**ŽILINSKÁ UNIVERZITA V ŽILINE**FAKULTA RIADENIA A INFORMATIKY

Vývoj aplikácií pre mobilné zariadenie  
  
**„Finančný kontrolór“**  
Študijný program: Informatika a Riadenie

Vypracoval:  
Hanko Radovan,  
5ZYR22

Akademický rok:  
2021/2022,  
Letný semester

1. **Špecifikácia zadania, definovanie problému**

Navrhnite aplikáciu pre android, v ktorej bude možné pridávať, upravovať, odstraňovať a zobraziť dáta o tom, aké množstvo peňazí ste minuli, kde ste ich utratili/ zarobili a kde ste ich zarobili a kedy sa to odohralo. Taktiež bude možné vložiť nejaký popis k jednotlivým dátam. Dáta sa budú ukladať do databázy. Jednotlivé dáta bude možné zobraziť cez graf v aplikácií. Riešenie bude možné spraviť cez javu alebo kotlin.

1. **Prehľad podobných aplikácií**

Pri navrhovaní aplikácie som spravil prieskum podobných dostupných aplikácií. Na trhu ich je celkom dosť, avšak na porovnanie som si vybral tieto aplikácie: 1Money a Money Manager.

Podstata týchto aplikácií je podobná, prípadne rovnaká, ako mnou navrhovanej aplikácie.

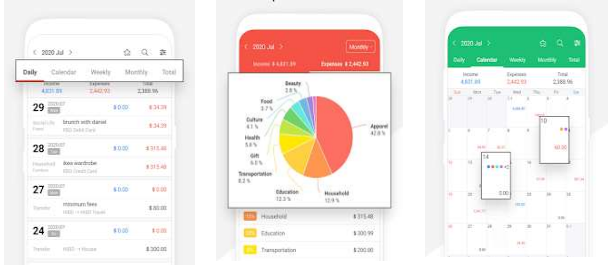
* 1. **1Money**

Tvorcom aplikácie je PixelRush



* 1. **Money Manager**

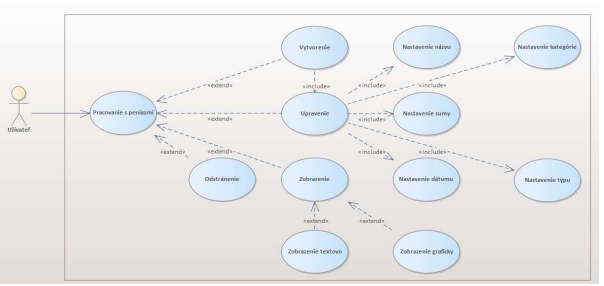
Tvorcom aplikácie je Realbyte Inc.



* 1. **Líšenie sa aplikácie od iných aplikácií (rovnakej povahy)**

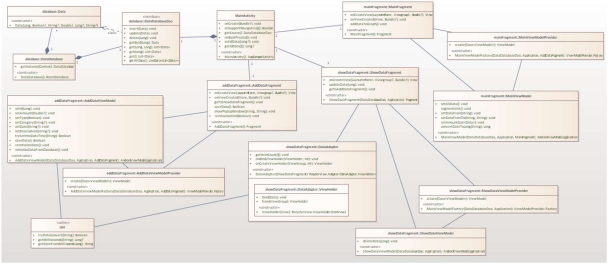
Moja aplikácia nie je tak prepracovaná ako iné aplikácie (napr. Money Manager, 1Money), avšak je to moja aplikácia a som na to hrdý, hoci niektoré veci som mohol spraviť aj lepšie.

1. **Analýza aplikácie**

****

Užívateľ bude môcť pri práci s peniazmi vytvoriť si informáciu, kedy, ako a kde ich minul alebo nadobudol, bude si taktiež môcť upraviť jednotlivé údaje, vymazať a taktiež zobraziť. Zobraziť si ich môže v grafe alebo textovo.

1. **Návrh aplikácie**

****

1. **Popis implementácie**
   1. **Databáza**
      1. **Verzia**

V aplikácií je využívaná databáza SQLite.

* + 1. **Popis tabuľky *data\_about\_money\_table***

V tabuľka obsahuje informácie o tom, akú sumu peňazí užívateľ minul/ získal, kategóriu, do ktorej sa zaraďuje minutá/ získaná suma, kedy a prípadne krátky popisok.

Jednotlivé entity sú zadefinované po anglicky, avšak sú to jednoduché a stručne názvy, preto si myslím, že nie je potrebné písať podrobnejšie o jednotlivých entitách.

Entita: Typ:   
data\_id Long not null primary key  
type Boolean  
category String  
amount Double  
date Long  
description String

* 1. **Program**
     1. **IDE**

Na vývoj aplikácie som využíval IDE Android Studio, ktoré podporuje programovanie v programovacom jazyku kotlin aj java. Celý kód obsahuje názvy parametrov, atribútov, metód, ... v angličtine pre univerzálnosť

* + 1. **Balík *database***

V balíku *database* sa nachádzajú triedy: *Data*, *DataDatabase* a *DataDatabaseDao*. Balík je dôležitý pre prácu s databázou.

Trieda *Data* obsahuje názvy pre jednotlivé entity v databáze.

Trieda *DataDatabase* je abstraktná trieda, ktorá slúži na vytvorenie databázy. Je potomkom triedy *RoomDatabase*.

Interface *DataDatabaseDao* je potrebný na prácu s databázou. Na vkladanie, editovanie, mazanie, zobrazovanie dát z databázy.

* + 1. **Balík *addDataFragment***

V balíku *addDataFragment* sa nachádzajú triedy: *AddDataFragment*, *AddDataViewModel* a *AddDataViewModelProvider*. Balík je dôležitý pre vkladanie alebo upravovanie dát do databázy.

Trieda *AddDataFragment* pracuje s fragmentom (layout) *fragment\_add\_data*, sú v ňom pomocou bindingu prepojené jednotlivé textové zoznamy a tlačidlá. Pomocou tejto triedy vie každé tlačidlo, čo robiť po stlačení. Taktiež má metódu na zobrazenie popUp okna definovaného cez AlertDialog, v prípade potreby. A metódu na vyčistenie alebo prepísanie textových zobrazení od textov.

Trieda *AddDataViewModel* je zodpovedná za spracovávanie dát, ktoré sú uložené cez MutableLiveData. Ich ukladanie, prepisovanie alebo vkladanie do databázy po príkaze od triedy *AddDataFragment*. V niektorých prípadov využíva pri práci s dátami aj viewModelScope, či triedu *Util*.

Trieda *AddDataViewModelProvider* je zodpovedná za vytvorenie objektov triedy *AddDataViewModel* pre objekty triedy *AddDataFragment*.

* + 1. **Balík *mainFragment***

V balíku *mainFragment* sa nachádzajú triedy: *MainFragment*, *MainViewModel* a *MainViewModelProvider*. Balík je dôležitý pre zobrazenie dát v grafe.

Trieda *MainFragment* pracuje s fragmentom (layout) *fragment\_main*, sú v ňom pomocou bindingu prepojené jednotlivé textové zoznamy a tlačidlo. Pomocou tejto triedy vieme zobraziť graf, v ktorom sú zobrazené celkové príjmy, či výdavky. Prípadne si vieme určiť odkedy chceme vidieť tieto dáta alebo odkedy až dokedy. Pomocou metódy v triede sa graf vždy prekreslí a zobrazí aktálne informácie.

Trieda *MainViewModel* je zodpovedná za spracovávanie dát, ktoré sú uložené cez MutableLiveData. V niektorých prípadov využíva pri práci s dátami aj viewModelScope, či triedu *Util*. Sú v nej uložené príjmy a výdavky za dané obdobie a segmenty pre graf.

Trieda *MainViewModelProvider* je zodpovedná za vytvorenie objektov triedy *MainViewModel* pre objekty triedy *MainFragment*.

* + 1. **Balík *showDataFragment***

V balíku *showDataFragment* sa nachádzajú triedy: S*howDataFragment*, *ShowDataFragment,* *ShowDataFragment DataAdapter*. Balík je dôležitý pre zobrazenie dát, či zmazanie jednotlivých dát.

Trieda *ShowDataFragment* pracuje s fragmentom (layout) *fragment\_show\_data*, sú v ňom pomocou bindingu prepojené jednotlivé dáta už spracované do layoutu *data\_list*. Pomocou tejto triedy vieme zobraziť všetky dáta, v ktorom sú zobrazené jednotlivé dáta z databázy. Prípadne, že je potrebné, tak je možné rolovať medzi dátami.

Trieda *ShowDataViewModel* je zodpovedná za spracovávanie dát, ktoré sú uložené cez MutableLiveData. Využíva pri práci s dátami aj viewModelScope, či triedu *Util*. Sú v nej uložené dáta a vie ich odstrániť.

Trieda *ShowDataViewModel* *Provider* je zodpovedná za vytvorenie objektov triedy *ShowDataViewModel* pre objekty triedy *ShowDataFragment*.

Trieda *DataAdapter* je snáď najdôležitejšia v tomto balíku, pretože väčšinu vecí sa koná v nej. Ukladá jednotlivé dáta do layoutu *data\_list*, umožňuje pomocou tlačidiel, ktoré sú v layoute *data\_list* ich upraviť, či vymazať.

* + 1. **Trieda *MainActivity***

Trieda *MainActivity* pracuje s aktivitou (layout) *activity\_main*. Trieda zodpovedá za správne fungovanie menu (menu) *main\_menu*, v ktorom je definovaný navigátor (navigation) *navigation*. A pomocou menu možno preklikávať medzi jednotlivými fragmentmi. Hlavným fragmentom je (layout) *fragment\_main*.

* + 1. **Trieda *Util***

Trieda *Util* je výpomocná trieda, v ktorej sú zadefinované niektoré funkcie, aby sa neopakovali v programe viackrát a zároveň sú viac všeobecné.