, and most importantly, teamwork. One of the pivotal learning experiences was our adoption of the *Agile Methodology* during the software development process. Embracing Agile's iterative approach, we frequently reassessed our progress and adapted to emerging requirements. This not only facilitated improved communication within the team but also ensured that our resources were efficiently allocated and that we remained aligned with our project goals. Through regular sprints and retrospectives, we cultivated a culture of continuous feedback and improvement, enabling us to navigate challenges more effectively.

This allowed our team to deliver the project we planned to create, overcoming various hurdles and hardships. Having been equipped with the knowledge and skills through this project, our team members have a kick-start towards software development and design if they decide to pursue this in the near future.

References

1. APIs. (n.d.). <https://lhncbc.nlm.nih.gov/RxNav/APIs/index.html>
2. Bhandaribhumin. (n.d.). *GitHub* -
bhandaribhumin/cordova-plugin-local-notification-12. GitHub.
<https://github.com/bhandaribhumin/cordova-plugin-local-notification-12>
3. Contributors, M. O. J. T. a. B. (n.d.-a). *Get started with Bootstrap*.
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
4. deLeeuw, A. (2022). 9 best Pill Reminder apps.
<https://www.medicalnewstoday.com/articles/pill-reminder#best-pill-reminder-apps>
5. Gopal, S. (n.d.). Impact of blue colour in UX design: A Closer look.
<https://www.linkedin.com/pulse/impact-blue-color-ux-design-closer-look-sachin-gopal/>
6. Google. (n.d.). *Gantt chart template*. Google Sheets.
https://docs.google.com/spreadsheets/d/1JcX4sHAuBRGsbXIgktxj5n72sMyFQutQyqJ7R_xQCCU/edit#gid=0
7. Overview - *Apache Cordova*. (n.d.).
<https://cordova.apache.org/docs/en/3.0.0/guide/overview/>
8. PharmD, C. a. K. P. a. L. T. W. (2014, January 12). Medication Adherence: A Literature review. *Medscape*. <https://www.medscape.com/viewarticle/514164>
9. Sumithra, S. (n.d.). Association of Low English Health Literacy with Medication Non-adherence among Elderly Singaporeans: Mediating Role of Uncertainty about Prescription Medication Labels.
[https://www.duke-nus.edu.sg/docs/librariesprovider3/iaru/13-poster_sumithra.pdf?sfvrsn=ac2c27d_2#:~:text=Elderly%20\(aged%20%E2%89%A5%2065%20years%20old\)%20in%20Singapore%3A&text=83%25%20take%20at%20least%201,\(%E2%89%A5%205%20prescribed%20medicines\).&text=61%25%20are%20unable%20to%20read%20in%20English.&text=English%20proficiency%20is%20unknown%20for%20those%20who%20can%20read%20English.&text=Will%20make%20up%20almost%2040%25%20of%20the%20population%20by%20year%202050](https://www.duke-nus.edu.sg/docs/librariesprovider3/iaru/13-poster_sumithra.pdf?sfvrsn=ac2c27d_2#:~:text=Elderly%20(aged%20%E2%89%A5%2065%20years%20old)%20in%20Singapore%3A&text=83%25%20take%20at%20least%201,(%E2%89%A5%205%20prescribed%20medicines).&text=61%25%20are%20unable%20to%20read%20in%20English.&text=English%20proficiency%20is%20unknown%20for%20those%20who%20can%20read%20English.&text=Will%20make%20up%20almost%2040%25%20of%20the%20population%20by%20year%202050)
10. *The 10 best medication Reminder apps | Online Doctor*. (2022, June 3). Online Doctor.
<https://www.onlinedoctor.com/best-medicine-reminder-apps/#MyTherapy>

Individual reflections

Jun Rong

I was part of the backend team for this project, and I worked on establishing the connection with Firebase to store and display the medicine details on the home page. Apart from working with Firebase, I helped implement the language change function for the application. Within the backend team, we often bounced ideas and features between each other to implement as well as solve issues faced. Some challenges faced were as we went about our own ways of implementing and creating these backend functions. When it was time to merge the features together there were often many conflicts due to different naming and methods. Moving forward, I definitely feel more confident in working in a team and before implementing any features, I would talk through the steps I will take and iron out any differences in things like variable naming and methods.

Shambhavi

I was a part of the frontend team that worked on *DoseManager*, and I had the chance to contribute to building the UI and overall user experience. This project has allowed me to grow both as an individual and as a computer scientist. It has allowed me to hone my skills in coding efficiently and debugging while also improving on my collaboration and time management skills. One of the key challenges I faced was designing a user-friendly application that could balance aesthetics with functionality. Throughout the project, the iterative process of conducting surveys to gain user insights helped us to tackle this challenge as it allowed us to prioritise the end users. Another issue I faced was during the merging of our commits when we could not resolve certain issues, therefore making me delete my pull requests. Therefore, I had to copy the code I wrote from my branch and send it to my teammates via text so they could incorporate my code into theirs, thus leading to my name not appearing in certain instances on the git logs. Aside from these issues, I am thankful for the supportive teammates who have worked on this project with me, pushing me to be better and to be resilient, for *DoseManager* would be non-existent if it weren't for them. The journey of building *DoseManager* was gratifying, and if provided with an opportunity to contribute to such a project again, I will gladly put in my time and effort for it.

Henry Zhang

My role in the project closely aligned with Shambhavi's tasks within the frontend team for *DoseManager*. This experience has been immensely valuable in my personal and professional growth. Being part of the frontend team offered me the chance to play a significant role in crafting the user interface and enhancing the overall user experience. Through this process, I've seen notable improvements in my coding proficiency and debugging skills. One of the pivotal challenges I encountered was striking the right balance between aesthetics and functionality to create a user-friendly application. To overcome this, we implemented an iterative approach involving user surveys to gather valuable insights. This helped us to build *DoseManager* even better. I have also learnt the various git commands and put them to use in this project. Not only that, I have gained my proficiency in BootStrap. The journey of building *DoseManager* has been deeply rewarding. I really had fun coding, and debugging and also contributed to the writing of the report. If given the

chance, I am committed to investing my time and expertise to contribute to similar initiatives in the future. I would also like to try out backend implementations from then on.

Zhong Da

Coming into this course, I had no experience with doing backend development, let alone mobile development. I had to learn along the way, and thus there were many different challenges which caught me off by surprise. I worked on linking the database from Firebase' firestore with the medical appointments page to add and display elements from the database into the appointments page.

At first, we split up our work into having an individual page to develop for each one of us, which was not very complicated, as we did not need to mind what the rest of us were doing. However, we were met with obstacles when merging our pages together, as many different conflicts arose due to our different naming of variables, classes and IDs. This caused us to have many bugs occurring, which we had to spend a lot of time debugging. On the positive side, this resulted in our backend team being close-knit, and we would also meet multiple times a week to help each other with debugging. Through this course, I have understood what it is like to work in a team and also learnt how difficult it is to develop an ambitious project with no knowledge at all.

We did pull through as a team, and I learned how important teamwork is in developing different software. I have also learnt to always communicate beforehand with team members on how to proceed with certain processes and iron out any issues before starting the work, otherwise, many different complications may arise. I would bring this experience with me to my next and future projects, possibly guiding other group members with less experience.

Benjamin Lee

As Benjamin, the Team Leader of Team 66, I'd like to reflect upon our journey of creating *DoseManager*. My role in this project has been multifaceted, and I feel a deep connection with the progress and challenges we've faced.

From the outset, I viewed my role not as a traditional leader but more as a coordinator. My primary responsibility was to ensure that our team communicated effectively and worked in a coordinated manner. In the initial week, I prioritised establishing a robust communication channel and choosing a platform for our work. This foundational step was crucial, especially considering the diverse backgrounds and skill sets of our team members.

Additionally, I brought to the table my experience as a Certified Scrum Master, transforming our weekly meetings to be structured as Scrum meetings where each member is updated on their development progress, what they plan to work on next or any potential issues.

Upon realising that the majority of my teammates lacked extensive programming expertise, I proposed using Apache Cordova for our project. This decision was based on its capability to allow development without native coding experience in a short time frame. As the only member with relevant development internship experience, I felt a strong obligation to support my teammates. This responsibility was not just about imparting technical knowledge but also about fostering an environment where everyone felt empowered to ask questions and share their ideas.

I took the lead in several technical aspects of *DoseManager*. From exploring backend database options, and collaborating with Jun Rong on its implementation, to spearheading the notification features and leveraging Cordova plugins, I was deeply involved in the development process. Furthermore, managing the GitHub repository, merging code, resolving conflicts, and ensuring that our application was tested thoroughly on mobile devices fell under my purview. Setting up the Cordova environment boilerplate to facilitate everyone's work was another task I initiated.

However, our journey was not without its challenges. One key challenge was the underestimation of the project's complexity. While Apache Cordova promised ease of use with its HTML, CSS, and JS development capabilities, we soon realised that many of its plugins and libraries were outdated. The deprecated SQLite plugin and the older Cordova notification repository by Katzer on GitHub are two notable examples. The fact that we had to navigate through these challenges, find updated libraries, and get them to work was a testament to our team's resilience and determination.

Another notable challenge was the configuration of Android permissions, especially for Android 13. With its specific internet permission and CORS rules, it took a significant effort to ensure our app functioned seamlessly across different Android versions.

In retrospect, the *DoseManager* project has been a tremendous learning experience. Not only did it hone my technical skills, but it also reinforced the importance of teamwork, communication, and adaptability. As the project coordinator, I'm proud of our achievements and grateful for the unique experiences and challenges that shaped our journey.