

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY



ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia
a informatiky

VÝVOJ APLIKÁCIÍ PRE MOBILNÉ ZARIADENIA

Študijný program: Informatika a riadenie

Vypracoval: Erik Beňuš
Študijná skupina: 5ZYR21
Školský rok: 2023/2024
Vyučujúci: doc. Ing. Patrik Hrkút, PhD.

I. POPIS APLIKÁCIE

Za cieľ som si dal vytvorenie aplikácie na správu osobných financií. Už dlhšie využívam na prehľad mojich financií Excel tabuľky, ktoré mám uložené na Google Disku. Vďaka tejto aplikácie bude môj prehľad o financiách dostupnejší a bude sa dať jednoduchšie aktualizovať.

ZÁKLADNÉ INFORMÁCIE O APLIKÁCIÍ

Názov aplikácie:	Moje Financie
Podobné aplikácie:	Správca peňazí
	TradingView
	Money Manager
Základné funkcie:	Pridanie, odstránenie príjmov/výdajov
	Spravovanie portfólia akcií a kryptomien
	Zobrazovanie aktuálneho stavu
Dostupné na:	Android

Tabuľka 1: Základné informácie o aplikácii



Obrázok 1: Logo aplikácie Moje Financie

II. SPRACOVANIE PREHĽADU DOSTUPNÝCH APLIKÁCIÍ PODOBNÉHO ZAMERANIA

Existuje viac aplikácií s podobným zameraním, na podrobnú analýzu som si vybral 3 aplikácie, a to sú Správca peňazí, TradingView a Money Manager:

Základné informácie na porovnanie podobnosti s navrhovanou aplikáciou:

Názov/Funkcia	Vytvorenie rozpočtu	Akciové portfólio	Vizualizácia financií	Slovenský jazyk
Správca peňazí	Áno	Nie	Áno	Áno
TradingView	Nie	Áno	Áno	Nie
Money Manager	Áno	Nie	Áno	Nie
Moje Financie	Áno	Áno	Áno	Áno

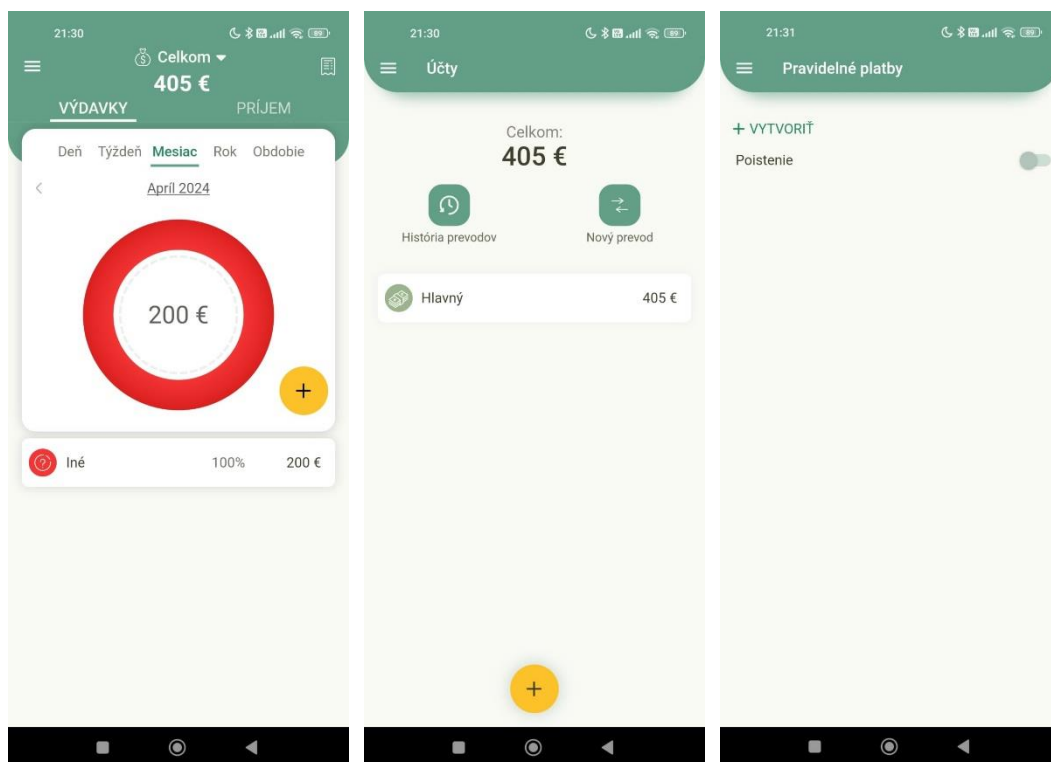
Tabuľka 2: Základné porovnanie aplikácie s konkurenciou

A. SPRÁVCA PEŇAZÍ

Jednoduchá aplikácia na sledovanie svojich príjmov a výdajov.

Aplikácia ponúka:

- Vizualizáciu dát napr. pomocou koláčového grafu,
- nastavenie pravidelných príjmov alebo výdajov,
- nastavenie viacerých účtov,
- zdieľanie s priateľmi,
- nastavenie pripomienok,
- rozdelenie príjmov a výdajov do kategórií,
- dostupnosť v slovenskom jazyku.



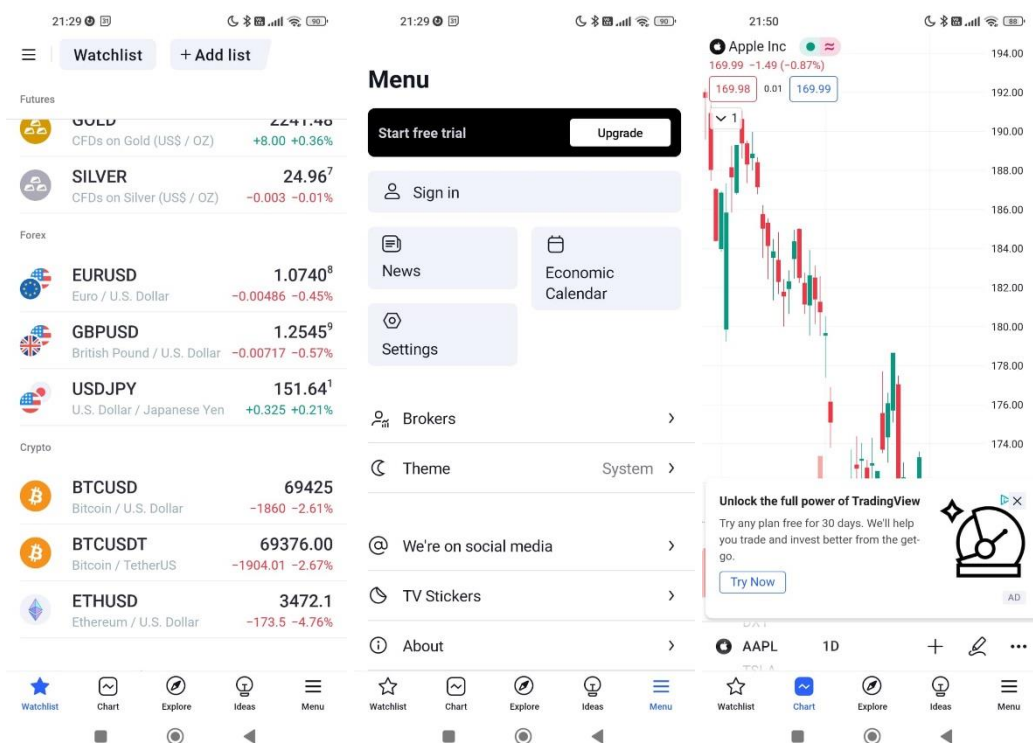
Obrázok 2: Užívateľské rozhranie aplikácie Správca peňazí

B. TRADINGVIEW

Aplikácia primárne zameraná na sledovanie akciového portfólia, taktiež je možné sledovať menové kurzy, ETF fondov a kryptomeny.

Aplikácia ponúka:

- Sledovanie cien komodít v reálnom čase,
- pridanie do tzv. watchlistu,
- zobrazovanie grafov,
- zaujímavé nápady a tipy,
- príbehy ľudí, ktorí investujú
- články zo sveta financií,
- rebríček podľa hodnotení užívateľov,
- ekonomický kalendár.



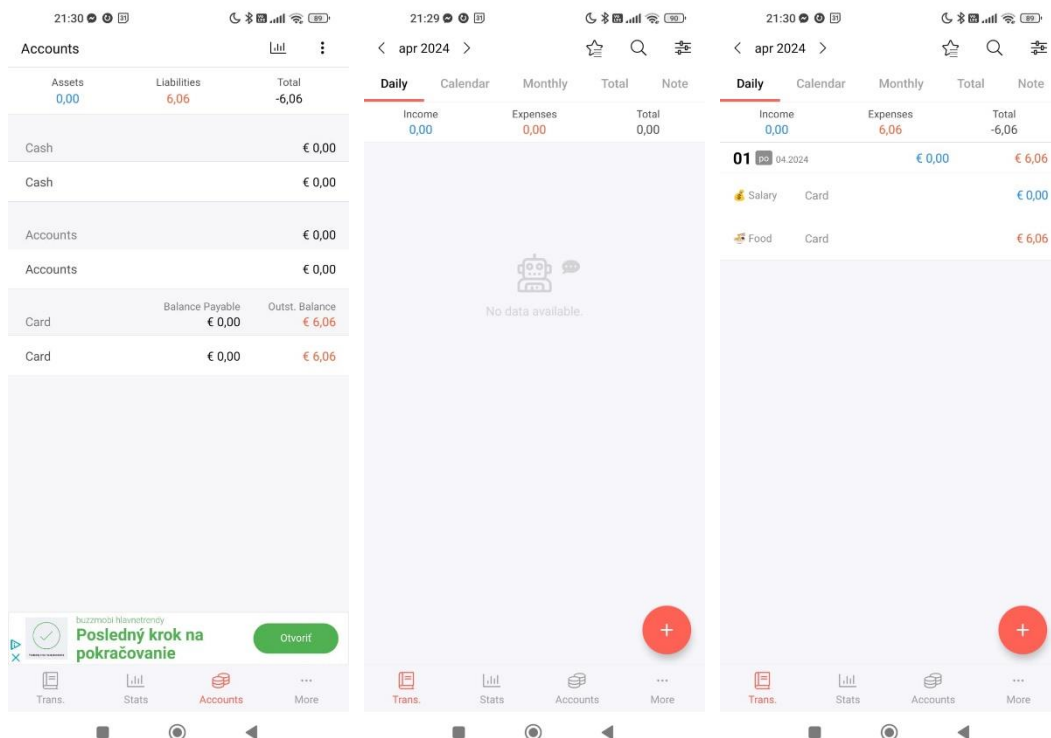
Obrázok 3: Užívateľské rozhranie aplikácie TradingView

C. MONEY MANAGER

Aplikácia zameraná na spravovanie osobných financií.

Aplikácia ponúka:

- Vizualizáciu dát napr. pomocou koláčového grafu,
- Možnosť správy aj v počítačovej verzii,
- nastavenie pravidelných príjmov alebo výdavkov,
- nastavenie viacerých účtov,
- nastavenie hesla,
- rozdelenie príjmov a výdavkov do kategórií,
- vybranie spôsobu platby ako hotovosť alebo karta.



Obrázok 4: Užívateľské rozhranie aplikácie Money Manager

III. ANALÝZA APLIKÁCIE MOJE FINANCIE

Aplikácia Moje Financie slúži na správu osobných financií, uľahčuje prehľad a aktualizáciu výnosov alebo výdavkov. Dostupný prehľad ohľadom portfólia akcií a kryptomien, a teda užívateľ bude vedieť akú hodnotu majú jeho portfólia, potom čo si ich manuálne aktualizuje.

VÝNOSY A VÝDAJE

Aplikácia ponúka nasledovné možnosti spracovania výnosov alebo výdajov:

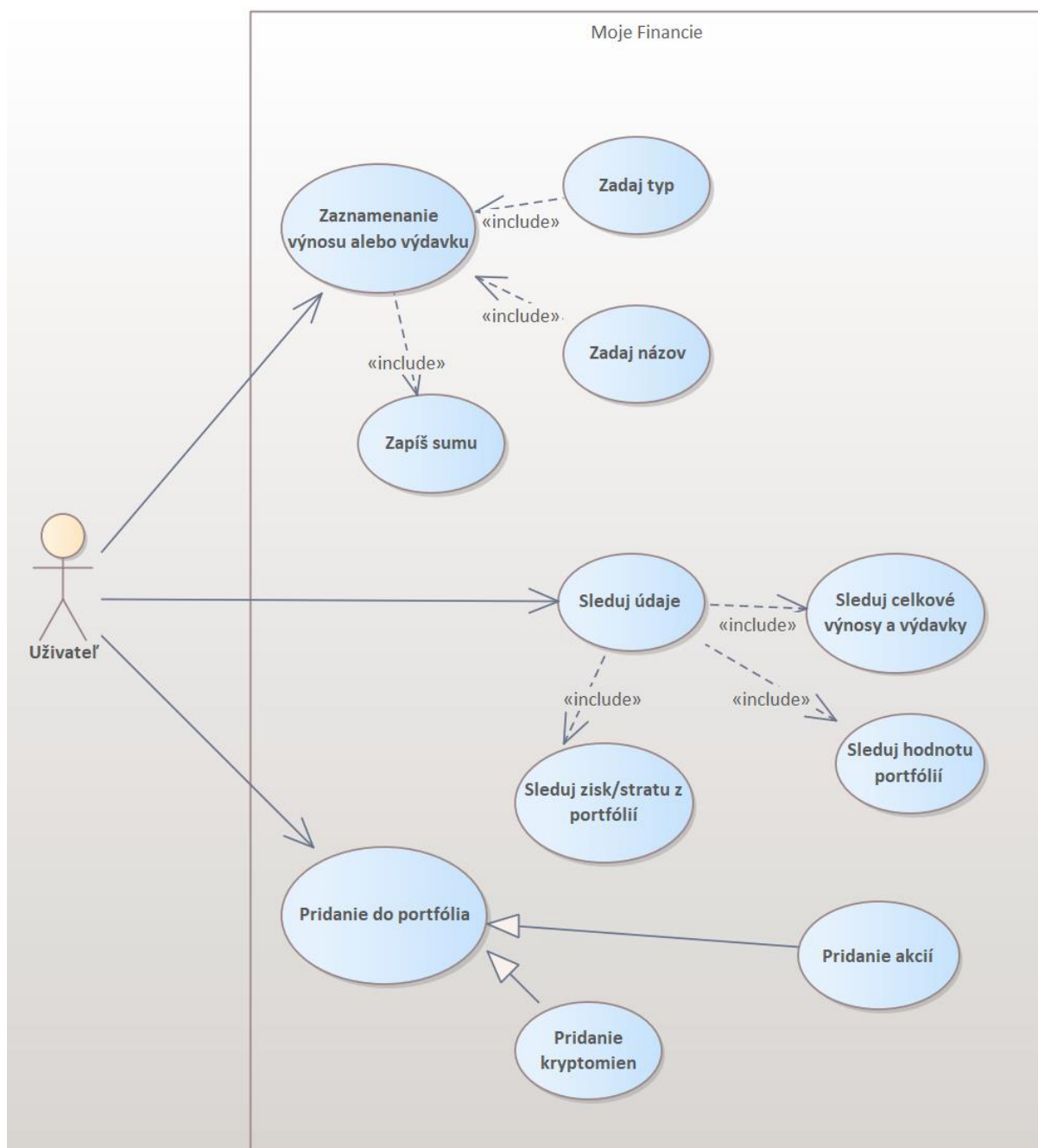
- Zápis sumy pre príjem alebo výdaj,
- zobrazenie celkových výdavkov a príjmov,
- osobitná kontrola jednotlivých príjmov alebo výdajov, možné upravenie alebo odstránenie

AKCIE A KRYPTOMENY

Aplikácia ponúka nasledovné možnosti spracovania dát ohľadom akciového trhu a kryptomien:

- Pridanie akcií do vlastného portfólia,
- pridanie kryptomien do vlastného portfólia,
- celková suma zo ziskov alebo strát investícií,
- manuálne aktualizovanie investícií

Bližšie ponúkané možnosti zobrazuje aj nasledujúci UML diagram prípadov použitia:



Obrázok 5: UML diagram prípadov použitia pre aplikáciu Moje Financie

IV. POPIS IMPLEMENTÁCIE APLIKÁCIE

Na implementovanie aplikácie boli využité AndroidX komponenty:

- Room (na vytvorenie SQLite databáz)
- Navigation (na prechod medzi obrazovkami)
- ViewModel (oddelenie logickej časti od užívateľského rozhrania)

OBRAZOVKY

Práca s obrazovkami fungovala spoločne s implementáciou ViewModelov a navigácie. Súčasťou aplikácie je menu (BottomMenu), s týmto menu spolupracuje 4 hlavné obrazovky:

- *HomeScreen*:
 - Domovská stránka, zobrazujú sa na nej základné údaje pomocou Card (Cards sú nastavené ako klikateľné, každá karta naviguje na príslušnú ďalšiu obrazovku)

```
NavHost(  
    navController = navController,  
    startDestination = Screens.HomeScreen.name,  
    modifier = Modifier  
        .padding(paddingValues)  
) { this: NavGraphBuilder  
    composable(route = Screens.HomeScreen.name) { this: AnimatedContentScope it: NavBackStackEntry  
        Home(navController = navController)  
    }  
}
```

Obrázok 6: Navigácia pre domovskú obrazovku

- *FinanceScreen*:
 - Zobrazujú sa na nej základné údaje pre príjmy a výdavky.
 - Taktiež obsahuje tlačidlo, ktoré naviguje na obrazovku *AddTransactionScreen* pre pridanie príjmu alebo výdavku.
 - Taktiež obsahuje Card, po ktorých kliknutí sa zobrazia detaily danej transakcie. Na to slúži obrazovka *TransactionDetailsScreen*.
 - Na obrazovke *TransactionDetailsScreen* môžete vymazať danú transakciu alebo ju po kliknutí na tlačidlo upraviť, po kliknutí na tlačidlo sa otvorí obrazovka na úpravu transakcií *EditTransactionScreen*.

```
floatingActionButton = {  
    FloatingActionButton(  
        onClick = { navController.navigate(Screens.AddTransactionScreen.name) },  
        modifier = Modifier.padding(20.dp)  
    ) {  
        Icon(  
            imageVector = Icons.Default.Add,  
            contentDescription = stringResource("Pridaj Transakciu")  
        )  
    }  
}
```

Obrázok 7: Navigácia pre floatingbutton na pridanie Transakcie

- *StocksScreen*:
 - Zobrazujú sa na nej portfólio už pridaných akcií.
 - Ak akcie nie sú pridané dá sa to urobiť cez tlačidlo, ktoré naviguje na obrazovku *AddStockScreen*.
 - Ak už pridaná akcia v portfóliu je možné ju editovať pomocou navigácie na obrazovku *EditInvestmentScreen* (Táto obrazovka slúži aj na úpravu kryptomien)

```
composable(
    route = InvestmentScreens.EditInvestmentScreenRoute,
    arguments = listOf(navArgument( name: "investmentId") { type = NavType.IntType })
) { this: AnimatedContentScope it: NavBackStackEntry
    EditInvestmentScreen(
        navigateBack = { navController.navigateUp() }
    )
}
```

Obrázok 8: Navigácia pre obrazovku na editovanie aj s argumentom

- *CryptoScreen* :
 - Zobrazujú sa na nej portfólio už pridaných kryptomien.
 - Ak kryptomeny nie sú pridané dá sa to urobiť cez tlačidlo, ktoré naviguje na obrazovku *AddCryptoScreen*.
 - Editovanie funguje identicky ako pre akcie.

```
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun AddCryptoScreen(
    modifier: Modifier = Modifier,
    navigateBack: () -> Unit,
    viewModel: CryptoViewModel = viewModel(factory = MyViewModelProvider.Factory)
) {
```

Obrázok 9: Získanie ViewModelu pre obrazovku AddCryptoScreen

PRÁCA S DATABÁZOU

Ukážky kódu pre prácu s databázou. Nasledujúce obrázky sú kusy kódu, kde sa pracuje na vytvorení tabuľky *Investment*, repozitár pre tabuľku a DAO:

```
@Entity(tableName = "investments")
data class Investment(
    @PrimaryKey(autoGenerate = true) val id: Int = 0,
    val name: String,
    val ticker: String,
    val startingAmount: Double,
    val currentAmount: Double,
    val type: InvestmentType
)

enum class InvestmentType {
    Stock,
    Crypto
}
```

Obrázok 10: Vytvorenie tabuľky *Investment* v databáze

```
interface InvestmentRepository {
    fun getInvestmentStream(id: Int): Flow<Investment?>
    fun getAllInvestmentsStream(): Flow<List<Investment>>
    fun getAllInvestmentsByTypeStream(type: InvestmentType): Flow<List<Investment>>
    suspend fun insertInvestment(investment: Investment)
    suspend fun deleteInvestment(investment: Investment)
    suspend fun updateInvestment(investment: Investment)
}
```

Obrázok 11: Repozitár pre tabuľku *Investment*

```
@Dao
interface InvestmentDao {
    @Insert(onConflict = OnConflictStrategy.IGNORE)
    suspend fun insert(investment: Investment)

    @Update
    suspend fun update(investment: Investment)

    @Delete
    suspend fun delete(investment: Investment)

    @Query("SELECT * FROM investments WHERE id = :id")
    fun getInvestment(id: Int): Flow<Investment?>

    @Query("SELECT * FROM investments ORDER BY name ASC")
    fun getAllInvestments(): Flow<List<Investment>>

    @Query("SELECT * FROM investments WHERE type = :type")
    fun getAllInvestmentsByType(type: InvestmentType): Flow<List<Investment>>
}
```

Obrázok 12: Návrhový vzor DAO

V. ZOZNAM POUŽITÝCH ZDROJOV

- Pre prácu s databázami som využil CodeLab:

[Android Basics Compose - Use Room for data persistence | Android Developers](#)

- Pre prácu s BottomMenu som využil Youtube tutorial:

[https://www.youtube.com/watch?v=L_6poZGNXOo&ab_channel=Foxandroid](#)