

Semestrálna práca z predmetu vývoj aplikácií pre mobilné zariadenia

vypracoval: Andrej Ondrášek  
študijná skupina: 5ZYI24

Popis a analýza riešeného problému

Špecifikácia zadania, definovanie problému

Aplikácia na tvorbu a kvízov, ktorá umožňuje užívateľom vytvárať, upravovať, zdieľať a hrať kvízy.

Podobné aplikácie

Kahoot – Pri tvorbe otázky mám moja aplikácia iba možnosť klasických odpovedí narozdieľ od kahootu, ktorý má veľa ďalších možností ale narozdieľ od kahootu môže mať jedna otázka neobmedzene veľa odpovedí .

Kahoot nastavuje čas osobitne otázkam a v mojej aplikácií sa nastavuje všeobecne celému kvízu.

Jednoduchosť dizajnu mojej aplikáciu ju robí výrazne viac prehľadnú ako Kahoot s jeho hromadou reklám a možností, ktoré aj tak môžu využívať iba platený užívatelia.

Quizziz – V mojej aplikácií sa dá nastaviť počet bodov každej jednotlivej odpovedi(u zápornej to značí koľko bodov hráč stratí) a v Quizziz sa dá nastaviť iba všeobecne počet bodov za otázku.

Quizziz má okrem klasického výberu otázok aj doplňovacie otázky a otázky kde sa môže hráč rozpísať ale u klasických ma maximum odpovedí iba 5.

Počas hrania ma Quizziz rôzne power upi a dáva možnosť záchrany opravou jednej zo zlých odpovedí čo moja aplikácia nerobí

Všeobecne moja aplikácia je výrazne jednoduchšia oproti ostatným aplikáciám, keďže ponúka len prihlásenie, zdieľanie kvízov , tvorbu kvízov kde je zadaná otázka a užívateľ vyberie koľko dá odpovedí a za koľko bodov a počas hrania len kliká na odpovede a na konci zistí koľko získal bodov. Ale vďaka tejto jednoduchosti je aplikácia viacej prehliadná, keďže neposkytuje žiadne prémiové verzie a hromadu možnosti ako si upraviť svoj účet keď chce človek iba rýchlo urobiť pár kvízov.

Návrh riešenia problému

Krátka analýza

Aplikácia sa dá využiť na:

* Vytvorenie a úprava vlastného kvízu
* Zdieľanie daného kvízu s ostatnými užívateľmi
* Hranie kvízov

Návrh aplikácie

Obrazovky:

* Authorization – Prihlasovanie do aplikácie.
* MainMenu – Výber medzi knižnicou, zdieľanými kvízmi a odhlásením sa.
* QuizLibrary – Knižnica užívateľových kvízov, kde užívateľ môže vytárať nové kvízy a upravovať už vytvorené.
* OnlineQuizzes – Zoznam všetkých zdieľaných kvízov kde si užívateľ môže zvoliť, ktorý chce hrať.
* QuizCreation – Tvorba kvízov.
* QuizGame – Hranie kvízu.

Každá obrazovka má svoj viewModel.

Dáta:

* AnswerData – Informácie jednotlivej odpovede.
* QuestionData – Informácie jednotlivej otázky.
* QuizData – Informácie jednotlivého kvízu.
* StatData – Štatistiky hráča v danom kvíze.

Database – Trieda, ktorá zabezpečuje komunikáciu s Firebase databázou.

MainScreenNavGraph – Zabezpečuje navigáciu medzi obrazovkami.

Popis implementácii

1.Povinné

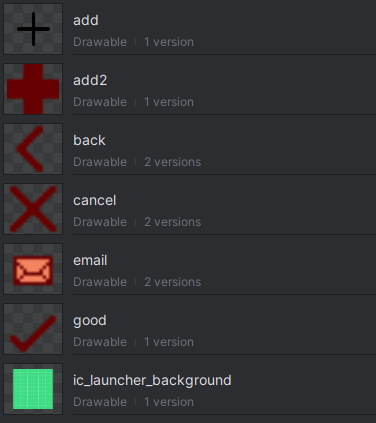
Na dobré reagovanie otáčania boli využite padding, fillMaxWidth, niekde weight aby boli dané komponenty vždy dobre rozdelené a viewModeli jednotlivých obrazoviek si pametajú potrebné informácie, vdaka čomu sa nevymažu napríklad informácie v textfieldoch.

Jednoduchy príklad pre rozloženie MainMenu:

@Composable  
fun MainMenu(  
 modifier: Modifier = Modifier,  
 onNavigateBack: () -> Unit = **{}**,  
 navigateLibrary: () -> Unit = **{}**,  
 navigateOnline: () -> Unit = **{}**,  
 mainMenuViewModel: MainMenuViewModel = viewModel()) {  
 Column (  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(25.*dp*),  
 verticalArrangement = Arrangement.Center,  
 horizontalAlignment = Alignment.CenterHorizontally  
  
 ) **{** OptionButton(  
 onClick = navigateLibrary ,  
 text = stringResource(R.string.*my\_library*),  
 icon = R.drawable.*library*,  
 modifier = Modifier  
 .*fillMaxWidth*()  
 .*padding*(bottom = 32.*dp*)  
  
 )...

Stringy su v resources :

<resources>  
 <string name="app\_name">SemestralnaPraca</string>  
 <string name="back">back</string>  
 <string name="username">Username</string>  
 <string name="password">Password</string>  
 <string name="log\_in">Log in</string>  
 <string name="play\_quizzes">Play Quizzes</string>  
 <string name="sign\_out">Sign out</string>  
 <string name="email">email</string>  
 <string name="my\_library">My Library</string>  
 <string name="quizzes">Quizzes</string>...

Obrázky:

2. Obrazovky

Aplikácia ma 6 obrazoviek:

Authorization, MainMenu, QuizLibrary, OnlineQuizzes, QuizCreation, QuizGame

3.Využitie AndroidX komponentov

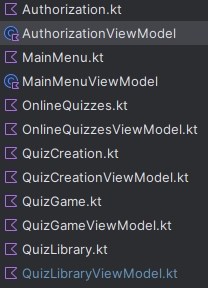
LifeCycles, ViewModel – každá obrazovka má svoj VewModel, ktorý spravuje v UIState dáta a vykonáva funkcie potrebné pre dobru funkcionalitu obrazovky.

class QuizLibraryViewModel: ViewModel() {  
 private val \_quizzesState = *MutableStateFlow*(QuizLibraryUiState())  
 val quizzesState: StateFlow<QuizLibraryUiState> = \_quizzesState  
  
 private val database = Database.getInstance()

...

fun rename() {  
 val updateInfo: HashMap<String, Any> = *hashMapOf*(  
 "name" *to* \_quizzesState.value.textForRenaming  
 )  
 *CoroutineScope*(Dispatchers.Main).*launch* **{** withContext(Dispatchers.IO) **{** database.updateContentInDatabase(  
 table = "quizzes",  
 childPath = *listOf*(\_quizzesState.value.quizID),  
 updateInfo  
 )  
 **}** \_quizzesState.value = \_quizzesState.value.copy(quizID = "")  
 loadQuizzesFromDatabase()  
 **}**}

...



Navigation – využitý na navigáciu medzi obrazovkami

@Composable  
fun MainScreenNavGraph(  
 modifier: Modifier = Modifier,  
 navController: NavHostController = rememberNavController()  
) {  
 val startDestination: String  
 val currentUser = Firebase.*auth*.*currentUser* if (currentUser != null) {  
 startDestination = Screens.*MainMenu*.name  
 } else {  
 startDestination = Screens.*Authorization*.name  
 }  
 System.*out*.println(currentUser)  
 NavHost(navController = navController,  
 startDestination = startDestination ) **{** *composable*(route = Screens.*Authorization*.name) **{** Authorization(onNavigateUp = **{** navController.navigate(Screens.*MainMenu*.name) **}**)  
 **}** *composable*(route = Screens.*MainMenu*.name) **{** MainMenu(onNavigateBack = **{** navController.navigate(Screens.*Authorization*.name) **}**,  
 navigateLibrary = **{** navController.navigate(Screens.*Library*.name) **}**,  
 navigateOnline = **{** navController.navigate(Screens.*Online*.name) **}**)  
 **}...**

7. Použitie externého frameworku / knižnice

Využitie firebase na prihlasovanie a databázu aby som mohol ukladať dáta online medzi zariadeniami a tým zabezpečil možnosť zdieľania kvízov.

Prihlasovanie:

import com.google.firebase.Firebase  
import com.google.firebase.auth.auth  
import kotlinx.coroutines.tasks.await  
class AuthorizationServices  
{  
 suspend fun logIn(email: String, password: String) {  
 Firebase.*auth*.signInWithEmailAndPassword(email, password).await()  
  
 }  
 suspend fun signUp(email: String, password: String) {  
 Firebase.*auth*.createUserWithEmailAndPassword(email, password).await()  
 }fun signOut() {  
 Firebase.*auth*.signOut()  
 }  
}

Databáza:

import android.util.Log  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.database.DatabaseReference  
import com.google.firebase.database.FirebaseDatabase  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.tasks.await  
import kotlinx.coroutines.withContext  
class Database {  
 private val database = FirebaseDatabase.getInstance()  
 companion object {  
 @Volatile  
 private var instance: Database? = null  
 fun getInstance(): Database {  
 return instance ?: *synchronized*(this) **{** instance ?: Database().*also* **{** instance = **it }  
 }** }  
 }

suspend fun addQuizToDatabase(quiz : QuizData): String {  
 return withContext(Dispatchers.IO) **{** val quizzesRef = database.getReference().child("quizzes")  
 val newQuizRef: DatabaseReference = if (quiz.quizId == "") {  
 quizzesRef.push()  
 } else {  
 quizzesRef.child(quiz.quizId)  
 }  
  
 var userID = ""  
  
 val currentUser = FirebaseAuth.getInstance().*currentUser* currentUser?.*let* **{** userID = currentUser.*uid* **}** val quizData = *hashMapOf*(  
 "userID" *to* userID,  
 "name" *to* quiz.quizName,  
 "sharedToPublicQuizzes" *to* quiz.shared.toString(),  
 "shareID" *to* quiz.shareID,  
 "numberOfQuestions" *to* quiz.numberOfQuestions,  
 "time" *to* quiz.time.toString()  
 //Add more data  
 )  
  
 newQuizRef.setValue(quizData)  
 .addOnSuccessListener **{** Log.d("addQuizToDatabase", "Quiz Added Succesfully")  
 **}** .addOnFailureListener **{** e **->** Log.e("addQuizToDatabase","ERROR adding quiz",e)  
 **}**.await()  
  
 newQuizRef.*key*.*toString*()  
 **}**}...

Databáza vo firebase



Zdroje

Využité tutoriály a podobne:

* Prihlasovanie pomocou Firebase Authentication - <https://firebase.google.com/docs/auth/android/start>
* Práca s firebase databázou - <https://firebase.google.com/docs/database>

+ a neskôr zmena na asynchrónne funkcie pomocou chatgpt

(tutorial na firebase je aj v AndroidStudiu v tools firebase)

* Odpočet v QuizGameViewModel – pomocou chatgpt
* Porovnanie času v QuizGameViewModel – pomocou chatgpt

Linky na obrázky z internetu:

* https://www.veryicon.com/icons/internet--web/billion-square-cloud/rename-5.html
* https://www.svgrepo.com/svg/126178/question-mark
* https://iconduck.com/icons/68205/share
* https://commons.wikimedia.org/wiki/File:OOjs\_UI\_icon\_trash\_apex.svg#/media/File:OOjs\_UI\_icon\_trash\_apex.svg
* https://commons.wikimedia.org/wiki/File:Plus\_symbol.svg
* https://iconscout.com/free-icon/back-arrow-1767523
* https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.svgrepo.com%2Fsvg%2F14478%2Femail&psig=AOvVaw1dl21oarZd0q3uTvrx-SCf&ust=1714400879265000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCIDOwI2P5YUDFQAAAAAdAAAAABAE
* https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.svgrepo.com%2Fsvg%2F505250%2Fplus&psig=AOvVaw2jeEPkC4M2S7UWO7c84XT6&ust=1714392109839000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCLCm9rfu5IUDFQAAAAAdAAAAABAd