Maintenance Guide

Purpose

This guide explains how to set up, run, maintain, and update the Smart Planner and Budget Coach application. It is intended for developers responsible for ongoing maintenance, troubleshooting, or future enhancements. The information provided outlines the required environment, project setup, and the best practices needed to support the application throughout its lifecycle.

Environmental Requirements

Application: Smart Planner and Budget Coach
Platform: Android 8.0+ (API 26)

Android SDK:

- o   Minimum API 26

- o   Target: Defined in build.gradle

Version Control: Git (GitLab Repository)

Testing: Android Emulator or Physical Device

Project Setup

1. Clone the GitLab Repo

2. Open the project in Android Studio

3. Confirm Gradle JDK = JDK 17

4. Allow Gradle to sync and install missing dependencies

5. Build the project to confirm successful setup.

Application Architecture

UI Layer: Activities and RecyclerView Adapters

Data Layer

- o Room entities

- o DAOs

- o Room database class

Persistence: Local SQLite database via Room framework

Database Maintenance

Entity changes require:

- o Updating the entity class.

- o Incrementing the Room database version.

- o Adding a migration strategy.

App data is cleared by uninstalling the app or wiping emulator data.

Common Maintenance Task

Feature Updates

- o UI changes: res/layout.

- o Logic updates: Activities and adapters.

- o Data Changes: Entities, DAOs, and Repositories.

- o Test all changes on the emulator or a physical device.

Dependency Updates

- o Update Gradle dependency versions.

- o Sync and rebuild the project.

- o Validate application stability.

Version Control

- o Use feature branches.

- o Commit with clear and descriptive messages.

- o Merge only after successful build and testing.

Testing

- o Verify CRUD functionality.

- o Confirm budget calculations update correctly.

- o Test orientation changes and app restarts.

- o Validate behavior across different screen sizes.

Troubleshooting

- o Build failures: verify JDK and SDK version and invalidate caches.

- o App crashes: Review Logcat output.

- o Database errors: Validate entity and DAO alignment.

Deployment

- o Run app using Android Studio Run option.

- o Generate APK via Build -> Build APK

Maintenance Best Practices

- Keep SDK and dependencies up to date.

- Test after every change.

- Maintain consistent project structure.

  Update documentation as features evolve.