

# Mobilné technológie a aplikácie

## Zadanie projektu LS 2025

Vytvorte vo dvojiciach mobilnú aplikáciu (iOS / Android) podľa Vášho zadania, ktoré si zvolíte na prvom cvičení. Zadanie bude preberané a hodnotené formou viacerých kontrolných bodov v priebehu semestra.

Výstupom zadania je mobilná aplikácia a samostatný backend, s ktorým aplikácia komunikuje.

## Mobilná aplikácia

Výstupom zadania je mobilná aplikácia vytvorená pre platformy iOS alebo Android. Aplikácie musia byť spustiteľné a funkčné na posledných verziách operačných systémov (Android 14, iOS 17.3) so spätnou kompatibilitou aspoň po (vrátane) Android 10 a iOS 15.

## Povolené technológie

Povolené technológie pre vytvorenie mobilnej aplikácie:

- Kotlin / Java (Android)
- Swift / Objective-C (iOS)
- Kotlin Multiplatform (cross-platform)
  - vo forme zdieľaného UI (Jetpack Compose)
  - vo forme separátneho natívneho UI pre platformu
    - Jetpack pre Android
    - SwiftUI pre iOS
- Flutter (cross-platform)
- React Native (cross-platform)
- Unity (cross-platform)

Výber inej technológie je možný po konzultácii a odsúhlasení cvičiacim na cvičeniach.

Explicitne zakázané sú všetky technológie používajúce WebView (HTML) na zobrazovanie UI (t.j. webové/hybridné platformy).

## Funkcionálne požiadavky

Aplikácia musí mať aspoň 5 funkčných obrazoviek (splash screen sa do počtu obrazoviek nepočíta), každá obrazovka pokrýva konkrétny use case. Musí byť pokrytý aspoň jeden scenár vytvorenia, úpravy a mazania dát (CRUD operácie). Komunikácia s vlastným backendom pomocou HTTP volaní na princípoch REST API.

## Povinné súčasti projektu

Implementácia musí obsahovať:

- user management (perzistentná autentifikácia používateľa)
- podporu pre fungovanie v offline režime
- podporu pre realtime komunikáciu (WebSockets/SSE - server sent events)
- podporu telefónneho a tabletového zobrazenia
  - separátny screen pre zariadenie - nepočíta sa zobrazenie telefónneho layoutu na tablete
- podporu dark mode
- podporu push notifikácií
- podporu permission handlingu

## Povinne voliteľné súčasti projektu

Implementácia musí obsahovať aspoň 3 z nasledujúcich požiadaviek:

- podpora pre ďalšie zariadenie
  - auto, hodinky, TV...
- podpora accessibility
  - čítanie obsahu, ovládanie gestami, dostatočný kontrast...
  - stačí na jednom druhu zariadenia
- podporu AR/VR
- integrácia Firebase služieb
  - analytika, crash reporting
  - výstupy z analytiky a crashreportingu budú súčasťou dokumentácie spolu s ich popisom
- location based service
  - GPS, geofencing...
- integrácia ďalšieho senzoru

- gyroskop, akcelerometer...
- background task scheduling
  - task sa spúšťa v pravidelných intervaloch bez nutnosti mať zapnutú aplikáciu
  - výstupy z analytiky jednotlivých metrík súčasťou dokumentácie spolu s ich popisom

## Nepovinné úlohy

Implementácia je na vás, môžete za ňu získať bonusové body:

- pokrytie testami
  - unit testy, UI testy, E2E testy
  - zoznam testov bude súčasťou dokumentácie spolu s ich popisom
- machine learning
  - image classification, ...
- výstup z performance analytics toolu ako súčasť dokumentácie zadania
  - Xcode, Android Studio (memory graph, energy usage, network callings...)
- generovaná dokumentácia z anotácií v kóde

## Backend

Backend aplikácie realizujte formou vlastného kódu a vlastnej inštalácie databázy. Výber použitej technológie, ako aj samotného prostredia (napr. aplikácia spustená priamo na vašom notebooku, Docker/Podman/Kubernetes (K3s, Kind, Minikube...), prostredie typu LAMPP/XAMPP, virtuálny stroj pomocou VirtualBox a i.) je na vás.

Backend odporúčame vyvíjať v jazyku JavaScript (Node.js/Deno/Bun), Python, PHP, Ruby, Go, Rust, Java, Kotlin, prípadne Swift. Ako databázu si môžete zvoliť MariaDB, MySQL, PostgreSQL alebo napríklad MongoDB či inú NoSQL technológiu. Ak máte záujem o inú databázu alebo technológiu backendu (Backendless, Firebase, atď.), podlieha schváleniu cvičiacim.

Aplikácia bude s backendom komunikovať prostredníctvom API endpointov (konkrétnych URL ktoré bude aplikácia volať prostredníctvom HTTP volaní). Backend musí mať implementovaných aspoň 5 volaní, pričom musí mať minimálne jedno volanie GET, PUT, POST a DELETE. Minimálne jedným volaním musí byť aplikácia schopná zapísať do databázy alebo do súborového systému backendu binárny obsah (obrázok, video, PDF, ...) a jedným volaním musí byť schopná aplikácia takýto obsah načítať. Nie je povolené zapisovať binárny obsah ako text v databáze (napr. base64).

Backend musí mať pripravené interaktívne prostredie umožňujúce jeho použitie (testovanie) aj bez Frontendu, napr. Postman, Insomnia, Hoppscotch, Swagger UI a pod.

## Dokumentácia k projektu

Súčasťou projektu je vypracovanie dokumentácie, ktorá definuje aký je účel aplikácie, akú funkcionality aplikácia pokrýva, prostredníctvom akých volaní komunikuje s backendom, aké dáta sa prenášajú. Súčasťou dokumentácie musí byť aj model databázy / štruktúra údajov a takisto opis všetkých integrácií s API zariadenia alebo API tretích strán. Dokumentácia API endpointov môže byť realizovaná formou ľubovoľného nástroja na správu API dokumentácie (PDF, HTML export, Swagger, Apiary, ...) . Ďalej sú súčasťou dokumentácie akceptačné testy pre frontend.

## Zdrojové kódy a rozdelenie práce

Každá dvojica si vytvorí dva GitHub repozitáre, jeden pre frontend, jeden pre backend. Každá dvojica si musí rovnomerne rozdeliť prácu aj na frontende aj na backende (rozdeliť si implementáciu endpointov, rozdeliť si implementáciu obrazoviek) a toto rozdelenie musí byť jasné z commitov v GitHub repozitári.

## Akceptačné testy

Každá dvojica navrhne 10 akceptačných testov, pričom

- 7 testov bude kladných (správne použite)
- 3 budú záporné (odolnosť voči nesprávnemu použitiu)

Každý akceptačný test frontendu musí pozostávať aspoň z 5 krokov, ktoré na seba nadväzujú a tvoria logický celok (dávajú zmysel, testujeme funkčnosť niečoho).

## Prezentovanie aplikácie

Pri prezentovaní je nevyhnutné aby aplikácia bežala na dvoch fyzických zariadení (vaše telefóny vo dvojici) medzi ktorými je vidno komunikáciu. Simulátor je možné použiť na prezentácie iných layoutov alebo funkcionality nepodporovaných fyzickými zariadeniami. Ak nie je možné prezentovať aplikáciu na dvoch zariadeniach, je potrebné si s cvičiacim dohodnúť prezentačný setup dostatočne dopredu.

# Základný systém bodovania projektu

## Milestone 1:

Návrh, wireframy, špecifikácia – 5 bodov

*akceptačné minimum 1 bod*

## Milestone 2:

Implementácia backendu (REST a DB) – 10 bodov

*akceptačné minimum 5 bodov*

## Milestone 3:

Splnenie povinných súčastí – 8 bodov

Splnenie troch povinne voliteľných úloh – 4 body za každú (max 12 bodov)

Celková kvalita implementácie frontendu – 5 bodov

Splnenie UAT testov – 5 bodov

Súlad dokumentácie s implementáciou – 5 bodov

*akceptačné minimum 15 bodov*

## Nepovinné úlohy / bonus:

Každá zo štyroch definovaných za max 5 bodov. Cvičiaci môže udeliť viac bodov za jednu, ak jej implementácia a kvalita spracovania výrazne prevyšuje očakávania. (*nie je nárokovateľné*)

Cvičiaci môže prideliť určitý počet bonusových bodov aj podľa vlastného uváženia za mimoriadne kvalitné spracovanie „niečoho navyše“ tak, že celkový počet bonusových bodov nikdy nepresiahne hodnotu 20.

Všetky body uvedené v bodovaní projektu sú maximálne bodové zisky, body prideľujú cvičiaci podľa vlastného uváženia podľa kvality spracovania daného celku (body sa udeľujú, nie strhávajú, bodové maximum získajú iba výborne vypracované projekty bez technických a návrhových problémov).

## Ukážka akceptačného testu

<b>Test 1: Vytvorenie nového článku</b>	
<b>Vstupné podmienky:</b>	Používateľ ktorý má rolu administrátora je prihlásený v aplikácii, aplikácia má pripojenie na Internet.
<b>Výstupné podmienky:</b>	V aplikácii pribudne článok, ktorý sa uloží do databázy a bude možné ho zobraziť.
<b>Postup:</b>	<ol style="list-style-type: none"><li>1. Používateľ stlačí tlačidlo MENU, ktoré sa vyroluje zľava</li><li>2. Používateľ stlačí tlačidlo NOVÝ ČLÁNOK</li><li>3. Zobrazí sa formulár na pridanie článku</li><li>4. Používateľ vyplní nadpis, text článku, a vyberie obrázok</li><li>5. Používateľ klikne na tlačidlo ULOŽIŤ</li><li>6. Zobrazí sa informácia o vytvorení článku</li><li>7. Po 3 sekundách je používateľ v aplikácii presmerovaný na nový článok</li></ol>
<b>Výsledok:</b> PASS / FAIL	

## Ukážky API dokumentácií:

- <https://openweathermap.org/current>
- <https://billdu.docs.apiary.io/#reference/documents/documents/list-all-documents>
- <https://www.zabbix.com/documentation/4.0/manual/api>