**Section 42 — Integrating AdMob (Complete Developer Notes)**

**1. Key Concepts from the Section**

1. **What is AdMob?**
   * Google’s platform for showing ads inside Android/iOS apps.
   * Revenue model: CPC (cost-per-click) and CPM (cost per thousand impressions).
   * Works via **App ID** (per app) and **Ad Unit ID** (per ad type in the app).
2. **Ad Types Covered:**
   * **Banner Ads** – persistent rectangular ads in UI.
   * **Interstitial Ads** – full screen, shown at logical breakpoints.
   * **Rewarded Ads** – give user a benefit after watching.
3. **Setup Requirements:**
   * AdMob account → Add app → Get App ID.
   * Create Ad Units → Get Ad Unit IDs for each type.
   * Add **Google Mobile Ads SDK** dependency.
   * Add Internet permissions.
   * Initialize the SDK in your app.
4. **Test vs Production Ads** – why you must use Google’s test IDs in development to avoid policy violations.
5. **Manifest Metadata** – linking App ID.
6. **Placement & Timing Best Practices** – avoiding user frustration.

**2. Step-by-Step Implementation**

**Step 1 — Add SDK Dependency**

In app/build.gradle:

dependencies {

// Google Mobile Ads SDK

implementation 'com.google.android.gms:play-services-ads:22.6.0' // Latest stable as of 2025

}

**Why?**  
This brings in the classes like AdView, InterstitialAd, RewardedAd needed to load and show ads.

**Step 2 — Add Permissions**

In AndroidManifest.xml (outside <application>):

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

**Why?** Ads need an internet connection and must detect network state.

**Step 3 — Add App ID Metadata**

Inside <application> in AndroidManifest.xml:

<meta-data

android:name="com.google.android.gms.ads.APPLICATION\_ID"

android:value="ca-app-pub-3940256099942544~3347511713"/> <!-- TEST App ID -->

**Note:** Replace with your **real App ID** from AdMob when going live.  
Use Google’s **test App ID** for development.

**Step 4 — Initialize SDK in MainActivity**

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Initialize Mobile Ads SDK — must be done before loading ads

MobileAds.initialize(this, initializationStatus -> {

// Optional: You can log init status

}); }

Kotlin equivalent:

MobileAds.initialize(this) { /\* init complete \*/ }

**Step 5 — Banner Ads Implementation**

**Layout (activity\_main.xml)**

<com.google.android.gms.ads.AdView

android:id="@+id/adView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

ads:adSize="BANNER"

ads:adUnitId="ca-app-pub-3940256099942544/6300978111" <!-- TEST Banner Unit -->

app:layout\_constraintBottom\_toBottomOf="parent"/>

**Activity Code**

AdView adView = findViewById(R.id.adView);

AdRequest adRequest = new AdRequest.Builder().build(); // Default ad request

adView.loadAd(adRequest); // Start loading banner ad

**Step 6 — Interstitial Ads Implementation**

**Java (latest SDK flow)**:

private InterstitialAd mInterstitialAd;

private void loadInterstitial() {

AdRequest adRequest = new AdRequest.Builder().build();

InterstitialAd.load(this,

"ca-app-pub-3940256099942544/1033173712", // TEST ID

adRequest,

new InterstitialAdLoadCallback() {

@Override

public void onAdLoaded(@NonNull InterstitialAd interstitialAd) {

mInterstitialAd = interstitialAd;

}

@Override

public void onAdFailedToLoad(@NonNull LoadAdError adError) {

mInterstitialAd = null;

}

});

}

private void showInterstitial() {

if (mInterstitialAd != null) {

mInterstitialAd.show(this);

}

}

**Best practice:** Load at app start → Show only at natural breaks.

**Step 7 — Rewarded Ads Implementation**

private RewardedAd rewardedAd;

private void loadRewardedAd() {

AdRequest adRequest = new AdRequest.Builder().build();

RewardedAd.load(this,

"ca-app-pub-3940256099942544/5224354917", // TEST ID

adRequest,

new RewardedAdLoadCallback() {

@Override

public void onAdLoaded(@NonNull RewardedAd ad) {

rewardedAd = ad;

}

@Override

public void onAdFailedToLoad(@NonNull LoadAdError adError) {

rewardedAd = null;

}

});

}

private void showRewardedAd() {

if (rewardedAd != null) {

rewardedAd.show(this, rewardItem -> {

int rewardAmount = rewardItem.getAmount();

String rewardType = rewardItem.getType();

// Give reward to user

});

}

}

**Tip:** Use rewards that matter — coins, extra lives, premium features.

**Step 8 — Test Ads During Development**

* Banner: ca-app-pub-3940256099942544/6300978111
* Interstitial: ca-app-pub-3940256099942544/1033173712
* Rewarded: ca-app-pub-3940256099942544/5224354917

**Never use real IDs in dev** → Google may ban your account.

**Step 9 — Organizing for Future Projects**

Instead of repeating code, create AdHelper:

public class AdHelper {

public static void loadBanner(AdView adView) {

adView.loadAd(new AdRequest.Builder().build());

}

public static void loadInterstitial(Context ctx, String unitId, Consumer<InterstitialAd> callback) {

AdRequest adRequest = new AdRequest.Builder().build();

InterstitialAd.load(ctx, unitId, adRequest, new InterstitialAdLoadCallback() {

@Override

public void onAdLoaded(@NonNull InterstitialAd ad) { callback.accept(ad); }

@Override

public void onAdFailedToLoad(@NonNull LoadAdError adError) { callback.accept(null); }

});

}

}

Makes integration **1–2 lines** per ad type.

**3. Tools, Libraries, APIs**

* **Google Mobile Ads SDK** (com.google.android.gms:play-services-ads)
* **AdView**, **InterstitialAd**, **RewardedAd** classes
* AdMob Web Console

**4. Best Practices (Industry-Level)**

* **Ad placement** → Don’t place ads where users will accidentally click.
* **Load early, show later** → Prevents UI delays.
* **Respect UX** → Limit interstitials to natural flow breaks.
* **Reward meaningful actions** → Improves user retention.
* **Check connectivity** before loading ads.
* Use **App-Open Ads** for splash screens (2025 trend).
* Combine with **Firebase Analytics** to track ad engagement.

**Part B — Additional Knowledge Not in the Section**

* **Native Advanced Ads**: Better CTR when styled to match your UI.
* **Adaptive Banners**: Auto-resizes for device width.
* **Mediation**: Increase fill rates by connecting multiple ad networks via AdMob.
* **Consent SDK**: Required for GDPR compliance in EU markets.
* **Server-Side Verification for Rewards**: Prevents fraud.
* **Frequency Capping**: Avoid showing too many ads.
* **Testing on Physical Device**: Use addTestDevice() for your dev phone.
* **Policy Compliance**: Read [AdMob Policy](https://support.google.com/admob) — violations can get your account banned permanently.