

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## MOBILNÍ APLIKACE PRO NAČÍTÁNÍ ÚČTENEK A SPRÁVU CEN PRODUKTŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JÁN PAULOVČÁK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# MOBILNÍ APLIKACE PRO NAČÍTÁNÍ ÚČTENEK A SPRÁVU CEN PRODUKTŮ

MOBILE APPLICATION FOR BILL SCANNING AND PRODUCT PRICE MANAGEMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JÁN PAULOVČÁK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2015

## **Abstrakt**

Tato bakalářská práce se zabývá návrhem a implementací mobilní aplikace pro operační systém Android. Obsahuje teoretický rozbor návrhu jednotlivých částí aplikace a popis využitých technologií. Při návrhu je kladen důraz na funkčnost aplikace spolu se zachováním jednoduchosti v ovládaní a přátelského uživatelského rozhraní.

## **Abstract**

This bachelor thesis deals with design and implementation of mobile application for Android operating system. It contains a theoretical analysis of the design of individual parts of the application and a description of used technologies. The design of application aims mainly to its functionality along with ease of control and friendly user interface.

## **Klíčová slova**

Android, OCR, aplikace, Otsu, spracování obrazu, Tesseract, REST, HTTP

## **Keywords**

Android, OCR, application, Otsu, image processing, Tesseract, REST, HTTP

## **Citace**

Ján Paulovčák: Mobilní aplikace pro načítání účtenek a správu cen produktů, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Mobilní aplikace pro načítání účtenek a správu cen produktů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szókeho, Ph.D.

.....  
Ján Paulovčák  
18. května 2015

## Poděkování

Ďekuji panu Ing. Igorovi Szókemu, Ph.D. za vedení bakalářské práce, odbornou pomoc, cenné připomínky a poskytnuté konzultace, které vedli k dosaženým výsledům.

© Ján Paulovčák, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>2</b>
1.1 Cieľ bakalárskej práce . . . . .	3
1.2 Členenie práce . . . . .	3
<b>2 Návrh aplikácie</b>	<b>4</b>
2.1 Diagram prípadov použitia . . . . .	5
2.2 Uživatelské rozhranie . . . . .	6
2.2.1 Úvodná obrazovka . . . . .	6
2.2.2 Hlavné menu . . . . .	6
2.2.3 Hlavná obrazovka . . . . .	7
2.2.4 Pridávanie nových produktov . . . . .	8
2.2.5 Detailný popis produktu . . . . .	10
2.2.6 Úprava existujúceho produktu . . . . .	13
2.2.7 Duplikácia existujúceho produktu . . . . .	14
2.2.8 Vyhľadávanie produktov . . . . .	14
<b>3 Implementácia</b>	<b>16</b>
3.1 Android Framework . . . . .	16
3.1.1 Komponenty . . . . .	17
3.1.2 Rôzne veľkosti obrazoviek . . . . .	19
3.2 OCR . . . . .	20
3.2.1 Pre-processing . . . . .	21
3.2.2 Rozoznávanie znakov . . . . .	23
3.2.3 Tesseract . . . . .	23
3.2.4 Pridávanie nových produktov pomocou OCR . . . . .	24
3.3 Databáza . . . . .	25
3.4 REST API . . . . .	25
3.5 Knižnice . . . . .	27
<b>4 Testovanie</b>	<b>28</b>
<b>5 Záver</b>	<b>31</b>
5.1 Dosiahnuté ciele . . . . .	31
5.2 Rozšírenia . . . . .	31
<b>A Obsah CD</b>	<b>34</b>
<b>B REST API</b>	<b>35</b>

# Kapitola 1

## Úvod

Za niekoľko posledných dekád prešli informačné technológie vo svete obrovským rozvojom. Tento prínos uľahčil ľuďom prácu takmer v každom smere, a jednou z oblastí sú nepochybne telekomunikácie a s nimi spojené mobilné telefóny. Tie však spočiatku splňali len jednu základnú funkciu, a to telefonovanie. Neskôr sa užívateľom umožnilo aj posielanie textových správ. Postupom času sa výkon a možnosti zvyšovali, pribúdali nové funkcie, a z luxusnej vymoženosti, určenej hlavne pre bohatých sa stal nástroj na komunikáciu dostupný pre takmer celú širokú verejnosť. Možnosť spojenia sa so svojimi blízkymi sa stala skutočnosťou, ktorá vo výraznej miere ovplyvnila životy ľudí. Nie je to však až tak dávno, čo sa na trhu po prvý krát objavili mobilné aplikácie. Tie boli nanešťastie dostupné výhradne pre vreckové počítače (angl. Personal Digital Assistant - PDA), a opäť ako pri mobilných telefónoch, ich kvôli ich vysokej cene využívali zväčša manažéri alebo počítačoví experti. No vďaka neustálemu rozvoju technológií a zvyšujúcemu sa výkonu zariadení, sa práve mobilné telefóny začali postupne týmto vreckovým počítačom vyrovnávať, ba dokonca, v súčasnej dobe ich takmer úplne vyradili z trhu a prináležil im názov inteligentné telefóny (angl. smartphones). Slovo "inteligentné" referuje najmä na obrovské spektrum služieb, ktoré nám tieto zariadenia ponúkajú, a to nielen v oblasti komunikácií ale aj hier, hudby, finančníctva, navigácií, fotografií, videa a mnoho ďalších.

Obrovský prevrat v používaní mobilných zariadení nastal v roku 2007, keď kalifornská firma Apple uviedla na trh operačný systém iOS a spolu s ním iPhone - jeden z prvých inteligentných telefónov, umožňujúci ovládanie viacerými dotykmi súčasne, tzv. multi-touch. Hlavnými výhodami bola veľká obrazovka reagujúca na priamy dotyk prsta, a úplna absencia stylusu a klávesnice, ktoré boli v tom čase typické pre tieto zariadenia. Konkurencia však na seba nenechala dlho čakať, a v nasledujúcom roku firma Google predstavila svoj vlastný operačný systém Android, spolu s inteligentným telefónom HTC Dream. Aj keď bol tento operačný systém výkonovo slabší než operačný systém firmy Apple, jeho popularita sa v roku 2010 ohromne rozšírila, a dnes je dominantným systémom na trhu. Android je taktiež narozdiel od iOS-u od svojho počiatku open-source a je dostupný pre širokú škálu zariadení, počínajúc mobilnými telefónmi, cez tablety, hodinky a nechýba ani medzi televíziami. Viac než milión nových zariadení s týmto operačným systémom je spustených každý deň, a približne každý mesiac je jeden a pol miliardy aplikácií stiahnutých z Obchodu Play. Aj vďaka týmto faktom, pôsobí Android ako najlepšia voľba pre vývoj aplikácie.

## 1.1 Cieľ bakalárskej práce

Hlavným cieľom tejto bakalárskej práce je vytvoriť aplikáciu pre mobilnú platformu Android umožňujúcu užívateľom uchovávať si jednotlivé ceny produktov, spravovať tieto produkty, poskytnúť im prehľad o cenách rovnakých produktov zakúpených inými užívateľmi, ako aj zobrazovať detailný popis jednotlivých produktov vrátane miesta nákupu, dátumu zakúpenia, názvu produktu a štatistiky o cenách. Pre uľahčenie pridávania produktov do databázy sa užívateľovi ponúka výber z dvoch možností a to pridanie pomocou vloženia konkrétnych údajov alebo vyfotenia pokladničného bloku a následného automatického rozčlenenia jednotlivých položiek. Úlohou je taktiež poskytnúť užívateľom možnosť priradenia čiarového kódu k produktu a následná možnosť rýchleho vyhľadávania práve pomocou naskenovania čiarového kódu.

Aplikácia je určená pre všetky vekové kategórie a je pre ľudí, ktorí chcú mať prehľad o cenách produktov, ktoré si zakúpajú. Prínosom je pre tých, ktorí si chcú porovnať aktuálne ceny s históriou ich nákupov a to pomocou grafu, ako aj porovnať si svoje nákupy s nákupmi iných ľudí a určiť, čo je pre nich najvýhodnejšie.

## 1.2 Členenie práce

Práca je rozčlenená do štyroch kapitol popisujúcich problematiku a návrhy riešenia pre daný problém. Kapitola 2 je zameraná najmä na návrh aplikácie, a to zobrazenie jednotlivých interakcií s aplikáciou a možností, ktoré sa poskytujú užívateľovi. Kapitola 3 je venovaná implementácii aplikácie, detailnému popisu problemových častí, použitých knižníc, ale aj operačného systému, pre ktorý je aplikácia vytvorená, a v ktorého prostredí bude pracovať. Popisuje tak základné prvky systému z pohľadu vývojára. Kapitola 4 je zameraná na oblasť testovania a zmien užívateľského rozhrania na základe spätných väzieb užívateľov.

## Kapitola 2

# Návrh aplikácie

Druhá časť práce je venovaná návrhu aplikácie, a to ako grafickému tak aj jednotlivým prípadom použitia. Cieľom bolo vytvoriť aplikáciu, ktorá bude spĺňať jednak špecifikované požiadavky, no súčasne bude jednoduchá na používanie. V nasledujúcej časti bližšie špecifikujem jednotlivé požiadavky aplikácie a doplním ich UML diagramom prípadov použitia pre lepšie objasnenie a pochopenie kontextu.

### 1. Vkladanie nových produktov

Aplikácia by mala umožňovať užívateľovi vkladanie nových produktov, a to dvoma nasledujúcimi spôsobmi:

- **Manuálne**

Možnosť vpísať jednotlivé vlastnosti produktu do predpripravených polí a jeho následné uloženie

- **Fotografickým spracovaním pokladničného bloku**

Umožniť užívateľovi vyfotografovať pokladničný blok, spracovať ho a automatizovane predpripraviť produkty na uloženie

### 2. Uchovávanie dát

Možnosť vložené produkty uchovávať aj po vypnutí aplikácie, vymazovať ich, ale aj prehliadať a upravovať.

### 3. Zdieľanie cien s inými užívateľmi

Aplikácia by mala byť schopná medzi užívateľmi zdieľať ceny nákupov a pri zobrazení detailného popisu produktu, v prípade dostupnosti tieto ceny, ktoré prisluchajú k danému produktu zobraziť.

### 4. Zobrazenie detailného popisu produktu

Vytvorenie detailného popisu produktu vrátane štatistických údajov cien, histórie nákupov, dátumu zakúpenia ako aj miesta nákupu. V prípade dostupnosti internetového pripojenia pomocou vzdialeného prístupu k databáze zobraziť jednotlivé ceny nákupov iných užívateľov pre daný produkt.

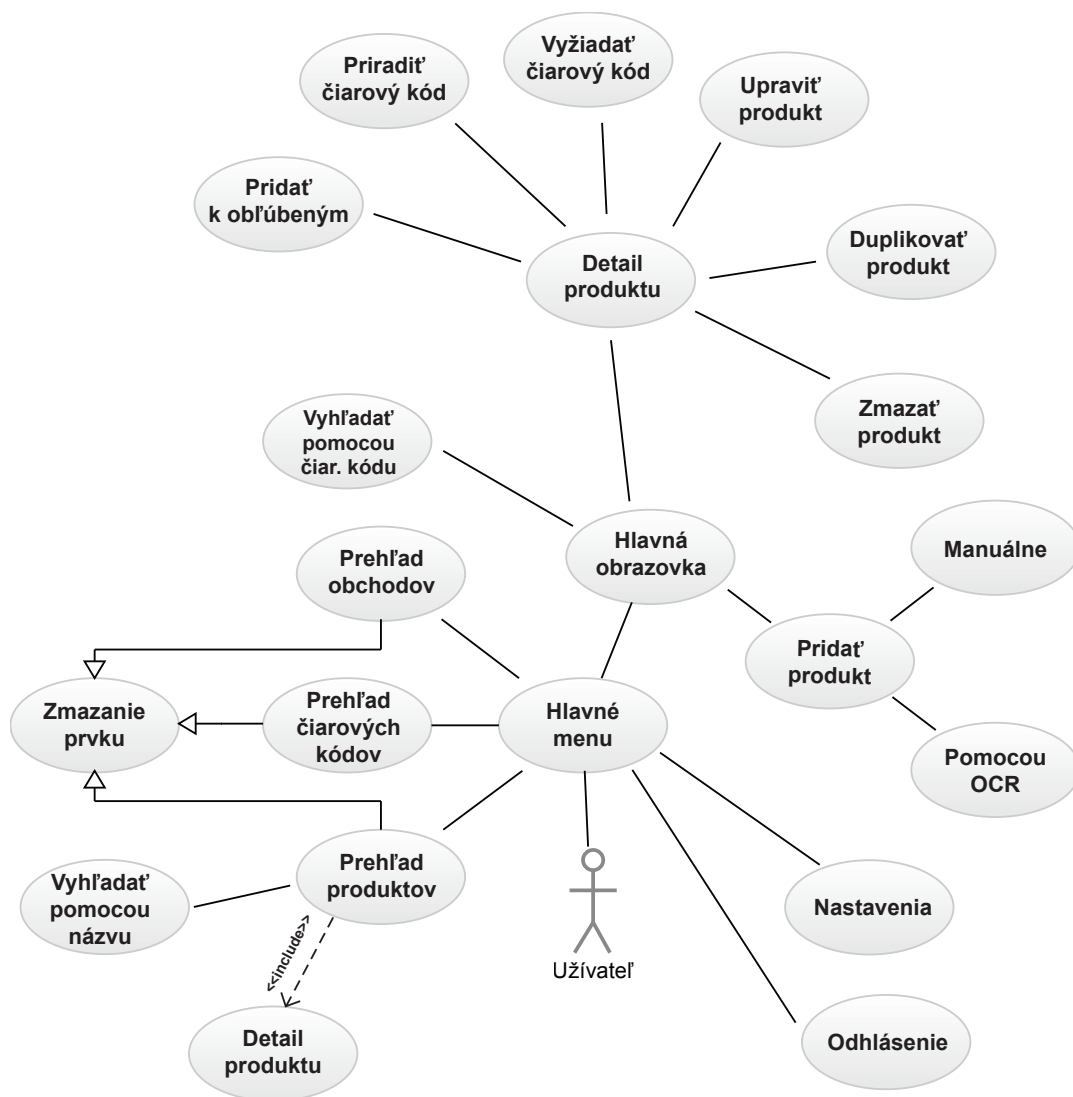
### 5. Priradenie čiarového kódu k produktu

Poskytnúť užívateľovi možnosť priradiť k produktu čiarový kód, pomocou ktorého bude následne možné produkt vyhľadať.



## 2.1 Diagram prípadov použitia

Obrázok 2.1 zobrazuje diagram prípadov použitia, ktorý zachytáva jednotlivé funkčné požiadavky na systém. Pracuje s ním len jediný typ užívateľa, a tým je samotný užívateľ aplikácie. Z hlavného menu má používateľ prístup k takmer všetkým prvkom aplikácie, a to k prehľadu obchodov, čiarových kódov a produktov, odkiaľ má možnosť zobraziť si detailný popis produktu, ako aj k nastaveniam aplikácie, odhláseniu sa z aplikácie a k hlavnej obrazovke, ktorá ponúka možnosť vyhľadávania pomocou čiarového kódu.



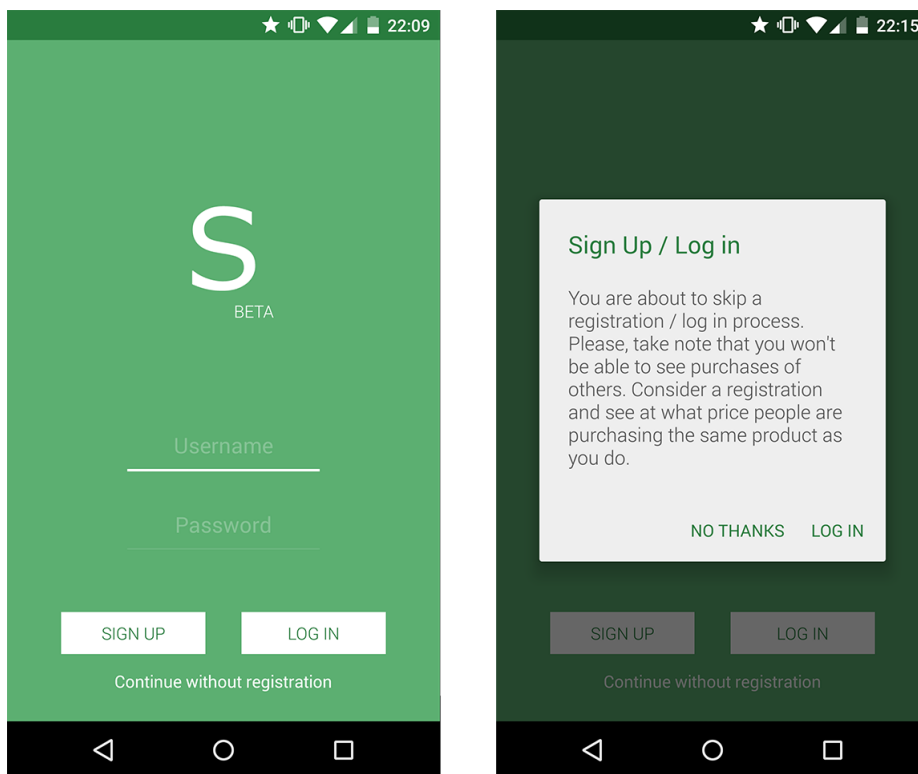
Obrázok 2.1: Diagram prípadov použitia pre aplikáciu

## 2.2 Uživatelské rozhranie

Táto časť práce popisuje návrh jednotlivých častí aplikácie spolu s popisom ako medzi sebou dané časti interagujú.

### 2.2.1 Úvodná obrazovka

Úplne prvou obrazovkou, ktorá sa užívateľovi naskytne po zapnutí aplikácie je úvodná obrazovka, zobrazená na obrázku 2.2.1 vľavo obsahujúca editačné polia pre vyplnenie registračných údajov. Ponúka možnosť zaregistrovať sa alebo prihlásiť sa už s existujúcim účtom. K týmto úkonom je však potrebné internetové pripojenie, nakoľko sa registračné údaje nachádzajú na vzdialenom serveri. Užívateľ taktiež nadobúda možnosť pokračovať vo využívaní aplikácie aj bez registrácie. Pred týmto úkonom je však oboznámený s obmedzením, ktoré ho pri tejto voľbe bude sprevádzať a to pomocou dialogového okna na obrázku 2.2.1 vpravo, ktoré sa zobrazí nad úvodnou obrazovkou. Týmto obmedzením je absencia zobrazovania cien nákupov iných užívateľov pri zobrazení detailného popisu produktu alebo vyhľadávani produktu. Systém však aj naďalej bude v prítomnosti internetového pripojenia anonymne odosielať informácie o nákupoch, ktoré sú následne dostupné registrovaným užívateľom.

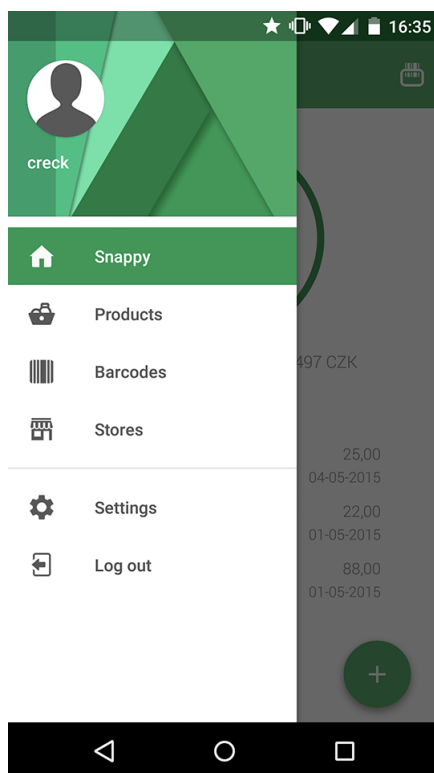


Obrázek 2.2: Úvodná obrazovka a varovné dialógové okno

### 2.2.2 Hlavné menu

Nasledujúcim prvkom grafického rozhrania je hlavné menu nachádzajúce sa stále nad aktuálnou obrazovkou. Zobraziť ho je možné dvoma spôsobmi, a to buď gestom pohybu -

potiahnutím prsta po obrazovke zľava doprava alebo kliknutím na názov aplikácie v ľavej hornej časti hlavnej obrazovky. Následné ukrytie je pomocou navigačného tlačidla späť, gestom pohybu prsta po obrazovke zprava doľava alebo kliknutím mimo plochu vyhradenú pre zobrazenie položiek nachádzajúcich sa v menu. V hornej časti je zobrazená fotografia používateľa spolu s menom, ktoré bolo vyplnené pri registrácii. Následne sú zobrazené tlačidlá k prístupom k jednotlivým častiam aplikácie, a to k prehľadu zakúpených produktov, prehľadu uložených čiarových kódov a obchodov. Napokon nasledujú nastavenia a odhlásenie sa z aplikácie. Na obrázku 2.2.2 je znázornená ukážka hlavného menu.

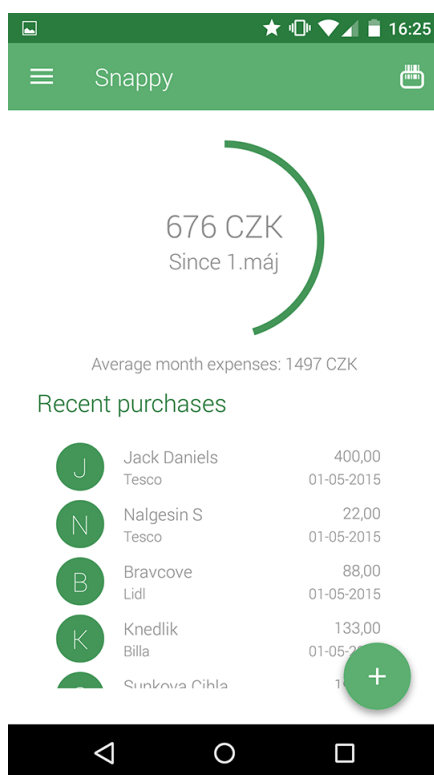


Obrázek 2.3: Hlavné menu aplikácie

### 2.2.3 Hlavná obrazovka

Na hlavnej obrazovke znázornenej na obrázku 2.2.3 je užívateľovi poskytnutý prehľad o naposledy zakúpených produktoch, aktuálnej spotrebe v danom mesiaci, a to počnúc prvým kalendárnym dňom, ako aj informácie o priemernej mesačnej spotrebe od vloženia prvého produktu. Percentuálny pomer aktuálnej mesačnej spotreby voči priemernej mesačnej spotrebe je vyobrazený pomocou kružnice, ktorej príslušná časť sa vykreslí v závislosti od vypočítanej hodnoty definujúcej veľkosť pomeru týchto dvoch hodnôt. V pravej dolnej časti obrazovky sa nachádza tlačidlo umožňujúce pridávanie nových produktov, a to pomocou manuálneho zadávania informácií o produkte a jeho zakúpení alebo využitím obrazového spracovania podkladničného bloku. V tomto prípade však užívateľ musí vyfotiť pokladničný blok pomocou kamery, zvoliť si časť pokladničného bloku, v ktorej sa nachádza zoznam zakúpených položiek a systém následne tieto položky spracuje a predpripraví ich užívateľovi na uloženie. V pravej hornej časti úvodnej obrazovky sa taktiež nachádza tlačidlo spúšťajúce

skener čiarových kódov, ktorý umožňuje komfortne a rýchlo vyhľadať požadovaný produkt, a to ako v lokálnej databáze produktov, tak aj vzdialenej za podmienky, že je užívateľ zaregistrovaný a prihlásený do aplikácie, a má prístup k internetovému pripojeniu.



Obrázek 2.4: Hlavná obrazovka aplikácie

#### 2.2.4 Pridávanie nových produktov

Po spustení aplikácie a zobrazení hlavného okna má užívateľ možnosť pridať do aplikácie nové produkty a tým rozšíriť svoje portfólio záznamov. Túto akciu je možné vykonať dvoma spôsobmi, a to ručným vyplnením vlastností produktu do predprípravených polí alebo pomocou automatizovaného systému, ktorý roztriedi jednotlivé produkty odфотографované z pokladničného bloku pomocou fotoaparátu. Obe z možností je možné vykonať pomocou tlačidla so znakom plus v pravom dolnom rohu hlavnej obrazovky, ktoré sa expanduje, a poskytne užívateľovi špecifickejšie tlačidlá. Obrázok 2.2.4 zobrazuje expandované tlačidlo pre pridávanie nových produktov.



Obrázek 2.5: Tlačidlá pre pridávanie nových produktov

- **Pridávanie nových produktov manuálne**

Možnosť ponúknutá užívateľovi pridávať nové produkty do aplikácie manuálne, spočíva v presmerovaní do okna aplikácie, ktoré obsahuje editačné polia, do ktorých následne užívateľ vpíše jednotlivé vlastnosti vkladaneho produktu. Každé z polí je povinné a bez jeho vyplnenia systém neumožní užívateľovi vložiť nový produkt do aplikácie. V prípade nekorektných alebo chýbajúcich údajov bude užívateľ o tomto stave oboznámený varovnou hláškou v spodnej časti obrazovky, ktorá však po čase zmizne. Užívateľ má takto možnosť upraviť chybné alebo chýbajúce polia a skúsiť vložiť produkt opätovne. Jednotlivé polia, ktoré je v prípade vkladania nového produktu manuálne potrebné vyplniť sú:

- *Názov produktu*  
Jedná sa o názov produktu, ktorý bude zobrazovaný v aplikácii, a pomocou ktorého sa vyhľadáva zhoda medzi produktami vloženými inými užívateľmi na vzdialenom serveri. Tento názov nemusí byť unikátny, nakoľko sa v prípade už existujúceho produktu s takýmto názvom priradí novovkládaný produkt do danej skupiny.
- *Cena produktu*  
Cena vkladaneho produktu sa zaokrúhľuje na dve desatinné miesta, a suma musí byť v rozmedzí 0 - 999 999 korún.
- *Obchod*  
Toto editačné pole slúži k špecifikovaniu, v ktorom z obchodov bol produkt zakúpený. Aplikácia si navyše uchováva jednotlivé názvy obchodov, ktoré užívateľ počas používania do tohto editačného poľa vpísal, a následne mu ich pri písaní zobrazuje ako návrhy pre rýchle a jednoduché vyplnenie.
- *Dátum zakúpenia* Dátum zakúpenia produktu je automaticky prednastavený na súčasný dátum, no naďalej je poskytnutá užívateľovi možnosť po kliknutí na hodnotu v tomto editačnom poli vloženú hodnotu upraviť. Spôsob zvolenia si nového dátumu zakúpenia je pomocou dialogového okna, ktoré graficky zobrazuje kalendár.

Po vyplnení a potvrdení všetkých potrebných údajov sa nový produkt vloží do databázy aplikácie, ako aj súčasne odošle na vzdialený server a čaká na odpoveď zo servera, ktorá obsahuje identifikačné číslo práve vloženého produktu. Toto číslo sa následne priradí k nami vytvorenému produktu v databáze aplikácie, a tým vznikne spojenie medzi produktom v lokálnej databáze a produktom vo vzdialenej databáze. Takto bude aj naďalej možné aktualizovať jednotlivé produkty súčasne v oboch da-

tabázach, ako aj vymazovať produkty v oboch databázach zároveň.

- **Pridávanie nových produktov pomocou odfotoграфovania pokladničného bloku**

Systém pridávania nových produktov pomocou odfotoграфovania pokladničného bloku spočíva vo vyfotení pokladničného bloku získaného z pokladne pri zakúpení tovaru od predajcu a jeho následnom spracovaní. Užívateľ má možnosť vyfotografovať získaný pokladničný blok pomocou vstavaného fotoaparátu v zariadení, ktoré sa aktivuje stlačením tlačidla so znakom fotoaparátu, zobrazeného na obrázku 2.2.4.

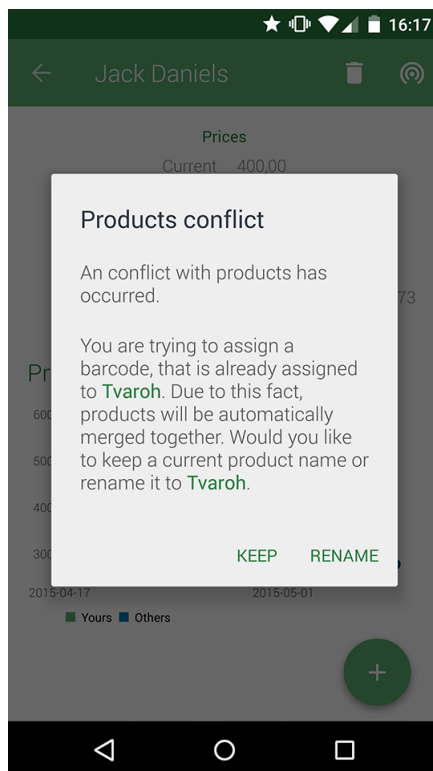
Tento spôsob poskytuje užívateľovi rýchle a komfortné vkladanie väčšieho množstva produktov len za pomoci niekoľkých kliknutí. Treba však poznamenať, že v dôsledku ešte stále sa vyvíjajúceho systému optického rozoznávania znakov je táto metóda nepresná a názvy a ceny produktov sa nemusia presne zhodovať s položkami na pokladničnom bloku. Úplný popis implementácie pridávania nových produktov pomocou odfotoграфovania pokladničného bloku je popísaný v časti implementácie v kapitole 3.2.4.

### 2.2.5 Detailný popis produktu

Detailný popis produktu zobrazený na obrázku 2.7 poskytuje užívateľovi podrobné informácie o vybranom produkte. Zobrazené sú taktiež štatistické údaje, a to priemerná cena produktu vypočítaná z danej skupiny produktov - teda zo skupiny produktov s rovnakým názvom, maximálna cena z tejto skupiny, a taktiež najvýhodnejšia cena, za ktorú bol tento produkt zakúpený. V spodnej časti obrazovky sa nachádza graf vývoja cien, v horizontálnej osi počínajúc najstarším produktom až po ten najnovší. V prípade, že užívateľ využil možnosť zaregistrovať sa a je prihlásený do aplikácie pod daným prihlasovacím menom a zariadenie má prístup k internetovému pripojeniu, v grafe sa navyše zobrazia ceny produktov s rovnakým názvom alebo čiarovým kódom, avšak zakúpených inými užívateľmi. Týmto sa užívateľovi poskytuje úplny prehľad o vývoji cien daného produktu, ako aj o rozdieloch cien v rôznych predajniach.

V ľavej spodnej časti obrazovky sa nachádza tlačidlo úprav, ktoré sa po kliknutí roztvorí a ponúka užívateľovi na výber niekoľko druhov úprav aktuálne zobrazeného produktu. Prvým z tlačidiel, označeným číslom 1 na obrázku 2.8, je tlačidlo úprav vlastností produktu. Nasledujúcim tlačidlom číslo 2, zobrazeným na obrázku 2.8, má užívateľ možnosť pridať produkt medzi obľúbené produkty. Táto voľba sa však vzťahuje na celú skupinu produktov s rovnakým názvom nielen na práve vyobrazený.

Každý skupine produktov s rovnakým názvom má užívateľ taktiež možnosť priradiť čiarový kód, a to za pomoci skenera, ktorý je možné aktivovať pomocou tlačidla číslo 3 na obrázku 2.8. Ak sa naskenovaný čiarový kód v lokálnej databáze zariadenia zatiaľ nenachádza, systém ho automaticky vloží ako nový záznam a priradí k nemu aktuálny názov produktu. Takýto záznam sa navyše v databáze označí ako **confirmed** - hlavný názov prisluhujúci čiarovému kódu. Ak by však nastala situácia, že sa naskenovaný čiarový kód v databáze už nachádza a je priradený k niektorému z produktov, aplikácia automaticky detekuje konflikt čiarových kódov a upozorní naň užívateľa. To sa vykoná pomocou dialogového okna, ktoré obsahuje označenie problému spolu s krátkym popisom vzniknutej situácie, kde sa užívateľovi zobrazí a priblíži vzniknutý konflikt. Obsahom okna zobrazujúceho upozornenie je taktiež názov produktu, s ktorým prišiel aktuálny produkt do konfliktu. Toto dialógové okno nachádzajúce sa nad detailným popisom produktu, zobrazené na obrázku 2.2.5, ponúka užívateľovi na výber z dvoch možností, ako vyriešiť vzniknutú situáciu.



Obrázek 2.6: Vzniknutý konflikt produktov

- **Zachovanie súčasného názvu produktu**

Jedným z riešení vyskytnutého sa konfliktu medzi čiarovými kódmi je, že sa produktu ponechá aktuálny názov. Avšak produkt, ktorý je v konflikte s aktuálnym produktom sa premenuje a zlúči s aktuálnym. Nakoľko sa v databáze naskenovaný čiarový kód už nachádzal, záznam sa upraví, a to tak, že sa z neho odstráni označenie **confirmed**, ktoré poukazuje na to, že ide o hlavný názov produktu, priradený k danému čiarovému kódu. Zároveň sa však vloží nový záznam už s označením **confirmed**, a to ako kombinácia názvu aktuálneho produktu a naskenovaného čiarového kódu. Týmto spôsobom sa vytvára databáza čiarových kódov, ako kombinácia čiarového kódu a názvu produktu, pričom stále iba jeden záznam v skupine čiarových kódov s rovnakým číslom je označený ako **confirmed**. Zvyšné záznamy uchovávajú alternatívne mená produktov prisluhujúce jednotlivým čiarovým kódom.

- **Premenovanie súčasného produktu**

Druhou z možností, z ktorých má užívateľ na výber pri vyskytnutom sa konflikte čiarových kódov je tá, že aktuálny produkt sa premenuje na názov produktu, s ktorým je v konflikte a následne sa tieto dve skupiny produktov zlúčia dohromady. Spolu s tým je spojená úprava záznamov čiarových kódov v databáze, a to týmto spôsobom, že záznam s čiarovým kódom, ktorého číslo je rovnaké ako číslo naskenovaného čiarového kódu sa neupravuje, keďže hlavný názov produktu, priradený k tomuto čiarovému kódu (záznam **confirmed**) sa nemení a zostáva mu naďalej priradený. Overí sa však, či sa v databáze nachádza kombinácia naskenovaného čiarového kódu a názvu súčasného produktu. V prípade, že by bol výsledok negatívny, do databázy sa vloží nový záznam, a to práve so spomínanou kombináciou naskenovaného čiarového kódu a ná-

zvom súčasného produktu. Takýto záznam avšak zostáva v databáze len ako záznam označujúci alternatívne meno produktu prisluhujúce k danému čiarovému kódu.

Okrem spomínaných dvoch prípadov priradovania čiarového kódu k produktu, kedy sa skenovaný čiarový kód v databáze vôbec nenachádza alebo ak sa čiarový kód už v databáze nachádza, a je priradený k istému produktu, existuje aj prípad, kedy sa naskenovaný čiarový kód v databáze síce nachádza ale nie je priradený k žiadnému z produktov. V tomto prípade dochádza k úprave záznamov v databáze, a to nasledovným spôsobom. Pre daný čiarový kód sa vyhledá záznam **confirmed** a zruší sa z neho toto označenie. Následne sa skontroluje či sa v databáze nachádza kombinácia čiarového kódu a názvu produktu. V prípade, že sa spomínaná kombinácia v databáze nenachádza, vytvorí sa nový záznam označený ako hlavný názov produktu k naskenovanému čiarovému kódu, teda záznam **confirmed**. Ak by sa takáto kombinácia čiarového kódu a názvu produktu v databáze už nachádzala, tento záznam sa len upraví, a to spôsobom označenia záznamu ako **confirmed**.

Následujúca tabuľka 2.1 zobrazuje ukážku princípu vkladania záznamov čiarových kódov do databázy.

<b>barcode</b>	<b>name</b>	<b>confirmed</b>
8718265743403	Tvaroh	true
8718265743403	Tvaroh mäkký	false
8718265743403	Tvaroh hrúdkovaný	false
8718265743403	Tvrh	false
8718265743403	Tvaroh light	false
8808992091400	Persil	false
8808992091400	Persil White	true

Tabuľka 2.1: Ukážka formátu vkladania záznamov čiarových kódov do databázy

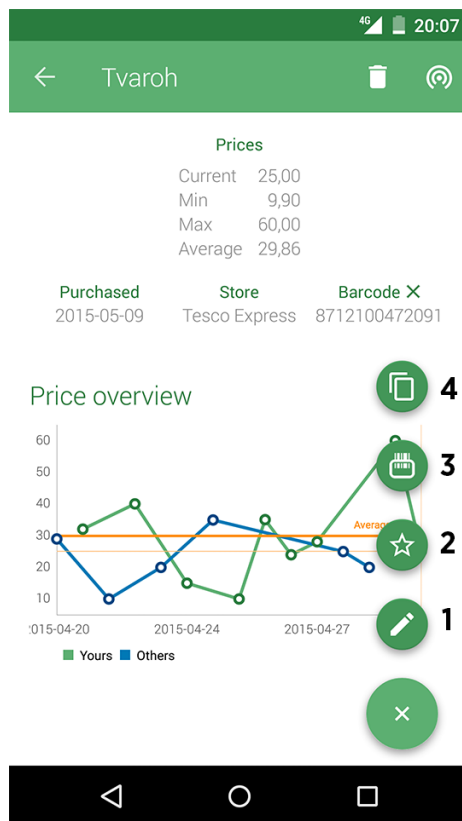
Dôvodom vytvárania a uchovávanía záznamov zobrazujúcich alternatívne mená produktov, pridelených k jednotlivým čiarovým kódom je ten, že v prípade, ak sa užívateľ rozhodne do aplikácie vložiť nový produkt a následne si zobrazí jeho detailný popis, systém sa automaticky na pozadí pokúsi vyhledáť v databáze čiarových kódov zhodu v názve produktu a názvoch pridelených k čiarovým kódom. V prípade zhody, sa následne zo zhodujúcich záznamov vyberú len tie čiarové kódy, ktoré sú označené ako **confirmed**, teda jedná sa o tie čiarové kódy, ktorých pridelené hlavné meno produktu je zhodné s vyhledávaným. Ak by sa však medzi zhodujúcimi sa záznammi nachádzali aj záznamy označujúce alternatívne meno produktu, teda nie sú označené ako **confirmed**, vyhledajú sa k týmto čiarovým kódom záznamy typu **confirmed**. Výsledná množina získaných záznamov sa pri zobrazení detailného popisu produktu poskytne užívateľovi ako návrh čiarových kódov, ktoré je možné priradiť k produktu, a to aj bez opätovného skenovania. Tento návrh je zobrazený v dialógovom okne nachádzajúcom sa nad detailným popisom produktu, a to s krátkym popisom vznikutej situácie, zobrazením návrhov čiarových kódov ako dvojíc tvoriacich číslom čiarového kódu a hlavným menom produktu, no taktiež aj s možnosťami, ktoré sa ponúkajú užívateľovi na výber. Jednou z týchto možností je práve spomínaná možnosť potvrdenia jedného z ponúkaných čiarových kódov, a tým poskytnutie užívateľovi efektného znovupoužitia už naskenovaných čiarových kódov. Toto priradenie však nie je záväzné a je doplnené o nasledujúce dve možnosti. Prvou z nich je odloženie výberu na neskôr, čo spôsobí, že pre aktuálne



zobrazenie detailu produktu sa toto dialogové okno skryje a zobrazí sa až pri opätovnom zobrazení detailného popisu tohto produktu. Druhou z možností je odmietnutie ponúkaných čiarových kódov úplne, a s tým spojené zabránenie opätovného získavania návrhov čiarových kódov pre daný produkt. V takomto prípade je však užívateľovi ponúknutá voľba dobrovoľne si vyžiadať návrhované čiarové kódy, a to stlačením tlačidla v pravom hornom rohu obrazovky, zobrazeného na obrázku 2.7. Práve zobrazený produkt je možné taktiež odstrániť z databázy, a to použitím tlačidla zobrazujúceho odpadkový kôš v hornej časti obrazovky. Týmto úkonom sa odstráni produkt nielen z lokálnej databázy, ale v prípade, že je daným produkt namapovaný aj do vzdialenej databázy, bude odstránený aj z nej.



Obrázek 2.7: Detailný popis vybraného produktu



Obrázek 2.8: Detailný popis produktu s expandovaným tlačidlom úprav

## 2.2.6 Úprava existujúceho produktu

Užívateľovi sa taktiež ponúka možnosť upravovať vlastnosti už existujúceho produktu, a to prechodom do editačného okna pomocou tlačidla číslo 1 na obrázku 2.8. Jednotlivé vlastnosti sa zobrazia vyplnené v editačných poliach, ktoré je následne možné prepísať. S výnimkou úpravy názvu produktu užívateľ nepotrebuje potvrdzovať žiadne dodatočné úkony, nakoľko sa jedná o úpravu práve zobrazeného produktu v detailnom popise z celej skupiny produktov. Pokiaľ sa užívateľ rozhodne zmeniť názov produktu, následne musí potvrdiť do akej miery sa má zmena názvu vykonať. Toto rozhodnutie má možnosť potvrdiť na dialogovom okne, ktoré sa automaticky vykreslí nad editačnou obrazovkou, a ktoré obsahuje krátky úvod do problému pri premenovávaní produktov. Problém spočíva v rozsahu

premenovávanie produktu, nakoľko sa môže jednať o požiadavku na zmenu názvu práve zobrazeného produktu alebo zmenu názvu celej skupiny produktov. Toto rozhodnutie má užívateľ možnosť potvrdiť práve v spomínanom dialógovom okne, a to buď výberom zmeny názvu pre práve zobrazený produkt alebo pre zmenu názvu celej skupiny produktov. V oboch prípadoch, pri premenovávaní produktu, je z produktu automaticky odstránený číarový kód a užívateľ ho teda musí opätovne produktu priradiť. V prípade, že sa užívateľ rozhodne premenovať celú skupinu produktov, systém automaticky overí či sa v databáze produktov produkt s takýmto názvom už nenachádza, a ak áno, obe skupiny produktov budú zlučené dohromady. Ak bola zvolená možnosť premenovať len práve zobrazený produkt, tento produkt je z pôvodnej skupiny produktov vyradený a systém sa mu pokúsi vyhľadať potrebnú skupinu medzi existujúcimi produktami. Ak takú skupinu nájde, odlúčený produkt sa k nej priradí a následne sa zobrazí detail tohto produktu už v novej skupine po premenovaní. Ak by sa však takáto vhodná skupina nenašla, produkt sa premenuje a sám tvorí novú skupinu produktov s daným názvom.

### 2.2.7 Duplikácia existujúceho produktu

Pre uľahčenie vytvárania nových produktov je užívateľovi ponúknutá možnosť duplikovať už existujúci produkt s následnou voliteľnou úpravou vlastností pred samotným vytvorením. Týmto spôsobom dochádza k efektívnemu a komfortnému vytváraniu nových záznamov a to len za pomoci dvoch kliknutí. Pri kliknutí na tlačidlo spúšťajúce proces duplikovania, ktoré je zobrazené ako tlačidlo číslo 4 na obrázku 2.8 dochádza k predvyplneniu vlastností produktu podľa práve zobrazeného produktu v detailnom popise. Užívateľ má následne možnosť jednotlivé vlastnosti ponechať, pozmeniť alebo ich zmeniť úplne. Pri stlačení navigačného tlačidla späť alebo stlačení tlačidla, ktoré zruší duplicitu sa údaje o produkte zahodajú a nedochádza k vytvoreniu nového produktu. Naopak, pri potvrdení vyplnených vlastností systém automaticky vyhľadá existujúcu skupinu produktov podľa názvu duplikovaného produktu a v prípade existencie takejto skupiny do nej priradí nový produkt. V opačnom prípade bude novovytvorený produkt tvoriť skupinu produktov sám.

### 2.2.8 Vyhľadávanie produktov

V dôsledku zachovania prehľadu pri zväčšujúcom sa množstve vložených produktov, ako aj rýchleho prístupu k vyžadovaným detailom sa užívateľom aplikácie ponúka možnosť vyhľadávania produktov ako v lokálnej, tak aj vo vzdialenej databáze. Toto vyhľadávanie je prístupné v dvoch častiach aplikácie.

- **Vyhľadávanie podľa názvu v zozname produktov**

Po prístupe k úplnému zoznamu produktov pomocou zobrazenia hlavného menu a zvolením voľby Produkty (angl. Products) sa užívateľovi zobrazí úplný zoznam skupín produktov, zoradený podľa ich dátumu zakúpenia zostupne. Nakoľko môže tento zoznam nadobudnúť veľkú dĺžku, je preň vytvorené vyhľadávanie podľa názvu produktu. Toto vyhľadávanie sa aktivuje po kliknutí na tlačidlo označené lupou v pravo hornom rohu obrazovky a umožní sa zadávanie názvu produktu do editačného poľa. Počas vpisovania jednotlivých písmen sa zoznam automaticky aktualizuje, čo umožňuje vyhľadávanie medzi produktami aj v prípade, že si užívateľ presne nepamätá úplný názov produktu, ale len jeho časť. Po uzavretí vyhľadávacieho panelu sa zoznam automaticky vráti do pôvodnej podoby, zobrazujúci úplný zoznam produktov.

Pri kliknutí niektorý z produktov v zozname bude užívateľ presmerovaný na detailné zobrazenie daného produktu, popísané v kapitole 2.2.5.

- **Vyhľadávanie pomocou nasnímania čiarového kódu**

Pri zobrazení hlavnej obrazovky, popísanej v kapitole 2.2.3, sa v pravom hornom rohu, pomocou tlačidla zobrazujúceho skener, ponúka užívateľovi špeciálny typ vyhľadávania produktov, a to pomocou skenera čiarových kódov. Toto tlačidlo spustí skener, ktorý po zameraní na čiarový kód, zosníma potrebné údaje a pokúsi sa podľa nich vyhľadať adekvátny produkt v aplikácii. V prípade, že je užívateľ zaregistrovaný a prihlásený do aplikácie, umožňuje sa mu vyhľadávanie produktov aj na vzdialenom serveri, čo v konečnom dôsledku umožní zobrazenie cien nákupov aj iných užívateľov pre daný produkt. Tento spôsob vyhľadávania si však vyžaduje, aby bol k produktu priradený čiarový kód. Pri úspešnom vyhľadávaní a podľa výsledkov vyhľadávania môže aplikácia zobraziť výsledné údaje dvoma spôsobmi.

1. ***Vyhľadávaný produkt sa nachádza v aplikácii***

Ak užívateľ do svojej aplikácie pridal produkt, ktorý sa následne snaží vyhľadať práve pomocou čiarového kódu, výsledné údaje sa mu zobrazia ako detailný popis produktu, v ktorého grafe vývoja cien budú zahrnuté aj nákupy iných užívateľov, ktorí k svojim produktom priradili práve hľadaný čiarový kód. Zobrazenie cien nákupov iných užívateľov sa však spája s nutnosťou prihlásenia do aplikácie a prítomnosti internetového pripojenia.

2. ***Vyhľadávaný produkt sa nenachádza v aplikácii***

Ak sa užívateľ pokúsi vyhľadať produkt na základe čiarového kódu, ktorý však nie je priradený k žiadnému z produktov uchovaných v aplikácii, systém sa automaticky pokúsi vyhľadať produkty s takýmto čiarovým kódom v vzdialenej databáze. K tomuto úkonu je však potrebné internetové pripojenie a nutnosť prihlásenia do aplikácie, nakoľko anonymným užívateľom je tento benefit odobraný. Ak niektoré z produktov vo vzdialenej databáze majú pridelený práve požadovaný čiarový kód, informácie o nich sa stiahnu do zariadenia a zobrazia v dialógovom okne nad hlavnou obrazovkou. Toto dialógové okno poskytuje stručné informácie o vyhľadávanom produkte, ako aj graf vývoja cien daného produktu. Táto možnosť vyhľadávania môže byť obzvlášť užitočná, ak sa užívateľ nachádza priamo v obchode a potrebuje rýchlo a efektívne získať informácie o cenách požadovaného produktu.

## Kapitola 3

# Implementácia

Táto kapitola sa venuje implementácií aplikácie pre uchovávanie a správu cien produktov. V úvodnej časti je opísaný framework operačného systému Android. Následne sú opísané jednotlivé technológie, využívané pri vývoji aplikácie, spolu s popisom implementácie služby fungujúcej na vzdialenom serveri.

### 3.1 Android Framework

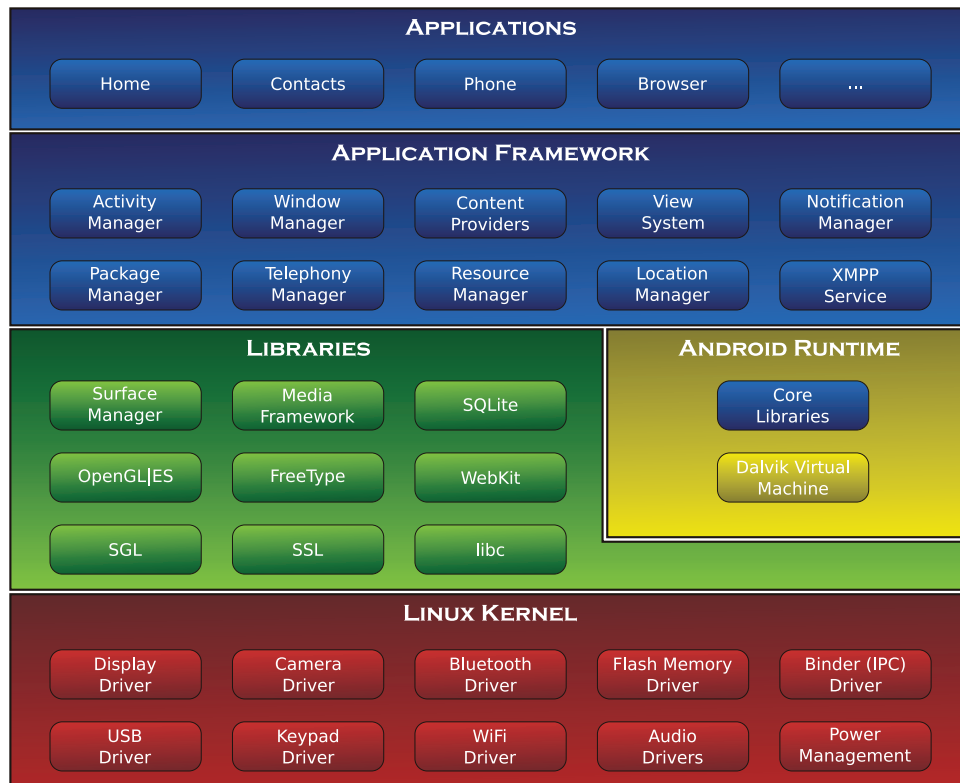
Operačný systém Android je open source projekt, založený na jadre Linuxu, ktorý je určený najmä pre mobilné zariadenia. Ide o najrozšírenejší operačný systém využívaný na mobilných zariadeniach, ktorý je vyvíjaný alianciou Open Handset Alliance a firmou Google Inc.. Avšak zavádzajúce je tvrdiť, že systém Android je postavený výlučne na jadre systému Linux, nakoľko sa nejedná len o samotné jadro, ale aj o medzivrstvu medzi jadrom a aplikáciami (tzv. middleware), knižnice, API a taktiež o samotné aplikácie. Všetky telefóny používajúce tento operačný systém obsahujú taktiež niekoľko predinštalovaných aplikácií.

Užívatelia majú ľahký prístup k získaniu nových aplikácií a to prostredníctvom inštaláčnych súborov uložených priamo v ich zariadení, alebo za pomoci služby Obchod Play. V súčasnosti služba Obchod play ponúka približne 1 300 000 aplikácií. Obe spôsoby si vyžadujú minimum technických zručností užívateľa nakoľko sa systém na pozadí automaticky postará o všetky potrebné inštaláčne procesy, a to dokonca aj o výber miesta, kde bude nová aplikácia nainštalovaná. Jedinou požiadavkou, ktorá sa kladie na samotného používateľa je súhlas s prístupom aplikácie k jednotlivým zdrojom zariadenia, ako sú napríklad kontakty, kalendár, senzory, fotografie a pod.

Nepochybne je dôležité spomenúť, že v systéme Android sú si všetky aplikácie rovné. Nezáleží na tom či sa jedná o aplikáciu vyvinutú treťou stranou alebo priamo tvorcami systému, obe pracujú v rovnakom prostredí a obe majú rovnaký prístup k zdrojom zariadenia. Prostredie, v ktorom pracujú sa nazýva sandbox, pričom ide o izolovanú časť systému, ktorá nemá prístup k zdrojom zariadenia, s výnimkou priameho udelenia práv užívateľom pri inštalácii danej aplikácie.

Narozdiel od iných operačných systémov určených pre mobilné zariadenia je Android open source, čo znamená, že zdrojové kódy sú prístupné verejnosti, to umožňuje vývojárom po celom svete nielen bližšie nahliadnúť a pochopiť čo sa deje na pozadí systému ale sa aj

zapojiť do vývoja samotného systému.



Obrázek 3.1: Architektúra systému Android

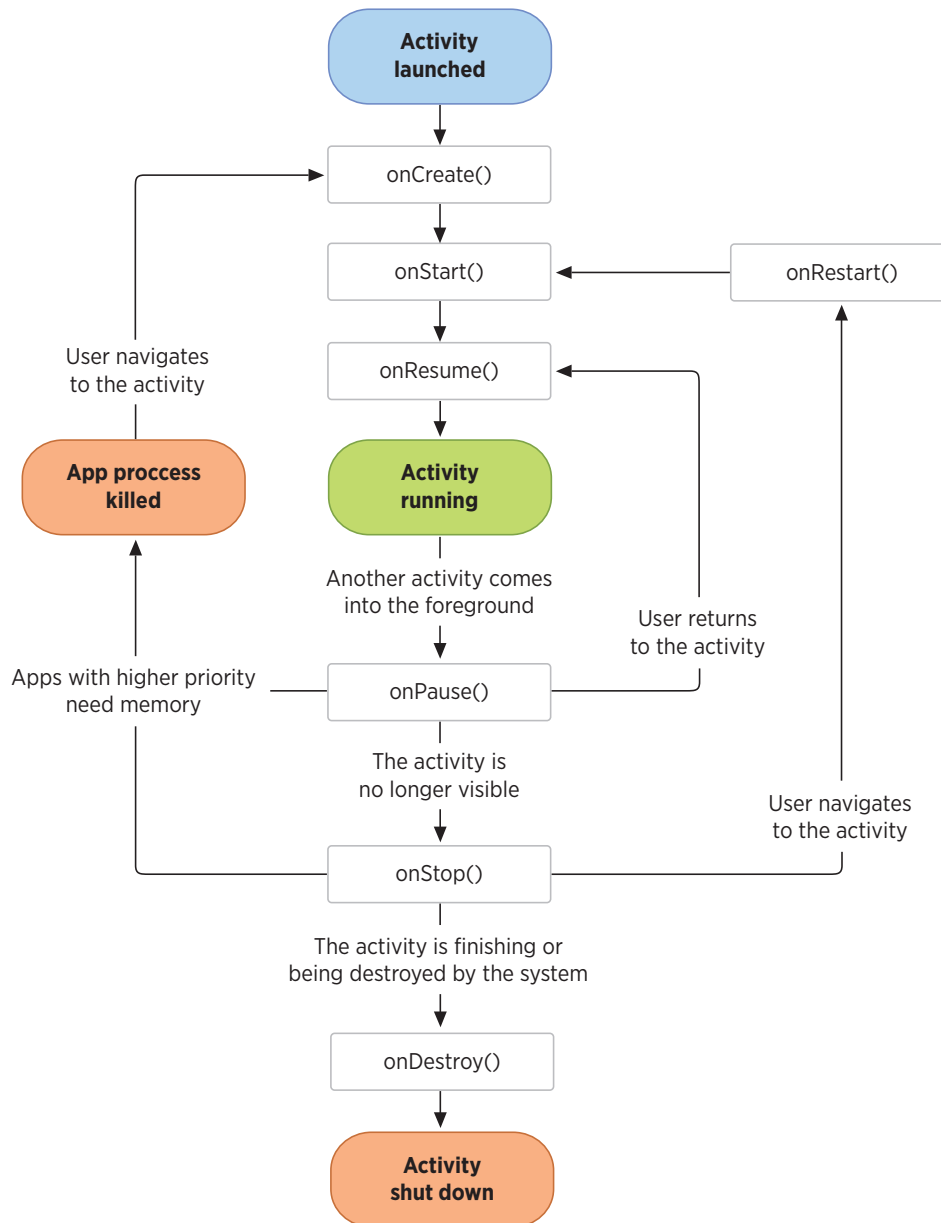
### 3.1.1 Komponenty

Každá aplikácia pozostáva zo základných komponentov. Android figuruje štyrmi druhmi komponentov, pričom každý z nich je iný, má iný účel a životný cyklus, ktorý určuje ako je komponent vytvorený a ako zaniká. Výnimočným aspektom systému Android je, že každá aplikácia môže spustiť komponent inej aplikácie.

- *Activity* - aktivity v systéme nie sú na sebe navzájom závislé, a to aj v prípade, že spoločne vytvárajú aplikáciu. Zvyčajne obsahujú grafické prvky a fragmenty na vytvorenie užívateľského rozhrania a interakcie s užívateľom. Aktivita sa môže nachádzať v stave:
  - *Aktívna* - v tomto štádiu je aktivita aktívna a nachádza sa na popredí obrazovky
  - *Pozastavená* - v prípade, že je aktivita čiastočne prekrytá inou (priehľadnou alebo zmenšenou) je aktivita pozastavená. V tomto štádiu je stále plne funkčná, naďalej obsahuje príslušné informácie, avšak pri extrémnom nedostatku pamäte môže byť systémom zrušená
  - *Zastavená* - tento stav nastáva ak je aktivita úplne prekrytá inou, teda už nie je naďalej viditeľná. Stále si však uchováva svoje informácie, no v prípade nedostatku pamäti je zrušená

- o *Neaktívna* - aktivita nachádzajúca sa v stave pred spustením, alebo ak bola zrušená buď na požiadanie o ukončenie systémom alebo v dôsledku nedostatku pamäte

Nakoľko ide o jednu zo základných komponentov aplikácie, nasledujúci obrázok 3.2 popisuje životný cyklus každej z aktivít, od vytvorenia až po samotný zánik.



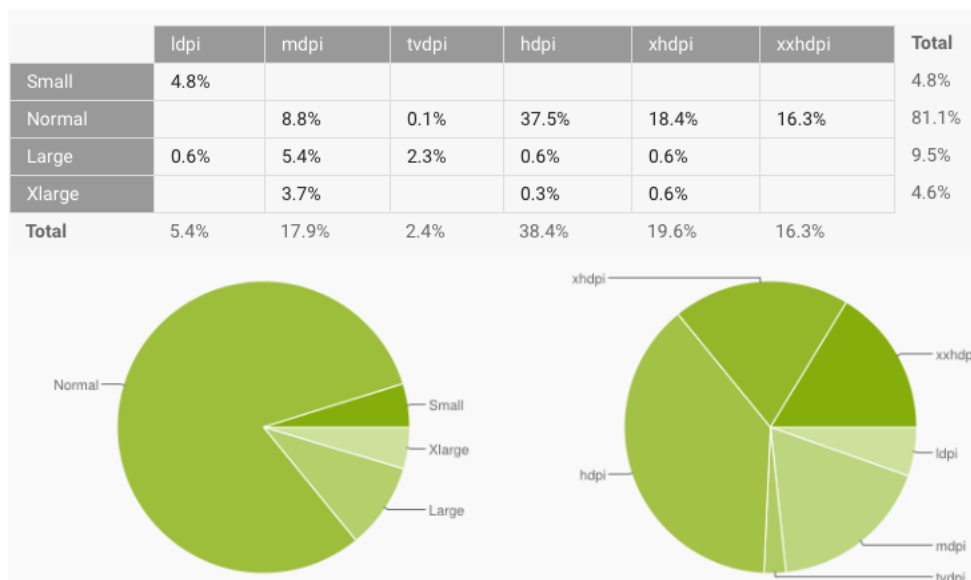
Obrázek 3.2: Životný cyklus aktivity

- *Service* - samostatná časť aplikácie, ktorá beží na pozadí a vykonáva úlohy bez vytvárania užívateľského rozhrania. V prípade zložitých výpočtov by mala pracovať na samostatnom vlákne. S ostatnými komponentami aplikácie komunikuje pomocou komponenty Broadcast Receiver a užívateľa je možné upovedomiť pomocou frameworku určeného na oznámenia.
- *Broadcast Receiver* - reaguje na správy zaslané inou aplikáciou, alebo samotným systémom. Tieto správy sa niekedy označujú ako events alebo intents. Tento komponent nemá vlastné užívateľské rozhranie, no určitým spôsobom môže informovať užívateľa o vykonávaní činnosti napríklad pomocou oznámení.
- *Content provider* - na požiadanie sprístupňuje dáta navzájom medzi aplikáciami. Dáta je možné ukládať do súborov, databázi alebo aj na vzdialené uložisko. Každá aplikácia pracuje s vlastnými právmi, procesmi a dátami, ktoré sú pred inými aplikáciami ukryté. Avšak v prípade potreby zdieľania dát medzi aplikáciami nachádza Content provider svoje uplatnenie. Pre vytvorenie požiadavky sa používa reťazec vo forme URI s nasledujúcim formátom:

`content://authority/path/id`

### 3.1.2 Rôzne veľkosti obrazoviek

Nakoľko ide o rozšírený operačný systém pracujúci na rôznych zariadeniach, pri vývoji aplikácie je potrebné mať na zreteli nielen škálu podporovaných verzií systému, ale aj špecifikácie cieľových zariadení. Pravdepodobne najdôležitejšou z týchto špecifikácií je veľkosť obrazovky a hustota pixelov na fyzickej časti displeja. Kým fyzická veľkosť obrazovky je rozdelená do štyroch základných kategórií `small`, `normal`, `large`, `xlarge`, hustota pixelov člení zariadenia do nasledujúcich skupín: `ldpi`, `mdpi`, `hdpi`, `xhdpi`, `xxhdpi`, `xxxhdpi`. Nasledujúci graf 3.3 zobrazuje, že väčšina zariadení disponuje obrazovkou s vysokou alebo normálnou hustotou pixelov, avšak je dobrým zvykom podporovať čo najširšiu škálu zariadení. V dôsledku rôznych veľkostí obrazoviek má Android špecifikovanú vlastnú jednotku vyjadrujúcu rozmer, nazývanú `density pixel` (`dp`). Jedná sa o abstraktnú jednotku založenú na hustote obrazových bodov fyzickej obrazovky. Jej veľkosť je relatívna k obrazovke s hustotou 160dpi, takže na takejto obrazovke vyjadruje 1 klasický obrazový bod taktiež jeden `density pixel`. Pomer zvyšovania veľkostí rozmeru jedného `density pixel` rastie spolu s veľkosťou hustoty obrazových bodov na obrazovke, avšak nie priamoúmerne.



Obrázek 3.3: Relatívny počet zariadení so špecifickou veľkosťou obrazovky a hustotou pixelov (5. január 2015), zdroj: *Android developers* <http://developer.android.com/about/dashboards/index.html>

## 3.2 OCR

System optického rozoznávania znakov z textu (z angl. Optical Character Recognition) je technológia, ktorá umožňuje konvertovať rôzne typy dokumentov do digitálnej podoby reprezentovanej postupnosťou ASCII<sup>1</sup> znakov. Prvé optické rozoznávanie textu siahá až do dôb vývoja telegrafov a vytvárania čítacích zariadení pre nevidiacich. [4] Koncom 20-tych rokov 19. storočia Emanuel Goldberg vyvinul prístroj, ktorý prehľadával archív mikrofilmov na základe optického rozoznávania znakov.

V súčasnosti však softvér využívajúci túto technológiu v sebe musí obsahovať rôzne lingvistické prvky, prvky umelej inteligencie, ako aj prvky strojového videnia, ktoré po spoločnej kooperácii dosiahnu požadovaný výsledok. Tento proces však pozostáva z dvoch krokov, pričom každý z nich je skupinou algoritmov aplikujúcich sa na vstupný súbor. Súbor teda prechádza pred samotným rozoznávaním textu niekoľkými fázami spracovania, čo prispieva k zvýšenej presnosti rozoznávania znakov zo vstupu. Každá z tejto fázy je však veľmi dôležitá, nakoľko celý proces rozoznávania môže zlyhať v prípade, že sa niektorá z fáz nevykoná správne. Každý z algoritmov musí pracovať správne na čo najväčšom počte rozličných obrázkov, čo je príčinou, že v súčasnosti existuje len niekoľko systémov, ktoré tieto podmienky spĺňujú. Na získanie čo najlepších výsledkov musí systém automaticky nastaviť dôležité parametre algoritmov pre korektné spracovanie obrázku, pričom niekedy je to jediný spôsob, ako zvýšiť kvalitu vstupného súboru. Nanešťastie, v súčasnosti však neexistuje žiaden systém na rozoznávanie znakov z textu, ktorý by bol zrovnateľný s ľudským okom.

<sup>1</sup>ASCII je kódovací systém znakov abecedy, čísel, a iných znakmi. Skratka pochádza z anglického American Standard Code for Information Interchange



### 3.2.1 Pre-processing

Prvou fázou spracovania vstupu, ktorým je súbor načítaný ako bitmapa je tzv. pre-processing, teda predpríprava vstupu pred samotným rozoznáváním. Na túto bitmapu sa aplikuje niekoľko filtrov, ktoré poupraví v prípade potreby originálny obrazok. Tieto filtre sú popísané v nasledujúcej časti.

- **Pootočenie**

V prípade, že je obrázok nesprávne natočený, systém automaticky pootočí v smere alebo v protismere hodinových ručičiek vstupný súbor tak, aby bol text úplne v horizontálnej alebo vertikálnej polohe.

- **Odstránenie šumu**

Ak vstupný obrazok obsahuje šum, či už náhodný alebo Gaussov šum, je potrebné tento nedostatok odstrániť a zvýšiť tak kvalitu obrázku pred spracovaním.

- **Binarizácia obrázku**

Slovo binarizácia referuje na prevod obrázku z farebného na čiernobiely, nazývaného tiež binárny obrazok, nakoľko sa v ňom nachádzajú len dve farby. Tento proces pozostáva z niekoľkých aplikovaní filtrov na vstupný súbor, čím sa docielí jeho prevod z farebného na čiernobiely. Úlohou binarizácie je oddelenie textu od pozadia obrázku.

<sup>2</sup>Nakoľko väčšina súčasných systémov optického rozoznávania znakov pracuje práve s binarizovanou formou obrázku je tento krok nevyhnutný. Prvým z filtrov je pozmenenie farieb obrázku na rôzne odtiene šedej a vytvorenie tzv. grayscale obrázku. To je v počítačovej grafike označované ako obrázok, ktorého hodnota každého pixelu je vzorkou obsahujúcou intenzitu farby. Tieto obrázky, taktiež označované ako čiernobiele sa skladajú výlučne z odtieňov šedej, pričom čierna farba označuje najslabšiu intenzitu, a biela naopak najsilnejšiu. [1] Každý farebný obrázok pozostáva z kombinácie intezít troch základných farieb - červenej, zelenej, modrej, a to v rozsahu od 0 do 1. Pre prevod z farebnej zložky na zložku obsahujúcu výlučne odtieň sivej je použitý vážený súčet týchto troch lineárnych zložiek intenzity farieb. Výsledná intenzita farby je následne vypočítaná podľa vzorca:<sup>3</sup>

$$Y = 0.2126R + 0.7152G + 0.0722B$$

Farebný obrázok prevedený na odtiene šedej je zobrazený na obrázku 3.4



Obrázek 3.4: Prevod farebného obrázku na odtiene šedej

---

<sup>2</sup>[http://webdocs.cs.ualberta.ca/~nray1/CMPUT605/track3\\_papers/Threshold\\_survey.pdf](http://webdocs.cs.ualberta.ca/~nray1/CMPUT605/track3_papers/Threshold_survey.pdf)

<sup>3</sup><http://www.w3.org/Graphics/Color/sRGB>

Binarizácie v sebe taktiež zahŕňujú prahovanie obrazu, ktoré zredukuje grayscale obrázkov na jeho binárnu formu. Táto metóda predpokladá, že objekty v obraze sa dajú rozlíšiť na základe intenzity jasu jednotlivých obrazových bodov. Problém však nastáva pri určovaní hodnoty prahu, ktorá je hlavným faktorom úspešnosti oddelenia jednotlivých objektov od pozadia obrazu. V ideálnom prípade má však histogram hĺbokú a ostrú priepasť medzi dvoma vrcholmi reprezentujúcimi objekty a pozadie obrazu, respektíve opačne, ktorej najmenšia hodnota reprezentuje ideálnu hodnotu prahu. Avšak, pri väčšine reálnych obrazov je zložitá vypočítať túto najmenšiu hodnotu v priepasti, obzvlášť v prípadoch, keď je táto priepasť plytká, alebo ak majú jednotlivé vrcholy extrémne nerovnomernú výšku čo môže mať za následok, že žiadna priepasť sa medzi vrcholmi nebude nachádzať.<sup>4</sup> K výpočtu ideálnej hodnoty prahu je použitá Otsu metóda, ktorej algoritmus spočíva v iterovaní cez všetky možné hodnoty prahu a vypočítania veľkosti rozptýlenia medzi jednotlivými obrazovými bodmi objektov a pozadia obrazu. Cieľom je nájsť hodnotu, kde suma rozptýlenia obrazových bodov patriacich objektom a pozadiu bude čo najmenšia. Takto získanú hodnotu následne použijeme pre vypočítanie výslednej farby obrazových bodov, a to dosadením intenzity farby obrazového bodu do funkcie,

$$f(c) = \begin{cases} 1 & c \geq \text{prah} \\ 0 & c < \text{prah} \end{cases}$$

ktorá vráti hodnotu 1 alebo 0. V prípade, že výsledná hodnota je 1, farba obrazového bodu bude biela, v opačnom prípade sa farba nastaví na čiernu. Na obrázku 3.5 je zobrazené aplikovanie prahovania na grayscale obrázkov a zobrazenie ideálnej hodnoty prahu z histogramu.



Obrázek 3.5: Aplikovanie prahovania na obrázkov rôznych hodnôt šedej

- **Odstránenie čiar**

Ak sa na obrázku nachádzajú súvislé čiary, ako napríklad podčiarknutie písma alebo ohraničenie textu, je potrebné ich odstrániť a zamedziť tak prípadným artefaktom, ktoré by mohli v dôsledku toho vzniknúť.

- **Zónovanie**

Zónovanie alebo analýza dispozície je mimoriadne dôležitá v prípade rozoznávania znakov z textu, ktorý obsahuje viacero riadkov. Ide teda o detekciu a indentifikovanie jednotlivých riadkov, odstavcov, nadpisov a iných častí textu do jednotlivých blokov.

- **Detekcia slov**

Jedná sa o určenie základnej línie pre slová a tvary znakov, ako aj rozdelenie slov v

<sup>4</sup><http://web-ext.u-aizu.ac.jp/course/bmclass/documents/otsu1979.pdf>

pripade potreby

### 3.2.2 Rozoznavanie znakov

Nasledujúcou fázou spracovania vstupného súboru je samotné rozoznavanie znakov. Po získaní pripraveného binárneho obrázku z fázy pre-processingu je tento obrázok podrobený rozoznavaniu znakov. Systém dekomponuje jednotlivé piktogramy do menších elementov, ako sú čiary, uzavreté slučky, smery čiar a ich priesečníky. Tie sú porovnávané s abstraktným vektorom reprezentujúcim písmeno. Následne sú použité klasifikátory na určenie najbližšieho suseda, ako napríklad algoritmus k-najbližších susedov, ktoré určia najbližšiu zhodu medzi elementami získanými z piktogramu obrázku a elementami uchovanými v systéme.

Softwéry ako Cuneiform a Tesseract používajú dvojchodové porovnávanie znakov, pričom druhý chod je označovaný ako adaptatívne rozoznavanie a používajú tvary písmen získaných z prvého prechodu, ktoré majú vysokú dôveryhodnosť na lepšie rozoznavanie zvyšných písmen pri druhom prechode. Toto rozoznavanie je výhodou najmä v prípadoch, keď je písmo neobvyklé alebo zdrojový obrázok je v nízkom rozlíšení.<sup>5</sup>

### 3.2.3 Tesseract

Použitým systémom pre optické rozoznavanie znakov v aplikácii, presnejšie na detekciu zakúpených produktov umožňujúcu rýchle pridanie produktov do aplikácie, je open-source projekt Tesseract, sponzorovaný spoločnosťou Google, a vydaný pod licenciou Apache License, verzia 2. Tesseract je v súčasnosti považovaný za jeden z najpresnejších systémov OCR, ktoré sú voľne dostupné. Dostupný je vo verzii 3.02. Obrovskou nevýhodou softvéru vo verziách menších než 2.0 bola absencia spracovania súborov iného formátu než TIFF<sup>6</sup>. Systém si taktiež nevedel poradiť so vstupným súborom, ktorý obsahoval text na viacerých riadkoch. Počiatočné verzie podporovali len anglický jazyk, no v súčasnosti sa táto podpora rozšírila o ďalších 37 jazykov, vrátane slovenčiny a češtiny. Nasledujúca časť zobrazuje originálny vstupný súbor - vyfotografovanú časť pokladničného bloku, vstupný súbor po aplikovaní pre-processingu a nakoniec výstupný text po rozoznaní systémom Tesseract.



15.0 NUDL.P KURECI VIMI 50G	6.10
15.0 CHERRY 250G	19.90
15.0 NUDL.P KURECI VIMI 50G	6.10
21.0 AO POKLADNI TASKA	1.90
15.0 AHB ORISOVA COKO 100G	9.90
15.0 POLICAN	25.00
219.00 x 0.114 KG	
15.0 SUNKOVA CIHLA	14.70
99.00 x 0.148 KG	
15.0 AHB ENERG.NAPOJ TROPICAL 0	6.80
15.0 AO TOUST. CHLEB SV. B. 500	16.90
15.0 ZLATY SYR GOUDA 48%	14.90
159.00 x 0.094 KG	
15.0 AHB NEKTAR POMERANC 50% 1L	13.90

Obrázek 3.6: Originálny vstupný súbor



15.0 NUDL.P KURECI VIMI 50G	6.10
15.0 CHERRY 250G	19.90
15.0 NUDL.P KURECI VIMI 50G	6.10
21.0 AO POKLADNI TASKA	1.90
15.0 AHB ORISOVA COKO 100G	9.90
15.0 POLICAN	25.00
219.00 x 0.114 KG	
15.0 SUNKOVA CIHLA	14.70
99.00 x 0.148 KG	
15.0 AHB ENERG.NAPOJ TROPICAL 0	6.80
15.0 AO TOUST. CHLEB SV. B. 500	16.90
15.0 ZLATY SYR GOUDA 48%	14.90
159.00 x 0.094 KG	
15.0 AHB NEKTAR POMERANC 50% 1L	13.90

Obrázek 3.7: Vstupný súbor po aplikovanej pre-processingu

<sup>5</sup><http://tesseract-ocr.googlecode.com/svn/trunk/doc/tesseracticdar2007.pdf>

<sup>6</sup>Tagged Image File Format - súborový formát pre ukladanie rastrovej počítačovej grafiky

## Výstupný text spracovaný systémom Tesseract

```
15.0 Noo1 P KURECI VIMI 500 6.10
15.0 CHERRY 2500 19.90
15.0 NUDL.P KURECI VIMI 500 0.10
21.0 Ao POKLADNI TASKA 1.90
15.0 AHB ORISKOVA COKO 1000 9.90
15.0 POLICAN 25.00
219.00 x 0.114 Ko
10.0 SUNKOVQ CIHLR 14.70
99.00 X 0 148 Ko
15.0 AHB ENERG.NAPOJ TROPICAL 0 6.60
15.0 Ao 1003?. CHLEB 3V. 0. 500 16.90
15.0 ZLATY SYR GOUDA 48 14.90
159.00 x 0.094 Ko
10.0 RHB NEKTAR POMERANE 00 1L 13.90
```

### 3.2.4 Pridávanie nových produktov pomocou OCR

Po stlačení tlačidla so znakom fotoaparátu na obrázku 2.2.4 sa inicializuje a spustí fotoaparát, pomocou ktorého užívateľ vyfotí pokladničný blok. Následne, ak je užívateľ s vytotografovaným blokom spokojný, potvrdí výber a aplikácia ho presmeruje na okno, v ktorom je potrebné vybrať z pokladničného bloku tú časť, ktorá obsahuje sumarizáciu zakúpených produktov. Výber sa vykonáva pomocou hľadáča obdĺžnikového tvaru, ktorého zvolená oblasť je ohraničená zeleným rámom, a zobrazuje sa svetlejšie, než oblasť, ktorá nebude spracovaná. Jednotlivé strany tohto obdĺžnikového útvaru je možné ľubovoľne zväčšovať alebo zmenšovať pomocou dotykov na obrazovke, pričom je potrebné dotknúť sa hrany obdĺžnika a následným pohybom do strán sa jeho obsah automaticky zväčší alebo zmenší na základe smeru pohybu. Po ukončení zmeny veľkosti výberu, sa obraz automaticky priblíži alebo oddiali, a to na základe zvolenej časti odфотоografovaného pokladničného bloku. Týmto spôsobom je možné presnejšie zvoliť si potrebnú časť z bloku.

Po potvrdení výberu oblasti pokladničného bloku, ktorá obsahuje zakúpené produkty, sa následne vybraná oblasť pôvodného obrázku oreže, a aplikujú sa naň grafické filtre, ktoré postupne upravujú obrázok do požadovanej podoby pre rozpoznávanie textu z obrázku. Po spracovaní zvoleného obrázku a získaní rozpoznávaného textu sa užívateľovi ponúkne dialógové okno, z ktorého má možnosť si vybrať obchod, z ktorého pochádza pokladničný blok. Tento výber však musí súhlasiť s pokladničným blokom, nakoľko každá z predajní má štruktúru pokladničných blokov rôznu a pri nesprávnom výbere by mohlo dôjsť k nekorektnému roztriedeniu zakúpených produktov. V súčasnosti je podporovaná predajňa Albert spoločnosti AHOLD Czech Republic, a.s. Po zvolení vhodnej predajne na roztriedenie položiek, systém automaticky na základe získaného textu z pokladničného bloku roztriedi jednotlivé položky a vytvorí schému produktov pripravených na vloženie do aplikácie. Takto vytvorené produkty sa však pred samotným vložením nachádzajú v zozname, v ktorom má užívateľ možnosť ich upraviť alebo v nevyhovujúcich prípadoch dokonca úplne z procesu pridávania odstrániť. Po potvrdení zoznamu sa následne každá z položiek vloží do aplikácie ako nový

produkt.

Nakoľko ide o ešte stále sa vyvíjajúci systém rozoznávania znakov a jeho výstup nemusí byť presný je pre získavanie presnejších názvov produktov pri spracovaní pokladničného bloku použité porovnávanie názvov produktov, a to medzi získaným názvom z pokladničného bloku a názvami už existujúcich produktov vložených v aplikácii. Toto porovnávanie je založené na Levenshteinovej vzdialenosti.

### 3.3 Databáza

Súčasťou takmer každej aplikácie je využívanie externého úložiska a spolu s ním spojených databáz k ukládaniu dát. Tieto množiny štruktúrovaných dát sú uložené v systéme a na komunikáciu s nimi je potrebný dopytovací jazyk. Typicky je v každej z databáz opis štruktúry vložených dát, ktorý sa nazýva logická schéma. Táto schéma obsahuje informácie o type objektov nachádzajúcich sa v databáze, ako aj vzťahmi medzi jednotlivými objektami. Android ako operačný systém ponúka tvorcom aplikácií využívanie databáz spolu s dopytovacím jazykom SQLite3. Avšak v prípade vzdialenej databázy, vytvorenej k zdieľaniu informácií o produktoch, je použitý MySQL.

SQLite je open source databáza, ktorá však podporuje štandardné funkcie relačných databáz ako SQL syntax a transakcie. K úspešnému fungovaniu databázy je potrebné minimum operačnej pamäti, čím sa stáva vhodným kandidátom najmä pre mobilné zariadenia. Medzi podporované dátové formáty SQLite databázy patrí **TEXT** reprezentujúci postupnosť znakov, **INTEGER** pre uchovávanie celých čísel ako aj **REAL** pre desatinné čísla. Všetky ostatné typy budú konvertované na jeden z vyššie popísaných.

Pri používaní SQLite databázy v operačnom systéme Android nie je potrebná prvotná inicializácia ani administrácia databázy. Avšak v dôsledku prístupu k údajom v databáze, ktorý môže byť pomalý, nakoľko sú informácie uchované ako súbory na disku zariadenia, je odporúčané pristúpiť k nim asynchrónne.

Navrhnutá databáza pre aplikáciu pozostáva zo štyroch tabuliek, ktoré uchovávajú jednotlivé informácie o produktoch a ich skupinách, obchodoch a čiarových kódoch. Súčasťou návrhu sú taktiež triggre, ktoré sa starajú najmä o konzistenciu dát.

### 3.4 REST API

Nakoľko API operačného systému Android nepodporuje priame získavanie údajov zo vzdialených databáz, je potrebné vytvoriť ho ručne. Pri výbere technológie, ktorá sprostredkuje komunikáciu medzi klientom a serverom bol kladený dôraz najmä na jednoduchú manipuláciu a tým čo najmenší počet zasielaných požiadaviek, a takisto zmenám odolné rozhranie. K tomuto úkonu je teda využívané REST API (z angl. Representational State Transfer), ktoré jednoducho umožňuje získavať, upravovať, vkladať alebo zmazať informácie zo vzdialeného servera pomocou HTTP<sup>7</sup> požiadavok. REST API využíva štyri HTTP metódy a každá z nich má iný účel.

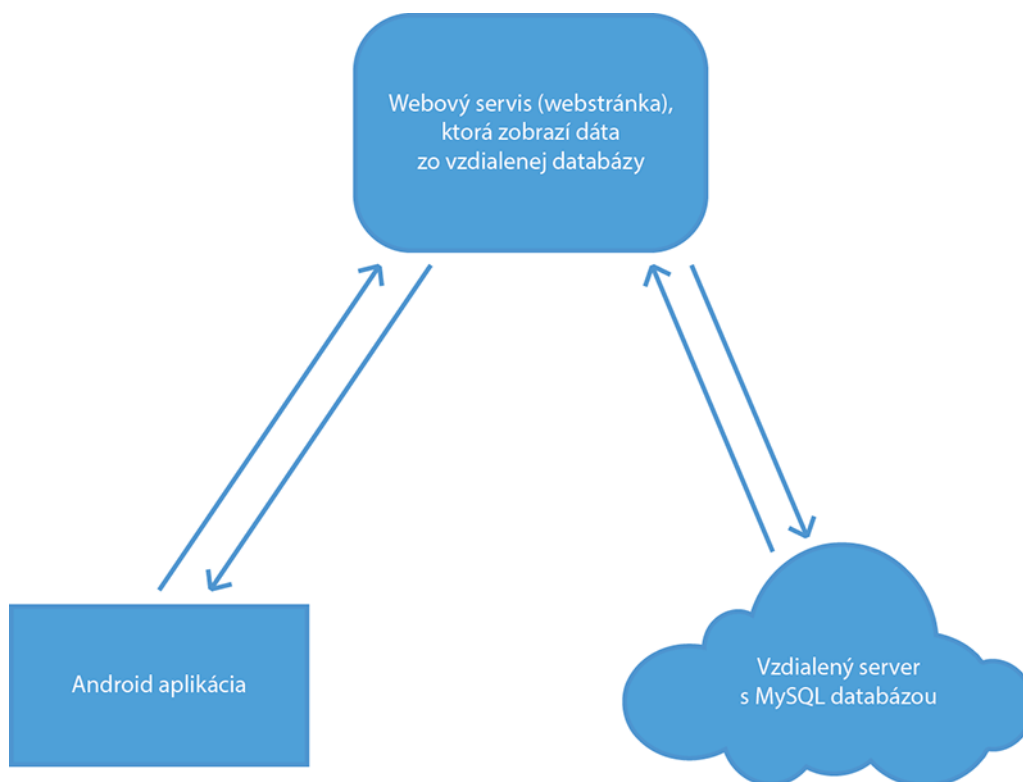
- **GET**
- **PUT**
- **POST**

---

<sup>7</sup>Hypertextový prenosový protokol definujúci požiadavky a odpovede medzi klientom a serverom

- **DELETE**

Pre dátovú výmenu medzi klientom a serverom REST využíva niekoľko štandardizovaných formátov, no aplikácia pracuje výhradne s formátom JSON a metódou POST. Princíp získavania dát medzi aplikáciou pracujúcou na zariadení a databázou na vzdialenom serveri je v odoslaní požiadavku zo strany klienta, teda aplikácie, na vzdialený server. Server túto požiadavku spracuje a v prípade odpovede zobrazí dáta na webovej stránke, ktoré následne zariadenie stiahne a spracuje. Pre implementáciu služby REST je použitý Slim framework, implementovaný v jazyku PHP, ktorý ponúka efektívne spracovanie dát a parsovanie jednotlivých požiadaviek. Nasledujúci obrázok zobrazuje grafický princíp komunikácie medzi zariadením a vzdialeným serverom.



Obrázek 3.8: Komunikácia medzi klientom a serverom použitím REST servisu

Ku korektnému fungovaniu komunikácie bolo potrebné nadefinovať jednotlivé požiadavky spolu s ich parametrami. Každá z nich obsahuje základnú adresu servera spolu s verziou API a príponou špecifikujúcou danú požiadavku.

Bázová adresa: <http://46.101.151.117/snappy/v1>

Úplný popis jednotlivých požiadaviek sa nachádza v prílohe, pričom budú pre jednoduchosť uvedené bez bázevej adresy.

## 3.5 Knižnice

Táto podkapitola je venovaná použitým knižniciam v aplikácii. Každá z knižníc je open source a vydaná pod licenciou Apache License v2. Nasledujúca časť obsahuje zoznam týchto knižníc spolu s URL odkazom, ktorý je možné pridať do zoznamu závislostí vo vývojovom prostredí Android Studio pri zostavovaní aplikácie.

- Butterknife - `com.jakewharton:butterknife:6.0.0`
- Robospice - `com.octo.android.robospice:robospice-retrofit:1.4.13`
- Float Action Button - `com.getbase:floatingactionbutton:1.9.0`
- Material Drawer - `com.mikepenz.materialdrawer:library:2.7.9@aar`
- Barcode Scanner - `me.dm7.barcodescanner:zbar:1.7`
- Material Edit Text - `com.rengwuxian.materialedittext:library:2.0.3`
- Android Chart - `com.github.PhilJay:MPAndroidChart:v2.0.9`
- Material Datetime Picker - `com.wdullaer:materialdatetimepicker:1.2.2`
- Material Progress Bar - `com.pnikosis:materialish-progress:1.5`
- Apache Commons - `org.apache.commons:commons-lang3:3.4`
- Android Support Library - `com.android.support:support-v4:22.0.0`
- Android AppCompat - `com.android.support:appcompat-v7:22.0.0`
- Tesseract OCR

## Kapitola 4

# Testovanie

Aplikácia bola zverejnená v Obchode Play avšak len vo forme beta verzie, nakoľko je ešte stále vo vývoji. Systém beta testovania spoločnosti Google zamärdzuje bežným užívateľom takéto aplikácie zobraziť v zozname vyhľadávaných položiek v Obchode Play, a je možné stiahnuť si ich len v prípade, že je užívateľ zaradený do skupiny beta-testerov, vytvorenej majiteľom aplikácie. Z tohto dôvodu táto kapitola popisuje výhradne subjektívne názory a spätné väzby desiatich ľudí, ktorí boli pridaný do skupiny testerov. Počas vývoja bola aplikácia testovaná na zariadeniach LG Nexus 5, One Plus One a Samsung S3 mini.

- **Užívateľské rozhranie**

Hlavným testovaným prvkom celej aplikácie bolo užívateľské rozhranie. To prešlo v priebehu vývoja niekoľkými rásnymi úpravami. Bol kladený dôraz najmä na jeho jednoduchosť a ľahkú orientáciu, čo sa v konečnom dôsledku podľa testerov podarilo splniť. V nasledujúcej časti je zobrazený priebeh vývoja časti užívateľského rozhrania spolu s krátkym popisom dôvodu, ktorý viedol k zmene. Naľavo je zobrazený pôvodný návrh a napravo jeho upravená verzia.

Prvá dvojica obrázkov zobrazuje zmenu v užívateľskom rozhraní v oblasti pridávania nového produktu. Pôvodny návrh sa javil byť chaotickým najmä v rozložení jednotlivých prvkov a kombinácii farieb. Výsledkom bolo preštylizovanie jednotlivých polí, presunutie tlačidiel do hornej časti obrazovky, a zmena farieb.

Nasledujúca dvojica zobrazuje detailný popis produktu, kde v pôvodnom návrhu boli užívatelia nespokojní najmä s nevýraznosťou grafu, s veľkým množstvom rozličných farieb a s použitím listu zobrazujúceho nákupy daného užívateľa. To vyústilo do pomerne rozsiahlej zmeny tohto zobrazenia produktu, pričom sa jednotlivé zoznamy zakúpených produktov, a to ako užívateľových tak aj iných užívateľov presunuli do grafu vývoja cien a rozlíšili sa farbami liniek. Do hornej časti sa presunuli štatistické údaje a naopak, do spodnej časti obrazovky sa pridala väčšia časť tlačidiel zabalených do jedného hlavného, ktoré svoj obsah po kliknutí expanduje.

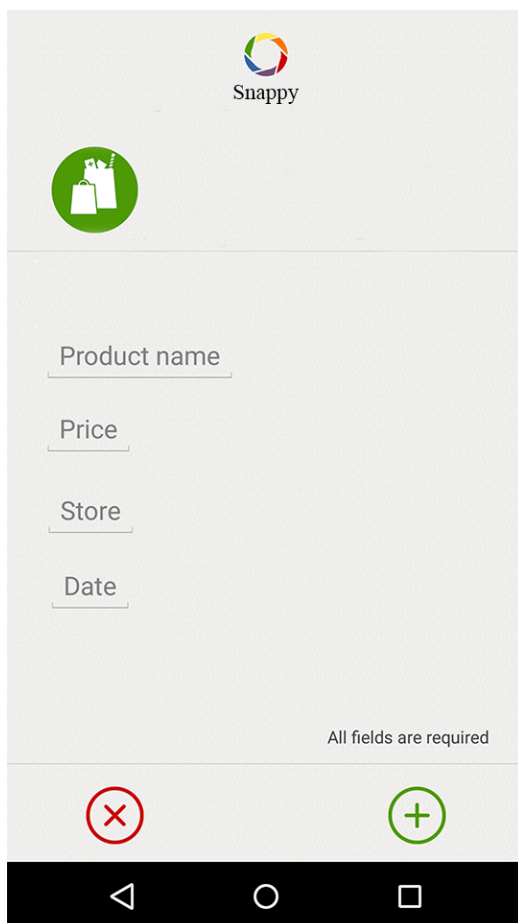
- **Stabilita**

Počas vyvíjania aplikácie ju zvolená skupina testerov testovala na stabilitu a funkčnosť v rôznych prípadoch a scenároch čo pomohlo odhaliť závažné chyby v niektorých prípadoch a to najmä pri priradzovaní čiarových kódov k produktom, a premenovávaním produktov na základe zmeny čiarových kódov. Chyby sa taktiež objavili pri zobrazovaní návrhov čiarových kódov pre produkty, ktoré obsahovali aj tie čiarové kódy, ktoré s daným produktom nemali nič spoločné. Všetke zistené problémy boli v poslednej verzii aplikácii odstránené a v súčasnosti sa javí aplikácia stabilne.

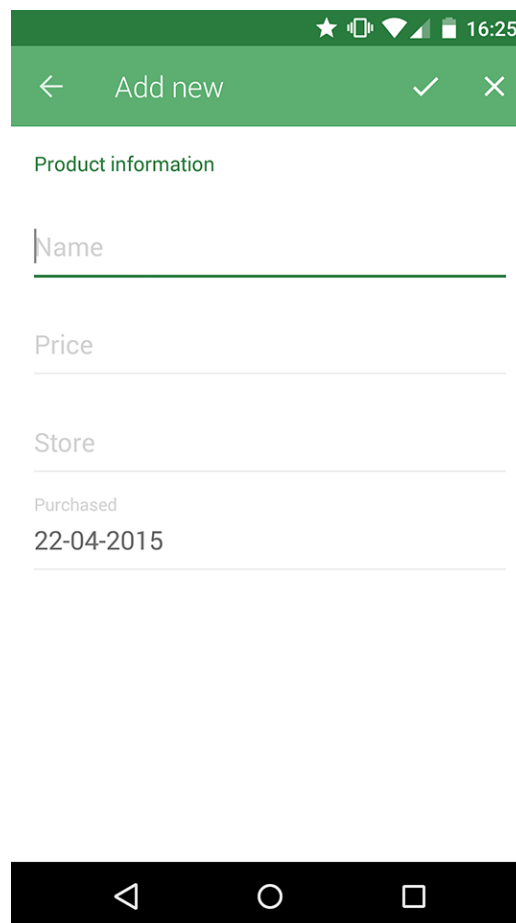


- **REST API a zobrazovanie nákupov iných užívateľov**

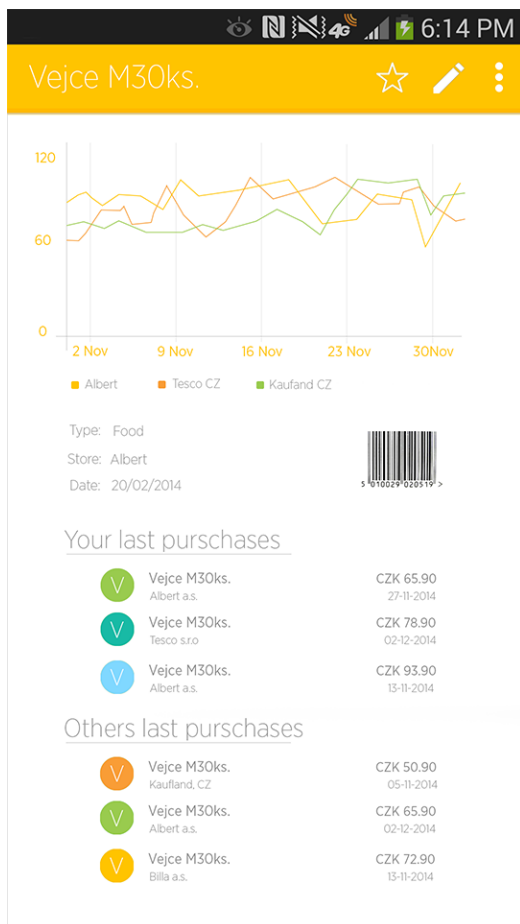
Cieľom týchto testov bolo vyskúšať korektné zobrazovanie nákupov iných užívateľom v prípade prihlásených ako aj anonymných užívateľov. Spolu s tým bolo potrebné otestovať aj vyčlenenie zobrazovania užívateľových vlastných nákupov zo vzdialeného servera v prípade, že bol prihlásený do aplikácie. Táto časť sa ukázala ako bezproblémová a nevyskytlo sa v nej žiadne nekorektné chovanie.



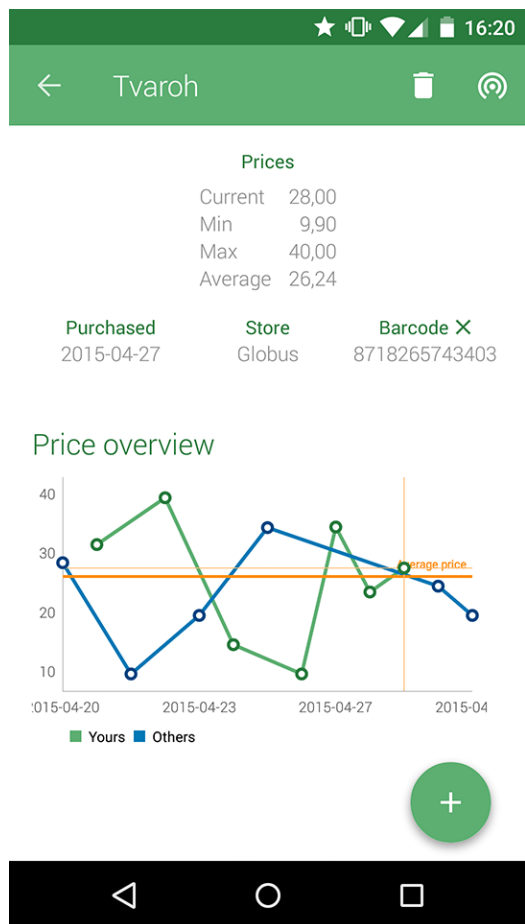
Obrázek 4.1: Pôvodný návrh umožňujúci pridávanie nových produktov



Obrázek 4.2: Upravený návrh umožňujúci pridávanie nových produktov



Obrázek 4.3: Původný návrh detailního popisu produktu



Obrázek 4.4: Upravený návrh detailního popisu produktu

# Kapitola 5

## Záver

Táto kapitola zhrňuje dosiahnuté výsledky a popisuje zoznam možných rozšírení implementácie v budúcnosti.

### 5.1 Dosiahnuté ciele

Cieľom tejto bakalárskej práce bolo vytvorenie reálnej aplikácie, čo úzko súvisí s nastudovaním si princípov programovania pre mobilnú platformu Android. Spolu s tým bolo taktiež potrebné preštudovať si a implementovať rozhranie služby REST na vzdialenom serveri, potrebné pre zdieľanie údajov o produktoch medzi užívateľmi. Výsledkom je aplikácia schopná uchovávať históriu vložených produktov, ako aj zobrazovať štatistické údaje a detailný popis jednotlivých produktov. V prípade prihlásenia užívateľa, ktoré však nie je povinné, je navyše poskytnutá možnosť zobrazovať ceny produktov od iných užívateľov a tým rozšíriť spektrum vývoja cien daného produktu. Aj napriek širokej škále možností, ktoré aplikácia ponúka, zostáva naďalej jednoduchou na ovládanie.

### 5.2 Rozšírenia

Aplikácia by v budúcnosti mohla byť rozšírená o nasledujúce prvky.

- **Rozšírenie škály podporovaných zariadení**  
Nakoľko bola táto aplikácia vyvíjaná a optimalizovaná pre mobilné zariadenie LG Nexus 5 a One Plus One, veľmi užitočným rozšírením by bola podpora ďalších zariadení.
- **Podpora ďalších mobilných platform**  
Implementácia aplikácie pre platformy Windows Mobile a iOS
- **Synchronizácia**  
Podpora synchronizácie so vzdialeným serverom
- **Možnosť prihlásenia prostredníctvom Facebook a Google účtu**  
V súčasnosti má takmer každý užívateľ mobilného zariadenia zriadený účet na sociálnej sieti Facebook alebo emailovú schránku v spoločnosti Google. Vďaka tomu by bolo možné poskytnúť užívateľom aplikácie komfortné a jednoduché prihlásenie, bez nutnosti registrácie.
- **Podpora push notifikácií**  
Pridanie podpory push notifikácií by mohlo byť obzvlášť prínosné v prípade, že si

uživateľ priradí produkt medzi obľúbené produkty a pri následnom vložení rovnakého produktu na vzdialený server iným užívateľom by bol o tejto skutočnosti informovaný práve pomocou týchto notifikácií.

- **Vytvorenie aplikačného servera**

Spolu s vytvorením push notifikácií je spojené aj vytvorenie aplikačného servera, ktorý by uľahčil prácu aplikácie a mohol jej poskytovať komplexnejšie informácie o jednotlivých produktoch a obsahu databázy.

- **Rozličné finančné meny**

Podpora rozličných finančných mien by umožnila využívanie aplikácie aj mimo Českej Republiky.

# Literatura

- [1] Johnson, S.: *Stephen Johnson on Digital Photography*. O'Reilly Media, 2006, ISBN 978-0596523701.
- [2] Meier, R.: *Professional Android 4 Application Development*. Wrox, třetí vydání, 2012, ISBN 978-1118102275.
- [3] Murphy, M. L.: *Android 2: Průvodce programováním mobilních aplikací*. Computer Press, 2011, ISBN 978-8025131947.
- [4] Schantz, H. F.: *The history of OCR, optical character recognition*. Recognition Technologies Users Association, 1982, ISBN 978-0943072012.

# Příloha A

## Obsah CD

Priložené CD obsahuje nasledujúcu štruktúru súborov.

- Snappy - Android Studio projekt vrátane použitých knižníc
- tex -  $\text{\LaTeX}$ súbory tejto práce
- rest - zdrojové súbory REST API pracujúcom na vzdialenom serveri
- poster - plagát
- video - video

# Příloha B

## REST API

- **/login**
  - *username, password*
  - Prihlásenie užívateľa
- **/username-exists**
  - *username*
  - Overí či užívateľ s takýmto daným menom existuje
- **/register**
  - *username, password*
  - Zaregistruje nového užívateľa a vráti jeho ID
- **/create-product**
  - *name, price, store, purchased, barcode, user\_id*
  - Vytvorí nový produkt v databáze a vráti jeho ID
- **/get-products**
  - *user\_id, name, barcode*
  - Vráti zoznam produktov s požadovaným názvom, alebo majú pridelený požadovaný čiarový kód a nepatria užívateľovi s daným ID
- **/delete-product**
  - *id, user\_id*
  - Zmaže produkt s požadovaným ID, no súčasne sa musí zhodovať aj zadané ID užívateľa
- **/update-product**
  - *id, name, price, store, purchased, barcode, user\_id*
  - Upraví produkt s daným ID