

Žilinská univerzita v Žiline

Fakulta riadenia a informatiky

št. program Informatika a Riadenie



ŽILINSKÁ UNIVERZITA V ŽILINE

Fakulta riadenia
a informatiky

Kvíz pre lietadlové motory

SEMESTRÁLNA PRÁCA

Predmet: VAMZ

Streda – 7:00

šk. rok: 2023/2024

Vypracoval: Maroš Papan

št. skupina: 5ZYR32

Obsah

Úvod	3
Dostupné aplikácie podobného zamerania	3
Analýza aplikácie	4
Návrh architektúry aplikácie	5
Ukážka návrhu obrazoviek	7
Domovská obrazovka	7
Obrazovka kategórií.....	8
Obrazovka informačného listu	9
Popis implementácie.....	10
View model a obrazovky.....	10
Navigation – navigácia v aplikácii	10
Databáza Realm – implementácia Singleton	11
Štruktúra aplikácie.....	12
Notifikácie	12
Záver.....	13
Zoznam použitých zdrojov	14

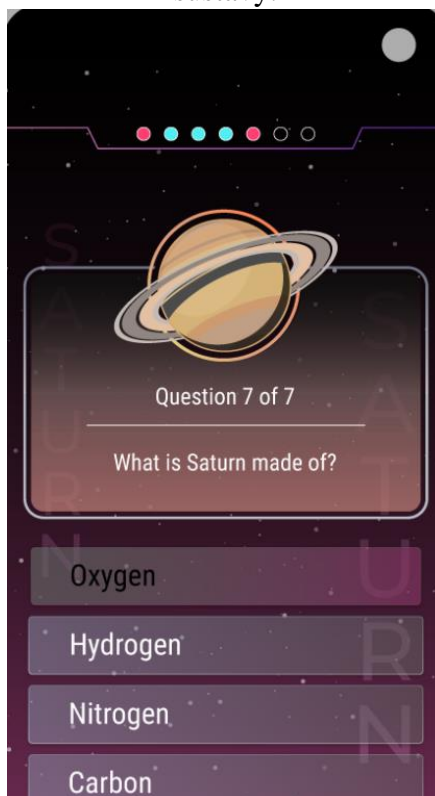
Úvod

Kvízová aplikácia je platforma, kde používateľ odpovedá na otázky z vybraného žánru, ktorý aplikácia ponúka. Aplikácia "Kvíz pre lietadlové motory" je zameraná špecificky na kvízy o lietadlových motoroch, konkrétne: turbovrtuľové motory, turbojet motory, turbofan motory a turboshaft motory. Používateľ si môže vybrať z týchto kategórií a otestovať svoje vedomosti o jednotlivých typoch lietadlových motorov. Po dokončení kvízu sa zobrazí výsledok v podobe pomeru správnych odpovedí k celkovému počtu otázok, čím poskytuje okamžitú spätnú väzbu o úspešnosti používateľa. Zodpovedané kvízy si môže používateľ uložiť, pričom záznamy sa ukladajú do internej databázy aplikácie. Tieto uložené záznamy sú prístupné v aplikácii pod sekciou "Záznamy kvízov", kde si ich používateľ môže kedykoľvek prezrieť. Táto funkcia je mimoriadne užitočná pre tých, ktorí si chcú sledovať svoj pokrok a analyzovať svoje vedomosti v priebehu času. S aplikáciou "Kvíz pre lietadlové motory" sa používatelia nielen zabavia, ale aj zlepšia svoje odborné znalosti o rôznych typoch lietadlových motorov.

Dostupné aplikácie podobného zamerania

V Obchode Play Store od spoločnosti Google sa nachádza množstvo aplikácií podobného charakteru, ktoré sa zaoberajú kvízmi z rôznych oblastí:

- Solar system quiz(Rapatundr): Aplikácia v ktorej odpovedáte na otázky z slnečnej sústavy.



- Knowledge quiz(Relity Games): Všeobecný vedomostný kvíz
- Quiz Planet(Lotum one GmbH): Aplikácia v ktorej si môžete vybrať kategóriu z vesmírnej teórie a riešiť kvíz z kategórie.

- Turbo: Car quiz(BOLD CAT): Kvíz o autách



- Flags and Capitals(VnS) – kvíz v ktorých si testujete svoju vedomosť z vlajok štátov a hlavných miest štátov.
- Quess the Logo Quiz(XoSelf) – kvíz v ktorom sa snažíte podľa obrázku uhádnuť logo firmy či auta.
- World Geography - Quiz game(Atoms Games Ent.) – Kvíz v ktorom sa podľa obrázkov snažíte uhádnuť určitý štát

Aplikácia kvíz pre lietadlové motory sa líši tým, že je špecificky zameraná len na lietadlové motory. Užívateľovi sa pri otvorení aplikácie neponúkne množstvo rôznych kategórií, z ktorých by si musel vybrať, ale okamžite uvidí kategórie týkajúce sa lietadlových motorov. Toto zjednodušenie robí aplikáciu užívateľsky príjemnejšou a umožňuje rýchlejší prístup k potrebným informáciám. Navyše, aplikácia obsahuje podrobné vysvetlenia a ilustrácie, ktoré pomáhajú lepšie pochopiť jednotlivé typy motorov.

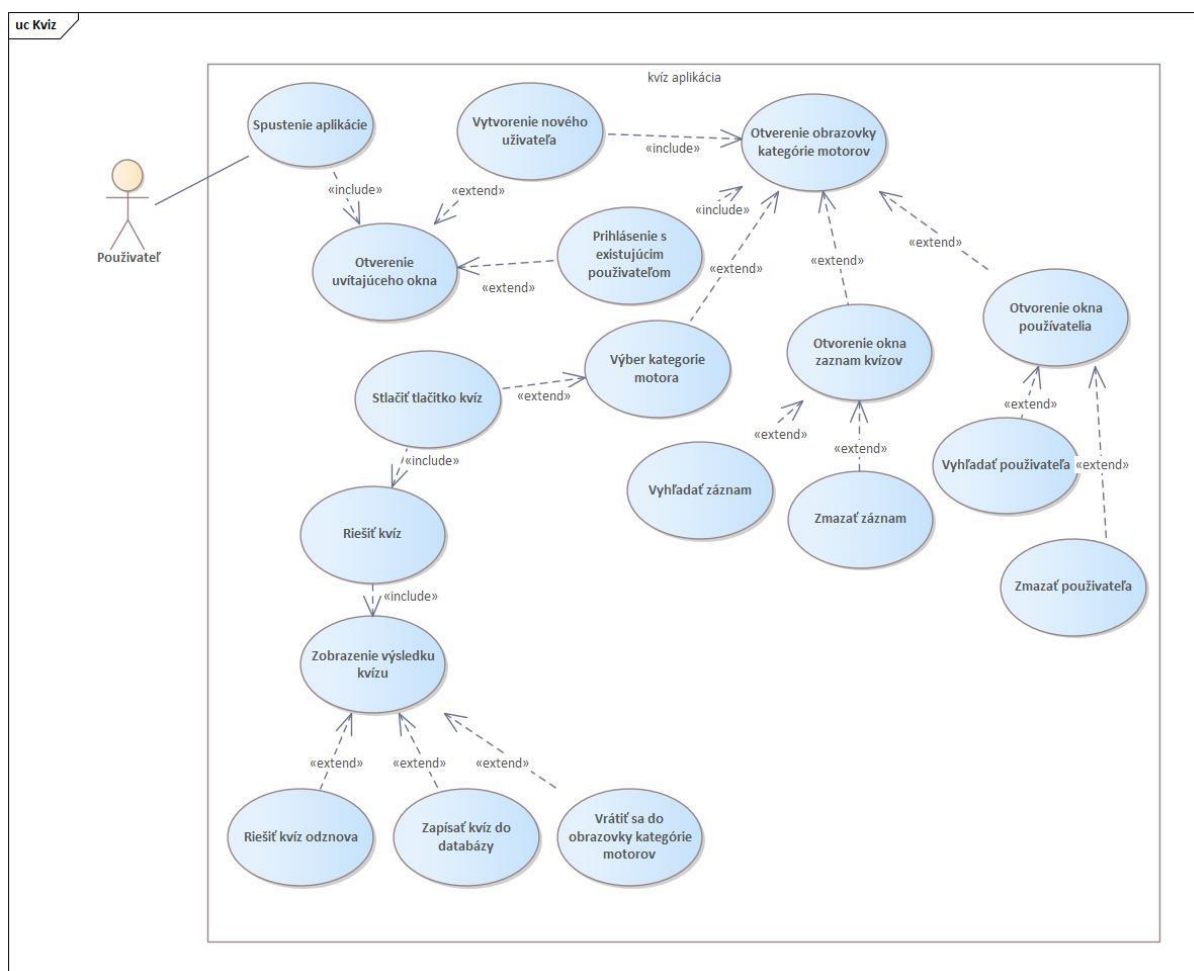
Analýza aplikácie

Aplikácia je určená pre otestovanie si svojich vedomostí o lietadlových motoroch. Kvíz o lietadlových motoroch sa bude môcť spustiť hocikedy pri používaní aplikácie. Po dokončení kvízu bude môcť užívateľ si pozrieť svoj výsledok na základe ktorého zistí svoje vedomosti o týchto motoroch.

Aplikácia bude taktiež ponúkať základne informácie o určitých lietadlových motoroch z ktorých budú pozostávať kvízové otázky, tak aby sa používateľ vedel 100% pripraviť na kvíz.

Aplikácia je dobrá pri tzv. skrátaní si času či už pri čakaní v dlhšom rade alebo pri cestovaní autobusom na dlhšie trasy, pre lietadlových nadšencov.

Taktiež bude aplikácia slúžiť na zlepšovanie si svojej pamäti schopnosti, kde používateľ bude využívať svoju rozumovú schopnosť uchovávaní a opätovného vybavovania informácií. Čo poskytuje pozitívnu odozvu hlavne u starších ľudí.



Návrh architektúry aplikácie

Prihlásenie užívateľa, zadanie mena pri otvorení aplikácie: Hneď po spustení aplikácie sa zobrazí okno so vstupom kde bude môcť zaregistrovať, prihlásiť sa užívateľ. Aby sa vedelo identifikovať kto robil daný kvíz.

Zoznam lietadlových motorov: Používateľ bude mať možnosť po spustení aplikácie a vyplnení si mena vybrať z kategórie motorov o aký má záujem a prečítať, naštudovať vedomosti o určitom motore. Kategórie budú postupne zoradené v hlavnom okne aplikácie. Tu bude mať používateľ aj možnosť sa prekliknúť na obrazovku všetkých zaregistrovaných používateľov alebo všetky kvízy doteraz spravené a zaregistrované používateľom.

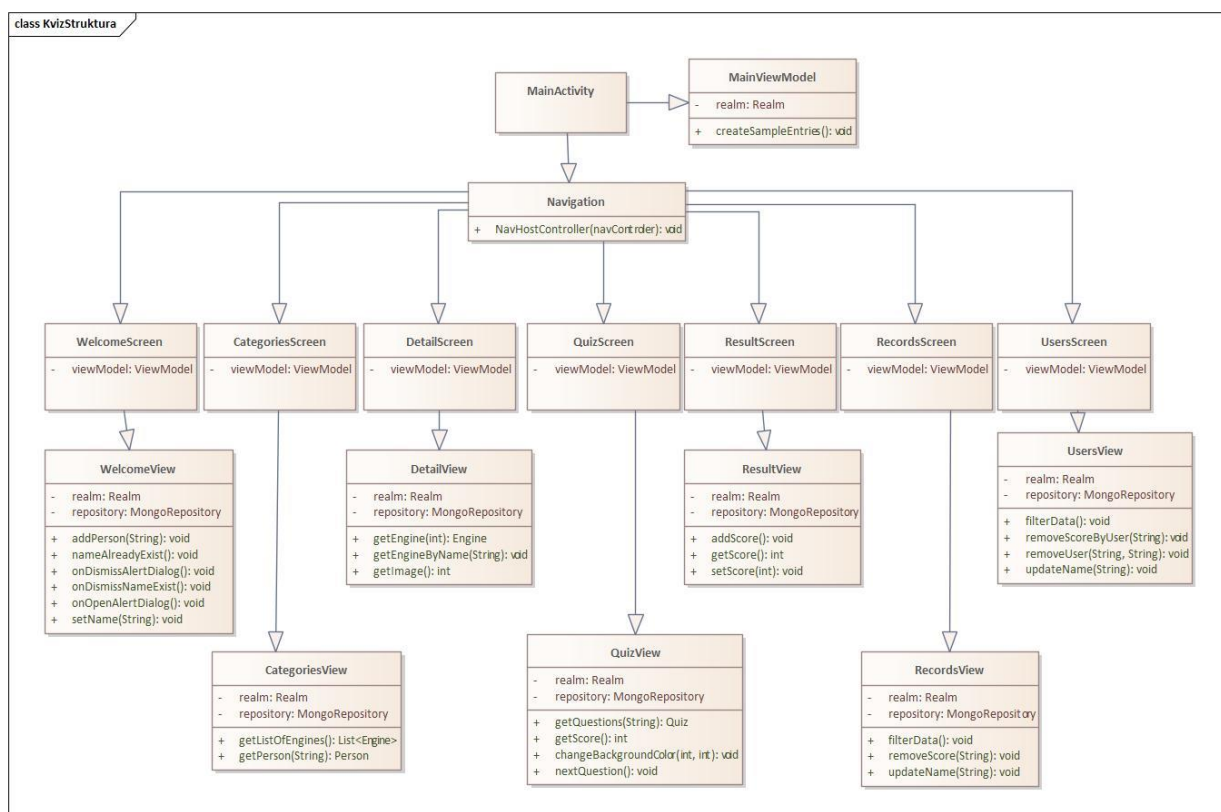
Riešenie kvízu: Po vybratí si kategórie bude mať používateľ možnosť si spraviť kvíz. V informačnom okne určitej kategórie bude tlačidlo, ktoré ho presunie do okna kvízu, kde bude užívateľ odpovedať na otázky. Každá otázka bude v novom okne.

Vyhodnotenie kvízu: Po zodpovedaní otázok v kvíze si bude môcť používateľ pozrieť svoj výsledok, ktorý sa zobrazí na novom okne v aplikácii po dokončení kvízu. Taktiež bude mať možnosť si tento kvíz zaznamenať do používanej databázy alebo bude mať možnosť vypracovať kvíz odznova alebo sa vrátiť na okno kategórií motorov.

Kvízové okno: Kvízové okno bude pozostávať z otázky z určitej kategórie a pod otázkou tri alebo štyri odpovede z ktorých bude len jedna správna. Po stlačení na určitú odpoveď sa otázka vyhodnotí či bola správna a prejde sa na ďalšiu otázku. Tento proces sa opakuje až po kým užívateľ nezodpovie na všetky otázky. Kde sa následne zobrazí okno s vyhodnotením kvízu.

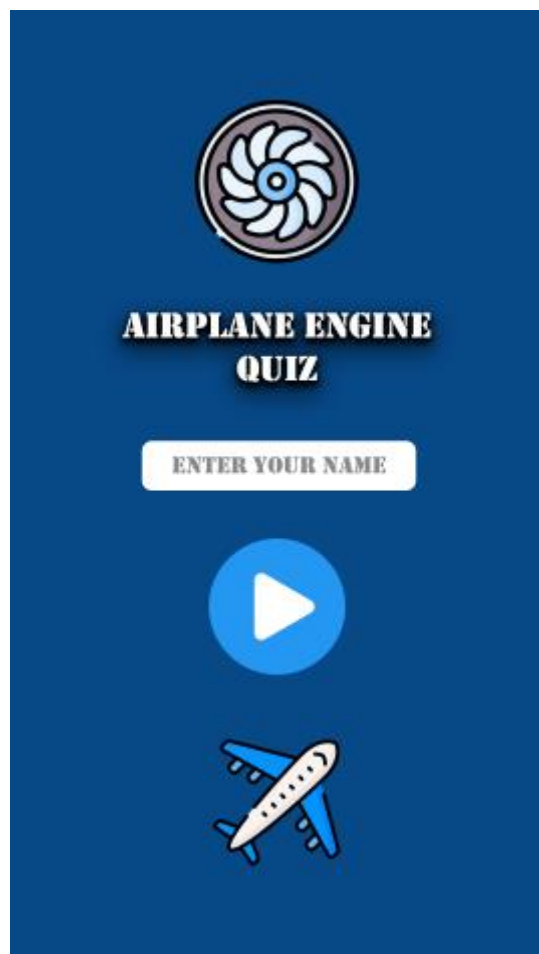
Okno všetci používatelia: Toto okno obsahuje záznam všetkých používateľov aplikácie. Používateľ môže filtrovať jednotlivých používateľov pomocou textového poľa do ktorého napíše akého meno používateľa, ktorého chce vyhľadať. Taktiež môže vymazať užívateľa.

Okno zaznamenaných kvízov: V tomto okne nájde užívateľ všetky zaznamenané kvízy, ktoré užívateľia tejto aplikácie robili. Nájde tu aký kvíz a ktorý užívateľ ho robil a jeho skóre.



Ukážka návrhu obrazoviek

Domovská obrazovka



Obrazovka kategorií



Obrazovka informačného listu

AIRPLANE ENGINES



TURBO FAN

ENGINE

BASIC INFORMATION

A turbofan engine has a large fan at the front, which sucks in air. Most of the air flows around the outside of the engine, making it quieter and giving more thrust at low speeds. Most of today's airliners are powered by turbofans. In a turbojet all the air entering the intake passes through the gas generator, which is composed of the compressor, combustion chamber, and turbine. In a turbofan engine only a portion of the incoming air goes into the combustion chamber. The remainder passes through a fan, or low-pressure compressor, and is ejected directly as a "cold" jet or mixed with the gas-generator exhaust to produce a "hot" jet. The objective of this sort of bypass system is to increase thrust without increasing fuel consumption. It achieves this by increasing the total air-mass flow and reducing the velocity within the same total energy supply.

Popis implementácie

Aplikácia kvíz lietadlových motorov bola tvorená pomocou určitých programátorských techník, ktoré zabezpečili rýchlu tvorbu a logicky jednoduchú implementáciu aplikácie. Použité programátorské techniky sú: View model, Navigation, implementovaná databáza pomocou singletonu, notifikácie, štruktúra rozdelenia do jednotlivých balíčkov v aplikácii, ktorá zabezpečila vytvorenie logickej štruktúry aplikácie.

View model a obrazovky

Každá obrazovka ma rozdelenú vnútornú logiku. A to tak, že pre každú obrazovku sa vytvoril jeden viewModel, ktorý spracováva logiku a dáta nasledovne zobrazíme na obrazovke. Všetky viewModely pre obrazovky sú uložené v jednom balíčku nazvanom views. A určitý viewModel sa potom volá v každej obrazovke. Obrazovky sú uložené v balíčku „ui“ v tomto balíčku sú ďalej vytvorené balíčky pre jednotlivé obrazovky. Napríklad obrazovka DetailScreen, ktorá zobrazuje detail jednotlivého zakliknutého motoru je uložená v balíčku „detail“, ktorý je uložený v balíčku „ui“.

Ukážka view modelu:

```
class DetailView: ViewModel() {  
    private val realm = DatabaseModule.provideRealm()  
    private val repository = DatabaseModule.provideMongoRepository(realm)  
  
    var typeOfEngine = mutableStateOf<String>{ value: "" }  
  
    fun getEngine1(engineID: ObjectId): Engine? {  
        return repository.getEngineById(engineID)  
    }  
}
```

Navigation – navigácia v aplikácii

Pre navigovanie v aplikácii sa použil nástroj Androidx Navigation. Celá navigácia je spracovaná v dvoch súboroch kde v súbore Screen.kt sú uložené jednotlivé cesty každej obrazovky ako objekty a v súbore Navigation.kt je spracovaná logika navigovania. Na implementovanie navigácie v aplikácii potrebujeme dva základne komponenty nástroja Navigation a to NavController, ktorým sa vytvorí navigačný ovládač, ktorý sleduje ktoré destinácie používateľ navštívil a umožňuje používateľovi sa pohybovať v aplikácii. Ďalší komponent v nástroji Navigation je NavHost, pomocou ktorého sa vytvorí navigačný graf aplikácie, ktorý NavController sleduje.

Ukážka navigácie:

Navigation.kt

```
@Composable
fun Navigation() {
    val navController = rememberNavController()

    NavHost(navController = navController, startDestination = Screen.WelcomeScreen.route) {
        composable(route = Screen.WelcomeScreen.route) {
            WelcomeScreen(navController = navController, modifier = Modifier)
        }
        composable(
            route = Screen.CategoriesScreen.route +("/{name}",
            arguments = listOf(
                navArgument(name = "name") {
                    type = NavType.StringType
                }
            )
        ) { this: AnimatedContentScope, entry ->
            val name = entry.arguments?.getString(key = "name").toString()

            CategoriesScreen(navController = navController, name = name)
        }
    }
}
```

Screen.kt

```
sealed class Screen(val route: String) {
    Maroš Papan
    object CategoriesScreen: Screen(route = "categories_screen")
    Maroš Papan
    object WelcomeScreen: Screen(route = "welcome_screen")
    Maroš Papan
    object DetailScreen: Screen(route = "detail_screen")
    Maroš Papan
    object QuizScreen: Screen(route = "quiz_screen")
}
```

Databáza Realm – implementácia Singleton

Realm databáza je vytvorená na podklade databázy MongoDB, to znamená, že dáta sú uložené vo forme jsonu ako dokument. Na vrchole hierarchie MongoDB je databáza, ktorá obsahuje niekoľko kolekcí a každá kolekcia uchováva niekoľko dokumentov v ktorých sú zaznamenané niekoľko záznamov. Tieto kolekcie sú podobné tabuľkám ktoré sú v relačnej databáze a dokumenty sú ako záznamy alebo riadky v tabuľke. V aplikácii sa Realm databáza implementovala pomocou návrhového vzoru singleton, ktorý zabezpečí volanie jednej inštancie z rôznych súborov, metód.

Pre implementáciu Realm databázy bolo potrebné vytvoriť tri základne veci a to model databázy, repozitár databázy a databázový modul, ktorý nám vytvára alebo poskytuje prístup k databáze.

Model databázy:

Skladá sa zo piatich objektov a to: Engine, Person, Question, Quiz, Score.

Engine(Motor) – je objekt realm databázy, ktorý slúži na ukladanie záznamu motorov, obsahuje stĺpce ako typ motoru, id motoru, popis motoru, obrázok motoru.

Person(Užívateľ) – je objekt realm databázy, ktorý slúži na ukladanie užívateľov aplikácie. Obsahuje stĺpce typu id človeka, meno, priezvisko, ktoré je nastavené na null takže ho nemusíme vyplniť pri ukladaní záznamu.

Question(Otázka) – je objekt realm databázy, ktorý slúži na ukladanie otázok pre quiz. Obsahuje stĺpce typu otázka, možnosť jedno, možnosť dva, možnosť tri, správna odpoveď, obrázok, referencia na motor.

Quiz(Kvíz) – je objekt realm databázy, ktorý slúži na ukladanie jednotlivých kvízov. Obsahuje stĺpce typu id kvízu, otázky referencia na question, typ kvízu.

Score(Skóre) – je objekt realm databázy, ktorý slúži na ukladanie skóre pri ukončení kvízu užívateľom. Obsahuje stĺpce typu id, skóre, person referencia na človeka, quiz referencia na kvíz.

Repozitár databázy

Tvorený dvoma súbormi prvý interface, kde sú určené aké funkcie bude vykonávať repozitár a druhý súbor, ktorý obsahuje implementáciu funkcií. Funkcie typu zapisovania a čítania dát z databázy.

Databázový modul

Modul, ktorý vytvára alebo poskytuje databázu použitý návrhový vzor singleton tak aby vždy poskytol jednu inštanciu databázy v celej aplikácii.

Štruktúra aplikácie

Aplikácia je rozdelená do jednotlivých logických balíčkov, ktoré umožňujú jednoduchú orientáciu v zdrojovom kóde. Konkrétne je aplikácia rozdelená do základných troch balíčkov: ui balíček, views balíček, domain balíček.

Ui balíček - Tento balíček obsahuje súbory, ktoré sa týkajú dizajnu aplikácie. Zodpovedá za to ako sa aplikácia vizuálne reprezentuje. V tomto balíčku sa taktiež nachádza balíček components(komponenty). Kde máme uložené komponenty ktoré viac krát požívame v jednotlivých obrazovkách.

Views balíček – spracováva logiku aplikácie. Zodpovedá za spracovanie vstupov od užívateľa, komunikáciu s databázou a následne zobrazenie výsledkov užívateľovi.

Domain balíček - Balíček domain je zodpovedný za nastavenie databázy. Obsahuje súbory, ktoré definujú štruktúru databázy, vrátane tabuliek, vzťahov medzi nimi a pravidiel pre manipuláciu s dátami.

Notifikácie

V aplikácii lietadlové motory notifikácie zabezpečujú plynulosť aplikácie a to tým, že sa zobrazia vždy pri nevynútenej chybe užívateľa a to napr. pri prihlasovaní ak užívateľ zabudol zadať meno tak aplikácia vyhodí notifikáciu, ktorá ho upozorní na nevyplnené textové pole. V aplikácii sa vyskytujú štyri druhy týchto notifikácií a to sú : nevyplnené

textové pole pri prihlasovaní, vymazanie užívateľa, ktorý je práve prihlásený, zapisovanie do databázy, neo značenie odpovede pri riešení kvízu.

Nevypísané textové pole pri prihlasovaní

Ak užívateľ zabudne pri prihlasovaní napísať užívateľské meno zobrazí sa notifikácia aby toto textové pole vyplnil.

Vymazanie užívateľa, ktorý je práve prihlásený

Ak sa užívateľ v okne all_users pokúsi vymazať samého seba vyhodí mu to notifikáciu, že sa účet, ktorý sa snaží vymazať práve používa.

Zapisovanie do databázy

Po dokončení kvízu je v aplikácii možnosť si zapísať skóre do databázy a tak zaznamenať svoje pokroky. Pri zapisovaní skóre vyhodí notifikáciu či sa skóre úspešne zapísalo do databázy.

Neoznačenie odpovede pri riešení kvízu

Pri riešení kvízu, ak užívateľ neo značí odpoveď a chce prejsť na ďalšiu otázku, vyhodí mu to notifikáciu aby najprv zodpovedal na konkrétnu otázku a nepustí ho na ďalšiu otázku.

Záver

Na záver tejto semestrálnej práce by som rád zhrnul hlavné aspekty a výsledky tvorby aplikácie kvíz pre lietadlové motory, ktorá bola vytvorená v Android Studio pomocou Jetpack Compose a jazyka Kotlin. Táto aplikácia demonštruje efektívne využitie moderných technológií a postupov v oblasti vývoja mobilných aplikácií.

Aplikácia bola navrhnutá s cieľom poskytovať používateľom interaktívny a vzdelávací zážitok, pričom jej implementácia zahŕňala niekoľko kľúčových technológií a architektonických vzorov:

- **Androidx Navigation:** Tento nástroj umožnil jednoduchú a intuitívnu navigáciu medzi obrazovkami, čo prispelo k plynulému a používateľsky prívetivému rozhraniu.
- **ViewModel pre každú obrazovku:** Použitie ViewModelu zabezpečilo efektívne riadenie dát a stavov aplikácie, čím sa zlepšila odozva a stabilita jednotlivých obrazoviek.
- **Implementácia databázy pomocou singletonu a repozitára:** Táto architektúra umožnila centralizované a efektívne spravovanie dát, pričom singletonový vzor zabezpečil, že databázové pripojenie bolo bezpečne a efektívne zdieľané v rámci celej aplikácie. Repozitár sprostredkoval prístup k databázovým operáciám, čo zjednodušilo ich údržbu a rozširiteľnosť.
- **Štruktúra aplikácie rozdelená do logických balíčkov:** Dôsledné rozdelenie kódu do logických balíčkov podporilo prehľadnosť a modularitu aplikácie, čo uľahčilo jej údržbu a budúce rozširovanie.
- **Notifikácie na zabezpečenie plynulosti aplikácie:** Implementácia notifikácií poskytla používateľom dôležité informácie a upozornenia, čím sa zlepšila celková používateľská skúsenosť a interakcia s aplikáciou.

Táto práca ukazuje, ako je možné pomocou Jetpack Compose a Kotlinu vytvoriť aplikáciu, ktorá je nielen vizuálne atraktívna, ale aj technologicky pokročilá a spoľahlivá.

Zoznam použitých zdrojov

Realm databáza použitý tutorial: <https://www.youtube.com/watch?v=7ukzzjtUCJA> – Full Guide to Mongo DB Realm as Local Database for Android(Kotlin SDK)