MEDICAL PILL REMINDER APP

Problem Statement:

To design an Android application to create a medical pill reminder and tracker app for multiple users/patients at once. The app aims to assist multiple users in managing their medication schedules efficiently, so that family members, especially the elderly can be benefitted by saving their schedules in one app in the family.

Software Requirements:

a) Functional Requirements:
   1. Multi-user Management: Allows multiple users to be added within same app.
   2. Medication Management: Add and edit medications with details like medicine pill name and the quantity of the medicine. Users can view these details effectively.
   3. Reminder Functionality: Users can create reminders and set alarms for each medication specifying the alarm time, repeat options, days of the week, and dosage quantity. This allows custom repeat options and dosage.
   4. Notification Functionality: When the alarm is triggered, a notification is displayed.
   5. Snooze/Turn Off Functionality: The notification/alarm can be snoozed or turned off.
   6. User Interface: A user-friendly interface for easy navigation and interaction, displaying a clear overview of all medications and their respective reminders for each user.

b) Non-Functional Requirements:
   1. Performance: The app responds promptly to user actions without significant delays. And utilizes device resources efficiently to ensure smooth performance.
   2. Security: Secure storage of user data, including medications and user profiles using SQLite Database.
   3. Reliability: Operates reliably without frequent crashes or errors, with effective error handling to handle unexpected scenarios gracefully..
   4. Scalability and Maintainability: App architecture is structured in a way to integrate potential future updates or enhancements and involves clean, well-documented code for ease of maintenance and future development.
   5. Offline Advantage : Uses an offline database which is stored in the device, no internet connection is required, offline storage of the database helps in faster read write operations.
   6. User Friendly UI : Simple yet effective UI that is pleasing to the eye.

Functionalities:

1. User Profile Authentication Module:

Allows users to add new users with necessary details and enable existing users to log in securely.

2. Medication Management Module:

Allows users to add new medications with details (name, dosage, time, etc).

3. Reminder Setup Module:

Allows users to create reminders for each medication specifying alarm time, repeat options, days of the week, and quantity, customizing the reminder settings for each medication. A list of scheduled reminders for each medication will be displayed.

4. Alarm and Notification Module:

Users can activate alarms based on scheduled medication reminders and view notifications when the alarm triggers. The alarm can also be turned off or snoozed.

5. Data Storage and Database Module:

Implemented database functionalities to store user profiles, medications, and reminder settings using SQLite Database to ensure secure storage of sensitive user information within the database.

Schema:

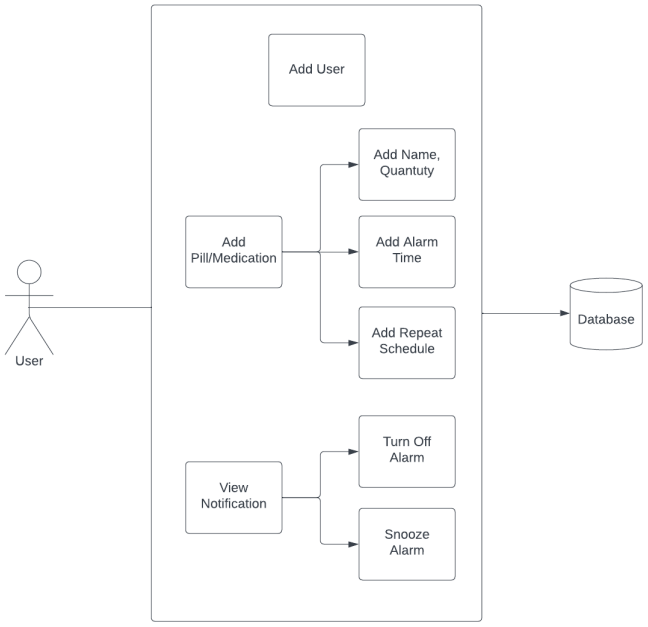
User

_id	User_Name

Medicine

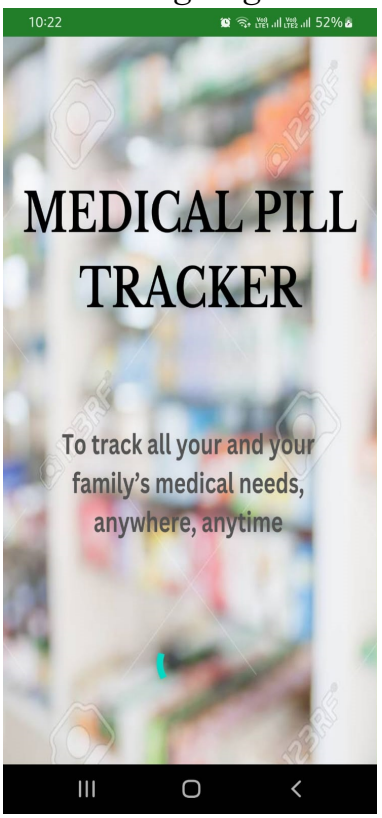
_id	Med_Name	Qty	Date_Time	Days	User_Id

Design:

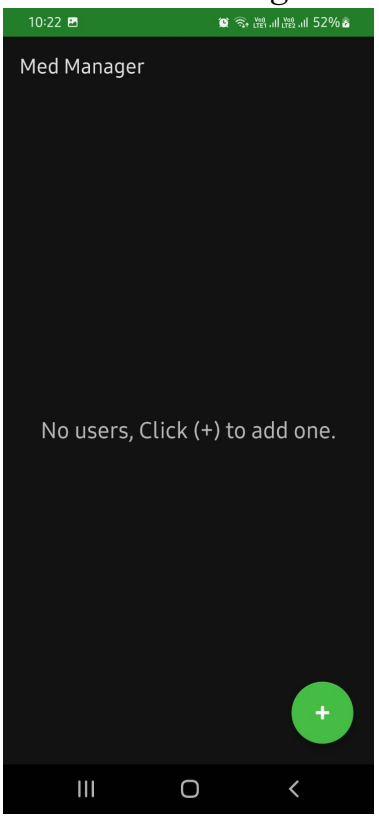


Output Screenshots:

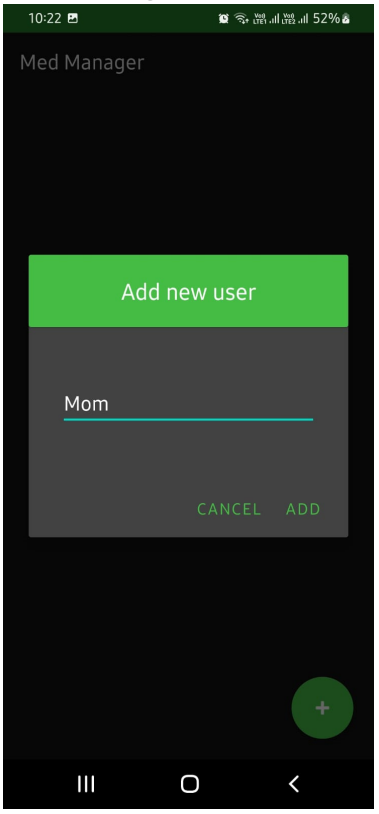
Loading Page



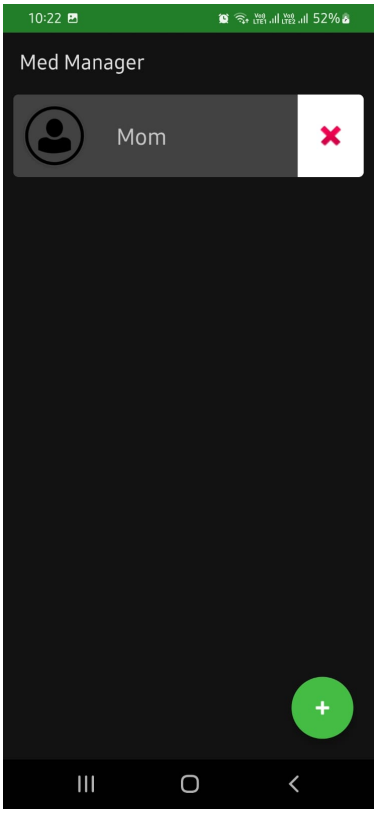
Home Page



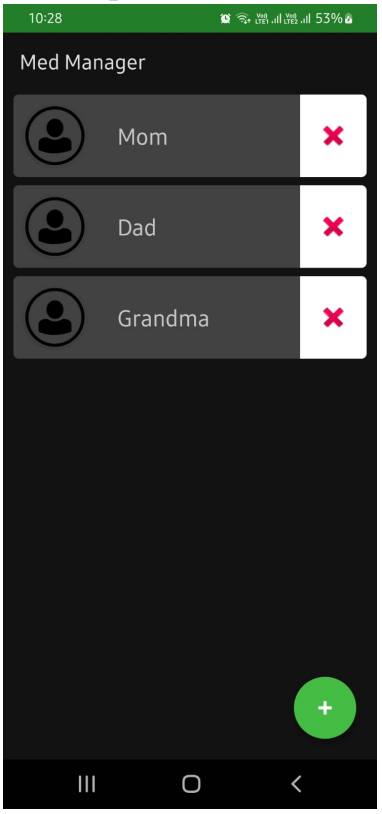
Adding New User



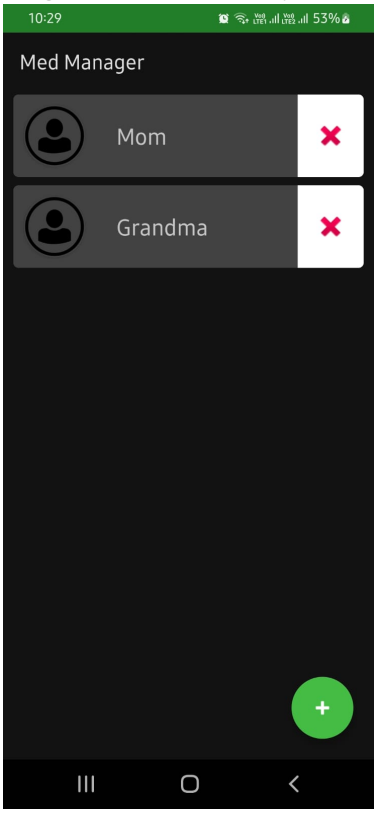
New User "Mom" Added



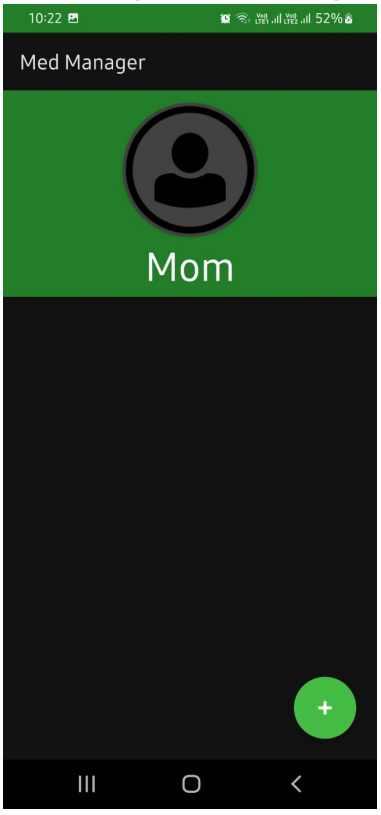
Multiple Users Added



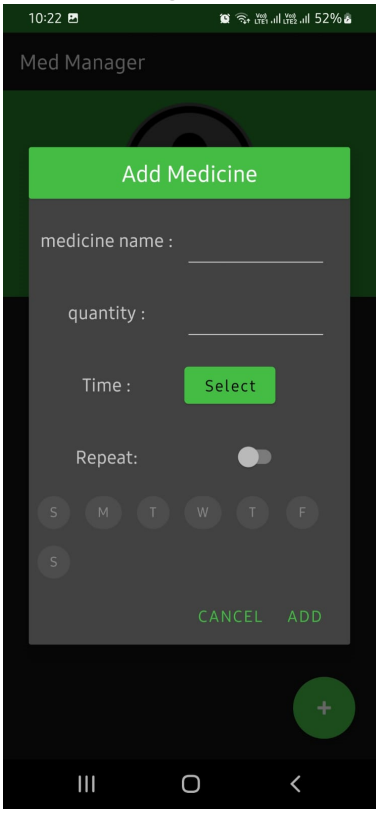
Deleting User "Dad" by Clicking "X"



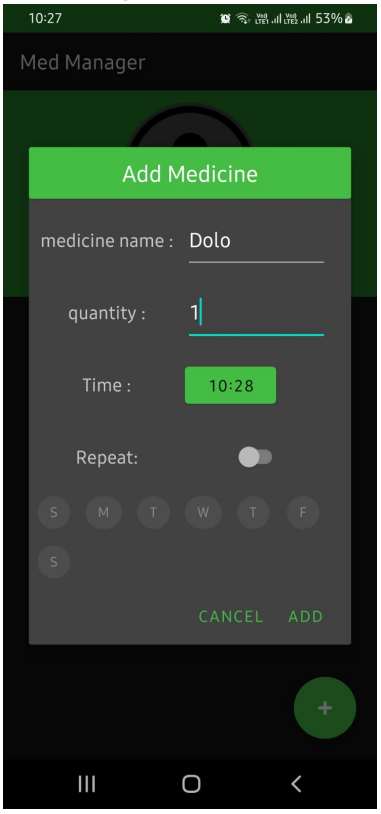
Medicine Page on Clicking "Mom"



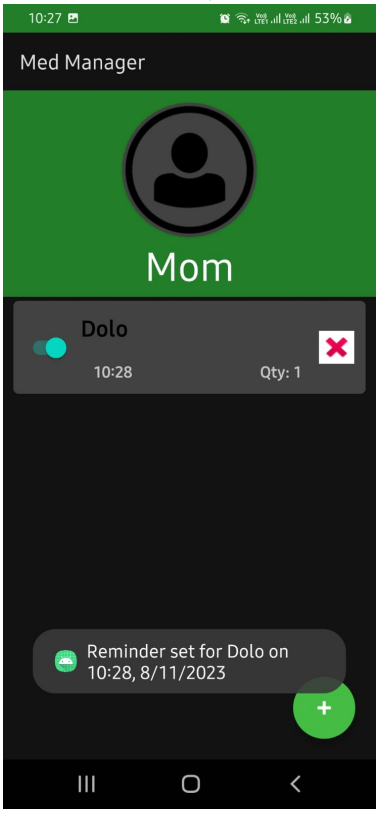
Adding New Pill



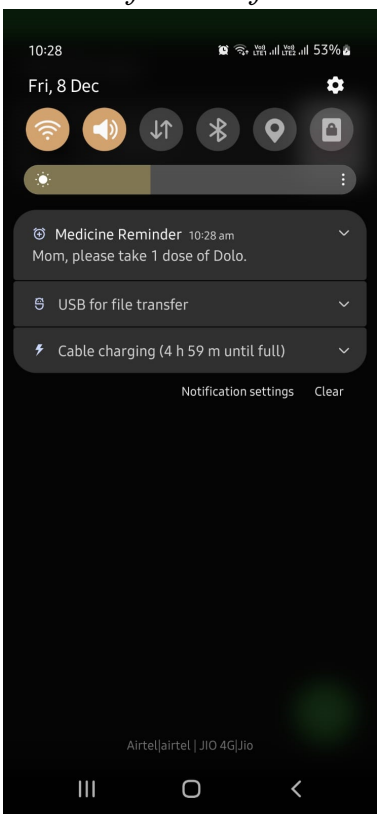
Entering New Pill Detail



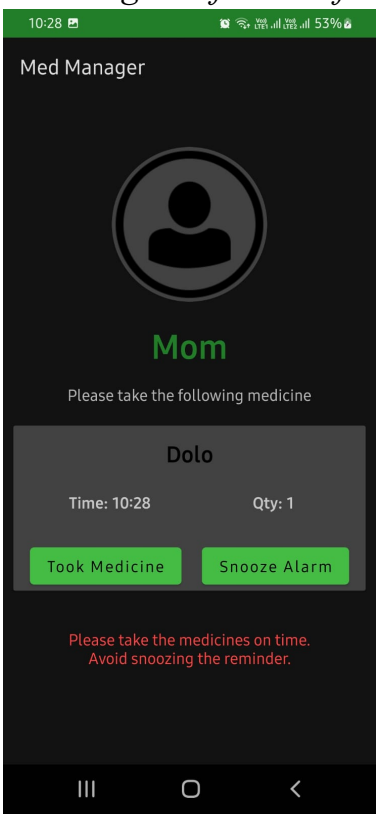
Alarm Activated for Pill "Dolo"



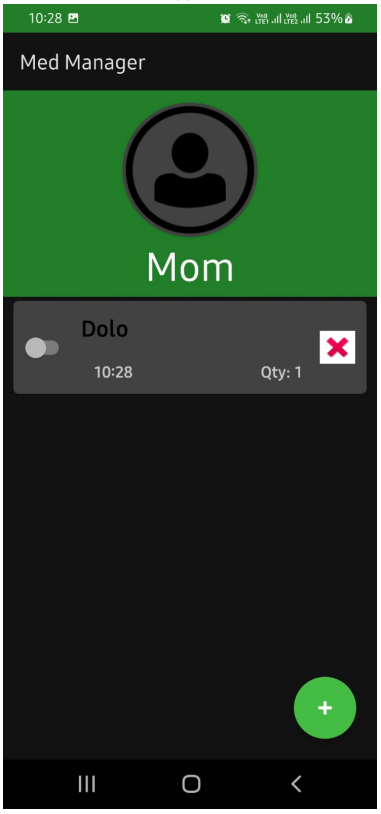
Alarm Notification for "Dolo"



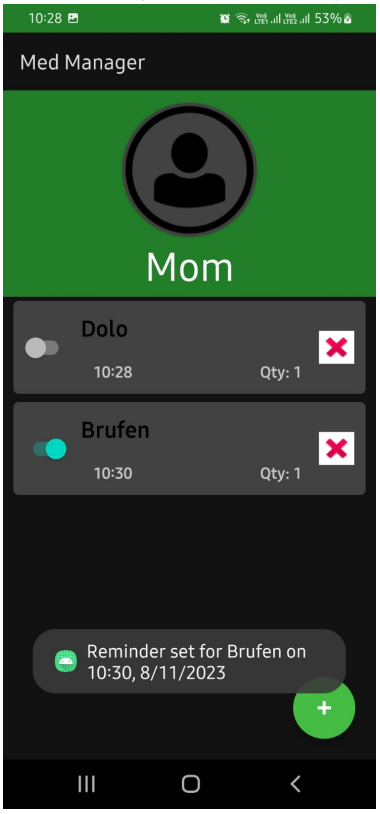
On Clicking Notification for "Dolo"



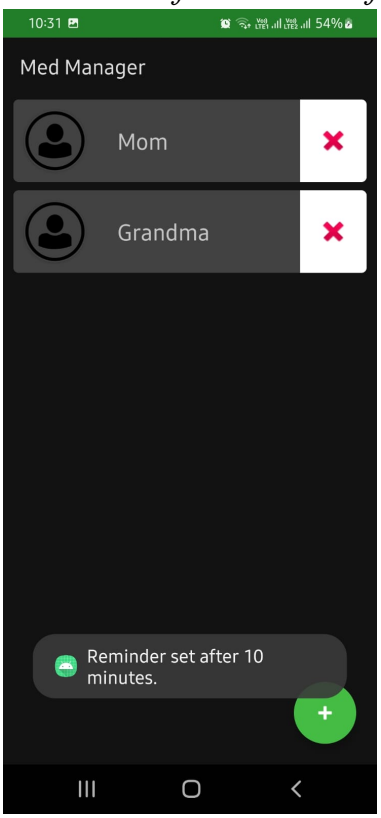
Turn Off Alarm



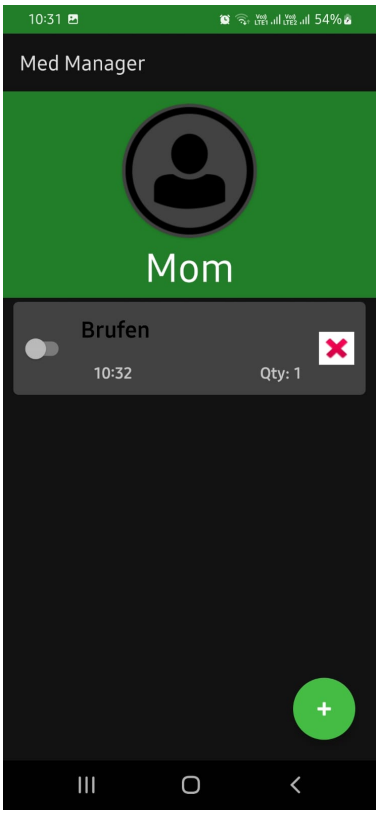
Set Alarm for Pill "Brufen"



Snooze Alarm for Pill "Brufen"



Delete Pills



Best Practices:

1. The code has been divided into modules and is clean for easier maintenance and scalability.
2. Code is clean, readable, and well-commented.
3. We used version control in Git to track changes, collaborate, and maintain different versions of the code.
4. The code was regularly tested and dependencies and libraries were updated often.
5. The project was documented in a neat comprehensive report.