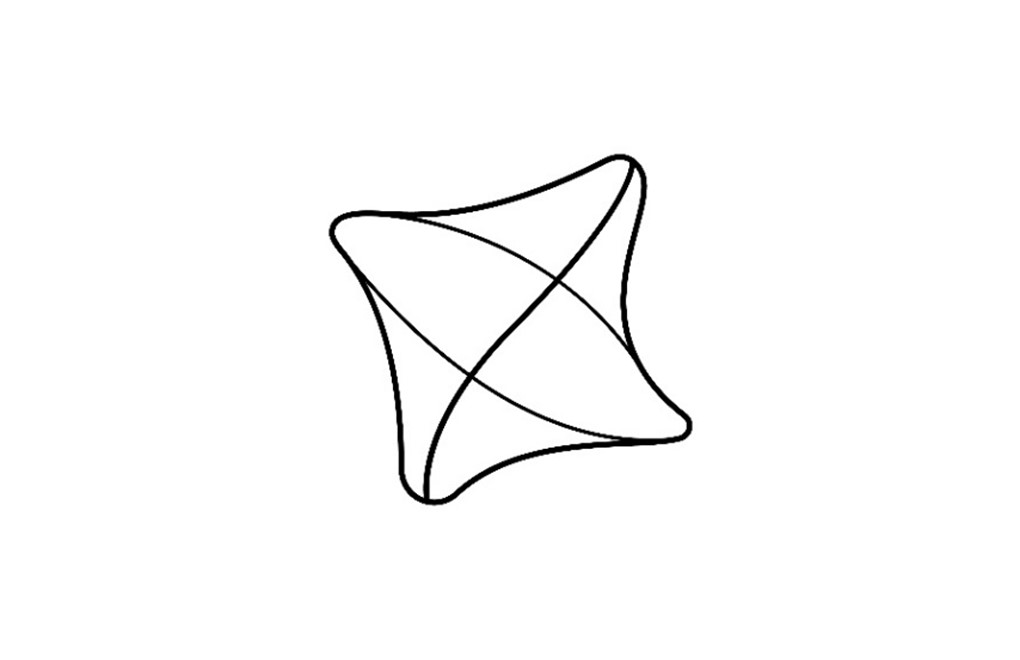
ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY



Semestrálna práca

Aplikácia TodoR

Vývoj aplikácií pre mobilné zariadenia

Vypracoval: Lukáš Jágrik

Študijný program: Informatika a riadenie

Študijná skupina: 5ZYR23

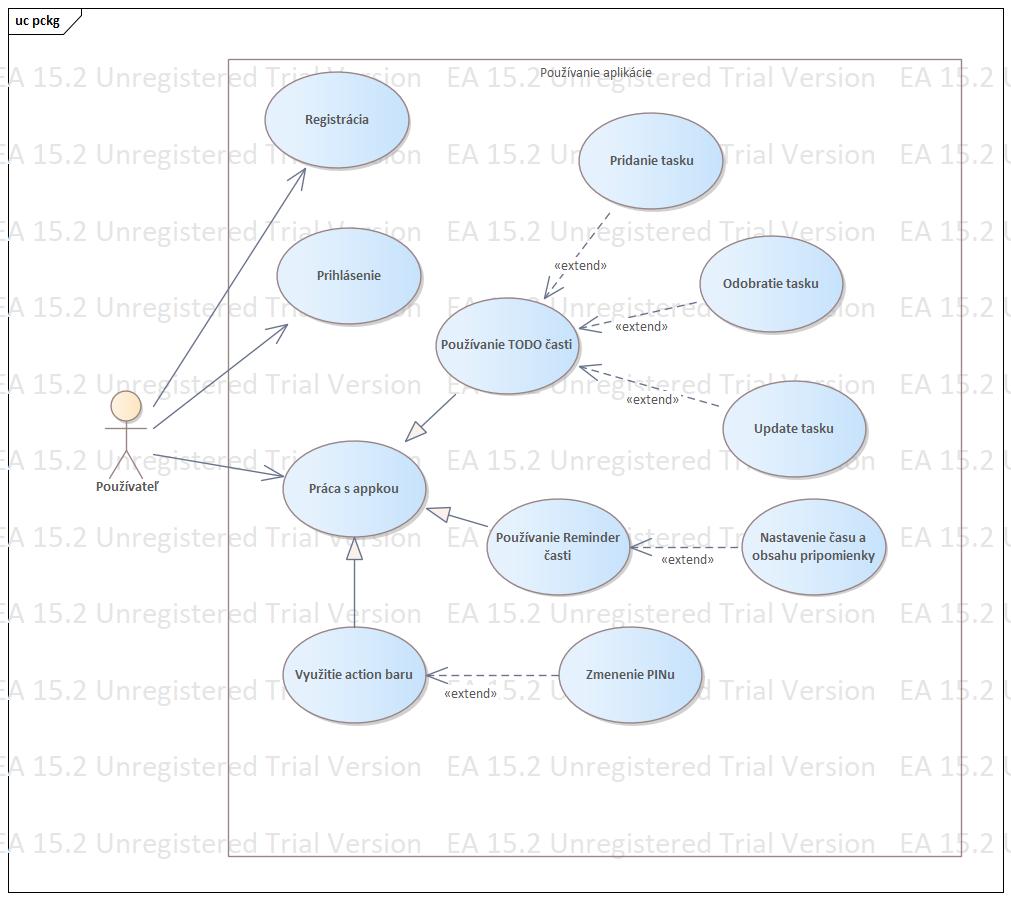
# **Popis a analýza riešeného problému**

Aplikácia TodoR je aplikácia pre Android zariadenia. Táto aplikácia umožňuje užívateľom zapisovať si úlohy (pridávanie taskov), upravovať ich, odfajknúť si ich splnenie, vymazanie a pridanie si upozornenia na konkrétny čas. Po prvom spustení si musí užívateľ zvoliť PIN (zaregistrovať) a následne sa pomocou neho prihlásiť. Po prihlásení PINom môže užívateľ používať aplikáciu.

# **Podobné aplikácie**

V Obchod Play je množstvo aplikácií zameraných na TODO listy alebo Remindere. Pre porovnanie som stiahol aplikáciu Todoist, ktorá slúži ako Todolist ale nikde som v nej nenašiel aby sa dalo nastaviť aby mi prišla nejaká notifikácia. Ďalšia aplikácia Reminder od Samsungu, poskytovala takúto možnosť. Skúsil som teda nejakým spôsobom spojiť tieto dve vymoženosti tak, že naviac od Todolistu sa viem prekliknúť do jednoduchého Remindru, kde si viem nastaviť upozornenie.

# **Návrh riešenia problému**



# **Popis implementácie**

Táto aplikácia bola naprogramovaná v prostredí Android studio s využitím jazyka Kotlin. Aplikácia pozostáva z 5tich aktivít a 5tich fragmentov.

Aktivity:

* MainActivity – je to základná aktivita, do ktorej sa užívateľ dostáva po prihlásení – je to vlastne Todolist, ktorý je základom tohto projektu
* ReminderActivity – aktivita reminderu v ktorej sa vie nastavovať upozorenie
* LoginActivity – aktivita prihlásenia sa do aplikácie
* RegisterActivity – aktivita registrovania sa do aplikácie
* PinChangeActivity – aktivita zmeny PINu

Fragmenty:

* TodolistFragment – fragment, ktorý uzko spolupracuje s TodolistAdapter
* AddNewTask – fragment v ktorom je implementovaná logika pridávania taskov
* RemoveTask – fragment v ktorom je implementovaná logika mazania taskov
* UpdateTask – fragment v ktorom je implementovaná logika aktualizovania textu tasku
* TaskStatusUpdate – fragment v ktorom je logika aktualizovania, či je task splnený alebo nie
* Taktiež využívam **RoomDatabase** na ukladanie jednotlivých taskov

Obrázok, na ktorom je text

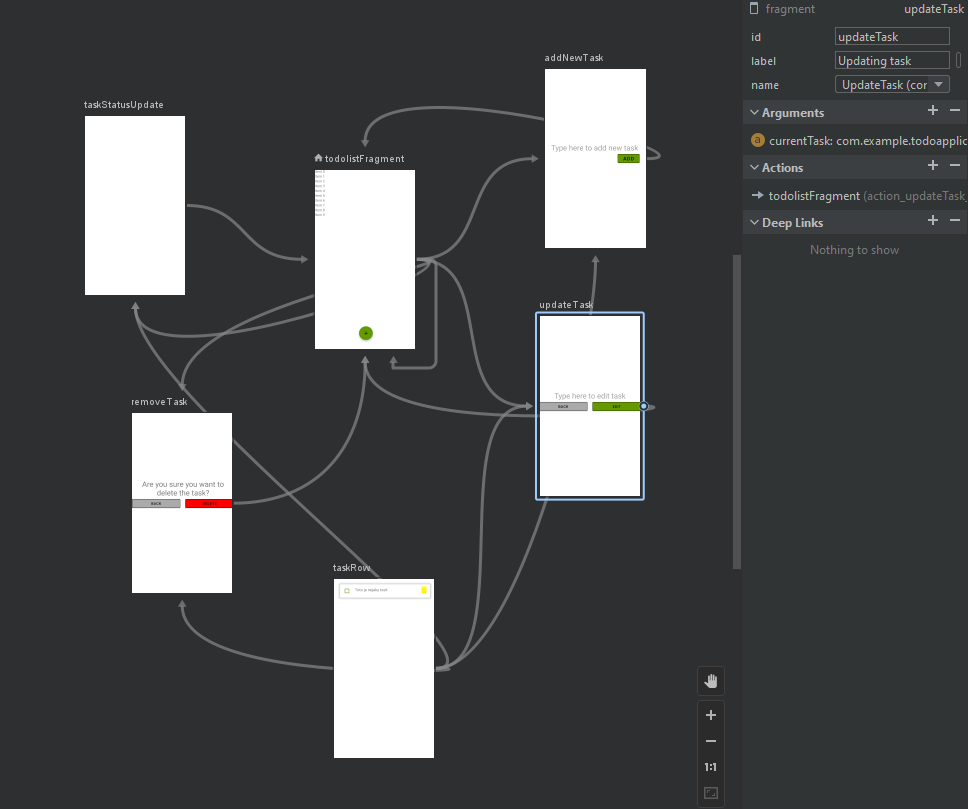
Automaticky generovaný popis

* Použil som taktiež **ViewModel**, v ktorom využívam **LiveData** pre List Taskov a **Korutiny** pri komunikácií s repozitárom.

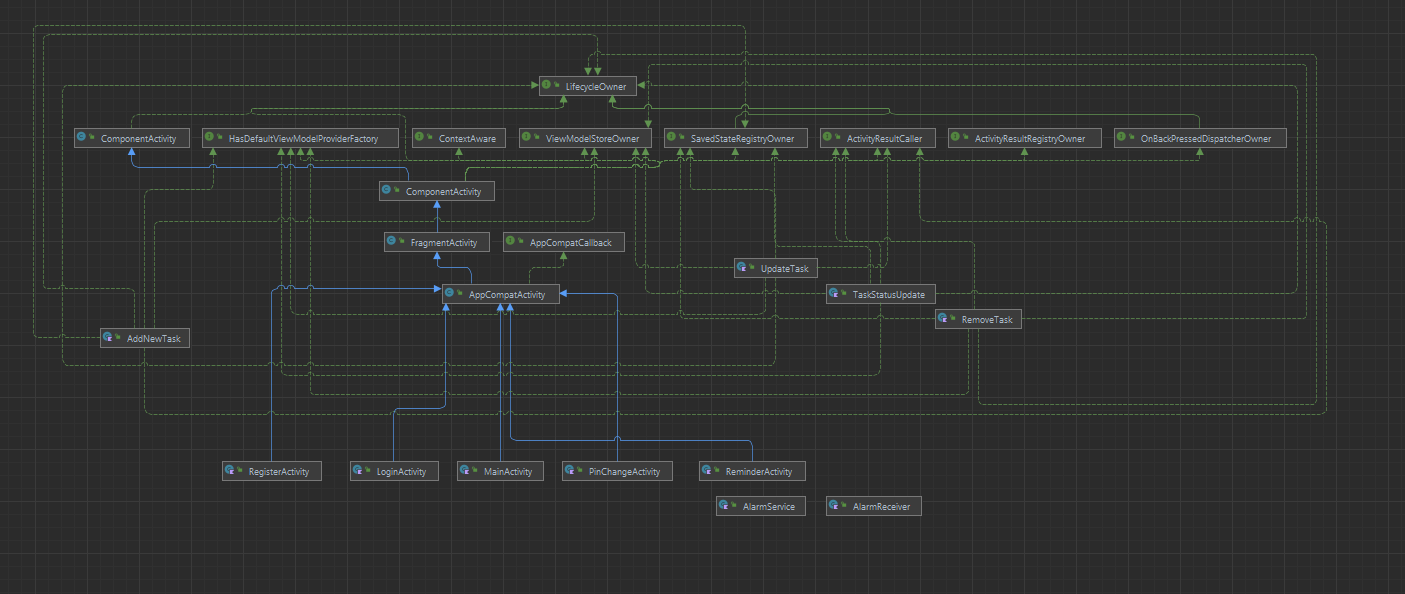
Obrázok, na ktorom je text

Automaticky generovaný popis

* Ďalej som využil **Navigation**, ktorý mi uľahčil presun medzi fragmentami, a tiež Args (navArgs , Parcelable Task) vďaka ktorým som mohol určiť po kliknutí na jednotlivý riadok RecyclerViewera, na ktorý task som klikol a následne s ním pracovať a pristupovať ku jeho hodnotám.



* Pri vytváraní notifikácie som použil **knižnicu Notify**, ktorá zjednodušila „skladanie“ notifikácie.
* Pri registrácií a prihlasovaní som využil **SharedPrefference**s, na ukladanie PINov, taktiež som SharedPrefferences využil aj pri spustení aktivity Register len prvý krat po štarte aplikácie.
* Na umiestnenie tlačidla na zmenu PINu som použil **ActionBar** a OptionMenu.



# **Zdroje**

Zdroje, ktoré mi pomohli pri tvorbe tejto aplikácie:

<https://developer.android.com/studio/write/image-asset-studio>

<https://stackoverflow.com/questions/7992216/android-fragment-handle-back-button-press?fbclid=IwAR3Jk_LqnfpajiqBGue7rtsk6Kmx4tJQ4y7eVwMJuvqeA-8h6P2KJT1qi9g>

<https://stackoverflow.com/questions/14382877/passing-parameters-to-the-broadcastreceiver>

<https://www.youtube.com/watch?v=44Nr3AT7fF4>

<https://www.youtube.com/watch?v=xSrVWFCtgaE>

<https://www.youtube.com/watch?v=nl-dheVpt8o&t=272s>

<https://www.youtube.com/watch?v=diaPAI0n-tw&t=1368s>

<https://www.youtube.com/watch?v=D0VpASTpgmw>

<https://stackoverflow.com/questions/71580441/the-emulator-process-for-avd-pixel-5-api-30-has-terminated>

<https://www.youtube.com/watch?v=HhmA9S53XV8>

<https://stackoverflow.com/questions/47742474/kotlin-call-a-function-to-update-ui-from-broadcastreceiver-onreceive>