



**ŽILINSKÁ UNIVERZITA V ŽILINE**

Fakulta riadenia  
a informatiky

Semestrálna práca z predmetu  
*vývoj aplikácií pre mobilné zariadenia*

**AUTOMATE**

**Vypracoval:** Jakub Novák

**Študijná skupina:** 5ZYR23

**Akademický rok:** 2024/2025

V Žiline dňa 8. 6. 2025



## Obsah

|   |    |
|---|----|
| Zoznam tabuliek .....                     | 3  |
| Úvod .....                                | 4  |
| Prehľad podobných aplikácií .....         | 4  |
| Fuelio (1) .....                          | 4  |
| Drivvo (2) .....                          | 4  |
| Simply Auto (3) .....                     | 5  |
| Analýza navrhovanej aplikácie .....       | 5  |
| Hlavné funkcie .....                      | 5  |
| Role .....                                | 5  |
| Návrh architektúry aplikácie .....        | 5  |
| Návrh vzhľadu obrazoviek .....            | 6  |
| Skutočný návrh riešenia problému .....    | 8  |
| Krátka analýza .....                      | 8  |
| Návrh riešenia .....                      | 10 |
| Databáza .....                            | 11 |
| Balíčky .....                             | 11 |
| Používateľské rozhranie (UI) .....        | 11 |
| Navigácia .....                           | 11 |
| Popis implementácie .....                 | 12 |
| Obrazovky .....                           | 12 |
| AndroidX komponenty .....                 | 12 |
| Externé Frameworky / Knížnice .....       | 12 |
| Iné použité komponenty .....              | 13 |
| Možnosti budúceho rozšírenia .....        | 14 |
| Cloud .....                               | 14 |
| Grafické zobrazenie spotreby paliva ..... | 14 |
| Notifikácie .....                         | 14 |
| Zoznam zdrojov .....                      | 15 |



## Zoznam obrázkov

|  |    |
|--|----|
| Obrázok 1 - Hlavná obrazovka, zobrazí sa po zapnutí aplikácie..... | 6  |
| Obrázok 2 – Obrazovka na pridanie vozidla .....                    | 6  |
| Obrázok 3 – Obrazovka zvoleného vozidla.....                       | 6  |
| Obrázok 4 - Obrazovka úpravy informácií o vozidle.....             | 7  |
| Obrázok 5 - Obrazovka prehľadu paliva .....                        | 7  |
| Obrázok 6 - Obrazovka prehľadu údržby.....                         | 7  |
| Obrázok 7 – Use Case Diagram.....                                  | 8  |
| Obrázok 8 - Class Diagram.....                                     | 10 |

## Zoznam tabuliek

|                              |    |
|------------------------------|----|
| Tabuľka 1 - Obrazovky.....   | 12 |
| Tabuľka 2 - Komponenty ..... | 12 |
| Tabuľka 3 - Knižnice.....    | 12 |



## Úvod

Mobilná aplikácia AutoMate je určená pre každého motoristu, ktorý chce mať prehľad o svojom vozidle, najmä o spotrebe paliva, nákladoch na tankovanie a servisných úkonoch. Hlavnou motiváciou pre vytvorenie aplikácie bolo zjednodušiť a zautomatizovať evidenciu údajov o vozidle, ktoré si používatelia často zapisujú manuálne.

Ja sám som aktívny vodič a doteraz som si údaje o tankovaní a výdavkoch zapisoval do Excel tabuľky. Myslím si, že aplikácia by bola oveľa praktickejšie a efektívnejšie riešenie.

## Prehľad podobných aplikácií

### Fuelio (1)

#### Funkcie

- Zaznamenávanie tankovania, servisov, nákladov, jžd.
- Výpočet spotreby, grafy a štatistiky.
- Export/import údajov, záloha cez Google Disk.
- GPS tracking a mapa výdavkov.

#### Výhody

- Prehľadné a intuitívne prostredie.
- Možnosť synchronizácie medzi zariadeniami.
- Viacero áut, podpora viacerých mien a jednotiek.

#### Nevýhody

- Pokročilejšie funkcie sú v platenej verzii.
- Reklamy v bezplatnej verzii.

### Drivvo (2)

#### Funkcie

- Evidencia tankovania, opráv, daní, poistenia.
- Pripomienky na plánované úkony (servis, STK).
- Export do Excelu, cloudu.

#### Výhody

- Vhodné aj pre firemné vozidlá.
- Rozšírené možnosti filtrovania a hľadania záznamov.
- Prispôsobiteľné kategórie výdavkov.

#### Nevýhody

- Aplikácia je viac zameraná na podnikateľov, nie vždy intuitívna pre bežného používateľa.
- Bezplatná verzia má obmedzený export a synchronizáciu.

## Simply Auto (3)

### Funkcie

- Evidencia tankovania, servisu, výdavkov, jász.
- Automatický import údajov z SMS.
- Zdieľanie údajov medzi viacerými používateľmi.

### Výhody

- Veľmi podrobné štatistiky a prehľady.

### Nevýhody

- Aplikácia je čiastočne v angličtine, nie všetky funkcie sú lokalizované.
- Náročnejšie ovládanie pre začiatočníkov.

Popisované aplikácie ponúkajú množstvo funkcií pre evidenciu výdavkov spojených s vozidlom. Sú robustné, často využívajú cloudové riešenia a cieľia aj na firemných používateľov. Výhodou je ich funkčná vyspelosť, avšak často sú dostupné len v prémiových verziách, obsahujú reklamy alebo majú zložitejšie používateľské rozhranie.

Oproti nim bude aplikácia AutoMate jednoduchšia a zameraná najmä na bežného používateľa. Jej hlavným cieľom je poskytnúť prehľad o spotrebe paliva a výdavkoch bez zbytočných komplikácií. Bude fungovať offline, bez reklám, a dáta sa budú ukladať lokálne do textového súboru. Používateľské rozhranie bude navrhnuté s dôrazom na jednoduchosť, prehľadnosť a rýchle zadávanie údajov.

## Analýza navrhovanej aplikácie

### Hlavné funkcie

- Pridanie vozidla – je možné pridať viacero vozidiel
- Pridanie tankovania – dátum, množstvo, cena, stav km
- Zobraziť štatistiky spotreby paliva
- Evidovanie servisných kontrol
- Upozornenie na servis/technickú kontrolu

### Role

- Používateľ – jediná rola, ktorá zadáva a zobrazuje údaje

## Návrh architektúry aplikácie

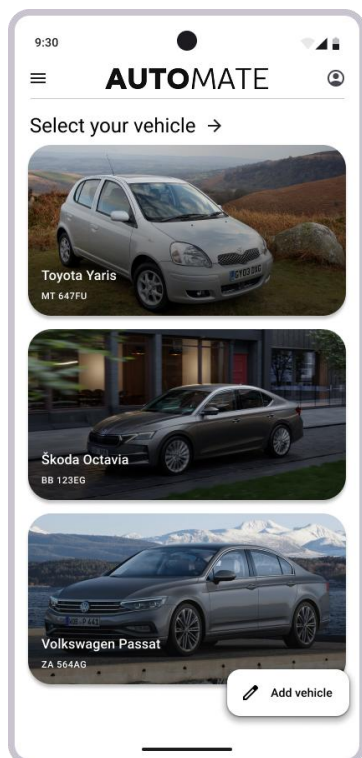
Architektúru tvorí jednoduché rozdelenie na UI, logiku a prácu s dátami. Využijem Kotlin s Jetpack Compose UI toolkit. Dáta sa budú ukladať do textového súboru.

Z dát o palive budú vypočítavané nasledujúce metriky:

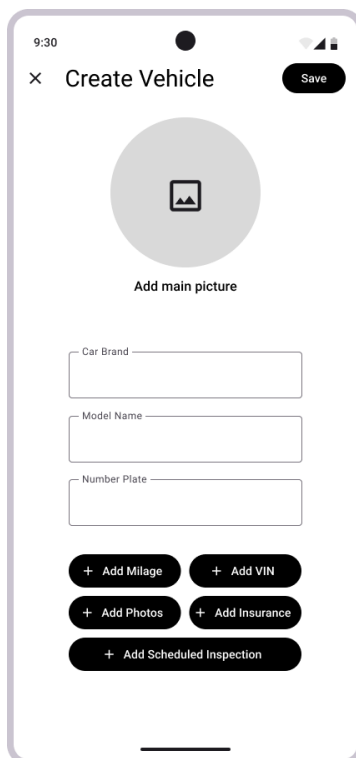
- Spotreba paliva (medzi tankovaniami, alebo na časovom úseku)
- Priemerná cena paliva
- Priemer nákladov na tankovanie
- Priemerné množstvo (v litroch) natankovaného paliva

Z dát o údržbe sa budú rátať celkové náklady.

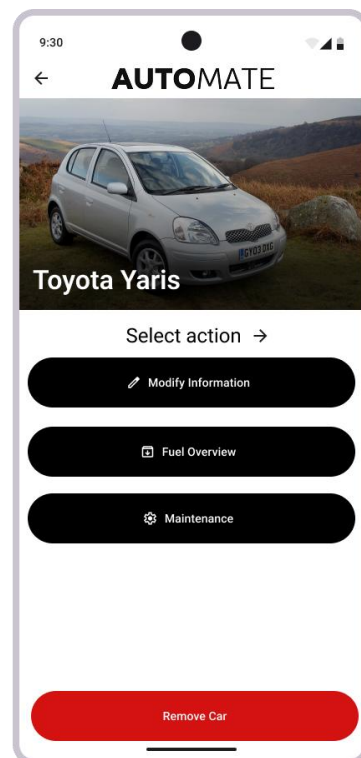
## Návrh vzhľadu obrazoviek



Obrázok 1 - Hlavná obrazovka, zobrazí sa po zapnutí aplikácie



Obrázok 2 – Obrazovka na pridanie vozidla



Obrázok 3 – Obrazovka zvoleného vozidla

Obrázok 1 – Hlavná obrazovka, zobrazuje vozidlá, dajú sa tu pridať vozidlá tlačidlom „Add vehicle“, po kliknutí na niektoré z vozidiel sa ukáže obrazovka na obrázku 3.

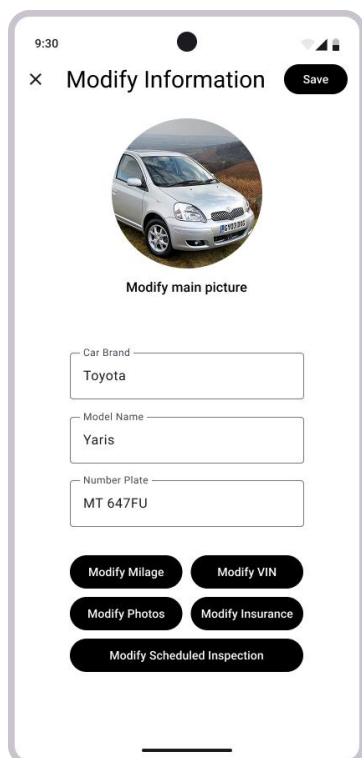
- LazyColumn – pre zobrazenie áut
- Text – „Select your vehicle“
- FloatingActionButton – „Add vehicle“

Obrázok 2 – Obrazovka vytvárania vozidla, zadávajú sa tu informácie o vozidle.

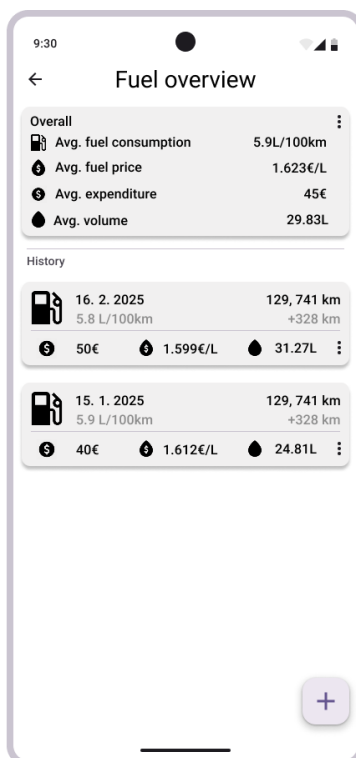
- ImagePicker (knižnica) – výber obrázku
- OutlinedTextField – značka, model, ŠPZ
- Button – pridať nájazd, VIN, poistenie..., Save

Obrázok 3 – Obrazovka, na ktorej sa dajú zvoliť akcie, ktoré sú možné vykonať s vozidlom.

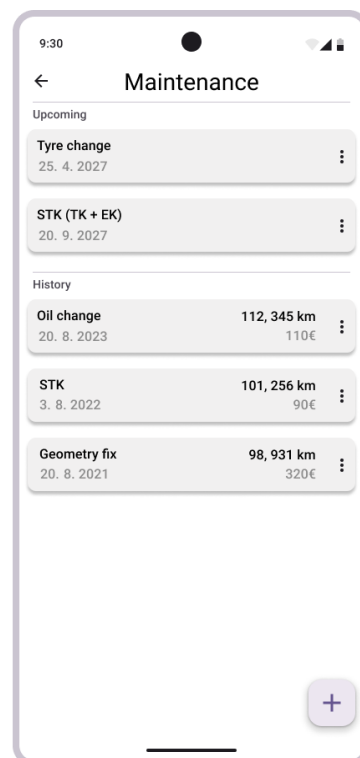
- Image, Text
- Button – Modify, Fuel, Maintenance, Remove Car



Obrázok 4 - Obrázovka úpravy informácií o vozidle



Obrázok 5 - Obrázovka prehľadu paliva



Obrázok 6 - Obrázovka prehľadu údržby

Obrázok 4 – Obrázovka na ktorej sa dajú zmeniť informácie o vozidle

- ImagePicker (knižnica) – výber obrázku
- OutlinedTextField – značka, model, ŠPZ
- Button – pridať nájazd, VIN, poistenie..., Save

Obrázok 5 – Obrázovka prehľadu paliva, zobrazujú sa tu individuálne informácie o jednotlivých tankovaniach, taktiež aj informácie za celú dobu používania vozidla alebo v určitom časovom úseku

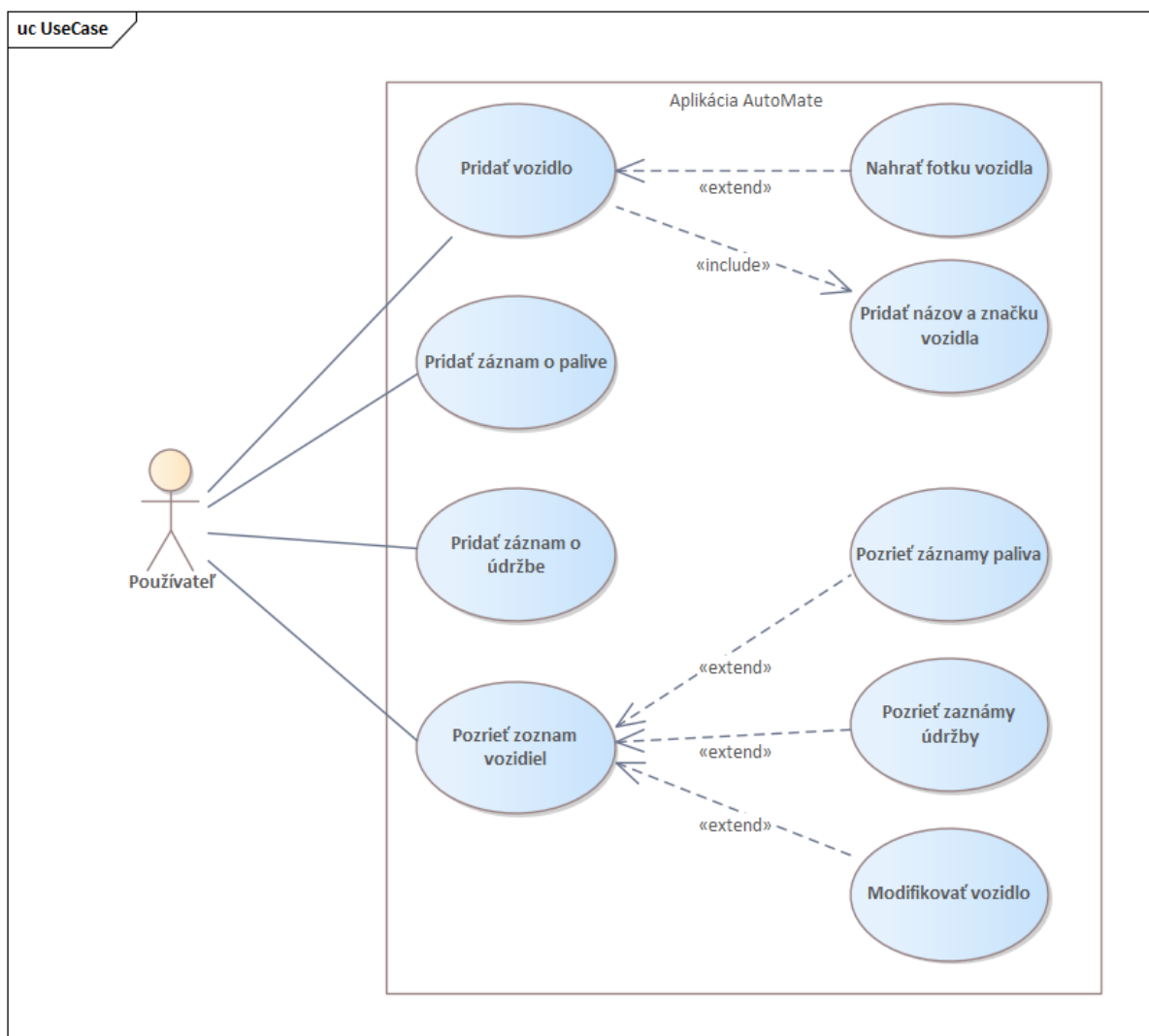
- Card, Text, Icon, Divider
- LazyColumn – zoznam tankovaní
- FloatingActionButton – pridať nové tankovanie

Obrázok 6 – Obrázovka prehľadu údržby, zobrazuje nasledujúce plánované údržby a aj históriu údržieb

- Card, Text, Icon, Divider
- LazyColumn – zoznam údržieb
- FloatingActionButton – pridať novú údržbu

## Skutočný návrh riešenia problému

### Krátka analýza



Obrázok 7 – Use Case Diagram

Aplikácia AutoMate umožňuje pridávať viacero vozidiel a ku každému zaznamenávať tankovania aj servis. Existuje len jedna hlavná rola, a to je používateľ, ktorý vykonáva všetky operácie nie je potrebné rozlišovať administrátorov či iné role.

### Používateľ má v aplikácii nasledujúce možnosti:

#### 1. Pridanie nového vozidla

Na úvodnej obrazovke *VehicleScreen* užívateľ môže kliknúť na tlačidlo „Add Vehicle“. Zobrazí sa obrazovka *AddVehicleScreen* kde používateľ vyplní nasledujúce údaje: značka, model, ŠPZ, aktuálny stav odometra, VIN číslo, vek vozidla, obrázok vozidla



## 2. Zobrazenie vozidla

Po kliknutí na kartu zvoleného vozidla sa zobrazí *VehicleDetailScreen*, kde sú dostupné možnosti:

- Tlačidlo „Modify Information“ – otvorí obrazovku na úpravu vozidla
- Tlačidlo „Fuel Overview“ – otvorí prehľad záznamov o tankovaní
- Tlačidlo „Maintenance“ – otvorí prehľad záznamov údržby
- Tlačidlo „Remove Vehicle“ – odstráni vozidlo a všetky záznamy o nich

## 3. Pridanie záznamu o tankovaní

Na obrazovke *FuelOverviewScreen*, ktorá sa dá otvoriť cez tlačidlo „Fuel Overview“, po kliknutí na tlačidlo „+“, môže užívateľ na obrazovke *AddFuelScreen* vyplniť:

- Dátum tankovania
- Stav odometra
- Objem natankovaného paliva
- Cena za tankovanie
- Poznámka

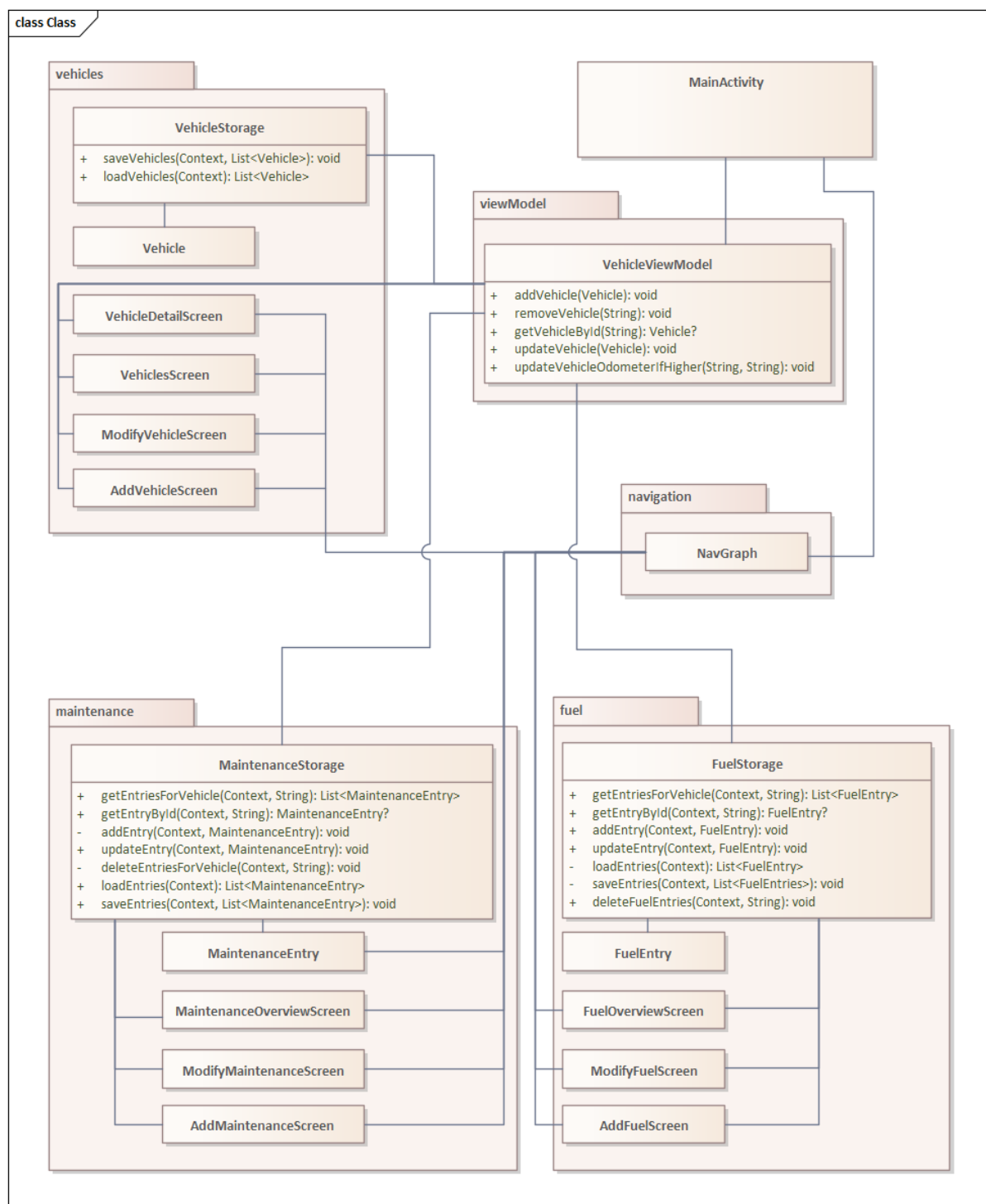
Následne sa na *FuelOverviewScreen* zobrazí daný záznam a celkové štatistiky o tankovaní.

## 4. Pridanie záznamu o údržbe

Na obrazovke *MaintenanceOverviewScreen*, sa zobrazia všetky záznamy údržby pre dané vozidlo, po kliknutí na tlačidlo „+“ môže užívateľ na obrazovke *AddMaintenanceScreen* zadať:

- Typ údržby
- Dátum
- Stav odometra
- Cena
- Poznámka

## Návrh riešenia



Obrázok 8 - Class Diagram

Aplikácia AutoMate bola vyvinutá v jazyku Kotlin s využitím Jetpack Compose ako nástroja pre deklaratívnu tvorbu používateľského rozhrania. Aplikácia funguje offline a ukladá údaje lokálne do vnútorného úložiska zariadenia.

Logická vrstva je postavená na architektonickom vzore MVVM (Model–View–ViewModel). V projekte je definovaný komponent *VehicleViewModel*, ktorý predstavuje jediný zdroj pravdy (StateFlow) pre stav aplikácie a sprostredkováva ho všetkým obrazovkám(4).



## Databáza

Ako dátovú vrstvu som zvolil lokálne úložisko vo forme JSON súborov. Aplikácia používa 3 JSON súbory:

- *vehicles.json*
- *fuel\_entries.json*
- *maintenance\_entries.json*

Prístup k týmto súborom je zapuzdrený v triedach *VehicleStorage*, *FuelStorage* a *MaintenanceStorage*. Na serializáciu a deserializáciu údajov sa používa knižnica *Gson*. Všetky dáta sú uložené vo vnútornom úložisku Androidu, čo zaručuje, že sú súkromné a neprístupné pre iné aplikácie – čo zodpovedá odporúčaniam pre bezpečné ukladanie údajov (5).

## Balíčky

Projekt je rozdelený do viacerých balíčkov:

- *vehicles* – obsahuje dátovú triedu *Vehicle*, obrazovky na zobrazenie a úpravu vozidiel a úložisko *VehicleStorage*
- *fuel* – obsahuje dátovú triedu *FuelEntry*, a príslušné obrazovky pre zobrazenie údajov o palive
- *maintenance* – obsahuje dátovú triedu *MaintenanceEntry*, a príslušné obrazovky pre zobrazenie údajov o údržbe
- *viewModel* – obsahuje *VehicleViewModel*, ktorý uchováva stav aplikácie a komunikuje s jednotlivými komponentami
- *navigation* – definuje navigačný graf *NavGraph* a prepája obrazovky cez roupe parametre

## Používateľské rozhranie (UI)

Rozhranie je implementované pomocou *Jetpack Compose*, každá obrazovka je funkcia typu *@Composable*, ktorá reaguje na zmeny stavu z *VehicleViewModel*.

## Navigácia

Navigácia je riešená pomocou *Jetpack Navigation Compose*, každá obrazovka má vlastný route.



## Popis implementácie

Aplikácia AutoMate bola vyvinutá v prostredí AndroidStudio s jazykom Kotlin. Pre UI som využil framework JetpackCompose. Minimálna verzia androidu pre AutoMate je Android 7.0.

Aplikácia funguje správne pri otočení obrazovky na výšku aj na šírku, ale preferovaný spôsob použitia je orientácia na výšku. Pri vyplňaní textových polí ostanú polia vyplnené aj po otočení obrazovky (6).

### Obrazovky

Tabuľka 1 - Obrazovky

| Obrazovka                 | Funkcia                               |
|---------------------------|---------------------------------------|
| VehiclesScreen            | Zoznam vozidiel                       |
| AddVehicleScreen          | Pridanie nového vozidla               |
| VehicleDetailScreen       | Detail vozidla + odkazy na podzáznamy |
| FuelOverviewScreen        | Zobrazenie záznamov o tankovaní       |
| AddFuelScreen             | Pridanie záznamu o tankovaní          |
| ModifyFuelScreen          | Úprava záznamu o tankovaní            |
| MaintenanceOverviewScreen | Zobrazenie záznamov o údržbe          |
| AddMaintenanceScreen      | Pridanie záznamu o údržbe             |
| ModifyMaintenanceScreen   | Úprava záznamu o údržbe               |

### AndroidX komponenty

Tabuľka 2 - Komponenty

| Komponent      | Použitie   |
|----------------|--|
| ViewModel (7)  | VehicleViewModel na správu stavu                           |
| Navigation (8) | Jetpack Navigation Compose                                 |
| LifeCycle (9)  | StateFlow pozorovaný pomocou collectAsStateWithLifecycle() |

### Externé Framworky / Knižnice

Tabuľka 3 - Knižnice

| Knižnica  | Funkcia                            |
|-----------|------------------------------------|
| Gson (10) | Serializácia a deserializácia JSON |



## Iné použité komponenty

### **Scaffold**

Použitie – Napr. *VehiclesScreen*

Účel – Poskytuje základný rozvrh obrazovky

### **LazyColumn**

Použitie - *FuelOverviewScreen*, *MaintenanceOverviewScreen*, *VehiclesScreen*

Účel – Efektívne zobrazenie záznamov s viacerými kartami

### **FloatingActionButton**

Použitie – Napr. *VehiclesScreen*

Účel – Pridanie nového záznamu (napr. Vozidlo, tankovanie)

### **DatePickerDialog**

Použitie – Napr. *AddFuelScreen*, *AddMaintenanceScreen*

Účel – Vybratie dátumu pomocou štandardného Androidového dialógu

### **LocalContext.current**

Použitie – Vo väčšine obrazoviek, kde sa načítavajú alebo zapisujú hodnoty zo súboru JSON

Účel – Získanie kontextu pre prácu so súborom



## Možnosti budúceho rozšírenia

### Cloud

Momentálne je celá databáza v internom úložisku, integrácia cloudu by umožnila zálohovanie dát, prístup viacerých zariadení a jednoduché obnovenie pri výmene telefónu.

### Grafické zobrazenie spotreby paliva

Ďalšie dobré vylepšenie by bola grafická vizualizácia pri spotrebe paliva. Užívateľ by tak mohol ľahko porovnávať jednotlivé tankovania.

### Notifikácie

Aplikácia by mohla používateľa upozorniť na nadchádzajúce servisné termíny na základe zadaného dátumu alebo stavu odometra.



## Zoznam zdrojov

1. **Fuelio.** Fuelio - Vehicle Log App. *Fuelio*. [Online] 2025. <https://www.fuel.io/>.
2. **Drivvo.** Manage your vehicle expenses. *Drivvo*. [Online] 04. 04 2025. <https://drivvo.com/>.
3. **Simply Auto.** Simply Auto - Car Maintenance App. *Simply Auto*. [Online] 04. 04 2025. <https://simplyauto.app/>.
4. **Android Developers.** Guide to app architecture – Architecture best practices. *Android Developers*. [Online] Google. <https://developer.android.com/topic/architecture>.
5. **OWASP Foundation.** Testing Data Storage – Overview. *Mobile Application Security Testing Guide (MASTG)*. [Online] OWASP Foundation.
6. **Android Developers.** Handle configuration changes. *Android Developers*. [Online] Google. <https://developer.android.com/guide/topics/resources/runtime-changes>.
7. —. Understand ViewModel. *Android Developers*. [Online] Google. <https://developer.android.com/topic/libraries/architecture/viewmodel>.
8. —. Navigation in Jetpack Compose. *Android Developers*. [Online] Google. <https://developer.android.com/jetpack/compose/navigation>.
9. —. Lifecycle-aware components in Jetpack Compose. *Android Developers*. [Online] Google. <https://developer.android.com/jetpack/compose/lifecycle>.
10. **Google.** Gson – GitHub Repository. *GitHub*. [Online] GitHub, Inc. <https://github.com/google/gson>.
11. **Android Developers.** Store app-specific files. *Android Developers*. [Online] Google. <https://developer.android.com/training/data-storage/app-specific>.
12. —. State in Jetpack Compose – Flows. *Android Developers*. [Online] Google. <https://developer.android.com/develop/ui/compose/state#flows>.
13. —. Images in Jetpack Compose. *Android Developers*. [Online] Google. <https://developer.android.com/jetpack/compose/graphics/images>.
14. —. MaterialDatePicker. *Android Developers*. [Online] Google. <https://developer.android.com/reference/com/google/android/material/datepicker/MaterialDatePicker>.
15. **Oracle.** Class UUID. *Java Platform, Standard Edition 8 API Specification*. [Online] Oracle Corporation. <https://docs.oracle.com/javase/8/docs/api/java/util/UUID.html>.
16. **Android Developers.** Scaffold layout in Compose. *Android Developers*. [Online] Google. <https://developer.android.com/develop/ui/compose/components/scaffold>.
17. —. Floating action button (FAB) in Compose. *Android Developers*. [Online] Google. <https://developer.android.com/develop/ui/compose/components/fab>.
18. —. Lists in Compose. *Android Developers*. [Online] Google. <https://developer.android.com/develop/ui/compose/lists>.