



ŽILINSKÁ UNIVERZITA V ŽILINE

**Fakulta riadenia
a informatiky**

Semestrálna práca z predmetu
vývoj aplikácií pre mobilné zariadenia

APLIKÁCIA NA QUIZ

Vypracoval: Ivan Kubovič

Študijná skupina: 5ZYI31

Akademický rok: 2024/2025

V Žiline dňa 8.4.2025



Obsah

Úvod	2
Prehľad podobných aplikácií	2
Analýza navrhovanej aplikácie	2
Návrh architektúry aplikácie	3
Návrh vzhľadu obrazoviek	4
Skutočný návrh riešenia problému	5
Krátka analýza	5
Návrh riešenia	5
MVVM a Compose	5
Pripojenie cez WebSockets	5
UI a UX	5
Popis implementácie	5
Zoznam zdrojov	7



Úvod

Táto mobilná aplikácia predstavuje kvízovú hru pre viacero hráčov, ktorá funguje po sieti v reálnom čase. Každý hráč sa pripája z vlastného zariadenia, odpovedá na otázky a vidí priebežné skóre ostatných. Aplikácia ponúka jednoduché, rýchle rozhranie a živý priebeh súťaže. Hlavnou motiváciou bolo prepojiť pôvodný webový koncept do dynamickej multiplayer aplikácie pre Android. Dôvodom, prečo som sa rozhodol vytvoriť túto aplikáciu je zlepšenie kvality kvízov organizovaných študentskou organizáciou UNIZA ESPORTS, ktorej som členom.

Prehľad podobných aplikácií

Pôvodne sa na tieto účely v organizáciách používali webové aplikácie ako Mentimeter, Slido a Kahoot.

Mentimeter je interaktívny prezentačný nástroj, ktorý umožňuje používateľom zapájať sa do prednášok, pomocou otázok, kvízov a hlasovaní.

Slido je nástroj určený na interaktívnu komunikáciu počas prezentácií, stretnutí a konferencií. Umožňuje publiku zapájať sa cez otázky, hlasovania a kvízy v reálnom čase.

Kahoot! je vzdelávacia platforma zameraná na interaktívne kvízy, ktoré môžu hráči hrať cez svoje zariadenia. Hrá sa v reálnom čase s bodovaním.

Analýza navrhovanej aplikácie

Navrhovaná mobilná aplikácia je určená pre viacerých hráčov, ktorí odpovedajú na otázky v reálnom čase prostredníctvom svojich mobilných zariadení. Aplikácia zabezpečuje zobrazenie otázok, spracovanie odpovedí a zobrazovanie priebežného skóre pre všetkých zúčastnených.

Funkcie (prípady použitia):

- Vytvorenie hernej miestnosti hosťiteľom
- Pripojenie hráčov k hre zadaním kódu miestnosti
- Zobrazenie otázky na všetkých zariadeniach naraz
- Odpovedanie hráča na otázku
- Vyhodnotenie správnosti odpovede
- Zobrazenie priebežného skóre všetkých hráčov v reálnom čase
- Zobrazenie výsledkov po skončení hry

Používateľské roly:

- Hostiteľ: vytvára hru a riadi jej spustenie
- Hráč: pripája sa do lobby, odpovedá na otázky a sleduje svoj aj cudzí postup

Mimo funkčné požiadavky:

- Reálny čas: dáta musia byť synchronizované takmer okamžite
- Zabezpečenie dát: odpovede hráčov by mali byť dočasne skryté, aby sa nedali kopírovať



- Škálovateľnosť: aplikácia by mala podporovať viacerých hráčov súčasne bez výrazného poklesu výkonu

Návrh architektúry aplikácie

Aplikácia je postavená na architektonickom vzore MVVM (Model-View-ViewModel), ktorý zabezpečuje oddelenie logiky, dát a používateľského rozhrania. Tento prístup uľahčuje testovanie a údržbu aplikácie.

Model reprezentuje:

- Dátové triedy ako Question, Player, GameSession
- Štruktúru otázok načítanú z JSON alebo synchronizovanú zo servera
- Aktuálne skóre a odpovede hráčov

ViewModel:

- Riadi priebeh hry (aktívna otázka, poradie hráčov)
- Spravuje stav odpovedí a skóre
- Komunikuje so serverom cez sieťové požiadavky (WebSocket)
- Aktualizuje UI

View:

- Zobrazuje otázky a možné odpovede
- Prijíma vstupy od hráča (výber možností alebo text)
- Dynamicky aktualizuje priebežné skóre hráčov
- Zobrazuje výsledky po skončení kvízu

Sieťová vrstva:

- Starostlivosť o synchronizáciu medzi hráčmi
- Pripojenie hráčov do jednej hry
- Priebežné odosielanie odpovedí a skóre na server
- Prijímanie živých aktualizácií o stave hry a skóre ostatných

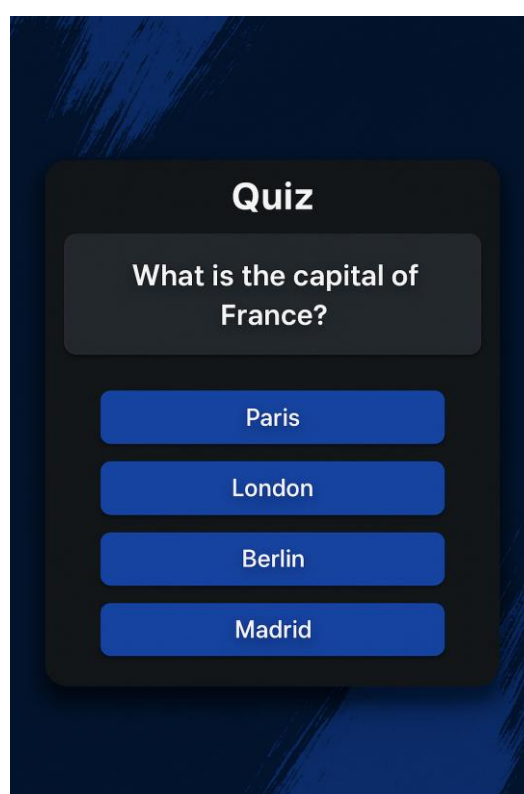
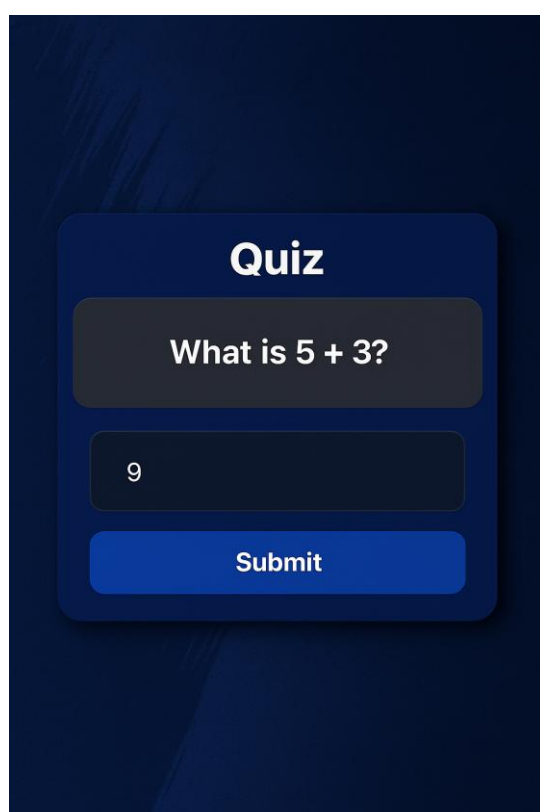
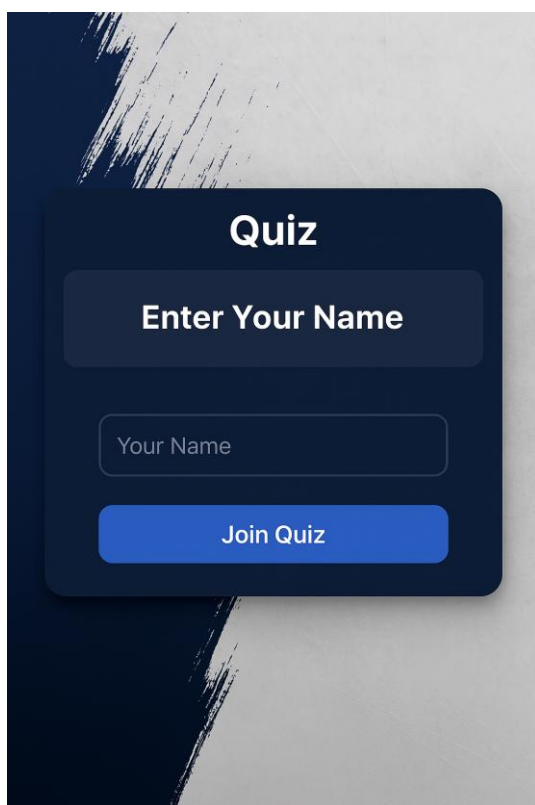
Dátový tok v aplikácii:

- Hráč sa pripojí do existujúcej hry alebo ju vytvorí
- Server rozpošle otázku všetkým hráčom naraz
- Hráči odpovedajú nezávisle
- Server vyhodnotí odpovede, aktualizuje skóre a rozpošle ho späť
- Aplikácia zobrazuje priebežné skóre až po konečný výsledok



Aplikácia nevyužíva databázu, všetky dáta sú uložené v pamäti alebo synchronizované cez sieť v reálnom čase.

Návrh vzhľadu obrazoviek





Skutočný návrh riešenia problému

Krátka analýza

Pri tvorbe tejto aplikácie som sa sústredil na jednoduché a rozšíriteľné riešenie, avšak keďže som sa sústredil na vytvorenie aplikácie priamo pripojenej k aktuálne vytvorenému serveru, bol som mierne limitovaný hlavne čo sa týka niektorých funkcionalít. Oproti pôvodnej analýze treba poznamenať, že vytvorenie hernej miestnosti sa vykonáva mimo mobilnej aplikácie. Pripojenie k miestnosti je riešené priamo bez zadávania kódu miestnosti, keďže server zatiaľ neposkytuje možnosť vytvorenia viacerých herných miestností. Zvyšne funkcionality spomenuté v pôvodnom návrhu sa mi úspešne podarilo vytvoriť. Používateľské role sú prevzaté priamo zo serveru, ktorý riadi celý priebeh kvízu a teda cez mobilnú aplikáciu je možné sa pripojiť iba ako hráč.

Návrh riešenia

MVVM a Compose

Návrh pre view model v mojej aplikácii spravuje stav a komunikáciu. Composable funkcie budú len zobrazovať dáta.

StateFlow spolu s collectAsState zabezpečujú reaktívnu synchronizáciu dát a UI bez manuálneho obnovovania.

Pripojenie cez WebSockets

V aplikácii som sa rozhodol implementovať Ktor Client pre trvalé spojenie so serverom a minimalizáciu latencie.

UI a UX

Pôvodne som pre vizuálnu stránku navrhol konzistentné s využitím Material3 karty avšak po tejto implementácii sa výsledok nezhodoval s návrhom. Preto som sa rozhodol navrhnúť pozadie vo forme obrázku, ktoré zodpovedalo pôvodnému návrhu.

Popis implementácie

Pri implementácii boli využité nasledovné bodované časti:

Otočenie obrazovky

- Všetky vstupné polia používajú rememberSaveable. Tým sa zabezpečí, že po otočení zariadenia nezmizne vložený text ani neprebehne reset obrazovky.
- Všetky funkcie čítajú údaje priamo z QuizViewModel, takže otočenie nevytvára duplicitné pripojenia ani nekontrolované stavy.



2. Obrazovky

- Entry Screen
Funkcia: Zadanie mena a inicializácia spojenia.
- Title Screen
Funkcia: Prechodové zobrazenie názvu kola.
- Waiting Screen
Funkcia: Čakanie na ďalší krok / ostatných hráčov.
- Choice Question Screen
Funkcia: Zobrazenie otázky s viacerými možnosťami a výber odpovede.
- Text Question Screen
Funkcia: Zobrazenie otázky a zber textovej odpovede od používateľa.
- Summary Screen
Funkcia: Zhrnutie výsledkov kola pre všetkých hráčov.

3. AndroidX komponenty

- ViewModel
Použitie: Spravuje celý stav aplikácie (meno hráča, aktuálny slide, odpovede, prepojenie so serverom).
- Lifecycle + StateFlow
Použitie: Reaktívna aktualizácia UI pri prichádzajúcich dátach.

4. Externé knižnice

- Ktor Client
Použitie: Komunikácia so serverom cez WebSockets
- Kotlinx.serialization (JSON)
Použitie: Konverzia dátových tried a surových JSON elementov na Slide objekty.

5. Použitie sieťovej komunikácie

- WebSockets
Použitie: Priama komunikácia so serverom.



Zoznam zdrojov

<https://www.slido.com/product>

<https://www.mentimeter.com/features>

<https://kahoot.com/business/features/how-it-works/>