Project Title: PLANANA ANDROID APPLICATION

Project Description:

🍌 Welcome to Planana Your fun, flexible, and focused weekly planner. Life gets busy — but staying organized doesn't have to be boring. Planana is your cheerful productivity companion designed to make planning your week feel smooth, satisfying, and even a little bit fun. With Planana, you can effortlessly manage your weekly schedule, jot down important events, keep track of your tasks, and stay on top of your to-do list — all in one clean and intuitive app. Planana helps you stay productive without the pressure, making planning feel light, fun, and truly yours.

Configuration

All versions must be updated to the latest version.

Development Environment:

- Android Studio Meerkat 2024.1.1
- Java 17

SDK Requirements:

- Minimum SDK: API 24 (Android 7.0 – Nougat)
- Target SDK: API 34+ (Android 14.0) – Future Proofing

Dependencies

- SQLite
- Firebase
- Junit
- Logcat

Background Services:

- AlarmManager
- Android Services

Notification Management:

- NotificationManager

Basic General Flow

User logs in → User views home screen → User creates task and save→ User sees ongoing tasks → User will be notified for the task created → User can see notifications in notification box → User can view entire task progress → User logs out

Requirements

1. Database Storage
   a. All user data, tasks, and schedules are stored locally in an SQLite DB.
2. Login
   a. Users must log in to access personalized planning tools.
   b. Store credentials in SQLite DB and hash the passwords.
3. Registration
   a. Users can register with an email and password.
4. Task Management
   a. Create, edit, and delete tasks.
   b. Tasks are tagged with date, time, category (e.g., Personal, School, Work).
   c. Push notifications or scheduled alarms are sent 15 minutes before a task starts.
   d. Notifications also appear in the notification tray.

5. Task Notifications
    a. Notification system alerts users of upcoming tasks.
    b. Updates come in every 5 seconds (to prevent network congestion and large data consumption).
    c. However, if the transporter is offline, then delivery progress may not be updated in real time.
6. Task Progress and Summary
    a. Users can view the number of tasks and current progress.
    b. Percentage-based progress for visual feedback.
7. Logs and History
    a. View all created, completed, and deleted tasks.
    b. Sort logs by date, priority, or category.

Entities

1. User
    a. Registers and logs in to access personal planning data.
    b. Has one-to-many relationships with tasks.
2. Tasks
    a. Contains title, description, date, time, and category.
    b. Belongs to one user.
    c. May include notification flags and visual customizations.

System Architecture Overview

Layer	Stack / Tool	Notes
Frontend UI	Android XML Layout, Java	Manages layout, user interaction, widgets, and visual feedback
App Logic	Java 17, ViewModels	Handles input validation, state management, event binding
(App) Service Layer	Custom Service Classes	Contains business logic, mediates between View and Model (Presenter)
Data Layer	SQLite,	Local database storage for tasks and user profiles
Session Management	SharedPreferences	Stores user login state and lightweight preferences
Background Jobs	Android Services, AlarmManager	Schedules task reminders and background syncing
Network / APIs	NotificationManager	Pushes local reminders to the notification tray
Push Notifications / Realtime	Firebase	Pushes local reminders to the notification tray
Testing	JUnit, Logcat	For testing individual modules and tracking runtime errors