

ŽILINSKÁ UNIVERZITA
V ŽILINE

Fakulta riadenia a informatiky

Geografický quiz

Semestrálna práca

Marko Krstič

Študijný program: Informatika a Riadenie

Študijný odbor: Informatika

Školiace pracovisko: Žilinská univerzita v Žiline,

Cvičiaci: doc. Ing. Patrik Hrkút, PhD. a Ing. Michal Ďuračík, PhD.

Žilina 2024

Obsah

1	Popis a analýzy riešeného problému	3
1.1	Špecifikácia zadania, definovanie problému	3
1.2	Podobné aplikácie	3
1.2.1	QuizLand.....	3
1.2.2	GeoGuessr.....	4
1.2.3	Europe Geography.....	5
1.2.4	Moja aplikácia	6
2	Návrh a riešenie problému.....	7
2.1	Návrh aplikácie	7
2.2	Krátka analýza.....	7
3	Popis implementácie	9
	Záver.....	14

1 POPIS A ANALÝZY RIEŠENÉHO PROBLÉMU

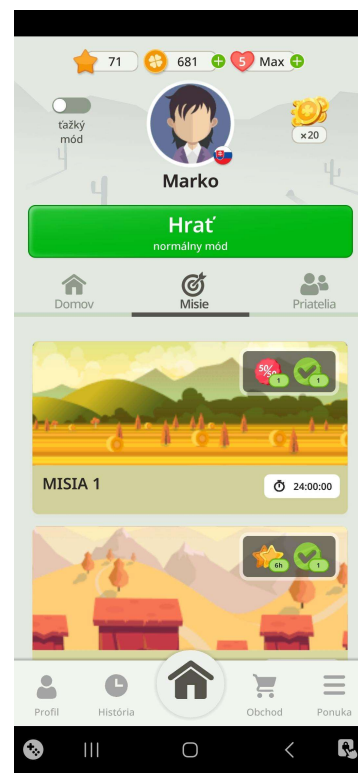
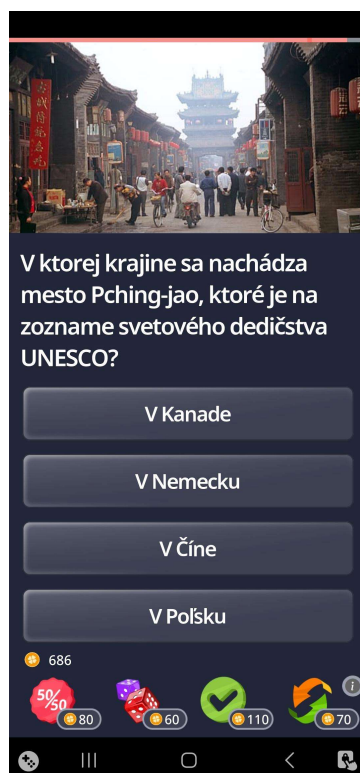
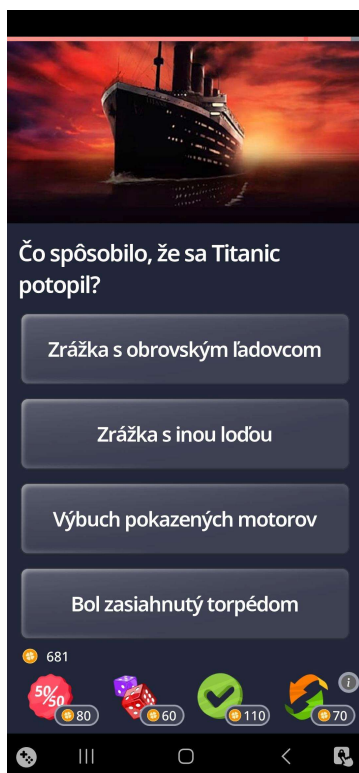
1.1 Špecifikácia zadania, definovanie problému

V dnešnej dobe sa zdá, že je čoraz viac zabúdané na dôležitosť všeobecného rozhľadu, ktorý zahŕňa aj geografické znalosti o Európe. Moja aplikácia sa snaží adresovať túto problematiku a poskytovať užívateľom príležitosť zlepšovať alebo overovať svoje poznatky v tomto smere. S cieľom posilniť vzdelávanie a zvýšiť povedomie o európskej geografii, táto aplikácia ponúka prostriedky a nástroje na interaktívne a zábavné učenie.

1.2 Podobné aplikácie

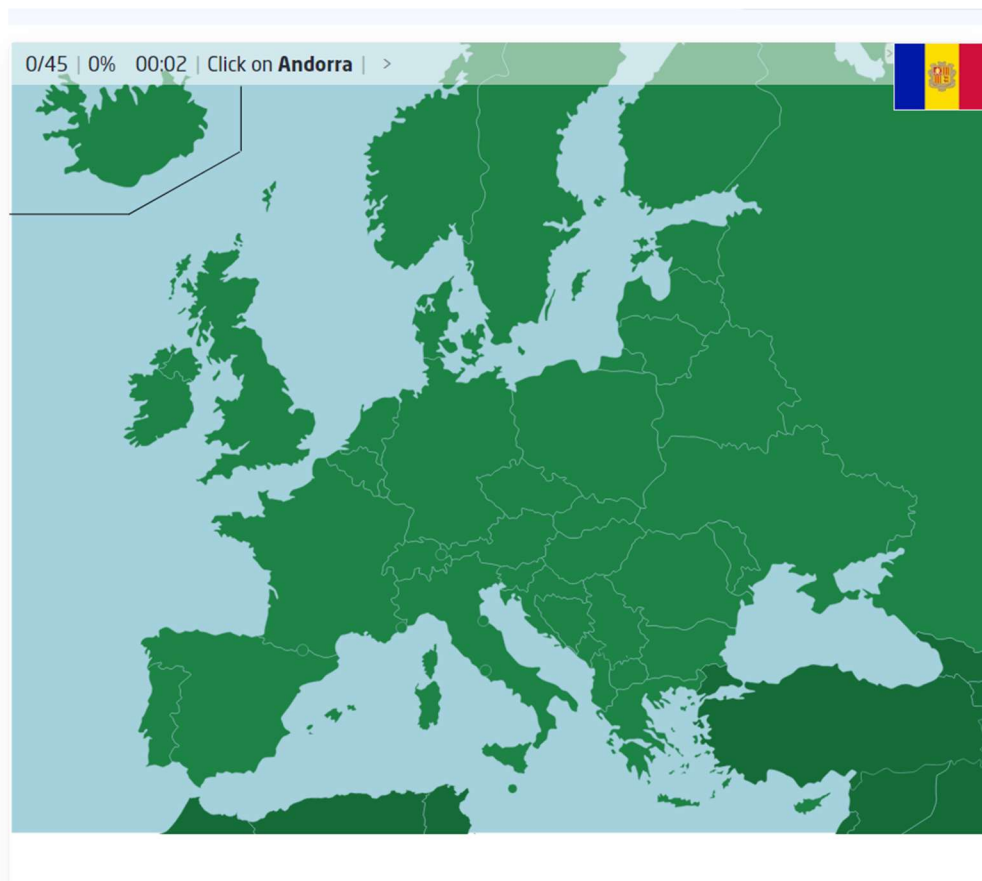
1.2.1 QuizLand

QuizzLand je Trivia hra, ktorá ponúka mnoho otázok, ktorou sú súčasťou aj otázky o geografie Európy. Aplikácia má jednoduchý dizajn a viacjazyčnú podporu. Hráči získavajú mince za správne odpovede a súťažia o ceny v podobe hviezdíček, ktoré určujú umiestnenie v rebríčkoch. Umožňuje hrať samostatne alebo s priateľmi, s interaktívnymi pomocníkmi.



1.2.2 GeoGuessr

Tento obrázok pochádza zo stránky GeoGuessr, ktorá slúži na overovanie geografických znalostí a lokalizáciu štátov v Európe. Práve z tejto stránky vzišla inšpirácia k vytvoreniu tejto aplikácie, keď mi kamarát ukázal jej koncept. Svedomie o svojej nedostatočnej geografickej znalosti ma motivovalo k zlepšeniu metódou vytvorenia aplikácie na overovanie geografických znalostí.



1.2.3 Europe Geography

Europe Geography je trivia hra zameraná čisto na geografické poznatky o Európe. Musíte v nej odpovedať na otázky, v ktorých identifikujete krajiny, ich hlavné mestá a symboly. Táto hra vám umožní rozšíriť svoje znalosti. Ponúka používateľovi viac ako 6000 otázok, 5000 obrázkov na identifikáciu alebo encyklopédiu. Avšak množstvo informácií, tlačidiel a výber vytvára neprehľadné prostredie.



1.2.4 Moja aplikácia

Moja mobilná aplikácia, zameraná na geografiu Európy, ponúka interaktívnych kvíz, ktorý pomáha užívateľom hlbšie sa vzdelávať ohľadom geografie Európy . Ako tvorca tejto aplikácie som sa tiež chcel osobne pri vzdelávať v tejto oblasti a zároveň poskytnúť ostatným užívateľom príležitosť robiť to isté. S jednoduchým a intuitívnym dizajnom je prístup k informáciám a otázkam ľahký a prehľadný. Moja aplikácia taktiež ako aplikácia Europe Geography ponúka štatistiky quizov.



2 NÁVRH A RIEŠENIE PROBLÉMU

2.1 Návrh aplikácie

Popis tried a ich vzťahov:

MainActivity: Hlavná aktivita aplikácie, ktorá inicializuje obsah a spravuje navigáciu medzi obrazovkami. Obsahuje metódy na vytvorenie notifikačného kanála a žiadosť o povolenie na notifikácie.

Notifikacia: Pomocná trieda pre notifikácie. Obsahuje metódu na vytvorenie notifikačného kanála.

HI_Menu: Composable funkcia, ktorá slúži na zobrazenie hlavného menu a slúži na hlavné navigovanie medzi obrazovkami

Nastavenia: Composable funkcia, ktorá slúži na nastavovanie údajov hráča a spravovanie databázy

Statistiky: Composable funkcia, ktorá slúži na zobrazenie štatistík hráča uložené v databáze

Quiz_Screen: Composable funkcia, ktorá slúži na zobrazovanie otázok a zberanie správnych alebo nesprávnych odpovedí hráča

Encyklopedia: Composable funkcia, ktorá slúži na vzdelávanie sa hráča ohľadom geografie európy

NacitavanieOtazokView: ViewModel slúžiaci na načítanie údajov z Json filu do otázok ktoré následne zobrazuje quiz_screen

2.2 Krátka analýza

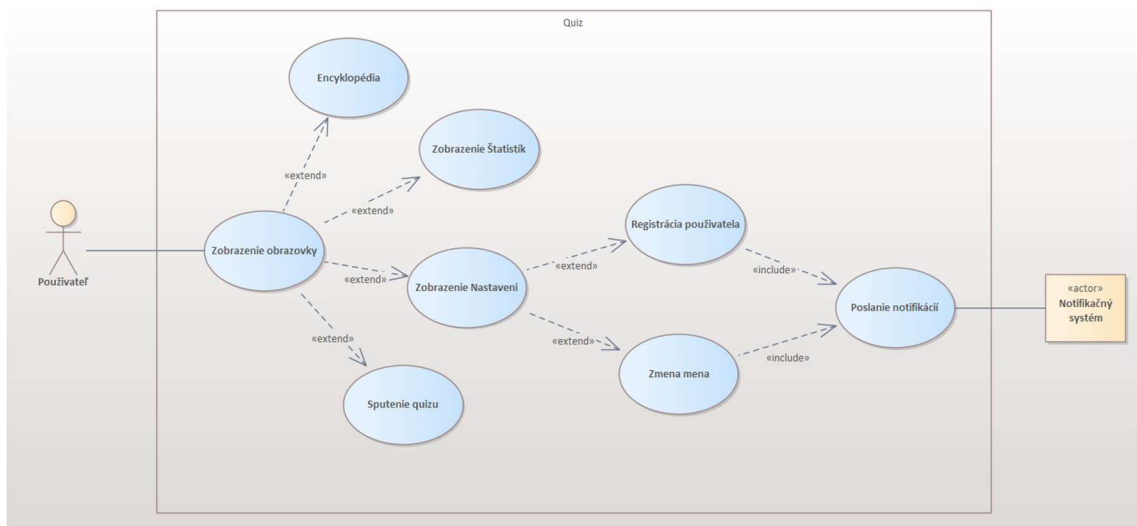
UseCase diagram zobrazuje hlavné funkcionality aplikácie mojej aplikácie a interakcie medzi používateľom a systémom. Diagram sa zameriava na tieto úlohy:

Registrácia používateľa v nastaveniach: Používateľ sa zaregistruje v systéme databázy cez nastavenia kde si môže zmeniť meno. Pri oboch týchto akciách príde používateľovi notifikácia o zmene v databáze.

Encyklopédia: Používateľ sa môže vzdelávať v oblasti geografie Európy.

Quiz: Používateľ spustí quiz počas ktorého odpovedá na otázky a ukladajú sa jeho správne a nesprávne odpovede do databázy.

Štatistiky: Používateľ si vie pozrieť svoj celkový počet otázok na ktoré odpovedal a počet otázok na ktoré odpovedal správne.



3 POPIS IMPLEMENTÁCIE

Reakcia na otočenie displeja: Aplikácia je navrhnutá tak , aby správne reagovala na otočenie displeja z portrait módu do landscape módu a naopak. Uchovávanie údajov v textových poliach zabezpečuje rememberSaveable. Taktiež používam LazyColumn aby som zabezpečil scrollovanie v prípade kedy je obrazovka nedostatočne veľká po otočení displeja.

```

└─ AirToggle *
@Composable
fun Nastavenia(navController: NavController, uzivatel: UzivatelData = UzivatelData()) {
    val configuration = LocalConfiguration.current
    val currentFocus = LocalFocusManager.current
    val isPortrait = configuration.orientation == android.content.res.Configuration.ORIENTATION_PORTRAIT

    val contextt = LocalContext.current
    var text by rememberSaveable { mutableStateOf(uzivatel.name) }

    Box(modifier = Modifier.fillMaxSize()) { this: BoxScope
        Image(
            painter = painterResource(id = R.drawable.pozadie2),
            contentDescription = "pozadie",
            contentScale = ContentScale.FillBounds,
            modifier = Modifier.matchParentSize()
        )
    }
}

```

Resources: Všetky obrázky a farby sú uložené v projekte pod zložkou resources. Fotky sú v priečinku res/Drawable.

Obrazovky: Aplikácia obsahuje viacero unikátnych obrazoviek, ktoré riešia rôzne funkcie aplikácie:

1. Hlavné menu:

HL_menu slúži na navigáciu medzi screenami ako Quiz, štatistiky, encyklopédia, nastavenia a button na ukočenie aplikácie

```

└─ AirToggle
@Composable
fun HL_menu(navController: NavController, uzivatel : UzivatelData = UzivatelData()) {
    val context = LocalContext.current

    Box(modifier = Modifier.fillMaxSize()) { this: BoxScope
        Image(
            painter = painterResource(id = R.drawable.pozadie2),
            contentDescription = "pozadie",
            contentScale = ContentScale.FillBounds,
            modifier = Modifier.matchParentSize()
        )
    }

    LazyColumn(
        modifier = Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.SpaceEvenly,
        horizontalAlignment = Alignment.CenterHorizontally
    ) { this: LazyListScope
        item { this: LazyItemScope

```

2. Quiz screen:

Slúži na spravovanie otázok načítaných z Json filu a následne ukladanie štatistík do databázy užívateľa.

```
@Composable
fun Quiz_Screen(navController: NavController, uzivatel: UzivatelData = UzivatelData()) {
    val context = LocalContext.current
    val viewModel: Quiz_ScreenView = viewModel()
    val numbers: MutableList<O tazka> = mutableListOf()

    LaunchedEffect(Unit) { this: CoroutineScope
        viewModel.loadQuestions(context)
    }

    val questions = viewModel.questions.value
    val currentQuestionIndex by viewModel.currentQuestionIndex
    val currentQuestion = questions.getOrNull(currentQuestionIndex)
    var help : Boolean = false
    val configuration = LocalConfiguration.current
    val isPortrait =
        configuration.orientation == android.content.res.Configuration.ORIENTATION_PORTRAIT
```

3. Štatistiky:

Slúži na zobrazenie štatistík užívateľa z databázy.

```

AirToggle
@Composable
fun Statistiky(navController: NavController, uzivatel: UzivatelData = UzivatelData()) {
    Box(modifier = Modifier.fillMaxSize(), contentAlignment = Alignment.TopCenter) { this: BoxScope
        Image(
            painter = painterResource(id = R.drawable.pozadie2),
            contentDescription = "pozadie",
            contentScale = ContentScale.FillBounds,
            modifier = Modifier.matchParentSize()
        )

        Text(
            text = " Štatistiky ",
            fontSize = 32.sp,
            color = Color.White,
            modifier = Modifier
                align(Alignment.Center)
                background(Color.DarkGray)
                padding(top = 16.dp)
        )
    }
}
```

4. Encyklopédia:

Používateľ si vie v tejto obrazovke zistiť dodatočné otázky ohľadom rôznych krajín v Európe ktoré sa môžu vyskytovať v quizoch. Táto trieda má vlastný viewModel a dataclass.

```

└─ AirToggle
@Composable
fun Encyklopedia(navController: NavController) {
    val viewModel: NacitavanieObsahuView = viewModel()
    val context = LocalContext.current
    var obsahy by remember { mutableStateOf<List<ObsahEncyklopedie>?>() (value: null) }
    LaunchedEffect(Unit) { this: CoroutineScope
        obsahy = viewModel.NacitajObsahy(context)
    }
}

```

5. Nastavenia:

Tento screen slúži na zmenu mena užívateľa a vloženie záznamu užívateľa do databázy alebo prípadne vymazanie databázy.

```

}
LazyColumn(
    modifier = Modifier.fillMaxSize(),
    verticalArrangement = Arrangement.SpaceEvenly,
    horizontalAlignment = Alignment.CenterHorizontally
) { this: LazyListScope
    item { this: LazyItemScope
        Text(
            text = "Nastavenia",
            fontSize = 24.sp,
            color = Color.White,
            modifier = Modifier.background(Color.Blue)
        )
    }
    item { this: LazyItemScope
        TextField(
            value = text,
            onValueChange = { newText ->
                val filteredText = newText.filter { it.isLetter() || it.isWhitespace() }
                text = filteredText
            },
            label = { Text(text: "Zmena mena") },
            keyboardOptions = KeyboardOptions.Default.copy(imeAction = ImeAction.Done),
            keyboardActions = KeyboardActions(onDone = { this: KeyboardActionScope

```

Využitie AndroidX komponentov:

ViewModel: použitý pre správu logiky obrazoviek napr. Quiz_ScreenView, NacitavanieObsahuView, nacitavanieOtazokView

Navigation: Použitý na navigáciu medzi obrazovkami aplikácie.

Notifikácie: táto trieda mi vytvorí notifikačný kanál do ktorého posielam notifikácie ohľadom meny užívateľa

```
1 AirToggle
2 fun createNotificationChannel(context: Context) {
3     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
4         val name = "Notifikácie"
5         val descriptionText = "Čo ja viem"
6         val importance = NotificationManager.IMPORTANCE_DEFAULT
7         val channel = NotificationChannel(id: "notif", name, importance).apply { this: NotificationChannel
8             description = descriptionText
9         }
10        val notificationManager = ContextCompat.getSystemService(context, NotificationManager::class.java)
11        notificationManager?.createNotificationChannel(channel)
12    }
13 }
14 }
```

Room databáza: mám vytvorenú databázu k nej interface a triedu do ktorej sa ukladajú jednotlivé riadky tabuľky

```
8
9
10 AirToggle
11 @Entity(tableName = "uzivatelData")
12 @Serializable
13 data class UzivatelData(
14     @PrimaryKey(autoGenerate = true) val id: Int = 0, // Primárny kľúč pre databázový záznam
15     @ColumnInfo(name = "name") var name: String = "Užívateľ",
16     @ColumnInfo(name = "pocetOtazok") var pocetOtazok: Int = 0,
17     @ColumnInfo(name = "pocetSpravnychOtazok") var pocetSpravnychOtazok: Int = 0,
18     @ColumnInfo(name = "celkovyCas") var celkovyCas: Int = 0
19 ) {
20
21     AirToggle
22     suspend fun uloziUdaje(context: Context) {
23         UzivatelDB.getInstance(context).uzivatelDao().insert(uzivatel: this)
24     }
25 }
```

Práca s Git serverom:

Aplikácia bola ukladaná na môj Git, s pravidelnými commitmi dokumentujúcimi pokrok vývoju aplikácie.

Finalna aplikacia2	AirToggle	Today 16:20
Finalna aplikacia	AirToggle	Today 16:00
2.Oprava Databázy	AirToggle	Today 12:59
1.Encyklopédia screen + view + json	AirToggle	Today 11:48
1.ŠtatistikaScreen a oprava databázy	AirToggle	Yesterday 23:38
1.Logika otázok + užívateľa	AirToggle	Yesterday 18:39
Merge remote-tracking branch 'origin/master'	AirToggle	9. 6. 2024 22:27
2. Notifikacia_nastavovanie_dat_uzivatela	AirToggle	9. 6. 2024 19:39
Create Lorem Ipsum	Marko Krstić	9. 6. 2024 20:00
1. Notifikacia_nastavovanie_dat_uzivatela	AirToggle	9. 6. 2024 19:39
1. Nastavovanie_dat_uzivatela	AirToggle	9. 6. 2024 17:43
1. tvorenie_dat_uzivatela	AirToggle	5. 6. 2024 23:38
3. Tvorenie_otazok	AirToggle	5. 6. 2024 15:37
2. Tvorenie_otazok	AirToggle	5. 6. 2024 14:02
1. Tvorenie_otazok	AirToggle	4. 6. 2024 17:23
1. Prechody_Obrazoviek	AirToggle	3. 6. 2024 21:25
1. uvodnáobraz_spravne	AirToggle	3. 6. 2024 16:06
1. commit	AirToggle	31. 5. 2024 20:02
1. commit	AirToggle	31. 5. 2024 19:55
1. commit	AirToggle	29. 4. 2024 9:09
1. commit	AirToggle	29. 4. 2024 9:09

Iné použité knižnice ako AndroidX:

implementation("com.google.code.gson:gson:2.10.1")

Použité zdroje mimo Android developer dokumentácie a kurzu predmetu:

<https://medium.com/@anandgaur22/expandable-card-with-animation-in-jetpack-compose-70849c33a391>

ZÁVER

Moja mobilná aplikácia, zameraná na geografiu Európy, pomáha používateľom hlbšie sa vzdelávať v tejto oblasti. Ako tvorca tejto aplikácie som sa chcel osobne vzdelávať v geografii a zároveň poskytnúť ostatným používateľom príležitosť robiť to isté. S jednoduchým a intuitívnym dizajnom je prístup k informáciám a otázkam ľahký a prehľadný. Cieľom aplikácie je motivovať používateľov k zlepšovaniu ich geografických vedomostí a poskytnúť im nástroje na dosiahnutie ich cieľov, čo prispieva k ich osobnému rastu a rozvoju. Bohužiaľ som nedosiahol úroveň aplikácie, akú som zamýšľal v pôvodnej dokumentácii, ale tento projekt mi dal dobrý základ na programovanie v Kotlině s použitím Jetpack Compose a zároveň som si zopakoval geografiu Európy.