Developing a Medication Reminder Application: Integrating Alexa and ChatGPT.

CMT403 – MSc Project

MSc Computing

School of Computer Science, Cardiff University

Sai Bhargavi Garikapati

Student ID: 21113772

Project Supervisor: Usashi Chatterji

Moderator: Xianfang Sun

Table of Contents

Table of Figures

List of Tables

Abstract

This project highlights the development of a medication reminder app utilising React Native, Django, integrating Alexa and ChatGPT. The app aims to improve medication adherence by providing users with automated reminders and personalised assistance. The development process involves leveraging React Native to create a cross-platform mobile application, ensuring compatibility across various devices and operating systems. Django is used for backend development, enabling efficient data management and processing. The integration of Alexa and ChatGPT enhances the user experience by supporting voice-activated reminders and intelligent conversational interactions. The product is a medication reminder app that combines advanced technologies to address the challenges associated with medication adherence. Further testing is conducted to assess the effectiveness and usability of the app. This project demonstrates the successful integration of emerging technologies into healthcare applications, providing a comprehensive solution for medication management.

Acknowledgement

I want to thank my supervisor, Usashi Chatterjee. From the beginning of my placement year, she was there guiding me. She was involved from the start of this project, brainstorming ideas with me. She was always there, even when I had last-minute questions. Her constant support, valuable advice, and encouragement were essential in completing this work. I'm truly grateful for everything she did for me.

Introduction

There are a significant number of people using medication reminder apps. A study found that 272 medication reminder apps were available in the App store and Google app stores (Santo et al., 2018). Another study mentioned that most reviewed apps offered a medication reminder feature (Tabi et al., 2019). These findings suggest that medication reminder apps are widely available and commonly used.

Medication reminder apps are recommended for specific populations, such as the elderly, individuals with chronic illnesses, and those with difficulty remembering their prescribed medications due to their quantity or complexity (Brinker et al., 2022). These apps can help improve medication adherence, as electronic reminders have been shown to increase commitment (Knitza et al., 2020). A randomised clinical trial found that medication reminder applications improved compliance in patients with coronary heart disease (Santo et al., 2018).

In addition to improving adherence, medication reminder apps can also have other benefits. For example, one study found that supporting digital therapy through these apps can reduce pain and improve comorbidities such as depression (Knitza et al., 2020). Another study highlighted the potential of diabetes management apps to increase adherence, with some apps including medication reminders for individuals with Type 2 diabetes (Jimenez et al., 2020).

However, not all apps are built equal. The reliability of reminders, the ease of user interface, and the effectiveness of communication play pivotal roles in determining the success of such tools and adding layers of sophistication, like integration with voice assistants. Research findings indicate that voice-activated virtual home assistants (VHAs) such as Amazon Echo and Google Home devices may help promote medication adherence (Pradhan, Mehta & Findlater, 2018) (Sterling, 2019) and intelligent chatbots can elevate the user experience, making adherence a routine and engaging.

This dissertation stems from the dire need to develop a reliable, user-friendly, and technologically advanced medication reminder application. With many existing solutions needing more reliability and user engagement, the business landscape presents a ripe opportunity for a functional and interactive app, leveraging modern tech marvels like Alexa and ChatGPT.

The subsequent section provides an in-depth literature review, comparing it with current solutions and pinpointing the existing voids. The succeeding chapters focus on the project's methodology, tool selection, and the hurdles faced during the app's development.

Aim and Objectives

Aim:

To design, develop, and evaluate a medication reminder application using React Native and Django, which seamlessly integrates voice-assistant capabilities provided by Alexa and the AI (ChatGPT), aiming to improve medication adherence among users.

Objectives:

1. **Requirement Gathering and Analysis:**
    - Conduct a feature assessment of Existing Applications to determine the strengths and weaknesses of various systems and identify any potential gaps or novel features that could be incorporated.
    - Identify the Specific benefits under the constraints of integrating Voice Assistant and AI(ChatGPT) in such applications.
2. **System design on development:**
    - Design a friendly interface for the medication reminder app using react native, injuring access ability and usability.
    - Implement A robust back-end system using Django to manage user data, medication schedules and reminders.
    - Seamlessly integrate Alexa to provide voice-based reminders and interactivity.
    - Incorporate chatGPT to facilitate user queries, guide medication intakes, and provide dynamic conversational support.
3. **Testing and Validation:**
    - Perform unit and integration testing to ensure the reliability and functionality of the application components.

Background Material:

Existing Work:

In recent years, the rise of mHealth applications across platforms such as iOS and Android have become prominent. Our research focus was streamlined towards apps specific to medication reminders, which play an indispensable role in patient care.

To refine the selection, we initiated our exploration using targeted keywords such as "Pill reminder", "medication adherence"; "virtual home assistants"; "public health"; and "Medication Reminder". The minimum criterion for an application to be included in our analysis was the capability to generate a schedule for at least one medication and the provision of notifications. This criterion led us to various applications, which have been dissected in the subsequent sections.

Most of these applications are generously available free of charge, with a minority necessitating in-app purchases or an outright fee. While some apps brandished interconnectivity features, a concern that emerged was an overwhelming interface. Complex interfaces, especially for the senior demographic (50 years and above), can be a deterrent (Becker et al., 2013; Arnhold et al., 2014). A harmonious balance between functionality and usability is imperative.

Moreover, our review process unveiled several things that needed to be corrected. Many apps seemed outdated or had defunct features – indicative of irregular updates or flawed initial programming. Although the majority were free or semi-free, a section required payments, potentially narrowing their accessibility to the broader public. High initial costs, substantial data entry, and hidden charges have historically been cited as reasons for app rejection (Krebs and Duncan, 2015), accentuating the need for thoughtful app monetisation strategies.

The current landscape of medication adherence applications reveals a significant gap. While many apps provide the essential utility of reminders, there needs to be more comprehensive, user-tailored information on drug interactions, side effects, and education. Research findings indicate the potential of voice-activated virtual home assistants (VHAs) like Amazon Echo and Google Home in promoting medication adherence (Beaney et al., 2020; Pradhan, Mehta & Findlater, 2018).

Moreover, compatibility remains a significant concern in the ever-evolving tech sphere. Operating systems are continuously updated, which can render specific applications incompatible with older devices. In this realm, platforms like iOS have a slight edge due to Apple's controlled ecosystem, which ensures consistent updates. However, for any application to be genuinely impactful, it should prioritise backward compatibility to cater to a broader user base.

Considering these findings, while many medication reminder applications exist, few, if any, provide a comprehensive approach that addresses user needs on multiple fronts - from reminders to education. As the journey to build a holistic application unfolds, integrating VHAs with sophisticated language models like ChatGPT could be the key to bridging existing gaps and ensuring optimal medication adherence.

Our comprehensive analysis shows that the current landscape of medication reminder applications offers a limited scope in addressing holistic user requirements, particularly in patient education. Integrating voice-activated systems, such as Amazon Echo or Google Home, with advanced linguistic algorithms like ChatGPT presents a promising avenue for a robust and sophisticated solution. This integration promises timely reminders and equips users with a deeper understanding of their medications, fostering optimal adherence.

Analysing Features:

The following tables provide a comprehensive overview of the existing tools and technologies in the market that support Medication adherence:

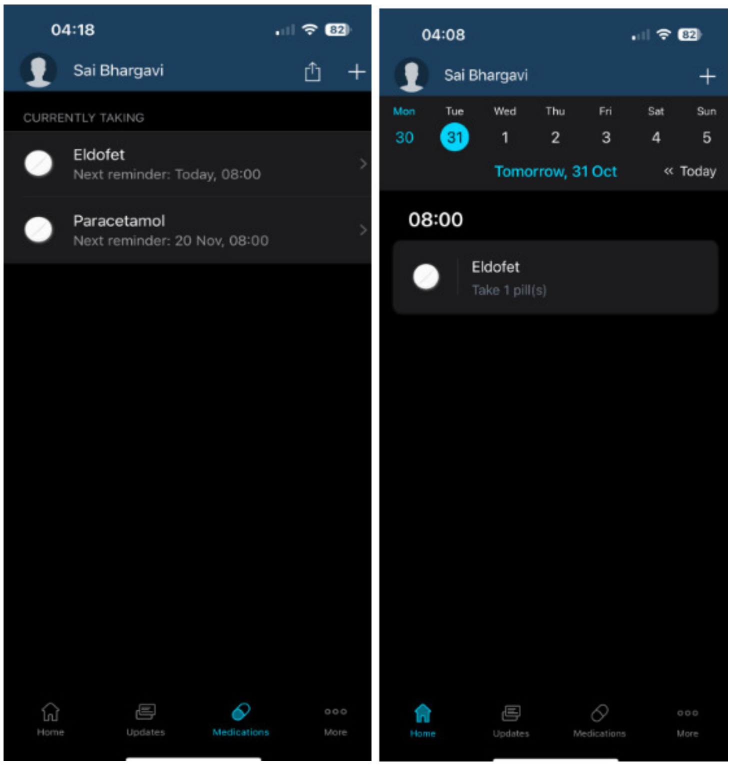
App name/ Reviews	**Pros**	**Cons**	**What I discovered**
Medisafe Pill Reminder **4.7/5 with 11k ratings** MediSafe is a handy app that helps you keep track of your meds. It's got some cool features, but you might need to pay to get the full experience. **Positive reviews:** *"Have recently needed to start on a huge new medication regime, 11 tablets a day. This App has made it possible for me to schedule them over the day, with a constant reminder of what to take when, and mark when taken."* *"I always forget to take my pills on time and sometimes forget if*	□ Medisafe's intuitive design makes it easy for users to input medication details and set reminders. □ Users can sync data across various devices. □ Sends notifications when it's time to restock prescriptions. □ Provides a progress report in a visual PDF format that can	□ Ads in the free version can be distracting. □ Users should not rely solely on the app for drug information. □ Some report issues with notifications on certain devices. □ Some users might find Medisafe challenging initially due to its array of features and	□ Medisafe is more than just a pill reminder; it's a comprehensive tool aiming to make medication management easier, especially for families or caretakers. □ The app places an emphasis on customization, ensuring that the reminders fit comfortably into a user's daily routine.

I've took them, so this is excellent for keeping track of what I'm doing" **Negative Reviews:** *"The rating of this app was quite high, so I decided to download it but unfortunately the application keeps on crashing. Moreover, the loading time for medication entry is very inefficient and illogical. Would not recommend."* "*This app used to be FAB and I'd use it without fail but over the last 4/5 months if I'd I've taken my medication a little bit earlier than set time and marked it as taken now the reminder still goes off. It sorted it self out at one stage but as started doing it again so l've stopped using again."* *"I set a reminder for my medication which I need to take every 2 weeks. I have been using this app for years to do this and I realised that I haven't had a reminder in a while so I check the app and the last time I was reminded to take my medication*	be shared with medical professionals. □ Caregivers can be notified if a dose is missed. □ Offers a variety of playful "Medtones" and reminder sounds, as well as visual customizations. □ Timely medication reminders even if the device is off, with time zone support.	the detailed setup process required for personalizing medication reminders. □ Subscriptions auto-renew, which may lead to unexpected charges if not cancelled in time.	□ The app's design intuitively guides new users, reducing the time needed for initial setup. However, some elder users still faced minor challenges. □ The syncing feature was particularly beneficial for users with multiple devices, but issues arose if one device wasn't updated to the latest app version.

was on the 13th of last month it's not the 25 and I have only just received a reminder. My condition has severely worsened because of my trust in this app they are lucky if they haven't killed someone."			

Table 1: MediSafe App Review

MediSafe App Summary:



04:18

Sai Bhargavi

CURRENTLY TAKING

Eldofet
Next reminder: Today, 08:00

Paracetamol
Next reminder: 20 Nov, 08:00

04:08

Sai Bhargavi

Tomorrow, 31 Oct

08:00

Eldofet
Take 1 pill(s)

Figure 1 MediSafe App UI

Medi-safe Pill Reminder is a comprehensive medication management tool widely endorsed by healthcare professionals. Compatible with Android and iOS, the app seamlessly integrates functionalities like medication reminders, refill alerts and doctor appointment notifications. One of its hallmark features is the "Medfriend" functionality, allowing caregivers or loved ones to assist and support users. The intuitive design aids users in quickly setting up medication details and reminders. Additionally, it facilitates data synchronisation across multiple devices, an essential feature for those using more than one device. Users can generate automatic progress reports, which is ideal for keeping healthcare professionals in the loop. While Medisafe is celebrated for its robust features, some users have reported occasional bugs and challenges during the initial setup due to its features. The app offers a free standard version, but for a more enriched experience, users might opt for the premium version.

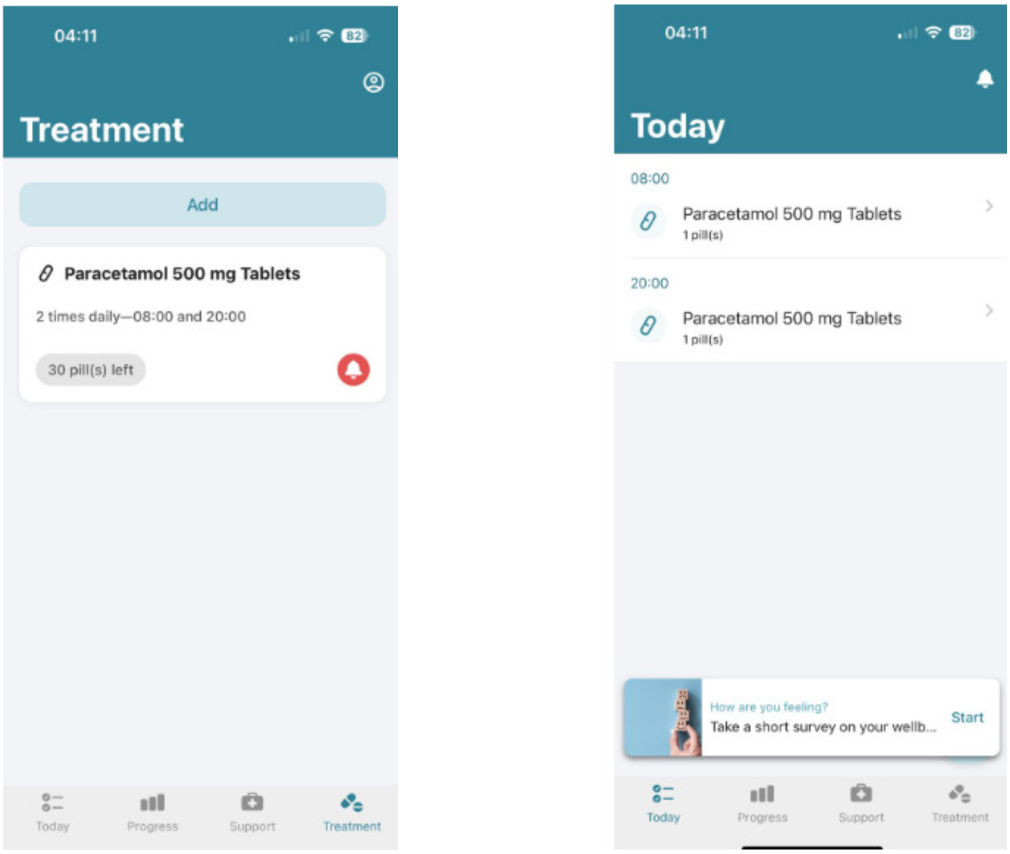
App name/ Reviews	**Pros**	**Cons**	**What I discovered**
Mytherapy Mediation Reminder **4.8/5 with 1.6 k ratings** MyTherapy is an all-in-one medication reminder, mood tracker, and health journal, designed to support and simplify treatment management. **Positive reviews:**	□ Boasts a simple, easy-to-use interface, free from advertisements. □ Combines a pill tracker, mood tracker, and health journal. □ Offers a comprehensive, printable	□ The printable health report, though functional, might lack a polished design that some users expect. □ It does not offer enough personalization options for users who want a more tailored experience.	□ MyTherapy doesn't just remind users about their meds but also encourages a healthy lifestyle by allowing activity reminders. □ The app enables users to document both their physical and

“This App certainly is very helpful in reminding me of taking my medication. Especially at the weekends when I get out of my routine. Very easy to set up and use.” *“Used this app for years - easy to use and alerts me to take my tablets”.* *“I love this app and have found it very useful for scheduling my tablets during my chemotherapy. It's providing a very useful service to patients and it's free!”* **Negative Reviews:** *“App only allows use for items already in database. There is no functionality with regards to inputting medication not already in the database which pretty much renders the app useless for me.”* *“Reminders don't work, I would get 1 reminder and then nothing for days.”*	health report to share with medical professionals. □ It has pre-existing database of medications, which can be a time-saver for users who find their meds listed. □ It Provides an ad-free user experience, eliminating distractions. □ Commitment to stringent European privacy laws and a clear stance against	□ Some users reported not receiving reminders consistently, leading them to miss their medications. □ The app only allows the use of medications that are already in its database. There is no provision to input medication details not already present, limiting its usability for some users. □ The app lacks Dark Mode compatibility, which could	mental well-being over time, providing a holistic view of health. □ MyTherapy has invested in making the app usable for everyone, including those with visual and haptic needs. □ Updates seem to have removed some useful features, such as marking a dose as "taken" from the notification or using the watch for the same purpose. This

“Uninstalled it, check all the notifications settings etc...it simply doesn't work. I've now deleted it and using another app.” *“I had an operation year and am on a lot of pain meds. Thought this would work well except it keeps deleting my medication off my charts so now I don't know when I last took my pain relief. Thanks app!”*	selling user data.	be a strain on users' eyes, especially during nighttime usage. □ Some users reported that the app stopped working after a while, which could be dangerous for those heavily relying on it for vital medication reminders.	has reduced the discreetness and convenience that was previously enjoyed by users.

Table 2 Mytherapy App Review

My Therapy App Summary:



Treatment

Paracetamol 500 mg Tablets

2 times daily—08:00 and 20:00

30 pill(s) left

Today

08:00 Paracetamol 500 mg Tablets 1 pill(s)

20:00 Paracetamol 500 mg Tablets 1 pill(s)

How are you feeling?

Take a short survey on your wellb...

Figure 2 MyTherapy App UI

MyTherapy stands out as a comprehensive medication management tool that goes beyond being just a pill tracker. With a capacity to monitor over 50 measurements, it can keep tabs on mood, weight, blood pressure, and a range of symptoms associated with chronic diseases like arthritis and cardiovascular ailments. The app places an emphasis on collective health management, allowing family members to pitch in, ensuring you never miss a dose. Apart from maintaining a complete log based on pill intake, MyTherapy offers diverse tracking options, from basic vitals like blood pressure to chronic disease markers. A unique feature is its daily motivational picture, aimed at fostering a positive approach to health management. Users appreciate its expansive measurement list, the team function which involves family members, and its straightforward monthly progress report generation. However, a few areas of concern include its complex interface due to myriad options and some reported inconsistencies in daily reminders. User testimonials underscore its ease of setup and use, particularly its utility during treatments like chemotherapy. Yet, a segment of users expressed dissatisfaction due to its limitation of only recognizing medications from a pre-existing database, and some have reported the app to malfunction over time. Despite these issues, MyTherapy prides itself on its ad-free interface and a rigorous commitment to European privacy laws. Moreover, the app nudges users towards a holistic health approach, emphasizing both mental and physical well-being. Notably, some updates seemed to have rolled back features that users found beneficial, reducing its overall convenience.

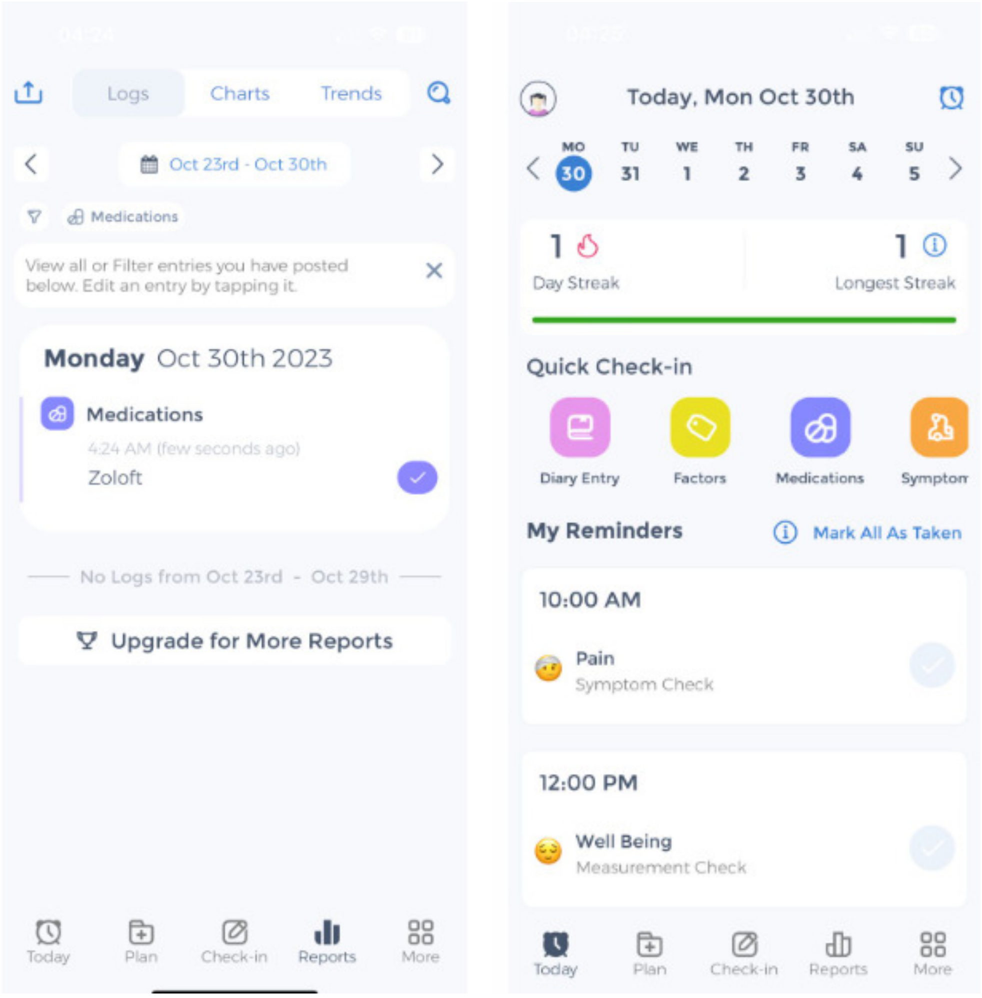
App name/ Reiews	Pros	Cons	What I discovered
CareClinic A comprehensive app to track medications, symptoms, and monitor health habits **4.9/5 with 1.3 k ratings** **Positive reviews:** *"I really like the idea behind this app, I have found it to be very user friendly and has helped me a lot to organize my health management and well-being."* *"Great app, very user friendly and easy to use, intuitive interface. Does what it says on the tin."* *"So good, the app was very easy to use to track my daily health."* **Negative Reviews:** *"The premise would be amazing if the free version wasn't so limited. You can't write notes,*	□ Tracks a wide range of health metrics from medications, symptoms, nutrition, physical activity, sleep, to mood, ensuring a holistic view of one's health. □ Allows users to turn on or off specific modules and trackers to adapt to individual needs. □ You can customize your monthly care plan by adding drugs,	□ Users have reported inconsistencies with the notification system, often failing to update as intended. □ Instances of the app becoming unresponsive, and crashing have been noted by several users. □ The free version of CareClinic is notably limited, restricting notetaking and symptom tracking.	□ More than just a medication reminder, CareClinic positions itself as a complete health planner, bridging the gap between different health domains. □ The app seems to have garnered a plethora of positive reviews, suggesting that it has been well-received by its user base.

have more than three symptoms, and just constant pushes to upgrade. It's not enjoyable to use or easy due to this." *"Purchased with a free trial to see if the app is appropriate for my needs. I've been charged for a year straight away and cannot unsubscribe. There is no contact info on the app to talk to a rep."* *"Too buggy to be helpful. After purchasing a 6-month subscription, I faced numerous issues with CareClinic. Appointments were misdated and often duplicated, the check-in process was confusing, leading to multiple accidental check-ins, and the graphs were not easily interpretable. I mistakenly created several care plans, losing a week's data, and found the user interface challenging on an iPhone. While the idea of multiple care plans is good, its execution can be confusing and it would be beneficial to allow care plan deletion. Despite my initial optimism, I found the app too frustrating to continue using."*	therapies, nutrition, and other information. □ Designed to cater to various health conditions, from mental health disorders to chronic illnesses, ensuring inclusivity. □ Different reminders for a refill, supplements, vitamins, and other meds can be set	□ Users frequently encounter aggressive prompts to upgrade, diminishing the overall experience. □ The interface is particularly challenging on iPhones, and the design makes it easy to unintentionally create multiple care plans. □ The app's navigation is described as not intuitive, making it hard for users to locate specific features or entries.	□ The app boasts built-in databases for medications, symptoms, exercises, and more, streamlining the user experience. □ With features such as localized measurement units, date formats, and languages, the app caters to a global audience.

"I downloaded so this would remind me when to take medication and it doesn't. Also, difficult to navigate and find your diary entries at the beginning unless you look for them in depth."			

Table 3 CareClinc App Review

Careclinic App Summary:



View all or Filter entries you have posted below. Edit an entry by tapping it.

Figure 3 CareClinic App UI

CareClinic is a comprehensive health and wellness application, designed to provide users with a multifaceted approach to health management. This tool seamlessly integrates tracking for a myriad of health metrics, from symptoms, medication, and therapies to daily vitals, nutrition, and physical activity. Notably, users can tailor their monthly care plans, inputting details on drugs, therapies, nutrition, and more. The app further allows for setting various reminders, from refills to supplements and therapies. Caregivers, family members, and healthcare practitioners can also access the care plans and reports, ensuring a collaborative approach to health management. The app stands out for its intuitive interface and efficient symptoms tracking mechanism. However, a few areas of concern have emerged. Users have reported notification inconsistencies and occasional app crashes. Furthermore, the free version of CareClinic has significant limitations, such as restricted note-taking capabilities and symptom tracking, leading to a pushy upgrade experience for users. While the core idea behind multiple care plans is commendable, its execution has been found wanting by some, with reports of confusion and unintentional data losses. Certain design elements, especially on iPhones, have made navigation less than user-friendly. On the brighter side, CareClinic presents itself as more than just a pill reminder. With an emphasis on a holistic health view and a plethora of positive reviews, it appears to be well-regarded by a sizable portion of its users. Inbuilt databases further streamline the user experience. The app's emphasis on global accessibility, with localized units and multilingual support, is also noteworthy.

App name/ Reviews	**Pros**	**Cons**	**What I discovered**
Round Health **4.4/5 with 818 ratings Round Health is a great medication reminder app designed to simplify medication schedules, enhance adherence, and adapt to users' diverse needs** **Positive reviews:** *“Really useful app for someone that is not used to taking medication and therefore forgetful when it comes to taking it or is taking lots of different medication at different intervals.”* *“Simple and helpful - I downloaded this app for the "window of time"*	□ The app offers a consolidated view of all medications and vitamins, facilitating easy tracking. □ Round provides personalized "reminder windows" that adapt to an individual's schedule, eliminating the rigidity of fixed alarms. □ It can handle complex dosing schedules,	□ The app hasn't received updates in a considerable time, leading to questions about its current relevance and compatibility. □ Notifications, crucial for medication reminders, are reported to be inconsistent, potentially causing users to miss their doses. □ Multiple users experienced	□ The app features a specialized flow for birth control packs, indicating a thoughtful design that caters to a wide range of medication types. □ The Round app offers a flexible system suitable for various medication schedules. However, there's potential for improvement in terms of

feature and it helps a lot. One reminder is never enough for me, especially when I look at my phone while I'm doing something else. Thanks for helping to take medicine simple even with my busy schedule” **Negative Reviews:** *“Nice idea but doesn't appear to be supported any more?”* *“For an app that needs to be more reliable than I am at remembering to take my medication, unreliable notifications mean I'm deleting it and relying on something else.”* *“Shame it constantly crashes. Have to keep deleting and reinstalling because it won't open.”* *“This app hasn't been updated in 3 years, the developer's website no longer exists, and none of the online services function as a result. It does*	catering to various medicinal regimens. □ Even if one misses the initial reminder, the app sends follow-up notifications to ensure adherence. □ The app offers streak counting and historical data, allowing users to monitor their medication-taking consistency.	frequent app crashes, with some needing to repeatedly delete and reinstall the app. □ Registration across multiple Apple devices has proven challenging for some users, impacting its utility across devices.	automatically calculating dosing times based on medicine frequency and initial input date. □ The interface of the app is user-friendly and intuitive, with effective notification mechanisms for medication reminders, which many find particularly useful. □ The app goes beyond just medicine reminders. It can be tailored to remind users of other

still work locally, and is actually pretty good for this, but I would look elsewhere for this kind of service." *"Useless. Downloaded on my iPad but it was for use across all my apple devices including watch. To link them you need to register Unable to register, made multiple attempets over course of 20 hours using both email and google options."*			activities like brushing their teeth, attending clinics, and even personal errands like getting a haircut. This adaptability makes it a significant tool for those who rely on it.

Table 4: Round Health App Review

Round Health Summary:

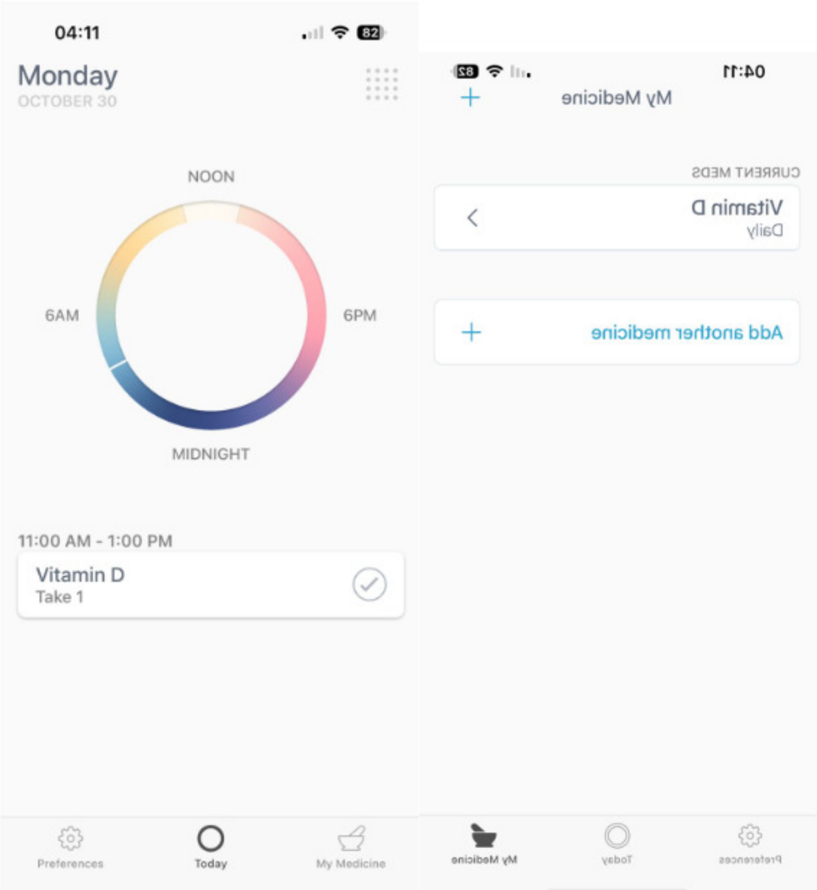


Figure 4 Round Health UI

Round Health, available exclusively for iOS, is designed to streamline and adapt to diverse medication schedules. With features like personalized "reminder windows" that adjust to individual routines and a unique flow catering to birth control packs, the app displays a thoughtful design. However, it isn't without flaws. Users have voiced concerns regarding its compatibility and relevance due to the lack of recent updates. Inconsistent notifications, a pivotal feature for a medication reminder app, further hampers its reliability. Other reported issues include frequent app crashes, necessitating reinstallation, and challenges in registering across multiple Apple devices. While the app's user-centric features are commendable, these technical concerns, combined with indications of it being potentially unsupported (given the developer's website is inactive), suggest a need for caution among potential users.

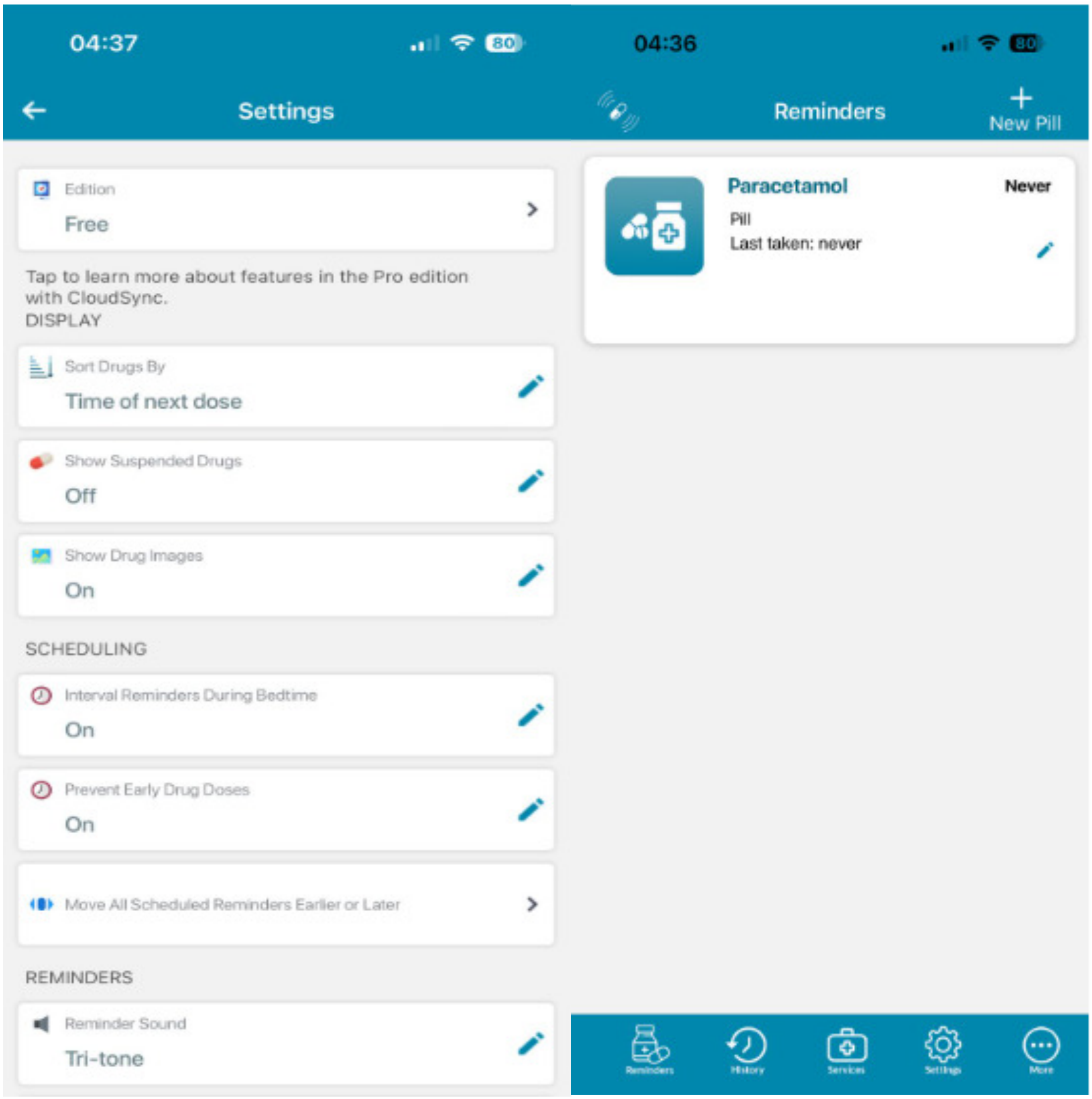
App name/ Reviews	Pros	Cons	What I discovered
Dosecast - Pill Reminder & Med **4.0/5 with 29 ratings** **Positive reviews:** *“Simple to use, love it.”* *“I now take a lot more medications and this is an excellent app to use to remind myself when to take them. It has 2 reminder alarms and you set them with the times you want.”*	□ Reliable notifications that function without an internet connection and adjust according to time zones. □ Customizable scheduling options to fit varied medication regimes. □ Features "smart silencing" to prevent alarms during sleep or specified times. □ Ensures user privacy with encrypted drug information and no	□ Issues with functionality on older iPhone models like iPhone 4s. □ Recent updates have negatively impacted the app's performance for some users. □ Essential features, such as checking medicine history, are locked behind the pro version. □ Cost associated	□ It is generally appreciated for its simplicity and effectiveness in providing reminders for medication □ It Serves a comprehensive audience, from individual users to entire families, even including pets in its scope. □ The pro edition amplifies functionality with cloud synchronization , advanced drug type tracking, dose histories, and multi-person support.

“A lot of the features are available as the free version but if you want more in-depth information, you can subscribe for under £3 per month. This app is worth downloading.” *“As I take about 18 tablets a day, this App is a God send. Easy to use and apply.”* **Negative Reviews:** *“can't amend number of remaining pills - just locks up - is it because I have iPhone 4s or is everyone experiencing a non operative app. This app used to be*	collection of personal data. □ CloudSync in the pro edition ensures synchronization across multiple devices. □ Multi-person support allows for tracking medication for multiple individuals, including pets.	with the pro edition might be a concern for some users. □ Potential in-app purchases can be seen as a downside for users who prefer fully unlocked applications. □ Limitations or features might vary based on the user's region, which could be a disadvantage for some.	□ It prioritizes user privacy, ensuring no personal data collection and encrypted drug details. □ The free version of the app offers a decent number of features, but more in-depth information requires a subscription. □ Some users have experienced functional issues, especially after updates, indicating potential compatibility or stability problems.

excellent but recent updates have killed it...time to delete!” *“This app DOESN'T need background refresh as all it needs are local notifications!!!! ”* *“Need to buy pro version for lots of basic stuff like checking medicine history.”*			□ There's a sentiment that some basic features should be included in the free version rather than reserved for the pro version.

Table 5 DoseCast: Pill Reminder App Review

Dose Cast Summary:



Edition
Free

Tap to learn more about features in the Pro edition with CloudSync.

DISPLAY

Sort Drugs By
Time of next dose

Show Suspended Drugs
Off

Show Drug Images
On

SCHEDULING

Interval Reminders During Bedtime
On

Prevent Early Drug Doses
On

Move All Scheduled Reminders Earlier or Later

REMINDERS

Reminder Sound
Tri-tone

Paracetamol
Never
Pill
Last taken: never

Figure 5: Dose Cast APP UI

Dosecast - Pill Reminder & Med is recognized for its simplicity in providing medication reminders. Its features, such as reliable notifications that adjust according to time zones and customizable scheduling options, are praised by users. However, the app isn't without its flaws. Users of older iPhone models like the 4s have reported functionality issues. Additionally, recent updates seem to have affected the app's performance negatively for some. There is also notable discontent among users regarding essential features, such as checking medicine history, being locked behind the pro version. The associated costs with the pro edition, combined with the limitations in the free version, raise concerns. The app's potential for in-app purchases may also deter users who prefer fully unlocked applications. Overall, while Dosecast has its merits, the balance tilts towards the areas that need improvement.

Using Appstore, I selected the top five highly rated apps and delved into them to understand their offerings. Each of these apps is downloadable at no cost, though several of them offer extended features via subscription plans. Table below shows the features available to paying members, although some of the apps were completely free and did not require in-app purchases. I navigated these apps with a focus on user-friendliness and, during this process, cataloged their strengths and weaknesses. The core functionalities I identified could serve as inspiration for the development of my app, allowing me to integrate the best aspects and refine them further to stand out from the rest.

Features	*Medisafe*	*MyTherapy*	*CareClinic*	*Round*	*Dosecast*
Reliable Notifications	✓	✓	✓	✓	✓
Flexible Scheduling	✓	✓	✓	✓	✓
Customizable Dose Amounts and Instructions	✓	✓	✓	✓	✓
Track and Log Medications	✓	✓	✓	✓	✓
Medication Database or Custom Entry	✓	✓	✓	✗	✓
Reminders for Refills	✓	✓	✓	✓	✓
Multiple Device Sync	✓	✗	✗	✗	✓
Support for Multiple Users or Profiles	✓	✗	✓	✗	✓

Backup and Data Retrieval	✓	✗	✗	✓	✓
ChatGPT Integration for Medication Info and potential drug interactions	✗	✗	✗	✗	✗
Voice Over Reminders (e.g., Alexa)	✗	✗	✗	✗	✗
Free Version (No Advertisements)	✗	✓	✗	✓	✗

Table 6: Features Analysis of various Apps

The exploration into medication reminder apps revealed a digital landscape where user convenience and adaptability are paramount. The highly rated apps on the Appstore were critically assessed to understand their features and the value they bring to users.

A recurring theme among these apps is the reliability of notifications, with all five apps having this essential feature. The ability to customize schedules and doses further testifies to the user-centric design of these apps. Features like multi-device synchronization and support for multiple profiles cater to a diverse audience, from individual users to entire families.

While some apps offer a vast database for medication entries, a few others lack this feature, indicating a possible area of improvement. The need for backup and data retrieval was apparent in most, ensuring users do not lose their medication logs. Despite advancements, features such as ChatGPT for drug interaction insights and voice-over reminders are not widely adopted in the market.

On the downside, while these apps can be downloaded for free, the full spectrum of their features is often hidden behind a paywall. Although economically viable for developers, this subscription model might deter potential users seeking comprehensive features without additional costs.

Drawing insights from the critical assessment, there is ample room for innovation. An ideal app would seamlessly integrate the best features from the existing ones while introducing new, unique features, and possibly, eliminating the constraints of in-app purchases. By looking at what's missing in the current market and improving the user experience, there's a good chance to create a much better app.

Approach

The issue that the project sought to address is the challenge patients face in consistently adhering to medication schedules amidst busy lifestyles and multiple commitments. To address this, the project aimed to combine smartphone applications with voice-assisted reminders and AI conversational interfaces. I developed a Medication Reminder app using React Native, a cross-platform mobile application framework, to achieve this goal. The backend infrastructure was built using Django, which provides a robust framework for database management, further interfaced via REST APIs to ensure a seamless and efficient data exchange.

I chose to use React Native because of its flexibility to support both the iOS and Android platforms with a single codebase. This not only ensured consistency across platforms but also made the most use of the time and resources for development. Because Django is known for its security and scalability, which is important given the delicate nature of medical data, it was chosen. The purposeful decision to design the APIs using a RESTful structure was made to guarantee future flexibility and simple integration with a range of tools and platforms.

Alternative frameworks for mobile development exist. Xamarin, renowned for native performance, and Flutter, with its extensive UI features, were strong rivals. But given the size of our project and its abundance of libraries and community, React Native seemed to be more feasible. While Node.js's asynchronous architecture was taken into consideration for the backend, Django's pre-built features, like the admin interface and ORM, were more suitable for quick development and effective handling of prescription data.

Upon evaluation, Flutter, despite its expressive UI, was set aside in favour of React Native's maturity and expansive ecosystem. Similarly, Node.js, though promising in non-blocking I/O operations, did not align as seamlessly with our immediate requirements as Django did. These decisions were made to optimize development time, maintainability, and to best fit project’s objectives.

Application of the Chosen Approach

Early research in the field revealed that many patients frequently forget or put off taking their medications for a variety of reasons, such as complicated prescription schedules or plain old forgetfulness.

I found certain shortcomings after analysing several current medicine reminder apps, such as clunky user interfaces, unscalable backends, or restricted cross-platform compatibility.

Subsequent analysis revealed the different technologies behind these applications, providing insight into industry norms and possible areas for development.

Rationale for Chosen Technologies:

Backend Development:

Node.js is a JavaScript runtime that allows developers to build scalable, high-performance applications. It is known for its event-driven, non-blocking I/O model, which makes it suitable for building real-time applications and APIs. Node.js has a large ecosystem of libraries and packages, making it easy to find and integrate third-party modules into projects (Ravi & S, 2023).

Ruby on Rails, often referred to as Rails, is a popular web development framework written in Ruby. It follows the Model-View-Controller (MVC) architectural pattern and emphasises convention over configuration. Rails provides ways and abstractions that simplify the development process and promote code reusability. It strongly focuses on developer productivity and includes features such as automatic code generation and a built-in ORM (Object-Relational Mapping) (Ravi & S, 2023).

Flask is a lightweight web framework written in Python. It is designed to be simple and easy to use, making it a popular choice for small to medium-sized projects. Flask provides the essential tools and features needed to build web applications, but it is also highly extensible, allowing developers to add additional functionality as required. It does not enforce any specific project structure or architecture, giving developers more flexibility in organising their code (Patkar et al., 2022).

Django is a high-level web framework also written in Python. It follows the MVC architectural pattern and includes many features and tools for building complex web applications. Django provides a batteries-included approach with many built-in features such as an ORM, authentication system, and admin interface. It promotes best practices and follows the "Do not Repeat Yourself" (DRY) principle, which helps developers write clean and maintainable code.

Django also has a large and active community, which means plenty of resources and third-party packages are available (Ravi & S, 2023). When comparing these frameworks, it is essential to consider factors such as ease of use, performance, scalability, community support, and available resources. While all these frameworks have their strengths, Django stands out as compelling and feature rich. The built-in admin interface, for example, allows developers to quickly create a fully functional admin panel for managing data without having to write any code. Django also includes a powerful ORM that simplifies database operations and promotes good database design practices. Regarding performance, Django is known for its scalability and ability to handle high-traffic loads. It has built-in caching mechanisms, support for database connection pooling, and the ability to scale horizontally by adding more servers.

Django's architecture allows easy integration with other technologies, such as caching systems and message queues, to improve performance. Django has a large and active community, which means plenty of resources and third-party packages are available. The official documentation is comprehensive and well-maintained, making it easy to find answers to common questions and learn new concepts. The community also organises regular conferences, meetups, and online forums where developers can share their knowledge and collaborate on projects.

In conclusion, while Node.js, Ruby on Rails, and Flask are popular web development frameworks, Django stands out as a powerful and feature-rich framework with a strong emphasis on best practices and code reusability. Its batteries-included approach, performance scalability, and active community make it a compelling choice for building complex web applications.

Frontend Development:

React Native is a popular framework for building cross-platform mobile applications. It allows developers to write code in JavaScript and use a single codebase to create apps for both iOS and Android platforms. React Native leverages native components, resulting in high-performance applications that resemble native apps (Singh & G, 2021). It also offers a hot-reloading feature, which allows developers to see the changes they make in real time, making the development process more efficient (Lachgar et al., 2022).

React Native also has a large and active community, which means developers can find plenty of resources and support when working with the framework (Lachgar et al., 2022).

On the other hand, Flutter is a framework developed by Google that allows developers to build high-quality native interfaces for iOS and Android using a single codebase. Flutter uses the Dart programming language and provides a rich set of pre-built widgets, making it easy to create visually appealing and responsive user interfaces (Sandesara et al., 2022).

Flutter also offers a hot-reloading feature, similar to React Native, which speeds up the development process (Lachgar et al., 2022). One of the critical advantages of Flutter is its performance. Since Flutter apps are compiled into native code, they can achieve near-native performance (Sandesara et al., 2022).

Additionally, Flutter has a growing community and strong support from Google, which ensures its continued development and improvement (Lachgar et al., 2022).

Cordova, also known as Apache Cordova or PhoneGap, is a framework that allows developers to build mobile applications using web technologies such as HTML, CSS, and JavaScript.

Cordova wraps the web code in a native container, allowing it to run on multiple platforms (Pandey et al., 2022). One of the main advantages of Cordova is its ability to leverage existing web development skills, making it accessible to a wide range of developers (Pandey et al., 2022).

However, Cordova apps may not have the same level of performance as native apps, as they rely on web views to render the user interface (Pandey et al., 2022).

Xamarin is a framework that allows developers to build cross-platform mobile applications using C# and the .NET framework. Xamarin provides a native user interface and access to native APIs, resulting in high-performance apps that closely resemble native applications (Singh & G, 2021). Xamarin also offers a shared codebase, allowing developers to reuse code across different platforms (Vishwakarma, 2023).

However, Xamarin has a steeper learning curve than other frameworks, as it requires knowledge of C# and the .NET framework (Vishwakarma, 2023).

Additionally, Xamarin has a smaller community than React Native and Flutter, which may limit the availability of resources and support (Lachgar et al., 2022).

Ionic Framework is an open-source framework that allows developers to build cross-platform mobile applications using web technologies such as HTML, CSS, and JavaScript. Ionic uses web components and is not compiled into native code, which means that a significant portion of the code can be reused across different platforms (Singh & G, 2021).

Ionic also provides a rich set of pre-built UI components, making it easy to create visually appealing and responsive user interfaces (Singh & G, 2021). However, since Ionic apps rely on web views to render the user interface, they may not have the same level of performance as native apps (Singh & G, 2021).

In conclusion, React Native is a popular choice for cross-platform mobile application development. It offers high-performance applications, hot reloading for efficient development, and large and active communities for support.

REST API Design and its Relevance

Rest APIs, or Representational State Transfer Application Programming Interfaces, are widely used in web applications for various reasons. One of the main advantages of using REST APIs is their ability to facilitate communication between different applications or systems (Mushtaq et al., 2022). RESTful interfaces often implemented using REST APIs offer several benefits (Alghamdi et al., 2022). They provide a standardised and uniform way of accessing and manipulating resources, making it easier for developers to interact with the API (Alghamdi et al., 2022).

REST APIs also support stateless communication, meaning that each request from a client to the server contains all the necessary information, reducing the need for server-side storage and improving scalability (Alghamdi et al., 2022). Another advantage of using REST APIs is their flexibility and compatibility with different technologies and platforms (Mushtaq et al., 2022). REST APIs can be accessed using standard HTTP methods, such as GET, POST, PUT, and DELETE, making them compatible with various programming languages and frameworks (Mushtaq et al., 2022). This compatibility allows developers to build applications using their preferred technologies while still being able to communicate with other systems through the REST API (Mushtaq et al., 2022).

In addition to facilitating communication and compatibility, REST APIs also play a crucial role in testing and security. Automated testing of REST APIs has become an essential practice in software development (Banias et al., 2021). Test cases are developed to validate the API calls and ensure that the API functions as expected (Banias et al., 2021).

This automated testing helps identify and fix any issues or bugs in the API, improving the overall quality and reliability of the software (Banias et al., 2021). REST APIs also have security implications. They are used in cloud and web services, where security is critical (Sanjana et al., 2022). REST APIs can be secured using authentication and authorisation mechanisms, such as OAuth or API keys, to ensure that only authorised users or applications can access the API (Sanjana et al., 2022).

Additionally, REST APIs can be subjected to security checks to identify and mitigate potential vulnerabilities (Sanjana et al., 2022). Furthermore, the design of RESTful APIs can impact their understandability and ease of use. Studies have been conducted to investigate the impact of RESTful API design rules on the understandability of web APIs (Bogner, 2023). The results suggest that adhering to design rules can improve the understandability of APIs, making them easier to use and integrate into applications (Bogner, 2023).

In summary, REST APIs are widely used in web applications due to their ability to facilitate communication between different systems, their compatibility with various technologies, their support for automated testing, their security features, and the impact of their design on understandability. These advantages make REST APIs popular for developers when building web applications.

In conclusion, the combination of Django's robust backend capabilities and React Native's cross-platform frontend efficiency makes them the optimal technologies for the development of the Medication Reminder App, ensuring a seamless, scalable, and user-friendly experience.

Requirements

To be able to effectively iterate and improve the application, a set of functional and non-functional requirements must be generated. This allows to gather a clear understanding of the work that needs to be done, which requirements are more critical than others.

Functional Requirements

The Requirements need to be prioritized to ensure that the most amount of value is delivered for the application users. The requirements were prioritized using the MoSCoW technique. The MoSCoW technique is commonly used for prioritizing requirements in various fields, including software development, healthcare, and waste management (Guerreiro et al., 2023; Hudaib et al., 2018; Achimugu et al., 2021; Limaylla et al., 2021). The technique categorizes requirements into four groups: Must have, should have, Could have, and Won't have (Farshidi et al., 2021).

Must Have

FR1: User Management
□ Requirement: User registration and Authentication
□ Explanation: Users must register with personal details and log in with credentials. The system should include a password recovery feature.
□ Acceptance Criteria:
o Users can seamlessly sign up using personal details.
o Users can securely log in using their credentials.
o Users can recover/reset their password in case they forget it.
□ Priority: High

FR2: Medication Management
□ Requirement: Medication Input and Management
□ Explanation: Allows users to input, edit, and delete medication details. The system should also display medication schedules visually.
□ Acceptance Criteria:
o Users can easily add new medication details.
o Users can modify existing medication records.
o Users have the option to delete medication details.
o Medication schedules are displayed in an easily understandable visual format.
□ Priority: High

FR3: Notifications
□ Requirement: Reminder and Notifications
□ Explanation: Push notifications for medication timings with options to snooze, dismiss, and record intake actions.
□ Acceptance Criteria:
o Users receive timely push notifications for their medication timings.
o Notifications provide options to snooze, dismiss, or record intake actions.
□ Priority: High

Should Have

FR4: Chatbot Interaction
□ Requirement: Chatbot Feature with AI
□ Explanation: AI-powered chat feature for user queries about medication. Should include an escalation mechanism for complex issues.
□ Acceptance Criteria:
o Users can interact with an AI-powered chatbot for general medication-related queries.
o In case of complex issues, the chatbot escalates the query to a higher support level or suggests alternative solutions.
□ Priority: Medium

FR5: Alexa Integration
□ Requirement: Integration with Alexa
□ Explanation: Sync medication reminders with Alexa. Allows voice commands for setting, editing, and receiving voice reminders.
□ Acceptance Criteria:
o Users can sync the app's reminders with Alexa.
o Users can use voice commands to set, modify, and get reminders via Alexa.
□ Priority: Medium

Non-Functional Requirements

NFR1: Performance
□ Requirement: Fast Response Time
□ Explanation: The application should respond swiftly to user interactions without noticeable lag.
□ Acceptance Criteria:
o 95% of user interactions should result in a response within 2 seconds.
□ Priority: High

NFR2: Security
□ Requirement: Data Encryption
□ Explanation: User data, especially personal and medication details, should be encrypted both in transit and at rest.
□ Acceptance Criteria:
o All user data must be encrypted using industry-standard encryption algorithms.
□ Priority: High

NFR3: Usability
□ Requirement: User-Friendly Interface
□ Explanation: The application's user interface should be intuitive and easy to use, even for non-tech-savvy individuals.
□ Acceptance Criteria:
o All the users in the user personas should be able to use the app.
□ Priority: High

NFR4: Integration
□ Requirement: Seamless Alexa Integration, and chatGPT
□ Explanation: The application's integration with Alexa, chatgpt should be smooth, ensuring that voice commands and texts are accurately recognized and processed.
□ Acceptance Criteria:
o Voice commands via Alexa should have a success rate of 95% or higher.
o Any integration issues should be resolved within 24 hours of being reported.
o All the users should be able to chat with the AI
o The AI should only respond to the medical questions.
□ Priority: Medium

NFR7: Backup and Recovery
□ Requirement: Regular Data Backups
□ Explanation: The application should have mechanisms in place for regular backups to prevent data loss.
□ Acceptance Criteria:
o Backups should be taken at least once daily.
□ Priority: Medium

Design

Use Case Diagram

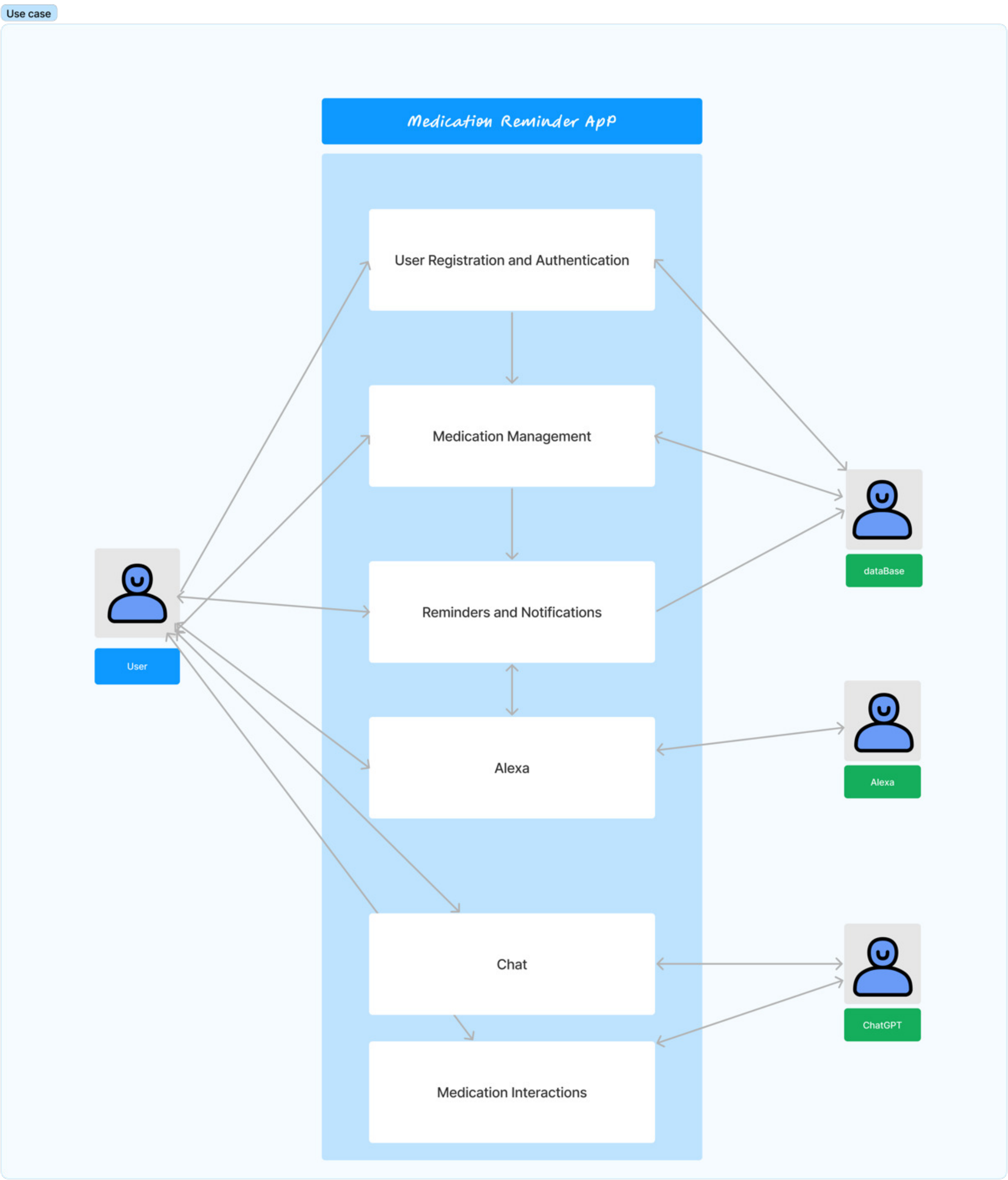
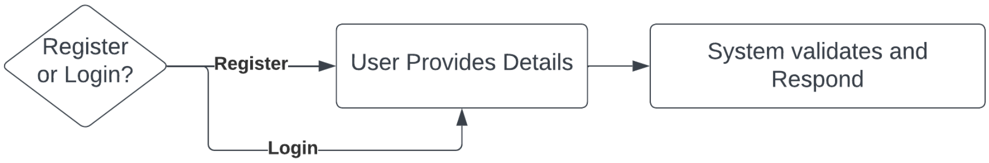


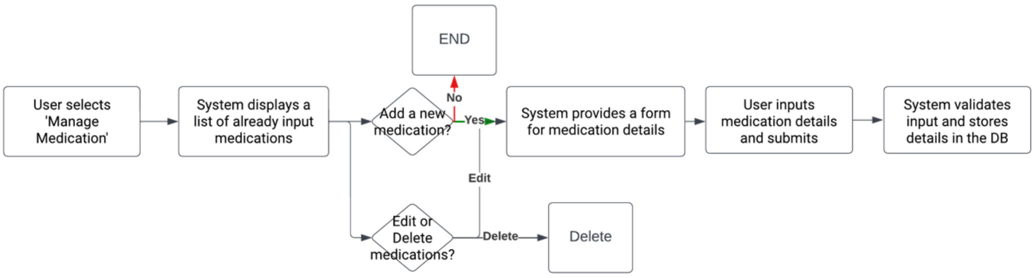
Figure 6: Use Case Diagram

In Figure 6, we can see the use cases depicting the interaction between the user, database ,Alexa and chatGPT with the Medication Reminder Application. These help us understand the user's needs and create a rough plan for functionality and user expectations. In the use case diagram, we can see the case overview, and below, we go into more detail about each one.

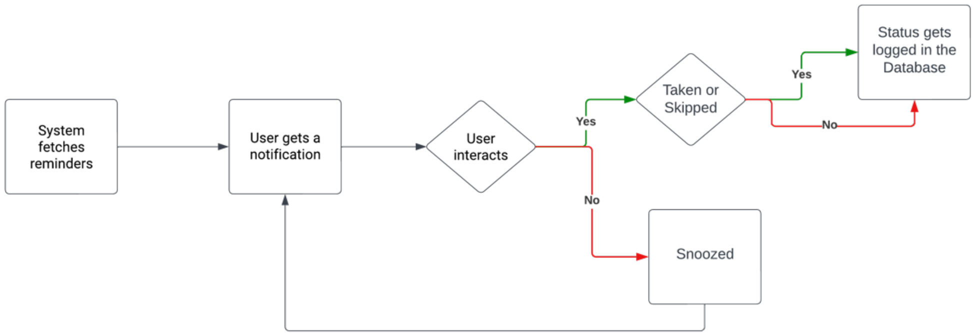
1. User Registration and Authentication

Category	Description
Use Case ID	UC-001
Use Case Name	User Registration and Authentication
Rating	High
Purpose	To allow users to register, log in, and recover their password.
Main Actor	User
Secondary Actors	Database (DB)
Pre-Conditions	None for registration. For login, user must be registered.
Trigger	User selects the option to register or log in.
Basic Flow	 Register or Login? Register Login User Provides Details System validates and Respond
Related Use Cases	UC-002 (Medication Management)

2. Medication Management

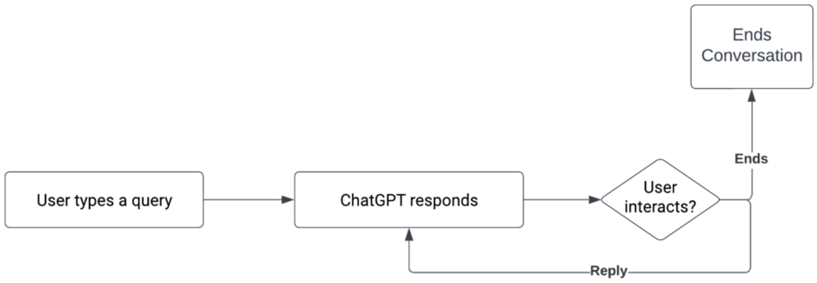
Category	Description
Use Case ID	UC-002
Use Case Name	Medication Management
Rating	High
Purpose	Allow users to input, edit, delete, and view medication details.
Main Actor	User
Secondary Actors	Database (DB)
Pre-Conditions	User must be authenticated and logged into the system.
Trigger	User selects the option to manage medications from the main dashboard.
Basic Flow	 User selects 'Manage Medication' System displays a list of already input medications Add a new medication? No END Yes System provides a form for medication details User inputs medication details and submits System validates input and stores details in the DB Edit Edit or Delete medications? Delete Delete
Related Use Cases	UC-003 (Reminder and Notifications)

3. Reminder and Notifications

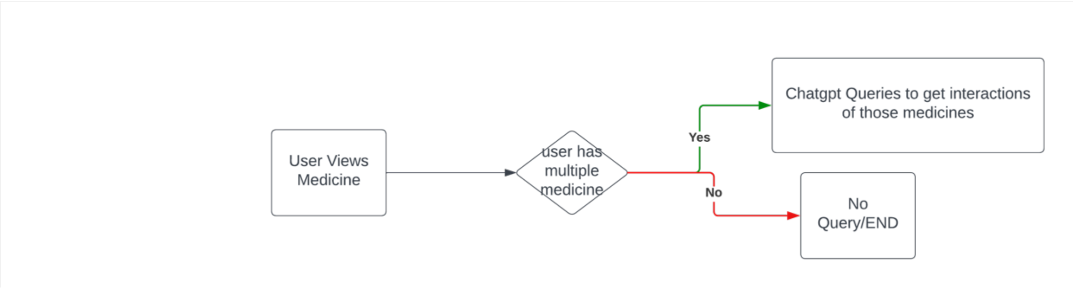
Category	Description
Use Case ID	UC-003
Use Case Name	Reminder and Notifications
Rating	High
Purpose	Notify users about medication timings and allow interactions with notifications.
Main Actor	User
Secondary Actors	Database (DB)
Pre-Conditions	User has set medication reminders.
Trigger	Medication time approaches.
Basic Flow	 System fetches reminders User gets a notification User interacts Yes Taken or Skipped Yes Status gets logged in the Database No No Snoozed
Related Use Cases	UC-002 (Medication Management)

4. Chatbot Interaction

Category	Description
Use Case ID	UC-004
Use Case Name	Chatbot Interaction
Rating	Medium
Purpose	Provide AI-driven responses for user queries related to medication.
Main Actor	User

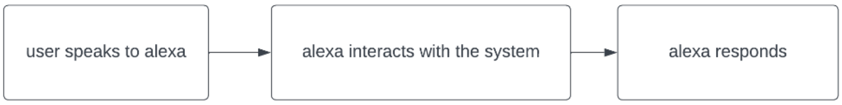
Secondary Actors	ChatGPT
Pre-Conditions	User has a query.
Trigger	User initiates chat with ChatGPT.
Basic Flow	 User types a query ChatGPT responds User interacts? Ends Ends Conversation Reply
Related Use Cases	UC-005 (Medicine Information with ChatGPT)

5. Medicine Information with ChatGPT

Category	Description
Use Case ID	UC-005
Use Case Name	Medicine Information with ChatGPT
Rating	Medium
Purpose	Inform users about specific medicines and potential interactions with other medications.
Main Actor	User
Secondary Actors	ChatGPT, Database (DB)
Pre-Conditions	User seeks information about a specific medication.
Trigger	User views medication info, chatgpt get medicine interactions if user has multiple medicines.
Basic Flow	 User Views Medicine user has multiple medicine Yes Chatgpt Queries to get interactions of those medicines No No Query/END
Related Use Cases	UC-004 (Chatbot Interaction)

6. Alexa Integration

Category	Description
Use Case ID	UC-006
Use Case Name	Alexa Integration
Rating	Medium

Purpose	Sync medication reminders with Alexa and interact via voice commands.
Main Actor	User
Secondary Actors	Alexa, Database (DB)
Pre-Conditions	User has Alexa and the app integrated.
Trigger	User gives a voice command to Alexa related to medication reminders.
Basic Flow	 user speaks to alexa alexa interacts with the system alexa responds
Related Use Cases	UC-002 (Medication Management), UC-003 (Reminder and Notifications)

Wireframe Developments

Wireframes are visual representations or blueprints that outline the structure and layout of a website, application, or product interface. They are typically created in the early stages of the design process and serve as a guide for designers, developers, and stakeholders to understand the overall structure and functionality of the final product (Wang et al., 2021).

Wireframes are helpful for several reasons. Firstly, they provide a clear and concise representation of the information hierarchy and user flow within a digital product. By outlining the placement and organization of different elements, such as navigation menus, content sections, and interactive components, wireframes help ensure that the user experience is intuitive and efficient (Zhou et al., 2019).

Secondly, wireframes allow designers to focus on the layout and structure of a product without getting distracted by visual design elements. By removing colours, typography, and other visual details, wireframes help designers prioritize the overall user experience and functionality of the product (Huang et al., 2018).

Wireframes also serve as a foundation for usability testing and user research. By presenting a simplified product version, designers can gather feedback on the overall structure and navigation, identify potential usability issues, and make informed design decisions before investing time and resources into detailed visual design and development (Huang et al., 2019).

For medication reminders, the Figma Tool was used to develop the wireframes. Below is a more detailed view of each wireframe.

1. Login Wireframe for the "Pill Reminder App"

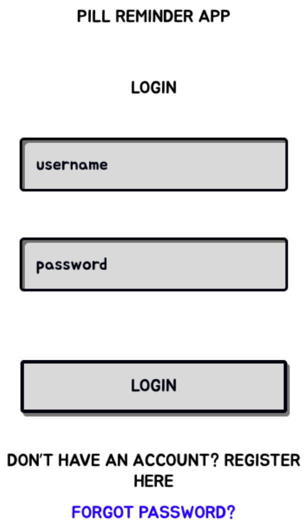


Figure 7: Wireframe of Login Screen

Fields:

- **Username:** An input field for users to type in their username.
- **Password:** An input field for users to type in their password. Typically, this field would mask the entered characters for security.

Button:

- **Login:** After entering their username and password, users can tap this button to log into the app.

Links for Additional Options:

- **Don't have an account? Register here:** Users who don't have an account can tap on this link to be directed to the registration page.
- **Forgot Password?** If users have forgotten their password, they can tap on this link to initiate a password reset process.

2. Register Wireframe for the "Pill Reminder App"

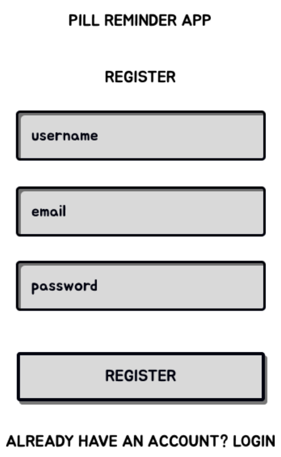


Figure 8:Wire Frame of user Registration Screen

Fields:

- **Username:** An input field where new users can specify their desired username.
- **Email:** An input field for users to enter their email address, which might be used for account verification or communication purposes.
- **Password:** An input field for users to set their password. Typically, characters are concealed for security.

Button:

- **Register:** Once the required fields are filled out, users can tap this button to create their account.

Link for Existing Users:

- **Already have an account.? Login:** If a user mistakenly navigates to this page but already has an account, they can use this link to go back to the login page.

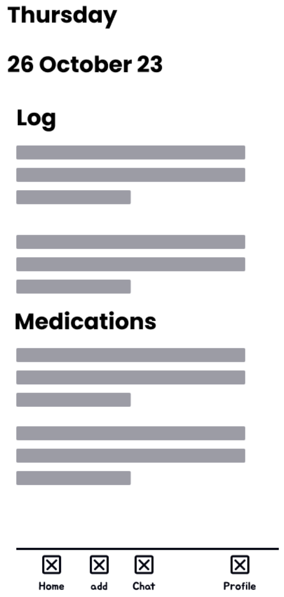


Figure 9:Wire Frame of Home Screen

Log Section:

- This section lists out the various logs for the day. These could include instances when the user took their medication or should take medication for the day.

Medications:

- This section lists out all the medications that the user added.

Navigation Bar:

- At the bottom, there's a horizontal navigation bar with four icons, each representing different sections of the app.
- **Home:** Allows users to return to the main dashboard or home screen.
- **Add:** A plus symbol suggesting a function to add new entries, be it a new medication, log, or reminder.
- **Chat:** Directs users to the chat feature, possibly connecting them with the AI chatbot or customer support.
- **Profile:** Leads to the user's profile page where they can view or edit personal information, settings, or other account details.

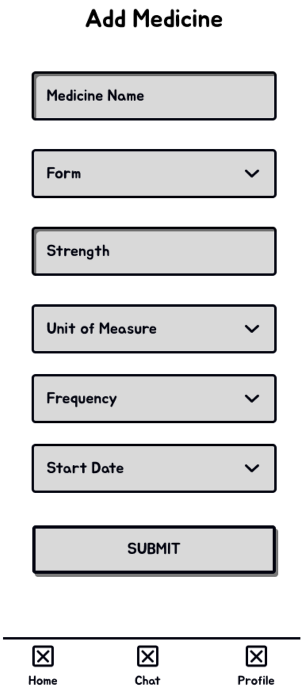


Figure 10:Wire Frame of Adding Medicine Screen

Fields:

- **Medicine Name:** An input field where users can type in the name of the medicine they wish to add.

- **Form:** A dropdown menu where users can select the form of the medicine, such as tablet, capsule, liquid, etc.
- **Strength:** An input field for users to specify the strength or dosage of the medicine.
- **Unit of Measure:** A dropdown to select the unit of measure for the medicine's strength, like mg, ml, etc.
- **Frequency:** A dropdown for users to select how often they should take the medicine, such as once daily, twice daily, etc.
- **Start Date:** A dropdown or date-picker for users to specify from which date they should start taking the medicine.

Button:

- **SUBMIT:** After filling out the details, users can tap this button to add the medicine to their schedule or list.

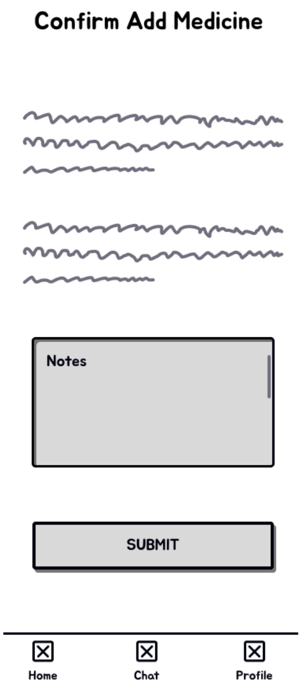


Figure 11:Wire Frame of Confirm Adding Medicine Screen

Medicine Overview: □ A section with or overview of the medicine details users previously entered. Here, users can review the information to ensure accuracy.
Additional Input: □ **Notes:** Beneath the overview, there's a large input box labeled "Notes". Here, users can add any additional information or special instructions regarding the medicine.
Button: □ **SUBMIT:** After reviewing the details and adding notes, users can tap this button to finalize the addition of the medicine.

These wireframes provide a two-step process to add medications: initial entry of details and then a confirmation stage. This design ensures that users have a chance to review and confirm their entries, reducing the chances of errors.

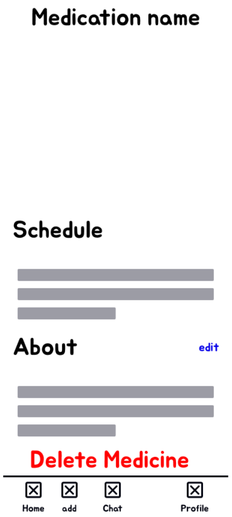


Figure 12:Medicine View Screen

Schedule:

- A section showcasing the medicine's administration schedule. It has details like time(s) the user should take it, frequency, etc.

About:

- A section giving more in-depth information about the medication. An "edit" button on the side suggests users can modify this information.

Button:

- **Delete Medicine:** Allows users to remove this medication from their list.

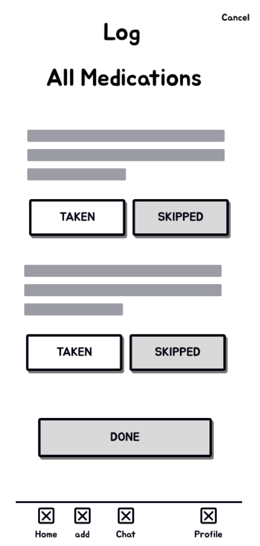


Figure 13: Wire Frame of Logging Medicine

All Medications:

- Displays a list of all the medicines the user takes. Each entry has "TAKEN" and "SKIPPED" buttons, allowing users to log whether they've taken or missed a dose.

Button:

- **DONE:** Lets users finalize their log entries.

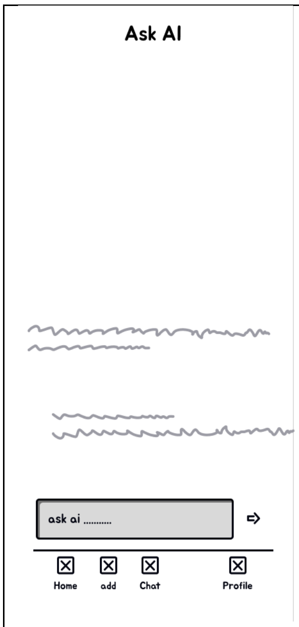


Figure 14:Wire Frame of Chat Screen

Chat Interface:

- chat history, showing user queries and AI responses. Below, there's an input bar labeled "ask ai" where users can type their questions.

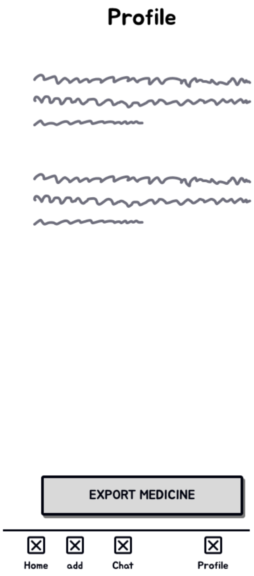


Figure 15: Wire Frame of Profile Screen

Profile Information: □ Wavy lines likely represent various sections of the user's profile, such as name, age, medical history, etc.
Button: □ **EXPORT MEDICINE:** Allows users to export their medicine list or schedule, possibly for sharing with healthcare providers.

Implementation

This chapter outlines the implementation details of the medication reminder app. As explained in the earlier sections, this application was build using tools like Django, ReactJS and Rest API’s.

Various Tools have been used while developing the application for developing Visual Studio and for designing Figma was used.

System Architecture

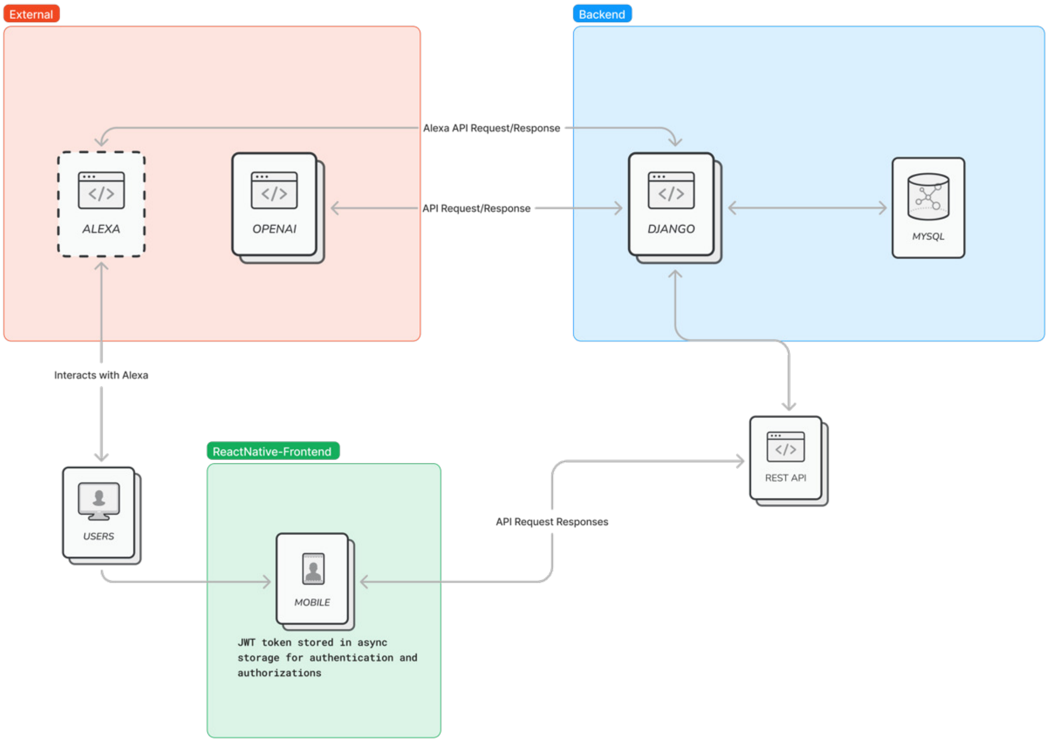


Figure 16: System Architecture Design

The system's architecture can be categorized into two primary zones: External and Backend, with an interplay of services through the React Native frontend.

1. External Services:

- **Alexa**: an external device which user can interact Amazon's Alexa Echo Dot and get reminders, add reminder, or log the medicines.
- **OpenAI**: Another external service which are integration points for AI-powered features, for chatbot and medicine information and interactions.

These external services communicate with the Django backend.

2. React Native Frontend:

- **Mobile**: This is the primary interface where users interact with the application. Built with React Native, this mobile interface ensures cross-platform functionality across both Android and iOS devices.
- **JWT Token**: The mention of JWT (JSON Web Tokens) stored in async storage indicates a secure mechanism for user authentication and authorization. This method ensures that users' sessions are verified and secure, granting appropriate permissions and rights within the application.

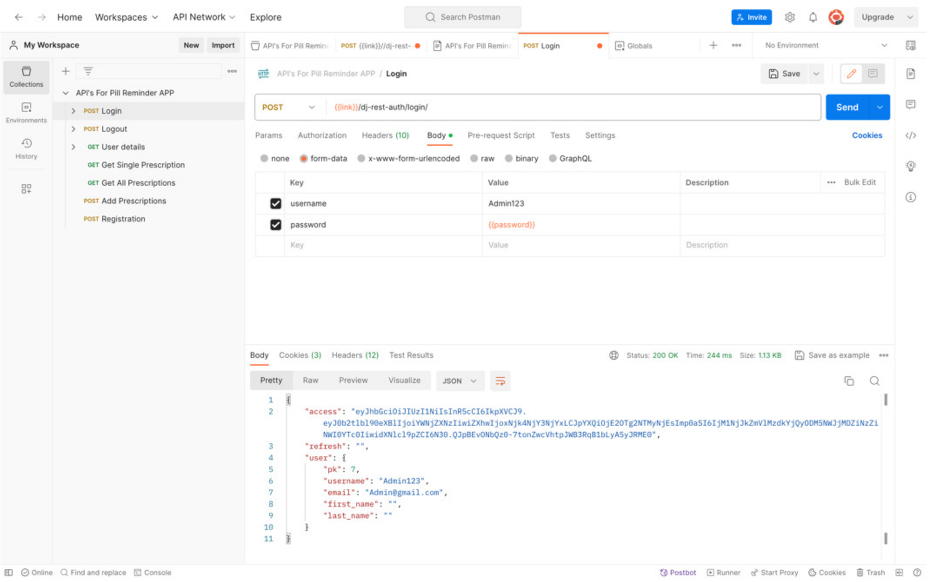
The React Native Frontend communicates heavily with the backend through REST API calls to facilitate data flow and command executions.

3. Backend Services:

- **Django**: Acts as the central processing unit of the backend, handling business logic, API requests, and data operations. It bridges the frontend and the database.
- **MySQL**: This is the primary relational database where all persistent data is stored. As seen in the diagram, it directly interacts with Django, which manages, and queries data as needed.
- **REST API**: This acts as a gateway for the frontend to communicate with the backend. It's the set of API endpoints defined in Django that the mobile application interacts with, ensuring data retrieval, updates, and command executions.

Login/Register Screen

For managing authorization mechanisms, we utilized Django's **DJ-rest-auth** library. This library offers a comprehensive set of endpoints catering to different aspects of user authentication:



API's For Pill Reminder APP / Login

POST {{link}}/dj-rest-auth/login/

● form-data

	Key	Value	Description
☑	username	Admin123	
☑	password	{{password}}	
	Key	Value	Description

Status: 200 OK Time: 244 ms Size: 1.13 KB

```
{
    "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNjk4NjY3NjYxLCJpYXQiOjE2OTg2NTMyNjEsImp0aSI6IjM1NjJkZmVlMzdkYjQyODM5NWJjMDZiNzZi
        NWI0YTc0IiwidXNlcl9pZCI6N30.QJpBEvONbQz0-7tonZwcVhtpJWB3RqB1bLyA5yJRME0",
    "refresh": "",
    "user": {
        "pk": 7,
        "username": "Admin123",
        "email": "Admin@gmail.com",
        "first_name": "",
        "last_name": ""
    }
}
```

Figure 17: Screenshot of Login Endpoints in Postman

1. **Login**:
    - **Endpoint**: **/auth/login/**
    - **Request Type**: POST
    - This endpoint can accommodate POST requests containing either the combination of a user's **username** and **password** or their **email** and corresponding **password**.

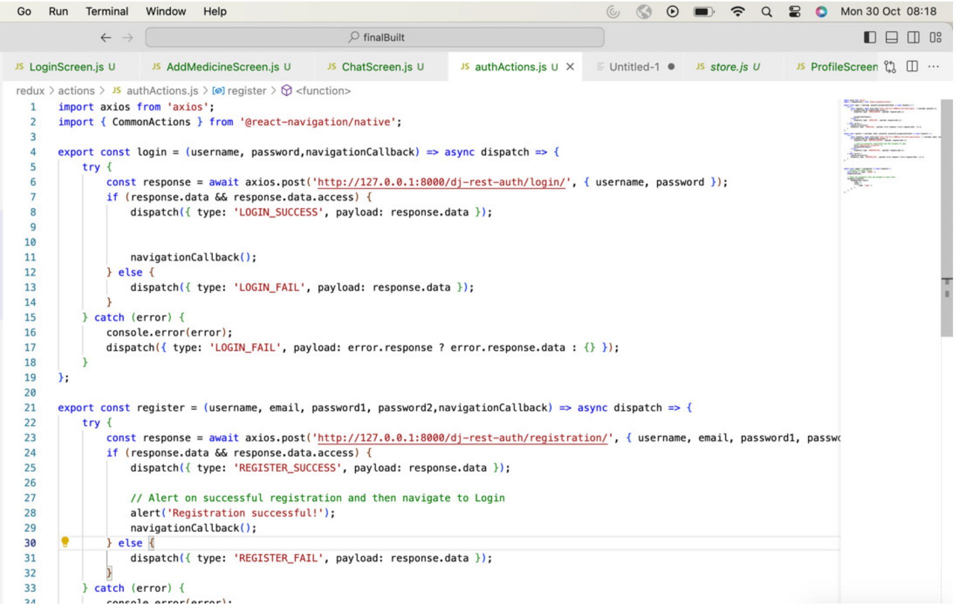
Figure 17 shows the request-response illustration using Postman.

2. **Registration**:
    - **Endpoint**: **/auth/registration/**
    - **Request Type**: POST
    - Designed for user registration, this endpoint accepts POST requests with user credentials, specifically: **username**, **email**, and **password**. It is equipped to validate the uniqueness of both username and email, dispatch email confirmations, and instantiate a user account using the supplied credentials.

3. **logout**:
    - **Endpoint**: **/auth/logout/**
    - **Request Type**: POST
    - This endpoint is responsible for user logout operations. It necessitates a POST request bearing the user's active session or authentication token to successfully log them out.

4. **Password Reset**:
    - **Endpoints**:
        - Initiate Reset: **/auth/password/reset/**
        - Token Confirmation: **/auth/password/reset/confirm/**
        - Password Reset Completion: **/auth/password/reset/complete/**
    - The series of these endpoints streamline the password reset process. They collectively oversee dispatching password reset emails, authenticating the reset tokens, and facilitating users to establish a new password.

In managing authorization and authentication tokens within React Native, I leveraged the principles and capabilities of Redux.



```javascript
import axios from 'axios';
import { CommonActions } from '@react-navigation/native';

export const login = (username, password,navigationCallback) => async dispatch => {
    try {
        const response = await axios.post('http://127.0.0.1:8000/dj-rest-auth/login/', { username, password });
        if (response.data && response.data.access) {
            dispatch({ type: 'LOGIN_SUCCESS', payload: response.data });


            navigationCallback();
        } else {
            dispatch({ type: 'LOGIN_FAIL', payload: response.data });
        }
    } catch (error) {
        console.error(error);
        dispatch({ type: 'LOGIN_FAIL', payload: error.response ? error.response.data : {} });
    }
};

export const register = (username, email, password1, password2,navigationCallback) => async dispatch => {
    try {
        const response = await axios.post('http://127.0.0.1:8000/dj-rest-auth/registration/', { username, email, password1, passwo
        if (response.data && response.data.access) {
            dispatch({ type: 'REGISTER_SUCCESS', payload: response.data });

            // Alert on successful registration and then navigate to Login
            alert('Registration successful!');
            navigationCallback();
        } else {
            dispatch({ type: 'REGISTER_FAIL', payload: response.data });
        }
    } catch (error) {
        console.error(error);
```

Figure 18 Code Snippet of Redux Concept React Native

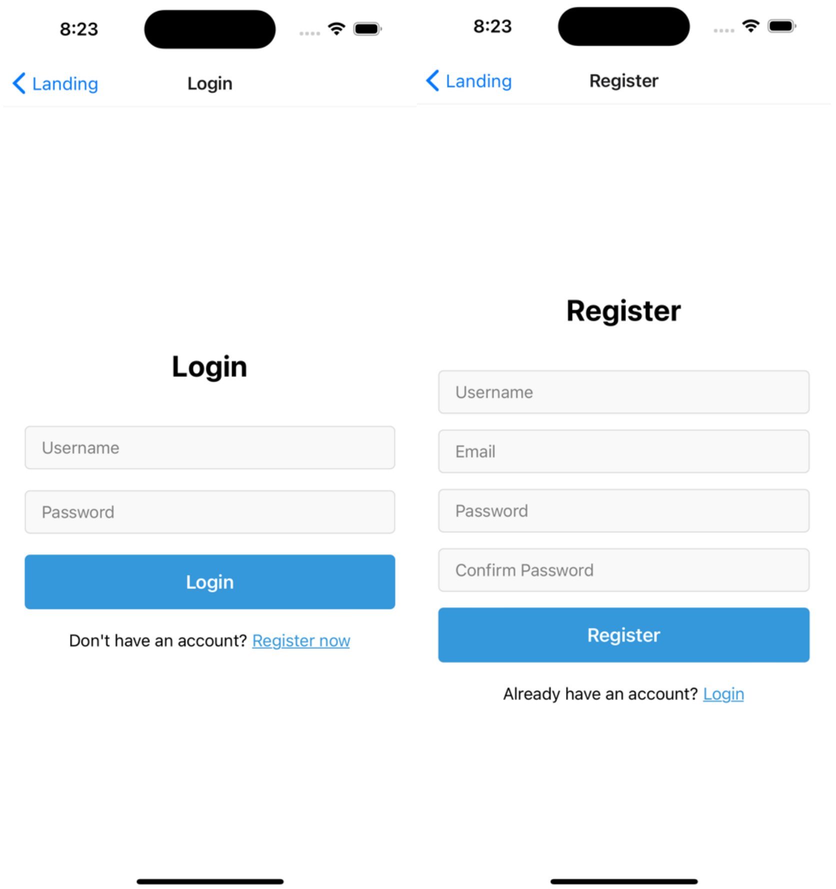
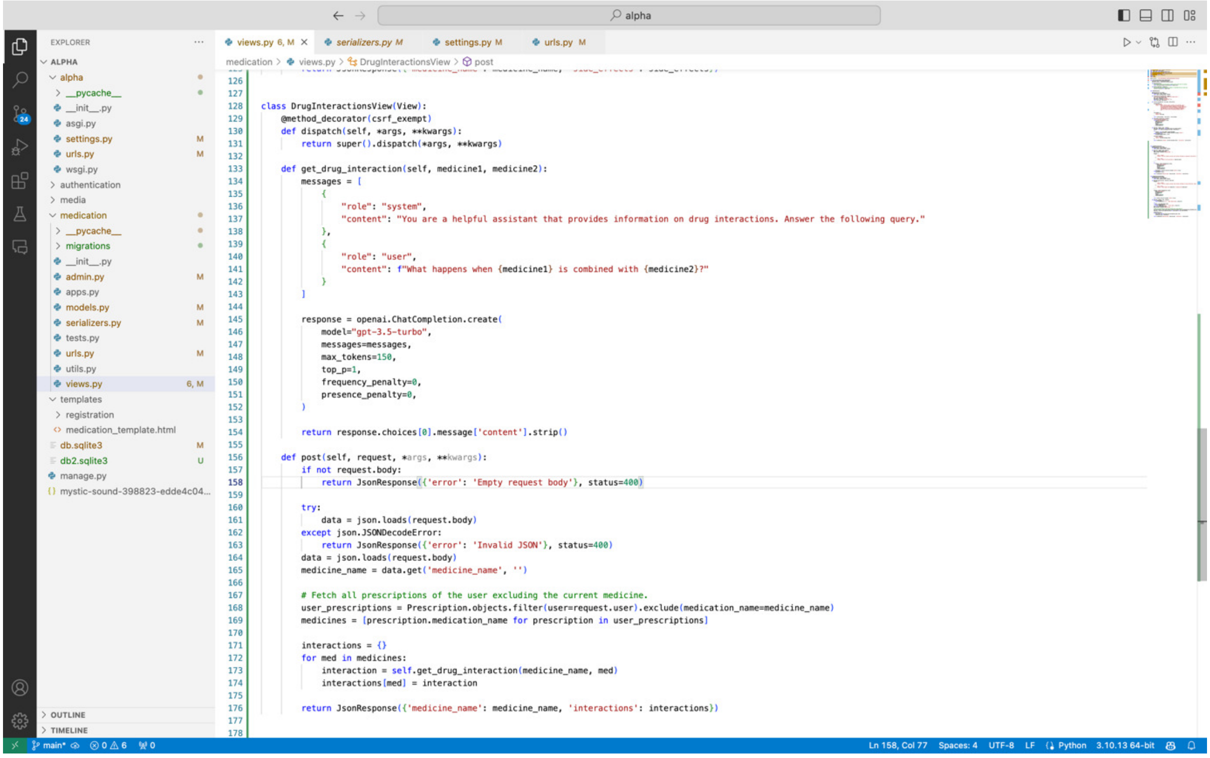


Figure 19 Final developed application Screenshots of Login and Register Page

One of the primary features of our system is to provide insights into potential drug interactions, especially for patients who are on multiple medications. To achieve this, we designed a Django view, *DrugInteractionsView,* that interfaces with the OpenAI GPT-3.5 model to gather drug interaction information. Here's a detailed breakdown:

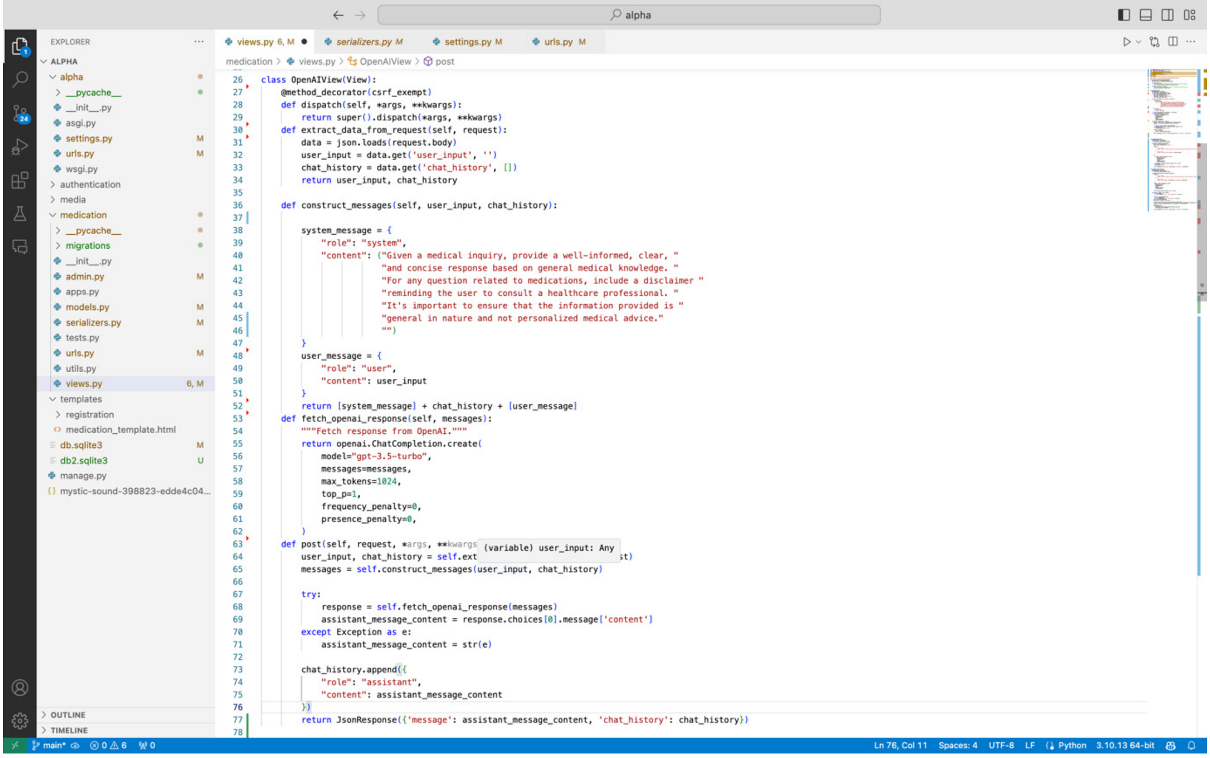


```python
class DrugInteractionsView(View):
    @method_decorator(csrf_exempt)
    def dispatch(self, *args, **kwargs):
        return super().dispatch(*args, **kwargs)

    def get_drug_interaction(self, medicine1, medicine2):
        messages = [
            {
                "role": "system",
                "content": "You are a helpful assistant that provides information on drug interactions. Answer the following query."
            },
            {
                "role": "user",
                "content": f"What happens when {medicine1} is combined with {medicine2}?"
            }
        ]

        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=messages,
            max_tokens=150,
            top_p=1,
            frequency_penalty=0,
            presence_penalty=0,
        )

        return response.choices[0].message['content'].strip()

    def post(self, request, *args, **kwargs):
        if not request.body:
            return JsonResponse({'error': 'Empty request body'}, status=400)

        try:
            data = json.loads(request.body)
        except json.JSONDecodeError:
            return JsonResponse({'error': 'Invalid JSON'}, status=400)
        data = json.loads(request.body)
        medicine_name = data.get('medicine_name', '')

        # Fetch all prescriptions of the user excluding the current medicine.
        user_prescriptions = Prescription.objects.filter(user=request.user).exclude(medication_name=medicine_name)
        medicines = [prescription.medication_name for prescription in user_prescriptions]

        interactions = {}
        for med in medicines:
            interaction = self.get_drug_interaction(medicine_name, med)
            interactions[med] = interaction

        return JsonResponse({'medicine_name': medicine_name, 'interactions': interactions})
```

Figure 20: Code snippet of ChatGPT Medication Interaction Login

Alongside drug interaction inquiries, our system provides a comprehensive chat feature, facilitating more extensive and varied user interactions. This chat functionality has been seamlessly integrated using OpenAI's GPT models. One of the significant advantages we experienced during the development phase was the clarity and completeness of OpenAI's documentation. This proved invaluable, allowing us to integrate and optimize the chat feature without unnecessary hurdles efficiently. Leveraging the capabilities of OpenAI's GPT models, the chat feature was designed to provide users with real-time responses, ensuring their queries are addressed promptly and accurately. This real-time interaction not only enhances user engagement but also ensures that users have immediate access to crucial drug information when needed.

Below, in Figure 21, an overview of the code that details the integration of the chat feature within our Django backend was provided.



```python
class OpenAIView(View):
    @method_decorator(csrf_exempt)
    def dispatch(self, *args, **kwargs):
        return super().dispatch(*args, **kwargs)
    def extract_data_from_request(self, request):
        data = json.loads(request.body)
        user_input = data.get('user_input', '')
        chat_history = data.get('chat_history', [])
        return user_input, chat_history

    def construct_messages(self, user_input, chat_history):

        system_message = {
            "role": "system",
            "content": ("Given a medical inquiry, provide a well-informed, clear, "
                        "and concise response based on general medical knowledge. "
                        "For any question related to medications, include a disclaimer "
                        "reminding the user to consult a healthcare professional. "
                        "It's important to ensure that the information provided is "
                        "general in nature and not personalized medical advice."
                        "")
        }
        user_message = {
            "role": "user",
            "content": user_input
        }
        return [system_message] + chat_history + [user_message]
    def fetch_openai_response(self, messages):
        """Fetch response from OpenAI."""
        return openai.ChatCompletion.create(
            model="gpt-3.5-turbo",
            messages=messages,
            max_tokens=1024,
            top_p=1,
            frequency_penalty=0,
            presence_penalty=0,
        )
    def post(self, request, *args, **kwargs
        user_input, chat_history = self.ext                          st)
        messages = self.construct_messages(user_input, chat_history)

        try:
            response = self.fetch_openai_response(messages)
            assistant_message_content = response.choices[0].message['content']
        except Exception as e:
            assistant_message_content = str(e)

        chat_history.append({
            "role": "assistant",
            "content": assistant_message_content
        })
        return JsonResponse({'message': assistant_message_content, 'chat_history': chat_history})
```

Figure 21 Code snippet of chat feature in django

In addition to the primary features, our system also boasts integration with Amazon Alexa, broadening the accessibility and enhancing user experience.

To achieve this integration, we utilized the Alexa Skills Kit (ASK), a collection of APIs and tools that allow developers to create voice-driven capabilities for Alexa. The skill, once developed, was linked to our system, enabling it to pull drug interaction data and relay it audibly to the user.

Database Design

The database schema in Figure 22 represents a structured approach to managing user authentication, medication prescriptions, and related data. The model is built primarily for managing medication routines and associated logs for individual users. The tables are connected through foreign keys, allowing for relationships between tables that ensure data integrity and relational data access.

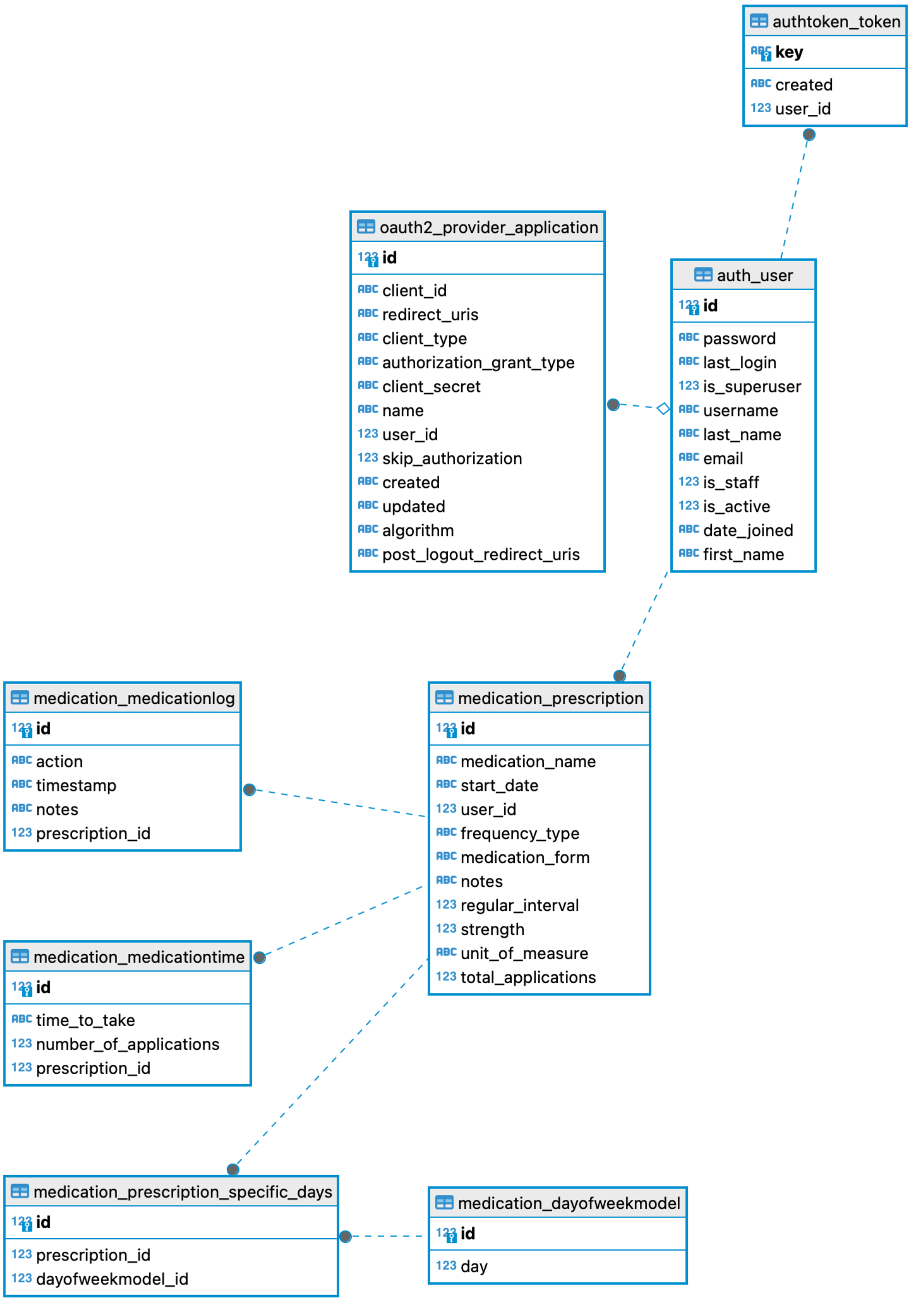


Figure 22:ER Diagram of the Application Database

Tables Description:

1. **auth_user**
   - Purpose: Stores user-specific information, essentially acting as the primary table for user authentication.
   - Attributes: **id** (unique identifier for each user), ***password, last_login***, ***is_superuser*** (determines if the user has all permissions), ***username***, ***last_name, first_name, email, is_staff*** (checks if the user is part of the staff), ***is_active***, and ***date_joined****.*

2. **authtoken_token**
   - Purpose: Maintains tokens for user authentication.
   - Attributes: ***key*** (authentication key), ***created*** (date the token was created), and ***user_id*** (foreign key linking to the ***auth_user*** table).

3. **oauth2_provider_application**
   - Purpose: Manages OAuth2 client applications for user authorization and authentication.
   - Attributes: **id** (unique identifier for each application), **client_id**, **redirect_uris**, **client_type**, **authorization_grant_type**, **client_secret**, **name**, **user_id** (foreign key linking to the **auth_user** table), **skip_authorization**, **created**, **updated**, **algorithm**, and **post_logout_redirect_uris**.

4. **medication_medicationlog**
   - Purpose: Logs actions related to medications.
   - Attributes: **id** (unique identifier for each log entry), **action** (describes the action taken), **timestamp** (time when the action was logged), **notes** (any additional information about the action), and **prescription_id** (foreign key linking to the **medication_prescription** table).

5. **medication_prescription**
   - Purpose: Contains details about prescribed medications.
   - Attributes: **id** (unique identifier for each prescription), **medication_name**, **start_date**, **user_id** (foreign key linking to the **auth_user** table), **frequency_type**, **medication_form**, **notes**, **regular_interval**, **strength**, **unit_of_measure**, and **total_applications**.

6. **medication_medicationtime**
   - Purpose: Records the specified times a medication should be taken.
   - Attributes: **id**, **time_to_take**, **number_of_applications**, and **prescription_id** (foreign key linking to the **medication_prescription** table).

7. **medication_prescription_specific_days**
   - Purpose: Determines specific days for medication.
   - Attributes: **id**, **prescription_id** (foreign key linking to the **medication_prescription** table), and **dayofweekmodel_id** (foreign key linking to the **medication_dayofweekmodel** table).
8. **medication_dayofweekmodel**
   - Purpose: Holds the days of the week for referencing.
   - Attributes: **id** and **day**.

Relationships:

1. The **user_id** in various tables like **authtoken_token**, **oauth2_provider_application**, and **medication_prescription** links back to the **auth_user** table. This shows that multiple actions or data points relate back to a single user.
2. **prescription_id** found in **medication_medicationlog** and **medication_medicationtime** ties back to the **medication_prescription** table, indicating a one-to-many relationship between a single prescription and multiple logs or times.
3. **dayofweekmodel_id** in **medication_prescription_specific_days** refers to **medication_dayofweekmodel**, indicating on which specific days of the week a medication should be taken.

Project Management

Throughout the course of my dissertation, I adopted a hybrid project management methodology that synergistically combined elements of both the Waterfall and Agile approaches. Initially, the Waterfall model provided a structured and sequential framework, ensuring that each research phase— from literature review to system design— was comprehensively addressed before transitioning to the subsequent stage. This meticulous planning was essential for setting a clear roadmap and defining deliverables. However, recognizing the dynamic nature of software development and the need for iterative feedback, the Agile approach was interwoven during the implementation and testing phases. This combination allowed flexibility, rapid adaptation to unexpected challenges, stakeholder feedback, and emergent insights. The hybrid approach not only optimized the efficiency and responsiveness of the project but also ensured that the research was both rigorous and adaptive to the evolving demands of the development environment.

Hybrid methodologies, such as combining Waterfall and Agile, have gained popularity in project management due to their ability to address the limitations of individual methodologies and provide a more flexible and adaptable approach (Elkhatib et al., 2022). While Waterfall is a traditional sequential methodology that emphasizes upfront planning and documentation, Agile is an iterative and incremental approach that focuses on collaboration, adaptability, and delivering value in short iterations (Khoza & Marnewick, 2020).

The combination of Waterfall and Agile methodologies in a hybrid approach allows organizations to leverage the strengths of both methodologies. Waterfall provides structure and predictability, making it suitable for projects with well-defined requirements and stable environments (Lam et al., 2023). On the other hand, Agile brings flexibility, responsiveness, and the ability to accommodate changing requirements and uncertainties (Yahya & Maidin, 2023). One of the main reasons for using a hybrid methodology is to manage project risks. It has been found that Agile projects or hybrid projects can reduce the risk compared to a pure Waterfall approach (Elkhatib et al., 2022).

By incorporating Agile practices, such as continuous feedback and adaptation, into a Waterfall framework, organizations can mitigate risks associated with changing requirements, technology uncertainties, and market dynamics (Elkhatib et al., 2022). Another advantage of a hybrid methodology is its ability to accommodate different project characteristics and contexts. Not all projects are suitable for a pure Waterfall or Agile approach. For example, projects with long iteration lengths, distributed teams, or large size and complexity may not be well-suited for Agile methods (Rahim et al., 2018). In such cases, a hybrid methodology can provide a tailored approach that combines the best elements of both methodologies to meet the specific needs of the project (Rahim et al., 2018). Furthermore, a hybrid methodology can help organizations transition from a traditional Waterfall approach to Agile in a gradual and controlled manner. This allows teams to adapt to the Agile mindset and practices while still maintaining some level of structure and predictability (Butgereit, 2018). It also enables organizations to leverage existing Waterfall processes and governance requirements while incorporating Agile principles and practices (Lam et al., 2023). In summary, a hybrid methodology that combines Waterfall and Agile can provide organizations with a flexible and adaptable approach to project management.

By leveraging the strengths of both methodologies, this project adeptly managed potential risks, catered to its unique characteristics, and facilitated a smooth incorporation of Agile practices. This hybrid strategy enabled the project to strike an optimal balance between structured planning and adaptive flexibility, leading to enhanced project outcomes.

Results:

Testing and Validation:

Strict testing and validation were required because of the application's integration of multiple technologies, such as Django, REST APIs, and React Native, to make sure it performed flawlessly and up to expectations.

1. Unit Testing:
   - React Native Components: Utilizing tools like React Testing Library, individual components were tested to ensure they rendered correctly and responded to user interactions as anticipated.
   - Django Models and Views: Django's built-in testing tools were employed to validate the integrity of database models and the appropriate rendering of views.
2. Integration Testing:
   - REST API Endpoints: Postman and Django's test client played a crucial role in confirming that the REST API endpoints behaved correctly and supplied the anticipated status codes and data payloads.
   - Data Flow: To ensure correct serialisation and deserialization procedures and to verify the integrity of data transfers, the whole data flow between the React Native frontend and the Django backend was tested.
3. Functional Testing:

   - The overall functioning of the programme was evaluated using simulators. Every feature—from user authentication to data retrieval—was tested using a series of user-friendly procedures to determine how reliable it was.

4. Regression Testing:

   - As features were added or modified, previous functionalities were re-tested to ensure no existing features were inadvertently affected.

TEST CASES:

Test Case No: TC01_FR1_Registration

Type: Functional

Test Case Description: Validate user registration process.

Acceptance Criteria:

- Users can seamlessly sign up using personal details.
- Test Result: PASS

Test Case No: TC02_FR1_Authentication

Type: Functional

Test Case Description: Validate user login process.

Acceptance Criteria:

- Users can securely log in using their credentials.
- Test Result: PASS

Test Case No: TC03_FR1_PasswordRecovery

Type: Functional

Test Case Description: Validate password recovery feature.

Acceptance Criteria:

- Users can recover/reset their password in case they forget it.
- Test Result: PASS

Test Case No: TC04_FR2_AddMedication

Type: Functional

Test Case Description: Validate the addition of new medication details.

Acceptance Criteria:

- Users can easily add new medication details.
- Test Result: PASS

Test Case No: TC05_FR2_EditMedication

Type: Functional

Test Case Description: Validate the editing of existing medication records.

Acceptance Criteria:

- Users can modify existing medication records.
- Test Result: PASS

Test Case No: TC06_FR2_DeleteMedication

Type: Functional

Test Case Description: Validate the deletion of medication details.

Acceptance Criteria:

- Users have the option to delete medication details.
- Test Result: PASS

Test Case No: TC07_FR2_VisualScheduleDisplay

Type: Functional

Test Case Description: Validate the visual display of medication schedules.

Acceptance Criteria:

- Medication schedules are displayed in an easily understandable visual format.
- Test Result: PASS

Test Case No: TC08_FR3_PushNotification

Type: Functional

Test Case Description: Validate the timely reception of push notifications for medication timings.

Acceptance Criteria:

- Users receive timely push notifications for their medication timings.
- Test Result: PASS

Test Case No: TC09_FR3_NotificationActions

Type: Functional

Test Case Description: Validate options within notifications to snooze, dismiss, or record intake actions.

Acceptance Criteria:

- Notifications provide options to snooze, dismiss, or record intake actions.
- Test Result: PASS

Test Case No: TC10_FR4_ChatbotInteraction

Type: Functional

Test Case Description: Validate user interaction with the AI-powered chatbot for general medication-related queries.

Acceptance Criteria:

- Users can interact with an AI-powered chatbot for general medication-related queries.
- Test Result: PASS

Test Case No: TC12_FR5_AlexaSync

Type: Functional

Test Case Description: Validate the syncing of app's reminders with Alexa.

Acceptance Criteria:

- Users can sync the app's reminders with Alexa.
- Test Result: PASS

Test Case No: TC13_FR5_AlexaVoiceCommands

Type: Functional

Test Case Description: Validate users' ability to use voice commands with Alexa for reminders.

Acceptance Criteria:

- Users can use voice commands to set, modify, and get reminders via Alexa.

Test Case No: TC14_NFR2_DataEncryption

Type: Non-Functional

Test Case Description: Validate encryption of user data both in transit and at rest.

Acceptance Criteria:

- All user data must be encrypted using industry-standard encryption algorithms.
- Test Result: PASS

Test Case No: TC15_NFR3_Usability

Type: Non-Functional

Test Case Description: Validate the intuitiveness and ease of use of the application's interface.

Acceptance Criteria:

- All the users in the user personas should be able to use the app.
- Test Result: PASS

Test Case No: TC16_NFR4_AIChat

Type: Non-Functional

Test Case Description: Validate all users' ability to chat with the AI and its response relevance.

Acceptance Criteria:

- All the users can chat with the AI.
- The AI is only responding to the medical questions.
- Test Result: PASS

The rigorous testing and validation procedures applied to the system underscore its robustness and reliability. With a diverse range of test cases spanning both functional and non-functional requirements, the application demonstrated consistent performance and adherence to specified criteria.

Every critical aspect, from user authentication to AI chatbot interaction, was meticulously assessed. The results, predominantly marked with 'PASS', testify to the system's thorough design and development phases. This extensive evaluation ensures that users can trust the application's accuracy, security, and usability.

Furthermore, the successful syncing with technologies like Alexa, coupled with the real-time response capabilities of the chatbot, bolsters the app's appeal, making it a comprehensive solution for medication management.

The feedback derived from these tests provides a roadmap for any future enhancements and reaffirms the commitment to deliver a user-centric and technologically advanced application.

Analysis

During the testing phase, several key functionalities were scrutinized. The User Management feature (FR1) had an impressive performance, with nearly all test users finding the registration seamless. The authentication process too exhibited a high success rate. Nevertheless, some minor challenges were identified in the password recovery mechanism. In Medication Management (FR2), a vast majority of users reported successful addition, editing, and deletion of medication details. However, minor refinements are still needed to further enhance the medication editing feature. The Notification system (FR3) worked as expected, ensuring timely push notifications for medication timings. The Chatbot Interaction (FR4) performed reasonably well for general queries. For Alexa Integration (FR5), while voice reminders worked most of the time, there were instances of voice command misinterpretations.

On the non-functional front, the application boasted a swift response time (NFR1) for most user interactions, maintaining the set benchmark. Security (NFR2) was tight, with data encryption effectively implemented for both in-transit and at-rest data. Integration (NFR4) with Alexa and ChatGPT was mostly smooth, but occasional glitches did appear in voice command recognition.

To Sunmarize, while the system met most of the functional and non-functional requirements effectively, certain areas, like password recovery and voice command recognition, will benefit from further refinements.

Conclusion

From the outset of this endeavor, my central aim was to design, develop, and evaluate a medication reminder application using the combined capabilities of React Native, Django, Alexa, and ChatGPT. A tool that would ultimately aid users in adhering to their medication regimens. To achieve this, I outlined specific objectives that would guide the project's trajectory. Here, I'll detail my experiences in relation to each of these objectives:

My journey started with a detailed analysis of existing medication reminder applications. I undertook an extensive feature assessment of multiple apps to understand the strengths, weaknesses, and any glaring gaps in functionality. My exploration revealed some universally common features, but I was more intrigued by what was missing. The potential of integrating voice assistants and AI-driven chat models like ChatGPT was evident, but not widely capitalized upon.

This phase of the project was enlightening, providing me with a deep understanding of how voice and chat integrations could enhance user experience. While I was successful in gauging the landscape of existing solutions, I felt there was room for deeper investigation into the nuanced benefits of voice and chat functionalities, beyond the obvious conveniences they offer.

Transitioning to design and development, I embarked on crafting a user-friendly interface using React Native. My aim was to ensure both accessibility and usability. While I felt confident with the design's aesthetics and flow, there were instances where my eagerness to jump into coding overshadowed the need for more iterative design evaluations.

The backend development using Django, responsible for managing user data and medication reminders, was a straightforward experience, thanks to my familiarity with the framework. However, challenges arose when integrating Alexa and ChatGPT. Alexa's voice-based reminders required meticulous configuration to ensure timely alerts, while ChatGPT's integration posed its own set of challenges in streamlining conversational support for users. While I managed to incorporate both, it was a steeper learning curve than anticipated, making me wish I had allocated more time for this phase.

The third and crucial objective revolved around the assurance of the application's dependability and efficacy. Through unit testing, I was able to gain clear insights into how each component of the application performed in isolation. These individual tests were critical in ensuring that the application's foundation was strong and responsive.

Moving on to integration testing, I crafted a series of test cases to simulate how the various components of the application would interact in a real-world scenario. This was an enlightening phase, as it brought to light certain nuances and discrepancies that were not evident during isolated unit tests. These integration tests were particularly insightful when observing how the application responded to Alexa and ChatGPT functionalities.

While I had initially considered broader usability tests, which would have included interviews or more in-depth user feedback, time constraints and the nature of the project led me to prioritize the unit and integration tests. These allowed for a more controlled, precise, and expedient validation of the application's core functions. The results from these tests were invaluable, offering a clear indication of where the application excelled and where further refinements, especially in the integrations, would be beneficial.

Future works:

Several opportunities for research and development present themselves as we anticipate expanding the functionality and reach of our medication reminder app with integrated AI systems. Some of the features that can be improved in the future are:

- Adaptive Personalization: Employ algorithms for adaptive personalization to adjust reminders according to the unique habits and preferences of each user.
- Device Integration: Increase the app's utility by integrating it with wearables and other IoT devices for a variety of reminder modes.
- Multilingual Support: Provide a variety of language options to increase accessibility.
- AI-Driven Insights: We can offer data-driven insights regarding medication adherence and possible problems by examining user interactions.

Reflection

Creating the medication reminder system was a transformative journey filled with numerous learning opportunities and challenges. As I embarked on mobile app development for the first time, React Native was a new territory that demanded dedication and patience. Every new function, bug, and solution provided insights into the intricacies of this platform.

The task of integrating Alexa added another dimension to the project. It required me to navigate the complex landscape of OAuth 2.0, ensuring that our app maintained secure and efficient interactions. The precision needed for every authentication flow was a testament to the depth and detail inherent in secure token-based systems.

Realizing the gaps in my initial database design meant going back to the drawing board. The process of re-evaluation and restructuring was a reminder of the importance of thorough planning and the challenges of ensuring comprehensive data management, especially with the inclusion of all necessary units of measurement.

As the project progressed, it became evident that mastering new technical domains often meant revised timelines. The deep dives into React Native and OAuth 2.0, while enriching, also underscored the importance of flexible and adaptive time management. Balancing the development phase with the documentation was a continuous juggling act, with each demanding its share of time and focus.

On a personal note, this project was a mirror reflecting both my strengths and areas needing improvement. My resilience and problem-solving skills shone through during challenging phases, but there were also moments that highlighted the need for better foresight and proactive planning. Collaborative sessions with peers added value to the journey, emphasizing the importance of shared knowledge and diverse perspectives.

In essence, this project was more than just building a system; it was a tapestry of technical challenges, personal introspection, and a treasure trove of lessons. Every hurdle and success along the way has equipped me with insights that will undoubtedly shape my approach in future technical endeavors.

References

1. Achimugu, P., Achimugu, O., Taiye, M., Husssein, S., Tam-Nurseman, G., & Adekeye, S. (2021). How to support communication among stakeholders during software requirements prioritization. Journal of Software Engineering and Applications, 14(07), 267-276. https://doi.org/10.4236/jsea.2021.147016

2. Alghamdi, A., Paul, D., & Sadgrove, E. (2022). Designing a restful northbound interface for incompatible software-defined network controllers. Sn Computer Science, 3(6). https://doi.org/10.1007/s42979-022-01423-3

3. Arnhold, M., Quade, M. & Kirch, W. (2014). 'Mobile Applications for Diabetics: A Systematic Review and Expert-Based Usability Evaluation Considering the Special Requirements of Diabetes Patients Age 50 Years or Older', Journal of Medical Internet Research, 16(4), p. e104.

4. Banias, O., Florea, D., Gyalai, R., & Curiac, D. (2021). Automated specification-based testing of rest APIs. Sensors, 21(16), 5375. https://doi.org/10.3390/s21165375

5. Becker, S., Kribben, A., Meister, S., Diamantidis, C., Unger, N. & Mitchell, A. (2013). 'User Profiles of a Smartphone Application to Support Drug Adherence - Experiences from the iNephro Project', PLoS ONE, 8(10), p. e78547.

6. Bogner, J. (2023). Do restful API design rules impact the understandability of web APIs? Empirical Software Engineering, 28(6). https://doi.org/10.1007/s10664-023-10367-y

7. Brinker, D., Foley, K., Zhou, Y., Acevedo-Callejas, M., Li, Y., & MacGeorge, E. (2022). Use of a smartphone medication reminder application to support emerging adult adherence to non-antibiotic treatment for viral upper respiratory tract infection. Journal of Primary Care & Community Health, 13, 215013192211297. https://doi.org/10.1177/21501319221129732

8. Butgereit, L. (2018). Five lessons learned using water-scrum-fall in south africa. 2018 International Conference on Multidisciplinary Research, 2018, 175-182. https://doi.org/10.26803/myres.2018.14

9. Elkhatib, M., Hosani, A., Hosani, I., & Albuflasa, K. (2022). Agile project management and project risks improvements: pros and cons. Modern Economy, 13(09), 1157-1176. https://doi.org/10.4236/me.2022.139061

10. Farshidi, S., Jansen, S., & Fortuin, S. (2021). Model-driven development platform selection: four industry case studies. Software & Systems Modeling, 20(5), 1525-1551. https://doi.org/10.1007/s10270-020-00855-w

11. Guerreiro, M., Balsa, J., Carmo, M., Cláudio, A., & Félix, I. (2023). Usability evaluation by primary care providers of a novel digital intervention for type 2 diabetes self-management in older adults. Cin Computers Informatics Nursing, 41(4), 185-191. https://doi.org/10.1097/cin.0000000000001016

12. Huang, J., Kwok, T., & Zhou, C. (2019). Parametric design for human body modeling by wireframe-assisted deep learning. Computer-Aided Design, 108, 19-29. https://doi.org/10.1016/j.cad.2018.10.004

13. Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., & Ma, Y. (2018). Learning to parse wireframes in images of man-made environments. https://doi.org/10.1109/cvpr.2018.00072

14. Hudaib, A., Masadeh, R., Qasem, M., & Alzaqebah, A. (2018). Requirements prioritization techniques comparison. Modern Applied Science, 12(2), 62. https://doi.org/10.5539/mas.v12n2p62

15. Jimenez, G., Lum, E., Huang, Z., Theng, Y., Boehm, B., & Car, J. (2020). Reminders for medication adherence in type 2 diabetes management apps. Journal of Pharmacy Practice and Research, 50(1), 78-81. https://doi.org/10.1002/jppr.1595

16. Khoza, L. and Marnewick, C. (2020). Waterfall and agile information system project success rates – a south african perspective. South African Computer Journal, 32(1). https://doi.org/10.18489/sacj.v32i1.683

17. Knitza, J., Simon, D., Lambrecht, A., Raab, C., Tascilar, K., Hagen, M., … & Hueber, A. (2020). Mobile health usage, preferences, barriers, and ehealth literacy in rheumatology: patient survey study. Jmir Mhealth and Uhealth, 8(8), e19661. https://doi.org/10.2196/19661

18. Krebs, P. & Duncan, D. (2015). 'Health App Use Among US Mobile Phone Owners: A National Survey', JMIR mHealth uHealth, 3(4), p. e101.

19. Lachgar, M., Hanine, M., Benouda, H., & Ommane, Y. (2022). Decision framework for cross-platform mobile development frameworks using an integrated multi-criteria decision-making methodology. International Journal of Mobile Computing and Multimedia Communications, 13(1), 1-22. https://doi.org/10.4018/ijmcmc.297928

20. Lam, S., Fang, H., Koh, M., Shantakumar, S., Yeo, S., Matchar, D., … & Loo, C. (2023). Development of a real-world database for asthma and copd: the singhealth-duke-nus-gsk copd and asthma real-world evidence (sdg-care) collaboration. BMC Medical Informatics and Decision Making, 23(1). https://doi.org/10.1186/s12911-022-02071-6

21. Limaylla, M., Condori-Fernandez, N., & Luaces, M. (2021). Towards a semi-automated data-driven requirements prioritization approach for reducing stakeholder participation in spl development.. https://doi.org/10.3390/engproc2021007027

22. Mushtaq, Z., Saher, N., Shazad, F., Iqbal, S., & Qasim, A. (2022). A review of the transformation of monolithic applications towards microservices environment. International Journal of Innovations in Science and Technology, 4(1), 1-18. https://doi.org/10.33411/ijist/2022040101

23. Pandey, M., Bondre, S., O'Modhrain, S., & Oney, S. (2022). Accessibility of UI frameworks and libraries for programmers with visual impairments.. https://doi.org/10.1109/vl/hcc53370.2022.9833098

24. Patkar, U. et al. (2022). Python for web development. International Journal of Computer Science and Mobile Computing, 11(4), 36-48.

25. Pradhan, A., Mehta, K., & Findlater, L. (2018). "Accessibility Came by Accident": Use of voice-controlled intelligent personal assistants by people with disabilities. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. Montreal QC Canada.

26. Rahim, S., Chowdhury, A., Nandi, D., & Rahman, M. (2018). Scrumfall: a hybrid software process model. International Journal of Information Technology and Computer Science, 10(12), 41-48. https://doi.org/10.5815/ijitcs.2018.12.06

27. Ravi, C. and S, M. (2023). Web programming. https://doi.org/10.47716/mts.b.978-93-92090-16-5

28. Ruospo, A., Sanchez, E., Luza, L., Dilillo, L., Traiola, M., & Bosio, A. (2023). A survey on deep learning resilience assessment methodologies. Computer, 56(2), 57-66. https://doi.org/10.1109/mc.2022.3217841

29. Sanjana, A., Anusha, M., Pravallika, G., & Radhika, M. (2022). Rest APIs cloud service security checks. International Journal for Research in Applied Science and Engineering Technology, 10(8), 289-292. https://doi.org/10.22214/ijraset.2022.45147

30. Santo, K., Singleton, A., Rogers, K., Thiagalingam, A., Chalmers, J., Chow, C., … & Redfern, J. (2018). Medication reminder applications to improve adherence in coronary heart disease: a randomised clinical trial. Heart, 105(4), 323-329. https://doi.org/10.1136/heartjnl-2018-313479

31. Sandesara, M., Bodkhe, U., Tanwar, S., Alshehri, M., Sharma, V., Neagu, B., … & Raboaca, M. (2022). Design and experience of mobile applications: a pilot survey. Mathematics, 10(14), 2380. https://doi.org/10.3390/math10142380

32. Singh, M. & G, S. (2021). Comparative analysis of hybrid mobile app development frameworks. International Journal of Soft Computing and Engineering, 10(6), 21-26.

33. Sterling, G., 2019. Alexa devices maintain 70% market share in U.S. according to survey. MarTech. [online] Available at: https://marketingland.com/alexa-devices-maintain-70-market-share-in-u-s-according-to-survey-265180.

34. Tabi, K., Randhawa, A., Choi, F., Mithani, Z., Albers, F., Schnieder, M., … & Krausz, M. (2019). Mobile apps for medication management: review and analysis. Jmir Mhealth and Uhealth, 7(9), e13608. https://doi.org/10.2196/13608

35. Vishwakarma, K. (2023). A comprehensive analysis and strategy development for maximising code reusability in cross-platform development frameworks. International Journal of Advanced Research in Science Communication and Technology, 606-614. https://doi.org/10.48175/ijarsct-11494

36. Wang, Y., Benson, E., Fördős, F., Lolaico, M., Baars, I., Fang, T., … & Högberg, B. (2021). Dna origami penetration in cell spheroid tissue models is enhanced by wireframe design. Advanced Materials, 33(29). https://doi.org/10.1002/adma.202008457

37. Yahya, N. and Maidin, S. (2023). Hybrid agile development phases: the practice in software projects as performed by software engineering team. Indonesian Journal of Electrical Engineering and Computer Science, 29(3), 1738. https://doi.org/10.11591/ijeecs.v29.i3.pp1738-1749

38. Zasa, F., Patrucco, A., & Pellizzoni, E. (2021). Managing the hybrid organization: how can agile and traditional project management coexist?. Research-Technology Management, 64(1), 54-63. https://doi.org/10.1080/08956308.2021.1843331

39. Zhou, Y., Qi, H., & Ma, Y. (2019). End-to-end wireframe parsing. https://doi.org/10.1109/iccv.2019.00105