Dokumentácia

k semestrálnej práci z predmetu

Vývoj aplikácií pre mobilné zariadenia

PREHĽAD SPOTREBOVANÉHO PALIVA

**vypracoval:** Martin Ján Kulich

**študijná skupina:** 5ZYI22 dňa 9.6.2024

# Popis a analýza

Táto časť je venovaná popisu projektu, definícii problému a porovnaniu s podobnými aplikáciami.

## Motivácia

Množstvo dnešných áut je schopné zobraziť aktuálnu alebo priemernú spotrebu. Žiaľ, ani jedna s týchto možností neposkytuje presnú informáciu o spotrebovanom množstve pohonnej hmoty, či cene cesty. Aktuálna spotreba predstavuje spotrebu vozidla v danom momente, teda má krátkodobé opodstatnenie. Zatiaľ, čo priemerná spotreba je hodnota hovoriaca: pri svojich jazdách spotrebúvaš priemerne asi x l/100km. Z nej môžeme vyhodnotiť ako úsporná je naša jazda z dlhodobejšieho hľadiska. Avšak priemerná spotreba môže byť pre mnohých zavádzajúca. Napríklad, ak sme prešli 100 km a priemerná spotreba je 5 l/100km, neznamená to ešte, že sme spotrebovali 5 litrov paliva. Často spotrebujeme viac.

Pri mojich cestách by som chcel vedieť presne koľko paliva som spotreboval, a teda koľko ma daná cesta stála. Preto som sa rozhodol pre tento projekt.

## Popis + operácia

Úlohou aplikácie je poskytnúť čo najpresnejší prehľad o nákladoch spojených so spotrebovaným palivom pri automobilovej preprave. Umožňuje organizáciu dvoch druhov údajov: tankovania a precestované cesty.

Pre výpočet nákladov zásadne potrebuje dve informácie. Množstvo spotrebovaného paliva s jeho cenou a k tomu príslušnú dĺžku trasy.

Množstvo spotrebovaného paliva sa dá zistiť nasledujúcim spôsobom. Nádrž vozidla sa naplní doplna. Prejde sa určitá trasa. Nádrž sa opäť naplní doplna. Množstvo natankovaného paliva počas druhého čapovania zodpovedá presne spotrebovanému palivu na danú trasu. Z tohto vyplývajú dve hlavné obmedzenia projektu:

* Pre presné pracovanie je potrebné pri každom zastavení na čerpacej stanice naplniť celú nádrž.
* Aplikácia je schopná poskytnúť presné výsledky len medzi návštevami púmp. Z nich sa vypočítavajú menej presné hodnoty pre jednotlivé cesty.

Príslušná vzdialenosť medzi tankovaniami bude vypočítaná ako súčet prislúchajúcich dĺžok ciest.

## Podobné aplikácie

* Fuelio

<https://play.google.com/store/apps/details?id=com.kajda.fuelio>

Bola hlavnou inšpiráciu pri mojej tvorbe. Presne ako moja aplikácia umožňuje zadanie tankovaní a k nemu prejdených ciest. Zdá sa, že jednou z jej funkcií je aj vytvorenie trasy na základe GPS sledovania (podobne ako som aj ja chcel), avšak táto funkcia je na mojom zariadení nefunkčná alebo pre správny chod vyžaduje prídavné aplikácie z ekosystému vývojára. Pri tejto príležitosti spomeniem aj, že pri analýze v 7. týždni som v nej vedel pridať cestu bez problémov. To sa však nedá povedať o súčasnom stave, kedy sa mi akt vykonať nepodarilo. Niektoré funkcie sú schované za predplatným. Musím však pochváliť absenciu reklám.

* Zapisovač km od Driversnote a Drivvo

<https://play.google.com/store/apps/details?id=com.driversnote.driversnote>

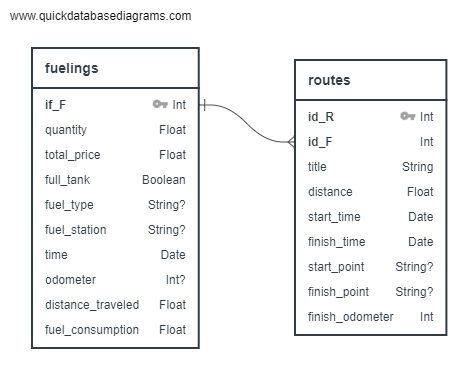
<https://play.google.com/store/apps/details?id=br.com.ctncardoso.ctncar&hl=sk>

Poskytujú služby zhodné s Fueliom. Prácu s nimi odmietam vyskúšať, pretože vyžadujú registráciu.

Uvedené aplikácie som vybral, pretože umožňujú zadávať cesty a tankovanie, s ktorých vytvárajú určitú časovú os, podobne ako moja aplikácia. Ostatné výsledky vyhľadávania v GooglePlay sú viac menej jednoduché kalkulačky spotreby.

# Návrh riešenia problému

Ako som už spomenul aplikácia uchováva záznamy o tankovaniach a cestách. Za ich organizáciu je zodpovedná lokálna databáza, ktorej dátový diagram je uvedený nižšie. Povšimnime si, že vzťah v ňom je: n ciest k jednému tankovaniu. Teda podporuje model na jedno natankovanie sa dá precestovať viac ciest.

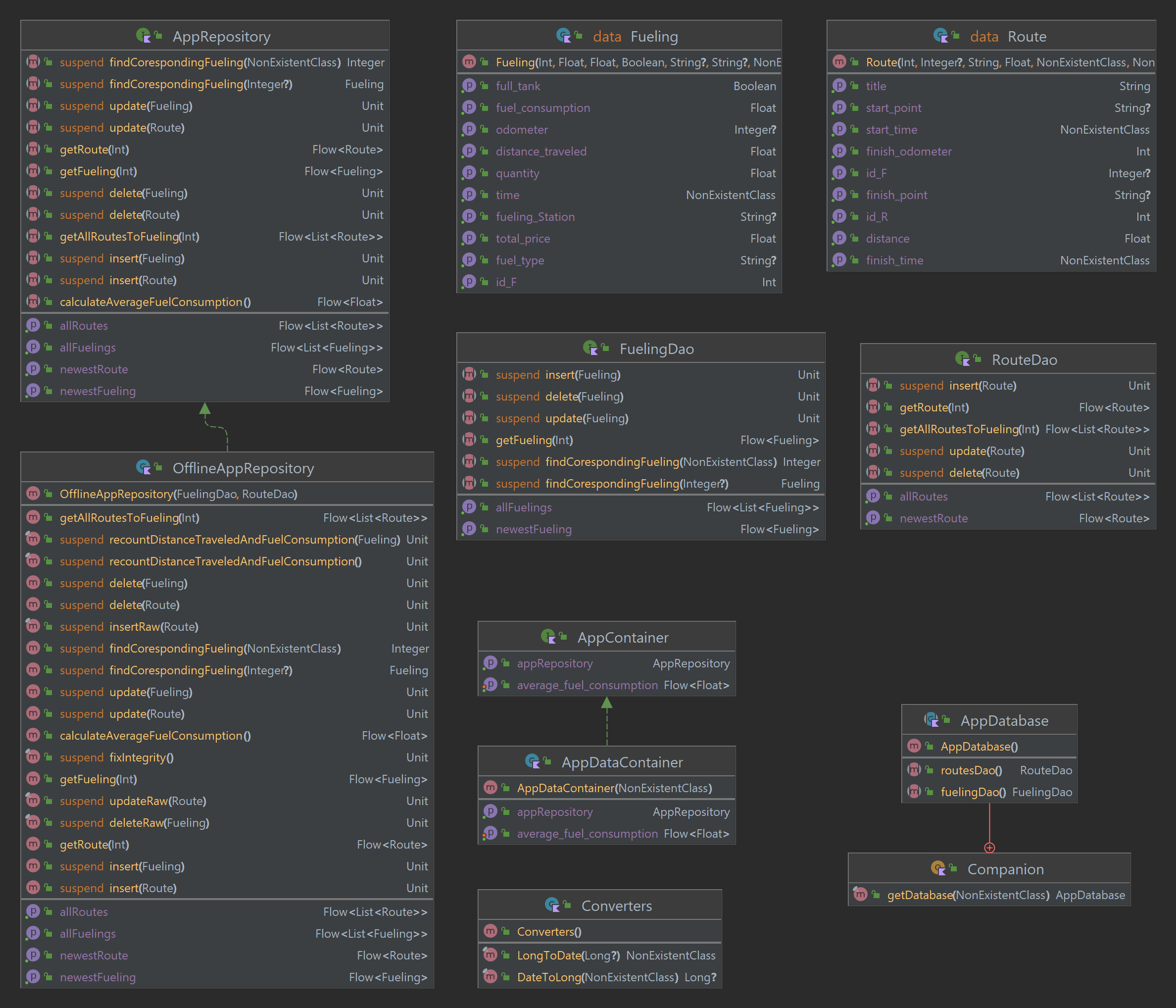


O dodržanie integrity sa stará OfflineAppRepository pracujúci s DAO vrstvou.

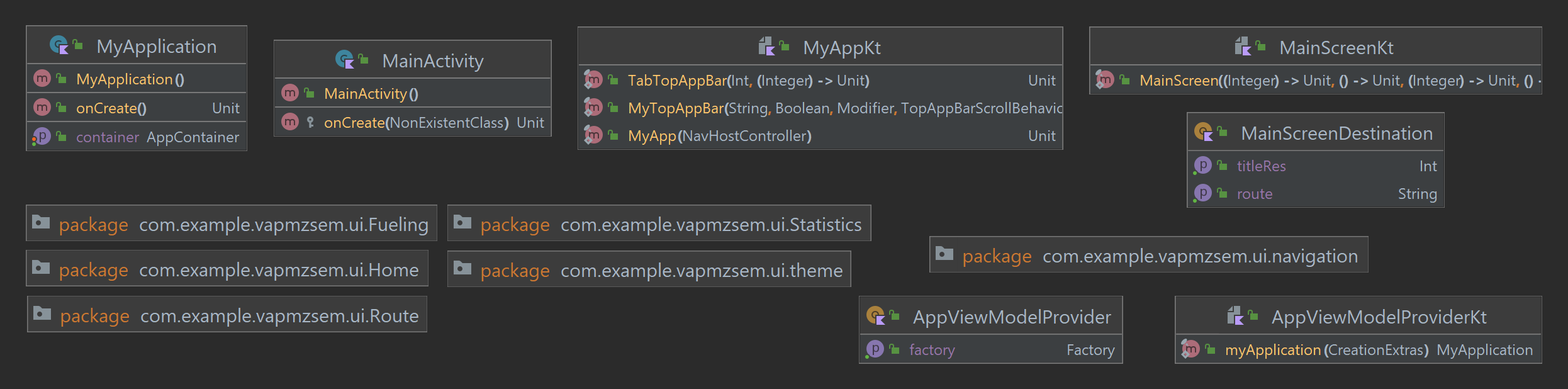
# Diagram tried

Tu sú vyobrazené diagramy tried. Jeden diagram by bol moc neprehľadný, preto každý diagram zodpovedá jednému balíčku

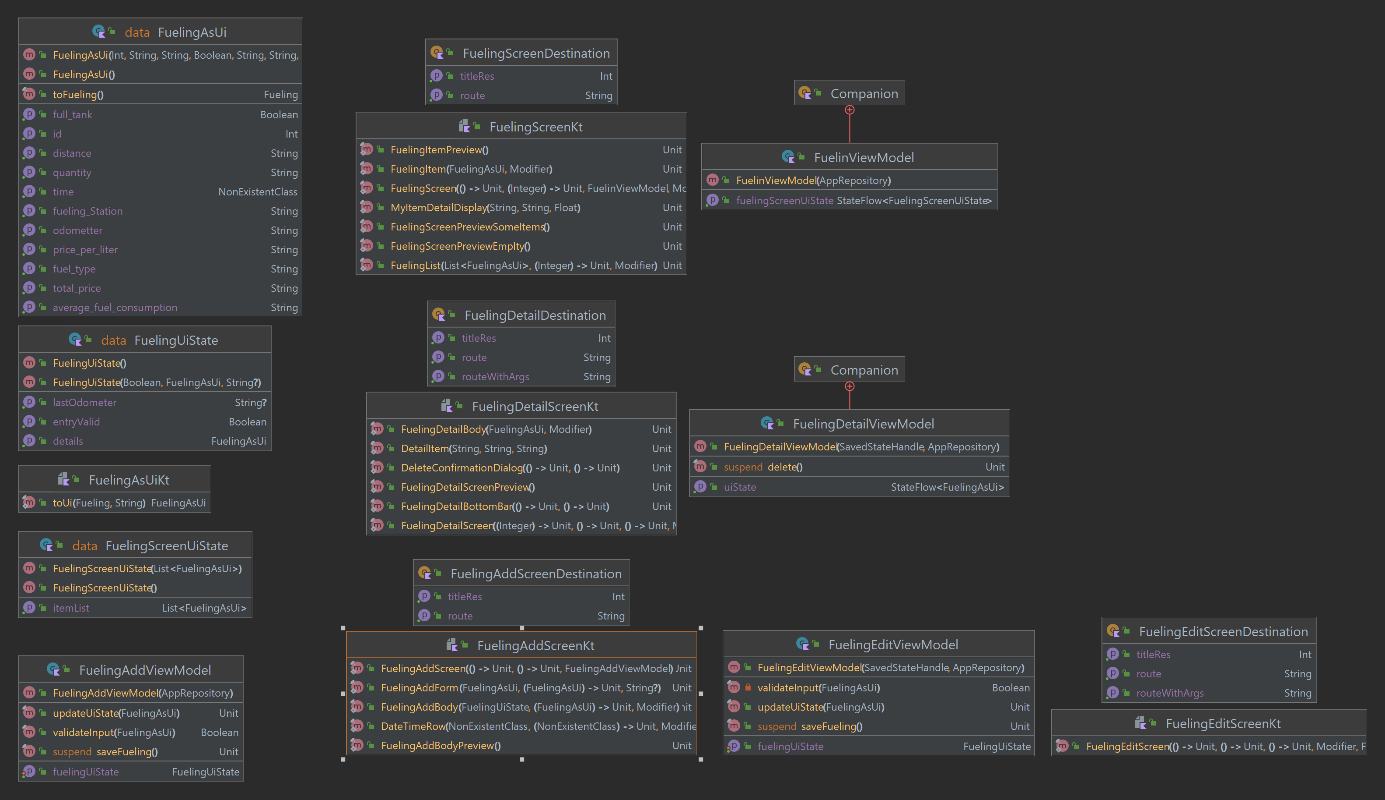
data



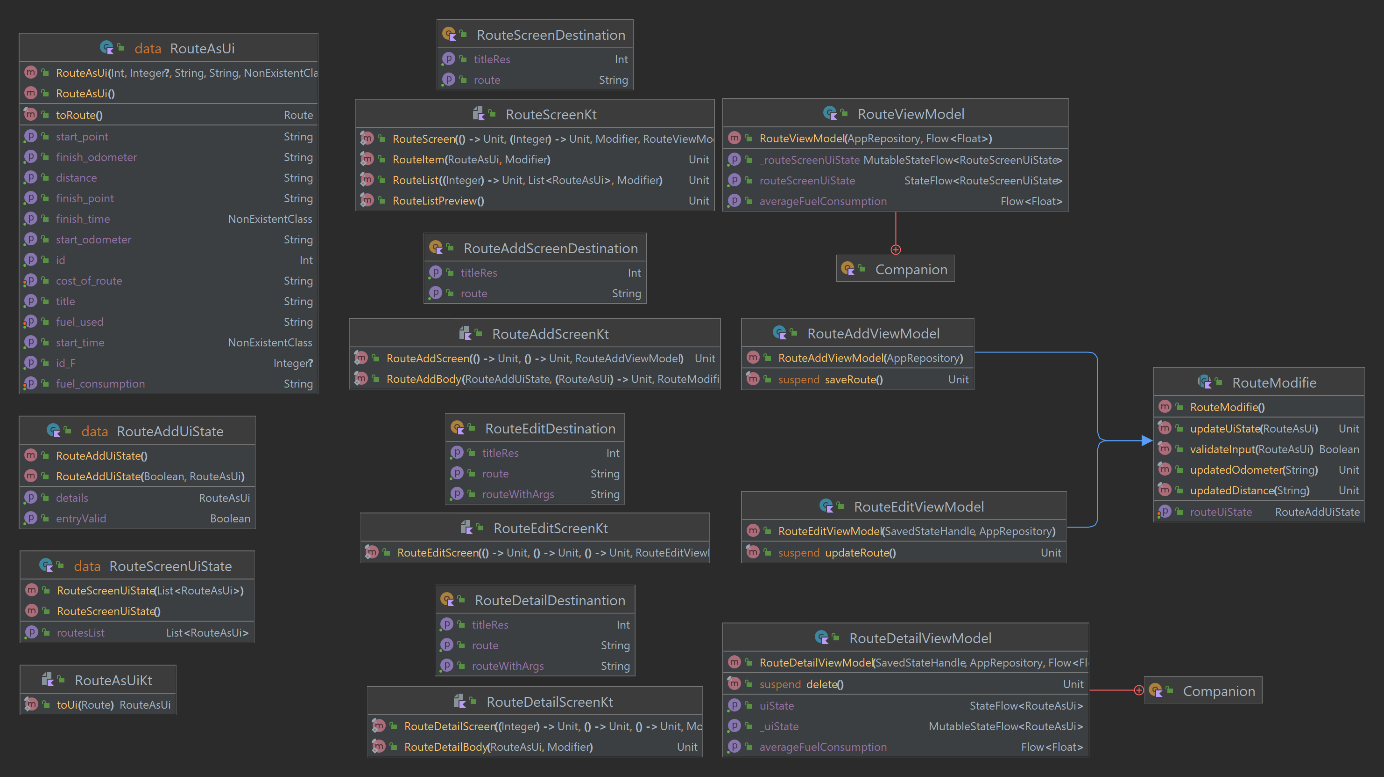
ui



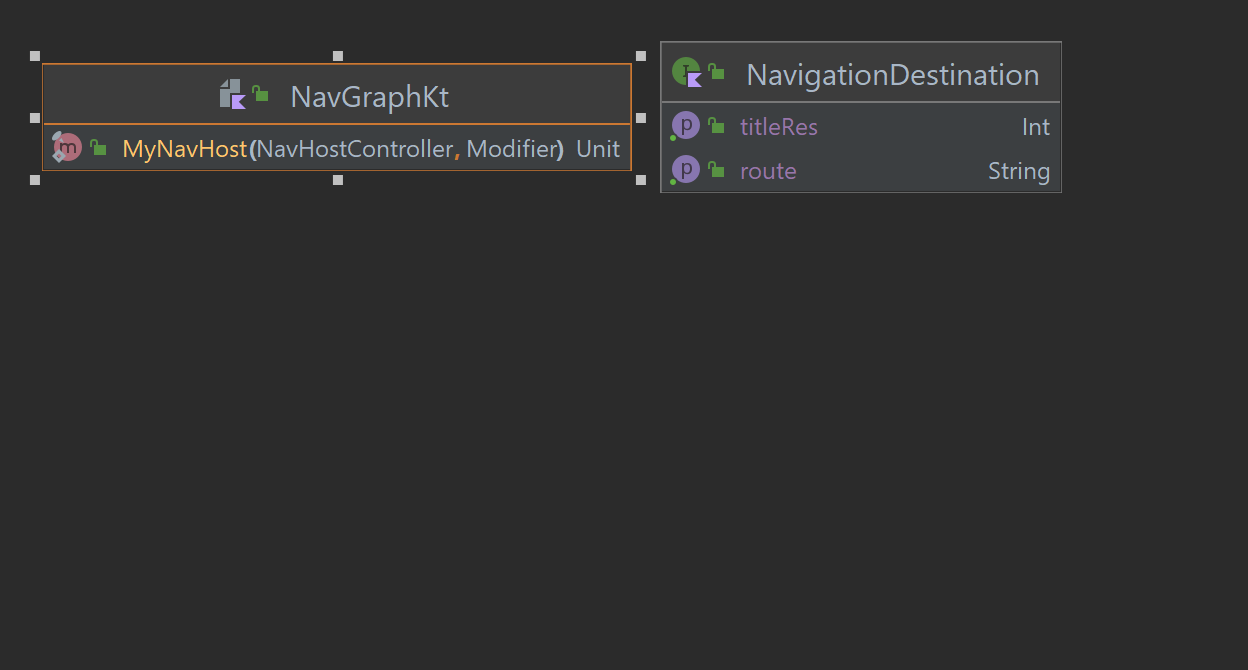
ui/Fueling



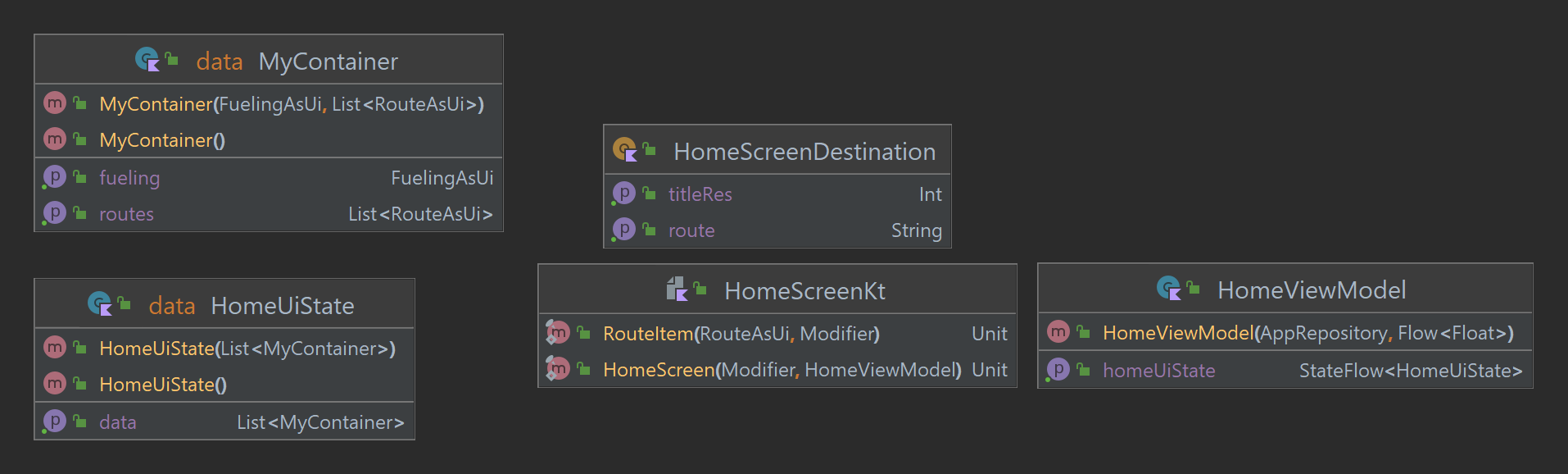
ui/Route



ui/navigation



ui/Home



# Popis implementácie

**LifeCycles** – využívam na vytvorenie a správu životného cyklu view modelov. Primárne však na dependenci injection. Tento prístup zabezpečuje tiež efektívne riadenie zdrojov.

Príklad:

object AppViewModelProvider {  
 val Factory = *viewModelFactory* **{** . . .// FuelongDetail  
 *initializer* **{** FuelingDetailViewModel(  
 this.*createSavedStateHandle*(),  
 *myApplication*().container.appRepository  
 )  
 **}**

**. . .**

**}**

@Composable  
fun FuelingDetailScreen(  
 . . .,  
 viewModel: FuelingDetailViewModel = viewModel(factory = AppViewModelProvider.Factory)  
){ . . . }

**Navigation** – využitý na navigáciu medzi obrazovkami. Každá ním navigovateľná obrazovka má svoj príslušný objekt implementujúci rozhranie NavigationDestination, podľa ktorého sa k nej naviguje.

**Room** – využitý na prácu s lokálnou databázou. Implementovaný podľa codelabu z 8. týždňa semestra. Umožňuje reaktívne aktualizácie UI pri zmene v databáze.

**ViewModel** – využitý na perzistentné držanie dát pre UI a enkapsuláciu biznis logiky. V podstatne pomáha každej obrazovke v aplikácií.