10. Návrh a vývoj mobilních aplikací

Vývojové platformy (iOS, Android, cross-platform)

Operační systémy iOS a Android jsou dvě hlavní platformy pro vývoj mobilních aplikací, které mají svá vlastní vývojová prostředí, nástroje a ekosystémy. Existují také multiplatformové vývojové nástroje, které umožňují vytvářet aplikace pro více platforem současně.

iOS

- Vývoj v Xcode, programování ve Swiftu nebo Objective-C.
- Distribuce přes App Store, aplikace jsou schvalovány Applem.
- **Výhody:** Stabilita, dlouhá podpora zařízení, rychlé aktualizace, silné zabezpečení.
- **Nevýhoda:** Uzavřený ekosystém (jen Apple zařízení).

Android OS

- Vývoj v Android Studio, jazyky Java nebo Kotlin.
- Otevřený systém, možnost publikace bez schvalování, obrovská rozmanitost zařízení.
- **Výhody:** Největší podíl na trhu, široká přizpůsobitelnost.
- **Nevýhoda:** Fragmentace zařízení (různé verze Androidu, různé parametry HW).

Cross-platform

- Frameworky: Flutter (Dart), React Native (JavaScript/TypeScript), Xamarin (C#), Unity (hlavně pro hry).
- Jeden kód lze spustit na iOS i Androidu, což šetří čas a náklady.
- **Nevýhoda:** Ne vždy plný přístup k nativním funkcím, případně složitější optimalizace a horší výkon v porovnání s nativním vývojem.

Vývojové prostředí, historie a trendy

iOS

- První verze vydána v roce 2007 s iPhonem.
- Od té doby rychlý vývoj – přidán App Store, multitasking, podpora nových zařízení (iPad, Watch).
- **Současnost:** Moderní iOS nabízí pokročilé funkce (např. ARKit pro rozšířenou realitu), vysokou bezpečnost a pravidelné aktualizace pro většinu zařízení.

Android

- První verze v roce 2008.
- Postupně přibývaly nové funkce, lepší rozhraní, podpora různých velikostí obrazovek, bezpečnostní prvky.
- **Současnost:** Android běží na nejširší škále zařízení (telefony, tablety, TV, automobily) a podporuje i více výrobců současně.

Cross-platform vývoj

- V minulosti se aplikace vyvíjely vždy odděleně pro každou platformu.
- Dnes se běžně používají multiplatformní frameworky (Flutter, React Native), které šetří čas i náklady při vývoji aplikací pro více systémů najednou.
- Přesto v některých případech zůstává výhodnější nativní vývoj (např. pro maximální výkon nebo specifické funkce dané platformy).

Základy programování v Javě (pro Android)

- Java je objektově orientovaný jazyk, určený pro robustní a přenositelné aplikace.
- **Třídy a objekty:** Třída je předloha, objekt je konkrétní instance třídy.
- **Dědičnost:** Umožňuje tvořit hierarchie tříd a sdílet společnou funkcionalitu.
- **Polymorfismus:** Objekty různých tříd mohou být zpracovávány stejným rozhraním.
- **Výjimky:** Konstrukce try-catch-finally pro zpracování chyb.
- Znalost Javy je základem pro porozumění Androidu (Kotlin je moderní alternativou, syntaxe je podobná).

Úvod do vývojového prostředí Android Studio (LogCat, ADB)

- **Android Studio** je oficiální IDE pro vývoj aplikací na Android.
- **LogCat:** Zobrazuje logy a výstupy běžících aplikací – důležité pro ladění a diagnostiku chyb.
- **ADB (Android Debug Bridge):** Nástroj pro komunikaci s fyzickým nebo emulovaným zařízením. Umožňuje instalaci aplikací, spouštění příkazů, získávání logů a ladění.

Android SDK

- Android SDK (Software Development Kit) obsahuje všechny nástroje potřebné pro vývoj aplikací pro Android včetně knihoven, rozhraní API a nástrojů pro ladění.
- Obsahuje také emulátory pro testování aplikací na různých verzích systému Android.

Senzory (WIFI, BT, GPS, GSM, kompas, gyroskop, akcelerometr)

Umožňují interaktivní a kontextové aplikace:

- **WIFI:** Umožňuje připojení k bezdrátovým sítím. API pro správu připojení, skenování dostupných sítí atd.
- **Bluetooth:** Umožňuje bezdrátovou komunikaci s jinými zařízeními. Podpora různých profilů jako klasického Bluetooth a Bluetooth Low Energy (BLE).
- **GPS:** Umožňuje určování polohy pomocí satelitů. API pro získání aktuální polohy, sledování polohy atd.
- **GSM:** Umožňuje komunikaci prostřednictvím mobilních sítí. API pro správu telefonních hovorů, posílání SMS atd.
- **Kompas:** Umožňuje určení směru. Na určení směru vzhledem k magnetickému poli Země používá magnetometr.
- **Gyroskop:** Měří orientaci a úhlovou rychlost. Používá se pro zjišťování otáčení a naklonění zařízení.
- **Akcelerometr:** Měří zrychlení a dokáže rozpoznat pohyb a otřesy. Používá se pro detekci pádů, změn orientace zařízení atd.

Android – struktura projektu (Manifest, kód, zdroje/resources, Gradle build toolkit)

Struktura projektu Android zahrnuje několik klíčových komponent:

- **Manifest:** Soubor AndroidManifest.xml obsahuje důležité informace o aplikaci, například oprávnění, aktivity a služby.
- **Kód:** Obsahuje zdrojový kód aplikace, obvykle v průčincích java nebo kotlin. Hlavní soubory jsou třídy, které definují chování aplikace.
- **Zdroje (resources):** Obsahuje soubory jako obrázky, rozložení, řetězce atd. Zdroje jsou uspořádány v průčincích jako například res/drawable, res/layout, res/values atd.
- **Gradle build toolkit:** Nástroj pro automatizaci procesu sestavování, správu závislostí a konfiguraci projektu.

Tvorba UI (XML, LinearLayout, RelativeLayout, styly, Activity, Fragment, změny konfigurace, optimalizace pro různá zařízení)

Vytvoření uživatelského rozhraní (UI) v systému Android zahrnuje:

- **XML:** Slouží k definování rozložení a vzhledu aplikace.
- **LinearLayout:** Uspořádá prvky do řádku nebo sloupce. Používá se pro jednoduché a pravidelné uspořádání prvků.
- **RelativeLayout:** Umožňuje relativní umístění prvků vůči sobě. Používá se při složitějším uspořádání prvků.
- **Styly:** Definují vzhled a styl prvků aplikace. Styly lze definovat v souboru res/values/styles.xml.

- **Activity:** Základní stavební prvek aplikace, kterým je jedna obrazovka s uživatelským rozhraním. Každá aktivita má svůj vlastní životní cyklus a může obsahovat více fragmentů.
- **Fragment:** Modulární část uživatelského rozhraní, kterou lze použít v rámci aktivity. Fragmenty umožňují flexibilní a opakovaně použitelné komponenty uživatelského rozhraní.
- **Změny konfigurace:** Odstraňování problémů souvisejících se změnami konfigurace, jako je například otáčení obrazovky. Možnosti zahrnují použití ViewModel na zachování údajů během změn konfigurace.
- **Optimalizace pro různá zařízení:** Použití různých rozložení, zdrojů obrázků a velikostí textu pro různé velikosti a rozlišení obrazovky. Responzivní design pomocí ConstraintLayout a dalších nástrojů.

Životní cyklus aplikace – Activity Lifecycle

Activity Lifecycle: Systém Android řídí životní cyklus aktivity prostřednictvím série stavů. Každý stav představuje určitou fázi života aktivity:

- `onCreate()`: Inicializace aktivity. Zde se obvykle nastavuje uživatelské rozhraní a inicializují se potřebné zdroje.
- `onStart()`: Aktivita se stává viditelnou pro uživatele, ale ještě není v popředí.
- `onResume()`: Aktivita se dostane do popředí a uživatel může interagovat.
- `onPause()`: Aktivita ztrácí fokus, ale je stále viditelná. Zde se obvykle ukládají změny, které by měly být trvalé.
- `onStop()`: Aktivita není viditelná. Systém může uvolnit zdroje.
- `onDestroy()`: Aktivita je zničena. Zde se uvolní všechny zbývající zdroje.

Komunikace mezi aktivitami (Intent, BroadcastReceiver)

- **Intent (Záměr):** Mechanismus pro zahájení nových činností nebo komunikaci mezi činnostmi. Záměry mohou být explicitní (zaměřené na konkrétní činnost) nebo implicitní (se všeobecným záměrem, který může být naplněn více aplikacemi).
- **BroadcastReceiver:** Komponent pro příjem a zpracování globálního vysílání. Vysílání mohou být systémová (např. změna stavu baterie) nebo definovaná aplikací.

Interakce s uživatelem – notifikace, toast zprávy, SnackBar, dialogy

- **Notifikace:** Informují uživatele o událostech mimo aplikaci. Zobrazují se v oznamovací liště. Systém Android poskytuje rozhraní API oznámení na vytváření a správu oznámení.
- **Toast zprávy:** Krátké zprávy, které se uživateli zobrazí na krátký čas. Neprerušují aktuální činnost uživatele.
- **SnackBar:** Rozšířená verze toastových zpráv s možností akce. Zobrazuje se ve spodní části obrazovky.
- **Dialogy:** Okna, která vyžadují interakci uživatele, jako například AlertDialog pro potvrzování akcí.

Trvalá data v aplikaci – SharedPreferences, SQLite databáze, správa souborů

- **SharedPreferences:** Ukládání jednoduchých dvojic klíč-hodnota údajů. Slouží pro ukládání nastavení a jednoduchých údajů.
- **Databáze SQLite:** Relační systém správy databází pro komplexnější ukládání údajů.
- **Správa souborů:** Ukládání a čtení údajů z interní nebo externí paměti zařízení.

Práce na pozadí – Threads, AsyncTask, Services

- **Threads:** běh kódu na pozadí.
- **AsyncTask:** (zastaralé) jednoduché úlohy na pozadí, dnes nahradily jiné mechanismy.
- **Services:** dlouhodobě běžící procesy bez přímé interakce s UI.

Android Design Guidelines – dokumentace, návody, optimalizace (best practices, používání knihoven, přehled nejpoužívanějších)

- **Dokumentace:** Oficiální dokumentace k systému Android obsahuje pokyny pro navrhování aplikací podle zásad Material Designu. Navrženo tak, aby poskytovalo konzistentní uživatelské prostředí napříč zařízeními a platformami.
- **Optimalizace:** Osvedčené postupy pro efektivní kódování, výkon a uživatelskou zkušenost. Efektivní využívání paměti a procesoru. Minimalizace spotřeby baterie. Optimalizace uživatelského rozhraní pro různé velikosti obrazovky.
- **Používání knihoven:** Populární knihovny jako Retrofit (pro HTTP požadavky), Glide (pro práci s obrázky) a Room (pro práci s databázemi).
    - **Retrofit:** Knihovna pro jednoduchou komunikaci s rozhraním REST API.
    - **Glide:** Knihovna pro efektivní vyhledávání a zpracování obrázků.
    - **Room:** Abstrakce nad databází SQLite pro jednodušší správu databázových operací.

Komunikace se senzory (WIFI, BT, GPS, GSM, kompas, gyroskop, akcelerometr)

Systém Android poskytuje rozhraní API pro komunikaci se snímači, která aplikacím umožňují přístup k údajům z různých snímačů:

- **WIFI:** API pro správu připojení k bezdrátovým sítím, skenování dostupných sítí a správu připojení.
- **Bluetooth (BT):** API pro komunikaci s jinými zařízeními prostřednictvím Bluetooth včetně klasického Bluetooth a Bluetooth Low Energy (BLE).
- **GPS:** API pro získávání aktuální polohy zařízení pomocí satelitů, sledování polohy a další funkce související s polohou.
- **GSM:** API pro správu mobilních sítí, telefonních hovorů, SMS a dalších funkcí souvisejících s mobilní komunikací.

- **Kompas:** Rozhraní API pro přístup k magnetometru zařízení s cílem určit jeho směr vzhledem k magnetickému poli Země.
- **Gyroskop:** API pro přístup k gyroskopu zařízení, který měří orientaci a úhlovou rychlost.
- **Akcelerometr:** API pro přístup k akcelerometru zařízení, který měří zrychlení a dokáže zjišťovat pohyb a otřesy.

Sdílení dat mezi aplikacemi/serverem – Content providers, SyncAdapter, REST API

- **Content providers:** Umožňují aplikacím sdílet údaje s jinými aplikacemi. Poskytují standardizované rozhraní pro přístup k údajům jako jsou kontakty, média a další.
- **SyncAdapter:** Mechanismus pro synchronizaci údajů mezi zařízeními a servery. Umožňuje automatizovanou a efektivní synchronizaci údajů.
- **REST API:** Rozhraní pro komunikaci mezi klientem a serverem pomocí požadavků HTTP. Používá se pro přenos údajů mezi aplikacemi a servery.

Podepisování aplikací a publikování na Google Play (zdarma vs. placené aplikace, In-App platby, reklamy)

- **Podpisování aplikace:** Proces zabezpečení pravosti a integrity aplikace před jejím zveřejněním. Podpísaná aplikace obsahuje digitální podpis, který ověřuje její původ a zaručuje, že s ní nebylo manipulováno.
- **Publikování v službě Google Play:** Proces nahrávání a distribuce aplikací prostřednictvím obchodu Google Play. Vývojáři si musí vytvořit účet Google Play Developer a nahrát soubor APK spolu s popisem aplikace, snímky obrazovky a dalšími informacemi.
- **Bezplatné vs. platené aplikace:** Rozhodnutí o strategii monetizace aplikace. Bezplatné aplikace mohou generovat příjmy prostřednictvím reklam nebo nákupů v aplikacích, zatímco platené aplikace vyžadují jednorázový nákup.
- **Reklamy:** Integrace reklamy na generování příjmů. Mezi oblíbené reklamní platformy patří Google AdMob, Facebook Audience Network a další.