

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103004/B/2019/36103158107173892

TVORBA MOBILNEJ APLIKÁCIE PRE
NEVIDIACICH – ČÍTANIE

Bakalárska práca

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

TVORBA MOBILNEJ APLIKÁCIE
PRE NEVIDIACICH – ČÍTANIE
Bakalárska práca

Študijný program: Hospodárska informatika
Študijný odbor: Hospodárska informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Vedúci práce: Ing. Jaroslav Kultán, PhD.

Bratislava 2019

Samuel Sarik

Čestné vyhlásenie

Čestne vyhlasujem, že bakalársku prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum: 2.5.2019

.....

Vlastnoručný podpis študenta

Pod'akovanie

Týmto by som chcel poďakovať vedúcemu mojej bakalárskej práce,
Ing. Jaroslavovi Kultanovi, PhD., za rady, čas a usmernenia pri tvorbe bakalárskej práce.

ABSTRAKT

SARIK, Samuel: Tvorba mobilnej aplikácie pre nevidiacich – čítanie. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra aplikovanej informatiky FHI. – Vedúci bakalárskej práce: Ing. Jaroslav Kultán, PhD. – Bratislava: FHI EU, 2019, počet strán 70.

Cieľom bakalárskej práce je analyzovať existujúce pomôcky pre zrakovo postihnuté osoby, vytvoriť aplikáciu na načítanie textov rôznej veľkosti pomocou mobilného telefónu a ich ozvučenie. Súčasťou práce je manuál, ktorý opisuje prácu s danou aplikáciou. Práca je rozdelená do piatich kapitol. Obsahuje 52 obrázkov, 5 tabuliek a 4 prílohy. V prvej kapitole je popísaná základná problematika, prehľad súčasného stavu a existujúce pomôcky pre zrakovo postihnutých. Následne je uvedený cieľ práce s rozdelením do troch podcieľov. V ďalšej kapitole sú popísané nástroje a programové vybavenie, ktoré sa použijú na vývoj mobilnej aplikácie. Ďalšia kapitola je zameraná na samotný vývoj a implementáciu mobilnej aplikácie. Predposledná kapitola je zameraná na diskusiu a možnosti vylepšenia v budúcnosti. V záverečnej kapitole je zoskupená použitá literatúra a zdroje. Výsledkom bakalárskej práce je mobilná aplikácia, ktorá umožňuje nasnímať obraz, prekonvertovať ho na text a následne vrátiť hlasový výstup. Práca popisuje jednotlivé kroky a funkčnosť mobilnej aplikácie.

Kľúčové slová: mobilná aplikácia, Android, zrakovo postihnutí, ozvučenie textu, čítanie textu

ABSTRACT

SARIK, Samuel: Creating a mobile app for the blind – reading. – University of Economics in Bratislava. Faculty of Economic Informatics; Department of Applied Informatics FHI. – Bachelor thesis supervisor: Ing. Jaroslav Kultán, PhD. – Bratislava: FHI EU, 2019, 70 pages.

The main aim of the bachelor thesis is to analyze existing devices for visually impaired people, create an application to retrieve text of different sizes using a mobile phone and their sound. Part of the thesis is a manual describing the work with the given application. The thesis is divided into 5 chapters. It contains 52 pictures, 5 tables and 4 appendices. The first chapter describes basic issues, an overview of the current situation and existing aids for the visually impaired. Subsequently, the aim of the work is introduced and divided into three subsections. The next chapter describes the tools and software that will be used to develop mobile apps. The next chapter focuses on the development and implementation of the mobile application itself. The fifth chapter is the discussion and the possibilities for improvement in the future. The final chapter deals with the literature and resources used. The result of the bachelor thesis is a mobile application that allows to capture the image, convert it into text and then return the voice output. The work describes the steps and the functionality of the mobile application.

Key words: mobile application, Android, visually impaired, sound refresh, text reading

Obsah

Úvod	9
1 Analýza základných programových prostriedkov a pomôcok na uľahčenie čítania textu	11
1.1 Analýza vlastností	11
1.1.1 Zrkové postihnutie	11
1.1.2 Sluchové postihnutie	12
1.1.3 Telesné postihnutie	12
1.1.4 Kognitívne postihnutie	13
1.1.5 Zhrnutie analýzy vlastností	13
1.2 Funkcionalita zrakového orgánu	14
1.2.1 Problematika zrakového postihnutých ľudí	15
1.2.2 Typy zrakových postihnutí	16
1.2.3 Príčiny zrakových postihnutí	17
1.2.4 Typy zrakových porúch	18
1.3 Prehľad pomôcok pre nevidiacich	19
1.3.1 Historický vývoj	19
1.3.2 Členenie existujúcich pomôcok	20
1.3.3 Optické pomôcky	20
1.3.4 Vybrané prostriedky IT	22
1.4 Existujúce mobilné aplikácie	27
1.4.1 OCR aplikácie na pomoc pre nevidiacich	27
1.4.2 Čítacie aplikácie na pomoc pre nevidiacich	30
1.4.3 Aplikácie s hlasovými povelmi na pomoc pre nevidiacich	31
1.5 Zhrnutie analýzy	33
2 Cieľ a metodika práce	34
3 Popis metód a nástrojov tvorby informačných systémov	35
3.1 Výber pomôcok a nástrojov, ktoré sa použijú v aplikačnom zariadení	35
3.1.1 Výber mobilných zariadení	35
3.1.2 Výber programovacích jazykov	39
3.1.3 Výber existujúcich podsystémov	41
3.2 Využívanie OCR v mobilných aplikáciách	42
3.2.1 ABBYY Real-Time Recognition SDK	42
3.2.2 Tesseract OCR	43
3.2.3 IRIS Mobile OCR SDK	43
3.2.4 Mobile Vision API	44
4 Návrh aplikačného zariadenia a implementácia	45

4.1 Algoritmus činnosti zariadenia	45
4.2 Návrh informačného systému	52
4.3 Používateľské rozhranie	55
4.4 Analýza výsledkov práce	56
4.5 Tvorba manuálu.....	60
5 Diskusia a možnosti vylepšenia v budúcnosti	61
5.1 Možnosti vylepšenia aplikácie.....	61
5.2 Vylepšenie pridaním novej funkcionality	63
Záver	64
Zoznam použitej literatúry.....	66
Zoznam príloh	70

Úvod

V súčasnosti narastá počet ľudí so zhoršenými zrakovými vlastnosťami. Je to spôsobené viacerými faktormi, ako sú nedostatok vitamínov, zlá životospráva alebo môže ísť o vrodenú poruchu zraku. Problémy, ktoré nevidiacim ľuďom vznikajú môžu byť väčšieho a zložitejšieho rozsahu alebo môže ísť o menšie problémy, ako je návšteva správnych dverí u lekára. Hlavným obmedzením pre ľudí so zrakovým postihnutím sú malé písmena na etiketách, plagátoch, vizitkách a tiež neprimerané farebné kontrasty a rozlíšenia obrazoviek. Vďaka tomu, že svet sa stáva čoraz viac uponáhľanejším ako kedykoľvek predtým, osobná starostlivosť a pomoc pre týchto ľudí sa hľadá veľmi ťažko.

Rozvoj informačných a komunikačných technológií prispieva k tomu, aby sa bariéry, ktoré vznikajú nevidiacim a slabozrakým ľuďom čo najviac odstraňovali. V dnešnej dobe je už samozrejmosťou, že väčšina ľudí vlastní smartfón, telefón s operačným systémom, ktorý obsahuje bohaté programové vybavenie. Vývojári a softvéroví inžinieri vytvárajú aplikácie, ktoré napomáhajú ľuďom v ich každodennom živote. Umožňujú načítať text a vrátiť používateľovi hlasovú odozvu. Trendom v poslednom období sa stáva GPS navigátor pre nevidiacich, skener QR kódov a skener čiarových kódov.

Pomocou mobilných aplikácií vedia ľudia prečítať ťažko čitateľnú vizitku na dverách lekára, zloženie výrobku, a rôzne ďalšie texty. Tieto aplikácie predstavujú pre ľudí s poruchou zraku pomocnú ruku, ktorá je im stále k dispozícii. Napríklad, ak človek nevidí názvy jedál v jedálnom lístku v reštaurácii, použije mobilnú aplikáciu a tá mu tento názov prečíta.

Aplikácie môžu byť vyvíjané na mobilných telefónoch s operačným systémom iOS, Android alebo Windows Phone. V mojej bakalárskej práci som sa zamerlal na vývoj mobilnej aplikácie pre nevidiacich a slabozrakých na čítanie práve na mobilné telefóny s operačným systémom Android. Podľa najnovšieho výskumu, ktorý bol zverejnený na webovej stránke Gsmarena som zistil, že počet ľudí s Androidom je o polovicu vyšší ako s konkurenčným iOS-om (tabuľka sa nachádza v prílohe). Dôležitú úlohu pri výbere operačného systému zohral fakt, že viac ľudí trpiacich poruchou zraku je v staršom veku, a tí uprednostňujú Android pred iOS-om.

Mojou motiváciou pre výber bakalárskej práce bola možnosť vytvorenia produktu, čiže vytvoriť určitú pridanú hodnotu, ale najmä skutočnosť, že v reálnom svete môže aplikácia pomôcť stovkám ľudí, ktorí sú buď nevidiaci, alebo slabozrakí. Pomôcť v zmysle

bežných vecí, ako je nakupovanie, návšteva lekára, čítanie novín a podobne, ktoré sú pre vidiacich ľudí bežné.

Nástroje a prostriedky, ktoré som použil na vývoj aplikácie som opísal v samostatnej kapitole, kde som sa snažil zachytiť ich podstatu a možnosti použitia. Najdôležitejším prvkom mobilného telefónu je kamera, ktorá zachytáva text z fotografie. Následne dochádza ku konverzii textu na zvukovú správu pomocou mikrofónu. Pri návrhu aplikácie som sa zameral na krátky popis, načrtnol som algoritmus, ktorý som použil, opísal som vhodné nástroje na vývoj mobilnej aplikácie a nakoniec som sa venoval testovaniu funkčnosti na reálnom zariadení.

1 Analýza základných programových prostriedkov a pomôcok na uľahčenie čítania textu

Bakalárska práca je zameraná na vytvorenie mobilnej aplikácie, pričom tento proces tvorby je možné rozdeliť do dvoch častí. Prvá je zameraná na načítanie textu pomocou fotoaparátu, druhá sa zameriava na poskytnutie hlasového výstupu pomocou mikrofónu zabudovaného v mobilnom zariadení. Pred vytvorením aplikácie je dôležitým krokom zmapovanie cieľovej skupiny ľudí, ktorým bude táto aplikácia určená. Následne je potrebné analyzovať aktuálny stav trhu. Či už sa jedná o kompenzačné pomôcky elektronické, optické, mechanické alebo moderné aplikácie, ktoré napomáhajú nevidiacim a slabozrakým ľuďom. Jadro celej kapitoly sa zameriava na vysvetlenie príčin zhoršenia zraku, rozdelenie ľudí s poruchou zraku a popis možností na pomoc ľuďom so zrakovým postihnutím.

1.1 Analýza vlastností

Analýza zahŕňa oblasti, ktoré chcem prostredníctvom mojej aplikácie zlepšiť a aj tie, ktoré chcem odstrániť. Je rozčlenená do viacerých častí z hľadiska typu postihnutia či už zrakového, sluchového, telesného alebo kognitívneho.

1.1.1 Zrakové postihnutie

Ľudia so zrakovým postihnutím, ktorí sú nevidiaci, farboslepí, majú nízku úroveň videnia alebo trpia úplnou slepotou, nemôžu vidieť obrazovku a ovládacie prvky v mobilnej aplikácii tak, ako to môžu vidiaci ľudia. Z dôvodu vizuálnych obmedzení sa väčšina ľudí so zrakovým postihnutím spolieha na čítačku obrazovky, na obsluhu počítačov a mobilných zariadení. Čítačka obrazovky je nápomocný softvér, ktorý slúži na konverziu informácií zobrazovaných na obrazovke zariadenia do zvukovej podoby. Niektoré čítačky obrazoviek majú v sebe zabudovanú ovládateľnosť funkcií dotykom a gestami, a tiež možnosť počuť popisy jednotlivých funkcií. Medzi softvérovú podporu pre ľudí so zrakovým postihnutím patrí:

- **čítačka obrazovky** – niektoré mobilné zariadenia majú v sebe zabudovanú čítačku obrazovky, ale tá je dostupná aj na trhoch s aplikáciami tretích strán (Google Play, App Store).
- **nastaviteľná veľkosť písma** – v súčasnosti je už vo väčšine zariadení zabudovaná funkcia na zväčšovanie písma mobilných aplikácií. Ak sa táto funkcia aplikuje, potom sa znižuje množstvo údajov zobrazených na obrazovke.

- **zväčšovanie obrazovky** – patrí medzi vstavané funkcie mobilných zariadení, pričom pomáha slabozrakým ľuďom prispôbiť si obrazovku podľa svojich predstáv. Je vhodné vtedy, ak postihnutie nie je veľké a zväčšenie nemusí byť niekoľkonásobné. Pri veľkom postihnutí sa znižuje informačná cennosť textu.

- **nastaviteľný jas a kontrast** – táto funkcia sa používa na zmenu farby popredia, zmenu pozadia obrazovky a tiež zmenu jasu individuálnych potrieb.

- **rozpoznávanie hlasu** – umožňuje používateľom ovládať mobilné zariadenia prostredníctvom hlasových povelov, čo uľahčuje prácu nielen nevidiacim a slabozrakým ľuďom, ale aj vidiacim ľuďom. Využíva sa napríklad pri šoférovaní, ak šofér chce zapnúť GPS navigáciu prostredníctvom hlasu.

1.1.2 Sluchové postihnutie

Osoby so sluchovým postihnutím nedokážu počuť volajúceho pri telefonovaní, zvukové upozornenia, audio obsah a pokyny, ktoré sú vo zvukovom formáte. Väčšina z týchto osôb sa preto sústreďuje na znakovú reč a uprednostňuje komunikáciu prostredníctvom odosielania správ. Poskytnutím dostatočného popisu znakovej reči pre zvukový obsah, vizuálna a vibračná výstraha umožňuje ľuďom so sluchovým postihnutím používať mobilné aplikácie. K softvérovej podpore pre ľudí so sluchovým postihnutím patria:

- **vibračné signály a vizuálne upozornenia** – pomocou tejto funkcie pri prijímaní správy alebo hovoru, mobilné zariadenie začne blikať, vibrovať alebo sa zobrazí na obrazovke upozornenie

- **titulky** – použitím titulkov pre video môžeme zobraziť obsah zvukovej konverzácie a popísať jednotlivé signály a zvukové správy ku ktorým došlo v jednotlivých video scénach

- **ovládanie hlasu** – táto funkcia je užitočná najmä pre tých ľudí, ktorí potrebujú naslúchadlo

- **mono audio** – pomáha ľuďom so stratou sluchu postihujúcou len jedno ucho, pričom využíva kombináciu oboch zvukových kanálov do jedného a podáva samotný obsah cez jeden zvukový kanál

1.1.3 Telesné postihnutie

Ľudia s problémami horných končatín a mobilitou rúk majú ťažkosti s vykonávaním gest, stláčaním tlačidiel a ovládaním dotykových obrazoviek. Poskytovanie pokročilého

softvéru rozpoznávania reči umožňuje nahrádzať ručné ovládanie mobilných zariadení. Niektorí ľudia s poruchou pohyblivosti horných končatín a ruky môžu používať iné časti tela na ovládanie mobilných zariadení pomocou špeciálneho pera. Softvérovú podporu pre pomoc osobám s telesným postihnutím predstavuje:

- **rozpoznávanie hlasu** – používatelia môžu ovládať mobilné zariadenia prostredníctvom zadávania hlasových príkazov
- **rýchlosť stláčania tlačidiel** – slúži na nastavovanie požadovanej rýchlosti prechádzania stránkami obrazovky, a tiež na možnosť zjednodušiť navigáciu spájaním náhodného dotyku s cieľným dotykcom

1.1.4 Kognitívne postihnutie

Ľudia s kognitívnou poruchou majú problémy súvisiace s pozornosťou, pamäťou, komunikáciou, a tiež s analytickými, výpočtovými a čitateľskými zručnosťami. Prejavujú sa u nich ťažkosti s pochopením komplexného textu alebo problémy s prácou podľa manuálu či návodu. Osoby, ktoré majú problémy s učením, môžu používať čítačky obrazoviek, aby lepšie pochopili to, čo sa snažia naučiť. Softvérovú podporu predstavuje najmä:

- **intuitívne používateľské rozhranie** – poskytnutím jednoduchého, jasného používateľského rozhrania a konzistentných prvkov rozhrania sa uľahčuje navigácia pri používaní mobilných aplikácií a tiež sa ponúka pomoc aj negramotným osobám.
- **čítanie textu so súčasným zvýraznením textu** – táto funkcia pomáha osobám s kognitívnymi poruchami posilniť porozumenie textu pri čítaní počas používania mobilnej aplikácie.
- **automatické dopĺňanie textových vstupov** – niektoré mobilné zariadenia majú túto funkciu, ktorá pomáha pri výmene textu s predinštalovanými textami, aby sa znížil počet stlačených kláves a aby sa zabránilo vzniku gramatických chýb.
- **nastaviteľný čas na dokončenie operácie** – poskytuje používateľom dostatok času na dokončenie funkcií mobilných aplikácií vlastným tempom, podľa vlastného uváženia [1].

1.1.5 Zhrnutie analýzy vlastností

Analýza vlastností je veľmi dôležitým krokom pred samotnou tvorbou mobilnej aplikácie. Treba si uvedomiť a pochopiť rôzne prekážky, obmedzenia a samotné správanie sa ľudí trpiacich nejakým postihnutím, či už zrakovým, sluchovým, telesným alebo

kognitívnym. Je potrebné zistiť čo treba do mobilnej aplikácii zahrnúť, aby vyhovovala a pomáhala čo najväčšiemu počtu ľudí. Je potrebné odstrániť prekážky, ktoré spôsobujú ťažkosti s ovládaním aplikácie ale aj problémy so samotnými funkciami. Nasledujúca tabuľka (Tabuľka 1.) prezentuje vlastnosti, ktoré chcem prostredníctvom mojej aplikácie posilniť a na druhej strane tie, ktoré chcem potlačiť.

Tabuľka 1. Analýza vlastností[autor]

vlastnosti, ktoré chcem posilniť	vlastnosti, ktoré chcem potlačiť
vnímateľnosť prostredia	nezrozumiteľnosť
vyššia samostatnosť	nedostatočné možnosti čítania
práca so smartfónom	bežné problémy (návšteva lekára)
jednoduchosť	nutnosť fyzickej pomoci
rýchlosť	
prispôsobenie	
uľahčiť život nevidiacim a slabozrakým	

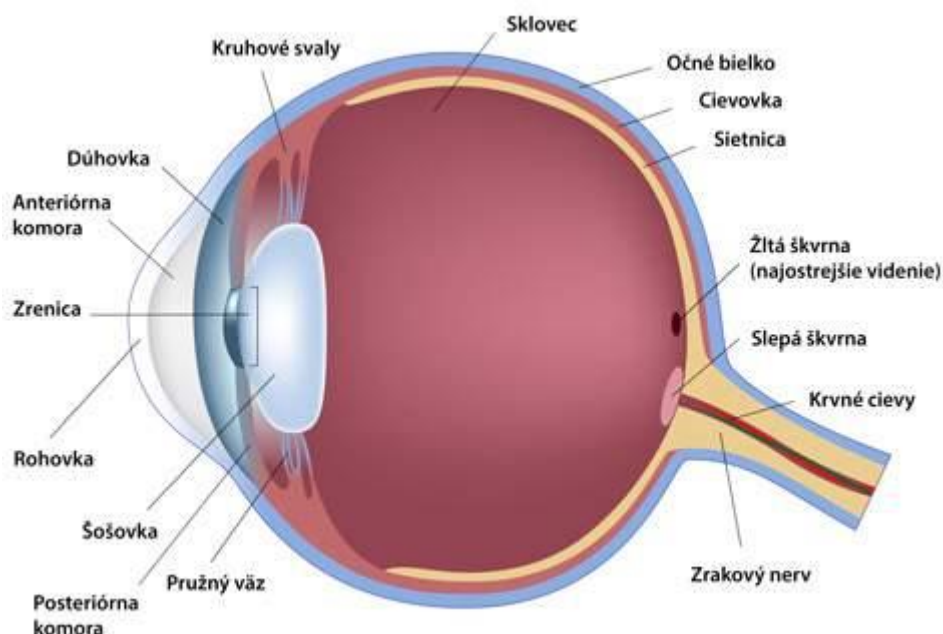
1.2 Funkcionalita zrakového orgánu

Zrak je spomedzi všetkých ľudských zmyslov ten, ktorý je pre človeka z hľadiska orientácie v priestore a vnímania svetla najdôležitejší. Oko je zmyslový orgán reagujúci na svetlo. Pomocou zraku dokážeme vnímať nielen svetlo, ale aj farby a tvary. Ľudské oko funguje ako fotoaparát, ktorý sníma okolitý svet a zachytáva jeho krásu. Oko je len nástrojom videnia, zvyšok je výsledok práce mozgu. To čo dokážeme vidieť pomocou očí sa prenáša cez zrakové nervy do mozgu vo forme nervových signálov, ktorý tieto signály spracuje a vyhodnocuje. Interpretácia týchto signálov predstavuje výsledok nášho videnia sveta.

Oko má v priemere 24 mm. V očniciach sú uložené a dobre chránené oči. Pracujú ako filmové kamery, a v závislosti od toho, že sú od seba vzdialené, každé z nich vidí trochu odlišný obraz. Obrazy, ktoré následne vznikajú si mozog spája. Oko je zložené z troch vrstiev:

- **vonkajšia väzivová vrstva** - pozostáva z rohovky, ktorá pomáha zaostrovať obraz a na udržanie veľkosti a tvaru očnej gule slúži bielko.
- **stredná vrstva** - pre prísun výživy a kyslíka je potrebné veľké množstvo pigmentových buniek a ciev, ktoré sa nachádzajú v ciehovke. Šošovka je prehl'adné pružné teliesko zavesené na vráskovcovom telese, ktoré umožňuje oku vytvárať ostrý obraz. Na reguláciu veľkosti rohovky slúži dúhovka v tvare kruhu. Zrenica je otvor v dúhovke, ktorá podľa množstva svetla, ktoré dopadá do oka sa zväčšuje alebo zmenšuje sťahmi svalov v dúhovke.
- **vnútorná vrstva** - v sietnici sa nachádzajú tyčinky a čapíky. Na rozlíšenie svetla, tmy, tvaru, pohybu slúžia tyčinky, ktoré snímajú len čiernu a bielu farbu. Na

farebné videnia za jasného svetla slúžia čapíky. Miesto najostrejšieho videnia je žltá škvrna, kde sa nachádza veľké množstvo čapíkov



Obrázok 1. Prierez oka [36]

Proces ľudského videnia sa začína vstupovaním svetelných lúčov pozorovaného objektu do oka zrenicou. Lúče putujú ďalej cez rohovku a šošovku. Lúče sú ohýbané šošovkou tak, že sa vytvára zmenšený a prevrátený obraz na sietnici. Na sietnici sú tyčinky a čapíky, ktoré premieňajú svetelné lúče na nervové impulzy. Tie prechádzajú do mozgu, ktorý vníma nervové impulzy ako obraz prevrátený správne nahor. Posledným štádiom je uvedenie si pozorovaného objektu [2].

Výsledná aplikácia má za cieľovú skupinu určenú ľudí so zrakovým postihnutím. Je dôležité poznať základné informácie ohľadom funkčnosti a činnosti oka. Vzhľadom na to, že chceme aby naša aplikácia vyhovovala čo najväčšiemu počtu ľudí z cieľovej skupiny, je potrebné vedieť chápať ako títo ľudia vnímajú okolitý svet, to znamená vedieť vcítiť sa do ich každodenného života. Významnú rolu zohráva ich rozdelenie do viacerých kategórií, podľa stupňa poškodenia oka.

1.2.1 Problematika zrakovo postihnutých ľudí

Podľa organizácie World Health Organization (WHO) sa odhaduje, že na svete žije 36 miliónov slepých a 124 miliónov slabozrakých osôb. Približne 85% ľudí, ktorí sú slepí alebo majú ťažšie poškodenie zraku sú vo vekovej hranici 50 rokov a viac. Najnovšie výpočty odhadujú, že počet ľudí s poškodením zraku by sa mohol strojnásobiť v dôsledku

rastu populácie a starnutia. Do roku 2050 sa očakáva, že počet slepých ľudí bude 115 miliónov [3].

1.2.2 Typy zrakových postihnutí

Pojem zrakové postihnutie nepredstavuje len samotná slepota. Poznáme zrakové postihnutia, z ktorých každé je niečím špecifické. Zrakové poškodenie sa môže prejavíť v troch smeroch: znížením zrakovej ostrosti, obmedzením alebo poškodením zorného poľa alebo poruchou farebného videnia. Podľa doby vzniku a príčiny vzniku rozoznávame nasledujúce typy porúch:

- funkčné poruchy – tupozrakosť
- orgánové poruchy – porucha v oku
- opakujúce sa poruchy – striedavé škúlenie
- dlhodobé poruchy – chronický zápal sietnice
- krátkodobé poruchy – zápal nervu
- získané poruchy – oslepnutie
- vrodené poruchy – kogenitálna slepota

Diferenciácia zrakovo postihnutých zohráva dôležitú úlohu pri vytváraní systému špeciálnych zariadení pre ľudí so zrakovými poruchami.

Tabuľka 2. Kategorizácia zrakových porúch podľa WHO [autor]

Kategória	Stupeň poruchy	Ostrosť zraku ¹
Normálny zrak	Žiadna Lahká	5/6, 6/7, 20/25 alebo lepší menej než 5/7, 6/7, 20/25
Slabý zrak	Stredná Silná	Menej než 5/15, 6/20, 20/80 Menej než 5/40, 20/160
Slepota	Hlboká Úplná	Menej než 5/100, 20/400 Žiadne vnemy svetla

Podľa oftalmologických kritérií sa rozdeľujú zrakovo postihnutí ľudia do troch hlavných skupín – nevidiaci, slabozrakí, binokulárne chybní (tupozrakí a škúľaví). Jesenský špecifikuje osoby so zrakovým postihnutím podľa druhu a stupňa postihnutia na:

- nevidiacich (úplná slepota, zachovalý svetlocit, zachovalé čiastočné videnie),
- slabozrakých (ťažký, stredný a ľahký stupeň) ,
- binokulárne chybných (tupozrakí, škúľaví, monokulus),
- osoby s kombinovaným postihnutím.

¹ Ostrosť zraku udáva pomer vzdialeností na ktoré sa pozerá postihnutý človek a človek zdravý

Prejavom zrakového postihnutia môže byť porucha zrakovej ostrosti, porucha binokulárneho videnia a farbosleposť. Sprostredkovanie informácií sa u nevidiacich ľudí realizuje prostredníctvom sluchového a hmatového kanálu. Písomná reč sa realizuje pomocou Braillovho písma, využitím tyflografických pomôcok a tyflotechniky. V dôsledku zrakových porúch dochádza k zmenám v stavbe a činnosti zrakových orgánov a procesu videnia.

Tupozrakosť patrí k poruchám zrakovej ostrosti. Je to funkčná porucha, pretože jedno oko vidí ostrejšie ako druhé. Takíto ľudia sa stávajú monokulom, to znamená, že vidia len jedným okom. Porucha rovnovážneho postavenia očí je **škul'avosť**. Prejavuje sa ako odchýlka jedného oka od druhého. Tupozrakosť a škulenie sú dva javy, ktoré spolu úzko súvisia a navzájom sa ovplyvňujú.

Slabozrakí ľudia tvoria samostatnú skupinu zrakovu postihnutých. Typickým znakom je znížená schopnosť zrakového vnímania, najmä čo sa týka presnosti a rýchlosti. Častým problémom je narušenie zorného poľa alebo farbecitu. Rozdiel medzi slabozrakými a tupozrakými ľuďmi je v tom, že slabozrakosť je poškodenie samotného zrakového orgánu a tupozrakosť je len poškodením jeho funkcie.

Čiastočne vidiaci sú skupina ľudí so zvyškami zraku, alebo prakticky nevidiaci. Využívajú zvyšky zraku, ale zrakový analyzátor nemá vedúce postavenie. Rozhodujúcu úlohu zohráva využívanie kompenzačných mechanizmov, pretože zrakové vnemy a predstavy sú obmedzené a skreslené. Čiastočne vidiaci majú najťažšiu slabozrakosť, ktorá sa prejavuje tak, že pri pozeraní obidvoma očami za pomoci okuliarov, rozoznávajú prsty až tesne pred očami.

Nevidiaci neprijímajú informácie z prostredia vizuálnou cestou, ani nevznikajú u nich žiadne zrakové pocity. Slepota sa považuje za najťažší zrakový defekt. Prejavuje sa stratou funkcie zrakového analyzátora pre chorobné stavy očí, nervových dráh alebo mozgového centra. **Farboslepí** sú skupina ľudí, ktorí nedokážu správne určiť farby, alebo majú problém len s rozoznávaním určitých farieb, buď si farby zamieňajú alebo môže ísť o totálnu farbosleposť. Tento druh postihnutia sa vzťahuje k obrázkom a textom, ktoré sú na farebnom podklade [4].

1.2.3 Príčiny zrakových postihnutí

Vizuálne poruchy väčšinou vznikajú z inej choroby alebo z podvýživy. Strata zraku môže byť zapríčinená zápalom spojivky alebo rohovky, ktorá vzniká pri zanedbaní osobnej hygieny. U detí je to zapríčinené nedostatkom vitamínov. Existujú aj prípady, že strata zraku

je spôsobená vrodeným nevytvorením určitých tkanív oka, čo sa prejavuje defektom dúhovky alebo ostatných častí oka [5].

Rizikovými faktormi sú:

- 1) **dedičnosť** – ak niektorý z rodičov trpí poruchou zraku, existuje vysoká pravdepodobnosť, že ju môže zdediť aj potomok
- 2) **vek nad 45 rokov** – pokročilejším vekom sa stáva, že oči nevykonávajú svoju funkciu tak ako by mali, a to vedie často k slabozrakosti
- 3) **poranenie oka** – môže vzniknúť pri športe, v práci alebo pri bežnej činnosti
- 4) **nedostatočná strava** – najmä v oblastiach, kde podmienky na život pre človeka nie sú priaznivé a nevychádza to z vlastnej vôle
- 5) **fajčenie, zafajčené prostredie** – dym z cigariet negatívne pôsobí na zrak

1.2.4 Typy zrakových porúch

Zrakové poruchy sa delia podľa toho, aká oblasť zrakových funkcií je porušená. Môže ísť o stratu zrakovej ostrosti, narušenie zorného poľa, okulomotorické poruchy, problémy spojené s farbami. Každá zraková porucha je bližšie špecifikovaná.

Ak ľudia nevidia zreteľne a majú problémy s rozlišovaním detailov, vtedy ide o **stratu zrakovej ostrosti**. Takto postihnutí ľudia nemusia mať problémy s identifikáciou veľkých predmetov. Stupeň postihnutia je rozdielny. Zraková ostrosť sa meria Snellenovými optotypmi.

Porucha zorného poľa sa spája s obmedzeným priestorom, ktorý ľudia vidia. Pri pohybe v priestore narážajú na predmety na tej strane, kde je práve výpadok zorného poľa. Môžu sa objaviť problémy s rozlišovaním farieb, môže sa zhoršiť videnie za šera a pri adaptácii na zmenu osvetlenia.

Pri chybnnej koordinácii pohybu očí vznikajú **okulomotorické poruchy**. Problémy vznikajú pri používaní obidvoch očí a pri sledovaní pohybujúceho sa predmetu. Predmet sleduje najprv jedno oko a až potom druhé. Typické sú aj ťažkosti s uchopovaním predmetov.

Problémy so spracovaním zrakových podnetov sú spojené s poškodením zrakových centier v kôre mozgu. Prejavujú sa problémami s interpretáciou zrakovéj informácie a jej spojením s ostatnými zmyslovými vnemami pri vytváraní zrakového obrazu u ľudí.

Farebné videnie je schopnosť oka rozlišovať rôzne dĺžky elektromagnetického vlnenia a vnímať ich ako farby. Množstvo farebných odtieňov vzniká kombináciou troch základných farieb a to červenej, modrej a zelenej. **Poruchy farebného videnia** môžu byť vrodené alebo získané počas života [6].

1.3 Prehľad pomôcok pre nevidiacich

Základnou podmienkou rozvoja každého jedinca je neustály prísun informácií, ktoré prijíma zo svojho okolia a určitým spôsobom ich spracováva. Poškodenie, porucha alebo strata zraku obmedzuje, deformuje alebo úplne vylučuje získavanie informácií prostredníctvom zraku. Kompenzačnou pomôckou pre postihnutých ľudí je nástroj alebo zariadenie špeciálne vytvorené a upravené tak, aby svojimi vlastnosťami a možnosťami použitia kompenzovalo niektorú nedostatočnosť spôsobenú daným postihnutím [7].

1.3.1 Historický vývoj

Spočiatku boli zrakovo postihnutí ľudia odkázaní na prístup k informáciám, ktoré boli sprostredkované vidiacimi ľuďmi. Prelomom sa stalo objavenie reliéfného písma čítaného hmatom. Bola to prvá informačná revolúcia umožňujúca nevidiacim nielen čítať, teda prijímať informácie, ale aj písať, čiže s nimi aktívne narábať. Celosvetovým trendom sa stalo Braillovo písmo. Najprv sa písalo len ručným „vypichovaním“ do papiera, postupne sa začali používať mechanické písacie stroje a tlač lisovaním pomocou matric a nakoniec počítačom riadené tlačiarne. Hlavnou nevýhodou Braillovho písma je nevratná trojrozmerná deformácia nosiča informácie - papiera.

Základom druhej informačnej revolúcie sa stalo rozhlasové vysielanie ako zdroj obrovského množstva aktuálnych informácií rovnocenných pre zrakovo postihnutých a vidiacich poslucháčov. Hovorené informácie prekonali informačnú bariéru v čase, keď sa na trhu objavili prvé magnetofóny určené na reprodukciiu a záznam zvukových nahrávok. Výhoda spočívala v možnosti budovať rýchlejšie, lacnejšie a väčšie knižničné fondy zvukových nahrávok dokumentov v porovnaní s fondami kníh v Braillovom písme. Avšak najväčšou výhodou je, že zvukový záznam môže využívať každý zrakovo postihnutý človek s použitím jednoduchého ovládateľného magnetofónu bez potreby špeciálneho zaučenia sa.

Tretia informačná revolúcia pre zrakovo postihnuté osoby predstavuje použitie výpočtovej techniky v oblasti informácií a telekomunikácií. Hlavnou myšlienkou je schopnosť samotnej výpočtovej techniky vytvoriť komunikačné rozhranie medzi človekom a strojom, zabezpečujúce obojsmernú výmenu informácií s obmedzenou zrkovou kontrolou

alebo bez nej. Nevidiaci a slabozrakí ľudia majú postupne prístup k informáciám bez sprostredkovateľa. Ponúka im rovnaký výber informácii ako majú vidiaci ľudia, bez nutnosti prekonávania svojej obmedzenej mobility [8].

1.3.2 Členenie existujúcich pomôcok

Táto časť je zameraná na poskytnutie stručného prehľadu pomôcok a zariadení, ktoré v každodennom živote pomáhajú ľuďom so zrakovým postihnutím. Kompenzačné pomôcky sa môžu deliť podľa viacerých kritérií. Podľa účelu použitia sa členia na:

- pomôcky uľahčujúce alebo umožňujúce orientáciu a pohyb nevidiaceho (napr. biela palica, akustické hlásiče, orientačný kompas)
- pomôcky pre každodenné použitie a do domácnosti (napr. hodinky, meracie prístroje, šablóna na bankovky)
- pomôcky na sprístupnenie, získavanie a spracovávanie informácií (napr. kamerové lupy, osobný počítač s hlasovým vstupom)

Základné a historicky staršie členenie prostriedkov na pomoc nevidiacim a slabozrakým ľuďom je na optické a elektronické. **Optické pomôcky** uľahčujú čítanie slabozrakým ľuďom. Najvýznamnejším predstaviteľom neelektronických pomôcok sú jednoznačne lupy. K **elektronickým pomôckam** patria televízne lupy pre pacientov so zbytkami zraku a slabozrakých. Môžu to byť kamerové zväčšovacie TV lupy alebo digitálne zväčšovacie TV lupy.

Pri výbere vhodnej pomôcky je dôležitá požiadavka pacienta, na aký účel potrebuje danú pomôcku. Účelom môže byť videnie do diaľky, čítanie na blízko alebo písanie. Veľmi dôležitým javom je psychické nastavenie pacienta, dostatočná vôľa pacienta a dostatočná inteligencia pre manuálnu zručnosť. Pre konečný výber pomôcky je zásadné testovanie používania danej pomôcky [9].

1.3.3 Optické pomôcky

Medzi hlavné optické prístroje ktoré slúžia na podporu činností nevidiacich patria šošovky, lupy, ďalekohľadové systémy a turmony. **Šošovky** sú vkladané do okuliarov, ide o monokulárny turmon, pričom okrem čítania informácií z rôznych kníh a textov slúžia nevidiacim aj v bežnom živote, aby boli schopní overiť informácie na diaľku.

Lupa je najjednoduchšia subjektívna pomôcka, vytvárajúca nereálny obraz, ktorý pozorujeme pod zväčšeným zorným uhlom. Slúži najmä na zjednodušenie čítania. Môže byť

tvorená jednou spojnou šošovkou alebo sústavou šošoviek s ohniskovou vzdialenosťou menšou ako je konvenčná pozorovacia vzdialenosť. Za optimálnu vzdialenosť pre pozorovanie blízkych predmetov alebo bodov sa považuje konvenčná vzdialenosť menšia ako 25 cm. Slabozrakí pacienti používajú lupu na čítanie v takej vzdialenosti, ako by čítali bez nej. Podľa spôsobu držania sa delia na:

1) **Binokulárne hlavové lupy** – konštrukcia pozostáva z ľahkého materiálu. Sú vybavené xenónovým svetlom a vďaka šošovkám vyrobeným z polyakrylátu poskytujú široké zorné pole. Nachádza sa tam mechanika na prispôsobenie obvodu hlavy, čo spôsobuje, že ruky môžu byť pri práci voľné. Využívajú sa v stomatológii, oftalmológii, veterine, kozmetike a v ďalších odvetviach.

2) **Závesné lupy** – po zavesení na krk poskytujú možnosť pracovať s oboma rukami. Používajú sa na všetky ručné práce ako je vyšívanie a pletenie.

3) **Príložné lupy** – sú známe ako „rybie oko“, majú stálu ohniskovú vzdialenosť. Sú určené najmä pre deti a starších ľudí, pretože eliminujú tras ruky alebo problémy s motorikou horných končatín.

4) **Vreckové lupy** – praktické využitie pri bežných činnostiach. Pre ľudí so zníženou zrakovou ostrosťou je k dispozícii farebné puzdro pre jednoduchšie hľadanie lupy.

5) **Stojanové lupy** – možnosť uchytenia o stôl pomocou klbového stojana. Sú nápomocné slabozrakým ľuďom pri čítaní, písaní a pri rôznych činnostiach, keďže stojan umožňuje meniť konvenčnú vzdialenosť.

6) **Digitálne USB lupy** – špeciálna lupa s LED osvetlením a výsuvnou CMOS makrokamerou a USB výstupom. Najviac je využívaná numizmatikmi, filatelistami ale aj v školách na študijné účely.



Obrázok 2. Stojanová lupa[37]



Obrázok 3. Digitálna USB lupa 200x[37]

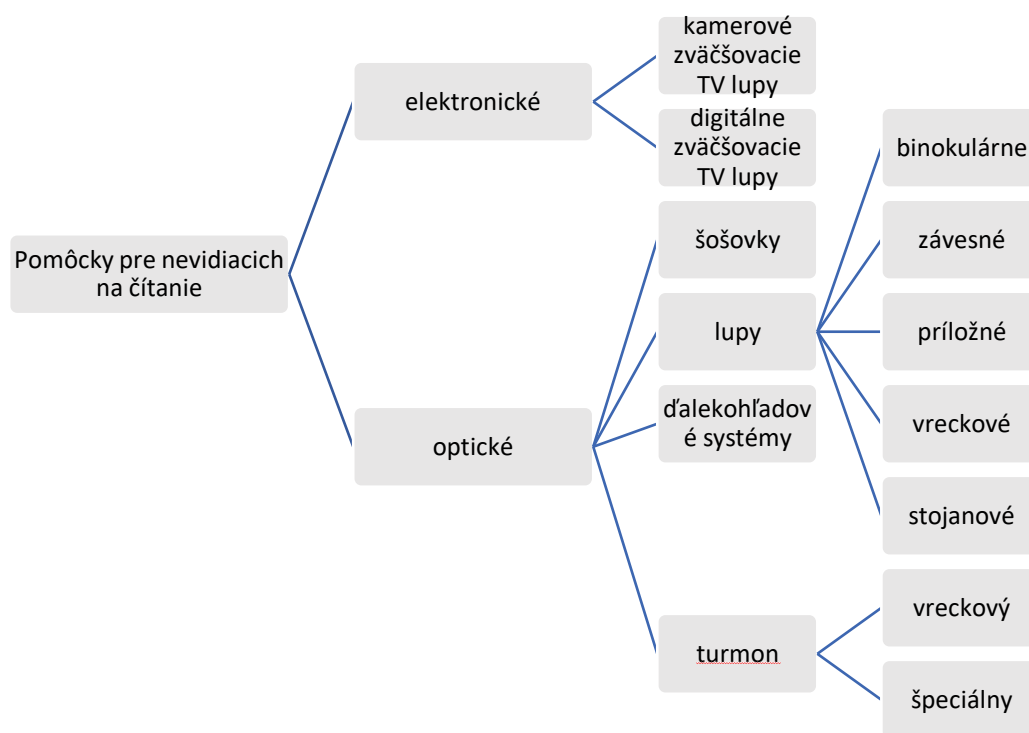


Obrázok 4. Turmon[37]

Optickými pomôckami pre strednú vzdialenosť a do diaľky sú **d'alekohľadové systémy**. Rozoznávame dva typy systémov podľa využitia a optických prvkov: Galileiov princíp a Keplerov princíp ďalekohľadu. Ďalekohľadový systém typu Galilei je afokálna sústava zložená z objektívu a okuláru so zápornou ohniskovou vzdialenosťou. Jeho zväčšenie je maximálne 6x. Ďalekohľadový systém typu Kepler je tiež afokálny systém, ktorý tvoria dve spojné šošovky, objektív a okulár.

Turmon je optická pomôcka s fixným či nastaviteľným zaostrením konštruovaná podľa Keplerovho systému. Je odporúčaný osobám s veľmi nízkou zrakovou ostrosťou (3/50 až 4/50). Existuje v dvoch variantoch, a to ako vreckový a špeciálny turmon [10].

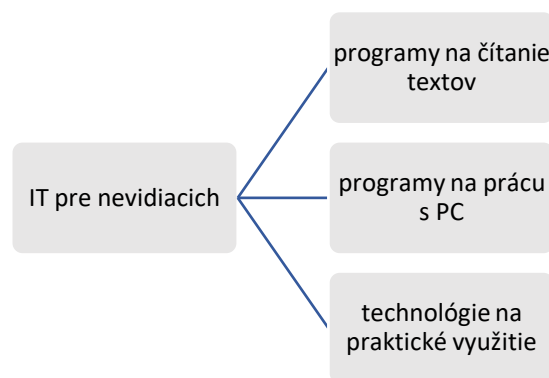
Zhrnutie rozdelenia pomôcok bez použitia prostriedkov IT sa nachádza na nasledujúcom diagrame (Obrázok 5).



Obrázok 5. Rozdelenie pomôcok na pomoc nevidiacim[autor]

1.3.4 Vybrané prostriedky IT

V danej časti sú uvedené len niektoré vybrané prostriedky pre slabozrakých a nevidiacich, ktoré využívajú informačné technológie. Prehľad slúži na analýzu možností zlepšenia a skvalitnenia života ľuďom s poruchami zraku. Pred tvorbou aplikácie je dôležité vedieť, aké zariadenia, programy, prístroje už existujú a ako pomáhajú zrakovu postihnutým ľuďom. Zariadení využívajúcich informačné technológie je v súčasnom modernom svete obrovské množstvo, no pre účely bakalárskej práce som opísal len niektoré z nich.



Obrázok 6. Využívanie IT slabozrakými a nevidiacimi[4]

Tyflosonar Ray je malý, ľahký a elektronický prístroj vydávajúci akustické alebo vibračné signály, navrhnutý ako doplnok k bielej palici. Vďaka ultrazvukovému snímaniu pomáha rozpoznávať prekážky až do vzdialenosti 2,85m a upozorňuje na ich prítomnosť. Jeho špeciálny „ESCAPE“ mode umožňuje používateľovi nájsť cestu aj pri veľkom množstve ľudí. Vibračná signalizácia umožňuje, že je vhodný na použitie v hlučnom prostredí [19].

Gala Tee je braillovská klávesnica pre nevidiacich, ktorú je potrebné pripojiť k počítaču alebo notebooku pomocou USB a hneď je pripravená k práci. Nie je nutnosť žiadneho špeciálneho programu. Počítač je ovládateľný len pomocou 17 klávesov. Má v sebe zabudovanú podporu pre všetky operačné systémy vrátane Linuxu.

Braillovský riadok pre nevidiacich **FOCUS 40 Blue** vyniká malými rozmermi, braillovskou klávesnicou a bezdrôtovým prenosom pomocou bluetooth. Ponúka 40 buniek zobrazujúcich jednotlivé znaky, ktoré sú veľmi dobré hmatateľné. Pomocou bezdrôtového prenosu môže pracovať až 20 hodín bez prerušenia.

Pen Friend je prístroj s hlasovou príručkou v slovenskom jazyku, ktorý umožňuje identifikáciu predmetov. K dispozícii je 870 kusov etikiet na ktoré je možné nahráť hlásku podľa predmetu, ktorý chcete označiť. Etiketa sa potom nalepí na zvolený predmet a pomocou tzv. hovoriaceho pera je možné predmet ľahko vyhľadať. Obsluha pera je jednoduchá nielen vďaka hlasovej pomoci, ale najmä kvôli dizajnu v tvare pera s ľahko hmatateľnými štyrmi tlačidlami [20].

Na rozpoznávanie farieb sa používa zariadenie **Colortest** so zabudovaným hlasovým výstupom. Videnie farieb je podobné tomu, ako ich vidí ľudské oko. **Colortest Memo** je zariadenie s podobným dizajnom a funkciami ako má Colortest, no rozšírením je rozlišovanie bankoviek. Rozpoznávanie eurobankoviek je jednoduché. Stačí prejsť

jednoduchým pohybom po eurobankovke a následne prostredníctvom hlasového výstupu Colortest Memo prečíta jej hodnotu [21].

MAGic je výkonný program na zväčšovanie informácie a navigáciu na obrazovke počítača. Poskytuje zväčšenie 2 až 36-krát s vyhladzovaním písma a farieb. Voliteľnosť inverzného farebného zobrazenia zlepšuje kontrast a uľahčuje čítanie. Umožňuje slabozrakým ľuďom efektívne využívať aplikačné programy pre voľný čas, vzdelávanie ale aj zamestnanie. **MAGic Plus** je rozšírením programu Magic o hlasovú podporu. Okrem prezerania informácie, dochádza aj k jej vyslovovaniu. Reč a zväčšenie môžu byť používané samostatne ale aj súčasne. Odozva klávesnice opakuje pri písaní znaky alebo slová, dokonca aj celé vety. Ozvučený ukazovateľ myši číta jednotlivé riadky a slová, keď cez nich prechádza [22].



Obrázok 7. Vybrané pomôcky pre nevidiacich[38]

Sara je skenovacie zariadenie pre nevidiacich a slabozrakých používateľov, pomocou ktorého si môžu prečítať knihy, časopisy, audio knihy a mnoho ďalších. Môže byť pripojené k externému monitoru, ktorý umožňuje čítanie textu vo zväčšenom písme. Jednoduché ovládanie funkcií a jednoduchá orientácia je vďaka veľkým rôznofarebným ovládacím tlačidlám. V prednej časti zariadenia sa nachádzajú stereo reproduktory [23].

Finereader 10 je program na optické rozpoznávanie písma. Systém OCR (Optické rozpoznávanie znakov) dokáže automaticky prevádzať nasnímané tlačené dokumenty do počítača v plno editovateľnej forme, ktoré potom dokáže program JAWS prečítať alebo zväčšiť program Magic. Pomáha nevidiacim a ťažko slabozrakým ľuďom samostatne čítať tlačené dokumenty. Umožňuje paralelne spracovávať viaceré úlohy, čo vedie k zvyšovaniu efektivity. Je ľahko ovládateľný a zachováva štruktúru dokumentu [24].

Novou, inovatívnou verziou elektronických stolových čítacích lúp je verzia od firmy Optelec – Tieman Group s názvom **ClearView+**. Základ tvorí pohyblivý čítací stolík nad ktorým je umiestnená snímacia kamera. Používa sa najmä na čítanie textov pre ťažko slabozrakých ľudí. Súčasťou stolíka je ovládací panel, takže ruky nie je potrebné pri čítaní premiestňovať. Kontrast, jas a zväčšenie lupy sa ovládajú elektronicky, pričom zaostrovanie je automatické. Prídavný modul umožňuje vykonávanie ďalších funkcií ako sú ukazovadlo a umelé zafarbenie. Pre seniorov je určený Zero-button, kde lupa nemá žiadne ovládacie prvky, všetko sa robí na základe intuície [25].

Compact+ patrí medzi vreckové elektronické čítacie lupy. Má vynikajúce funkčné vlastnosti a sofistikovaný dizajn. Doba prevádzky pri neustálom používaní je približne 3 hodiny. V balíčku sa okrem samotného produktu nachádza nabíjačka, prenosné puzdro a šnúrka na zavesenie. Veľkou výhodou je jeho hmotnosť, ktorá nepresahuje 300g [26].

Prenosnou elektronickou čítacou lupou, ktorá dokáže zväčšiť text a obrázky je zariadenie **FarView**. Prostredníctvom viacúčelovej kamery je možné používanie lupy v dvoch režimoch – v režime snímania na blízko a v režime snímania do diaľky. V reálnom, živom režime je možné obraz zväčšiť až 42 krát a v prezerajúcom režime až 14 krát. Dizajn a funkcie lupy sú prispôsobené na každodenné používanie bez ohľadu na priestor, ktorý je práve k dispozícii. Významným znakom je režim čítania dokumentov, pomocou ktorého sa dá posúvať v texte, ukotviť v čítanom texte a tiež návrat na začiatok textu prostredníctvom jedného tlačidla [27].

Na zväčšovanie snímaného textu alebo objektu 3,4 až 16x slúži prenosná elektronická čítacia lupa **Sapphire**. Je navrhnutá a určená pre ľudí s poruchami zraku, ktoré boli zapríčinené napríklad šedým alebo zeleným zákalom, degeneráciou makuly a podobne. Má veľmi jednoduché používanie, ktoré sa začína stlačením tlačidla, pokračuje umiestnením na objekt, ktorý chceme zväčšiť, a končí sa posúvaním žltého tlačidla pomocou ktorého sa nastaví požadované zväčšenie. Môže byť dokonca napojená na televíznu obrazovku. Vďaka svojmu tvaru sa ľahko používa a prenáša. Nevýhodou je hmotnosť, ktorá oproti Compact+ je vyššia, no na druhej strane poskytuje dlhšiu výdrž batérie [28].



Obrázok 8. Vybrané prostriedky pre nevidiacich – čítacie[38]

Najrozšírenejším a najpopulárnejším čítačom PC obrazovky pre nevidiacich je program **JAWS**, ktorý je určený pre operačný systém Windows. Zabezpečuje prístup k programom a Internetu aj napriek tomu, že ľudia nemusia vidieť obrazovku. Spolupráca so softvérovým syntetizérom reči a zvukovou kartou počítača zabezpečuje čítanie všetkých potrebných informácií. Je populárna najmä kvôli svojej špecifickej črte, ktorou je rozprávanie syntetickou rečou v slovenčine, češtine a v iných svetových jazykoch. Používateľ tohto programu dokáže sám priradzovať alebo modifikovať braillovské aj pomocné hlásenia pre rôzne objekty nachádzajúce sa na obrazovke [29].

Daisy je flash prehrávač nachádzajúci sa v malom, multifunkčnom a šikovnom PDA zariadení nazývanom **Tomboy**, ktorý okrem spomínaného DAISY-ho obsahuje aj zvukový záznamník, rádio a kalendár. Tomboy dokáže prehrávať zvukové formáty wav, aac, mp3 ako aj textové txt súbory. Prevod textového vstupu na rečový výstup je možný v slovenčine ako aj v češtine. Pre používateľa je výhodné najmä to, že zariadenie má v sebe zabudovaný návod, ktorý pomáha pri jeho používaní. Výdrž lítium polymérovej nabíjacej batérie je

približne 20 hodín, čo je dostatočný čas na prehratie všetkých potrebných informácií, ktoré majú slúžiť na pomoc ľuďom, ktorých trápia problémy so zrakom [30].

MP3 prehrávač **Transcend 860** sa používa na prehrávanie zvukových kníh, ktoré si nevidiaci ľudia prostredníctvom internetu sťahujú zo Slovenskej knižnice pre nevidiacich v Levoči vo formáte WMA DRM 10. Má kapacitu pamäte 4GB, čo umožňuje uloženie približne 12 zvukových kníh. Významným znakom je možnosť počúvania bez slúchadiel, vďaka vstavanému reproduktoru [31].



Obrázok 9. JAWS[39]



Obrázok 10. Tomboy[40]



Obrázok 11. Transcend 860[41]

1.4 Existujúce mobilné aplikácie

Vývoj mobilnej aplikácie, ktorá dokáže pomáhať ľuďom so zrakovým postihnutím je veľmi zaujímavý nielen z pohľadu spoznania sa s novými technológiami, ale hlavným dôvodom je, že v reálnom svete môže pomáhať nevidiacim a slabozrakým ľuďom pri rutinných činnostiach, ktoré by bez prítomnosti tejto aplikácie nedokázali zvládnuť. V poslednej dobe počet mobilných vývojárov narastá, s čím súvisí aj narastajúci počet existujúcich mobilných aplikácií, preto sú analyzované len niektoré z nich. Vychádzajúc z toho, že finálna aplikácia bude pozostávať z troch častí a to OCR, teda spracovanie obrazu z kamery a jeho prevodu na text, potom nasleduje spracovanie textu a jeho prevod na zvukový výstup, a posledná časť predstavuje zadávanie zvukových príkazov, teda analýza hlasu, zameral som sa na analýzu mobilných aplikácií zapadajúcich do tejto kategórie.

1.4.1 OCR aplikácie na pomoc pre nevidiacich

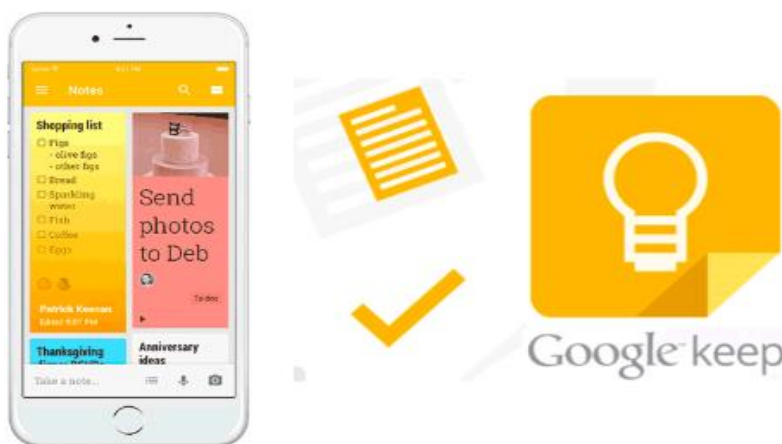
Optické rozpoznávanie znakov, označované skratkou OCR, je proces rozpoznávania a automatickej konverzie existujúcich znakov z obrázka do textového formátu, ktorý sa potom môže použiť v rôznych aplikáciách. Ide o rozsiahly proces pozostávajúci zo šiestich krokov. Jednotlivé kroky na seba nadväzujú a vytvárajú akúsi postupnosť, ktorá je

zodpovedná za konverziu textu z obrázkov. Výsledkom tejto postupnosti je získaný text z obrázka. Medzi jednotlivé kroky OCR patrí:

- optické snímanie obrazu
- umiestnenie a segmentácia
- preprocessing (predspracovanie)
- extrakcia funkcií
- klasifikácia
- postprocessing [11]

Google Keep

Táto aplikácia od spoločnosti Google umožňuje uchovávať všetky záznamy v účte Google. Pomocou aplikácie je možné nastaviť pripomienky a čas na dokončenie úloh. Taktiež je možné pridávať fotografie, obrázky a vyhľadávať poznámky podľa automaticky vytvorených tém. Významným prvkom je možnosť synchronizácie poznámok s telefónom, tabletom a hodinkami. Po odfovení obrázka je k dispozícii možnosť „Grab image text“, ktorá slúži na vybratie textu z obrázka.



Obrázok 12. Náhľad do aplikácie Google Keep[42]

Google Goggles

Aplikácia je ponúkaná vyhľadávacím nástrojom Google, ktorý používa bežné metódy OCR na úplne inej úrovni. Android OCR aplikácia poskytuje informácie o objekte jednoducho cez kameru smartfónu a dokáže konvertovať text, ktorý číta okamžite aj do iného jazyka. V praxi sa používa vtedy, keď chceme vedieť, kde sa daný výrobok predáva, odfoťme jeho názov a logo, a následne nám aplikácia vráti lokalitu možného nákupu. Tiež

je dostupná funkcia, ktorá po odfovení čiarového kódu dokáže zistiť, aký to je výrobok a pomocou vyhľadávača umožňuje zisťovať dodatočné informácie.



Obrázok 13. Náhľad do aplikácie Google Goggles[43]

Office Lens

Android OCR aplikácia je vytvorená vývojármi z Microsoftu, čo zaručuje, že ide o certifikovanú aplikáciu s dokonalou funkcionalitou. Oproti ostatným OCR aplikáciám ponúka výnimočnú vstavanú funkciu, známu pod názvom „Whiteboard mode“, ktorá umožňuje vyrezávať a odstraňovať tieň a prenikavé svetlo. Mnohí ľudia aplikáciu kritizovali kvôli jej názvu, ale aj napriek tomu je väčšina ľudí s ňou veľmi spokojná a radi ju používajú.



Obrázok 14. Náhľad do aplikácie Office Lens[44]

Text Fairy

Vynikajúca OCR aplikácia slúžiaca na zjednodušenie života ľudí, poskytujúca možnosť konvertovať písomný text do digitálneho súboru extrahovaním textu. Aplikácia umožňuje okamžité skopírovanie textu do niektorej zo schránok smartfónu a následne môže byť využívaná ostatnými aplikáciami na činnosti ako sú odosielanie správ alebo odosielanie emailov. Najväčšou výhodou aplikácie je, že dokáže rozpoznať viac ako 50 rôznych jazykov,

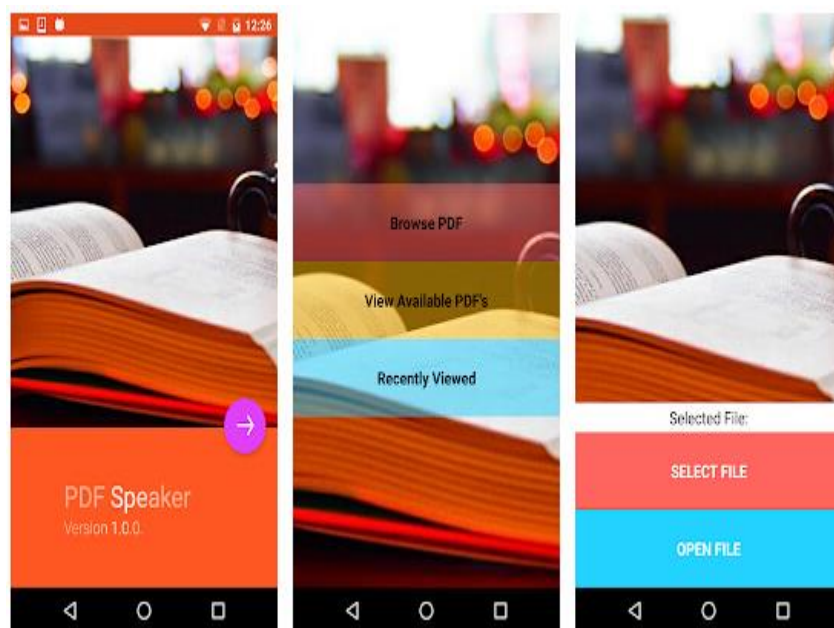
a tiež to, že je bezplatná a neobsahuje žiadne reklamy. Poskytuje možnosť konvertovania extrahovaných dát priamo do textového súboru alebo do pdf súboru [12].

1.4.2 Čítacie aplikácie na pomoc pre nevidiacich

Analýza zahŕňa aplikácie, ktoré boli vytvorené za účelom pochopenia textu v rôznych jazykoch a jeho následné prevedenie do hlasového výstupu. Pomocou týchto aplikácií môžu aj nevidiaci a slabozrakí ľudia mať potešenie z čítania rôznych kníh, článkov a publikácií. Patria sem aj aplikácie, ktoré slúžia na vytváranie zvukových záznamov z rôznych textových vstupov.

PDF Speaker

Aplikácia vytvára zvukový súbor z pdf súboru. Rozhranie aplikácie sa podobá súborovému systému, ktorý zobrazuje jednotlivé zložky usporiadané v telefóne. Vyznačuje sa jednoduchou orientáciou medzi jednotlivými pdf súbormi, čo umožňuje výber tých súborov, ktoré si aktuálne chcete vypočuť. Medzi výhody patrí, že nie je potrebné internetové pripojenie na realizáciu konverzie a tiež možnosť ukladania zvukovej nahrávky vo WAW formáte. Najväčšou nevýhodou sa javia reklamy nachádzajúce sa v aplikácii, ktoré spomaľujú prácu s aplikáciou, čo sa odzrkadľuje aj na jej výkone.



Obrázok 15. Náhl'ad do aplikácie PDF Speaker[13]

@Voice Aloud Reader

Aplikácia slúži na prevádzanie textu na zvuk, pričom je bezplatná, čiže je k dispozícii všetkým, ktorí ju potrebujú. Podporuje textové, pdf a html súbory. Umožňuje používateľovi

nastaviť rýchlosť zvuku, hlasitosť a mnoho ďalších funkcií, z čoho vyplýva vlastnosť prispôsobiteľnosť používateľovi a jeho potrebám. Významnou črtou je možnosť zmeny jazyka alebo hlasového výstupu. Dostupnými jazykmi sú angličtina, francúzština, nemčina, taliančina, kórejčina, portugalčina a ruština.



Obrázok 16. Náhľad do aplikácie @Voice Aloud Reader[45]

Natural Reader Text to Speech

Ide o aplikáciu určenú na konverziu pdf súboru na hlasový výstup. K dispozícii je viacero prirodzených hlasov a tiež možnosť synchronizácie dokumentov v cloudovom úložisku aj v CC režime. Podporuje viacero služieb ako sú Dropbox, OneDrive či Google Disk. Výhodami aplikácie sú jednoduché a ľahké ovládanie a tiež fakt, že sa tam nenachádzajú reklamy. Nevýhodou je nedostatok offline podpory pre konverziu textu na zvuk [13].



Obrázok 17. Náhľad do aplikácie Natural Reader Text to Speech[13]

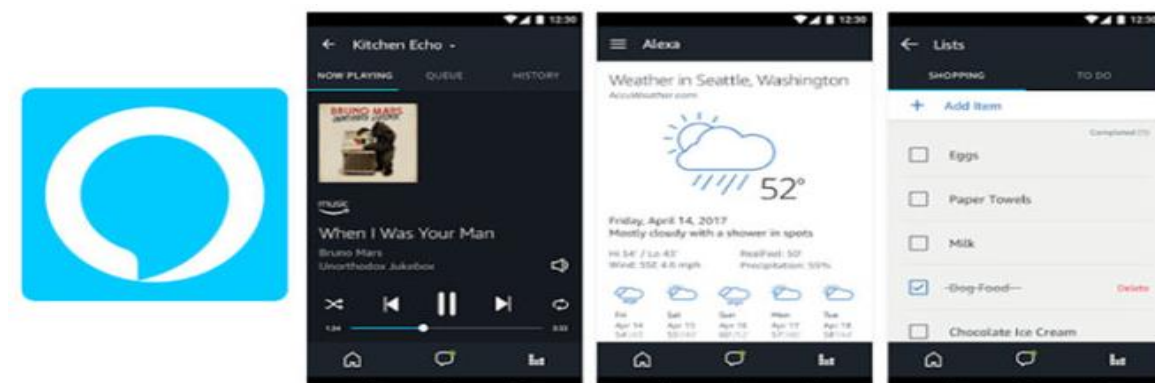
1.4.3 Aplikácie s hlasovými povelmi na pomoc pre nevidiacich

Aktuálne mobilné operačné systémy majú v sebe zabudovanú hlasovú asistentku. V súčasnosti už existujú aplikácie, ktoré sú zamerané na pomoc pre nevidiacich využívaním

hlasových povelov. Nevidiacim a slabozrakým ľuďom stačí povedať kľúčové slovo alebo vetu, a mobilný telefón vykoná danú akciu, či už vytočí telefónne číslo, odfoťí a mnoho ďalších funkcií.

Amazon Alexa

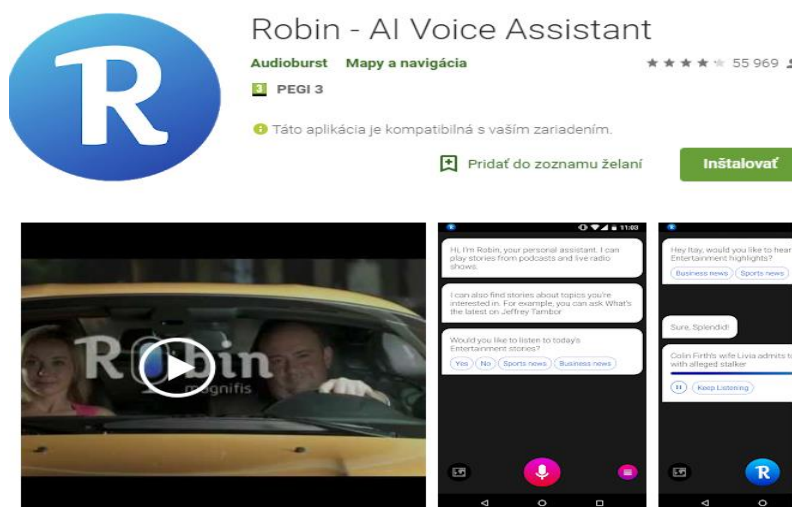
Aplikácia pre Android umožňuje streamovať playlist, dokáže povedať, aké bude počasie, prečíta novinové správy a dokonca aj povie vtip. Ak sa v domácnosti nachádzajú inteligentné produkty od Amazonu, a tie sú integrované s ekosystémom Amazonu, potom aplikácia Alexa ponúka skvelý spôsob ako to všetko ovládať.



Obrázok 18. Náhľad do aplikácie Amazon Alexa[46]

Robin

Vývojári aplikácie Robin sa snažili zámerne vytvoriť takú aplikáciu, ktorá bude veľkou konkurenciou pre digitálneho asistenta Siri od spoločnosti Apple. Ľudia ktorým sa nepáči Siri, sa môžu zamerať na aplikáciu pre Android s názvom Robin, ktorá nie je jej kópiou. Robin nie je len vodičským asistentom, ale je spôsobom prehľadávania a vyhľadávania informácií v telefóne. Zohráva úlohu kamaráta, ktorý hľadá informácie a vykonáva úlohy pre vás [14].



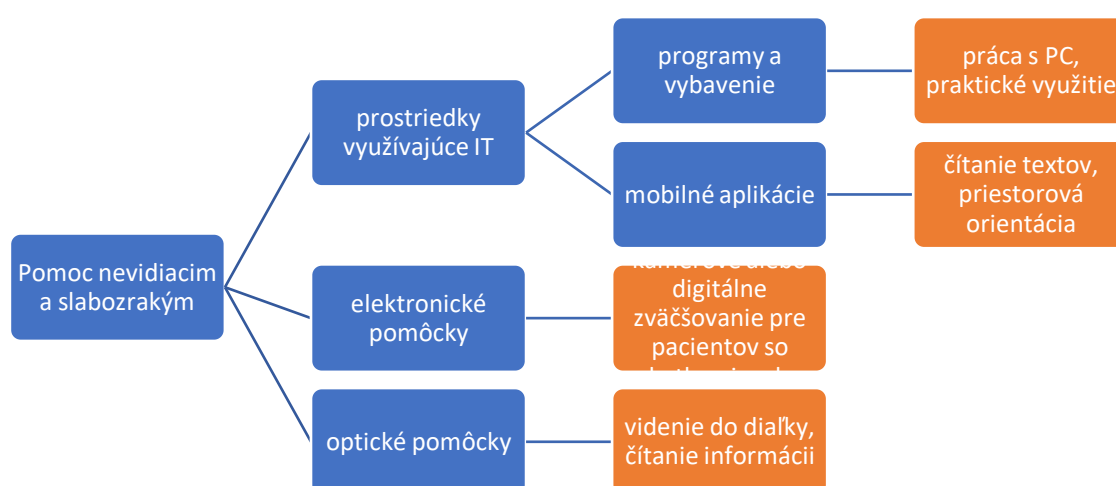
Obrázok 19. Náhľad do aplikácie Robin[47]

1.5 Zhrnutie analýzy

Podrobná analýza pomôcok a aplikácií poskytuje náhľad na súčasnú situáciu na trhu. Predstavenie jednotlivých prístrojov a aplikácií, vyzdvihnutie ich kladných a záporných stránok môže pomôcť pri tvorbe vlastnej aplikácie. Dôležité je pochopiť chyby niektorých aplikácií a vyvarovať sa ich pri tvorbe vlastnej aplikácie. Táto analýza predstavuje pre moju bakalársku prácu inšpiráciu, od ktorej sa môžem odrážať, no dôležité je prísť na trh s niečím novým, čiže vytvoriť pridanú hodnotu.

Cieľovou skupinou mobilnej aplikácie sú slabozrakí a nevidiaci ľudia, pričom je dôležité aplikáciu prispôbiť ich požiadavkám. Najvyššiu prioritu má použiteľnosť a používateľské rozhranie. Je potrebné poskytnúť jednoduchý dizajn s ľahkým ovládaním. Funkcionalita sa bude držať hesla: „V jednoduchosti je krása!“ Aj keď aplikácia bude mať v sebe zabudovanú zložitú funkcionality, pre používateľa musí navonok pôsobiť jednoducho, aby sa v nej dokázal zorientovať.

Aplikácia musí obsahovať podporu pre zrkovité postihnutie, to znamená, že má byť ovládateľná pomocou čítačky obrazovky a texty majú byť prispôbené systémovým veľkostiam. Taktiež majú byť k dispozícii hlasové inštrukcie, ktoré umožnia spustiť fotoaparát, nasnímať obraz a následne ho previesť do zvukovej podoby. Taktiež aplikácia bude disponovať históriou, kde budú ukladané predošlé výsledky vykonanej činnosti aplikácie. Pre lepšiu predstavu analýzy pomôcok a možností ich využitia slúži nasledujúci diagram (Obrázok 20.).



Obrázok 20. Zhrnutie analýzy pomôcok pre nevidiacich a slabozrakých[autor]

2 Cieľ a metodika práce

Cieľom bakalárskej práce bolo realizovať analýzu existujúcich pomôcok pre slabozrakých a vytvorenie mobilnej aplikácie na čítanie textov pre nevidiacich, ktoré bude využívať súčasné moderné technológie a nie zaťažovať ľudí. Hlavný cieľ som rozdelil na niekoľko menších úloh.

Úvodná časť: **analyzovať** existujúce zariadenia, porovnať ich silné a slabé stránky. V tejto časti som sa snažil rozdeliť prístroje a pomôcky na optické, elektronické a využívajúce informačné technológie a tiež na mobilné aplikácie. Výstupom tejto časti je prehľad, ktorý má byť nápomocný k naplneniu hlavného cieľa.

Ďalšia časť: **urobiť** výber pomôcok a nástrojov, ktoré budú použité v aplikačnom zariadení. Medzi konkrétne výbery patrí výber mobilných zariadení, programovacích jazykov a existujúcich podsystémov. Výber mobilných zariadení predstavuje výber operačného systému, pre ktorý bude aplikácia určená. Pri programovacích jazykoch je dôležité porovnať ich vlastnosti a podrobnejšie popísať ten, ktorý bude použitý na vývoj mobilnej aplikácie pre nevidiacich na čítanie. Výber existujúceho podsystému predstavuje opis vývojového prostredia v ktorom sa bude programovať naša aplikácia.

Posledná časť: **navrhnuť** samotné aplikačné zariadenie a uviesť ho do praxe, čiže splniť tretiu úlohu našej bakalárskej práce. Na začiatku treba vyhotoviť samotný návrh, potom je potrebné tento návrh implementovať, testovať a nakoniec treba vytvoriť manuál pre používateľov, ktorým je určené zariadenie. Dôležitým krokom v procese vývoja softvéru je vykonať testovanie, ktoré sa realizuje na reálnych zariadeniach a testuje sa ním funkcionálnosť, ktorá je naprogramovaná v aplikácii. Bez testovania sa nemôže aplikácia uvádzať do praxe, pretože testovanie je kľúčovým bodom vo vývoji softvéru. Vytvorenie manuálu pomáha a uľahčuje prvotnú prácu s danou aplikáciou. Jednotlivé kroky, ktoré smerujú od návrhu až po vytvorenie manuálu sa používajú pri vývoji akéhokoľvek softvéru a tieto kroky som aplikoval aj na moju bakalársku prácu.

3 Popis metód a nástrojov tvorby informačných systémov

Pri písaní bakalárskej práce aplikujem viacero základných metód skúmania, akými sú analýza, syntéza a komparácia. Analýza a syntéza informácií boli využité najmä pri zisťovaní zariadení a pomôcok na pomoc pre nevidiacich. Komparácia sa využíva pri porovnávaní podobných alebo odlišných javov.

Vývoj informačného systému musí dodržiavať základné pravidlá životného cyklu. Tento životný cyklus informačného systému sa skladá z jednotlivých etáp. Začína sa analýzou súčasnej situácie. V ďalšej etape dochádza ku špecifikácii požiadaviek. Nasleduje návrh riešenia, pričom tieto prvé tri etapy sa realizujú bez programovania. Potom sa vyvíja systém, teda sa programuje, na čo priamo nadväzuje testovanie. Po týchto dvoch fázach sa dostáva produkt ku koncovým používateľom.

Hlavnej problematike, ktorou je pomoc ľuďom so zrakovým postihnutím zameraná na čítanie textov, sa venovalo už mnoho softvérových firiem a vývojárov. Ich riešenie predstavujú pre tvorbu mojej mobilnej aplikácie konkurenciu, no na druhej strane sú dôležité z hľadiska vyvarovania sa chýb. Pomáhajú lepšiemu pochopeniu logiky aplikácie, samotnému dizajnu a tiež sa dá vyčítať hlavný algoritmus týchto aplikácií. Najdôležitejšie je byť lepší ako konkurencia, odlišovať sa od ostatných riešení, a tým vytvoriť kvalitnejšiu mobilnú aplikáciu.

3.1 Výber pomôcok a nástrojov, ktoré sa použijú v aplikačnom zariadení

V dnešnej dobe je vlastníctvo mobilného telefónu bežnou záležitosťou. Preto je potrebné vykonať výber mobilných telefónov podľa používania a rozšírenia na trhu. To priamo nadväzuje na výber operačného systému mobilného telefónu. Pri vývoji aplikácii, prichádza do úvahy použitie viacerých programovacích jazykov. Dôležité je vybrať z nich ten najvhodnejší, ktorý ponúkne plánované výsledky. Výmena informácií a skúseností umožňuje používanie už predpripravených existujúcich podsystémov, ktoré možno zahrnúť do samotného vývoja. Jednotlivé výbery sú roztriedené do samostatných kategórií.

3.1.1 Výber mobilných zariadení

Pri výbere mobilných zariadení zohráva dôležitú úlohu zvolenie si operačného systému, pre ktorý chceme aplikáciu navrhnuť. Keďže operačný systém Windows Phone nie je veľmi rozšírený, tak sa v porovnaní neuvádza. V nasledujúcej tabuľke sa nachádza porovnanie jednotlivých mobilných operačných systémov.

Tabuľka 3. Analýza operačných systémov[autor]

Vlastnosť / OS	Android	iOS
zdrojový kód	open source	uzatvorený
OS rodina	Linux	OS X, UNIX
prispôbitelnosť	veľká, môže sa meniť skoro všetko	obmedzená
vývojár	Google, Open Handset Alliance	Apple Inc.
dostupné jazyky	100+	34
prenos súborov	ľahší ako pri iOS, pomocou USB kľúča	zložitý proces pomocou desktopovej aplikácie iTunes
dostupnosť	veľký počet telefónov a tabletov najmä Samsung, LG, Sony	iPhone, iPad, Apple TV, iPod
internetový prehliadač	Google Chrome	Mobile Safari
obchod s aplikáciami	Google Play	Apple App Store
hlasové príkazy	Google Assistant	Siri
životnosť batérie	vyššia	nižšia, skôr sa vybíja
správca súborov	áno (Stock Android File Manager)	nie je k dispozícii
rozhranie	dotyková obrazovka	dotyková obrazovka

Na základe vytvoreného prehľadu som si vybral operačný systém Android aj napriek slabšiemu dizajnu oproti iOS. Kľúčovým bodom bola pre mňa cieľová skupina ľudí, ktorým bude aplikácia určená a tiež kompatibilita zariadenia. Prioritnou vlastnosťou pri výbere bola aj životnosť batérie, ktorá predurčuje operačný systém Android.

Veľkou výhodou Androidu je množstvo zariadení, ktoré ho využívajú. Používatelia iOS majú k dispozícii oveľa menej možností ako majú používatelia Androidu. Cenová úroveň vedie k tomu, že väčšina ľudí z rôznych kútov sveta používa Android, pretože nie všetci ľudia sú ochotní platiť vysoké ceny za produkty od spoločnosti Apple. Možnosť výberu z množstva Android zariadení je výhodou pre tých používateľov, u ktorých smartfón odzrkadľuje ich osobnosť. Základným prvkom prispôsobenia Android telefónu sú widgety, ktoré sa okrem funkčnej stránky podieľajú aj na estetikej stránke zariadenia. Stále viac a viac populárnymi sa stávajú alternatívne launchery, ktoré sa môžu neustále meniť a sú k dispozícii v neobmedzenom množstve. Veľmi zaujímavým faktom je existencia väčšieho množstva hier a služieb zdarma oproti tým v iOS. Často sa stáva, že aplikácia ktorá je v operačnom systéme iOS platená, je v Androide zdarma [15].

Pred tvorbou Android aplikácie je dôležité poznať základnú architektúru Androidu, ktorá sa delí do štyroch častí. Prvá, vrchná časť je aplikačná, pod ňou sa nachádza stredná vrstva nazývaná middleware, ktorá pozostáva z aplikačných frameworkov, knižníc a Android Runtime. Spodná časť obsahuje Linuxové jadro. Každú z týchto častí som bližšie špecifikoval.

Používatelia sú charakterizovaní vysokým záujmom o aplikácie. Android neustále dodáva na trh veľké množstvo užitočných aplikácií. Medzi hlavné užitočné aplikácie patrí Prehliadač, Kontakty, Správy a mnoho ďalších. Všetky aplikácie sú naprogramované v programovacom jazyku Java.



Obrázok 21. Aplikačná vrstva[16]

Aplikačný Framework je súbor vysoko-úrovňovej tvorby blokov na vytváranie aplikácií. Na Android zariadeniach je tento framework predinštalovaný a pozostáva z nasledujúcich komponentov:

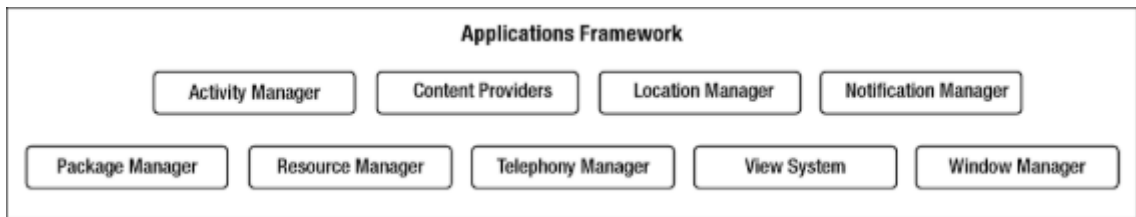
- **Activity Manager:** poskytuje životný cyklus aplikácie a udržiava zdieľaný súbor aktivít na navigáciu v rámci aplikácii a medzi aplikáciami
- **Content Providers:** zapuzdruje dáta, ktoré môžu byť zdieľané medzi jednotlivými aplikáciami
- **Location Manager:** je zodpovedný za fyzickú lokáciu zariadenia
- **Notification Manager:** jeho činnosť zahŕňa informovanie používateľa o rôznych udalostiach a skutočnostiach vo forme notifikácií
- **Package Manager:** umožňuje aplikácii aby sa dozvedela o ďalších aplikačných balíčkoch nainštalovaných na zariadení
- **Resource Manager:** zabezpečuje prístup k zdrojom aplikácie
- **Telephony Manager:** zodpovedá za telefónne služby zariadenia a tiež za prijímanie a vytváranie hovorov
- **View System:** spravuje jednotlivé prvky používateľského rozhrania a generuje udalosti orientované na používateľské rozhranie

Window Manager: organizuje reálny odhad obrazovky v oknách, priradzuje miesta na kreslenie a ďalšie úlohy súvisiace s oknami

Komponenty Aplikačného Frameworku pri vykonávaní svojich funkcií sú závislé od knižníc programovacieho jazyka C/C++. Vývojári využívajú najmä nasledovné knižnice:

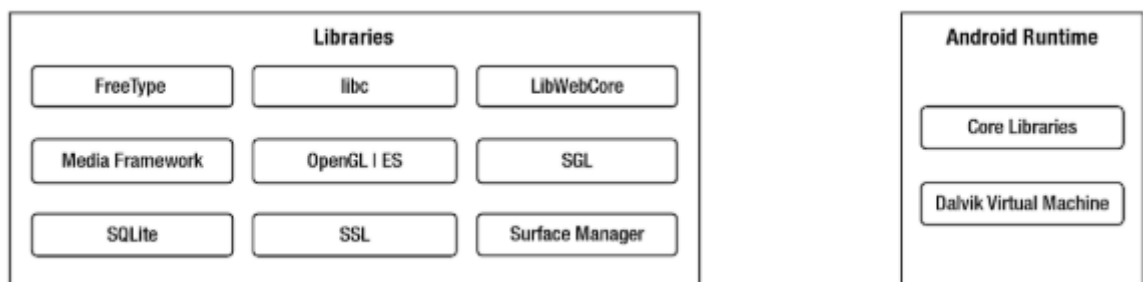
- **FreeType:** táto knižnica umožňuje vykresľovanie bitmapového a vektorového písma.

- **libc**: ide o BSD derivovanú implementáciu štandardnej systémovej knižnice programovacieho jazyka C.



Obrázok 22. Aplikačný Framework[16]

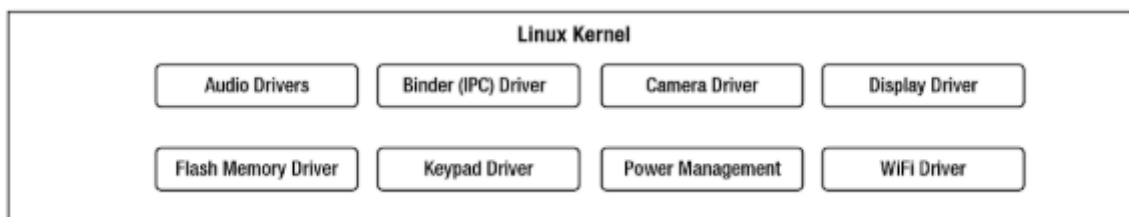
- **LibWebCore**: ponúka moderný a rýchly webový prehliadač, ktorý ovláda Android prehliadač.
- **Media Framework**: podporuje nahrávanie a prehrávanie rôznych audio a video formátov, ako aj prácu so statickými obrázkovými súborami. Medzi podporované formáty patrí napríklad MPEG4, JPEG, PNG a GIF.
- **OpenGL | ES**: sú to 3D grafické knižnice poskytujúce OpenGL implementáciu založenú na OpenGL ES 1.0/1.1/2.0/3.0 aplikáciách.
- **SGL**: poskytuje podkladový 2D grafický engine.
- **SQLite**: zabezpečuje relačný databázový engine, ktorý je dostupný všetkým aplikáciám.
- **SSL**: úlohou tejto knižnice je zabezpečiť ochranu socketov a ochranu internetovej komunikácie.
- **Surface Manager**: umožňuje zobrazenie podsystému, ale aj kompozitné 2D a 3D grafické vrstvy z viacerých aplikácií.



Obrázok 23. Knižnice C/C++[16]

Android poskytuje Runtime Environment, ktorý pozostáva z hlavných knižníc a z Dalvik Virtual Machine. Každá aplikácia od Androidu beží v jeho vlastnom Linuxovom

proces, v ktorom je umiestnená inštancia Dalvik. Tento virtuálny stroj bol navrhnutý takým spôsobom, aby zariadenia mohli efektívne pracovať s viacerými virtuálnymi strojmi [16].



Obrázok 6. Linux Kernel[16]

3.1.2 Výber programovacích jazykov

Mobilné aplikácie sa dajú programovať dvoma spôsobmi. Prvý je založený na vytvorení aplikácie pomocou jazyka HTML5, ktorá bude kompatibilná pre akékoľvek zariadenie. Druhá zahŕňa programovanie aplikácie špecifickej pre jednotlivé platformy, buď Android alebo iOS. Vychádzajúc zo skutočnosti, že som sa rozhodol pre operačný systém Android, pripadajú do úvahy programovacie jazyky Java, C++ a C#.

Pre moju bakalársku prácu som si vybral Javu, objektovo orientovaný jazyk od spoločnosti Oracle. Veľkou výhodou je fakt, že zdrojové programy sa nekompilujú do strojového kódu, ale do medzistupňa, tzv. „byte-code“, ktorý nie je závislý od konkrétnej platformy. Ďalšou významnou črtou Javy je veľké množstvo balíkov, a tiež jej popularita vo sfére programovania.

Pred výberom programovacieho jazyka Java bolo potrebné vykonanie porovnania programovacieho jazyka Java s konkurenčnými programovacími jazykmi. Prvá tabuľka obsahuje porovnanie Javy s jazykom C++. Komparácia bola zameraná na vlastnosti ako sú prenositeľnosť, správa pamäte, dedenie a ďalšie [17].

Tabuľka 4. Porovnanie programovacích jazykov Java a C++[autor]

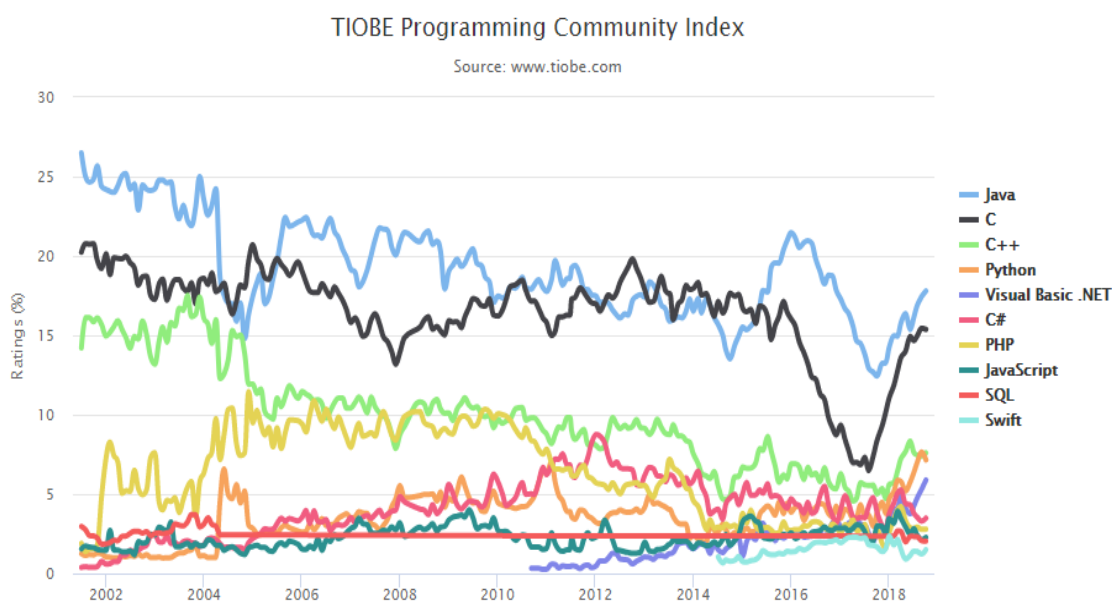
Programovací jazyk	Java	C++
Správa pamäte	nepoužíva ukazovatele, podporuje vlákna a rozhrania	používa ukazovatele, podporuje štruktúry a unióny
Dedenie	nepodporuje viacnásobnú dedičnosť, používa koncept rozhrania	podporuje jednoduchú a viacnásobnú dedičnosť
Mechanizmus zisťovania chýb pri spustení	systémová zodpovednosť	programátorská zodpovednosť
Knižnice	poskytuje širokú škálu tried pre rôzne služby na vysokej úrovni	dostupné s nízko úrovňovou funkcionalitou
Sémantika dátových typov	rozdielnosť medzi primitívnymi a objektovými typmi	podpora medzi primitívnymi a objektovými typmi
Prenositeľnosť	používa koncept bytecode, ktorý je nezávislý na platforme a môže byť použitý s platformou JVM	platformová závislosť, zdrojový kód musí byť kompilovaný pre inú platformu

Polymorfizmus	automaticky, používa statickú a dynamickú väzbu	explicitne pre metódy podporuje zmiešané hierarchie
---------------	---	---

Druhá tabuľka je zameraná na porovnanie Javy s programovacím jazykom C#. Obsahuje analýzu vlastností oboch programovacích jazykov ako sú koncepty na ktorých sú postavené, aký typ aplikácie sa pomocou nich vytvára, kde sa najčastejšie používajú, čiže na aký účel a ďalšie [18].

Tabuľka 5. Porovnanie programovacích jazykov Java a C#[autor]

Programovací jazyk	Java	C#
Paradigma	objektovo orientovaný jazyk, založený na triedach, odvodený z jazyka C++	objektovo orientovaný jazyk, založený na komponentoch a funkciách
Aplikácia	komplexná webová aplikácia, veľmi súbežná	webové aplikácie a herný priemysel, obľúbený pre mobilný vývoj
Projekt	vhodný pre komplexný, súbežný, webový projekt	najvhodnejší pre projekty herného vývoja
Použitie	správy, webové aplikácie, vysoko súbežné aplikácie	hry, mobilný vývoj, virtuálna realita
Inštalácia	vyžaduje JDK (Java Development Kit) pre spustenie Javy	.NET framework poskytuje obrovskú knižnicu kódov používaných v C#
Nástroje	Eclipse, NetBeans, IntelliJ IDEA	Visual Studio, MonoDevelop, #develop



Obrázok 25. Graf populárnosti programovacích jazykov[48]

Ďalším dôležitým krokom pri výbere programacieho jazyka je aj popularita medzi používateľmi. Do úvahy pripadá graf od spoločnosti TIOBE, ktorý sa nazýva TIOBE Programming Community Index. Tento graf je indikátorom popularnosti programovacích jazykov. Pri detailnejšom pohľade na tento graf môžeme vidieť jeden z dôvodov, prečo som si vybral programovací jazyk Java.

3.1.3 Výber existujúcich podsystémov

Vývojové prostredie, v ktorom bude vyvíjaná mobilná aplikácia sa tiež rozlišuje v závislosti od operačného systému, pre ktorý je aplikácia určená. Xcode8 je vývojové prostredie od spoločnosti Apple, ktoré podporuje programovacie jazyky C, C++, Java, Python a ďalšie. Pre náš Android je určené vývojové prostredie Android Studio založené na IntelliJ IDEA, ktoré vyžaduje mať nainštalovanú originálnu Javu od Oracle. Android Studio ponúka:

- integrované nástroje pre profiling
- testovanie kompatibility
- podpisovanie aplikácií
- wizardy založené na šablónach
- editor s možnosťou navrhovať užívateľské rozhranie metódou drag-and-drop



Obrázok 7. Úvodná obrazovka Android Studio[autor]

Android Studio je určené na používanie vývojovými tímami, bez ohľadu na to, či ide o jednotlivcov alebo globálne tímy. Obsahuje podporu pre verzionovací systém Git, ktorá umožňuje prepojenie projektu do repozitáru v Gite. Workflow pre Android Studio je postavený na koncepte kontinuálnej integrácie, ktorá umožňuje programátorom testovať svoj kód vždy, keď si kontrolujú svoju prácu. Tento koncept priebežného kontrolovania kódu poskytuje vývojárom spätnú väzbu s cieľom rýchlejšie nahrat' verzie mobilného riešenia od obchodu Google Play App Store. Taktiež podporuje Google App Engine, ktorý slúži na rýchlu cloudovú integráciu nových aplikácií a funkcií. Vývojové prostredie si môže stiahnuť každý, a používať zadarmo na akejkoľvek platforme. Android Studio poskytuje nástroje na vybudovanie riešení pre telefóny a tablety, ako aj pre nové technológie ako sú Android TV, Android Wear, Android Auto a mnoho ďalších.

3.2 Využívanie OCR v mobilných aplikáciách

OCR alebo Optické rozpoznávanie znakov je metóda, ktorá slúži na preklad textu z obrázka do textovej podoby, napr. do ASCII znakov abecedy. V minulosti sa na rozoznávanie znakov využívali rôzne metódy, no v súčasnosti sa používa výhradne výpočtová technika. OCR softvér obsahuje prvky umelej inteligencie a strojového videnia, ako aj lingvistické prvky, ktoré kontrolujú správnosť preložených slov a viet. Presnosť výstupu je determinovaná zložitou použitým softvérom. Pre OCR softvér je dôležité nastaviť jazyk, v ktorom je text napísaný, aby správne vyhodnotil diakritické znaky a aby pre výstupný text použil správny jazykový slovník. Známymi OCR softvérmi sú: ABBYY Real-Time Recognition SDK, Tesseract OCR, IRIS Mobile OCR SDK a Mobile Vision API.

3.2.1 ABBYY Real-Time Recognition SDK

Umožňuje vývojárom pridávať do mobilných aplikácií funkciu okamžitého zachytávania textu. Údaje o platbách na faktúrach, osobné údaje na identifikačných kartách, kódy vytlačené na bežných tovaroch ako aj výstražné štítky je možné okamžite zachytiť a použiť v mobilnej aplikácii, používateľ musí len zaostriť kameru na text, ktorý chce rozpoznať. Poskytuje možnosť rozpoznávania 63 cudzích jazykov.

Medzi znaky zachytávania textu v reálnom čase patrí:

1. **rýchle zadávanie textu** – k extrakcii textu dochádza priamo z obrazovky smartfónu, je to oveľa rýchlejšie ako snímanie obrázku, po ktorom nasleduje aplikovanie OCR alebo manuálne zadávanie textu.
2. **vysoké bezpečnostné požiadavky** – v dôsledku toho, že text nie je potrebné fotografovať, ukladať do pamäte, ukladať do cloudu alebo externých serverov, tento spôsob zachytávania informácií je vhodný pre organizácie s najvyššími štandardmi bezpečnosti údajov.
3. **zlepšenie výkonu podniku** – zachytávanie textu v reálnom čase z dokumentov ako sú faktúry, nákladové listy a ďalšie, možno rýchlo implementovať do firemných aplikácií v bankách, dopravných spoločnostiach, poisťovniach a iných organizáciách.
4. **vynikajúca presnosť rozpoznávania** – ide o spojenie osvedčených technológií ABBYY s novými inteligentnými algoritmami, ktoré rozpoznávajú text zo živých video streamov [33].

3.2.2 Tesseract OCR

Je to open-source OCR knižnica, vyvíjaná spoločnosťou HP na rozpoznávanie textov z obrázka. Proces rozpoznávanie prechádza dvoma krokmi. Prvý krok predstavuje pokus o rozpoznanie každého slova. Všetky slová, ktoré sú uspokojivé sa odovzdávajú adaptívnemu klasifikátoru ako tréningové údaje. V druhom kroku dochádza k prechádzaniu obrázka znova, a slová, ktoré neboli doposiaľ dostatočne dobre rozpoznané sa znova rozpoznávajú. Finálna fáza rieši fuzzy medzery a kontroluje alternatívne hypotézy pre najmenšie texty.

Tesseract sa používa na budovanie mnohých aplikácií pre Android, ako je skenovanie vizitiek, importovanie textu do zoznamu kontaktov, skenovanie obrazu a prehratie nahratého textu pre zrakovo postihnutých. Vyššie spomenutý klasifikátor nevie pracovať s poškodenými znakmi, pretože na poškodené znaky nebol vyškolený. Klasifikátor bol v realite vyškolený na dvadsiatich vzorkách z 94 znakov, čo predstavuje rozpoznávanie ôsmich fontov v jednej veľkosti, ale so štyrmi atribútmi.

Vždy keď modul rozpoznávania slov zvažuje novú segmentáciu, jazykový modul vyberie najlepší dostupný reťazec slov v každej z nasledujúcich kategórií: najviac používané slová, čísla, adresáre, slová s veľkými písmenami a malými písmenami. Konečné rozhodnutie pre danú segmentáciu je slovo s najnižším hodnotením celkovej vzdialenosti, kde každá z vyššie uvedených kategórií sa vynásobí inou konštantou. V súčasnosti patrí knižnica Tesseract OCR medzi najpresnejšie komerčné nástroje na rozpoznávanie textov z obrázka. Najväčšou silnou stránkou je veľký výber funkcií. Kľúčovou slabinou je použitie polygonálnej aproximácie ako vstup pre klasifikátor [34].

3.2.3 IRIS Mobile OCR SDK

Vývojári môžu jednoducho integrovať zobrazovacie funkcie a schopnosti OCR do ich mobilných aplikácií. IRIS OCR SDK umožňuje vytvárať vlastné mobilné zobrazovacie aplikácie na konverziu všetkých obrázkov a fotografií do indexovaných a upravovateľných dokumentov pre operačné systémy iOS a Android. Na konverziu dát, zachytávanie a získavanie dokumentov, čítanie a prekladanie textov je k dispozícii viac ako 137 jazykov.

Medzi kľúčové znaky patrí:

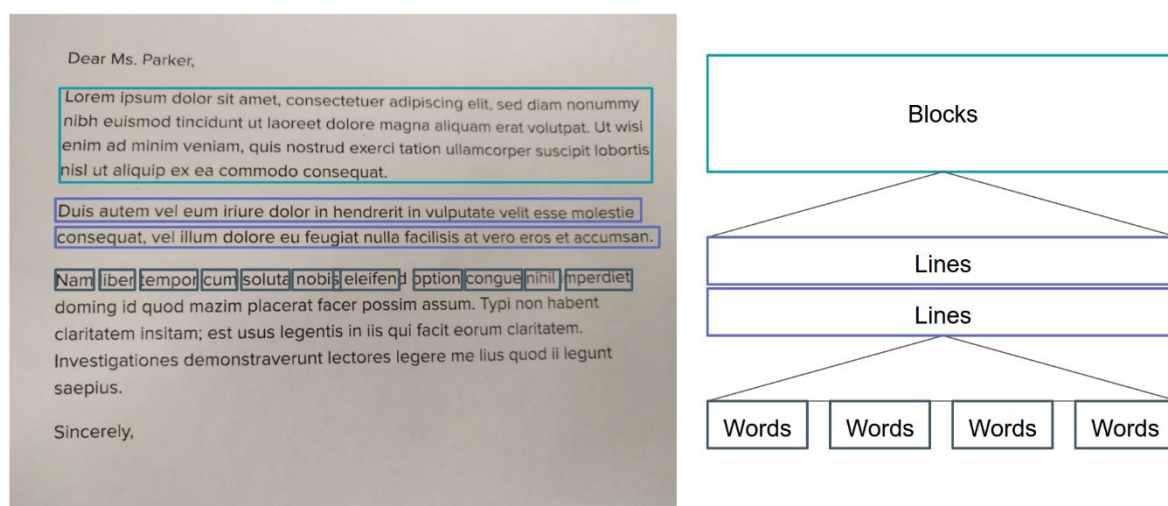
1. **zobrazovací modul** – zachytávanie snímok a obrázkov automaticky pomocou mobilného zariadenia. Načítavanie a ukladanie obrázkov v BMP, PNG, PDF a JPEG.

2. **modul predspracovania** – automatická detekcia okrajov a 3D korekcia na vyrovnanie obrazu. Rôzne optimalizačné funkcie ako je adaptívna binarizácia a rotácia dokumentov. Umožňuje odstrániť tmavé okraje, odstránenie riadkov a výpadok farieb je k dispozícii v samostatnom rozšírení.
3. **rozpoznávanie textu** – veľmi rýchle a presné optické rozpoznávanie znakov vo viac ako 137 jazykoch, vrátane ázijských jazykoch.
4. **výstup dokumentu** – formáty výstupu sú PDF, HTML, TXT, CSV a ďalšie. Tiež obsahuje kompresný modul na komprimovanie súborov [32].

3.2.4 Mobile Vision API

Mobile Vision poskytuje knižnicu na vyhľadávanie objektov nachádzajúcich sa na fotkách a videách. Táto knižnica obsahuje detektory, ktoré lokalizujú a opisujú vizuálne objekty v snímkach a videách. V súčasnosti obsahuje detektor tváre, skener čiarových kódov, textové detektory, ktoré môžu byť použité spolu alebo samostatne. Aplikácie môžu dokonca sledovať skutočné objekty, napríklad čítanie čísel zastávok vo vlaku.

Rozpoznávanie textu pomocou Mobile Vision API rozdeľuje text do blokov, riadkov a slov. Blok je súvislý set riadkov, napríklad paragraf. Riadok je súvislý set slov na rovnakej vertikálnej úrovni. Slovo je súvislý súbor alfanumerických znakov, ktoré sú na tej istej vertikálnej osi. Na nasledujúcom obrázku je znázornené vysvetlenie toho, ako pracuje rozpoznávanie textu, označované ako Text Recognizer [35].



Obrázok 41. Textová štruktúra rozpoznávania textu[35]

4 Návrh aplikačného zariadenia a implementácia

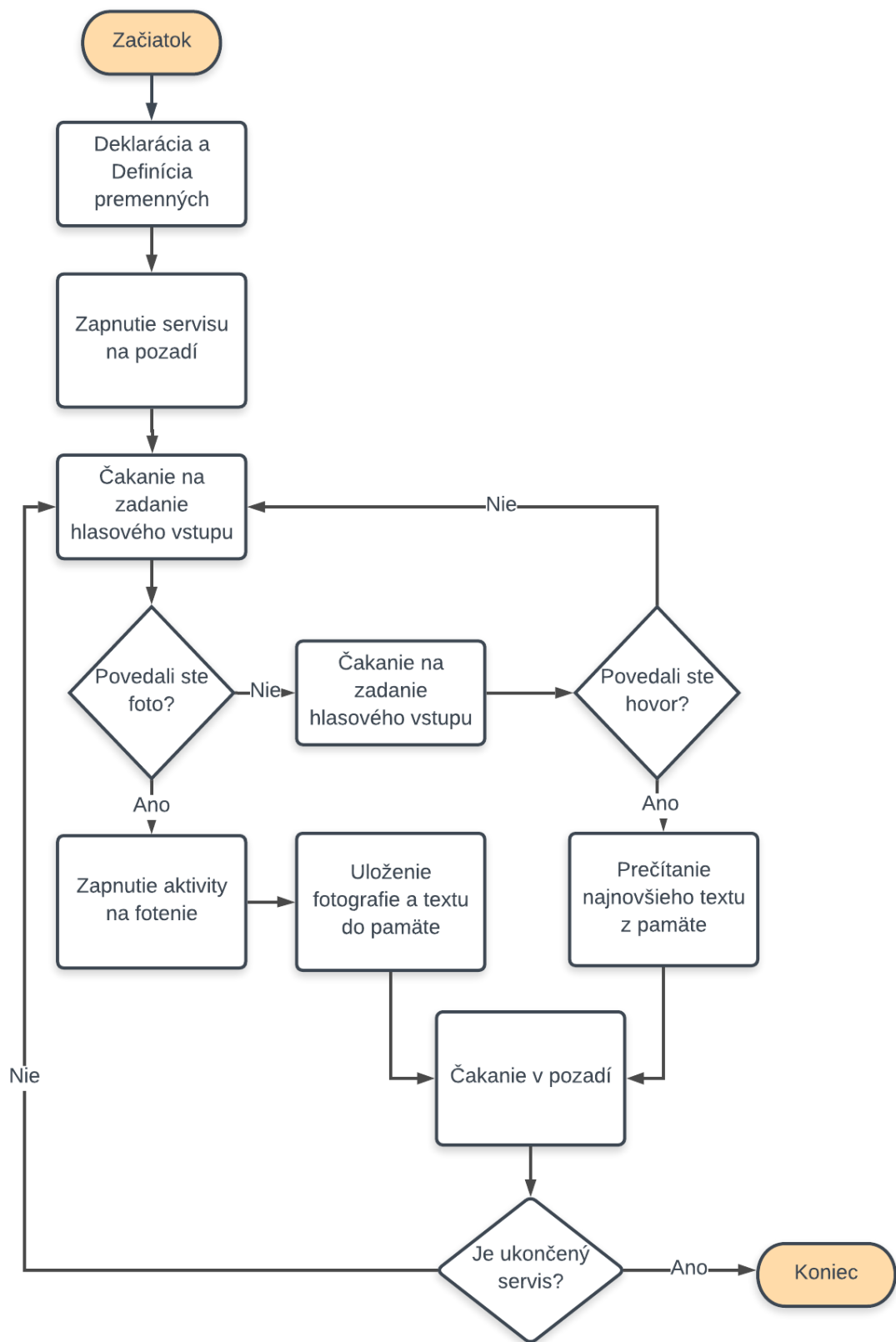
Táto časť práce je zameraná na návrh a tvorbu aplikácie a zároveň na detailnejšie popísanie knižníc ktoré boli implementované a tvoria hlavnú časť logiky aplikácie. Je tu popísaný celý postup tvorby aplikácie, jednotlivé komponenty a technológie, ktoré boli použité a tiež algoritmus činnosti aplikácie. V záverečnej časti je poskytnutý náhľad, ako vyzerá finálna aplikácia. Okrem popísania celého postupu vývoja, je vytvorená aj dokumentácia, ktorá slúži ako manuál k danej aplikácii. Tento manuál má byť nápomocný, a informačný, má pomáhať pri prvotnom styku človeka s mobilnou aplikáciou.

Úlohou predchádzajúcej kapitoly bolo vytvorenie prehľadu o jednotlivých mobilných operačných systémoch, programovacích jazykoch a o existujúcich podsystémoch. Na základe tohto prehľadu som pre vývoj vlastnej aplikácie vybral programovací jazyk Java, mobilný operačný systém Android a vývojové prostredie na tvorbu aplikácie Android Studio. Každý výber bol sprevádzaný zdôvodnením a vysvetlením, prečo pre tvorbu našej aplikácie najviac vyhovuje, respektíve poskytne najlepšie výsledky.

Aplikácia je určená pre nevidiacich a slabozrakých ľudí, čo znamená maximálne uspokojenie požiadaviek tejto skupiny ľudí. Je dôležité brať do úvahy fakt, že títo ľudia budú využívať mobilnú obrazovku len minimálne. Najdôležitejšou časťou aplikácie je zvuk a kamera telefónu. Pomocou nich sa dá poskytnúť ľuďom so zrakovým postihnutím pomoc pri bežných každodenných činnostiach.

4.1 Algoritmus činnosti zariadenia

Programovanie Android aplikácií je založené na tvorbe aktivít. Každá aktivita je zodpovedná za vykonanie nejakej akcie alebo viacerých akcií, pričom z jednej aktivity sa môže volať ďalšia aktivita. Takisto platí, že jedna aktivita môže posilať ďalšej aktivite dodatočné informácie. Vo svojej podstate je algoritmus postupnosť dobre definovaných inštrukcií na splnenie určitej úlohy. Algoritmické postupy sa dajú znázorniť mnohými spôsobmi. Grafické znázornenie algoritmu úlohy sa nazýva vývojový diagram. Vývojové diagramy sú určené na popísanie algoritmu úlohy pomocou grafických značiek. Na popísanie činnosti mojej aplikácie som zostrojil vývojový diagram. Takisto ako každý diagram má svoj začiatok a koniec. Tento diagram bol vytvorený v programe Lucidchart, ktorý je dostupný online. Je to riešenie, určené na vizuálnu komunikáciu a zostrojovanie vývojových diagramov, procesných máp, UML diagramov a iných grafov. Na nasledujúcom obrázku sa nachádza môj vývojový diagram, popisujúci činnosť mobilnej aplikácie.



Obrázok 52. Algoritmus činnosti zariadenia[autor]

Algoritmus aplikácie pozostáva z viacerých, do seba zapadajúcich častí, tzv. aktivít:

- **MainActivity** – je prvá aktivita, ktorá sa spúšťa po inštalácii aplikácie. Je zodpovedná za spustenie a ukončenie servisu, ktorý beží na pozadí. Dochádza v nej ku kontrole oprávnenia prístupu ku kamere mobilného telefónu a tiež k zapisovaniu do externej pamäte mobilného telefónu. Najdôležitejšia metóda v tejto aktivite je `onClickStartService`. Je zodpovedná za zapnutie, respektíve spustenie servisu, pričom jej definícia vyzerá nasledovne.

```
//zapnutie servisu
public void onClickStartService(View V) {
    //start servisu
    Intent i = new Intent( packageContext: MainActivity.this, Recorder.class);
    startService(i);
}
```

Obrázok 27. Metóda spustenia servisu na pozadí[autor]

Metóda `onClickStopService` ukončuje servis na pozadí. V `AndroidManifest` súbore je určené, že táto trieda sa spustí ako prvá. V každom Android projekte musí byť práve jedna trieda označená na to, aby sa mohla spúšťať ako prvá.

- **OcrActivity** – je hlavná aktivita na ktorej je postavená logika celej aplikácie. Má zodpovednosť za získanie textu z nasnímaného obrázka. Na zabezpečenie tejto činnosti využíva knižnicu `Mobile Vision API`. Ďalšou funkciou ktorú zabezpečuje je uloženie nasnímaného obrázka do externej pamäte mobilného telefónu, aby sa v prípade nezrovnalostí, na tento obrázok mohol pozrieť človek bez zrkovitého postihnutia. Obsahuje metódu `takeImage`, ktorá je zodpovedná za odfotenie obrázka a jeho uloženie do externej pamäte mobilného zariadenia. Pri ukladaní fotky som zistil, že dochádza k rotácii obrázka. Musel som preto rotovať obrázok pred jeho uložením o 90 stupňov, aby obrázok bol v takom tvare, ako bol aj naozaj odfotený. Celý algoritmus ukladania obrázka do externej pamäte telefónu sa nachádza na obrázku (Obrázok 28.).

Ďalej sa v tejto aktivite kontroluje to, či je povolený prístup ku kamere telefónu. Na to aby kamera telefónu fotila automaticky, bez akéhokoľvek klikania je potrebné nastaviť nášmu `CameraSource` niekoľko parametrov, vid' na obrázku (Obrázok 29.). Najdôležitejším sa javí nastavenie `setAutoFocusEnabled` na `true`, ktorý je zodpovedný za automatické fotenie bez kliknutia. Automatické fotenie pomáha ľuďom so zrkovým postihnutím, aby mohli aj oni vytvárať fotky, bez nutnosti stláčania tlačidiel telefónu.

```

// konvertuj pole bytov na bitmap
Bitmap loadedImage = null;
Bitmap rotatedBitmap = null;
loadedImage = BitmapFactory.decodeByteArray(bytes, offset: 0, bytes.length);

// zabezpeč rotácia obrázka
Matrix rotateMatrix = new Matrix();
rotateMatrix.postRotate( degrees: 90);
rotatedBitmap = Bitmap.createBitmap(loadedImage, x: 0, y: 0,
    loadedImage.getWidth(), loadedImage.getHeight(),
    rotateMatrix, filter: false);
String state = Environment.getExternalStorageState();
File folder = null;
if (state.contains(Environment.MEDIA_MOUNTED)) {
    folder = new File( pathname: Environment
        .getExternalStorageDirectory() + "/Samuel_Pictures");
}

boolean success = true;
if (!folder.exists()) {
    success = folder.mkdirs();
}

String name = new SimpleDateFormat( pattern: "yyyyMMdd_HHmms").format(new Date());

if (success) {
    imageFile = new File( pathname: folder.getAbsolutePath()
        + File.separator
        + name
        + ".jpg");

    imageFile.createNewFile();
}

```

Obrázok 28. Uloženie obrázka do externej pamäte[autor]

```

//Inicializacia CameraSource na pouzivanie vysokeho rozlisenia a nastavit Autofocus na On.
mCameraSource = new CameraSource.Builder(getApplicationContext(), textRecognizer)
    .setFacing(CameraSource.CAMERA_FACING_BACK)
    .setRequestedPreviewSize( w: 1280, h: 1024)
    .setAutoFocusEnabled(true)
    .setRequestedFps(2.0f)
    .build();

```

Obrázok 29. Nastavenie CameraSource[autor]

Ku detekcii, respektíve rozpoznaniu textu z obrázka za použitia OCR technológie dochádza v metóde `receiveDetections`. Veľmi dôležitým krokom na efektívne ukladanie textu z obrázka je použitie dátového typu `StringBuilder` pred obyčajným `String`om, pretože `StringBuilder` disponuje metódou `append`, ktorá slúži na vkladanie ďalších reťazcov. Na obrázku (Obrázok 30.) sa nachádza celá metóda `receiveDetections`, ktorá prijíma jeden parameter.

Treba dbať na to, aby sme na konci metódy zavolať na našu inštanciu `stringBuilder` metódu `toString`, ktorá zabezpečí, že výsledok bude typu `String`, a môže sa s ním ďalej pracovať ako s obyčajným reťazcom. Ak by som nezavolať túto metódu na pretypovanie premennej, dostal by sme chybu, a aplikácia by sa nedala skompilovať. AndroidStudio je

natol'ko inteligentné, že tento kúsok kódu by označilo červenou farbou, čo by znamenalo nutnosť prepísať alebo upraviť kód.

```
@Override
public void receiveDetections(Detector.Detections<TextBlock> detections) {
    final SparseArray<TextBlock> items = detections.getDetectedItems();
    if (items.size() != 0){

        mTextView.post(new Runnable() {
            @Override
            public void run() {
                StringBuilder stringBuilder = new StringBuilder();
                for(int i=0;i<items.size();i++){
                    TextBlock item = items.valueAt(i);
                    stringBuilder.append(item.getValue());
                    stringBuilder.append("\n");
                }
                mTextView.setText(stringBuilder.toString());
                writeExternalStorage(mCameraView);
                takeImage();
            }
        });
    }
}
```

Obrázok 30. Metóda *receiveDetections*[autor]

- **VoiceActivity** – ako už samotný názov aktivity hovorí, ide o aktivitu, ktorá je zodpovedná za hlasový výstup, čiže za prečítanie textu získaného z nasnímaného obrázka. Hlasový výstup sa dá nastaviť na rôzne jazyky. Nastavenie jazyka sa realizuje tak, že sa vytvorí premenná typu *Locale*. *Locale* je Android vstavaná trieda, ktorá obsahuje konštanty pre viaceré svetové jazyky. Jednou z konštánt je aj náš slovenský jazyk, ktorý som si vybral a použil v aplikácii. Nastavenie jazyka je znázornené na nasledujúcom obrázku.

```
Locale loc = new Locale ( language: "sk", country: "SK");
textToSpeech.setLanguage(loc);
```

Obrázok 31. Nastavenie slovenského jazyka pre hlasový výstup[autor]

Metóda, ktorá zabezpečuje hlasový výstup sa nazýva *TextToSpeechFunction*. Algoritmus tejto metódy je realizovaný tak, že do reťazca sa uloží text, ktorý som si uložil do textového súboru. Využívam pri tom pomocnú triedu *FileHelper* a jej metódu *readExternalStorage*. Následne používam na vytvorenú inštanciu typu *TextToSpeech* funkciu *speak*, ktorá vykoná hlasový výstup. Funkcia *speak*, prijíma tri parametre a to text zo súboru, ktorý som uložil do reťazca, konštantu typu *TextToSpeech* *QUEUE_FLASH*, ktorá je zodpovedná za plynulý chod hlasového výstupu, a tretí parameter je voliteľný, a slúži na zabezpečenie toho, aby sa vytvorilo unikátne id. Jedinečný identifikátor slúži na zabezpečenie plynulého chodu aplikácie. Celá metóda sa nachádza na obrázku.

```

public void TextToSpeechFunction() {
    HashMap<String, String> map = new HashMap<String, String>();
    map.put(TextToSpeech.Engine.KEY_PARAM_UTTERANCE_ID, "UniqueID");

    String textholder = (FileHelper.readExternalStorage());
    textToSpeech.speak(textholder, TextToSpeech.QUEUE_FLUSH, map);
    txtContent.setText(textholder);
}

```

Obrázok 32. TextToSpeech funkcia[autor]

Táto aktivita sa vykoná vtedy, keď je zadaný hlasový vstup pomocou kľúčového slova hovor. Vyvolanie aktivity nevedie k spusteniu používateľského rozhrania, pretože všetko sa deje na pozadí. Zabezpečujú to tieto riadky môjho kódu.

```

Intent intent1 = new Intent(Intent.ACTION_MAIN);
intent1.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
intent1.addCategory(Intent.CATEGORY_HOME);
startActivity(intent1);

```

Obrázok 33. Zabránenie spusteniu používateľského rozhrania[autor]

- **FileHelper** – je pomocná trieda, ktorá je určená na čítanie textu z textového súboru. Tento textový súbor obsahuje údaje, ktoré boli prekonvertované pomocou OCR technológie. V tejto triede sa nachádzajú dve metódy. Prvá je určená na čítanie z externej pamäti a druhá je určená na vybratie najnovšieho súboru z externej pamäti mobilného zariadenia. Na čítanie z externej pamäte sa používa jednoduchý algoritmus znázornený na obrázku.

```

public static String readExternalStorage() {
    File root = Environment.getExternalStorageDirectory();
    File Dir = new File( pathname: root.getAbsolutePath() + DIRECTORY);
    File file = lastFileModified(Dir.toString());
    String datafromfile;
    StringBuffer stringBuffer = new StringBuffer();

    try {
        FileInputStream fileInputStream = new FileInputStream(file);
        InputStreamReader inputStreamReader = new InputStreamReader(fileInputStream);
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

        while ((datafromfile = bufferedReader.readLine()) != null) {
            stringBuffer.append(datafromfile + "\n");
        }
    } catch (IOException e) {
        Log.e(TAG, e.getMessage());
    }

    return stringBuffer.toString();
}

```

Obrázok 34. Program na čítanie z externej pamäte[autor]

Celý algoritmus je uložený vo funkcii `readExternalStorage`. Na zabezpečenie toho, aby bol prečítaný najnovší pridaný súbor sa používa metóda s názvom `lastFileModified`, ktorá ako vstupný parameter prijíma `String`. Táto metóda má návratovú hodnotu typu `File` a je znázornená na nasledujúcom obrázku.

```
public static File lastFileModified(String dir) {
    File fl = new File(dir);
    File[] files = fl.listFiles(new FileFilter() {
        public boolean accept(File file) {
            return file.isFile();
        }
    });
    long lastMod = Long.MIN_VALUE;
    File choice = null;
    for (File file : files) {
        if (file.lastModified() > lastMod) {
            choice = file;
            lastMod = file.lastModified();
        }
    }
    return choice;
}
```

Obrázok 35. Program na vrátenie najnovšieho súboru[autor]

- **Recorder** – je trieda, ktorá je servisom, ktorý beží na pozadí. Celá trieda je veľmi rozsiahla, pretože obsahuje viaceré metódy, ktoré slúžia napríklad na vytvorenie notifikácie nachádzajúcej sa v top bare. To, že trieda `Recorder` je servisom, je definované aj v súbore `AndroidManifest` viď na nasledujúcom obrázku.

```
<service
    android:name=".Recorder"
    android:enabled="true" />
```

Obrázok 36. Recorder v `AndroidManifest` súbore[autor]

Na vytvorenie notifikácie, ktorá signalizuje, že naša aplikácia beží sa používa metóda `buildForegroundNotification`. Táto metóda je typu `Notification`, a jediné čo vracia je buildovaná notifikácia. Ukážka celej metódy sa nachádza na obrázku (Obrázok 37.).

Najdôležitejšou metódou celej triedy `Recorder` je `onResults`. V tejto triede dochádza k rozpoznávaniu hlasových vstupov. Aplikácia používa dve kľúčové slová, ktoré slúžia na vykonanie určitých akcií, ktoré sú rozpoznávané pomocou `SpeechRecognizera`, ktorý slúži na rozpoznávanie hlasu. Ak používateľ aplikácie povie: „foto“, tak sa spustí aktivita na odfotenie obrázka, pre možnosť lepšieho rozpoznania hovoreného slova bola pridaná aj kratšia verzia slova. Druhým slovom, ktoré je kontrolované či bolo povedané je: „hovor“, a tiež jeho kratšia verzia. Po rozpoznaní tohto slova sa spustí aktivita, ktorá prečíta prekonvertovaný text nachádzajúci sa v súbore. Trieda `Recorder` obsahuje metódu

onStartCommand, ktorá slúži na inicializáciu systémového servisu WakeLock. Jeho úlohou je zabezpečiť, aby procesor CPU vykonával svoju prácu, zatiaľ čo obrazovka je vypnutá.

```
//metody
//vytvorenie notifikacie
private Notification buildForegroundNotification(String filename) {
    //inicializacia buildera pre Notifikáciu
    NotificationCompat.Builder b = new NotificationCompat.Builder( context: this);
    //po kliknutí na ikonu notifikácie
    Intent resultIntent = new Intent( packageContext: this, MainActivity.class);
    PendingIntent resultPendingIntent =
        PendingIntent.getActivity(
            context: this,
            requestCode: 0,
            resultIntent,
            PendingIntent.FLAG_UPDATE_CURRENT
        );
    //zapnut notifikáciu
    b.setOngoing(true);
    //detaily notifikácie
    b.setContentTitle("Počúvanie...Pre ukončenie klikni")
        .setContentText(filename)
        .setSmallIcon(R.drawable.phonefinder)
        .setTicker("Počúvanie...Pre ukončenie klikni");
    b.setContentIntent(resultPendingIntent);

    //buildovanie notifikácie
    return(b.build());
}
```

Obrázok 37. Program na vytvorenie notifikácie[autor]

- **MyBroadcastReceiver** – táto trieda dedí z triedy BroadcastReceiver. Je zodpovedná za to, že po bootovaní mobilného zariadenia, respektíve po jeho reštarte, sa automaticky spustí moja mobilná aplikácia. Obsahuje metódu onReceive, ktorá slúži na spustenie servisu. Dôležité je pridať do súboru AndroidManifest tento BroadcastReceiver, ako je znázornené na obrázku.

```
<receiver android:name=".MyBroadcastReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

Obrázok 38. MyBroadcastReceiver v AndroidManifest súbore[autor]

4.2 Návrh informačného systému

Pred tvorbou aplikácie je potrebné po analýze a výbere programovacích jazykov a operačného systému prejsť k samotnému návrhu informačného systému. Ten pozostáva z viacerých častí:

- **OCR engine** – je technológia na optické rozpoznávanie znakov, ktorá umožňuje konvertovať rôzne typy dokumentov, napríklad nasnímané fotky mobilným telefónom. Činnosť skenera spočíva v tom, že vytvorí snímku dokumentu, ktorá je len

súborom čiernych a bielych alebo farebných bodov (tzv. rastrový obraz). Na to aby sa mohli používať údaje z rôznych dokumentov a snímok je potrebný softvér OCR, ktorý vyextrahuje písmená z obrázka, prevedie ich na slová a slová na vety, čím dochádza k úprave obsahu originálneho dokumentu alebo snímky. V mobilnej aplikácii využívam Google Mobile Vision API, ktorá je výkonná a spoľahlivá OCR knižnica. Taktiež sa táto knižnica používa pri detekcii tváre alebo môže slúžiť ako skener čiarových kódov. Na začiatku tvorby aplikácie som používal na rozpoznávanie textu knižnicu OCR Tesseract. Dôvodom výberu Mobile Vision API bolo najmä to, že Tesseract knižnica neposkytovala presné výsledky, mala problémy s čítaním jednoduchého textu, text z obrázka bol v nezrozumiteľnej podobe. Preto, aby aplikácia bola čo najviac nápomocná, som sa rozhodol používať technológiu Mobile Vision API. Pred používaním tejto knižnice musí byť pridaná do časti Dependency ďalšia závislosť, vid' na nasledujúcom obrázku.

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:28.0.0'  
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'  
    implementation 'com.google.android.gms:play-services-vision:17.0.2'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'com.android.support.test:runner:1.0.2'  
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
}
```

Obrázok 39. Pridanie Google Mobile Vision do projektu[autor]

Taktiež pred použitím OCR skenera je potrebné do súboru AndroidManifest.xml pridať meta-dáta pre knižnicu OCR. AndroidManifest je hlavným súborom Android aplikácie, pretože obsahuje všetky základné údaje o aplikácii, uvádzajú sa v ňom rôzne oprávnenia na používanie mobilného fotoaparátu, čítanie a zapisovanie do externej pamäte a mnoho ďalších. Pridanie meta-dát sa nachádza na nasledujúcom obrázku.

```
<meta-data  
    android:name="com.google.android.gms.vision.DEPENDENCIES"  
    android:value="ocr" />
```

Obrázok 40. Pridanie meta-dát do projektu[autor]

- **TextToSpeech** – slúži na vytvorenie hlasového výstupu. Potom, čo aplikácia odfoťí obrázok a prekonvertuje ho na text, používateľ aplikácie si môže vypočúť daný text z obrázka. Nie je potrebné pridať žiadnu dodatočnú knižnicu, pretože sa používa knižnica od Androidu, ktorej celý názov je android.speech.tts.TextToSpeech. Hlasový výstup je

možné vyprodukovať v rôznych jazykoch. Ja využívam náš slovenský jazyk. Na tomto obrázku sa nachádza ukážka toho, ako vyzerá nastavovanie slovenského jazyka.

```
Locale loc = new Locale ( language: "sk", country: "SK");
textToSpeech.setLanguage(loc);
btnRead.setEnabled(true);
TextToSpeechFunction();
```

Obrázok 42. TextToSpeech nastavenie slovenského jazyka[autor]

Medzi dôležité konštanty, ktoré používa TextToSpeech patrí: QUEUE_FLASH, SUCCESS, ERROR A ERROR_NETWORK. Pre používanie hlasového výstupu v mojej aplikácii, je potrebné implementovať TextToSpeech.OnInitListener. Ide o rozhranie, pomocou ktorého budú v projekte implementované všetky dôležité metódy pre vytvorenie hlasového výstupu. Ide o metódy, ktoré sú volané pri spustení tejto aktivity, pri jej ukončení alebo počas jej behu. To, čo sa má vykonať pri spustení tejto aktivity alebo pri jej ukončení sa definuje v jednotlivých metódach, ktoré boli implementované z rozhrania TextToSpeech.OnInitListener.

- **rozpoznávanie hlasu** – slúži na to, aby používateľ aplikácie, ktorý má zhoršený zrak alebo rôzne poruchy zraku, nemusel pre čítanie vizitky na dverách lekára, textov nachádzajúcich sa v novinách, na plagátoch a podobne, stláčať nejaké tlačidlo, ale aby pomocou svojho hlasu dokázal vyvolať aktivitu na odfotenie obrázka, a potom tiež vyvolať aktivitu, ktorou sa spustí hlasový výstup. V mojej aplikácii využívam technológiu od Androidu SpeechRecognizer, ktorý slúži na rozpoznávanie hlasu. Pre používanie tejto technológie je potrebné implementovať rozhranie RecognitionListener. Toto rozhranie obsahuje viacero metód, z ktorých najdôležitejšia je metóda onResults. V tejto metóde sú definované základné hlasové povely, s ktorými sa bude v aplikácii pracovať. RecognitionListener bude hľadať zhodu medzi tým čo povieme, a tým čo uvedieme v metóde onResults. Ak dôjde k zhode, čiže podmienka bude pravdivá, vykoná sa príslušná aktivita, ktorá je definovaná vo vnútri podmienky.

- **Background service** – slúži na to, aby aplikácia mohla pracovať na pozadí. Vo všeobecnosti je servis aplikačný komponent, ktorý môže vykonávať dlhodobé operácie na pozadí a neposkytuje používateľské rozhranie. Moja aplikácia rovnako ako Facebook, či Instagram bude bežať na pozadí, aby neustále mohol používateľ aplikácie s ňou komunikovať, a zadávať hlasové povely na vykonanie aktivít. V prípade, že používateľ spustí inú aplikáciu, servis beží stále na pozadí. Servis môže dokonca spracovávať sieťové

transakcie, prehrávať hudbu, vykonávať vstupno-výstupné operácie so súbormi, a to všetko na pozadí. Na používanie servisu je potrebné ho deklarovať v súbore `AndroidManifest.xml`. Trieda, ktorá má byť servisom, musí dediť od triedy `Service`. Je potrebné implementovať viaceré metódy, v ktorých sa určí, čo sa vykoná pri spustení servisu, pri jeho ukončení, a tiež metódu, ktorá slúži na vzdialené volanie procedúr (RPC).

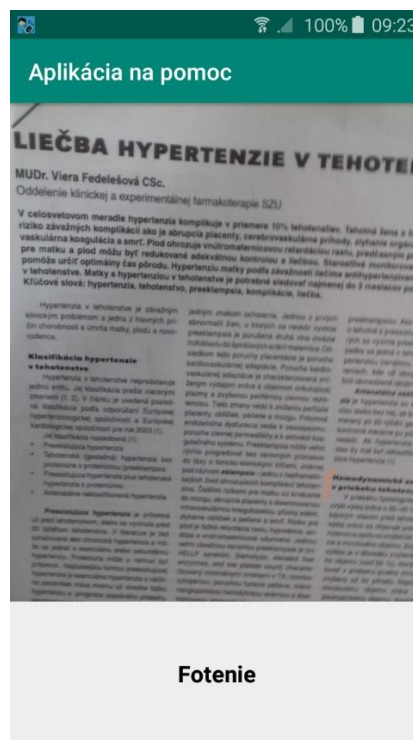
- **Ukladanie fotiek a textu** – po odfotení snímky používateľom aplikácie, sa automaticky obrázok uloží do adresára priamo v mobilnom zariadení, a taktiež prekonvertovaný text z obrázka sa uloží do textového súboru v adresári mobilného zariadenia. Táto funkcionality je dôležitá z pohľadu toho, že používateľmi aplikácie budú ľudia so zhoršenými zrakovými vlastnosťami, a keďže môžu nastať prípady, že sa im nepodarí dobre odfotiť to, čo chcú prečítať alebo OCR poskytne zlé výsledky, potom môžu tento obrázok alebo text ukázať ľuďom bez zrakového postihnutia, aby im prečítali čo je tam uvedené. Výhodou je fakt, že fotky aj text, ktoré sa ukladajú majú názov poskladaný z aktuálneho dátumu a času. Človek bez zrakového postihnutia potom môže na základe času, ktorý mu používateľ aplikácie povie, pozrieť na túto fotografiu alebo text, ktorý sa na nej nachádza. Fotky a texty sú ukladané v poradí od najnovších po najstaršie. Typickým príkladom je situácia, keď človek so zhoršenými vlastnosťami príde domov od lekára, niekto sa ho opýta na priezvisko lekára, a aplikácia mu dala zlý výsledok, tak mu povie nech sa pozrie do adresára v telefóne, a nech vyhladáva čas v ktorom bola odfotená vizitka lekára.

4.3 Používateľské rozhranie

Keďže samotná mobilná aplikácia je vytvorená pre používateľa so zrakovým postihnutím, používateľské rozhranie nezohráva dôležitú úlohu. Väčšiu prioritu má teda samotná funkcionality aplikácie či ponúka stanovené výsledky, či dostatočne rýchlo reaguje na požiadavky používateľa. Pri spustení aplikácie obsahuje používateľské rozhranie úvodnú obrazovku s inštrukciami ku mojej aplikácii a s dvoma tlačidlami pre spustenie a vypnutie. Používateľské rozhranie sa mení vtedy, keď používateľ zadá hlasový vstup. Pri zadaní kľúčového slova pre fotenie bude zavolané zmenené používateľské rozhranie, ale pri zavolaní aktivity na prečítanie textu, nebude viditeľné žiadne používateľské rozhranie, keďže používateľ si len vypočuje text, ktorý bol na obrázku. Obidva prípady sú znázornené na týchto obrázkoch.



Obrázok 43. Používateľské rozhranie aplikácie[autor]



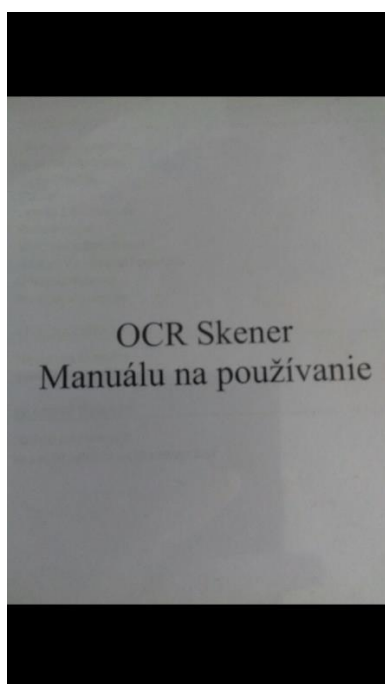
Obrázok 44. Fotenie[autor]

4.4 Analýza výsledkov práce

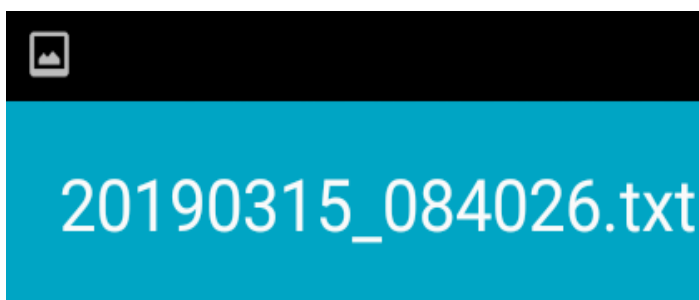
Táto časť je zameraná na zhodnotenie výsledkov činnosti mojej aplikácie. Obsahuje zhodnotenie každej časti funkcionality mobilnej aplikácie. Analýza výsledkov má veľký význam, pretože poskytuje informácie, ktoré môžu byť nápomocné pri ďalšom vyvíjaní a zlepšovaní aplikácie. Tvorba aplikácie nekončí jej vytvorením, ale neustále pokračuje pri hľadaní vylepšenia a zlepšenia funkcionality. Pri vývoji softvéru platí, ak existujú nové možnosti, ktoré prinesú zlepšenie výsledkov hoci len o jedno percento, treba ich použiť a aplikovať. Dôležité je to aj z pohľadu toho, ak chceme držať krok s konkurenciou, chceme ju predbehnúť, tak neustále musíme zlepšovať existujúce možnosti.

OCR – pri optickom rozpoznávaní znakov som sa rozhodol medzi knižnicou Tesseract a Mobile Vision API. Dôležité bolo vyskúšať obe knižnice, aby som zistil, ktorá z týchto knižníc poskytuje lepšie výsledky. Po preskúmaní výsledkov som dospel k záveru, že použitie Mobile Vision API prináša lepšie a kvalitnejšie výsledky. Výhodou bolo najmä to, že nie je nutné, aby používateľ musel klikať na tlačidlo fotenia alebo aby musel vykonávať dodatočnú interakciu. Všetko sa vykoná automaticky vďaka nastaveniu AutoFocusu na pravdivú hodnotu. Používateľ po odfovení počuje zvuk, ktorý signalizuje, že obrázok bol odfotený, a on môže ísť späť na domovskú stránku telefónu. Výsledky použitia

tejto OCR technológii vo väčšine prípadov poskytovali stopercentné výsledky. Niekedy sa stalo, že došlo k zámene slova alebo chybe v jednom slove, ale to nemenilo nič na kontexte celého obsahu, čiže používateľ aj napriek malej chybe vedel, čo tento text znamená, respektíve čo prináša táto správa. Jednotlivé výsledky, ktoré boli použité na testovanie aplikácie sa nachádzajú v prílohe mojej bakalárskej práce. Uviedol som tam všetky moje pokusy, pomocou ktorých som sledoval, ako pracuje OCR technológia v mojej aplikácii. Jednoduchý príklad toho, ako OCR knižnica Mobile Vision API prekonvertuje text z obrázka sa nachádza na nasledujúcich obrázkoch.



Obrázok 45. Vstupný
obrázok[autor]



OCR Skener
Manuálu na používanie

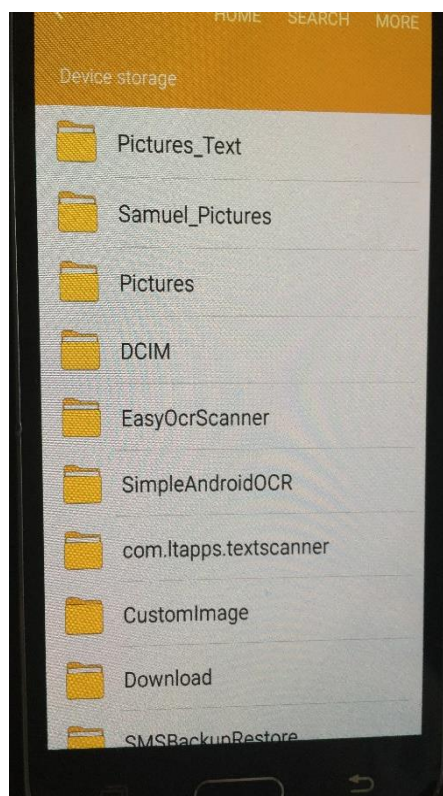
Obrázok 46. Prekonvertovaný text pomocou OCR[autor]

Na prvom obrázku sa nachádza text, ktorý bol odfotený prostredníctvom mojej aplikácie. Druhý obrázok je textový súbor, do ktorého sa ukladá prekonvertovaný text pomocou OCR technológii. Na takomto príklade je možné vidieť, že Mobile Vision API prináša skvelé výsledky. Používatelia tejto aplikácie ju budú používať na riešenie podobných problémov, napríklad keď nebudú schopní prečítať vizitku na dverách lekára alebo text nachádzajúci sa na plagáte.

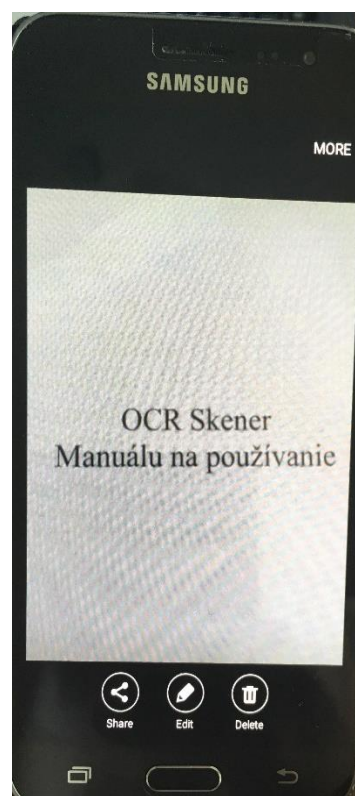
Text To Speech – na vytvorenie hlasového výstupu som použil knižnicu od Androidu, ktorej celý názov je `android.speech.tts.TextToSpeech`. Veľkou výhodou je možnosť nastavenia jazyka výstupu, takže som si vybral slovenský jazyk. Táto zabudovaná technológia bola výborným riešením, pretože jej výsledky sú stopercentné. Veľmi významným prvkom je možnosť určovania čo sa vykoná keď aplikácia dohovori text.

Spočiatku vznikali problémy s určením konca hlasového výstupu. Nakoniec po preštudovaní Android dokumentácie som zistil, že je potrebné nastaviť UtteranceProgressListener. Tento listener má v sebe zabudovanú funkciu onDone, v ktorej je určené to, čo sa vykoná po skončení hlasového výstupu.

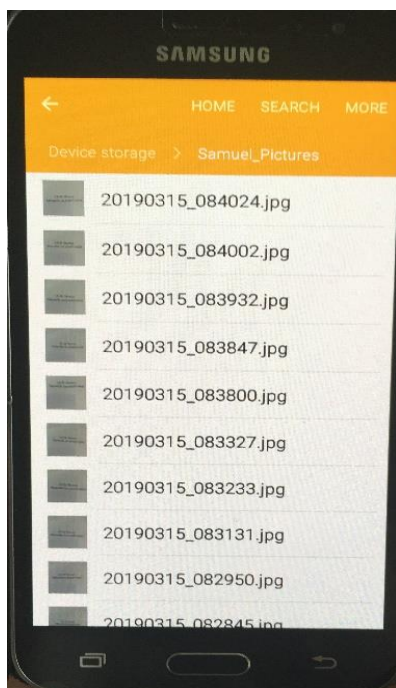
Ukladanie do pamäte – pri ukladaní do pamäte sa využíva algoritmus, ktorý je všeobecne známy a poskytuje správne výsledky. Keďže sa ukladá aj odfotený obrázok, aj text z neho, vytvoril som dva adresáre. Prvý je na ukladanie obrázkov a má názov Samuel_Pictures, druhý je určený na ukladanie textov, pričom jeho názov je Pictures_Text. Pri oboch typoch sa využíva ako názov systémový čas. Obedva typy súborov sa teda líšia iba svojím formátom. Zatiaľ čo obrázky sú ukladané vo formáte jpg, tak texty z obrázkov sa ukladajú vo formáte txt. Používateľ aplikácie, ktorý bude neskôr v kontakte s človekom bez poruchy zraku, mu len povie, aby sa pozrel do adresára, kde sú uložené odfotené fotky a kde sú uložené texty, a to len v prípade, ak si nie je istý tým, čo odfoťil alebo aplikácia mu poskytla zlé výsledky. Túto funkčnosť ukladania do pamäte mobilného zariadenia nie je potrebné vylepšovať alebo zdokonaľovať, keďže poskytuje to, čo od nej očakávam. Príklad ukladania obrázku a textu do pamäte je znázornený na obrázkoch.



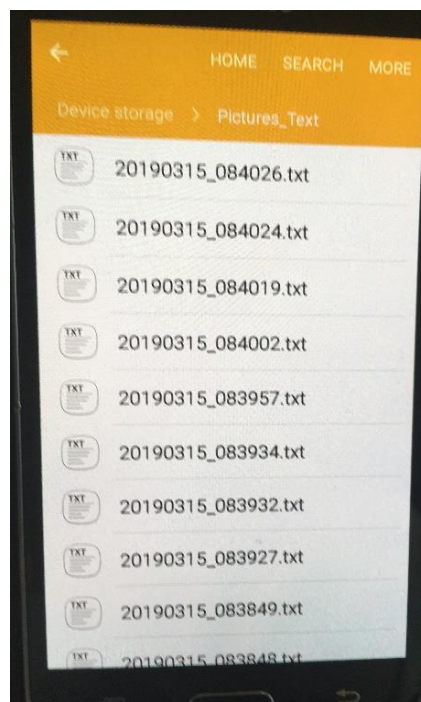
Obrázok 47. Ukážka adresárov[autor]



Obrázok 48. Odfotený obrázok[autor]



Obrázok 49. Uloženie obrázka[autor]



Obrázok 50. Uloženie textu[autor]

Hlasové príkazy – vytváranie algoritmu na zadávanie hlasových príkazov predstavovalo najťažšiu časť z hľadiska tvorby aplikácie. Pre zabezpečenie plynulej možnosti zadávania hlasových príkazov bolo potrebné nastaviť AudioManager tak, aby nedochádzalo k pípaniu ako to bolo na začiatku tvorby aplikácie. Pre zabezpečenie toho, aby zariadenie počas nahrávania zvuku nepípalo a nevibrovalo môžu jednotlivé funkcie AudioManager, ktoré sú na tomto obrázku.

```
mAudioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
mAudioManager.setStreamMute(AudioManager.STREAM_NOTIFICATION, state: true);
mAudioManager.setStreamMute(AudioManager.STREAM_SYSTEM, state: true);
mAudioManager.setStreamMute(AudioManager.STREAM_MUSIC, state: true);
mAudioManager.adjustVolume(AudioManager.STREAM_NOTIFICATION, AudioManager.ADJUST_MUTE);
mAudioManager.adjustVolume(AudioManager.STREAM_SYSTEM, AudioManager.ADJUST_MUTE);
mAudioManager.adjustVolume(AudioManager.STREAM_MUSIC, AudioManager.ADJUST_MUTE);
mSpeechRecognizer = SpeechRecognizer.createSpeechRecognizer(this);
mSpeechRecognizer.setRecognitionListener(new SpeechRecognitionListener());
mSpeechRecognizerIntent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
mSpeechRecognizerIntent.putExtra(RecognizerIntent.EXTRA_CALLING_PACKAGE, this.getPackageName());
```

Obrázok 51. AudioManager a jeho funkcie[autor]

Výsledky zadávania hlasových príkazov sú takmer stopercentné, problémy vznikajú až vtedy, keď je v blízkosti nadmerný hluk alebo dochádza k zlej výslovnosti kľúčových slov. Takéto zadávanie hlasových príkazov je prioritne orientované pre ľudí s poruchami zraku. Mojm cieľom bolo čo najviac uspokojiť požiadavky týchto ľudí. Som si vedomý toho, že nie sú schopní stláčať nejaké tlačidlá alebo prehľadávať celé menu nachádzajúce sa v mobilnom zariadení, preto som sa rozhodol pre zadávanie hlasových príkazov.

Background Service – k rozhodnutia toho, že moja aplikácia bude bežať na pozadí ma viedla inšpirácia v podobe aplikácii ako sú Facebook či Instagram. Používateľ môže popri práci s inými aplikáciami neustále zadať hlasový príkaz a aplikácia sa dostane do popredia. Aplikácia na pozadí funguje tak, ako som očakával a pre používateľov umožňuje možnosť pracovať s ňou kedykoľvek ju budú potrebovať.

4.5 Tvorba manuálu

Pred tým ako sa aplikácia dostane k ľuďom so zrakovým postihnutím je potrebné vytvoriť aj manuál pre správne používanie aplikácie. Vo všeobecnosti platí, že človek pri prvom styku s nejakou novou technológiou, aplikáciou, zariadením a podobne musí byť oboznámený s tým, ako aplikácia funguje, čo potrebuje stlačiť, zadať a mnoho ďalších činností, ktoré sa nachádzajú v manuáli aplikácie. Pre moju mobilnú aplikáciu bol vytvorený používateľský manuál, po prečítaní ktorého, bude práca s touto aplikáciou oveľa jednoduchšia. Tento manuál obsahuje popis inštalácie a základnú funkcionality mobilnej aplikácie pre nevidiacich na čítanie.

5 Diskusia a možnosti vylepšenia v budúcnosti

Posledná kapitola je zameraná na možnosti vylepšenia do budúcnosti. Nachádzajú sa v nej úvahy o tom, ktoré technológie by mohli byť vylepšené, nahradené alebo odstránené z aplikácie. Je dôležité hľadiť do budúcnosti, a snažiť sa vytvoriť si plán zlepšenia aplikácie. Keďže moderný svet sa neustále zlepšuje a zdokonaľuje, treba prihliadať aj na zlepšenie vlastnej aplikácie. Najhoršia vec, ktorá môže nastať je, že aplikácia bude stagnovať, nebude sa posúvať dopredu, čiže nebude prinášať žiadnu pridanú hodnotu. Preto je dôležité vytvoriť si plán, ktorý bude obsahovať možnosti zlepšenia do budúcnosti.

5.1 Možnosti vylepšenia aplikácie

Pri možnostiach vylepšenia aplikácie sa berú do úvahy jednotlivé funkcionality aplikácie. Predpokladám, že vylepšenia funkcionalít by viedli k celkovému zvýšeniu pridanej hodnoty aplikácie. V konkurenčnom boji s aplikáciami podobného typu je dôležité pridávať vždy nové vylepšenia, ktoré môžu zlepšiť samotný chod vlastnej aplikácie. Za vylepšenie sa považujú aj malé zmeny, ktoré v konečnom dôsledku povedú k samotnému zlepšeniu aplikácie. Zlepšenia sa môžu objaviť v jednotlivých funkcionalitách mobilnej aplikácie.

OCR – Na optické rozpoznávanie textu z obrázka som použil knižnicu Mobile Vision API. Je to riešenie od spoločnosti Google, ktoré je dostupné zdarma. Táto knižnica má schopnosť rozpoznávať objekty na fotkách a videách. Pomocou tejto knižnice sú rozpoznávané tvary, čiarové kódy a text. Hoci sú k dispozícii viaceré cloudové riešenia, ja som si vybral túto knižnicu, ktorá nepoužíva cloud. Hlavnou výhodou je to, že toto riešenie nevyžaduje žiadne internetové pripojenie. Veľmi nápomocným sa javí fakt, že dokáže rozpoznať text v akomkoľvek latinskom jazyku. Najväčšou limitáciou tejto knižnice je to, že detektor textu číta text náhodne. Existuje veľká pravdepodobnosť, že vráti text v odlišnom poradí ako vyzerá text na obrázku. Treba tiež poznamenať, že detektor textu nie je vždy stopercentný. Do budúcnosti si myslím, že by sa zvýšila pridaná hodnota aplikácie, ak by sa nahradila existujúca knižnica lepšou OCR technológiou, ktorá by poskytovala ešte lepšie výsledky. Dôležité pre človeka so zrakovým postihnutím je skutočnosť, že OCR technológia musí poskytovať stopercentné výsledky. Nesmie dochádzať k problémom s rozpoznávaním textu, poradím slov a podobne. Preto si myslím, že by v budúcnosti bolo vhodné a pre moju aplikáciu rozumné vybrať cloudové riešenie alebo rozhodnúť sa pre technológiu využívajúcu neurónové siete.

Text To Speech – Hlasový výstup prečítaného textu bol realizovaný prostredníctvom zabudovanej knižnice v Androide. Aj napriek výborným poskytovaným výsledkom je potrebné túto funkcionality vylepšiť. Niekedy sa stáva, že používateľ nepochopí celý obsah hlasového výstupu, nestihne zaregistrovať celý obsah správy. Vtedy by bolo vhodné pridať funkcionality, ktorá zabezpečí pomalší hlasový výstup, respektíve automatické zopakovanie správy. Takáto funkcionality by bola veľmi nápomocná, pretože najmä starší ľudia môžu mať problémy pri zachytení hlasových výstupov. Ďalším vylepšením v tejto oblasti by mohlo byť automatické zvyšovanie hlasitosti, ak je v blízkosti veľký hluk a používateľ aplikácie by mal problémy s prijatím a pochopením správy.

Ukladanie do pamäte – Ukladanie obrázkov a prekonvertovaného textu z obrázka sa realizuje prostredníctvom jednoduchého algoritmu. V tejto oblasti nie je potrebné žiadne vylepšenie, pretože ak by sme sa na ukladanie pozreli z pohľadu toho, či ukladanie prebehlo úspešne, výsledky by boli vždy stopercentné. To, že sa ukladajú obrázky a text do samostatných adresárov je prospešné najmä kvôli lepšej orientácii medzi jednotlivými súbormi. Ak si niekto bude chcieť pozrieť len odfotený obrázok, pôjde do adresára v ktorom ukladáme obrázky a ak niekto zasa bude chcieť vidieť len text, pôjde do adresára s uloženými prekonvertovanými textami. Takisto názvy jednotlivých súborov sa pomenovávajú podľa systémového času, kedy bol obrázok odfotený, čiže ani v tejto oblasti nie sú potrebné zmeny.

Hlasové príkazy – Pri zadávaní hlasových príkazov, by bolo potrebné ich rozšírenie, aby zahŕňalo široké spektrum pôsobnosti. Aplikácia by bola viac nápomocná, ak by umožňovala vykonávanie viacerých aktivít prostredníctvom hlasu. Toto rozšírenie by obsahovalo hlasové príkazy na telefonovanie blízkej osobe, vytvorenie hlasovej správy a tiež na spustenie GPS navigácie, ktoré by pomohlo pri lepšej orientácii človeku so zrakovým postihnutím. Vylepšenie a rozšírenie hlasových príkazov by viedlo k viacerým možnostiam práce s aplikáciou. Napríklad používateľa mojej aplikácie by ju nevyužívali len pri čítaní textov z obrázka, ale taktiež by ju využívali aj ako pomôcku, pomocou ktorej by sa mohli dostať k danému obrázku. Veľmi nápomocným by bolo zavedenie hlasových príkazov pre filtrovanie textov z obrázkov. Ak si používateľ nechce vypočuť len najnovší prekonvertovaný text, tak zadá kľúčové slovo pre filtrovanie dopredu a dozadu, a môže tak ľahko filtrovať medzi jednotlivými výsledkami. Taktiež rozumným vylepšením by bolo aj zavedenie vyhľadávania na základe povedaného slova. To znamená, že ak používateľ odfotil obrázok, na ktorom bola vizitka lekára, tak môže zadať hlasový príkaz pre vyhľadanie slova lekár a ako výsledok mu bude prečítaný text, ktorý obsahuje slovo lekár.

5.2 Vylepšenie pridaním novej funkcionality

V tejto časti sú analyzované nové možnosti, ktoré by mohli byť pridané do mojej aplikácie a vďaka ktorým by sa mohla zvyšovať celková pridaná hodnota aplikácie do budúcnosti. Dôležitým vylepšením je ukladanie obrázkov a prekonvertovaných textov z nich do databázy. Všetky odfotené obrázky a texty, ktoré sa získali prostredníctvom OCR technológie sa automaticky odošlú do databázy. To by umožňovalo, aby nejaká blízka osoba používateľa tejto aplikácie mohla kdekoľvek kontrolovať to, čo používateľ v ten ktorý deň odfotil, respektíve čo potreboval prečítať.

Ďalším rozšírením by bolo zamerania sa na používateľov, ktorý majú operačný systém iOS. Zavedenie kompatibility aplikácie pre používateľov s operačným systémom iOS by viedlo k pokrytiu takmer celého trhu potenciálnych používateľov. Zameranie sa len na jeden operačný systém vedie k strate potenciálnych možných používateľov aplikácie. Mojim cieľom je pomôcť ľuďom so zrakovým postihnutím, čiže treba brať do úvahy aj rozšírenie o ďalší operačný systém.

Významným vylepšením aplikácie by bola možnosť implementovať hlasového asistenta Google. To by viedlo k možnosti ovládať celé mobilné zariadenie, a tiež moju aplikáciu. Používateľ by sa cítil oveľa komfortnejšie, keby mohol pomocou hlasu ovládať celý systém, nielen túto aplikáciu.

Záver

Na začiatku bakalárskej práce bolo potrebné preštudovať si celú problematiku ľudského oka, jeho funkčnosť a špecifikáciu. Boli analyzované jednotlivé druhy postihnutí, pričom hlavný dôraz sa kládol na zrakové postihnutie. V samostatných podkapitolách boli analyzované jednotlivé typy zrakových postihnutí, ich príčiny a tiež typy zrakových porúch. Taktiež boli analyzované pomôcky a prostriedky, ktoré uľahčujú, respektíve napomáhajú ľuďom so zrakovým postihnutím. Najdôležitejšou časťou bola analýza existujúcich mobilných aplikácií, keďže mojím cieľom bolo vytvorenie mobilnej aplikácie pre nevidiacich. Na konci kapitoly som poskytol zhrnutie jednotlivých pomôcok vo forme diagramu, ktorý poskytuje celkový pohľad na jednotlivé pomôcky a prostriedky určené pre nevidiacich a ľudí so zrakovým postihnutím.

V samostatnej kapitole som venoval pozornosť metódam a nástrojom tvorby informačných systémov. Musel som vykonať výber nástrojov tvorby informačného systému, ktorý sa rozčlenil do troch častí. Prvou bol výber mobilného zariadenia, pre ktorý budem vyvíjať mobilnú aplikáciu. Po zvážení všetkých výhod a nevýhod som dospel k záveru, že si vyberiem operačný systém, postavený na jadre Linuxu a to Android. Pri ďalšom výbere som sa rozhodoval medzi programovacími jazykmi. Porovnal som jazyk Java s C++ a C#, a po preskúmaní jednotlivých vlastností som vybral programovací jazyk Java. Programovanie Java je v súčasnosti stále na popredných priečkach pri porovnávaní jazykov, ktoré potvrdil aj priložený graf od spoločnosti Tiobe.

V poradí štvrtá a najdôležitejšia kapitola odzrkadľuje celý proces tvorby mojej aplikácie. Na začiatku som sa venoval algoritmu činnosti zariadenia, ktorý pozostáva z viacerých na seba nadväzujúcich aktivít. Každá aktivita je zodpovedná za vykonanie určitej akcie a spolu vytvárajú komplexnú aplikáciu. Pri návrhu informačného systému boli popísané jednotlivé technológie, ktoré boli použité. Najprv som popísal OCR engine, ktorý využívam a poskytol som základné princípy práce s ním. Potom som opísal TextToSpeech technológiu, ktorú som využil v mojej aplikácii. Táto aplikácia pracuje na pozadí, čiže bolo potrebné implementovať Background service, aby aplikácia stále bežala. Jednotlivé nasnímané fotky a prekonvertované texty z nich sa ukladajú do pamäte telefónu, do samostatných adresárov. Použil som algoritmus na ukladanie fotiek a textov a taktiež som musel vytvoriť metódu na pomenovanie jednotlivých súborov, aby ich názov tvoril celý aktuálny dátum a čas. Pozornosť som venoval aj zadávaniu hlasových príkazov, musel som najprv deklarovať a inicializovať AudioManager, ktorý je zodpovedný za prácu

s nastavovaním hlasitosti a zvonením, aby pracoval v súlade s mojimi očakávaniami. V krátkosti bol poskytnutý náhľad používateľského rozhrania mobilnej aplikácie, priložil som aj obrázky, ako to v realite vyzerá. Jadrom tejto kapitoly je zhrnutie analýzy výsledkov práce. Jednotlivé použitia aplikácie sa nachádzajú v prílohe bakalárskej práci, no stručne som charakterizoval aj prácu jednotlivých použitých technológií. Nakoniec bol vytvorený aj manuál k aplikácii, ktorý slúži na poskytnutie pomoci pri prvom styku s aplikáciou a je uložený v prílohe k bakalárskej práci.

Piata kapitola je skôr úvahová, poskytuje myšlienky a nápady vylepšenia aplikácie. Je veľmi dôležité z pohľadu správneho chodu aplikácie, neustáleho vylepšovania, pridávania nových funkcionalít a podobne. Na to, aby aplikácia mohla napredovať a držať krok s konkurenciou je potrebné ju vylepšovať a prinášať nové možnosti použitia.

Výsledok mojej práce predstavuje informačný systém v podobe mobilnej aplikácie, ktorá slúži na pomoc ľuďom so zrakovým postihnutím na čítanie, využívajúc pri tom OCR engine na rozpoznanie textu z obrázka a text vracia v podobe hlasového výstupu. Celý proces chodu aplikácie beží na pozadí, čo umožňuje jednoduchú a neustálu prácu s aplikáciou. Táto bakalárska práca odzrkadľuje proces tvorby mobilne aplikácie využívajúc kooperáciu viacerých funkcionalít do jednej aplikácie. Mojim zámerom bolo aj ukázať, čo všetko sa dá použiť v mobilnej aplikácii, aby sa čo najviac prispôsobila požiadavkám skupiny používateľov, v tomto prípade ľuďom s poruchami zraku. Myslím si a verím, že moja aplikácia pomôže čo najviac ľuďom a odmenou pre mňa bude to, že s ňou budú spokojní.

Zoznam použitej literatúry

- [1] The Government of the Hong Kong Special Administrative Region. Mobile Application Accessibility Handbook, July 2017. Dostupné na internete: https://www.ogcio.gov.hk/en/our_work/community/web_mobileapp_accessibility/promulgating_resources/maahandbook/doc/mobile_app_handbook.pdf
- [2] doc. Mgr. MUDr. Furdová, Alena, PhD., MPH, MSc. Prečo oko vidí a čo nevidí?. [cit. 2018-29-09]. Dostupné na internete: http://www.dukonline.sk/tmp/asset_cache/link/0000018698/6.%20predn%C3%A1%C5%A1ka.pdf
- [3] World Health Organization. Visual impairment and blindness. [cit. 2018-29-09]. Dostupné na internete: <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- [4] Albertová, Brigita. Vplyv aktívnej pohybovej činnosti na rozvoj priestorovej orientácie nevidiacich: rigorózna práca. Školiteľ: Ing. Jaroslav Kultán, PhD. Ružomberok, 2017. 151s.
- [5] Jašková, Ľudmila. Zrakovo postihnutí a práca so základným softvérom. IKT vo vzdelávaní zdravotne postihnutých. [cit. 2018-29-09]. Dostupné na internete: <http://edi.fmph.uniba.sk/~jaskova/IKTH/tema02/tema02.html>
- [6] Hamadová, Petra, Kvetoňová, Lea, Nováková Zita. Oftalmopedie, Texty k distančnému vzdelávaní. 2.vyd., Brno: PAIDO, 2007. 125s. ISBN 978-80-7315-159-1
- [7] Keblová, A. Kompenzační pomůcky pro zrakovo postižené žáky ZŠ. 2.vyd., Praha: SEPTIMA, 1999. 28s. ISBN 80-7216-104-0
- [8] Mamojka, Branislav, Kobolková, Martina. Informačné zdroje na internete môžu byť prínosom aj pre nevidiacich a slabozrakých ľudí. 2001. [cit. 2018-29-09]. Dostupné na internete: http://itlib.cvtisr.sk/archiv/2001/4/informacne-zdroje-na-internete-mozu-byt-prinosom-aj-pre-nevidiacich-a-slabozrakych-ludi.html?page_id=2209
- [9] Kompenzační pomůcky se zameraním pro zrakovo postižené. Fakulta informatiky Masarykovy Univerzity. 2003. [cit. 2018-28-09]. Dostupné na internete: <http://www.fi.muni.cz/usr/jkucera/pv109/2003/xsukova03.htm>
- [10] Matušeková, Ivana. Korekčné pomôcky pre slabozrakých: bakalárska práca. Školiteľ: MUDr. Radek Girgle. Brno, 2012. 50s.
- [11] Neto, Roberto, Fonseca, Nuno. Camera Reading For Blind People. 2004.
- [12] Pandey, Amrit. 7 Best Android OCR Apps to Scan Image to Text. [cit. 2018-10-28]. Dostupné na internete: <https://mashtips.com/android-ocr-scanner-apps/>

- [13] Williams, Elise. Top 5 Android PDF to Speech Converter Apps, [cit. 2018-10-28]. Dostupné na internete: <https://pdf.wondershare.com/pdf-knowledge/pdf-to-speech-android.html>
- [14] King, Bertel. 7 Siri Alternatives for Android: Google Assistant, Cortana, Alexa, and More. [cit. 2018-10-28]. Dostupné na internete: <https://www.makeuseof.com/tag/7-siri-alternatives-android-google-now-cortana/>
- [15] Kapusta, Matúš. 10 dôvodov, prečo je Android stále lepší ako iOS. [cit. 2018-10-29]. Dostupné na internete: <https://www.mojandroid.sk/10-dovodov-android-lepsi-ios/>
- [16] Friessen, Jeff. Learn Java for Android Development. 3.vyd., Apress, 2003. 780s. ISBN 978-1-4302-5723-3
- [17] C++ vs Java. [cit. 2018-11-19]. Dostupné na internete: <https://www.educba.com/c-plus-plus-vs-java/>
- [18] Java vs C#. [cit. 2018-11-19]. Dostupné na internete: <https://www.educba.com/java-vs-c-sharp/>
- [19] Tyflosonar Ray. [cit. 2019-03-19]. Dostupné na internete: <http://www.tyflocomp.sk/produkty/tyflosonar.html>
- [20] Penfriend. [cit. 2019-03-19]. Dostupné na internete: <http://www.tyflocomp.sk/produkty/penfriend.html>
- [21] Colortest. [cit. 2019-03-19]. Dostupné na internete: <http://www.tyflocomp.sk/produkty/colortest.html>
- [22] MAGic. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/magic>
- [23] SARA. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/sara-zariadenie-pre-skenovanie-a-citanie>
- [24] Finereader 10. [cit. 2019-03-19]. Dostupné na internete: <https://zive.azet.sk/clanok/48954/test-abbyy-finereader-10-home-edition-spolahlivy-prevod-na-text/>
- [25] ClearView+. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/elektronicke-citacie-lupy-tv-lupy/clearview>
- [26] Compact+. [cit. 2019-03-19]. Dostupné na internete: <https://uk.optelec.com/products/comp-wrld-compact.html>
- [27] FarView. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/elektronicke-citacie-lupy-tv-lupy/farview>

- [28] Sapphire. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/elektronicke-citacie-lupy-tv-lupy/sapphire>
- [29] JAWS. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/jaws>
- [30] Tomboy. [cit. 2019-03-19]. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/tomboy-daisy-flash-prehravac>
- [31] Transcend 860. [cit. 2019-03-19]. Dostupné na internete: <https://www.transcend-info.com/About/press/1340>
- [32] IRIS Mobile OCR SDK. [cit. 2019-03-19]. Dostupné na internete: <https://www.irislink.com/EN-SK/c1895/IRIS-Mobile-OCR-SDK.aspx>
- [33] ABBYY Real-Time Recognition SDK. [cit. 2019-03-29]. Dostupné na internete: <https://www.abbyy.com/en-gb/real-time-recognition-sdk/>
- [34] Smith, Ray, Google Inc. An Overview of the Tesseract OCR engine. [cit. 2019-03-29]. Dostupné na internete: <https://www.github.com/tesseract-ocr/docs/blob/master/tesseracticdar2007.pdf>
- [35] Mobile Vision API. [cit. 2019-03-29]. Dostupné na internete: <https://developers.google.com/vision/>

Obrázky:

- [36] Prierez oka. Dostupné na internete: <https://www.zdravie.sk/choroba/20824/degeneracia-oka>
- [37] Stojanová lupa, Digitálna USB lupa 200x, Turmon. Dostupné na internete: https://is.muni.cz/th/evrhb/Korekcne_pomocky_pre_slabozrakych_bc..pdf
- [38] Vybrané pomôcky pre nevidiacich. Dostupné na internete: www.tyflocmp.sk/produkty/produkty.html
- [39] JAWS. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/jaws>
- [40] Tomboy. Dostupné na internete: <http://tyflocomp.sk/produkty-php/specialne-pocitace-upravy-a-prislusenstvo/tomboy-daisy-flash-prehravac>
- [41] Transcend 860. Dostupné na internete: <https://www.transcend-info.com/About/press/1340>
- [42] Náhľad do aplikácie Google Keep. Dostupné na internete: https://medium.com/@chiphm_29386/google-keep-is-it-a-keeper-7b4a0c182ac4

- [43] Náhl'ad do aplikácie Google Googles. Dostupné na internete:
<https://theamericangenius.com/gadgets/smartphones-gadgets/google-goggles-launches-new-amazing-version-of-their-android-app/>
- [44] Náhl'ad do aplikácie Office Lens. Dostupné na internete:
<https://pcworld.com/article/2927332/microsofts-office-lens-scanner-app-for-android-leaves-beta.html>
- [45] Náhl'ad do aplikácie @Voice Aloud Reader. Dostupné na internete:
www.maxapk.com/2018/10/voice-aloud-reader-tts-reader-1545.html
- [46] Náhl'ad do aplikácie Amazon Alexa. Dostupné na internete:
<https://www.intellectdigest.in/amazon-alexa-app-india-34535/>
- [47] Náhl'ad do aplikácie Robin. Dostupné na internete:
<https://play.google.com/store/apps/details?id=com.magnifis.parking&hl=cs>
- [48] Graf populárnosti programovacích jazykov. Dostupné na internete:
<https://www.tiobe.com/tiobe-index/>
- [49] Porovnanie nákupov. Dostupné na internete:
https://www.gsmarena.com/global_smartphone_sales_flat_chinese_companies_gain_ground-news-34532.php

Zoznam príloh

Príloha č.1: Manuál na používanie (aj na CD)

Príloha č.2: Výsledky práce OCR (na CD)

Príloha č.3: Zdrojový kód aplikácie (na CD)

Príloha č.4: Návod na spustenie aplikácie z vývojového prostredia (na CD)

Company	Q3 2018 sales (in thousands)	Q3 2018 Market share	Q3 2017 sales (in thousands)	Q3 2017 Market share
Samsung	73,360.1	18.9	85,605.3	22.3
Huawei	52,218.4	13.4	36,501.8	9.5
Apple	45,746.6	11.8	45,441.9	11.8
Xiaomi	33,219.7	8.5	26,853.2	7.0
Oppo	30,563.4	7.9	29,449.2	7.7
Others	153,960.0	39.6	159,742.0	41.6
Total	389,068.2	100.0	383,593.4	100.0

Obrázok 52. Porovnanie nákupov smartfónov[49]