

Žilinská univerzita v Žiline
Fakulta riadenia a informatiky

Aplikácia na správu projektov
SEMESTRÁLNA PRÁCA

Peter Čerešňa

2022/2023

OBSAH

<i>Analýza aplikácií</i>	3
<i>Analýza navrhovanej aplikácie</i>	4
<i>Návrh architektúry aplikácie</i>	5
<i>Ukážka návrhu obrazoviek</i>	6
<i>Implementácia riešenia</i>	9

Analýza aplikácií

Timesheet – time tracker

Kladné:

- Možnosť spravovať viac než projekty
- Funguje na hodinkách
- Mesačný prehľad, môže sa prehadzovať view na mesačný, týždenný alebo denný

Záporné:

- Neprehľadné
- Môže byť komplikované pre niektorých užívateľov
- Možnosť spravovať viac než projekty je v tomto prípade aj jedna z negatív

Zhrnutie:

Timesheet je skvelá aplikácia na spravovanie pokiaľ máte viac než len projekty, avšak v tomto prípade to môže byť mátku pre ostatných užívateľov, ktorý by si spravovať aj veci z osobného života. Hlavnou výhodou je, že aplikácia funguje aj na hodinkách, jednoducho stačí stlačiť tlačidlo na spustenie časovača.

Clockify

Kladné:

- Pekne spracované UI
- Možnosť spravovať rôzne projekty
- Priebežne reporty

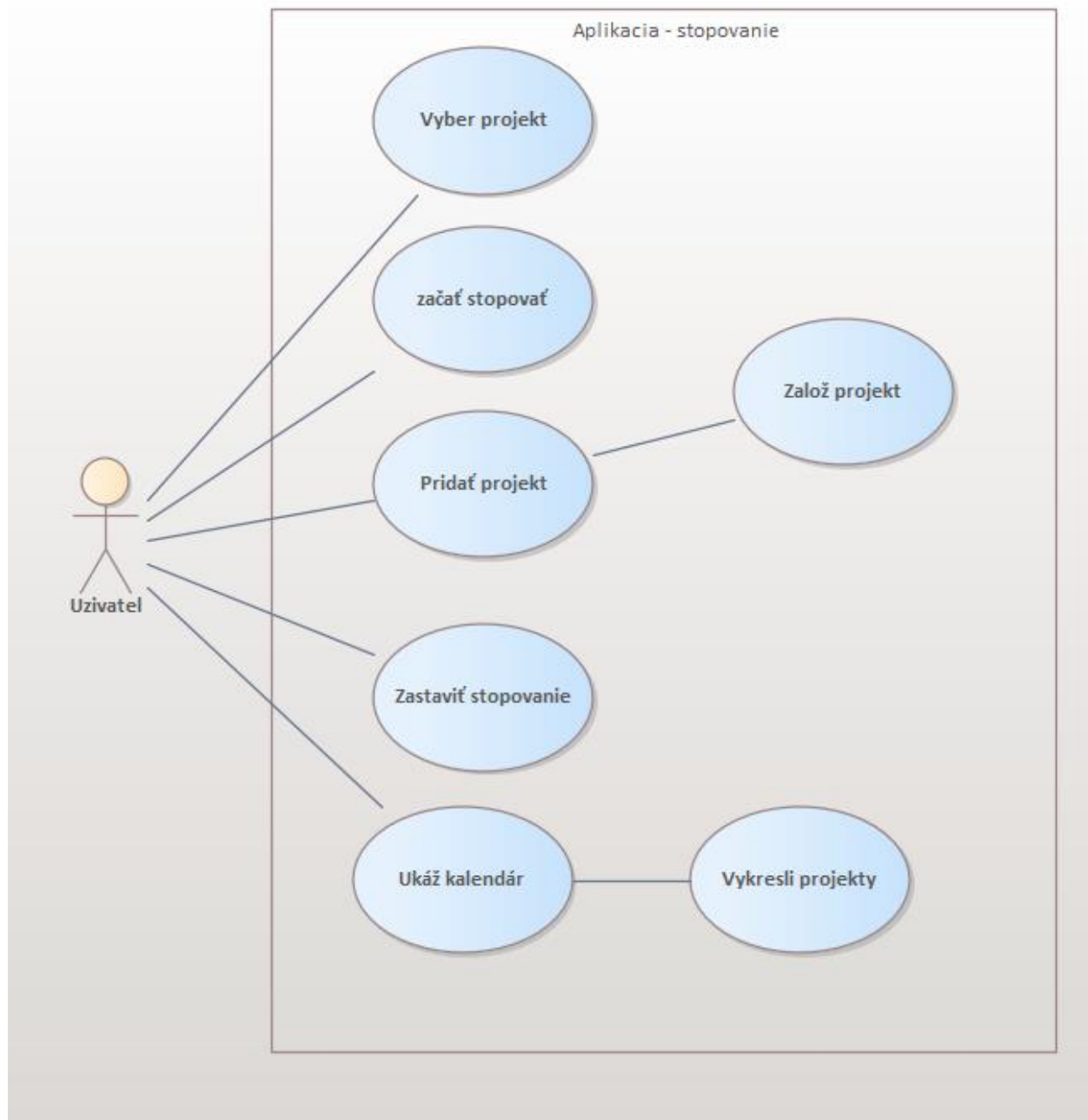
Záporné:

- Neobsahuje stopky ale časovač
- Miestami moc prepracované

Zhrnutie:

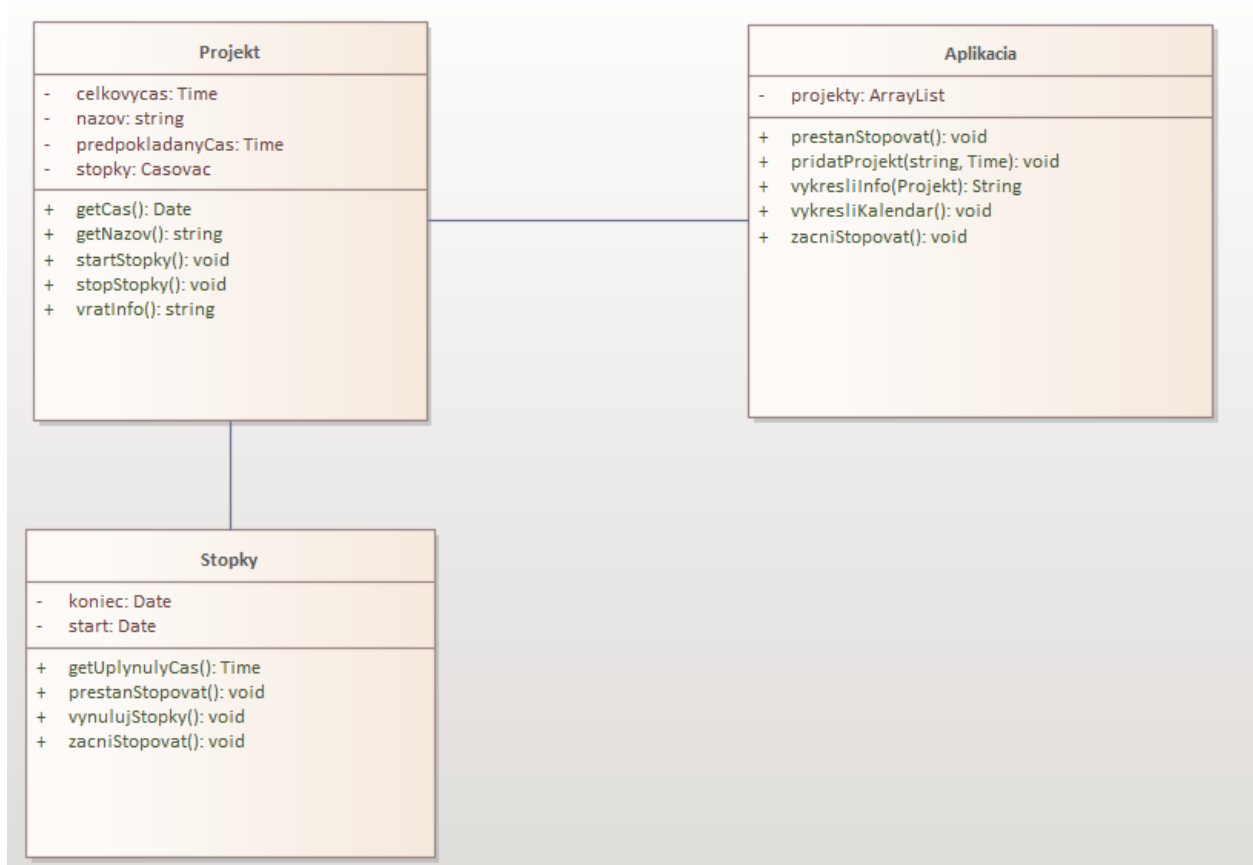
Clockify sa hodí v prípade, že potrebujete fixne dodržiavať daný predpis a čas strávený na projekte. Neprispôbuje sa dobre na náhle zmeny vzhľadom k tomu, že neobsahuje časovač a neprepne automaticky projekty. Pravidelné reporty sú však ale plusom v prípade keď chcete zistiť ako sa vám darí na danom projekte.

Analýza navrhovanej aplikácie



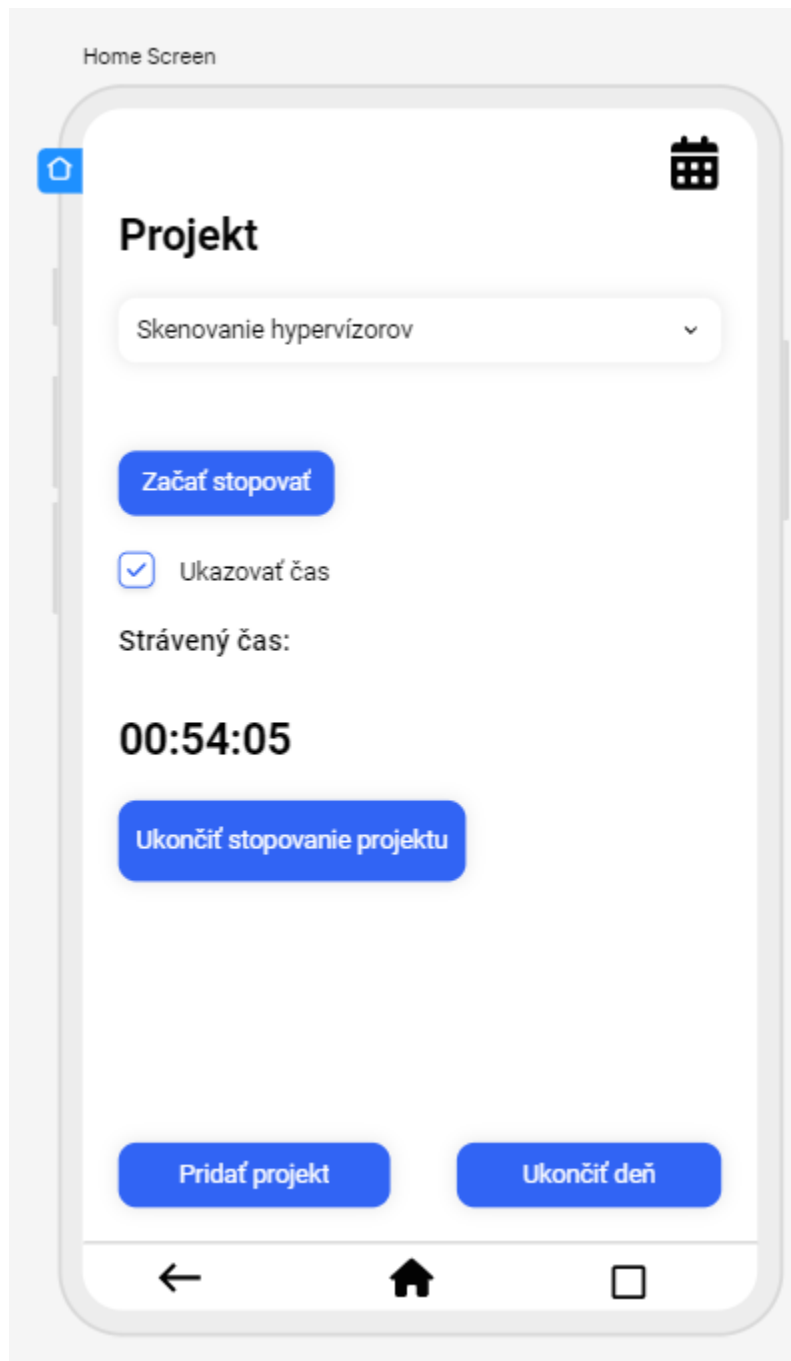
Navrhovaná aplikácia by vedela vybrať projekt z rôznych ďalších projektov, čo si užívateľ taktiež môže pridať. Najdôležitejšiou vecou je spúšťanie a zastavovanie stopovania projektu, kde automaticky sa hodnota na stopkách uloží do projektu, aby bolo následne možné vykresliť objekt do kalendára. Posledná vec, je ukázanie priebehu projektov za deň vo forme kalendára, kde vykreslí všetky projekty, ktoré má aplikácia k dispozícii.

Návrh architektúry aplikácie

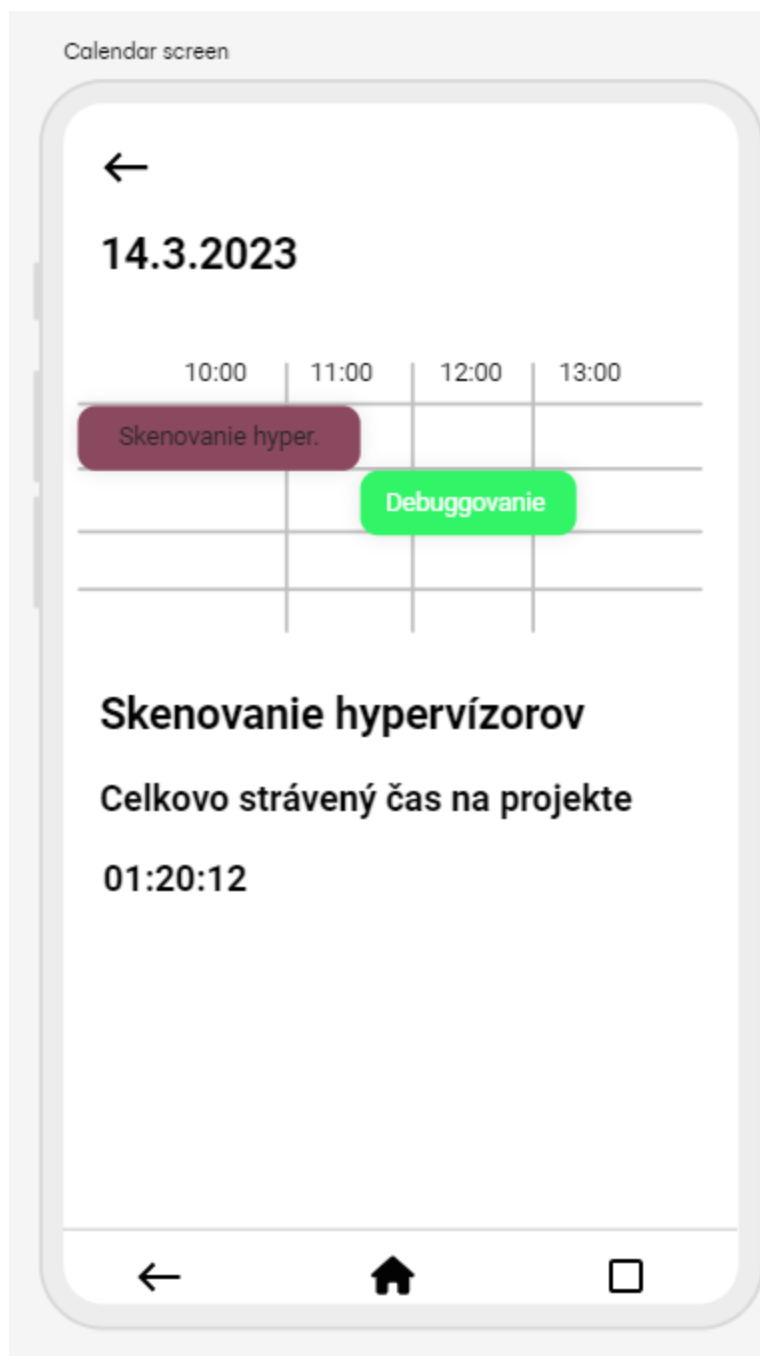


Aplikácia by mala 3 hlavné triedy. Trieda projekt, ktorá by si udržiavala informácie ako napríklad koľko času strávil užívateľ na danom projekte, predpokladaný čas, ktorý by užívateľ mal na projekte stráviť a názov samostatného projektu. Ideou je, že každý projekt si bude v sebe držať aj vlastné stopky aby sa vyhlo zbytočným komplikáciám so zdieľaním jedných stopiek. Významá trieda je Aplikácia kde bude prebiehať logika samostatnej aplikácie ako napríklad pridávanie projektov a podobne.

Ukážka návrhu obrazoviek



Domovská obrazovka – Na tejto obrazovke si užívateľ môže vybrať projekt na ktorom bude zrovna pracovať a spustiť alebo ukončiť stopovanie projektu. Užívateľ sa môže aj priamo rozhodnúť či má zobrazovať čas alebo nie. Taktiež je na výber možnosť pridať projekt, ktoré je možné riešiť aj cez widget namiesto buttonu. A nakoniec je možné zobrazovať projekty v kalendári.



Obrazovka s kalendárom – Táto obrazovka zobrazuje projekty vo forme Gantt Chartu. Po kliknutí na projekt zobrazí celkový strávený čas na projekte.

AddNew Project Screen

Názov projektu

Odhadovaný čas

☒ Pripomienka

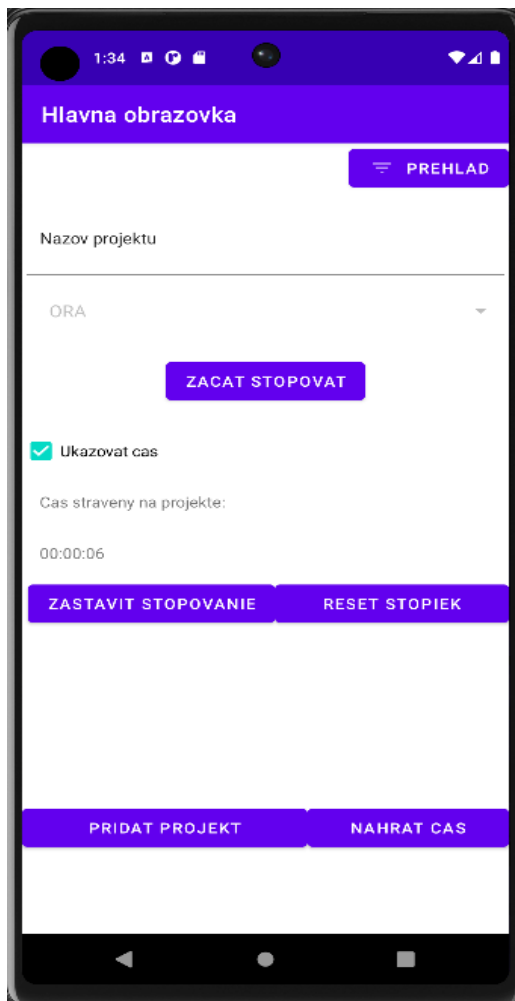
Pridať

← 🏠 □

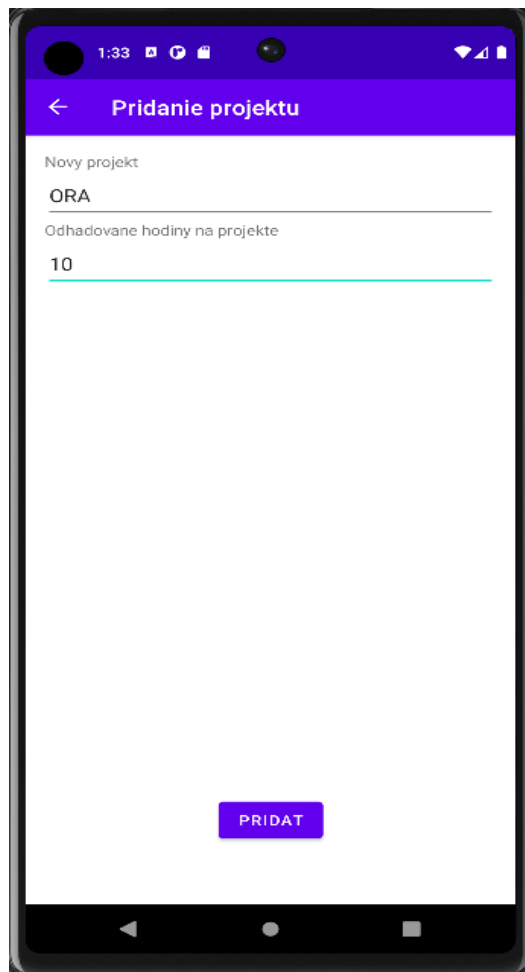
Obrazovka na pridanie projektu – Na tejto obrazovke si užívateľ môže pridať ľubovoľné projekty. Taktiež si môže nastaviť pripomienku, ktorá je formou notifikácie, ktorá upozorní užívateľa, že čas, ktorý strávil na projekte sa blíži predpokladanému času.

Implementácia riešenia

V práci sme využili jedne layout na hlavnú aktivitu aplikácie a niekoľko fragmentov na zobrazovanie rôznych obrazoviek. Obrazovky sú robené podľa návrhu vyššie, avšak nie vždy je možné pripraviť užívateľské rozhranie presne podľa predstáv. Aplikácia pozostáva z troch fragmentov. Fragmenty AddProject, MainScreen a Summary. MainScreen fragment predstavuje hlavnú obrazovku aplikácie, kde sa odohráva meranie času strávenom na projekte. Vo fragmente je možné spustiť stopky, zastaviť stopky alebo nahráť čas do databázy k danému projektu. Táto funkcionálnosť funguje vďaka Handleru, ktorý zabezpečuje počítanie sekúnd pomocou oneskorenia na 1000 milisekúnd, čo je 1 sekunda. Fragment MainScreen obsahuje taktiež komponent Spinner, ktorý uchováva v sebe názvy projektov, ktoré sa nachádzajú v databáze. Fragment taktiež obsahuje checkbox, podľa ktorého rozhoduje či zobrazovať alebo nezobrazovať meraný čas.



Ďalším fragmentom je AddProject. Tento fragment slúži na pridávanie nových projektov do databázy. Fragment obsahuje dva Inputy na zadanie mena projektu a odporúčaných hodín, ktoré by ste mali stráviť na projekte a jedno tlačidlo. Pokiaľ projekt s daným menom neexistuje, tak projekt nepridá a input na názov projektu vyhlási chybu. V opačnom prípade vytvorí nový projekt a pridá ho do databázy s tým, že budeme presmerovaný opäť na hlavnú obrazovku.



1:33

< Pridanie projektu

Nový projekt

ORA

Odhadované hodiny na projekte

10

PRIDAT

Posledným fragmentom je SummaryFragment. Tento fragment slúži na ukávanie prehľadu dostupných projektov. Na zobrazenie projektov využívame RecyclerView. Pre krajšie zobrazenie projektov využívame taktiež aj CardView. V danom CardView sa nachádzajú 3 TextView komponenty na zobrazenie informácií o danom projekte. O projekte sa môžeme dozvedieť jeho meno, odpracované hodiny vo formáte HH:MM:SS a odporúčané hodiny na projekt. Dole sa nachádza dodatočne tlačidlo na vymazanie projektov z databázy.



V celej aplikácii využívame AndroidViewModely na uchovávanie dát a správne správanie aplikácie pri rotovaní obrazovky a na správne fungovanie a kooperáciu s databázou.

```
package com.example.semestralka

import ...

class MainScreenViewModel(val database: ProjectDatabaseDao, application: Application) : AndroidViewModel(application) {
    private var projectsNames = MutableLiveData<List<String>>()

    var timeStarted = false
    private var handler = Handler(Looper.getMainLooper())
    lateinit var runnable: Runnable
    var seconds = 0
    val projects: MutableLiveData<List<String>> get() = projectsNames

    init {
        initializeProjects()
    }

    /**
     * Metoda, ktora spusti cas s oneskorenim 1000 milisek
     *
     * @param textView komponent TextView, ktora sa bude menit pri spusteni stopiek
     */
    fun startTimer(textView: TextView) {
        if (!timeStarted) {
            runnable = Runnable {
                seconds++
                textView.text = getTimeStringFromIntInSeconds(seconds)
                handler.postDelayed(runnable, delayMillis: 1000)
                Log.i(tag: "TIMER", msg: "Stale bezi")
            }
            handler.postDelayed(runnable, delayMillis: 1000)
            textView.text = getTimeStringFromIntInSeconds(seconds)
            timeStarted = true
        }
    }
}
```

Ako už bolo spomenuté využívame taktiež aj databázu vo forme RoomDatabase, ktorá funguje ako lokálna databáza zariadenia. Dáta z databázy slúžia aj ako LiveData pre dané fragmenty. Napríklad Spinner komponent bude vedieť kedy sa zmenili dáta a prispôbi sa podľa toho.