



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MOBILNÁ APLIKÁCIA PRE ZAZNAMENÁVANIE  
SPOTREBY AUTOMOBILU**

MOBILE APP FOR LOGGING OF AUTOMOBILE COSTS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN MOKRÝ**

**VEDOUcí PRÁCE**

SUPERVISOR

**prof. Ing. ADAM HEROUT, Ph.D.**

BRNO 2017

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2016/2017

**Zadání bakalářské práce**

Řešitel: **Mokrý Martin**

Obor: Informační technologie

Téma: **Mobilní aplikace pro zaznamenávání spotřeby automobilu**  
**Mobile App for Logging of Automobile Costs**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Prostudujte a popište existující mobilní aplikace pro sběr uživatelských dat a statistik.
2. Navrhněte funkčnost a principy užívání mobilní aplikace pro zaznamenávání spotřeby automobilu.
3. Prototypujte jednotlivé prvky uživatelského rozhraní řešené aplikace a prvky její funkčnosti. Testujte je na uživateli.
4. Implementujte řešenou aplikaci a testujte ji na uživateli.
5. Inkrementálně vylepšujte aplikaci na základě testů a podle možností implementujte pokročilou funkcionalitu (sdílení dat mezi uživateli, automatické připomenutí, atp.).
6. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

**Literatura:**

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2, značné rozpracování bodů 3 a 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Herout Adam, prof. Ing., Ph.D.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



---

doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Cielom tejto práce je navrhnuť a implementovať mobilnú aplikáciu pre zaznamenávanie spotreby automobilu. Aplikácia je určená pre operačný systém Android. Má umožniť užívateľom jednoducho a efektívne vkladať nové záznamy o čerpaniach paliva. Na základe týchto záznamov im následne zobrazí celkový prehľad o nákladoch ich vozidla. V tejto práci je zachytený postup vytvárania takejto aplikácie, jednotlivé etapy návrhu, implementačné detaily a testovanie výslednej aplikácie. Tiež sú tu uvedené a rozobraté podobné existujúce riešenia.

## Abstract

The goal of this thesis is to design and implement a mobile application for the recording of a car's consumption. The application is made for the Android operating system. It lets the user easily and effectively input new refuel data. Based on this data the application can show the user an overall view of their car's costs. This thesis focuses on the creation of this application, individual stages of design, implementation details and the testing of the application. It also states and analyzes similar existing applications.

## Kľúčové slová

mobilná aplikácia, android aplikácia, zber užívateľských dát, Android, štatistické údaje, štatistika, prehľad o nákladoch, aplikácia, Fuelinfo

## Keywords

mobile app, android app, collecting user data, Android, statistics data, statistics, costs overview, application, Fuelinfo

## Citácia

MOKRÝ, Martin. *Mobilná aplikácia pre zaznamenávanie spotreby automobilu*. Brno, 2017. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

# Mobilná aplikácia pre zaznamenávanie spotreby automobilu

## Prehlásenie

Prehlasujem, že som túto prácu vypracoval samostatne pod vedením pána profesora Adama Herouta. Uviedol som všetky literárne pramene, z ktorých som čerpal.

.....

Martin Mokrý  
17. mája 2017

## Podakovanie

Rád by som poďakoval pánovi Prof. Ing. Adamovi Heroutovi, Ph.D., ktorý mi poskytoval cenné rady a dodal mi motiváciu k riešeniu tejto práce. Zároveň by som rád poďakoval mojim rodičom, ktorí ma počas celého obdobia riešenia tejto práce veľmi podporovali.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Mobilné aplikácie zbierajúce užívateľské dáta a štatistiky</b>	<b>3</b>
2.1	Štyri typy mobilných aplikácií zbierajúcich užívateľské údaje . . . . .	3
2.2	Aplikácia Fuelinfo . . . . .	5
<b>3</b>	<b>Tvorba mobilných aplikácií pre Android</b>	<b>8</b>
3.1	Android . . . . .	8
3.2	Aplikácie . . . . .	11
3.3	Vzhľad Android aplikácie . . . . .	12
3.4	Nástroje na vývoj . . . . .	13
<b>4</b>	<b>Návrh vlastného riešenia a GUI</b>	<b>14</b>
4.1	Návrh prototypu – wireframe . . . . .	14
4.2	Návrh aplikácie . . . . .	16
<b>5</b>	<b>Implementačná časť riešenia</b>	<b>22</b>
5.1	Ovládacie prvky . . . . .	22
5.2	Práca s dátami . . . . .	25
5.3	Práca s grafmi . . . . .	26
5.4	Aktualizovanie zobrazovaných údajov . . . . .	26
5.5	Nástroje pre uľahčenie vývoja . . . . .	27
<b>6</b>	<b>Testovanie aplikácie a zhodnotenie výsledkov</b>	<b>28</b>
6.1	Testovanie na užívateľoch . . . . .	28
6.2	Možnosti pokračovania v projekte . . . . .	29
6.3	Zhodnotenie dosiahnutých výsledkov . . . . .	30
<b>7</b>	<b>Záver</b>	<b>31</b>
	<b>Literatúra</b>	<b>32</b>
<b>A</b>	<b>Obsah CD</b>	<b>34</b>

# Kapitola 1

## Úvod

Táto práca sa zaoberá návrhom a následnou tvorbou mobilnej aplikácie na zaznamenávanie spotreby automobilu – Fuelinfo. Taktiež sú v práci rozoberané existujúce riešenia mobilných aplikácií na zbieranie užívateľských dát, štatistík a konkurenčných riešení aplikácií na zaznamenávanie spotreby automobilov.

Dôvodom vzniku tejto práce bol fakt, že existujúce riešenia takýchto aplikácií sú často až príliš prepracované, a preto sa stávajú neprehľadné. Na druhej strane existujú aj aplikácie, ktoré sú jednoduché, no svojou funkčnosťou nespĺňajú základné požiadavky užívateľa na takýto typ aplikácie.

Cieľom tejto práce bolo vytvoriť také riešenie, ktoré bude zdravým kompromisom medzi týmito dvoma stranami. Svojou jednoduchosťou splní užívateľské požiadavky a zlepší tak užívateľom prehľad o ich finančných nákladoch na prevádzku za svoje vozidlo, najmä v oblasti nákladov za pohonné hmoty.

Táto práca sa ďalej zaoberá testovaním takto vytvorenej aplikácie na testovacích užívateľoch a následným vyhodnotením týchto testov. Konečné vytvorené riešenie je výsledkom štúdie existujúcich aplikácií s podobným zameraním, záverom vyvodeným z testov uskutočnených na testovacích užívateľoch a kreativity autora.

## Kapitola 2

# Mobilné aplikácie zbierajúce užívateľské dáta a štatistiky

Mobilné telefóny sú prenosné zariadenia, ktoré majú ľuďom umožniť primárne komunikovať, no v dnešnej dobe ich používame na oveľa viac činností. Veľa krát si ani neuvedomujeme, že dnešné smartphony, konkrétne aplikácie, ktoré pomocou nich používame, o nás zbierajú dáta a tie následne vyhodnocujú vo forme štatistických údajov.

### 2.1 Štyri typy mobilných aplikácií zbierajúcich užívateľské údaje

V nasledujúcich štyroch podkapitolách uvediem aplikácie, ktoré zbierajú alebo inak využívajú nazbierané užívateľské dáta. Každá sa líši spôsobom získavania takýchto dát, ako aj ich následným využitím. V každej podkapitole by mohlo byť uvedených viac aplikácií fungujúcich na podobnom princípe zbierania, vyhodnocovania a použitia dát, avšak ich základné črty by boli podobné. Z toho dôvodu som zvolil na opísanie každého typu takejto aplikácie práve jeden konkrétny príklad.

#### 2.1.1 Facebook

Asi najznámejšia a najpoužívanejšia sociálna sieť na svete [17], má aj vlastnú mobilnú rovnomennú aplikáciu. Samotná mobilná aplikácia zbiera veľké množstvo údajov o užívateľovi [8]<sup>1</sup>. Nazbierané údaje sa následne vyhodnocujú a sú využívané na ciele reklamu [9].

Tento typ zbierania a spracovania užívateľských dát zostáva užívateľom do istej miery nespozorovaný, prípadne o ňom užívateľ nemusí vedieť vôbec. Samotný užívateľ nemôže nijako nazbierané dáta využiť vo svoj prospech, prípadne si prezrieť vlastné štatistiky. Tieto slúžia iba spoločnosti Facebook a jej zákazníkom.

---

<sup>1</sup>Konkrétne povolenia je možné nájsť v zdroji, v sekcii povolenia

### 2.1.2 Zdravie - iOS

Toto je aplikácia [4], ktorá je priamo vstavaná v operačnom systéme *iOS*<sup>2</sup> od verzie osem. Aplikácia je zameraná na informovanie užívateľa o jeho zdravotnom stave, prípadne sa mu snaží dodávať motiváciu k fyzickej aktivite. Aplikácia teda kombinuje dva prístupy získavania dát. Prvý spôsob je, že užívateľ vloží informácie o svojom zdravotnom stave do aplikácie, teda svoju výšku, váhu, vek, prípadne zdravotný stav a podobne. Druhý spôsob zabezpečuje už samotná aplikácia, a to je zbieranie dát zo senzorov mobilného telefónu, prípadne inteligentných hodínok, ako je počet krokov, počet schodov, dĺžka behu, prejdená vzdialenosť. Na základe oboch druhov získaných informácií následne aplikácia užívateľovi zobrazí štatistiky o jeho zdravotnom stave.

Tento spôsob využitia užívateľských údajov je prospešný pre samotného užívateľa, a ten môže štatistické údaje využiť vo svoj prospech.

### 2.1.3 Účty, Platby

Tieto dve aplikácie [15] slúžia k sprístupneniu internetového bankovníctva prostredníctvom mobilného telefónu. Pomocou aplikácie Platby je možné vykonávať online platby. Aplikácia si následne uchováva históriu takýchto platieb a na jej základe vytvára kontakty, čo umožňuje užívateľovi opakovať platby na rovnaké účty komfortnejšie.

Aplikácia *Účty* na druhú stranu iba zbiera údaje o platbách, ich výšku, smer, čísla účtov a podobne. Na ich základe vytvára štatistiku, kde môže užívateľ veľmi prehľadne vidieť aké sumy finančných prostriedkov sa na jeho účte pohybujú, a mať tak prehľad o svojich financiách. Táto aplikácia vytvára štatistické údaje o všetkých pohyboch na účte, ako sú platby kartou, výbery z bankomatov, bezhotovostné prevody a taktiež aj platby mobilnou aplikáciou *Platby*.

Tento typ zbierania užívateľských údajov v sebe teda spája ako údaje, ktoré užívateľ zadal do samotného telefónu, tak aj údaje, ktoré sú dostupné na vzdialenom serveri.

### 2.1.4 AndroMoney

Aplikácia [3] určená na zlepšenie prehľadu o použitých finančných prostriedkoch. Užívateľ si sám, podľa svojho uváženia zadáva do aplikácie dáta, na základe ktorých mu aplikácia vytvorí požadované štatistické údaje a zlepší mu tak prehľad o tom, za aké typy vecí mína peniaze. Pri zadávaní nového výdavku si užívateľ zvolí dátum, jeho kategóriu, popis a presnú sumu. Môže si tiež nastaviť mesačný rozpočet, ktorý neplánuje prekročiť.

Tento typ aplikácie zaznamená užívateľove dáta, avšak iba tie, ktoré do nej užívateľ sám vloží. Všetky dáta sú uschované v lokálnej databáze a na ich základe sa vytvárajú štatistické údaje, ktoré slúžia iba pre informovanosť samotného užívateľa.

Na rozdiel od predchádzajúceho typu aplikácií dokáže táto aplikácia vytvárať detailnejšie záznamy a štatistiky o finančných výdavkoch a príjmoch, ale to iba do takej miery, do ktorej ju užívateľ využíva. Taktiež užívateľ nepotrebuje mať zriadený účet v banke ani v žiadnej inej inštitúcii, ktorá by viedla záznamy o jeho financiách.

---

<sup>2</sup><https://www.apple.com/iphone-7/ios/>



## 2.2 Aplikácia Fuelinfo

Aplikácia by mala slúžiť ako jednoduchý zápisník, do ktorého budú užívatelia vkladať záznamy o čerpaniach pohonných hmôt a z nich následne aplikácia vyhodnotí požadované štatistické údaje. Dala by sa teda zaradiť do typu popísaného v kapitole 2.1 v sekcii 2.1.4. Keďže sa jedná o čerpanie paliva, najčastejšie sa budú zadávať položky: množstvo načerpaného paliva, cena paliva za jeden liter a stav počítadla automobilu. Týmto položkám je treba prispôbiť formulár, do ktorého sa budú zadávať, a to takým spôsobom, aby boli položky dostatočne veľké, výrazné a ľahko sa na obrazovke hľadali. Samozrejme je potrebné rátať aj s položkami ako samotný automobil a taktiež s možnosťou jednotlivé záznamy a automobily upravovať.

Zadanie troch položiek pri novom zázname je pomerne jednoduchý úkon. Je teda potrebné, aby bolo zadávanie čo najjednoduchšie, a aby aplikácia sama užívateľovi niektoré položky už vopred vyplnila. Väčšina užívateľov navštevuje rovnaké čerpacie stanice. Ceny paliva sa menia raz za určité obdobie, a to iba o centy, stav tachometra sa taktiež zmení iba o pár stoviek kilometrov medzi čerpaniami. Z toho dôvodu je možné použiť údaje z minulých záznamov. Ak bude zadanie nového záznamu dostatočne rýchle a pohodlné, nebude užívateľov odrádzať a aplikáciu použijú vždy po načerpaní paliva.

Aplikácia všetky záznamy vyhodnotí a vykreslí na ich základe príslušné grafy, vypíše štatistické údaje a zdokonalí tak predstavu užívateľa o jeho nákladoch na prevádzku automobilu.

### 2.2.1 Operačný systém, pre ktorý je aplikácia určená

V predchádzajúcich kapitolách a podkapitolách som sa venoval všeobecným poznatkom o mobilných aplikáciách, ktoré zbierajú údaje o užívateľoch. Taktiež som sa venoval aj samotným princípom aplikácie zaznamenávajúcej spotrebu automobilu, tomu ako by mala fungovať, čo je jej cieľom a ako by toho mala dosiahnuť.

Keďže existuje viacero mobilných operačných systémov, je potrebné spomenúť, že táto aplikácia je vyvíjaná pre operačný systém Android, viac v kapitole 3.1. Tomu zodpovedá jej vzhľad, funkčnosť, implementácia a všetky vlastnosti, ktoré sa s týmto operačným systémom spájajú a budú popísané v nasledujúcich kapitolách.

### 2.2.2 Porovnanie existujúcich riešení

Pred tým ako sa začnem venovať samotnému opisu operačného systému Android, návrhu aplikácie a implementačným detailom, spomeniem dve zaujímavé riešenia mobilných aplikácií na zaznamenávanie spotreby a jedno nevydarené riešenie.

Takýchto aplikácií existuje pomerne veľké množstvo. Vzhľadom k tomu, že aplikácia Fuelinfo je určená pre systém Android, bude táto kapitola venovaná aplikáciám pre tento operačný systém. Pokúsím sa čo možno najobjektívnejšie zhodnotiť kladné aj záporné stránky jednotlivých aplikácií, a to najmä z pohľadu GUI<sup>3</sup>, zadávania nových záznamov, prehľadnosti zobrazovaných štatistík a celkovej prehľadnosti týchto aplikácií.

---

<sup>3</sup>Graphical User Interface – Grafické používateľské rozhranie

- **Fuelio**

Táto aplikácia [16] je z hľadiska designu veľmi pekne spracovaná a dodržiava základné pravidlá pre štandard Material Design, viac v kapitole 3.3.1. Návrh GUI je taktiež veľmi kvalitný a prehľadný, niektoré prvky by bolo možné ale stále vylepšiť. Ako príklad uvediem zadávanie nového záznamu čerpania paliva, pri ktorom je potrebné ručne vyplniť položky počítadlo, cena a množstvo paliva. Položky počítadlo a cena by bolo možné vopred vyplniť a užívateľ by vykonal už iba dodatočné úpravy. V prípade ceny za palivo by položku nemusel meniť vôbec, pokiaľ sa cena nezmenila. Toto by uľahčilo zadávanie nového záznamu.

Taktiež v prípade zobrazenia záznamov pre užívateľom zvolené auto z domovského okna, sa už nie je možné vrátiť späť šípkou v toolbare<sup>4</sup>, ale je potrebné tak urobiť cez bočné výsuvné menu.

Veľkou výhodou tejto aplikácie je zobrazenie veľkého množstva štatistických údajov v záložke Štatistiky.

- **Drivvo**

Aplikácia Drivvo [6] je taktiež veľmi dobre graficky spracovaná a dodržiava štandard Material Design. Na rozdiel od aplikácie Fuelio už vopred vypĺňa aj určité položky pri zadávaní nového záznamu a tým uľahčuje prácu užívateľovi. Avšak pri zadávaní nového záznamu je potrebné vyplniť aj celkovú cenu načerpaného paliva, napriek tomu, že užívateľ zadá množstvo a cenu za jednotku objemu. Aplikácia by toto mala vyplniť automaticky, čo aj urobí, ale až po zobrazení chybového hlásenia.

Návrh a spracovanie GUI je kvalitné, ale pre lepšiu orientáciu je potrebné aplikáciu používať dlhší čas, aby si prácu s ňou užívateľ dostatočne osvojil. Je to spôsobené až príliš veľkým množstvom funkcií, ktoré aplikácia ponúka. Tento fakt môže byť pri aplikácii na monitorovanie spotreby niekedy na škodu a môže užívateľa od používania aplikácie odradiť.

Táto aplikácia taktiež ponúka možnosť exportovať dáta vo forme CSV<sup>5</sup>/Excel, ale iba v platenej PRO verzii.

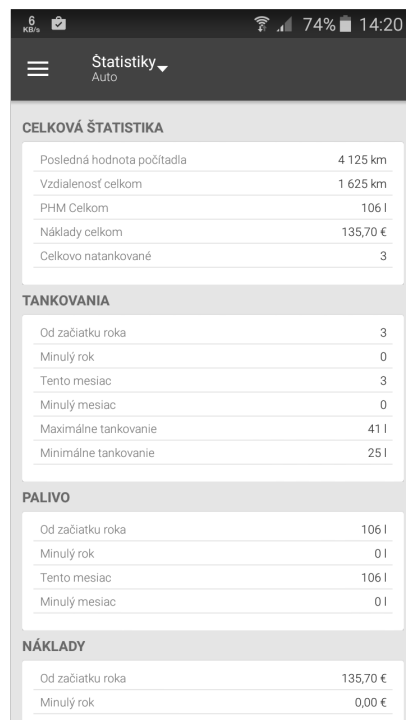
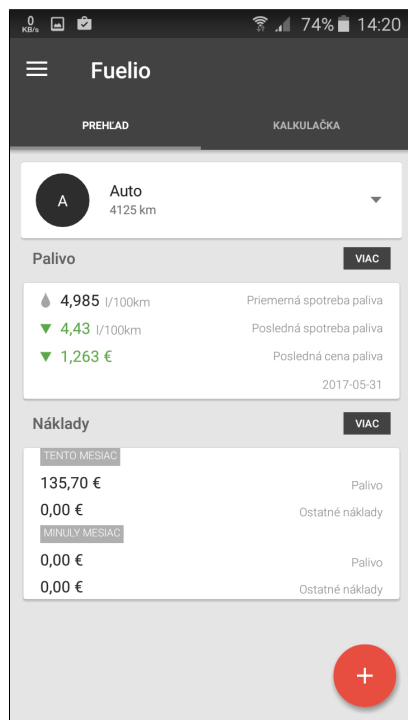
- **Fuel Consumption**

Toto je príklad aplikácie, ktorá je spomínaná v úvode tejto práce ako aplikácia, ktorá je jednoduchá, no nespĺňa užívateľské požiadavky. Rovnako slúži na vkladanie záznamov o čerpaní paliva ako vyššie uvedené aplikácie, avšak jej GUI je pomerne nešťastne spracované, neprehľadné a veľmi nepohodlné na prácu s ním. Jediné štatistické údaje, ktoré aplikácia ponúka sú graf priemernej spotreby a priemerná spotreba vyjadrená číselnou hodnotou. Graf zobrazuje hodnoty iba po hodnotu 10l/100km, ak je hodnota väčšia, tak sa zobrazí mimo grafu.

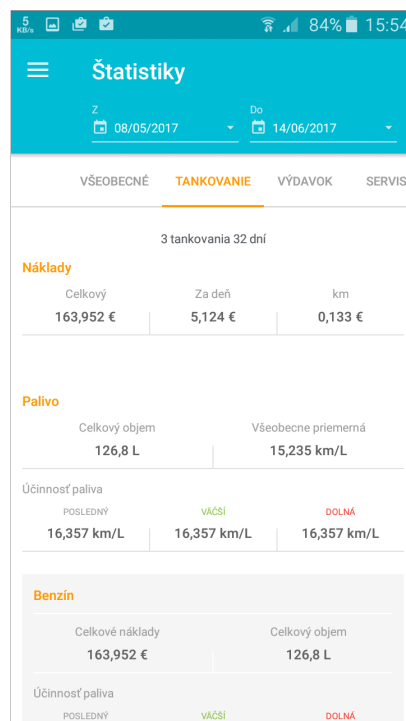
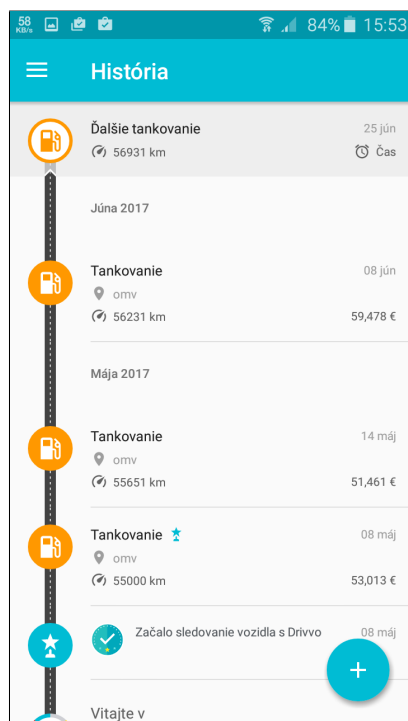
---

<sup>4</sup><https://developer.android.com/reference/android/support/v7/widget/Toolbar.html>

<sup>5</sup>Comma-separated values - Hodnoty oddelené čiarkou



Obr. 2.1: V ľavej časti obrázka je možné vidieť domovskú stránku aplikácie Fuelio. V pravej časti zase zobrazenie veľmi detailných štatistických údajov



Obr. 2.2: V ľavej časti obrázka je možné vidieť základnú obrazovku aplikácie Drivvo. V pravej časti zase zobrazenie štatistík o tankovaní.

## Kapitola 3

# Tvorba mobilných aplikácií pre Android

Aplikácie pre Android sú vyvíjané podobne ako pre iné mobilné operačné systémy so zámerom, aby boli pre užívateľa pohodlné na používanie a užitočné. Mali by teda poskytovať rovnakú UX<sup>1</sup> a funkčnosť. Vývoj sa líši v použitých technológiách a designe. Aplikácie pre Android sú programované v jazyku *Java*<sup>2</sup> a na uľahčenie ich výroby môže byť použitý Android SDK<sup>3</sup>, ktorý poskytuje API<sup>4</sup>.

### 3.1 Android

V predchádzajúcich kapitolách bol spomenutý operačný systém Android. V tejto kapitole bude v skratke vysvetlené ako tento systém funguje, aké su jeho časti a ako sú tieto časti spojené do funkčného celku.

Android ako operačný systém sa začal vyvíjať v roku 2005 spoločnosťou Android inc., ktorú neskôr odkúpila spoločnosť Google a v roku 2008 vyšla prvá verzia tohto operačného systému. Je založený na linuxovom jadre a je možné ho rozdeliť na 6 vrstiev [7]:

1. **Linux Kernel** – ako je možné vidieť na obrázku 3.1, kernel je základom systému Android. Poskytuje mu ovládače k hardwaru, prístup k súborovému systému a správu procesov. Je mierne odlišný od klasického Linux kernelu, a to tzv. Androidmami (*osviežovanie aplikácií na pozadí, anonmyná zdieľaná pamäť, ochrana proti nedostatku pamäte...*), ktoré ho lepšie prispôsobujú pre mobilné zariadenie.
2. **HAL – Hardware Abstraction Layer** – už názov napovedá, že sa jedná o vrstvu, ktorá obsahuje rôzne knižnice na prácu so zariadeniami telefónu. Vytvára tak akési rozhranie na jednoduchší prístup k týmto zariadeniam. Ako príklad zariadenia môže byť kamera alebo Bluetooth.

---

<sup>1</sup>User experience, User experience design

<sup>2</sup><https://www.oracle.com/java/index.html>

<sup>3</sup>Software Development Kit, (*Android Studio*)

<sup>4</sup>Application programming interface

3. **Základné C/C++ knižnice** – mnohé systémové komponenty sú postavené na knižniciach jazykov C/C++, a preto ich potrebujú pre svoju funkčnosť. Prípadne je možné pri vytváraní aplikácie používať rozhranie *OpenGL ES*<sup>5</sup> cez *Java OpenGL API* [2] a mnohé iné, ktoré vyžadujú tieto knižnice.
4. **Android Runtime (ART)** [12] – od verzie Android 5.0 nahradil starší systém nazývaný DVM<sup>6</sup>. Ten spúšťal DEX<sup>7</sup> súbory a pracoval na princípe JIT<sup>8</sup>, teda kompilácia prebiehala až pri spustení. Novší systém ART funguje na princípe AOT<sup>9</sup>, ktorý už počas inštalácie aplikácie skompiluje spustiteľné súbory a uloží ich do trvalej pamäte aplikácie. To zabezpečuje rýchlejšie spúšťanie aplikácií, keďže kód sa nemusí pred spúšťaním kompilovať.
5. **Java API Framework** – všetky funkcie, ktoré Android ponúka sú dostupné cez tieto knižnice. Využívajú sa pri vytváraní nových aplikácií. Vďaka nim môžu vývojári používať napríklad dizajnové prvky, upozornenia, prístup k dátam iných aplikácií a zásobník aktivít.
6. **Vrstva aplikácií** – sem patria už samotné aplikácie, či vstavané alebo nainštalované užívateľom. Tieto využívajú celú architektúru operačného systému, aby poskytli užívateľovi čo možno najlepšiu skúsenosť s Android zariadením.

---

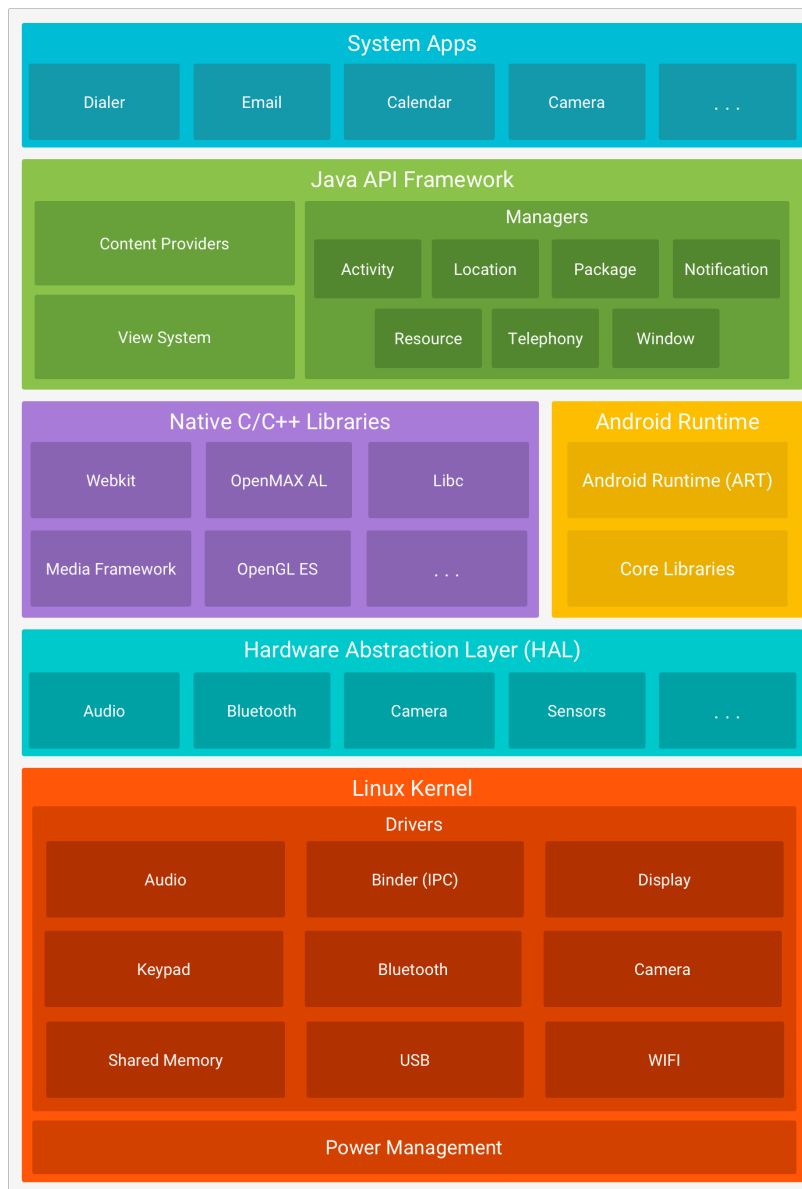
<sup>5</sup><https://developer.android.com/reference/android/opengl/package-summary.html>

<sup>6</sup>Dalvik Virtual Machine

<sup>7</sup>Dalvik Executable

<sup>8</sup>Just In Time

<sup>9</sup>Ahead Of Time



Obr. 3.1: Znáznorenie rozloženia jednotlivých vrstiev operačného systému Android [1]

## 3.2 Aplikácie

Na rozdiel od tradičných aplikácií sú Android aplikácie [7] voľnou kombináciou komponentov. Každý komponent môže mať viac prístupových miest a prístupy k nim môžu byť vykonané ako úkonmi užívateľa, tak aj systémom. Všetky komponenty a prístupy k nim sú definované v manifest súbore `AndroidManifest.xml`, ktorý sa pridá do aplikácie pri jej inštalácii. Najdôležitejším prvkom tohto súboru je meno balíčka aplikácie, ktoré je podpísané unikátnym kľúčom vývojára. To zabraňuje prepísaniu aplikácie inou rovnomennou aplikáciou. Toto sú štyri najdôležitejšie komponenty aplikácií:

1. **Aktivita** – jedna aktivita reprezentuje jednu „obrazovku“ v aplikácii, ktoré obsahujú užívateľské rozhranie. Sú to teda akési základné stavebné kamene, ktoré tvoria každú aplikáciu. Aktivity sú navrhované, aby boli zobrazované v určitom poradí, avšak je možné ich spustiť z akéhokoľvek miesta v aplikácii, prípadne mimo aplikácie. Spúšťajú sa pomocou správ, ktoré sú inštaniami triedy `Intent`. Tieto správy môžu obsahovať aj dodatočné informácie, ktoré je možné do nich vložiť.

Každá aktivita má svoj životný cyklus. To znamená, že prechádza rôznymi stavmi od svojho vzniku (*onCreate*) až po ukončenie/zmazanie (*onDestroy*). Ak užívateľ ukončí aktuálnu aktivitu, táto nemusí byť ihneď zmazaná systémom. Môže prejsť do stavu (*onPause*, *onStop*), v ktorom čaká na prípadné obnovenie (*onStart*, *onRestart*) alebo zmazanie.

2. **Služby** – bežia na pozadí a nemajú žiadne užívateľské rozhranie. Často sa používajú pre dlho-trvajúce procesy, ktoré majú bežať na pozadí a nerušiť užívateľa. Príkladom môže byť sťahovanie súborov, inštalácia nových aplikácií, prehrávanie hudby. Môže tiež definovať rozhranie pre komunikáciu s inými aplikáciami za využitia AIDL<sup>10</sup> a poskytovať im tak svoje služby.
3. **Content providers** – vytvárajú rozhranie pre aplikácie, aby mohli pristupovať k dátam, ktoré sú väčšinou uložené v súboroch alebo databáze. Často sa používajú na poskytovanie obsahu iným aplikáciám, vzhľadom na to, že kontrolujú prístupové práva ku konkrétnym častiam dát.
4. **Broadcast receivers** – odpovedajú na celo systémové udalosti, ktoré sa nazývajú *broadcast*. Ten môže pochádzať buď zo systému alebo z iných aplikácií a slúži na šírenie udalostí všetkým aplikáciám súčasne.

---

<sup>10</sup>Android Interface Definition Language

## 3.3 Vzhľad Android aplikácie

V kapitolách 3.1 a 3.2 som rozoberal technické detaily o systéme Android a tiež ako fungujú aplikácie na tomto systéme. V tejto kapitole sa zameriam na to, ako by mali aplikácie vyzerat a aké prvky k tomu využívajú. Taktiež uvediem akým spôsobom sa tento design programovo prenáša do samotnej aplikácie a aké nástroje môžu byť použité.

### 3.3.1 Material Design

Vývoj moderných mobilných aplikácií sa snaží o jednotný štýl ovládacích prvkov a o ich jednoduchosť. Ak užívateľ začne používať novú aplikáciu očakáva, že jej ovládacie prvky sa správajú rovnako, prípadne podobne ako tie, na ktoré je zvyknutý. Ak klikne na tlačidlo, tak očakáva určitú reakciu aplikácie. Taktiež sa zjednocujú aj kombinácie farieb aplikácií, rozloženie ovládacích prvkov a podoba formulárov.

Tento fakt sa snaží využívať väčšina vývojárov mobilných aplikácií, nezáleží pri tom pre aký operačný systém sú určené. Jedná sa teda o akýsi štandard a v prípade Android aplikácií sa nazýva Material Design [13].

Material Design prvý krát predstavil Google v roku 2014 a od vtedy si tento štandard získal veľa priaznivcov ako v radoch užívateľov, tak aj vývojárov mobilných aplikácií. Názov je akýmsi obrazným pomenovaním a má symbolizovať, že prvky v aplikácii by sa mali správať tak, ako by boli vyrobené z reálneho materiálu. Teda by mali mať určité vlastnosti ako výšku, povrch a okraje a mali by vrhať tieň. Avšak môžu aj meniť svoj tvar, spájať sa s inými prvkami alebo sa pohybovať.

### 3.3.2 Vzhľadové prvky

Každá aktivita obsahuje určité prvky, ktorých umiestnenie na obrazovke a ich typ, sú predpísané v *XML* súbore danej aktivity. Na toto sa používa značkovací jazyk *XML*<sup>11</sup>, ktorý nedefinuje nové prvky, iba o nich predáva informácie.

Existujú dva typy takýchto prvkov:

1. **Widgety**<sup>12</sup> – sú prvky, ktoré sa zobrazujú na obrazovke príslušnej aktivity. Takýchto prvkov je pomerne veľké množstvo, preto uvediem iba tri príklady: `Button`, `ImageButton`, `TextView`.
2. **Kontajnery**<sup>13</sup> – alebo tiež `layouty`, sú neviditeľné prvky, do ktorých sa dajú umiestňovať viditeľné prvky, teda widgety a taktiež aj ďalšie kontajnery. V Android aplikáciách existuje niekoľko druhov `layoutov` a každý má iné vlastnosti akými sa v ňom dá s widgetmi manipulovať.

`Layout`, ktorý môže, ale nemusí, obsahovať prvky sa pridá do aktivity pomocou metódy `setContentView(int id)`, ktorej predáme identifikačné číslo nami požadovaného `layoutu`. Metóda sa zvyčajne volá pri vytváraní aktivity v stave `onCreate`. Po pridaní prvkov, ktoré sa nachádzajú v `layoute`, pomocou metódy `findViewById(int id)`, môžeme následne pracovať aj s nimi.

---

<sup>11</sup>Extensible Markup Language

<sup>12</sup><https://developer.android.com/guide/topics/appwidgets/index.html>

<sup>13</sup><https://developer.android.com/guide/topics/ui/declaring-layout.html>



### 3.4 Nástroje na vývoj

Keďže programovanie aplikácie pre Android vyžaduje pomerne veľký projekt, je preto dobré zvoliť si určitý nástroj na vývoj – IDE<sup>14</sup>. Existuje viacero takýchto nástrojov. V čase písania tejto práce je asi najpoužívanejšie Android Studio<sup>15</sup>, ktoré je aj oficiálnym IDE od Google.

---

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Example"/>
</RelativeLayout>
```

---

Kód 3.1: Ukážka popisu widgetu *TextView* v *RelativeLayout* značkovacím jazykom XML

---

<sup>14</sup>Integrated Development Environment

<sup>15</sup><https://developer.android.com/studio/index.html>

## Kapitola 4

# Návrh vlastného riešenia a GUI

V tejto kapitole budem rozoberať ako som postupoval pri návrhu aplikácie Fuelinfo (kapitola 2.2), od návrhu prototypu a jeho testovania na užívateľoch až po výsledný vzhľad aplikácie. Vysvetlím, z akých aplikácií som bral inšpiráciu pri návrhu a tiež z akých chýb v iných aplikáciách som sa snažil poučiť.

### 4.1 Návrh prototypu – wireframe

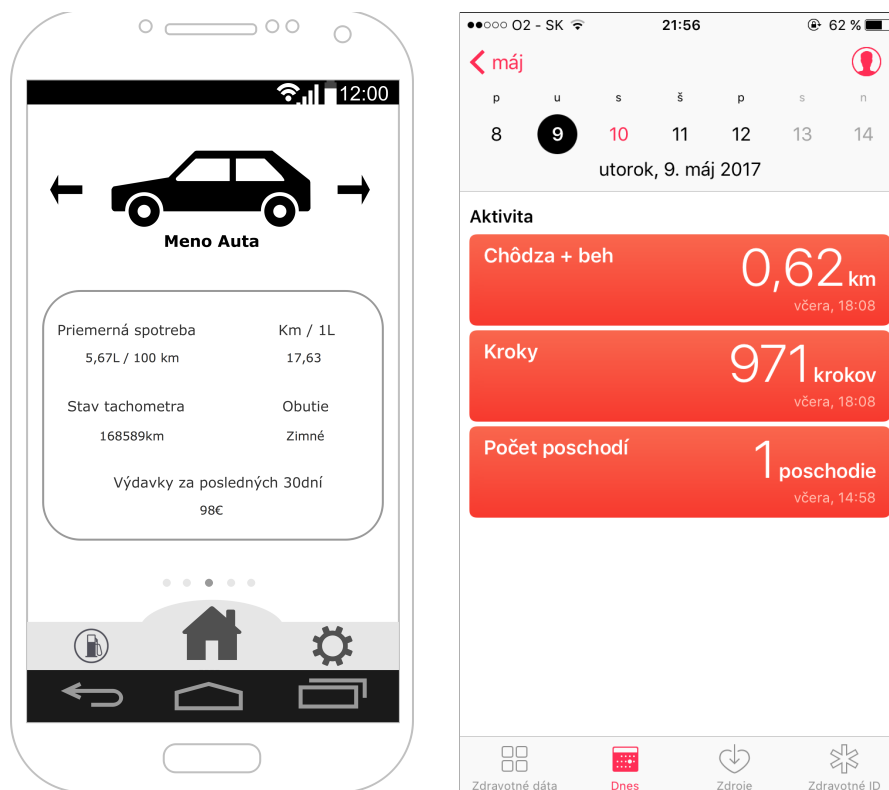
Na začiatku práce som si musel ujasniť, čo od aplikácie očakávajú užívatelia a aké funkcie by mala podporovať. Aplikácia mala zobrazovať dostatočne prehľadne a výrazne štatistické informácie o spotrebe automobilu a dovoliť jednoducho vkladať nové záznamy.

#### 1. Etapa návrhu

Keďže aplikácia mala podporovať aj pridávanie viacerých automobilov, bolo potrebné jasne rozlíšiť, s ktorým autom užívateľ práve pracuje. V tomto kroku som sa inšpiroval aplikáciou Zdravie (kapitola 2.1.2), v ktorej je veľmi dobre spracovaný prehľad jedného dňa a činností vykonaných v ňom. V hornej časti obrazovky je zobrazený kalendár s dňami v týždni a aktuálny deň je vždy zvýraznený.

Z toho dôvodu som sa rozhodol do hornej časti obrazovky umiestniť názov aktuálne vybraného automobilu a pod neho jeho štatistické údaje. Aktuálne vybraný automobil je možné šípkami po stranách meniť a zobrazíť tak postupne údaje o všetkých automobiloch.

Po predložení tohto návrhu (Obr. 4.1) testovacím užívateľom a vedúcemu mojej práce pánovi profesorovi Adamovi Heroutovi, som musel návrh prehodnotiť. Dôvodov bolo viac, v hornej časti obrazovky ikona auta zaberala príliš mnoho miesta a neprinášala žiadnu podstatnú informáciu, taktiež chýbala možnosť zobrazíť iné štatistické údaje.



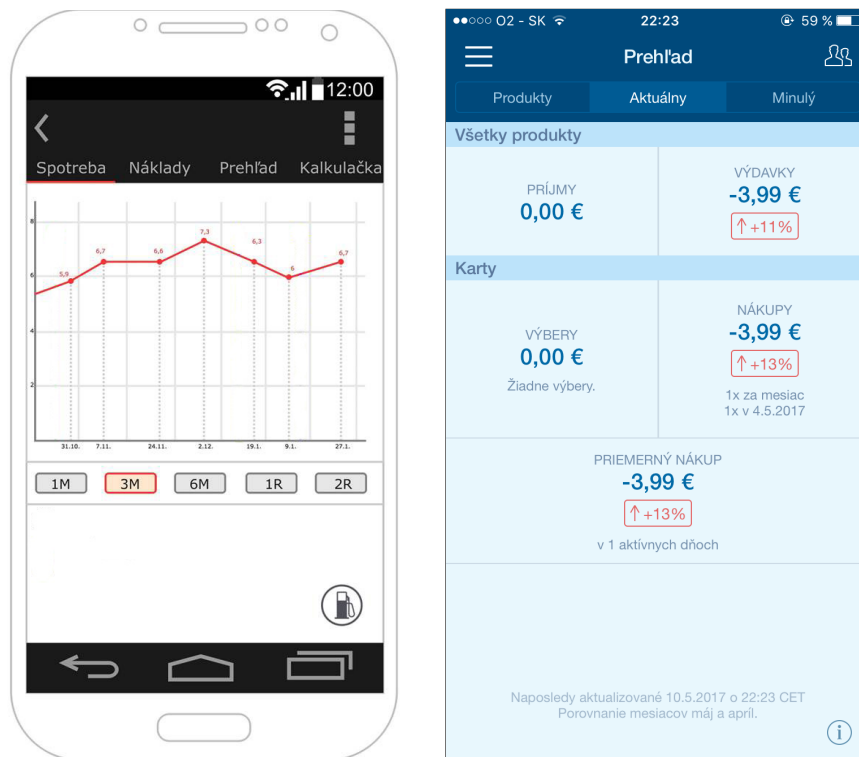
Obr. 4.1: V ľavej časti obrázka je možné vidieť návrh z 1. etapy vývoja aplikácie – v hornej časti možnosť zmeniť aktuálne vybrané auto, pod ním jednoduché štatistické informácie. V pravej časti je aplikácia Zdravie, taktiež v hornej časti je možnosť výberu dňa a pod ňou jednoduché štatistické informácie

## 2. Etapa návrhu

V tejto etape som sa inšpiroval aplikáciou Účty (kapitola 2.1.3) a to z dôvodu, že je navrhnutá ako singlepage aplikácia, takže všetky potrebné údaje zobrazuje na jednej obrazovke s možnosťou prechodu medzi záložkami. Týmto krokom som tiež znížil miernu neprehľadnosť tohto druhu aplikácií, o ktorej je písané v úvode tejto práce. Taktiež som sa rozhodol pridať graf, ktorý by zobrazoval priemernú spotrebu automobilu medzi čerpaniami paliva.

Tento návrh (Obr. 4.2) som ale taktiež zmenil. V spodnej časti grafu boli tlačidlá na zobrazenie časového rozsahu pomerne zbytočné, keďže knižnica na vytváranie grafov, ktorú som neskôr pri implementácii použil mala tento problém ošetrený.

Zo spätných väzieb testovacích užívateľov, ktorým som tento návrh ukázal, som zistil, že nemali prehľad, s ktorým automobilom práve pracujú. To samozrejme v prípade, že automobilov bolo viac. Z toho dôvodu som zväzil vrátenie prvku výberu automobilu, ale v menej rušivej podobe rozbalovacieho menu. K návrhu aplikácie ako singlepage boli spätné väzby pozitívne, preto som sa rozhodol túto zmenu aj uplatniť.



Obr. 4.2: V ľavej časti obrázka je možné vidieť návrh z 2. etapy vývoja aplikácie. V hornej časti pribudli záložky a na hlavnej obrazovke je zobrazený graf. Na pravej časti obrázka je aplikácia Účty ktorou som sa inšpiroval pri vytváraní tohto návrhu.

Spätné väzby boli získavané od skupiny testovacích užívateľov, ktorých bolo päť. Postupoval som na základe znalostí nadobudnutých v predmete Tvorba užívateľských rozhraní (ITU [5]) a knihy *Handbook of Usability Testing* [14]. Užívateľov som sa pýtal otázkou ohľadom jednotlivých prvkov na obrazovke a snažil sa zistiť, či vedia, čo má byť ich významom a ako sa chovajú po použití. Ak sa odpovede jednotlivých užívateľov, nezávisle od seba, zhodovali vyhodnotil som výsledok ako pozitívny. V opačnom prípade ako negatívny. Wireframe modely boli vytvorené pomocou webového nástroja draw.io<sup>1</sup>.

## 4.2 Návrh aplikácie

V tejto kapitole rozoberiem ako som využil návrhy prototypov a poznatky, ktoré som z nich získal, na vytvorenie vzhľadu výslednej aplikácie. Taktiež som sa inšpiroval prvkami z už existujúcich riešení z kapitoly 2.2.2. Konkrétne z aplikácie Fuelio som prevzal vzhľad číselných štatistických údajov (Obr.2.1), pretože bol veľmi prehľadný a graficky príjemne spracovaný.

V aplikácii Drivvo (Obr. 2.2) som sa inšpiroval funkčným prvkom – samostatné vyplnenie položiek pri zadávaní nového záznamu o čerpaní paliva.

<sup>1</sup><https://www.draw.io/>

### 4.2.1 Základná obrazovka

Ako som písal v predchádzajúcich kapitolách, aplikácia Fuelinfo je singlepage aplikácia. Základná obrazovka aplikácie je teda dôležitá, pretože obsahuje všetky podstatné prvky. Obsahuje tiež v hornej časti obrazovky aj menu so záložkami a každá záložka zobrazuje iné štatistické informácie. Všetky majú tieto základné prvky:

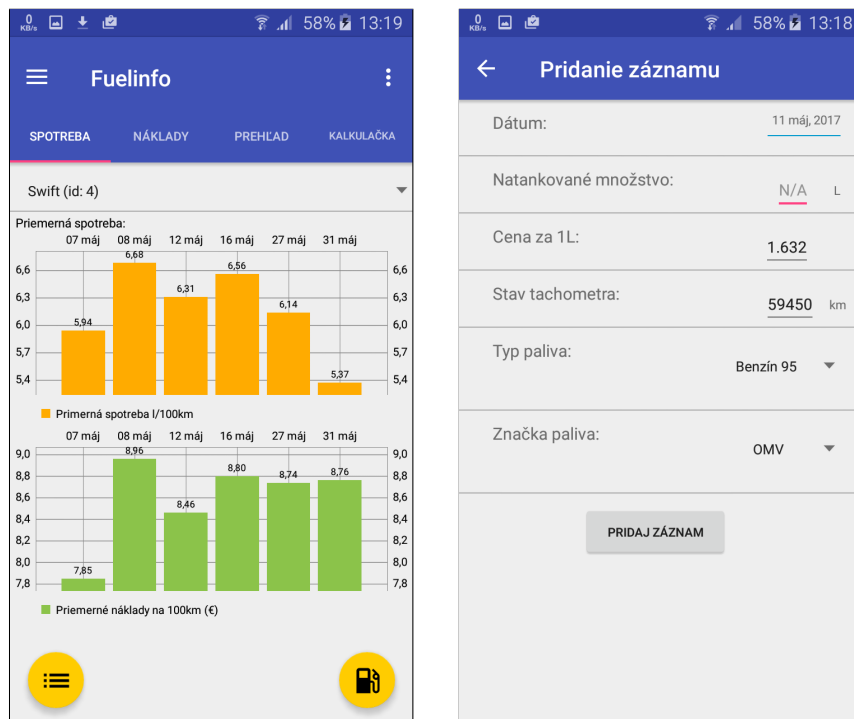
1. **Tlačidlo pre prídanie nového záznamu** – nachádza sa v pravej spodnej časti obrazovky, aby bolo ľahko dostupné. Toto je z dôvodu, že väčšina ľudí je pravákov, a teda sa dokážu k pravému okraju mobilného telefónu dostať palcom pohodlnejšie, ak ho držia v pravej ruke. Toto tlačidlo je teda prioritné a očakáva sa, že bude využívané pri načerpaní paliva. Ikona benzínového stojanu, ktorá je na tlačidle má evokovať, že sa jedná o prídanie nového záznamu.
2. **Tlačidlo pre zobrazenie záznamov** – nachádza sa v ľavej spodnej časti obrazovky a obsahuje ikonu záznamov.
3. **Rozbaľovacie menu** – slúži na výber automobilu, s ktorým užívateľ aktuálne pracuje. Tento automobil je po celý čas zobrazený v hornej lište a mal by evokovať, že akákoľvek aktivita vykonaná na hlavnej obrazovke sa vzťahuje k práve vybranému automobilu. Z tohto dôvodu je aj tlačidlo pre zobrazenie záznamov po celý čas zobrazené na základnej obrazovke. Je zreteľné, že záznamy, ktoré sa po kliknutí na toto tlačidlo zobrazia, patria k zobrazenému automobilu v rozbaľovacom menu. Rovnako aj prídanie nového záznamu. Toto menu tiež obsahuje možnosť prídania nového automobilu. Tá je vždy zobrazená ako posledná.

Ak užívateľ nastaví v menu iné auto, všetky štatistické údaje zobrazené v záložkách sa aktualizujú a zobrazia hodnoty pre novo zvolený automobil.

Záložka základnej obrazovky obsahuje ďalej dva grafy. Oproti návrhu, ktorý je na obrázku 4.2, boli ale grafy prerobené z lineárnych grafov na grafy stĺpcové. Hodnoty priemernej spotreby a nákladov sa totiž môžu meniť o niekoľko desiatok percent, to spôsobuje veľké výkyvy. Na lineárnom grafe to môže pôsobiť rušivým dojmom.

Horný graf vyjadruje priemernú spotrebu automobilu od posledného čerpania paliva v jednotkách  $[l/100km]$ . Spodný zase priemerné náklady na vzdialenosť  $100km$ , podľa cien paliva.

Tento návrh je možné vidieť na obrázku 4.3 v ľavej časti.



Obr. 4.3: V ľavej časti obrázka je zobrazená domovská obrazovka aplikácie Fuelinfo. V pravej časti je zobrazená obrazovka pre pridanie nového záznamu o čerpaní paliva.

#### 4.2.2 Pridávanie nových záznamov

Táto časť už z podstaty aplikácie musí byť jednoduchá, dobre funkčná a prehľadná. Do tejto obrazovky bolo potrebné vložiť niekoľko prvkov:

- dátum vloženia záznamu
- množstvo načerpaného paliva
- cena paliva
- stav počítadla automobilu

Tieto položky boli potrebné pre ďalšie spracovanie. Rozhodol som sa pridať aj dve dodatočné položky a to: typ paliva a jeho značku.

Keďže nové záznamy budú užívatelia vkladať pravidelne, je vhodné niektoré položky vyplniť vopred. Prvá takáto položka je dátum. Automaticky sa nastaví na aktuálny dátum v čase, kedy je okno otvorené. Ak by ho chcel užívateľ zmeniť, môže tak urobiť kliknutím naň. Následne sa zobrazí kalendár, kde vyberie požadovaný dátum.

Položky cena paliva a stav počítadla sa vyplnia podľa posledného vloženého záznamu. Stav počítadla užívateľ jednoducho upraví zmenou posledných hodnôt čísel, taktiež vidí aký posledný stav zadal. To mu v prípade, ak má v aplikácii viac automobilov uľahčí rozoznať, o ktorý sa jedná. Do tohto poľa nie je možné zadať menšie číslo ako bol posledný stav počítadla.

Položky typ paliva a značka paliva sa taktiež vyplnia podľa posledného vloženého záznamu. Je možné ich ľahko zmeniť. Na výber má užívateľ niekoľko značiek a typov palív. V prípade, že mu ani jedna možnosť nevyhovuje, môže zvoliť možnosť Iné. Formulár na vkladanie nových položiek je možné vidieť na obrázku 4.3 v pravej časti.

### 4.2.3 Upravovanie záznamov

Všetky záznamy je možné aj upravovať. Užívateľ si najprv zobrazí zoznam všetkých záznamov k danému automobilu. Vyberie jeden, ktorý chce upraviť kliknutím naň. Následne sa mu zobrazí nové okno na upravenie záznamu. Formulár v tomto okne vyzerá rovnako ako formulár pre pridanie nového záznamu, no s tým rozdielom, že všetky položky sú vyplnené a užívateľ ich má možnosť upraviť.

Po skončení úprav sa zmeny uložia kliknutím na tlačidlo *Uložiť Zmeny*. Tiež je možné celý záznam vymazať tlačidlom *Vymazať Záznam*.

### 4.2.4 Pridávanie a upravovanie automobilov

Nový automobil môže užívateľ vložiť cez rozbalovacie menu. Poslednou položkou tohto menu je možnosť pridanie automobilu. Po vybratí tejto možnosti sa zobrazí nové okno, v ktorom užívateľ vyplní všetky údaje o svojom automobile. Prvé dve položky (meno auta a stav počítadla) sú povinné. Ostatné sú nepovinné a slúžia iba pre informáciu užívateľa.

Automobil je tiež možné aj upraviť. Formulár na upravenie je rovnaký ako formulár na pridávanie automobilov. Samozrejme obsahuje aj možnosť zmazať automobil.

V prípade, že sa užívateľ rozhodne daný automobil zmazať, zmažú sa s ním aj všetky záznamy, ktoré k tomuto automobilu patria. Tieto záznamy už nie je možné obnoviť. Pred zmazaním sa preto zobrazí dialógové okno, ktoré musí užívateľ potvrdiť a súhlasí tak s tým, že automobil a záznamy budú zmazané.

Tento formulár je možné vidieť na obrázku 4.6 v pravej časti.

### 4.2.5 Záložky náklady a prehľad

- Náklady – v tejto záložke je zobrazený prehľad nákladov za načerpané palivo. V hornej časti je graf vývoja cien paliva, aby si užívateľ lepšie vedel vysvetliť zmeny vo svojich nákladoch. Pod týmto grafom je jednoduchý zoznam štatistických údajov, ktoré sa týkajú cien paliva a nákladov na čerpanie paliva za určité obdobie.
- Prehľad – táto záložka je rozdelená do troch skupín. Je tu zobrazená celková štatistika za celé obdobie vedenia záznamov k vybranému automobilu. Ďalej je tu priemerná štatistika k tomuto automobilu. A ako posledná položka je najčastejšie čerpané palivo – značka a typ.

Vzhľad je zobrazený na obrázku 4.5.

#### 4.2.6 Záložka kalkulačka

Slúži na predbežný odhad budúcich nákladov za palivo. Užívateľ musí vyplniť počet kilometrov, za ktoré chce vypočítať odhadované náklady. Cena paliva a spotreba automobilu sa vyplnia z údajov, ktoré boli vložené v minulosti, a to nasledovne:

cena paliva je použitá z posledného záznamu, spotreba automobilu je dopočítaná z posledných troch záznamov, konkrétne z ich aritmetického priemeru. V prípade, že užívateľ chce použiť iné hodnoty, môže tak urobiť jednoduchým prepísaním týchto položiek.

Táto záložka je zobrazená na obrázku 4.6 v ľavej časti. Bola pridaná až pri testovaní funkčnej aplikácie na základe užívateľských testov ako pokročilá funkcionality, ale keďže patrí k návrhu aplikácie, je popísaná v tejto kapitole.

Ak vybrané auto neobsahuje žiadne záznamy, všetky položky ktoré by sa mali vopred vyplniť zostávajú prázdne. Taktiež grafy zobrazia informáciu, že neexistuje dostatočné množstvo dát na zobrazenie údajov.

#### 4.2.7 Logo aplikácie

K aplikácii som samozrejme navrhol aj vhodné logo. To by malo evokovať, o akú aplikáciu sa jedná. Logo bolo vytvorené v programe *Adobe Photoshop CS6*<sup>2</sup>. Ako základ bol použitý obrázok zo stránky *depositphotos*<sup>3</sup>.

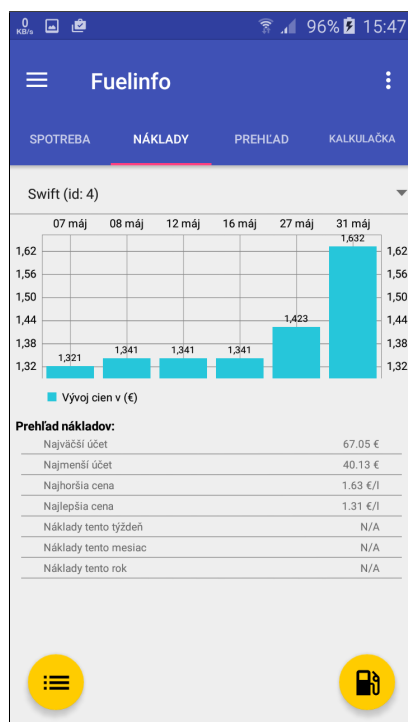


Obr. 4.4: Ikona aplikácie Fuelinfo

<sup>2</sup><http://www.adobe.com/sk/products/photoshop.html>

<sup>3</sup><https://depositphotos.com/100326094/stock-illustration-vector-outline-filling-gun.html>





**Swift (id: 4)**

**Celková štatistika:**

Stav počítadla	59450 km
Celkovo prejdene vzdialenosť	4450 km
Celkovo natankované	244.0 l
Celkové náklady	336.16 €
Počet tankovaní	7

**Priemerná štatistika:**

Priemerná spotreba	5.55 l/100km
Priemerné náklady na deň	84.04
Priemerná platba	48.02 €
Priemerné tankovanie	34.86 l
Priemerné náklady na 1km	0.0755 €/1km

**Palivo:**

Najčastejšie tankované palivo	Benzin 95
Najčastejšie tankovaná značka paliva	OMV

Obr. 4.5: Vľavo je možné vidieť obrazovku záložky náklady. Vpravo záložku prehľad.

**Swift (id: 4)**

**Vložte údaje:**

Vzdialenosť v km	—
Cena za liter	1.632 €
Spotreba	6.168 l/100km

**Výsledok:**

Cena paliva	N/A
Množstvo potrebného paliva	N/A

**Pridanie nového auta**

Meno auta:	<u>Povinné</u>
Počiatkový stav počítadla:	<u>Povinné</u>
Popis:	<u>Nepovinné</u>
Značka	<u>Nepovinné</u>
Model:	<u>Nepovinné</u>

**PRIDAJ AUTO**

Obr. 4.6: Vľavo je záložka Kalkulačka. Vpravo zase formulár na pridanie nového automobilu.

## Kapitola 5

# Implementačná časť riešenia

Aplikácia bola písaná v jazyku *Java* a pri písaní bolo použité vývojové prostredie *Android Studio*. Keďže na začiatku vývoja aplikácie som nemal žiadne skúsenosti s vývojom aplikácií ako všeobecne, tak aj pre systém Android, nemusia niektoré časti kódu spĺňať všetky zvyklosti tohto prostredia. Informácie som čerpal z rôznych online zdrojov ako aj z knihy: Vývoj aplikácií pro Android [11].

Počas vývoja boli použité aj externé knižnice na uľahčenie práce s niektorými prvkami. Všetky budú uvedené v tejto časti dokumentácie a tiež bude uvedený aj spôsob, akým som ich využíval. Grafická časť aplikácie, teda vzhľad aktivít, je písaná v jazyku *XML*.

### 5.1 Ovládacie prvky

Predchádzajúca kapitola bola venovaná návrhu aplikácie po designovej stránke, preto by som rád v tejto kapitole pokračoval a rozoberiem jednotlivé prvky z hľadiska kódu a implementácie. Popíšem tiež ako sa jednotlivé prvky používajú a v akých častiach programu je potrebné ich deklarovať. Venovať sa budem iba tým najdôležitejším, prípadne niektoré ukážem ako príklad.

#### 5.1.1 Floating Action Button

Je to jeden z prvkov štandardu Material Design a používa sa na dobre viditeľných miestach. Väčšinou býva umiestnený v spodných okrajoch obrazovky a je zobrazený ako keby sa vznášal nad jej obsahom. Do vnútra tohto tlačidla sa vkladá obrázok (*drawable*), väčšinou vektorový pre lepšie prispôbenie sa iným rozlíšeniam.

V aplikácii Fuelinfo sú použité takéto tlačidlá dve a ich umiestnenie už bolo rozoberané v kapitole 4.2.1.

---

```
<android.support.design.widget.FloatingActionButton
    android:id="@+id/tankovanie"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_gravity="end|bottom"
    android:scaleType="center"
    app:srcCompat="@drawable/ic_local_gas_station"
    app:backgroundTint="#ffcc00"/>
```

---

Kód 5.1: Pridanie Floating Action Button do layoutu

V ukážke 5.1 je kód, ktorým sa takéto tlačidlo v súbore .xml popisuje. Konkrétne sa jedná o tlačidlo pre zadanie nového záznamu.

---

```
FloatingActionButton tankovanie_button = (FloatingActionButton)
    findViewById(R.id.tankovanie);
    tankovanie_button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent tankuj_activy = new Intent(MainActivity.this,
                TankujActivity.class);
            startActivityForResult(tankuj_activy,1);
        }
    });
```

---

Kód 5.2: Práca s tlačidlom v jazyku Java

V ukážke 5.2 je naznačené ako sa následne pracuje s týmto tlačidlom v jazyku Java. Najprv je deklarovaný objekt triedy `FloatingActionButton`, do ktorého sa podľa popisu z kapitoly 3.3.2 pridá odkaz na tlačidlo vytvorené v .xml súbore.

V kóde je tak isto naznačené ako sa tlačidlu pridá `ClickListener`, ktorý zistí kliknutie užívateľom na takéto tlačidlo a vykoná určitú akciu. V tomto prípade vytvorí novú inštanciu pre komponent aktivity a následne túto aktivitu spustí zaslaním správy.

Táto ukážka vytvorenia a následného použitia tlačidla by mala slúžiť ako ukážka, keďže veľmi podobne sa pracuje aj s ostatnými viditeľnými prvkami, ktoré je možné vložiť do aktivít.

### 5.1.2 ToolBar Menu

Ovládacím prvkom je tiež aj *ToolBar Menu* v hornej časti obrazovky. Je možné do neho pridať prvky, tlačidlá v podobe `item` a následne s nimi pracovať. Toto zníži množstvo tlačidiel priamo na obrazovke a dokáže aktivitu viac sprehľadniť.

Prvým krokom je vytvorenie opäť .xml súboru, v ktorom sa popíše, aké má mať menu vlastnosti a aké prvky má obsahovať.

---

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context="com.example.martin.merac_sпотреby1.MainActivity">
    <item
        android:id="@+id/action_car_list"
        android:title="@string/car_list" />
    <item
        android:id="@+id/valute"
        android:title="@string/valute" />
</menu>
```

---

Kód 5.3: Vytvorenie menu pre ToolBar

Do .xml súboru aktivity, do ktorej sa takéto menu pridáva, je potrebné popísať položku (widget) Toolbar a vytvoriť jej id.

Pri práci s Toolbarom v aktivite sa postupuje rovnako ako v prípade tlačidla z kapitoly 5.1.1. Rozdiel je v následnej práci s položkami tohto menu, kedy je potrebné prekryť metódu `onCreateOptionsMenu` a v nej pridať meno vytvoreného .xml súboru.

Na zistenie, ktorá položka bola užívateľom zvolená, je potrebné vytvoriť podobnú metódu ako pri tlačidle, avšak nedefinuje sa priamo na inštanciu Toolbar. Miesto toho pokrýva metódu `onOptionsItemSelected`.

---

```
@Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_car_list:
            Intent car_list = new Intent(this, CarListActivity.class);
            startActivity(car_list);
            return true;
        case R.id.valute:
            popupWindow();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
}
```

---

Kód 5.4: Ukážka práce s prvkom menu v toolbare

### 5.1.3 EditText

Aby bolo možné zaznamenať užívateľský vstup z klávesnice existuje widget *EditText*<sup>1</sup>. Používa sa podobne ako tlačidlo z kapitoly 5.1.1. Užívateľ teda zadá do takto pripraveného poľa určitú hodnotu a po vykonaní akcie (napríklad kliknutí na tlačidlo) je možné s touto hodnotou cez objekt triedy `EditText` pracovať. Je možné nastaviť, aké typy hodnôt budeme v danom poli očakávať, či sa bude jednať o text, čísla, desatinné čísla a podobne.

Text je možné z inštancie vziať cez jej metódu `getText` a skonvertovať ju na nami požadovaný typ.

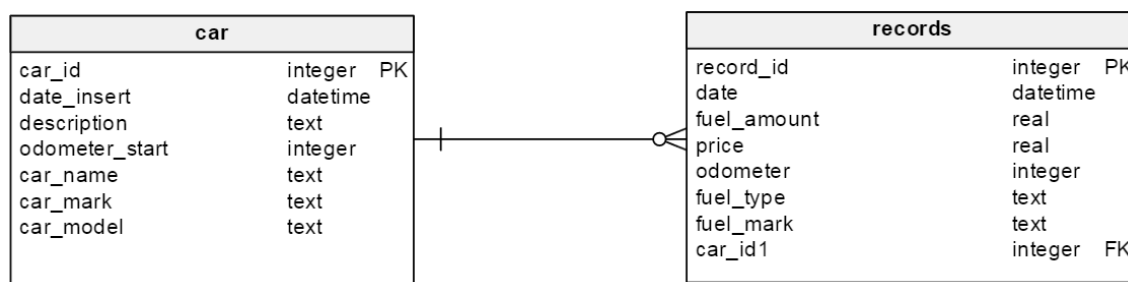
---

<sup>1</sup><https://developer.android.com/reference/android/widget/EditText.html>

## 5.2 Práca s dátami

Ovládacie prvky z predchádzajúcej sekcie umožňujú užívateľovi pracovať s aplikáciou. Pri aplikácii, ktorá musí zbierať užívateľské dáta a vytvárať z nich štatistiky je potrebné, aby dáta zostali zachované aj po vypnutí aplikácie, prípadne mobilného telefónu, to znamená, aby dáta boli perzistentné.

Tento fakt som riešil vytvorením internej offline databázy pomocou *SQLite*<sup>2</sup>. Tento typ databázy je priamo podporovaný operačným systémom Android a nie je problém s ním pracovať. Pre potreby mojej aplikácie bolo potrebné vytvoriť v databáze dve tabuľky, jednu pre automobily a druhú pre záznamy k týmto automobily. Príčinom druhá tabuľka - pre záznamy, obsahovala cudzí kľúč pre tabuľku automobilov.



Obr. 5.1: ER diagram databázy, ktorá bola použitá pre aplikáciu Fuelinfo a obsahuje dve entity.

Pre prácu s takouto databázou v prostredí Android slúži trieda *SQLiteOpenHelper*. Pomocou nej je možné vykonávať nad databázou operácie, a to metódami `execSQL` a `rawQuery`. Tieto metódy majú ako argument reťazec, ktorý obsahuje príkaz v jazyku *SQLite*.

Pre vyberanie hodnôt z databázy slúži metóda `rawQuery`, ktorej návratová hodnota je objekt typu `Cursor`. Argumentom tejto metódy je nejaký databázový dotaz vo forme: `SELECT meno_stĺpca FROM meno_tabuľky;`. Ten sa jej predá ako reťazec.

V objekte `cursor` sa následne dá pohybovať pomocou metód `moveTo` a z konkrétnej pozície vyberať dáta pomocou metód `getDouble`, `getInt` a podobne.

---

```
Cursor c = ...
while(c.moveToNext())
{
    ...
}
```

---

Kód 5.5: Postupná iterácia cez `Cursor`

Vkladanie dát sa zabezpečovalo pomocou metódy `execSQL`, ktorej sa ako argument predal reťazec obsahujúci *SQLite* príkaz vo forme: `INSERT INTO meno_tabuľky(meno_stĺpca) VALUES (hodnoty);`

<sup>2</sup><https://www.sqlite.org/>

## 5.3 Práca s grafmi

Na zobrazenie grafov bola použitá knižnica MPAndroidChart [10], čo je externá knižnica a musela byť do aplikácie pridaná. Ponúka niekoľko rôznych druhov grafov, ja som si zvolil grafy stĺpcové a približne popíšem ako som s nimi pracoval. Na začiatku sa vytvorí `List<BarEntry>`, do ktorého sa postupne pridávajú hodnoty vo formáte (*os x, os y*). Po naplnení tohto listu sa vytvorí objekt triedy `BarDataSet` a predá sa mu list so záznamami aj s popisom, vo forme refazca, o aké hodnoty sa jedná. Objektu `BarDataSet` je možné nastaviť požadovanú farbu, prípadne iné vlastnosti. Po nastavení všetkých požadovaných vlastností sa vytvorí nový objekt typu `BarData` a jeho konštruktoru sa predá objekt `BarDataSet`.

Na začiatku je samozrejme potrebné popísať graf v aktivite a priradiť ho svojmu objektu. Rovnako ako tlačidlo v kapitole 5.1.1 Tomuto grafu sa následne predá objekt `BarData` a po zavolaní metódy `invalidate()` sa hodnoty aktualizujú a zobrazia na grafe.

Priamo do objektu `BarEntry` nie je možné vkladať popisy na os *x*. V prípade, ak na túto os chceme dostať popis a nie číselné hodnoty, ako to bolo v mojom prípade, je potrebné použiť metódu:

---

```
BarChart graf = ...;  
graf.getXAxis().setValueFormatter(new IndexAxisValueFormatter(pole));
```

---

Kód 5.6: Zmena popisov na osi x

Ktorá hodnoty na ose *x* [0,1,2...] zamení za hodnoty z poľa [0,1,2...].

## 5.4 Aktualizovanie zobrazovaných údajov

V aplikácii Fuelinfo sú všetky hodnoty zobrazené na jednej obrazovke, konkrétne vo fragmentoch<sup>3</sup>, a nad nimi sa nachádza rozbaľovacie menu vo forme spinneru<sup>4</sup>, na ktorom môže užívateľ ľubovoľne meniť automobily. Je preto potrebné zaistiť, aby sa všetky zobrazované hodnoty spolu so zmenou automobilu aktualizovali.

Na príklade grafov popíšem, ako som požadovaného efektu dosiahol. Rovnako som postupoval aj pri ostatných prvkoch hlavnej obrazovky a všetkých fragmentov.

Pre každý fragment som vytvoril novú triedu, v ktorej som definoval metódy na prácu s databázou ako je popísané v kapitole 5.2. V týchto triedach som vybrané hodnoty spracoval do požadovaného tvaru. V prípade grafov sa jednalo o tvar:

```
[[spotreba, datum, cena], [spotreba, datum, cena], ...].
```

Následne som ich priamo vložil do objektov, ktoré reprezentovali prvky na obrazovke.

V aktivite, v ktorej sa nachádzali fragmenty, u mňa to bola `MainActivity`, som vytvoril objekty týchto tried a zavolať v nich metódu, ktorá zabezpečila aktualizovanie zobrazovaných údajov znovu načítaním dát z databázy. Túto aktualizáciu som vykonával vždy pri zmene výberu automobilu, po vložení nového záznamu o čerpaní paliva, pri upravení záznamov a pri zmene výberu záložky.

<sup>3</sup><https://developer.android.com/guide/components/fragments.html>

<sup>4</sup><https://developer.android.com/guide/topics/ui/controls/spinner.html>

## 5.5 Nástroje pre uľahčenie vývoja

Počas vytvárania aplikácie som používal aj rôzne nástroje pre uľahčenie vývoja. Najdôležitejším nástrojom pre mňa bolo prostredie Android Studio, ktoré obsahuje veľa už vstavaných a užitočných nástrojov. Okrem iného umožňuje ľuďom, ktorí nevlastnia zariadenie s Android operačným systémom, používať virtuálny emulátor<sup>5</sup>. Je možné stiahnuť si niekoľko druhov emulátorov, ktoré sa líšia verziou operačného systému Android aj rozlíšením obrazovky zariadenia.

Stretol som sa s názormi, že emulátor nie je dobré riešenie na testovanie a ladenie aplikácie. Údajne aplikácie na ňom fungujú inak ako v skutočnom telefóne, padajú, zamrzajú. Takéto chyby som nespozoroval a emulátor som využíval počas celého vývoja aplikácie, v kombinácii s mobilným telefónom Samsung Galaxy J3.

Pre prácu s ikonami a obrázkami je veľmi dobrý nástroj *Android Drawable Importer*<sup>6</sup> od firmy *JetBrains*, využíval som ho pravidelne a obsahuje veľké množstvo rastrových a vektorových ikoniek.

---

<sup>5</sup><https://developer.android.com/studio/run/emulator.html>

<sup>6</sup><https://plugins.jetbrains.com/plugin/7658-android-drawable-importer>

## Kapitola 6

# Testovanie aplikácie a zhodnotenie výsledkov

Po vytvorení funkčnej aplikácie ju bolo potrebné otestovať aj na užívateľoch. K testovaniu som pristupoval dvomi metódami a obe tu uvediem. Z časových dôvodov som sa však nedostal až k testovaniu v Obchode Play, čo je tiež jedna z položiek, ktorými sa dá pokračovať pri vývoji tejto aplikácie.

### 6.1 Testovanie na užívateľoch

Najskôr som aplikáciu testoval na mojich známych, a to priamo na mojom telefóne. Toto testovanie však nebolo veľmi úspešné. Väčšinou som zistil, že aplikácia je dobrá a nemajú jej čo vytknúť. Tento typ spätnej väzby teda neprinášal žiadnu podstatnú informáciu.

Pristúpil som teda k inému spôsobu [14] testovania. V Android Studio som vytvoril inštalačný balíček .apk a poslal ho testovacím užívateľom. K aplikácii som nedodal žiadne inštrukcie ako ju používať, iba som im predal popis toho, čo je jej podstatou. Zistil som nasledujúce fakty:

- **Nevýrazný dátum**

Pri pridávaní nového záznamu užívateľa nevedeli, že môžu zmeniť aj dátum jeho vloženia. Tento bol totiž do formulára vložený iba ako widget TextView, a tak neevokoval možnosť, že je možné na neho kliknúť. Túto skutočnosť som opravil pridaním podčiarknutia, aby pole dátum vyzeralo podobne ako ostatné polia, ktoré je možné upravovať. Problém sa už neopakoval.

- **Zadanie príliš dlhého mena automobilu pri jeho vytváraní**

Toto bola implementačná chyba aplikácie, kde som neošetril dĺžku zadaného reťazca. V dôsledku zadania príliš dlhého reťazca sa do textu rozbaľovacieho menu tento reťazec vložil celý, čo výrazne narúšalo vzhľad aplikácie a jej použiteľnosť. Taktiež tento problém bol aj pri číselných hodnotách, avšak už nebol iba estetický, ale aplikácia kvôli nemu spadla. Oba problémy som vyriešil nastavením maximálnej dĺžky reťazca a už sa neopakovali.



- **Možnosť zmeniť zobrazovanú menu**

Po otestovaní aplikácie užívateľmi z Českej republiky bol zistený problém, že aplikácia pracuje iba s menou € a nie Kč. Problém som vyriešil pridaním možnosti zmeniť zobrazovanú menu aj na České koruny. Aplikácia je lokalizovaná iba do slovenského jazyka, preto som zatiaľ neuvažoval o ostatných menách, ale iba o Českej republike z dôvodu neexistujúcej jazykovej bariéry.

Toto je tiež jedna z možností, kde by bolo vhodné v aplikácii pokračovať a pridať lokalizácie do iných jazykov spolu s menami.

- **Pridanie štatistických údajov**

V záložke náklady bol pôvodne zobrazený iba graf vývoja cien paliva. Užívateľom ale chýbali štatistické informácie o ich nákladoch za určité obdobia, prípadne najhoršia a najlepšia cena paliva. Tieto údaje som do záložky doplnil a problém tak vyriešil.

## 6.2 Možnosti pokračovania v projekte

Možností pokračovania v projekte je pomerne dosť. Niektoré som už uviedol v predchádzajúcej podkapitole 6.1. Na tomto projekte som nestihol implementovať všetky plánované prvky, a tak nie sú v tejto správe ani zahrnuté. Uvediem ich teda v tejto časti.

- **Možnosť zdieľania dát medzi užívateľmi** – Táto funkcia by mohla byť pomerne užitočná najmä v prípadoch, že viac užívateľov používa rovnaký automobil. Pri pridaní takejto možnosti do aplikácie by bolo nutné urobiť to prehľadne a efektívne. Aplikácia Fuelinfo je jednoduchá, malá a do istej miery intuitívna. Pridanie takejto možnosti rozšírenia by znamenalo, že užívatelia by museli mať v aplikácii vytvorené svoje účty, museli by sa do aplikácie prihlasovať a pri prvom spustení registrovať. Tento fakt a tiež aj samotný systém zdieľaných automobilov by mohol aplikáciu urobiť menej prehľadnou a odradiť tak užívateľov od jej používania.
- **Automatické pripomenutie** V prípade, že by užívatelia zabúdali na pridávanie nových záznamov o čerpaní paliva, aplikácia by im to mohla pripomenúť. Táto funkcia by mohla byť implementovaná dvomi spôsobmi.

Aplikácia by podľa pravidelnosti pridávania záznamov vypočítala priemerný čas medzi pridávaniami. Ak by čas, za ktorý užívateľ nepridal záznam prekročil priemerný čas, aplikácia by mu zobrazila upozornenie. Táto implementácia má nevýhodu, ak užívateľ dlhší čas nejazdí autom, prípadne je odcestovaný alebo z iného dôvodu nezadáva záznamy, tak by ho aplikácia upozorňovala zbytočne a mohlo by ho to odradiť od jej používania.

Druhý spôsob by bol využitie GPS modulu telefónu. V prípade, ak by sa užívateľ priblížil k čerpacej stanici a zotrval by na nej nejaký čas, tak by aplikácia toto vyhodnotila ako nové čerpanie paliva a zobrazila by upozornenie, aby užívateľ nezabudol toto čerpanie zaznamenať. Nepresnosti GPS modulov by ale mohli spôsobovať, že takéto upozornenia by užívateľ dostával pravidelne aj v prípade, že sa na čerpacej stanici nenachádza, čo by ho tiež mohlo odradiť od používania aplikácie.

- **Možnosť konverzie meny** Ako som už spomenul v časti 6.1, tak by aplikácia mohla podporovať aj iné meny. V prípade, že užívateľ veľa cestuje automobilom a pravidelne čerpá palivo v zahraničí, kde sa používa iná mena, mohla by aplikácia ponúkať konverziu týchto mien. Užívateľ by mal tak vždy prehľad o svojich nákladoch vo svojej mene a nemusel by toto prepočítavať ručne. Táto možnosť by sa ešte dala rozšíriť, aby užívateľ aktuálny kurz nemusel zadávať aplikácii, ale aplikácia by si ho stiahla sama z internetu. Toto rozšírenie by bolo potrebné zapracovať efektívne, keďže užívateľov, ktorí by takúto možnosť využívali je menšina. Užívatelia, ktorí by túto možnosť nepotrebovali, by ju mohli skryť alebo iným spôsobom odstrániť.

### 6.3 Zhodnotenie dosiahnutých výsledkov

Počas tohto projektu som si vyskúšal prácu v oblasti užívateľských rozhraní a osvojil si základy práce s mobilným operačným systémom Android. Vývoj mobilných aplikácií nie je súčasťou bakalárskeho štúdia na Fakulte informačných technológií Vysokého Učení Technického v Brně, a preto považujem túto skúsenosť za veľký prínos pre svoje znalosti. Pri práci na tomto projekte som zužitkoval poznatky, ktoré som na tejto škole nadobudol a prakticky som ich aplikoval.

Výsledkom tohto projektu je aplikácia Fuelinfo, ktorá dokáže zaznamenávať užívateľom zadané dáta a na ich základe mu zobrazí prehľadnú štatistiku. Napriek tomu, že aplikácia nebola vydaná v obchode Google Play, tak je aktívne používaná dvoma testovacími užívateľmi.

# Kapitola 7

## Záver

V tejto práci boli naštudované a popísané existujúce mobilné aplikácie na zaznamenávanie užívateľských dát a štatistík. Tiež boli popísané aj existujúce mobilné aplikácie na zaznamenávanie spotreby automobilu. Inšpirované týmito aplikáciami boli navrhnuté funkčnosť a princípy aplikácie Fuelinfo. Jej jednotlivé prvky boli prototypované a následne otestované na testovacích užívateľoch. Počas týchto testov bola do návrhu pridaná aj pokročilá funkcionálna vo forme kalkulačky na výpočet predbežnej spotreby automobilu.

Na základe spätných väzieb užívateľov dospel autor tejto práce k záveru, že návrh aplikácie Fuelinfo je dostatočný, a tak ju mohol implementovať. Funkčná aplikácia bola následne testovaná na užívateľoch. Na základe týchto testov v nej boli opravené prípadné chyby alebo doplnené požadované funkčné prvky. Autor tohto projektu bude aj naďalej pokračovať vo vývoji aplikácie Fuelinfo.

# Literatúra

- [1] *Platform Architecture*. [Online; navštívené 10.05.2017].  
URL <https://developer.android.com/guide/platform/index.html>
- [2] *android.opengl*. [Online; navštívené 10.05.2017].  
URL <https://developer.android.com/reference/android/opengl/package-summary.html>
- [3] AndroMoney: *AndroMoney ( Expense Track )*. [Online; navštívené 08.05.2017].  
URL <https://play.google.com/store/apps/details?id=com.kpmoney.android&hl=sk>
- [4] Apple inc.: *Zdravie*. [Online; navštívené 08.05.2017].  
URL <https://www.apple.com/ios/health/>
- [5] Beran, V.: *Vyhodnocení uživatelského rozhraní*. Fakulta informačních technologií Vysokého učení technického v Brně, [Online; navštívené 11.05.2017].  
URL <https://www.fit.vutbr.cz/study/courses/ITU/private/labs/design/itu-vyhodnoceni-poznamky.pdf>
- [6] Drivvo: *Drivvo - Správa vozidiel*. [Online; navštívené 08.05.2017].  
URL <https://play.google.com/store/apps/details?id=br.com.ctncardoso.ctncar&hl=sk>
- [7] Elenkov, N.: *Android Security Internals - An In-Depth Guide to Android's Security Architecture*. No Starch Press, 2014, ISBN 978-1-59327-581-5, 1 - 12 s.
- [8] Facebook: *Facebook*. [Online; navštívené 08.05.2017].  
URL <https://play.google.com/store/apps/details?id=com.facebook.katana>
- [9] Facebook: *Facebook ads*. [Online; navštívené 08.05.2017].  
URL <https://www.facebook.com/business/products/ads>
- [10] Jahoda, P.: *MPAndroidChart*. [Online; navštívené 12.05.2017].  
URL <https://github.com/PhilJay/MPAndroidChart>
- [11] Luboslav Lacko: *Vývoj aplikací pro Android*. Albatros Media, 2015, ISBN 978-80-251-4347-6.
- [12] Mahmudul, H.; Ariful, H.: *Mobile Application Development Approaches: Recommendation for E-commerce Enterprises*. Bachelor's thesis, Laurea University of Applied Sciences, [https://www.theseus.fi/bitstream/handle/10024/120281/Thesis\\_Paper\\_Final.pdf?sequence=1](https://www.theseus.fi/bitstream/handle/10024/120281/Thesis_Paper_Final.pdf?sequence=1), Prosinec 2016, [Online; navštívené 10.05.2017].

- [13] Mew, K.: *Learning Material Design*. Packt Publishing, 2015, ISBN 9781785289811.
- [14] Rubin, J.; Chisnell, D.: *Handbook of Usability Testing, Second Edition: How to Plan, Design, and Conduct Effective Tests*. Wiley Publishing, Inc., 2008, ISBN 978-0-470-18548-3, 181 - 185 s.
- [15] Slovenska sporitelna: *Mobilné aplikácie SporoApps*. [Online; navštívené 08.05.2017].  
URL <https://www.slsp.sk/sk/ludia/non-product/aplikacie-pre-smartfony>
- [16] Sygic: *Fuelio - palivo a náklady*. [Online; navštívené 08.05.2017].  
URL <https://play.google.com/store/apps/details?id=com.kajda.fuelio&hl=sk>
- [17] The Statistics Portal: *Most famous social network sites worldwide as of April 2017, ranked by number of active users*. [Online; navštívené 08.05.2017].  
URL <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

# Príloha A

## Obsah CD

Priložené CD obsahuje túto adresárovú štruktúru:

- **Fuelinfo** - obsahuje súbor .apk a manuál k jeho spusteniu na mobilnom telefóne.
- **Fuelinfo - zdrojové súbory** - obsahuje zložku s projektom vytvoreným v Android Studiu a všetky zdrojové súbory
- **Plagát** - obsahuje plagát k aplikácií vo formáte PNG
- **Technická správa** - obsahuje túto technickú správu a jej tlačiteľnú kópiu
- **Technická správa - zdrojový latex kód** - obsahuje zdrojové latex súbory tejto technickej správy
- **Video** - obsahuje video k aplikácií