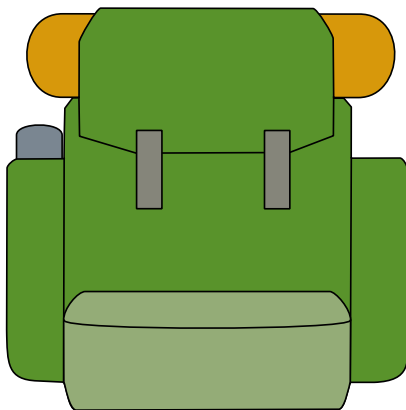




ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

Semestrálna práca z predmetu *Vývoj aplikácií pre
mobilné zariadenia*

PACKASSIST



Vypracoval: Lucia Sochorová

Študijná skupina: 5ZYI23



Popis a analýza aplikácie

Aplikácia PackAssist slúži na pomoc pri balení na rôzne akcie (udalosti). Umožňuje používateľom vytvárať a upravovať kolekcie predmetov, pridávať a upravovať udalosti, k udalostiam vytvoriť zoznamy vecí na zbalenie, do ktorých sa dajú ľahko vybrať predmety z vytvorených kolekcií, vyhľadávať a zobrazovať kolekcie a udalosti a základné informácie o nich.

Spravovanie kolekcií

- Používateľ by mal byť schopný vytvárať, upravovať a mazať kolekcie.
- Kolekcia by mala mať názov a minimálne jeden predmet.
- Pri vytváraní novej kolekcie by mal používateľ vedieť importovať predmety z inej už existujúcej kolekcie.

Spravovanie udalostí

- Používateľ by mal byť schopný vytvárať, upravovať a mazať udalosti.
- Udalosť by mala mať ako informácie názov a môže mať dátum, miesto a poznámky.

Zoznamy na zbalenie

- Používateľ by mal vedieť v jednotlivých udalostiach vytvoriť, mazať a upravovať zoznamy vecí na zbalenie.
- Pri vytváraní nového zoznamu by sa malo dať importovať predmety z už existujúcej kolekcie.
- Pri zobrazení zoznamov v udalosti sa dá pri každom predmete poznačiť, či je zbalený.

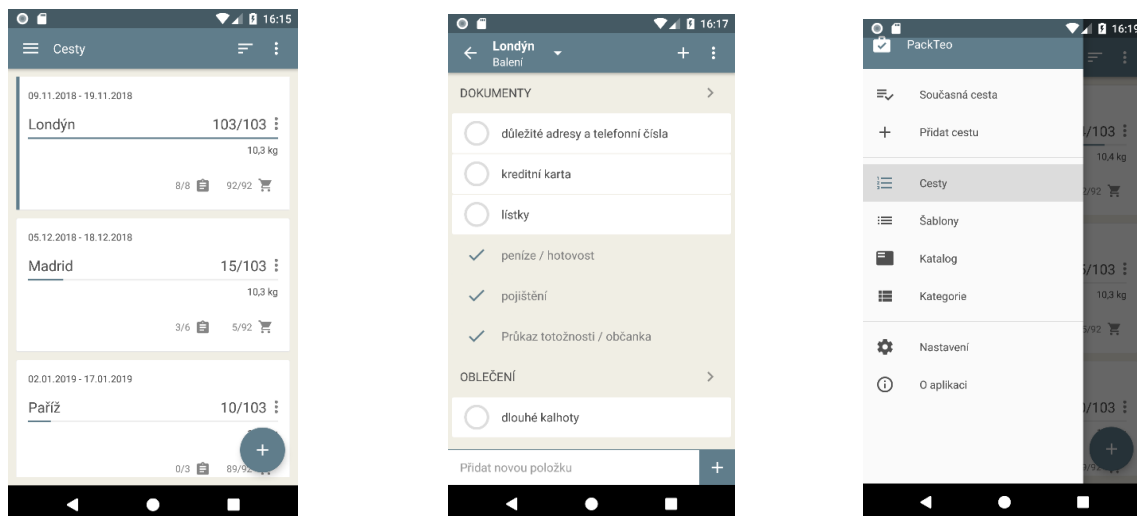
Zobrazovanie udalostí a kolekcií

- Používateľ by si mal vedieť zobraziť zoznam, ktorý je usporiadaný pre ľahšie prechádzanie, udalostí a kolekcií.
- Pri udalosti by sa mali dať zobraziť jednotlivé informácie o nej a zoznamy vecí na zbalenie.
- V zozname kolekcií alebo udalostí by sa malo dať vyhľadávať.

Podobné aplikácie

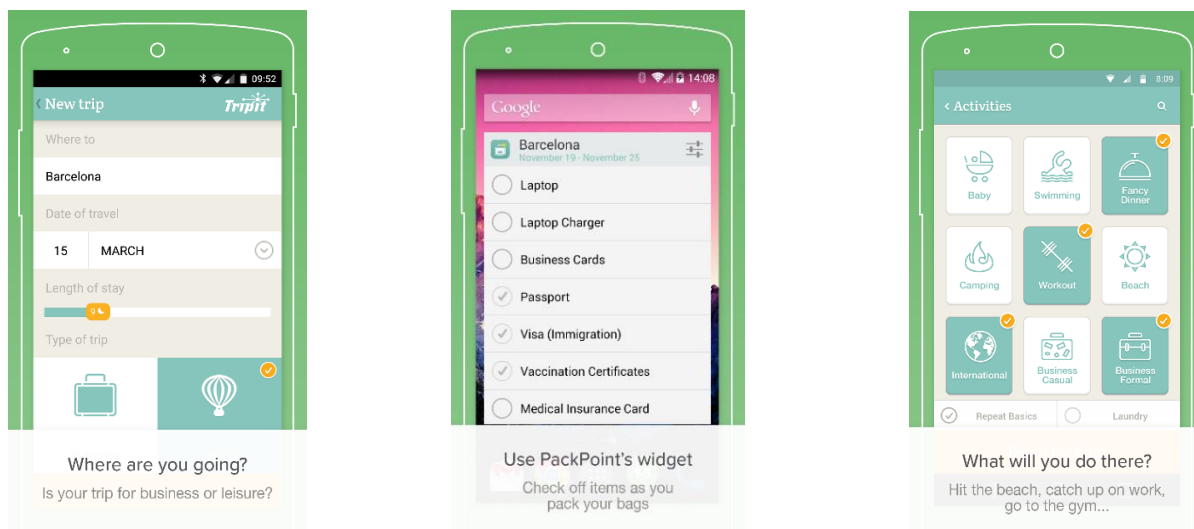


PackTeo - Seznam věcí



Moja aplikácia sa líši v tom, že je jednoduchšia - má menej možností na kategorizovanie predmetov, a dajú sa poznámky písať ku samotnej udalosti (tripu) a nie k jednotlivým predmetom a nemá možnosť výpočtu predpokladanej váhy predmetov.

PackPoint travel packing list



Moja aplikácia neurčuje používateľovi, čo si má zbalíť na základe výberu aktivít, ale sám si to zadáva. Pri výbere dátumu ponúkne výber z kalendára, v PackPoint je to manuálne zadanie dňa a mesiaca. V mojej aplikácii je prehľadnejší prehľad všetkých výletov.

Návrh riešenia

Vytvorenie kolekcie

1. Používateľ vyberie možnosť „pridať kolekciu“.
2. Zobrazí sa formulár na zadanie názvu a písanie položiek kolekcie a možnosť importu položiek z existujúcich kolekcií.



3. Po potvrdení, ak má kolekcia meno a aspoň jednu položku, sa kolekcia uloží do databázy. Pri zrušení sa nič neuloží.
4. Systém skryje formulár.

Upravenie kolekcie

1. Používateľ vyberie existujúcu kolekciu.
2. Zobrazí sa formulár na úpravu názvu a položiek a možnosť na pridanie novej položky.
3. Po výbere možnosti potvrdenia zmien sa zmeny uložia. Po výbere zrušenia zmien sa zmeny neuložia.
4. Systém skryje formulár.

Zmazanie kolekcie

1. Používateľ vyberie existujúcu kolekciu.
2. Zobrazí sa formulár s informáciami o kolekcii.
3. Po zvolení možnosti vymazať sa ukáže potvrdzovací dialóg.
4. Po potvrdení sa kolekcia vymaže.

Vytvorenie udalosti

1. Používateľ vyberie možnosť „pridať udalosť“
2. Zobrazí sa formulár na zadanie názvu.
3. Po potvrdení zadaného názvu sa udalosť uloží
4. Prejde sa na Úpravu udalosti

Úprava udalosti

1. Používateľ vyberie existujúcu udalosť
2. Zobrazí sa formulár s informáciami o udalosti s možnosťou zmeny.
3. Pri navigácii späť sa zmeny uložia do databázy

Zmazanie udalosti

1. Používateľ vyberie existujúcu udalosť.
2. Zobrazia sa informácie o udalosti.
3. Po vybratí možnosti „zmazať“ sa zobrazí potvrdzovací dialóg
4. Po potvrdení sa celá udalosť zmaže

Vytvorenie zoznamu na zbalenie

1. Používateľ v udalosti vyberie možnosť „pridať nový zoznam“.
2. Otvorí sa totožný formulár ako pre vytvorenie kolekcie.



3. Po vyplnení potrebných informácií a potvrdení sa zoznam uloží do databázy.

Úprava zoznamu na zbalenie

1. Používateľ vyberie konkrétny zoznam predmetov.
2. Zobrazí sa formulár totožný s formulárom na úpravu kolekcie.
3. Po potvrdení zmien sa zmeny uložia.

Vymazanie zoznamu na zbalenie

1. Používateľ vyberie konkrétny zoznam predmetov udalosti.
2. Zobrazí sa formulár s informáciami o zozname.
3. Po vybratí možnosti „zmazať“ sa zobrazí potvrdzovací dialóg.
4. Po potvrdení sa zoznam zmaže.

Zobrazenie kolekcií

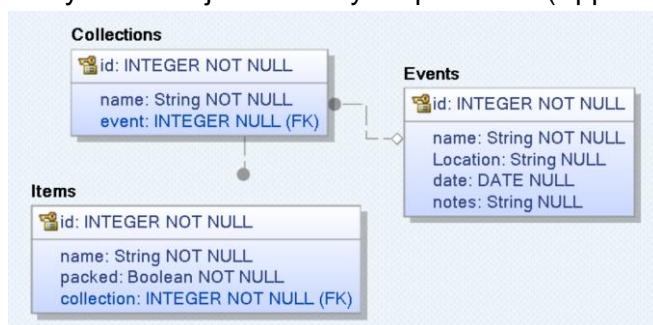
1. Ak chce používateľ zobraziť zoznam kolekcií, zobrazí sa abecedne usporiadaní zoznam kolekcií.

Zobrazenie udalosti

1. Ak chce používateľ zobraziť zoznam udalostí, zobrazí sa usporiadaný zoznam udalostí zoradený podľa dátumu konania a názvu.

Návrh aplikácie

Na ukladanie dát sa využívajú data class: Collection – predstavujúce kolekcie a zoznamy na balenie, Item – položky kolekcií a zoznam, Event – udalosti. Tieto triedy tvoria entity lokálnej databázy aplikácie (AppDatabase), jej schéma je na obrázku.



Každá entita má svoje DAO. Ku každému DAO prislúcha trieda offline repository, ktorá realizuje príslušné repository, k týmto triedam ďalej majú prístup triedy ViewModel cez triedu PackAssistApplication, ktorá detí z Application, a má atribút typu AppContainer. Každý ViewModel má atribút triedy typu data class s atribútmi reprezentujúce stav častí príslušnej obrazovky. Trieda NavGraphBuilder má extension functions pre každú obrazovku, v nej obrazovke sprostredkováva metódy ViewModelu a UI stav. V triede MainActivity sa potom cez funkciu PackAssistApp vytvorí navigačný graf. Na obrázku sú znázornené triedy ich atribúty a funkcie.



Popis implementácie

Aplikácia je rozdelená do dvoch vrstiev, dátovej a UI.

Dátová vrstva pozostáva z lokálnej databázy a tried určených na prácu s ňou, databáza je vytvorená za pomoci knižnice Room.

UI vrstva pozostáva hlavne z obrazoviek ich viewModelov a navigácie. Každéj obrazovke prináleží viewModel a navigácia. V navigácii sa pomocou extonsion function pre NavGraphBuilder vytvorí časť navigačného grafu pre danú obrazovku, ktorá v parametroch dostane metódy z viewModelu a svoj UI stav. Taktiež pre jednoduchšiu navigáciu na danú obrazovku sa vytvorí extension function pre NavController, ktorá obsahuje príkaz na navigáciu na danú obrazovku. Ukážka navigácie pre obrazovku na zmenu kolekcie:

```
const val COLLECTION_ID_ARG = "collectionId"
private const val COLLECTION_EDIT_ROUTE = "collectionCreation/{${COLLECTION_ID_ARG}}"

/**
 * Adds the collection edit screen to the navigation graph.
 *
 * @param onBackPressed Callback to navigate back to the previous screen.
 * @receiver The NavGraphBuilder object that is used to construct the navigation graph.
 */
@Lucia
fun NavGraphBuilder.collectionEditScreen(
    onBackPressed: () -> Unit
) {
    composable(
        route = COLLECTION_EDIT_ROUTE,
    ) { this: AnimatedContentScope, it: NavBackStackEntry
        val viewModel: CollectionEditViewModel =
            viewModel(factory = CollectionEditViewModel.Factory)

        val uiState = viewModel.state

        CollectionEditScreen(
            uiState = uiState,
            onNameChange = viewModel::onNameChange,
            onNewItemChange = viewModel::onNewItemChange,
            onChangeItem = viewModel::onChangeExistingItem,
            onEditItem = viewModel::ifEmptyDeleteItem,
            saveCollection = viewModel::saveCollection,
            addItemAction = viewModel::addItem,
            onDelete = viewModel::deleteCollection,
            navigateBack = onBackPressed
        )
    }
}

/**
 * Navigates to the collection edit screen.
 *
 * @param collectionId The ID of the collection to edit.
 * @receiver The NavController object that is used to navigate between screens.
 */
@Lucia
fun NavController.navigateToCollectionEdit(collectionId : Int){
    this.navigate(COLLECTION_EDIT_ROUTE.replace( "oldValue:" "${COLLECTION_ID_ARG}", collectionId.toString()))
}
```



Samotný navigačný graf sa vytvorí v PackAssistApp.

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

    NavHost(
        navController = navController,
        startDestination = EVENT_LIST_ROUTE,
        modifier = Modifier
    )
    { this: NavGraphBuilder
        collectionsListScreen(
            onEditCollection = { collectionId -> navController.navigateToCollectionEdit(collectionId) },
            onAddNewCollection = { navController.navigateToCollectionCreation() },
            onNavigateToRoute = navController::navigateToBottomBarRoute
        )

        collectionCreationScreen(
            onNavigateBack = {
                navController.navigateUp()
            }
        )

        collectionEditScreen {
            navController.navigateUp()
        }
    }
}
```

Ukážka navigačného grafu

ViewModel každej obrazovky obsahuje jej UI stav, metódy na jeho zmenu a zmenu dát databázy (uloženie, aktualizácia, vymazanie). UI stav je vo viewModeli napríklad ako public atribút s privátnym setterom `MutableStateOf(ScreenUIstateDataClass)` alebo ako public immutable atribút `StateFlow< ScreenUIstateDataClass >`, ktorý sa zbiera pomocou `collectAsStateWithLifecycle`.

Obrazovky:

Obrazovka na zobrazenie kolekcií – zobrazuje zoznam vytvorených kolekcií, obsahuje pole na vyhľadávanie podľa názvu kolekcie alebo predmetu a tlačidlo na pridanie novej kolekcie

Obrazovka vytvorenia kolekcie – Obsahuje polia na zadanie názvu kolekcie, názvu nového predmetu, po pridaní nového predmetu aj pole tohto predmetu. Má tlačidlo na import predmetov z existujúcej kolekcie, teda zobrazenie zoznamu možných kolekcií z ktorých sa dá importovať, tlačidlo na zrušenie, teda vrátenie sa späť bez vytvorenia novej kolekcie,



a tlačidlo na vytvorenie (uloženie) novej kolekcie. Využíva sa tiež na vytvorenie zoznamu vecí na zbalenie

Obrazovka úpravy existujúcej kolekcie – Obsahuje vyplnené textové polia s informáciami o názve kolekcie a jej položkách, jedno prázdne pole slúži na pridanie nového predmetu, tlačidlo na vymazanie, zrušenie zmien a tlačidlo na potvrdenie zmien.

Obrazovka na zobrazenie udalostí – zobrazuje jednotlivé udalosti s ich vyplnenými základnými informáciami, obsahuje pole na vyhľadávanie a tlačidlo na pridanie novej udalosti.

Dialóg na vytvorenie novej udalosti – dialóg ktorý obsahuje pole na zadanie názvu udalosti, tlačidlo na uloženie (vytvorenie) a tlačidlo zrušenia.

Obrazovka na zobrazenie a úpravu udalosti – obsahuje polia na zadanie alebo zmenu základných informácií, zoznam zoznamov predmetov na zbalenia, ktoré sa dajú zbaliť a rozbaliť a pri položkách zaškrtnúť ich zbalenie, ak je všetko zo zoznamu zbalené pri jeho názve sa objaví symbol „Like“. Obsahuje tlačidlo späť, kedy sa uložia zmeny a FAB na manažovanie zoznamov.

Obrazovka pre manažovanie zoznamov udalosti – obsahuje abecedne zoradený zoznam kariet zoznamov predmetov na zbalenie, každá karta má tlačidlo na úpravu daného zoznamu. Obrazovka obsahuje tlačidlo späť a FAB na pridanie nového zoznamu.

Zoznam použitých zdrojov

- <https://stackoverflow.com/questions/44364240/android-room-get-the-id-of-new-inserted-row-with-auto-generate> - ako pristúpiť ku práve vloženému prvku do databázy pri jeho neznámej hodnote PK
- <https://github.com/android/compose-samples/tree/main/Jetsnack> - inšpirácia pri tvorbe navigation bottom bar
- <https://issuetracker.google.com/issues/336842920#comment8> – riešenie *FATAL EXCEPTION CompositionLocal LocalLifecycleOwner not present* pri *collectAsStateWithLifecycle*