CS Final Project
University of Haifa

Final Report

PILL MINDER PLUS

Rima Hleihil & Aya Mahagna

203333539 314774639

Table of Contents

The Problem

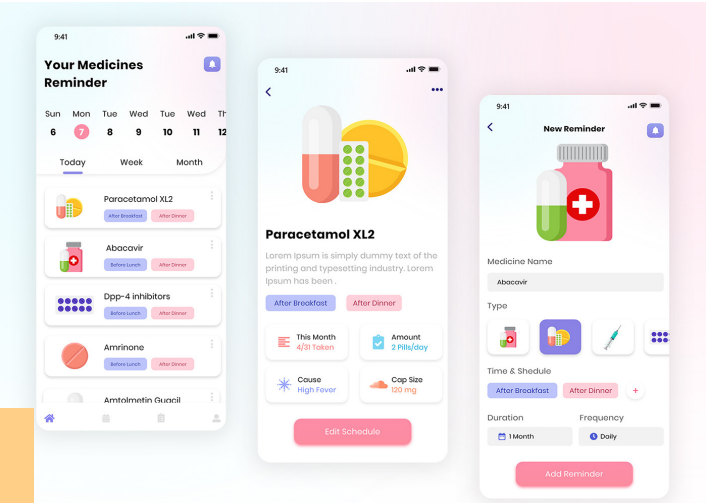
Many patients, especially the elderly, conveniently forget to take their medicine and pills a lot of times, or even don't remember whether they have taken them or not.
As we all know, taking medicines is vital, but sometimes we just forget. You may be at risk of serious health problems if you fail to take your medicines on time.

Medical errors are the third leading cause of death after heart disease and cancer, according to studies.
As a result, if a patient forgets to take his medicine, then he might be concerned and not know what to do, so real-time support and professional advice may be needed.

Moreover, patients, especially the elderly, can't always go to pharmacies and get their medicine, and they may not always have someone living with them or nearby who can help. Consequently, this might be another reason for forgetting to take medicine.

The importance of taking medicine at a suitable time and dose as well as tracking measurements, readings, and other vital signs cannot be overstated. Nevertheless, it's not as easy as it sounds, especially when we're stressed out, we might forget, and that's a serious issue.

Solution - Project Definition

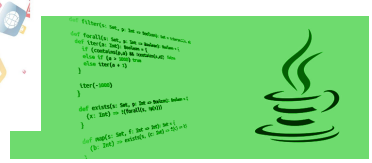


As part of this project, we propose building a mobile application that helps people remember to take their medicines on time. It's a must-have application for the elderly because they usually forget to take them on time. This app reminds them to take medication whenever the body requires them.

The app provides **medication reminders** for patients to take the required pills on time and in proper quantity.
Also to **alert** them if they want to take a medicine that they already took and forgot as well as important drug interaction warnings, missed medication alerts, refill reminders when you're running low, and family scheduling tools. That way you and a caregiver will get notifications.

In addition, it allows users to **track tablets, doses, measurements, overall health, healthy habits, goals, and activities** if they want to, also they can **share** the information with relatives or pharmacists, as well as **other features** (like **online order and delivery** that may significantly facilitate patient's life) we describe later in the features section in this report.

Implementation

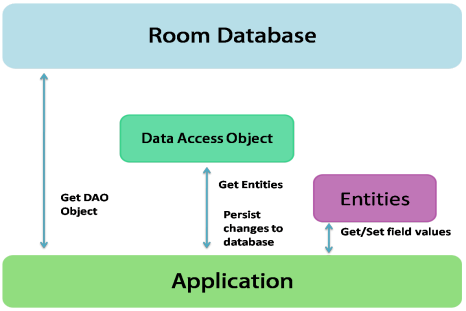


We chose to implement this app as an Android app.

Java- We've used Java programming language since it is appropriate for creating an Android app.

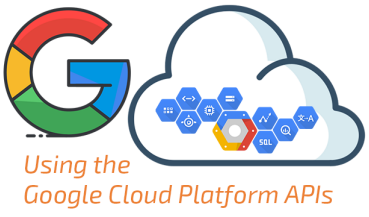
Android Studio- is our working environment and develop using Java

Rooms- Room Database is quite perfect for saving database locally and offline so the reminders will work also offline so.



Firebase- A collection of hosting services called Firebase may host any kind of application. It provides real-time and NoSQL hosting for databases, content, social authentication, and alerts, as well as other services like a real-time communication server.

Google APIs



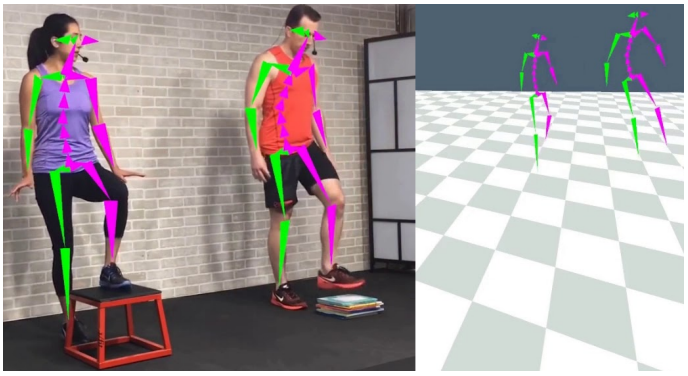
Features

- Register / Sign in / Reset password by sending mail
- Pills taking reminder
- Editing/deleting medicine reminders
- Prescription refill notices and reminders
- Medication history/calender can be shared with caregivers
- Tracking a patient's medication history/calender
- Intuitive visual interface
- A calendar to note your upcoming medical appointments
- Suspend notifications if needed
- Adding caregiver as a friend
- Map that shows nearby medical centers and pharmacies
- Snoozing the reminder for 5 minutes
- Navigate the chosen pharmacy or medical center in google maps

Future Improvements

Here are points we thought about:

- Observing and tracking a patient's body's movements and facial expressions so that it can help early recognize a problem.



- Supporting more languages.

- Voice support.

- An app that supports the iOS system as well.

Our Learning and Implementation

So far, the whole field (Application Development) was new to us, so we had to dig in deeply before we could actually implement something.

We gained experience creating applications. We had no prior experience or training in building apps; we had to learn it on the job.

We learned about the apps architecture and operations, as well as the development environment and its tools.

As far as we are concerned, the main work and efforts on this project was to understand the environments we are working with.

Additionally, we learned:
- how to use the JSON file format, which is used in Java to create and read objects.

- how to use the Google API to work with maps

- about Firebase, We gained knowledge about how the Firebase Real Time Database works and how to use it with Android Studio.

The programming language we used, Java (although the project gave us a deeper understanding of it), and the OOP methodology we employed were things we understood in advance.

We worked on the app on the Android Studio workspace using a USB connection to the phone while developing it and also virtual devices.

The components of our application (or any app in genral), are made up of a variety of Activities, each of which are linked to layouts.
We also have fragments and adapters. A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments cannot live on their own--they must be hosted by an activity

or another fragment.
An Adapter object acts as a bridge between an AdapterView and the underlying data for that view. The Adapter provides access to the data items. The Adapter is also responsible for making a View for each item in the data set.

Bottleneck - The Main difficulty

At start, we had difficulty to decide which programming language and working environments to choose. It was not easy at all since we did not try to develop an application before.
In addition, we had a hardship how to store retrieve and divide the data between the local database(room) and firebase, so we can get the best performance.

Link for our code:

https://drive.google.com/file/d/1hcEY1YbUtuosNXsxcfFC6_7H_ll_X48O/view?usp=sharing

- ★ The documentation for the code is written in the code itself as comments. You can open the classes in the drive link above and see all the documentation for a better understanding of our implementation.