UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

FAKULTA PRÍRODNÝCH VIED A INFORMATIKY

MOBILNÁ APLIKÁCIA PRE DISTRIBUOVANÚ TVORBU NÁKUPNÝCH ZOZNAMOV

BAKALÁRSKA práca

**2024 MATEJ TUREK**

UNIVERZITA KONŠTANTÍNA FILOZOFA V NITRE

FAKULTA PRÍRODNÝCH VIED A INFORMATIKY

**MOBILNÁ APLIKÁCIA PRE DISTRIBUOVANÚ TVORBU NÁKUPNÝCH ZOZNAMOV**

BAKALÁRSKA práca

Študijný odbor: 18. Informatika

Študijný program: Aplikovaná informatika

Školiace pracovisko: Katedra informatiky

Školiteľ: RNDr. Ján Skalka, PhD

Nitra 2024 Matej Turek

Obrázok, na ktorom je text, snímka obrazovky, písmo, list

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, písmo, dizajn

Automaticky generovaný popis

**POĎAKOVANIE**

Na tomto mieste si dovoľujem poďakovať môjmu školiteľovi a vedúcemu Katedry informatiky RNDr. Jánovi Skalkovi, PhD. za pripomienky a odbornú pomoc pri vypracovaní práce. Vďaka jeho ochote pomôcť, cenným radám, ale aj času strávenému korigovaním práce som mu povďačný za skompletizovanie mojej bakalárskej práce.

**ABSTRAKT**

TUREK, Matej: Mobilná aplikácia pre distribuovanú tvorbu nákupných zoznamov. [Bakalárska práca]. Univerzita Konštantína Filozofa v Nitre. Fakulta prírodných vied a informatiky. Školiteľ: RNDr. Ján Skalka, PhD. Stupeň odbornej kvalifikácie: Bakalár odboru Aplikovaná informatika. Nitra: FPVaI, 2023. .... s. (uviesť počet strán ZP).

Cieľom práce je analyzovanie existujúcich aplikácii pre tvorbu nákupných zoznamov, následné ich porovnanie a na záver implementácia vlastnej aplikácie pre tvorbu nákupného zoznamu. Aplikácia bude implementovaná pre operačný systém Android a bude komunikovať so serverom pre získavanie a posielanie dát vo formáte JSON. Z funkcionálneho hľadiska aplikácia bude umožňovať prihlásenie a registráciu používateľom, tvorbu nákupných zoznamov a podzoznamov, zdieľanie zoznamov s ostatnými používateľmi, radenie a vyhľadávanie položiek v zozname.

--------

Abstrakt musí byť písaný ako **jeden odsek**. **Abstrakt by mal mať rozsah maximálne jednej strany.** Abstrakt musí byť pre čitateľa zrozumiteľný, aj bez prečítania textu celej práce. **Abstrakt nie je len úvod práce, obsahuje zmienku o cieľoch, postup riešenia, výsledky a prínos.** Je potrebné zachovať základné informácie a súlad abstraktu s textom práce. Treba byť čo najstručnejší, dodržiavať požiadavky na obsahovú stránku abstraktu, ale vyhnúť sa nepresnostiam. Nie je vhodné uvádzať informácie alebo tvrdenia, ktoré sa nevyskytujú v samotnom texte práce. V abstrakte sa používa tretia osoba jednotného alebo množného čísla. Vždy, keď je to možné, treba použiť slovesá v prítomnom čase a v činnom rode. Prispieva to k jasnému, stručnému a pôsobivému písaniu. Trpný rod možno použiť na indikatívne konštatovania a na informatívne konštatovania, ak treba súčasne zdôrazniť osobu, na ktorú sa činnosť zameriava. *Terminológia:* V abstrakte sa používajú plnovýznamové slová z textu, ktoré pomôžu aj pri automatickom vyhľadávaní v textoch. Treba sa vyhnúť neznámym termínom, skratkám, akronymom alebo symbolom alebo ich treba v odôvodených prípadoch definovať hneď, keď sa prvý raz vyskytnú v abstrakte. Používajú sa jednotky, symboly, terminológia ISO vždy, keď je to možné. Ak nie sú k dispozícii, používajú sa národné normy.

Kľúčové slová: Nákupný zoznam. Mobilná aplikácia. Android.

**ABSTRACT**

TUREK, Matej: Mobile application for distributed creation of shopping lists. [Bachelor Thesis]. Constantine the Philosopher University in Nitra. Faculty of Natural Sciences and Informatics. Supervisor: Title. RNDr. Ján Skalka, PhD. Degree of Qualification: Bachelor of Applied Informatics. Nitra: FNSaI, 2023. .... p.

The abstract must be written as a single paragraph. The abstract should be a maximum of one page. It must be comprehensible to the reader, even without reading the full text of the paper. The abstract is not just an introduction to the thesis, but it includes a mention of the objectives, the solution procedure, the results and the contribution. It is necessary to keep the background information and consistency of the abstract with the text of the thesis. It is necessary to be as concise as possible, adhering to the content requirements of the abstract but avoiding inaccuracies. It is not appropriate to include information or statements that do not appear in the text of the thesis itself. The abstract should be written in passive voice. The imperfect gender can be used for indicative statements and for informative statements when the person on whom the action is directed needs to be emphasised at the same time. Terminology: the abstract uses full-meaning words from the text, which will also help with automatic text searches. Unfamiliar terms, abbreviations, acronyms or symbols should be avoided or, where justified, defined as soon as they first appear in the abstract. Units, symbols, ISO terminology shall be used whenever possible. Where they are not available, national standards shall be used.

Keywords: Shopping list. Mobile application. Android.

**OBSAH**

[Úvod ................................................................................................................................10](#_Toc162902728)

[1 Analýza súčasného stavu 11](#_Toc162902729)

[1.1 Súčasný stav prístupu používateľov na internet 11](#_Toc162902730)

[1.2 mObilné operačné systémy 12](#_Toc162902731)

[1.3 Android 13](#_Toc162902732)

[1.4 Natívne aplikácie vs. mobilné webové aplikácie 14](#_Toc162902733)

[1.5 Nákupný zoznam 16](#_Toc162902734)

[1.5.1 Počet produktov na nákupných zoznamoch 17](#_Toc162902735)

[1.5.2 Dôvody písania nákupného zoznamu 18](#_Toc162902736)

[1.5.3 Užívateľské rozhranie 18](#_Toc162902737)

[1.5.4 Aplikácie na vytváranie nákupného zoznamu 19](#_Toc162902738)

[2 Ciele záverečnej práce 22](#_Toc162902739)

[3 Návrh a metodika 23](#_Toc162902740)

[3.1 Návrh softvéru 23](#_Toc162902741)

[3.1.1 Funkčné požiadavky 23](#_Toc162902742)

[3.1.2 Nefunkčné požiadavky 23](#_Toc162902743)

[3.2 Definovanie aktérov a prípadov použitia 23](#_Toc162902744)

[3.3 Architektúra 25](#_Toc162902745)

[3.4 Databázový model 26](#_Toc162902746)

[3.5 Návrh užívateľského rozhrania 27](#_Toc162902747)

[3.6 Použité technológie 29](#_Toc162902748)

[3.6.1 JAVA 29](#_Toc162902749)

[3.6.2 JSON 29](#_Toc162902750)

[3.6.3 PHP 30](#_Toc162902751)

[3.6.4 MySQL 30](#_Toc162902752)

[4 Výsledky 31](#_Toc162902753)

[4.1 Implementácia 31](#_Toc162902754)

[4.1.1 Tvorba endopintov 31](#_Toc162902755)

[4.1.2 Implementácia aplikácie 32](#_Toc162902756)

[4.2 Testovanie Aplikácie 44](#_Toc162902757)

[4.2.1 Testovacie prípady 44](#_Toc162902758)

[4.2.2 Testovanie aplikácie 45](#_Toc162902759)

[Záver 46](#_Toc162902760)

[Zoznam bibliografických odkazov 47](#_Toc162902761)

[Zoznam príloh 51](#_Toc162902762)

# Úvod

Existuje množstvo aplikácii v službe Google Play a Apple Store, ktoré zľahčujú nákupy v obchodoch. Ich spoločnou funkciou je vytvorenie nákupné zoznamu, avšak v čom sa líšia sú rozšírené funkcie ako napríklad zdieľanie s ostatnými členmi, kategorizácia položiek zoznamu a zaujímavým dizajnom.

V úvodnej časti sa budeme venovať operačným systémom pre mobilné zariadenia a ich porovnaním. Detailnejšie si popíšeme operačný systém Android, ktorému sa táto práca ďalej venuje. Zhodnotíme si aj typy mobilných aplikácii z hľadiska architektúry, aké majú výhody a aké nevýhody. Ďalej sa budeme skúmať existujúce riešenia aplikácii založené na podobnom princípe.

V nasledujúcej kapitole si vymedzíme ciele, ktorými sa budeme riadiť pri tvorbe aplikácie.

V tretej kapitole si navrhneme aplikáciu od funkčných a nefunkčných požiadaviek, zadefinujeme aktérov a prípady použitia. Navrhneme architektúru aplikácie a komunikácie s ostatnými časťami systému a navrhneme užívateľské rozhranie. Takisto si zhodnotíme použité technológie

Posledná časť aplikácie popisuje samotnú implementáciu aplikácie a API endpointov. Popíšeme si základné triedy a všetky aktivity aplikácie spolu s ich funkciami. Tiež si zadefinujeme vylepšenia, ktoré by bolo možné aplikovať. Dôležitou časťou tejto kapitoly je samotné testovanie, ktoré zahŕňa testovacie prípady a praktické testovanie aplikácie.

# 1 Analýza súčasného stavu

Heinrichs (2011) definovalo nakupovanie ako jednou z najčastejších úloh v našom každodennom živote a tvorba a správa zoznamov nákupov sú dôležitou súčasťou tohto procesu. Vzhľadom na nedávny rozvoj mobilných zariadení už nie sme obmedzení len na perá a papier, pretože máme k dispozícii množstvo digitálnych nástrojov a aplikácií.

Li (2022) tvrdí, že ľudia teraz môžu využívať aplikácie na svojich smartfónoch na prístup k rôznym internetovým službám, vrátane okamžitého posielania správ (napríklad WhatsApp, WeChat), online sociálnych médií (napríklad Twitter, Weibo), elektronického nákupu (napríklad Amazon, Taobao) a online platobných systémov (napríklad PayPal, Alipay). Tieto služby sa stali dôležitou súčasťou infraštruktúry moderného informačného spoločenstva, čím sa stali aplikácie na smartfónoch nevyhnutnosťou v každodennom živote. Podľa správy od spoločnosti Statista sa počet aplikácií dostupných v obchode Google Play, oficiálnej obchodnej platforme pre Android, exponenciálne zvýšil z 16 000 v decembri 2009 na 2 893 806 v júli 2021. Taký rozsiahly a dôležitý trh s aplikáciami prilákal vývojárov a poskytovateľov služieb k skúmaniu správania používateľov aplikácií, aby lepšie vyvíjali a dodávali mobilné aplikácie.

## 1.1 Súčasný stav prístupu používateľov na internet

V októbri 2023 bolo na celom svete 5,3 miliardy používateľov internetu, čo predstavuje 65,7 % svetovej populácie. Údaje zo štatistickej stránky Statcounter (2023) ukazujú rôzny pomer návštevníkov internetu zo stolných počítačov a mobilných zariadení v rokoch 2010 - 2023. Z grafu 1 vyplýva, že napriek zvyšovaniu sa počtu mobilných návštev na internet, tempo rastu sa v posledných rokoch spomalilo. Rovnako graf ilustruje, že prudký pokles internetových používateľov zo stolných počítačov do roku 2017 vystriedalo obdobie kulminácie po súčasnosť. V prípade používateľov internetu prostredníctvom tabletu nárast zaznamenaný v rokoch 2012 – 2015 vystriedal pomalý pokles a v súčasnosti počet používateľov je menší ako 2 %. I napriek týmto skutočnostiam možno konštatovať, že najväčší počet používateľov internetu predstavujú mobilné zariadenia. Konkrétne v mesiacoch január – október 2023 sa ich návštevnosť pohybovala v rozmedzí 58,5 - 50,7 %.

Obrázok, na ktorom je text, diagram, rad, vývoj

Automaticky generovaný popis

*Graf 1 Celosvetový počet používateľov Internetu (Chaffey, 2023)*

## 1.2 Mobilné operačné systémy

Mobilný operačný systém je systémový softvér, ktorý je špeciálne navrhnutý tak, aby fungoval na vreckových zariadeniach ako sú mobilné telefóny, PDA. Je to platforma na aplikačné programy bežiacich na mobilných zariadeniach. Každý operačný systém má svoju vlastnú architektúru.

Tabuľka 1 zobrazuje historické aj súčasné operačné systémy na mobilné zariadenia, kde najpoužívanejšími sú Android a iOS. Android má na trhu približne 70 % zastúpenie a napríklad v roku 2021 bol inštalovaný na 83,32 % mobilných zariadeniach ([Devarakonda](https://www.researchgate.net/profile/G-K-Devarakonda?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InByb2ZpbGUiLCJwYWdlIjoicHVibGljYXRpb24ifX0), 2015).

*Tabuľka 1 Historické a súčasné operačné systémy*

|  |  |
| --- | --- |
| **Historické platformy** | **Súčasné platformy** |
| Symbian OS | Android |
| Palm OS | iOS |
| Maemo OS | Windows Phone |
| Meego OS | Firefox OS |
|  | Black Berry OS |

Opierajúc sa o štatistické údaje aj empirická časť práce sa bude používať vhodný programovací jazyk a vývojové prostredie pre mobilné aplikácie, konkrétne pre Android aplikácie.

## 1.3 Android

Operačný systém Android je založený na jadre Linux a jeho vývoj prebieha pod spoločnosťou Google. Primárne je určený pre smartfóny a tablety. Vzhľadom na svoju otvorenú povahu sa stal najrýchlejšie rastúcim mobilným operačným systémom. Táto otvorenosť ho urobila obľúbeným medzi mnohými spotrebiteľmi a vývojármi. Softvéroví vývojári môžu ľahko upravovať a pridávať vylepšené funkcie, aby vyhoveli najnovším požiadavkám mobilnej technológie. Používatelia Androidu sťahujú z obchodu Google Play viac ako 1,5 miliardy aplikácií a hier každý mesiac. Vďaka svojmu výkonnému vývojovému prostrediu majú používatelia a vývojári možnosť vytvárať vlastné aplikácie pre širokú škálu zariadení.

V súčasnosti má Android 2,5 miliardy aktívnych používateľov vo vyše viac ako 190 krajinách. Služba Google Play za posledné desaťročie enormne vzrástla a v roku 2020 dosiahla príjmy 38,6 miliardy USD. V roku 2020 bolo v obchode k dispozícii viac ako 2,9 milióna aplikácií, ktoré boli stiahnuté 108 miliárd krát (Curry, 2023).

Z architektonického hľadiska je operačný systém Android rozdelený do štyroch vrstiev:

* jadro,
* vrstva knižníc,
* vrstva aplikačného frameworku,
* aplikačná vrstva (Obrázok 1).

Jadro Androidu je modifikovanou verziou jadra Linux 2.6, ktoré je pravidelne aktualizované s rôznymi verziami Androidu. Knižnice poskytujú podporu pre grafiku, multimediálne schopnosti a ukladanie dát. Android runtime, zakorenený vo vrstve knižníc, obsahuje virtuálny stroj Dalvik, ktorý poháňa aplikácie. Ako náhradu za Dalvik Android predstavil nový Android RunTime s kompiláciou vopred, čo zlepšuje jeho výkon. Všetky aplikácie využívajú aplikačné rozhranie aplikačného rámca na prístup k najnižšej úrovni architektúry.

Obrázok, na ktorom je text, snímka obrazovky, číslo, písmo

Automaticky generovaný popis

*Obrázok 1 Štruktúra operačného systému Android (Khan, 2015)*

## 1.4 Natívne aplikácie vs. mobilné webové aplikácie

Rakestraw (2013) zadefinoval dva hlavné typy mobilných aplikácii: natívne a mobilné webové. Natívne sú integrované priamo do operačného systému zariadenia a dokážu komunikovať s jeho hardvérom. Natívne aplikácie sú schopné využívať výhody lokálnych API s cieľom maximalizovať funkčnosť pri zachovaní efektívnosti. Mobilné webové aplikácie sú aplikácie, ktoré sa spúšťajú priamo z online rozhrania, ako je napríklad webová lokalita. Tieto aplikácie zvyčajne nedokážu manipulovať s hardvérom zariadenia a sú obmedzené na rozhrania API webovej aplikácie a nie na programové balíky nachádzajúce sa v telefóne. Mobilná webová lokalita je séria webových stránok vytvorených výlučne na účely zobrazenia vo webovom prehliadači mobilného zariadenia. Tieto stránky sú často vytvorené pomocou HTML, ale niektoré operačné systémy ako iOS alebo Android sú vybavené webkitom. Tieto webkity umožňujú vykresľovanie webových stránok, ktoré rozširujú funkčnosť ďaleko nad rámec typických mobilných webových aplikácií; umožňujú manipuláciu s hardvérom, škálovanie používateľského rozhrania a ďalšie.

Tabuľka 2 porovnáva natívne a mobilné webové aplikácie.

*Tabuľka 2. Porovnanie natívnych a mobilných webových aplikácii (Jobe, 2013)*

|  |  |  |
| --- | --- | --- |
|  | **Natívne aplikácie** | **Mobilné webové aplikácie** |
| Vytvárať vs. konzumovať obsah | Natívne aplikácie sú vhodnejšie pre tvorbu obsahu kvôli výkonnosti a prístupu k hardvéru. | Mobilné webové aplikácie sú menej vhodné pre tvorbu obsahu, ale rovnako vhodné pre konzumáciu obsahu. |
| Používateľská skúsenosť | Plynulá integrácia s natívnym operačným systémom | Obmedzená integrácia, vyžaduje externé rámce |
| Frekvencia aktualizácii | Aktualizácie sú formálne prostredníctvom obchodov s aplikáciami | Aktualizácie sú viac neformálne a ekvivalentné aktualizáciám webových stránok |
| Výkon | Maximálny výkon a prístup k hardvéru zariadenia | Výkon závisí od vykresľovania JavaScriptu a mobilných webových prehliadačov, obmedzený prístup k hardvéru zariadenia |
| Funkcionalita | Všetka funkcionalita v mobilnom operačnom systéme je dostupná | Väčšina funkcionality mobilného operačného systému je dostupná |
| Vývoj | Vyžaduje špecifický vývoj pre každý mobilný operačný systém | Otvorené webové jazyky a prehliadače umožňujú vývoj "Napíš raz, bež kdekoľvek" |
| Ziskovosť | Rámec na monetizáciu aplikácií je dostupný prostredníctvom obchodov s aplikáciami | Nie je jasná, jednotná stratégia pre monetizáciu |

Sumarizácia výhod a nevýhod natívnej a webovej aplikácie je uvedená v tabuľke 3.

*Tabuľka 3. Výhody a nevýhody natívnej a webovej aplikácie*

|  |  |  |
| --- | --- | --- |
| **Aplikácia** | **Výhody** | **Nevýhody** |
| Natívna | * lepší výkon a rýchlosť * ochrana a zabezpečenie vzhľadom na fakt,  že natívna aplikácia je schválená príslušnou platformou * možná aj práca offline * interaktívnosť * intuitívnosť. | * vyššie náklady na vývoj * vyžaduje si skúsených vývojárov * neodporúča sa pre jednoduché aplikácie. |
| Webová | * nákladovo efektívna * vyžaduje nízku údržbu * môže byť spustená v akomkoľvek formáte * nevyžaduje manuálnu aktualizáciu. | * slabší používateľský zážitok, * nutnosť internetového pripojenia * v porovnaní s natívnymi aplikáciami pomalosť a spomalené reakcie * obmedzená interakcia a menej intuitívne rozhranie * nie je možné nainštalovať cez App Store či Google Play. |

Hybridná webová aplikácia je aplikácia, ktorá nie je úplne mobilnou webovou aplikáciou ani natívnou aplikáciou. Je to aplikácia napísaná pomocou webových technológii (HTML5, JavaScriptovými API a CSS), ale beží vnútri kontajnera natívnej aplikácie od tretích strán. Kľúčovými charakteristikami hybridnej aplikácie sú, že sú vyvinuté pomocou štandardných webových jazykov, ale zvyčajne majú prístup k natívnym API a hardvéru zariadenia. Niektoré z dobre známych a používaných hybridných mobilných frameworkov sú PhoneGap, Appcelerator a Appspresso.

## Nákupný zoznam

Thomas a Garland (2004) tvrdia, že aj v súčasnosti ľudia využívajú tradičné spôsoby tvorby nákupných zoznamov (obrázok 2a) ako je ich písanie na papier alebo sa snažia o zapamätanie si nakupovaných položiek, čo je však dosť nespoľahlivé a vedie k strate času a peňazí.

Adaji (2018) však pripomenul, že je však pravdepodobné, že ľudia si so sebou vezmú mobilné telefóny všade, kam prídu, preto je k dispozícii veľké množstvo digitálnych nástrojov a aplikácií (obrázok 2b).

Obrázok, na ktorom je text, rukopis, kancelárske potreby, list

Automaticky generovaný popis

b)

*Obrázok 2. Príklad nákupného zoznamu a) na papier a b) v mobilnej aplikácii (Katuk, 2019)*

Nakupovanie potravín a životný štýl človeka boli zaujímavými pre mnohé výskumy s cieľom skúmania a odhalenia korelácií medzi vzormi ľudského správania a predajom. Niektoré štatistické štúdie sú založené na údajoch pre konkrétne špecifické regióny, kým iné diskutujú o koncepcii v širšom rozsahu. Tieto analýzy správania odhaľujú fakty, ktoré sú užitočnými informáciami k identifikácii procesne orientovaných požiadaviek a možné zlepšenia, ktoré môžu obohatiť používateľskú skúsenosť (Harsha, 2017).

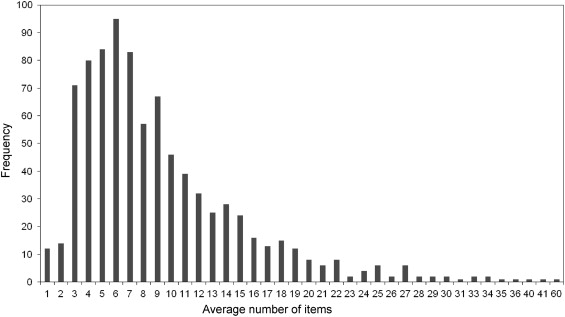
Aiolfi (2022) vo svojej práci uviedol, že za posledných pár rokov rozšírené mobilné pripojenie výrazne ovplyvnilo rozhodovací proces spotrebiteľov. Spotrebitelia používajú svoj mobil mimo predajne na zhromažďovanie informácií pred vstupom na miesto predaja. Naopak, spotrebitelia môžu použiť svoj mobilný obchod v obchode ako sprievodcu pri nakupovaní – napríklad na kontrolu svojich digitálnych nákupných zoznamov, porovnávanie cien online, nahliadanie do digitálnych letákov alebo používanie mobilných maloobchodných aplikácií.

Výskum PYMNTS (2019) uvádza, že spomedzi 2300 amerických spotrebiteľov 48 % vlastnia smartfóny, ktoré používajú pri nakupovaní v obchodoch alebo na činnosti súvisiace s nakupovaním. Napríklad 46,8 % ich používa na prístup k zľavám, 43,3 % na vyhľadávanie informácií o produktoch, 33,6 % ich používa na porovnanie cien s konkurenciou a 31,1 % na kontrolu recenzií produktov. Okrem toho nedávny prieskum spoločnosti Statista (2020) uviedol, že približne 46 % spotrebiteľov na celom svete sa cíti pohodlne pri používaní vlastného mobilného telefónu na činnosti v obchode; konkrétne 73 % respondentov prieskumu sa cíti bezpečne pri používaní svojho mobilného zariadenia na činnosti súvisiace s nakupovaním, ako je napríklad vyhľadávanie informácií o produktoch v obchode.

Nedávna štúdia Bellini a Aiolfiho (2017) zistila, že spotrebitelia, ktorí používajú smartfóny na prípravu nákupného zoznamu robia menej nákupov ako spotrebitelia, ktorí mobilné zariadenia nepoužívajú. Zistilo sa, že smartfóny uľahčujú spotrebiteľom nákupy potravín a udržiavajú ich na správnej ceste počas nákupu.

### 1.5.1 Počet produktov na nákupných zoznamoch

Podľa Schmidtovho výskumu (2012), výsledky ktorého sú na obrázku 2, bolo zistené frekvenčné rozdelenie počtu položiek na nákupných zoznamoch. Frekvenčné rozdelenie je „naklonené“ doprava a má tak rovnakú asymetrickú distribučnú charakteristiku ako tradičné rozdelenie príjmov. Väčšina spotrebiteľov si zapíše pomerne skromné množstvo produktov (medzi 4 až 15), zatiaľ čo malý počet spotrebiteľov si zapíše oveľa viac produktov. Horná hranica v tejto štúdii bola 60 položiek.



*Obrázok 4 Počet položiek nákupných zoznamov (Schmidt, 2012)*

### 1.5.2 Dôvody písania nákupného zoznamu

Hlavným dôvodom, prečo sú položky zapísané do nákupného zoznamu je, že sú z nejakého dôvodu potrebné. Najčastejšie spomínaný dôvod v rámci nákupov respondentov bol, že položky boli zapísané do nákupného zoznamu, pretože doma chýbajú/spotrebovali sa (78,6%). Pre približne tretinu nákupov boli položky zapísané z dôvodu, že boli potrebné na špeciálnu udalosť, ako je napríklad večierok.

Dôvody, prečo boli položky zapísané do nákupných zoznamov podľa Blockovho prieskumu (1999):

1. zapísal som veci, pretože boli spotrebované (78,6 %),
2. zapísal som veci, na ktoré som mal kupón (46,8 %),
3. zapísal som veci, ktoré som videl(a) v reklamných letákoch obchodu alebo v miestnych novinách (42,8 %),
4. zapísal som veci z dôvodu predvídaného očakávaného špeciálneho podujatia, ako je napríklad večierok alebo návšteva hostí (39,9 %).

### 1.5.3 Užívateľské rozhranie

Chen (2023) vo svojom výskume porovnával základné typy užívateľského rozhrania:

* svetlý a tmavý mód,
* zoznamový štýl a maticový štýl.

Výsledky odhalili, že ženy vykonávajú rýchlejšie vyhľadávanie informácií v svetlom móde, zatiaľ čo muži sú rýchlejší v tmavom móde (Obrázok 5). Rozloženie informácií ovplyvňuje vizuálnu vyhľadávaciu schopnosť a subjektívne hodnotenie používateľa; ženy uprednostňujú štýl zoznamu, zatiaľ čo muži uprednostňujú maticový štýl. Účastníci (ako muži, tak ženy) vnímali maticový štýl ako populárnejší v tmavom režime, avšak výsledok sa obrátil v svetlom režime.

Obrázok, na ktorom je mobilný telefón, text, snímka obrazovky, mobilné zariadenie

Automaticky generovaný popis

*Obrázok 5. Definovanie užívateľského rozhrania*

Zistenia z výskumu môžu slúžiť ako dobrá referenčná hodnota pre vývoj používateľského zážitku pri návrhu užívateľského rozhrania mobilných nákupných aplikácií.

### 1.5.4 Aplikácie na vytváranie nákupného zoznamu

V Tabuľke 4 Katuk (2019) porovnal mobilné aplikácie, ktoré pomáhajú spotrebiteľom pri ich nákupoch potravín.

*Tabuľka 4. Zoznam aplikácii podľa Katuka*

|  |  |
| --- | --- |
| Názov | Popis |
| Smart Grocer (Jain, 2018) | Smart Grocer poskytuje personalizovaný systém nákupného zoznamu tým, že zbiera históriu nákupov a rozpočet používateľov a udržiava informácie o zásobách potravín vrátane položiek, ktoré brzy vyprší, a položiek na predaj v preferovanom potravinovom obchode.  Systém môže automaticky generovať nákupný zoznam na základe receptu, ktorý používatelia vybrali po vytvorení týždenného plánu jedál. |
| MyNutriCart (López, 2017) | MyNutriCart je systém, ktorý si kládol za cieľ zlepšiť potraviny zakúpené v potravinových obchodoch s ohľadom na rozpočet používateľa a zároveň dosiahnuť zdravú stravu odporúčanú Smernicami pre Američanov v oblasti výživy. Systém môže generovať nákupný zoznam zdravých potravín, ktoré zodpovedajú zdravotnému stavu používateľov a členov rodiny. |
| The Smart Shopping List (DeSimone, 2017) | The Smart Shopping List uľahčuje vytváranie nákupného zoznamu poskytovaním základných funkcií pridávania, odoberania alebo odškrtnutia položiek z  nákupného zoznamu. Slúži tiež ako systém odporúčaní, ktorý odporúča používateľom potravinový obchod, ktorý by mohol mať najlepšie ponuky na základe zoznamu potravín uvedeného v zozname. Poskytuje tiež pripomienku o možných chýbajúcich alebo zabudnutých položkách. |
| AR-Assisted Mobile Grocery Shopping  (Ahn, 2015) | AR-Assisted Mobile Grocery Shopping app poskytuje používateľom podporu pri navigácii po uliciach potravín v obchode. Technológia rozšírenej reality je použitá na označenie potravín farbami na ulici, aby ukázala zdravé a nezdravé potraviny. Aplikácia skracuje čas  používateľov pri hľadaní zdravých a nezdravých potravín počas nákupov potravín. |

# 2 Ciele záverečnej práce

Cieľom tejto bakalárskej práce je vytvoriť aplikáciu na operačný systém Android, ktorá bude plniť funkciu nákupného zoznamu. Pre výmenu dát bude aplikácia komunikovať s Apache serverom, ktorý bude pristupovať k MySQL databáze.

Boli stanovené nasledovné podciele, v súlade s postupnosťou krokov k dosiahnutiu zadefinovaného hlavného cieľa práce:

* návrh konceptu aplikácie pre tvorbu nákupných zoznamov,
* definícia dátovej vrstvy,
* vytvorenie skriptov obluhujúcich endpointy pre prácu s údajmi,
* implementácia spojenia v mobilnej aplikácii s Apache serverom,
* návrh používateľského rozhrania mobilnej aplikácie,
* implementácia aplikácie,
* testovanie aplikácie a testovanie používateľského rozhrania ľudským používateľmi.

# 3 Návrh a metodika

## 3.1 Návrh softvéru

Cieľom aplikácie je vytvoriť systém, ktorý umožňuje každému používateľovi možnosť vytvoriť a spravovať vlastný účet. Po úspešnom prihlásení používateľa do aplikácie by mal užívateľ vidieť všetky nákupné zoznamy, ku ktorým má prístup. Buď tie, ktoré sám vytvoril alebo tie, ktoré mu boli nezdieľané inými užívateľmi. Tieto zoznamy môže ľubovoľne upravovať, mazať alebo pridávať nové. Súčasťou zoznamu sú položky zoznamu. Tie môžu byť priradené priamo do zoznamu alebo zoskupené do kategórii. Kategória je teda nejaká skupina položiek zoznamu, ktorú užívateľ môže vytvárať, mazať, pridávať, presúvať a odstraňovať z nej položky. Každý zoznam spolu s položkami a kategóriami môže používateľ zdieľať s inými používateľmi.

### 3.1.1 Funkčné požiadavky

Od aplikácie sú očakávanie nasledovné funkčné požiadavky:

* Aplikácia by mala umožniť používateľom registrovať sa.
* Aplikácia by mala umožniť používateľom prihlásiť sa.
* Aplikácia by mala umožniť vytvoriť nákupný zoznam.
* Aplikácia by mala vytvoriť kategórie a podkategórie v nákupnom zozname.
* Aplikácia by mala umožniť pridať položky do nákupného zoznamu.
* Aplikácia by mala umožniť editovať a odstraňovať položky nákupného zoznamu.
* Aplikácia by mala umožniť zdieľať nákupný zoznam iným používateľom.

### 3.1.2 Nefunkčné požiadavky

K očakávaným nefunkčným požiadavkám aplikácie patria:

* Aplikácia by mala fungovať na akomkoľvek zariadení s operačným systémom Android.
* Aplikácia by mala byť stále dostupná pre používateľov, aby mohli zaznamenávať alebo upravovať svoje nákupy.
* Aplikácia by mala mať silné zabezpečenie, aby bolo ťažké pre neoprávnené osoby získať prístup k osobným údajom a nákupným zoznamom používateľov.
* Aplikácia by mala mať schopnosť spracovať veľké množstvo nákupných zoznamov alebo veľkého počtu používateľov.

## 3.2 Definovanie aktérov a prípadov použitia

Vo všeobecnosti sú aktérmi vonkajšími entitami alebo systémami, ktoré interagujú so systémom alebo aplikáciou. V tomto konkrétnom prípade je jediným aktérom používateľ, ktorý má definované špecifikácie prípadov použitia v tabuľke 5.

*Tabuľka 5 Špecifikácia prípadov použitia pre používateľa*

|  |  |
| --- | --- |
| **Názov** | **Popis** |
| Registrovať sa | Ako používateľ chcem mať možnosť registrovať sa |
| Prihlásiť sa | Ako používateľ chcem mať možnosť prihlásiť sa |
| Vytvoriť nákupný zoznam | Ako používateľ chcem mať možnosť vytvoriť nákupný zoznam |
| Editovať nákupný zoznam | Ako používateľ chcem mať možnosť editovať nákupný zoznam |
| Zmazať nákupný zoznam | Ako používateľ chcem mať možnosť zmazať nákupný zoznam |
| Zdieľať nákupný zoznam | Ako používateľ chcem mať možnosť zdieľať nákupný zoznam |
| Vytvoriť kategóriu alebo podkategóriu v zozname | Ako používateľ chcem mať možnosť vytvoriť kategóriu alebo podkategóriu v nákupnom zozname |
| Zmazať kategóriu alebo podkategóriu v zozname | Ako používateľ chcem mať možnosť zmazať kategóriu alebo podkategóriu v nákupnom zozname |
| Editovať kategóriu alebo podkategóriu v zozname | Ako používateľ chcem mať možnosť editovať kategóriu alebo podkategóriu v nákupnom zozname |
| Pridať položky zoznamu | Ako používateľ chcem mať možnosť pridať položky zoznamu |
| Odstrániť položky zoznamu | Ako používateľ chcem mať možnosť odstrániť položky zoznamu |
| Editovať položky zoznamu | Ako používateľ chcem mať možnosť editovať položky zoznamu |

Na obrázku 6 je nakreslený diagram prípadov použitia.

Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popis

*Obrázok 6 Diagram prípadov použitia*

## 3.3 Architektúra

Návrh architektúry pozostáva z viacerých vrstiev, ktoré sú na obrázku 7. Medzi ne patria:

* Android aplikácia,
* Backendová vrstva spolu s endpointami (API) backendového rozhrania,
* Databáza.

Obrázok, na ktorom je snímka obrazovky, diagram, dizajn, písmo

Automaticky generovaný popis

*Obrázok 7 Architektúra riešenia*

Jedná sa o klasickú klient-server architektúru, v ktorej klient (aplikácia) pošle požiadavky na server, ktorý následne spracováva tieto požiadavky a poskytuje klientovi odpovede. Jedná sa o typický prístup pri vývoji webových aplikácií, kde klient môže byť webový prehliadač alebo Android aplikácia. Webový server je softvérový program, ktorý prijíma HTTP požiadavky od klientov a poskytuje odpovede. Server môže hosťovať webové stránky, API alebo iné služby. Pri vývoji Android aplikácií ako klienta je bežné, aby server poskytoval API, ktoré aplikácia využíva na získanie alebo odoslanie dát.

MySQL je relačný databázový systém, ktorý ukladá dáta. Databázový server obsahuje tabuľky, ktoré obsahujú záznamy (riadky) s dátami.

PHP skripty vytvárajú spojenie s MySQL serverom a vykonávajú SQL dotazy na čítanie, aktualizáciu alebo vytváranie údajov v databáze.

## 3.4 Databázový model

Na obrázku 8 bola navrhnutá databáza pomocou ER diagramu, kde základnými entitami je používateľ, zoznam, kategória a položky v zozname. Podstatou zdieľania listu medzi používateľmi predstavuje tabuľka pozvánok, ktorá umožňuje prístup ostatným používateľom (okrem vlastníka zoznamu) ku konkrétnemu zoznamu.

Obrázok, na ktorom je text, snímka obrazovky, diagram, číslo

Automaticky generovaný popis

*Obrázok 8 Návrh databázy*

## 3.5 Návrh užívateľského rozhrania

Pomocou online nástroja Figma bolo navrhnuté užívateľské rozhranie. Keďže v aplikácii bude viacero obrazoviek (tzv. aktivít), bolo nutné navrhnúť každú zvlášť.

**a) Prihlasovanie a registrácia**

Tieto aktivity, zobrazené na obrázku 9, definujú funkčné vlastnosti pre registráciu, prihlásenie a zabudnuté heslo.

Obrázok, na ktorom je text, snímka obrazovky, písmo, dizajn

Automaticky generovaný popis Obrázok, na ktorom je text, snímka obrazovky, písmo, číslo

Automaticky generovaný popis

*Obrázok 9 Prihlasovanie a registrácia používateľa*

**b) Profil**

Táto aktivita na obrázku 10 zobrazuje základné informácie o používateľovi s možnosťou editácie profilu a resetovanie hesla.

Obrázok, na ktorom je text, snímka obrazovky, písmo, dizajn

Automaticky generovaný popis

*Obrázok 10 Profil a editácia*

**c) Zoznamy a položky zoznamu**

Návrh pre zoznamy a položky zoznam definovaný na obrázku 11 zobrazuje zoznamy zoznamov a položky jednotlivých zoznamov. Je to najhlavnejšou časťou aplikácie.

Obrázok, na ktorom je text, snímka obrazovky, písmo, číslo

Automaticky generovaný popis Obrázok, na ktorom je text, snímka obrazovky, číslo, softvér

Automaticky generovaný popis

*Obrázok 11 Zoznamy*

## 3.6 Použité technológie

### 3.6.1 JAVA

Java je jedným z najpopulárnejších programovacích jazykov na vývoj Android aplikácií. Android používa vlastnú implementáciu Javy, známu ako Android Runtime (ART), pre vykonávanie Java kódu.

Java je všeobecný programovací jazyk, ktorý podporuje objektovo orientované programovanie (OOP). Z toho dôvodu môže byť využitý pre širokú škálu aplikácií a projektov.

Java má rozsiahlu komunitu vývojárov a množstvo dostupných knižníc, nástrojov a rámcov (frameworks). V prípade Androidu sú k dispozícii Android SDK (Software Development Kit) a mnoho ďalších nástrojov, ktoré uľahčujú vývoj.

Jedná sa o kompilovaný jazyk, ktorý je kompilovaný do medzičasového kódu (bytecode) a potom interpretovaný Java Virtual Machine (JVM) alebo v prípade Androidu do Android Runtime (ART). Tým sa dosahuje vysoký výkon a optimalizácia kódu.

Okrem vývoja mobilných aplikácií na platforme Android, môže byť Java použitá na vytváranie webových aplikácií, desktopových aplikácií, herných aplikácií a ďalších druhov softvéru.

### 3.6.2 JSON

JSON (JavaScript Object Notation) je jednoduchý textový formát určený na výmenu údajov. Je to ľahko čitateľný a ľahko zapisovateľný formát, ktorý je populárny pre použitie v dátových výmenách medzi programami. JSON je nezávislý od programovacieho jazyka, čo znamená, že ho môžete používať v rôznych jazykoch.

Formát JSON je založený na dvoch štruktúrach:

* Zoznam (Array): Usporiadaná kolekcia hodnôt, ktoré môžu byť čísla, reťazce, logické hodnoty, zoznamy, objekty alebo null.
* Objekt (Object): Neusporiadaná kolekcia párov kľúč-hodnota. Kľúče sú reťazce a hodnoty môžu byť čísla, reťazce, logické hodnoty, zoznamy, objekty alebo null.

JSON sa často používa pri výmene dát medzi webovými službami a klientmi, pre ukladanie konfiguračných súborov a pre rôzne dátové štruktúry v programovaní. Je ľahko spracovateľný v mnohých programovacích jazykoch, a preto je veľmi obľúbený vo svete vývoja softvéru.

### 3.6.3 PHP

PHP (Hypertext Preprocessor) je serverový skriptovací jazyk, ktorý je často používaný na vytváranie dynamickej webovej stránky a spracovávanie dát na strane servera. Je to otvorený, skriptovací jazyk so syntaxou podobnou C, Java a Perl, a je jedným z najobľúbenejších jazykov na webovom vývoji.

PHP podporuje HTTP metódy ako GET, POST, PUT, DELETE a ďalšie. Pri tvorbe API je často využívaný RESTful prístup, ktorý definuje jasné operácie pre rôzne zdroje.

Tvorba API (Application Programming Interface) v PHP môže byť dosiahnutá pomocou viacerých techník a nástrojov.

### 3.6.4 MySQL

Pri výbere typu databázy sa naskytla otázka, či vybrať SQL alebo NoSQL databázu.

Výhody relačných SQL databáz oproti nerelačným NoSQL databázam môžu byť v prípade tejto aplikácie nasledovné:

* Štruktúrované dáta - relačné databázy ako MySQL majú štruktúrovanú schému, ktorá umožňuje presne definovať tabuľky a vzťahy medzi nimi.
* Komplexné dotazy a transakcie - relačné databázy poskytujú robustnú podporu pre zložité SQL dotazy a transakcie, zatiaľ čo NoSQL databázy sa často viac zameriavajú na rýchlosť a škálovateľnosť
* Relačné vzťahy medzi dátami – ak sú v aplikácii rôzne typy dát (napr. produkty, kategórie, používatelia), a tieto dáta majú medzi sebou definované vzťahy, relačné databázy sú ideálne na zachovanie týchto vzťahov.

# 4 Výsledky

## 4.1 Implementácia

### 4.1.1 Tvorba endopintov

Pri implementácii endpointov sa vychádzalo z HTTP metód POST a GET. Metódy POST a GET sú dva hlavné spôsoby, ako prenášať dáta medzi klientom (aplikácia) a serverom v rámci protokolu HTTP (Hypertext Transfer Protocol). GET posiela dáta cez URL adresu, takže sú viditeľné a majú obmedzenú dĺžku. Používa sa na získanie údajov. Na druhej strane POST posiela dáta v tele požiadavky, čo znamená, že nie sú viditeľné v URL a nemajú obmedzenú dĺžku. To je užitočné pre odosielanie väčších objemov dát a citlivých informácií, a pre zmenu stavu na serveri. Tabuľka 6 uvádza zoznam endopointov.

*Tabuľka 6 Zoznam endpointov*

|  |  |  |  |
| --- | --- | --- | --- |
| **Názov** | **Metóda** | **Celok** | **Popis** |
| login | POST | účet | Prihlásenie používateľa na základe prihlasovacích údajov |
| register | POST | účet | Registrácia používateľa |
| editAccount | POST | účet | Editácia účtu |
| getAccountDetails | GET | účet | Získanie dát o účte |
| createCategory | POST | kategória | Vytvorenie novej kategórie v zozname |
| deleteCategory | POST | kategória | Vymazanie kategórie |
| getCategories | GET | kategória | Získanie všetky kategórie zoznamu |
| setCategory | POST | kategória | Úprava dát kategórie |
| acceptInvitation | POST | pozvánka | Prijatie pozvánky |
| createInvitation | POST | pozvánka | Vytvorenie pozvánky |
| deleteInvitation | POST | pozvánka | Vymazanie pozvánky |
| getInvitations | GET | pozvánka | Získanie pozvánok pre daný zoznam |
| getMyInvitations | GET | pozvánka | Získanie všetkých pozvánok pre daného používateľa |
| createItem | POST | položka | Vytvorenie položky zoznamu |
| getItems | GET | položka | Získanie položiek zoznamu |
| setItem | POST | položka | Úprava dát položky v zozname |
| setItemStatus | POST | položka | Nastavenie stavu pre položku |
| createList | POST | zoznam | Vytvorenie prázdneho zoznamu |
| deleteList | POST | zoznam | Zmazanie zoznamu |
| getListDetails | GET | zoznam | Získanie dát o zozname |
| getLists | GET | zoznam | Získanie zoznamov, ku ktorým môže používateľ pristupovať |
| setList | POST | zoznam | Editácia zoznamu |

### 4.1.2 Implementácia aplikácie

Medzi základné triedy patria:

Network Manager – trieda, slúži na správu sieťových požiadaviek. Popisuje funkcionalitu pre vykonávanie HTTP GET a POST požiadaviek na daný server. Pri vykonaní asynchrónnych požiadaviek používajú spoločného execútora (executor), ktorý vykonáva úlohy na samostatnom vlákne. Po dokončení požiadavky vracajú výsledky cez rozhranie ResultCallback. Metóda POST má vstupné parametre odkaz na danú URL adresu, POST dáta vo formáte JSON objektu, a callback.

public static void performPostRequest(String endpoint, JSONObject requestData, ResultCallback callback) {  
 *executor*.execute(() -> {  
 try {  
 URL url = new URL(*BASE\_URL* + endpoint);  
 HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();  
  
 urlConnection.setRequestMethod("POST");  
 urlConnection.setRequestProperty("Content-Type", "application/json");  
 urlConnection.setDoOutput(true);  
  
 OutputStream outputStream = urlConnection.getOutputStream();  
 outputStream.write(requestData.toString().getBytes());  
 outputStream.flush();  
 outputStream.close();  
  
 int responseCode = urlConnection.getResponseCode();  
 if (responseCode == HttpURLConnection.*HTTP\_OK*) {  
 BufferedReader reader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));  
 StringBuilder response = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 response.append(line);  
 }  
 reader.close();  
  
 *mainHandler*.post(() -> callback.onSuccess(response.toString()));  
 } else {  
 Log.*e*(*TAG*, "HTTP error code: " + responseCode);  
 *mainHandler*.post(() -> callback.onError("HTTP error code: " + responseCode));  
 }  
  
 } catch (IOException e) {  
 Log.*e*(*TAG*, "Error during POST request", e);  
 *mainHandler*.post(() -> callback.onError("Error during POST request"));  
 }  
 });  
}

Metóda GET je podobná, ale s tým rozdielom, že namiesto POST dát prijíma objekt mapy (dvojica kľúč – hodnota), ktoré pridáva do URL adresy, tak ako je definovaná GET metóda.

public static void performGetRequest(String endpoint, Map<String, String> queryParams, ResultCallback callback) {  
 *executor*.execute(() -> {  
 try {  
 String urlString = *buildUrl*(*BASE\_URL* + endpoint, queryParams);  
 URL url = new URL(urlString);  
 HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();  
 urlConnection.setRequestMethod("GET");  
 int responseCode = urlConnection.getResponseCode();  
 if (responseCode == HttpURLConnection.*HTTP\_OK*) {  
 BufferedReader reader = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()));  
 StringBuilder response = new StringBuilder();  
 String line;  
 while ((line = reader.readLine()) != null) {  
 response.append(line);  
 }  
 reader.close();  
 *mainHandler*.post(() -> callback.onSuccess(response.toString()));  
 } else {  
 Log.*e*(*TAG*, "HTTP error code: " + responseCode);  
 *mainHandler*.post(() -> callback.onError("HTTP error code: " + responseCode));  
 }  
 } catch (IOException e) {  
 Log.*e*(*TAG*, "Error during GET request", e);  
 *mainHandler*.post(() -> callback.onError("Error during GET request"));  
 }  
 });  
}

Shared Preferences Manager – trieda, ktorá získava a ukladá globálne dáta aplikácie. Objekt má súkromný konštruktor, čo znamená, že nemôže byť vytvorený inštanciou tejto triedy. To zabezpečuje, že objekt bude vždy použitý ako singleton. Poskytuje metódy na ukladanie a získavanie rôznych typov dát do/z SharedPreferences, ako sú userId, email a heslo. Tieto metódy sú statické, čo umožňuje ich volanie bez potreby vytvorenia inštancie triedy. Taktiež používajú editor SharedPreferences na zmenu hodnôt a ich aplikáciu. Poskytuje metódu clearData, ktorá vymaže všetky uložené údaje z SharedPreferences.

### 4.1.3 Aktivity Aplikácie

**MainActivity** - táto aktivita slúži ako hlavné rozhranie aplikácie, kde sa zobrazujú zoznamy a interaguje s nimi. Je zodpovedná za inicializáciu používateľského rozhrania vrátane zoznamu, tlačidiel a interaktívnych prvkov. Používateľ môže kliknúť na ikonu profilu na zobrazenie svojho profilu, tlačidlo pre vytvorenie nového zoznamu alebo tlačidlo pre zobrazenie ich pozvánok. Aktivita tiež spracováva interakcie s používateľom, ako je kliknutie na položku zoznamu pre zobrazenie detailov zoznamu, vytvorenie, úprava alebo odstránenie zoznamu. Okrem toho táto aktivita vykonáva HTTP volania na server na získanie údajov o zoznamoch a aktualizáciu ich stavu. To zahŕňa prihlásenie používateľa, získavanie zoznamov, ich odstránenie a pozývanie ďalších používateľov do zoznamov. Celkovo táto aktivita zabezpečuje správne fungovanie zobrazenia a interakcie s používateľom v aplikácii a zabezpečuje komunikáciu s backendom na získanie a aktualizáciu údajov (obrázok 12).

Obrázok, na ktorom je text, snímka obrazovky, číslo, softvér

Automaticky generovaný popis

*Obrázok 12 Ukážka MainActivity*

**LoginActivity** - táto aktivita slúži na spracovanie prihlasovania používateľov do aplikácie. V inicializačnej metóde onCreate() sa nastavuje používateľské rozhranie s poliami pre zadanie e-mailu a hesla a tlačidlami pre prihlásenie a registráciu. Po kliknutí na tlačidlo prihlásenia sa získajú údaje z polí a vykoná sa HTTP volanie na server pre overenie údajov. Úspešné prihlásenie vytvorí reláciu používateľa a jeho údaje sa uložia do lokálnej pamäti. V prípade chyby sa zobrazí chybová správa. Po úspešnom prihlásení sa aktivita uzavrie s výsledkom RESULT\_OK. Používateľ má možnosť kliknúť na tlačidlo pre registráciu, čo ho presmeruje na obrazovku pre registráciu. Ak používateľ stlačí tlačidlo späť, aktivita sa uzavrie a použije sa animácia pre prechod späť na predchádzajúcu obrazovku. Celkovo táto aktivita zabezpečuje proces prihlasovania a navigáciu v aplikácii (obrázok 13).

Obrázok, na ktorom je text, snímka obrazovky, dizajn

Automaticky generovaný popis

*Obrázok 13 Ukážka LoginActivity*

**RegisterActivity** - aktivita zabezpečuje registráciu nových používateľov do aplikácie. Po inicializácii používateľského rozhrania, kde sú zadávané údaje ako e-mail, meno a heslo, a následnom kliknutí na tlačidlo registrácie, sa overujú platnosť vstupných údajov. Kontroluje sa platnosť e-mailovej adresy, vstupného mena, dĺžka hesla a zhoda s opakovaným heslom. Ak sú údaje platné, vykonáva sa HTTP volanie na server na registráciu nového používateľa. V prípade úspešnej registrácie sa aktivita ukončí a používateľ je presmerovaný späť na prihlasovaciu obrazovku. V prípade chyby počas registrácie sa zobrazí chybová správa. Používateľ môže kedykoľvek stlačiť tlačidlo späť na zariadení a aktivita sa uzavrie s animáciou pre prechod späť na predchádzajúcu obrazovku.

Obrázok, na ktorom je text, snímka obrazovky, dizajn

Automaticky generovaný popis *Obrázok 14 Ukážka RegisterActivity*

**ProfileActivity** – úlohou tejto aktivity je spravovať profil používateľa v aplikácii. Po inicializácii používateľského rozhrania, kde sú zobrazované a upravované údaje ako e-mail a meno, a po kliknutí na tlačidlá pre uloženie zmien alebo odhlásenie, sa vykonávajú príslušné akcie. Metóda isValidInput() overuje platnosť zadaného mena pred jeho úpravou. Pri získaní platného mena sa vykonávajú úpravy profilu pomocou metódy editProfile(), ktorá odosiela požiadavku na server na úpravu údajov používateľa. V prípade úspešnej úpravy sa zobrazia nové hodnoty mena v profilovom rozhraní. Metóda getUserDetails() získava aktuálne údaje používateľa z servera a vyplňuje ich do príslušných polí v rozhraní.

Obrázok, na ktorom je text, snímka obrazovky, dizajn

Automaticky generovaný popis

*Obrázok 15 Ukážka ProfileActivity*

**ListActivity** - aktivita riadi zoznam položiek v aplikácii. Po inicializácii používateľského rozhrania, kde sú zobrazené položky zoznamu a možnosti pridania nových položiek, sa vykonávajú rôzne akcie, ako pridanie nových položiek, vytvorenie kategórií alebo úprava profilu. Používateľ môže pridať novú položku zadaním jej názvu a výberom príslušnej kategórie. Ak je vstup platný, položka je vytvorená a zobrazená v zozname. Rovnako je možné zobraziť ponuku možností pomocou tlačidla menu. Ak používateľ vyberie možnosť vytvoriť novú kategóriu, otvorí sa dialógové okno, v ktorom môže zadať názov novej kategórie alebo vybrať existujúcu. Aktivita tiež získava údaje o kategóriách a položkách z backendu. Po úspešnom získaní údajov sa vyplnia zoznamy kategórií a položiek. Celkovo táto aktivita umožňuje používateľovi spravovať zoznam položiek, pridávať nové položky a vytvárať kategórie, aby bol zoznam lepšie organizovaný. Pre správne zobrazovanie zoznamu bol implementovaný ListAdapter (obrázok 15).

Obrázok, na ktorom je text, snímka obrazovky, softvér, číslo

Automaticky generovaný popis *Obrázok 16 Ukážka ListActivity*

**ListAdapter** v aplikácii slúži na spracovanie a zobrazenie zoznamu kategórií a položiek. Tento adaptér poskytuje metódy na triedenie kategórií a položiek, generovanie zlúčených dát, zisťovanie typu zobrazenia položiek a kategórií, ako aj vytvorenie zobrazenia pre každú položku alebo kategóriu.

Funkcia triedenia triedi kategórie a položky podľa ich názvov. Funkcia generovania zlúčených dát spojí kategórie a položky do jedného zoznamu na zobrazenie. Metódy pre zistenie typu zobrazenia a vytvorenie zobrazenia pre každú položku alebo kategóriu sú implementované podľa ich typu. Tento adaptér tiež obsahuje metódy na spracovanie interakcií používateľa, ako je označenie položiek, vytvorenie, úprava a odstránenie položiek alebo kategórií.

Pri vytváraní, úprave alebo odstraňovaní položiek alebo kategórií adaptér spúšťa príslušné požiadavky na server a reaguje na odpoveď servera v závislosti na úspechu alebo neúspechu operácie. Taktiež obsahuje metódu na spustenie akcie na hlavnom vlákne pre zobrazenie správy alebo upozornenia používateľovi.

Aktivita **ListManagementActivity** (obrázok 17) je zodpovedná za správu zoznamov. Po inicializácii používateľského rozhrania umožňuje vytvárať nové zoznamy alebo upravovať existujúce. Používateľ môže zadať názov zoznamu a prípadné poznámky a následne ich odoslať. Ak sú vstupy platné, vytvorí sa nový zoznam alebo sa upravia informácie o existujúcom zozname. V prípade, že je aktivita spustená v editačnom režime, získa sa aj zoznam pozvánok pre daný zoznam a zobrazí sa na obrazovke. Aktivita tiež komunikuje s backendom a získava údaje o zozname a pozvánkach. Po úspešnom získaní údajov sa tieto údaje zobrazia v príslušných poliach. Celkovo táto aktivita umožňuje používateľovi spravovať svoje zoznamy a manipulovať s nimi prostredníctvom jednoduchého a intuitívneho používateľského rozhrania.

**Adapter InvitationAdapter** slúži na zobrazenie zoznamu pozvánok. Každá položka v zozname obsahuje informácie o pozvánke, ako je e-mailová adresa a stav pozvánky. Stav pozvánky je znázornený ikonou, ktorá zobrazuje, či bola pozvánka prijatá alebo čaká na potvrdenie. Pri dlhom stlačení na položku pozvánky sa zobrazí kontextové menu, ktoré obsahuje možnosť odstránenia pozvánky. Po kliknutí na túto možnosť sa zobrazí dialóg s potvrdením, ktorý umožňuje používateľovi potvrdiť alebo zrušiť odstránenie pozvánky. Ak je pozvánka úspešne odstránená, adapter aktualizuje zoznam pozvánok a upozorní na to používateľa pomocou rozhrania CallbackListener. Ak nastane chyba pri odstránení pozvánky, zobrazí sa oznámenie o chybe.

Obrázok, na ktorom je text, snímka obrazovky, softvér, operačný systém

Automaticky generovaný popis

*Obrázok 17 Ukážka ListManagementActivity*

**MyInvitationsActivity** (obrázok 18)je aktivita zodpovedná za zobrazenie pozvánok, ktoré používateľ prijal. Po inicializácii používateľského rozhrania získa aktuálne pozvánky používateľa a zobrazí ich v zozname. Pre získanie pozvánok sa komunikuje s backendom, kde sa na základe používateľovho ID získa zoznam pozvánok. Tieto pozvánky sa potom spracujú a zobrazia na obrazovke v zozname. Používateľ má možnosť prijať alebo odmietnuť pozvánky priamo z aktivity. Po vykonaní akcie na pozvánke sa zavolá metóda onInvitationAction(), ktorá obnoví zoznam pozvánok, aby zobrazila aktuálny stav. Celkovo táto aktivita poskytuje používateľovi prehľad o všetkých pozvánkach, ktoré prijal, a umožňuje mu prijímať alebo odmietnuť tieto pozvánky priamo z aplikácie. K tejto aktivite bol vytvorený MyInvitationsAdapter, ktorý slúži na zobrazenie zoznamu pozvánok. Každá položka v zozname obsahuje informácie o pozvánke, ako je e-mailová adresa a možnosti akceptovať alebo odstrániť pozvánku. Pri vytvorení zobrazenia pre každú položku adaptér overuje stav pozvánky a prispôsobuje zobrazenie podľa toho, či je pozvánka ešte neakceptovaná alebo už bola akceptovaná. V prípade neakceptovanej pozvánky umožňuje používateľovi akceptovať ju alebo odmietnuť. Akceptovanie alebo odmietnutie pozvánky spôsobuje odstránenie možnosti akceptovať a zobrazenie iba možnosti odstrániť. Pri odstránení pozvánky sa vykoná príslušná akcia na serveri a adaptér aktualizuje zoznam pozvánok podľa odpovede servera.

Obrázok, na ktorom je text, snímka obrazovky, softvér, operačný systém

Automaticky generovaný popis *Obrázok 18 Ukážka MyInvitationsActivity*

**Animácie** pri prechode medzi aktivitami aplikácie sú definované vo forme XML súborov, ktoré sú využité pri vytvorení alebo zrušení aktivity. Použitie je overridom metódy onBackPressed() a pri vytvorené novej aktivity:

Intent intent = new Intent(MainActivity.this, ProfileActivity.class);  
startActivity(intent);  
overridePendingTransition(R.anim.*slide\_in\_right*, R.anim.*slide\_out\_left*);

@Override  
public void onBackPressed() {  
 super.onBackPressed();  
 overridePendingTransition(R.anim.*slide\_in\_left*, R.anim.*slide\_out\_right*);  
}

### 4.1.4 Ostatné implementačné detaily

**Preklady:** Strings.xml v Android aplikáciách slúži na centralizované uchovávanie textových hodnôt a reťazcov, ktoré sa používajú v aplikácii. Jeho hlavnou úlohou je zjednodušiť údržbu, lokalizáciu a referencovateľnosť textových hodnôt v aplikácii. Centralizované uchovávanie textových hodnôt znamená, že všetky textové hodnoty sú uložené v súbore string.xml, čo umožňuje ich jednoduchú správu na jednom mieste. Taktiež umožňuje zoskupovanie textových reťazcov do jedného súboru, čo uľahčuje zmenu, aktualizáciu a údržbu aplikácie. Strings.xml tiež podporuje lokalizáciu aplikácie pre rôzne jazyky a regionálne nastavenia. Pre každý jazyk alebo regionálnu variantu sa môže vytvoriť samostatný súbor string.xml s príslušnými lokalizovanými textovými hodnotami. V kóde aplikácie sa na textové reťazce odkazuje pomocou identifikátorov definovaných v string.xml. Tento prístup zvyšuje čitateľnosť kódu a znižuje riziko chýb pri zmene textových hodnôt. Aktuálne je aplikácia preložená len do anglického jazyka, avšak táto implementácia umožňuje ľahké doplnenie ostatných jazykov.

**Ikonky:** Pre funkciu ikoniek boli využite preddefinované inkonky z ponuky vývojového prostredia Android Studio.

**Práva aplikácie:** Pre správne fungovanie aplikácie je vyžadovaný prístup na internet. V XML súbore „manifest.xml“ je to definované takto:

<uses-permission android:name="android.permission.INTERNET" />

**Error handling: V** prípade, že je ktorýkoľvek request neúspešný je zalogovaná daná chyba do logov aplikácie. Pokiaľ sa jedná o nesprávne údaje od používateľa napríklad pri prihlásení, vyskočí používateľovi zodpovedajúca správa. Pokiaľ sa však jedná o chybu v aplikácii napríklad kvôli prístupu na internet, vyskočí nová chybová aktivita „ErrorActivity“, ktorá naviguje na hlavnú aktivitu aplikácie (obrázok 19).

Ukážka princípu spracovávania requestov je na obrázku 20.

Obrázok, na ktorom je text, snímka obrazovky, štvorec, dizajn

Automaticky generovaný popis

*Obrázok 19 Ukážka ErrorActivity*

NetworkManager.*performGetRequest*(Endpoints.*GET\_LISTS*.getEndpoint(), queryParams, new NetworkManager.ResultCallback() {  
 @Override  
 public void onSuccess(String result) {  
 runOnUiThread(() -> {  
 try {  
   
 } catch (Exception e) {  
 Toast.*makeText*(MainActivity.this, "Get list error", Toast.*LENGTH\_LONG*).show();  
 Log.*e*("GET LIST REQUEST", "Error parsing JSON", e);  
 }  
 });  
 }  
  
 @Override  
 public void onError(String error) {  
 runOnUiThread(() -> {  
 Toast.*makeText*(MainActivity.this, "Get list error", Toast.*LENGTH\_LONG*).show();  
 Log.*e*("GET LIST REQUEST", error);  
 Intent errorIntent = new Intent(MainActivity.this, ErrorActivity.class);  
 startActivity(errorIntent);  
 finish();  
 });  
 }

*Obrázok 20 Ukážka princípu spracovávania requestov*

### 4.1.5 Možné vylepšenia

Boli navrhnutú tieto vylepšenia, ktoré by aplikáciu zlepšili:

* pridanie práv používateľom – každému používateľovi pri pozvánke uviesť k akým právam má prístup (napr. mazanie zoznamu, úprava zoznamu, čítanie zoznamu),
* zabezpečenie http metód – aktuálne riešenie HTTP metód nie je úplne zabezpečené, a ľahko sa dajú zneužiť zmenení dát pomocou POST metód a odchytenie dát pomocou GET metód
* notifikácie – implementácia zabudovaných Android notifikácii keď sa zmení status položky zoznamu, ku ktorému má užívateľ prístup,
* podpora viacerých jazykov,
* práca v offline režime – zahŕňalo by to vytvorenie lokálnej databázy aj na strane Android aplikácie (SQLite), ktorá by sa pravidelne synchronizovala s dátami na serveri.

## 4.2 Testovanie Aplikácie

Testovanie aplikácie je kritickým krokom v procese vývoja softvéru, ktorý má za cieľ zistiť a overiť, či aplikácia funguje správne a spĺňa požiadavky a očakávania používateľov. Testovanie aplikácie prebiehalo už pri samotnom vývoji aplikácii, kde sa odhalilo najväčšie množstvo chýb. Taktiež sa testovania zúčastnili osoby, ktoré nezasahovali do vývoja aplikácie.

### Testovacie prípady

*Tabuľa Testovacie prípady*

|  |  |
| --- | --- |
| Ako používateľ chcem mať možnosť registrovať sa | Správne |
| Ako používateľ chcem mať možnosť prihlásiť sa | Správne |
| Ako používateľ chcem mať možnosť vytvoriť nákupný zoznam | Správne |
| Ako používateľ chcem mať možnosť editovať nákupný zoznam | Správne |
| Ako používateľ chcem mať možnosť zmazať nákupný zoznam | Správne |
| Ako používateľ chcem mať možnosť zdieľať nákupný zoznam | Správne |
| Ako používateľ chcem mať možnosť vytvoriť kategóriu alebo podkategóriu v nákupnom zozname | Správne |
| Ako používateľ chcem mať možnosť zmazať kategóriu alebo podkategóriu v nákupnom zozname | Správne |
| Ako používateľ chcem mať možnosť editovať kategóriu alebo podkategóriu v nákupnom zozname | Správne |
| Ako používateľ chcem mať možnosť pridať položky zoznamu | Správne |
| Ako používateľ chcem mať možnosť odstrániť položky zoznamu | Správne |
| Ako používateľ chcem mať možnosť editovať položky zoznamu | Správne |

### Testovanie aplikácie

Pri testovaní používatelia prechádzali aplikáciu bežným scenárom, ktorý pozostával z krokov:

* vytvorenie účtu,
* prihlásenie,
* vytvorenie zoznamu,
* naplnenie zoznamu,
* zdieľanie zoznamu,
* simuláciu práce pri nákupe.

Počas toho boli nájdené chyby pri padaní aplikácie spôsobené pri nedostupnosti siete. Vyskytli sa chyby, pri nesprávnom zobrazovaní zoznamu a položiek zoznamu. Ďalšie chyby, ktoré používatelia našli, boli neaktualizované položky zoznamu pri práce s nimi.

# Záver

V analýze sme popísali súčasný aktuálny trend využívania mobilných aplikácii vo svete, ktoré vedie k implementácii stále novších a lepších aplikácii. Boli porovnané operačné systémy na mobilné zariadenia ako historické, tak aj súčasné, kde veľká časť venovala operačnému systému Android. Ďalej boli popísané typy mobilných aplikácii z architektonického hľadiska. Nakoniec bol v analýze popísaný a definovaný nákupný zoznam spolu s príkladmi.

V časti návrh a metodika boli zadefinované funkčné a nefunkčné požiadavky aplikácie spolu s aktérmi a prípadmi použitia. Bola navrhnutá architektúra, ktorá zahŕňala komunikáciu aplikácie so serverom a bol zadefinovaný databázový model. V poslednej kapitole sa venovalo návrhom užívateľského rozhrania.

V záverečnej kapitole nájdeme podrobný popis samotného vývoja aplikácie spolu s vytvorenými triedami a samotnými aktivitami aplikácie, ktoré obsahovali aj vizualizáciu užívateľského prostredia.

Dôležitá časť bola samotné testovanie aplikácie, kde sme testovali spolu s nezainteresovanými používateľmi testovacie prípady, z čoho výsledkom bolo viacero opráv aplikácie.

Výsledkom praktickej časti je funkčná aplikácia pre tvorbu nákupných zoznamov, avšak existuje viacero vylepšení, ktoré by bolo možné zapracovať a zlepšili by používateľský zážitok. V súčasnom stave však spĺňa požiadavky, ktoré sme si zadefinovali.

# Zoznam bibliografických odkazov

Kliknite alebo ťuknite sem a zadajte text.

==============================================================

ADAJI, Ifeoma, Kiemute OYIBO a Julita VASSILEVA, 2018. List It: A Shopping List App That Influences Healthy Shopping Habits. V: [online]. Dostupné na: doi:10.14236/ewic/HCI2018.81

AHN, Junho, James WILLIAMSON, Mike GARTRELL, Richard HAN, Qin LV a Shivakant MISHRA, 2015. Supporting Healthy Grocery Shopping via Mobile Augmented Reality. *ACM Transactions on Multimedia Computing, Communications, and Applications* [online]. 2015, roč. 12, č. 1s, s. 1–24. ISSN 1551-6857. Dostupné na: doi:10.1145/2808207

AIOLFI, Simone, Silvia BELLINI a Benedetta GRANDI, 2022. Using mobile while shopping in-store: a new model of impulse-buying behaviour. *Journal of Consumer Marketing* [online]. 2022, roč. 39, č. 5, s. 432–444. ISSN 0736-3761. Dostupné na: doi:10.1108/JCM-05-2020-3823

BELLINI, Silvia a Simone AIOLFI, 2017. The Impact of Mobile Device Use on Shopper Behaviour in Store: An Empirical Research on Grocery Retailing. *International Business Research* [online]. 2017, roč. 10, č. 4, s. 58. ISSN 1913-9012. Dostupné na: doi:10.5539/ibr.v10n4p58

BLOCK, Lauren G. a Vicki G. MORWITZ, 1999. Shopping Lists as an External Memory Aid for Grocery Shopping: Influences on List Writing and List Fulfillment. *Journal of Consumer Psychology* [online]. 1999, roč. 8, č. 4, s. 343–375. ISSN 1057-7408. Dostupné na: doi:10.1207/s15327663jcp0804\_01

BRAHLER STEFAN, 2010. *Analysis of the Android Architecture*.

CURRY DAVID, 2023. Android Statistics. *https://www.businessofapps.com/data/android-statistics/*.

DESIMONE DEANNA, 2017. *RELATIONSHIP OF GROCERY SHOPPING HABITS TO BODY MASS INDEX OF ADULTS IN WESTERN NEW YORK*. Buffalo, NY. Faculty of D’Youville College Division of Academic Affairs.

HALL MARK a HOSCH WILLIAM, 2023. Google. *https://www.britannica.com/topic/Google-Inc*.

HARSHA JAYAWILAL, W. A. a Saminda PREMERATNE, 2017. The smart shopping list: An effective mobile solution for grocery list-creation process. V: *2017 IEEE 13th Malaysia International Conference on Communications (MICC)* [online]. B.m.: IEEE, s. 124–129. ISBN 978-1-5386-3132-4. Dostupné na: doi:10.1109/MICC.2017.8311745

HEINRICHS, Felix, Daniel SCHREIBER a Johannes SCHÖNING, 2011. The hybrid shopping list. V: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* [online]. New York, NY, USA: ACM, s. 251–254. ISBN 9781450305419. Dostupné na: doi:10.1145/2037373.2037411

CHAFFEY DAVE, 2023. *Mobile marketing statistics compilation 2023* [online] [cit. 19. november 2023]. Dostupné na: https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/

CHEN, Chien-Hsiung a Weimin ZHAI, 2023. The Effects of Information Layout, Display Mode, and Gender Difference on the User Interface Design of Mobile Shopping Applications. *IEEE Access* [online]. 2023, roč. 11, s. 47024–47039. ISSN 2169-3536. Dostupné na: doi:10.1109/ACCESS.2023.3274575

JAIN ROSHNI, 2018. *SmartGrocer: A Context-Aware Personalized Grocery System*. University of Victoria. b.n.

JOBE, William, 2013. Native Apps Vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies (iJIM)* [online]. 2013, roč. 7, č. 4, s. 27. ISSN 1865-7923. Dostupné na: doi:10.3991/ijim.v7i4.3226

KATUK NORLIZA, JAYASANGAR TAMILARASI a YUSOF YUHANIS, 2019a. Design and Development of Smart List: A Mobile App for Creating and Managing Grocery Lists. *Baghdad Science Journal* [online]. 2019, roč. 16, č. 2(SI), s. 0462. ISSN 2411-7986. Dostupné na: doi:10.21123/bsj.2019.16.2(SI).0462

KHAN JAMIL a SHAHZAD SARA, 2015. Android Architecture and Related Security Risks. *Asian Journal of Technology & Management Research*. 2015, roč. 5, č. 2.

KRISHNA SUNDARA a DEVARAKONDA MOHAN, 2015. A Survey on Architectures of Mobile Operating Systems: Challenges and Issues. *International Journal of Research Studies in Computer Science and Engineering*. 2015, roč. 2, č. 3, s. 73–76.

LI, Li, Tegawendé F. BISSYANDÉ, Mike PAPADAKIS, Siegfried RASTHOFER, Alexandre BARTEL, Damien OCTEAU, Jacques KLEIN a Le TRAON, 2017. Static analysis of android apps: A systematic literature review. *Information and Software Technology* [online]. 2017, roč. 88, s. 67–95. ISSN 09505849. Dostupné na: doi:10.1016/j.infsof.2017.04.001

LI, T, T XIA, H WANG, Z TU, S TARKOMA, Z HAN a P HUI, 2022. Smartphone App Usage Analysis: Datasets, Methods, and Applications. *IEEE Communications Surveys & Tutorials* [online]. 2022, roč. 24, č. 2, s. 937–966. ISSN 1553-877X. Dostupné na: doi:10.1109/COMST.2022.3163176

LÓPEZ, Desiree, Michelle TORRES, Jammy VÉLEZ, Jhensen GRULLON, Edwin NEGRÓN, Cynthia M. PÉREZ a Cristina PALACIOS, 2017. Development and Evaluation of a Nutritional Smartphone Application for Making Smart and Healthy Choices in Grocery Shopping. *Healthcare Informatics Research* [online]. 2017, roč. 23, č. 1, s. 16. ISSN 2093-3681. Dostupné na: doi:10.4258/hir.2017.23.1.16

PYMNTS, 2019. *Consumers Who Use Mobile In-Store Are More Frequent Brick-And-Mortar Shoppers* [online] [cit. 13. november 2023]. Dostupné na: https://www.pymnts.com/news/mobile-payments/2019/in-store-smartphones/#:%E2%88%BC:text=In%20the%20study%2C%20nearly%20half,percent)%20look%20up%20product%20information

RAJINDER SINGH, 2014. An Overview of Android Operating System and Its Security Features. *Journal of Engineering Research and Applications* [online]. 2014, roč. 4, č. 2, s. 519–521 [cit. 4. november 2023]. Dostupné na: https://www.gadgetgyani.com/wp-content/uploads/2016/03/android-features-pdf.pdf

RAKESTRAW THOMAS L., EUNNI RANGAMOHAN V. a KASUGANTI RAMMOHAN R., 2013. The mobile apps industry. *Journal of Business Cases and Applications* [online]. 2013 [cit. 12. november 2023]. Dostupné na: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8b668d99bbb2b762e98578b1ab664a0d9b2cab22

SCHMIDT, Marcus, 2012. Retail shopping lists: Reassessment and new insights. *Journal of Retailing and Consumer Services* [online]. 2012, roč. 19, č. 1, s. 36–44. ISSN 09696989. Dostupné na: doi:10.1016/j.jretconser.2011.08.006

STATCOUNTER, 2023. *Desktop vs Mobile vs Tablet Market Share Worldwide* [online] [cit. 19. november 2023]. Dostupné na: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet

STATISTA, 2020. *Share of consumers worldwide comfortable with using their mobile phones at different points in the in-store customer journey as of October 2019* [online] [cit. 13. november 2023]. Dostupné na: www.statista.com/statistics/517424/global-in-store-mobile-shopping-activities

THOMAS, Art a Ron GARLAND, 2004. Grocery shopping: list and non‐list usage. *Marketing Intelligence & Planning* [online]. 2004, roč. 22, č. 6, s. 623–635. ISSN 0263-4503. Dostupné na: doi:10.1108/02634500410559015

# Zoznam príloh

Príloha A – Zdrojový kód + odkaz na GIT

Príloha B – Fotodokumentácia k prípravnej fáze záverečnej práce ........

PRÍLOHA A

**Príloha A**

* A1 Odkaz na GIT: https://github.com/matejturek/ShoppingList
* A2 ...
* A3 ...
* A4 ...