Project Title: MediTrack (Smart Medication Management & Health Reminder app)

Introduction

In this fast-paced life, people tend to forget to take medicine on time, and for patients having chronic diseases and elder, it poses a serious health problem. There are a few reminders apps available, but they are usually made needlessly complicated and packed with features that are unnecessary.

The project suggests the design of the MediTrack as a simple and efficient easy-to-use Android App that facilitates medication and health reminder plan for the users. App will stick to the necessary features

Objectives

- To design and implement a mobile application that provides timely medication reminders.
- To create an easy-to-use interface suitable for users of all ages.
- To allow users to track medication intake history.
- To demonstrate the application of standard software development principles within a 5-week timeline.

Functional Requirements

These define the core functions the app must provide.

1.1 User Management

- The system shall allow users to launch and use the app without requiring login (single-user mode).

1.2 Medication Management

- The system shall allow users to add medications with details:
  - Name of medication
  - Dosage (e.g., 1 pill, 5 ml)
  - Frequency (times per day)
  - Start and end date (optional)
  - Reminder times
- The system shall allow users to edit or delete medications.
- The system shall store medication data in a local database (Room)**.**

1.3 Reminder Notifications

- The system shall send local notifications at the scheduled times.
- The notification shall display:
    - Medication name
    - Dosage
    - Action buttons: Mark as Taken or Snooze**.**
- The system shall re-notify after a snooze period (default 10 minutes).

1.4 User Interface

- The system shall use Material Design guidelines.
- The main navigation shall include:
    - Home (Today's Medications)
    - Add Medication
    - History

Sprint 1 (Week 1) – Planning & Setup

Backlog Items:

- Backlog 1: Define requirements and prepare wireframes.
- Backlog 2: Initialize Android Studio project.
- Backlog 4: Implement app navigation skeleton (Home, Add Medication, History).

Deliverable: Working skeleton app with basic navigation and empty screens.

Sprint 2 (Week 2) – Database & Models

Backlog Items:

- Backlog 1: Create `Medication` entity class.
- Backlog 2: Implement Room Database schema and DAO..
- Backlog 3: Connect database with navigation skeleton.
- Backlog 4: Test with placeholder medication entry (hardcoded first).

Deliverable: App can store and retrieve medication data from database.

Sprint 3 (Week 3) – Core Features (CRUD)

Backlog Items:

- Backlog 1: Build "Add Medication" UI form.
- Backlog 2: Implement "Edit Medication" feature.
- Backlog 3: Implement "Delete Medication" option.
- Backlog 4: Integrate RecyclerView to display medication list.
- Backlog 5: Validate user inputs (name, dosage, time).

Deliverable: User can fully Add, View, Edit, and Delete medications.

Sprint 4 (Week 4) – Notifications & History

Backlog Items:

- Backlog 1: Implement local notifications (AlarmManager / WorkManager).
- Backlog 2: Add "Mark as Taken" and "Mark as Missed" actions in reminders.
- Backlog 3: Store taken/missed logs in history table.
- Backlog 4: Build History screen (last 7 days' logs).
- Backlog 5: Test reminder functionality in background.

Deliverable: App sends reminders at correct times + logs history.

Sprint 5 (Week 5) – Testing & Polish

Backlog Items:

- Backlog 1: Test app on emulator (multiple Android versions).
- Backlog 2: Test app on physical device.
- Backlog 3: Fix bugs & crashes from testing feedback.
- Backlog 4: Improve UI with Material Design (buttons, colors, spacing).
- Backlog 5: Prepare project documentation (features, architecture, screenshots).
- Backlog 6: Generate and package final APK.

Deliverable: Stable app with documentation and APK ready for submission.

Features (MVP)

The app will include the following features:

1. **Add / Edit / Delete Medications**
    - Users can input medicine name, dosage, frequency, and time.
2. **Medication Reminder Notifications**
    - Local notifications will alert users when it's time to take their medicine.
3. **Mark as Taken / Missed**
    - Users can log whether they took or skipped the dose.
4. **History Log**
    - View a simple list of recent medication records.
5. **Basic UI**
    - Simple, clean interface following Material Design guidelines.

Tools and Technology

- **IDE:** Android Studio
- **Programming Language:** Java
- **Database:** Room (SQLite wrapper for Android)
- **Notification Handling:** WorkManager
- **UI Components:** Material Design, RecyclerView, CardView

Target Users

- Elderly individuals who need consistent reminders for daily medications.
- Chronically ill patients who take multiple medicines.
- Caregivers who want to help family members manage their medication schedules.
- General users who want simple health or vitamin reminders.

Software Development Model

The Agile Scrum model will be most suitable for this project because:

- The project is time-constrained (5 weeks), requiring fast iterations.
- Features can be divided into small deliverables (sprints).
- Feedback and testing can be integrated at the end of each sprint.
- Flexibility allows scope adjustment if time runs short.

Conclusion

MediTrack addresses a clear and meaningful need which is, helping users reliably take their medications. While remaining achievable within the five-week timeframe. By focusing on a small, high-impact MVP (medication CRUD, a simple history log) and following an Agile/Scrum approach, the project balances rapid iteration with quality deliverables. Expected outputs at the end of the course are a tested Android APK,